



TIC FORUM²⁰¹⁹
IN MOTION

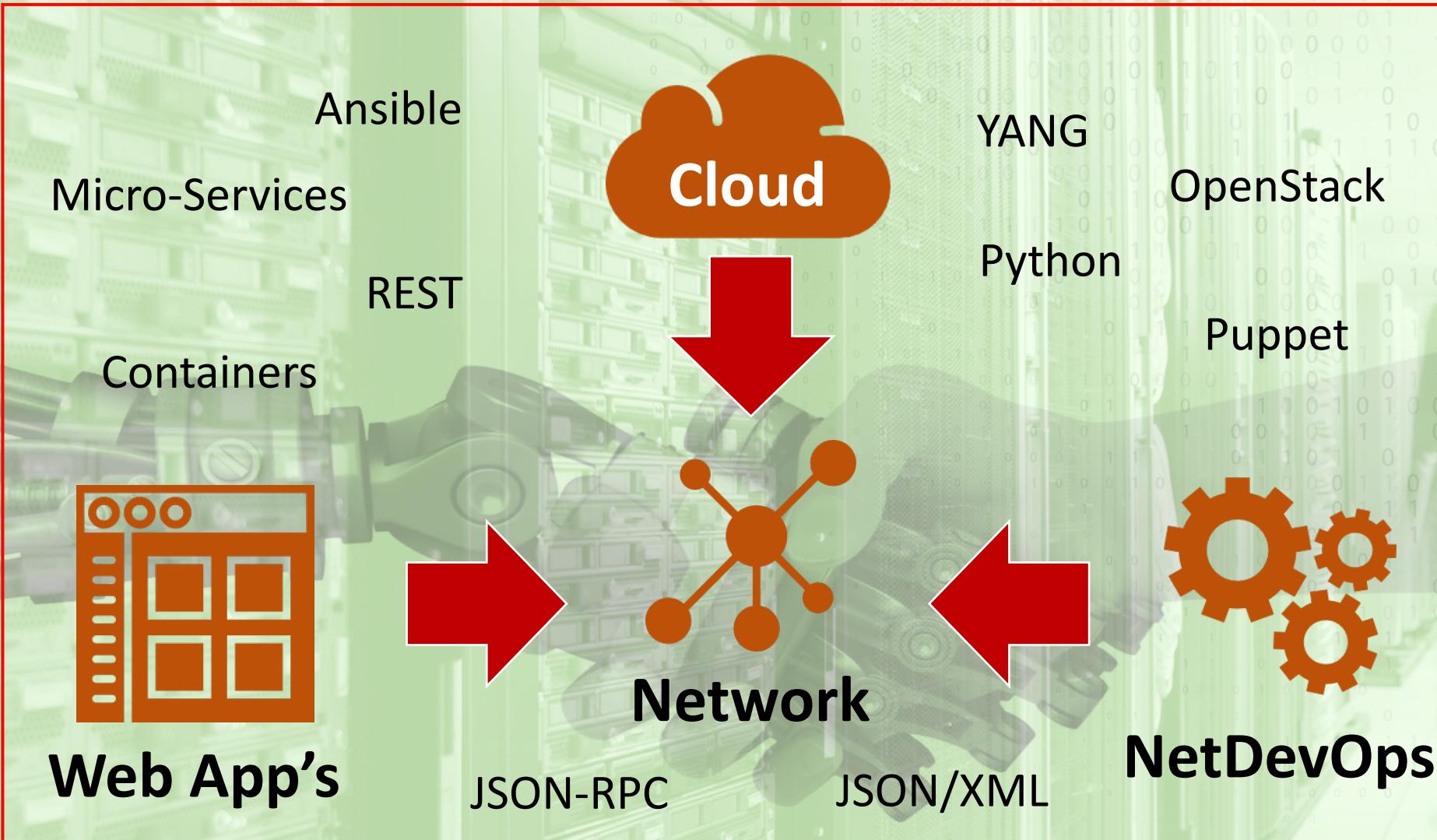
 LOGICALIS

Network Programmability & Automation

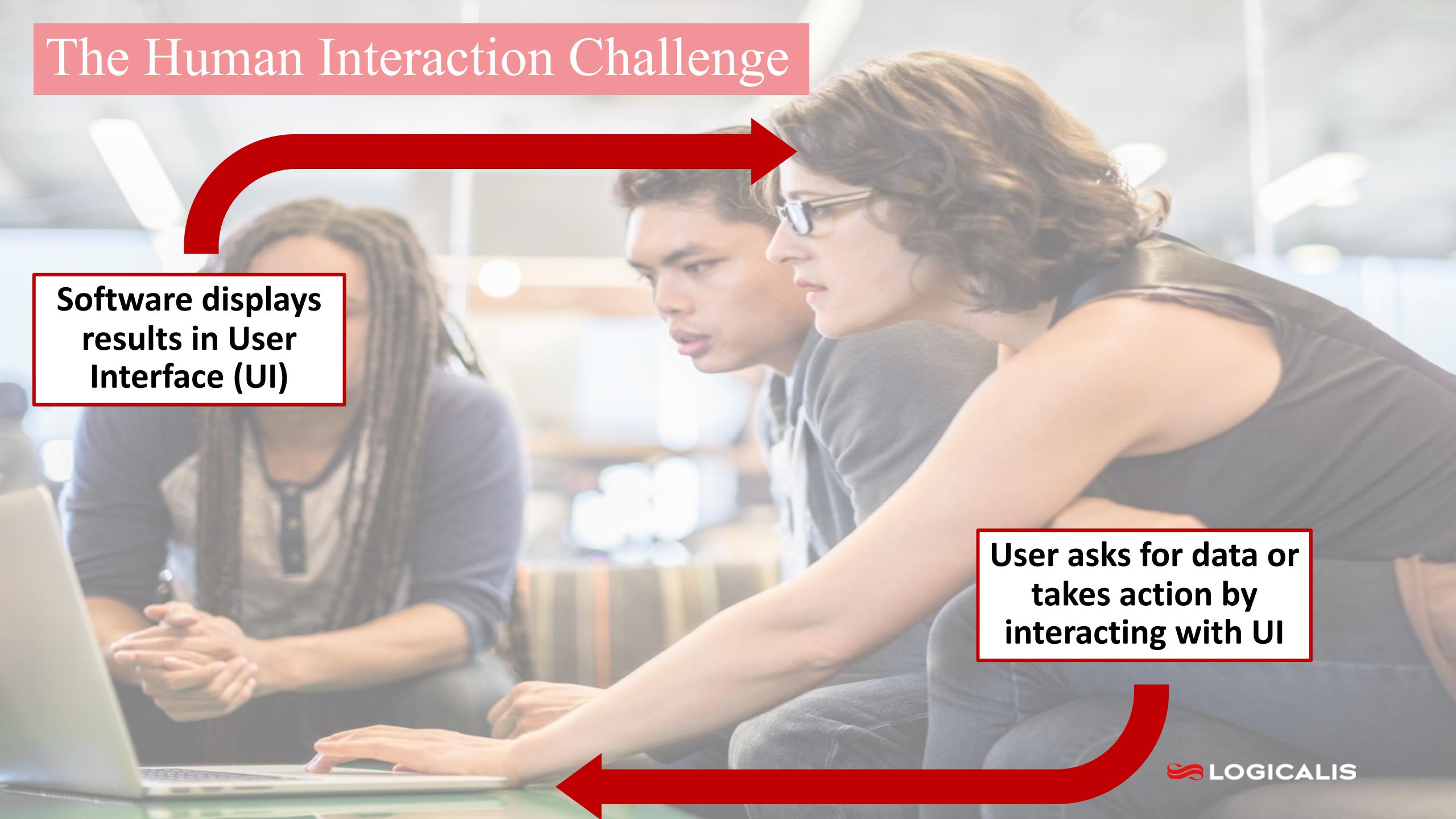
#Ed Scrimaglia

#Leo Malvar

The Network is No Longer Isolated



The Human Interaction Challenge



Software displays results in User Interface (UI)

User asks for data or takes action by interacting with UI

The Value-Proposition for Programmability

```
# By Ed Scrimaglia
# Status dispositivos

import requests
import sys
import json
from pprint import pprint

def getOrganizaciones(_apikey):
    url = "https://api.meraki.com/api/v0/123456/devicesStatuses"

    headers = {
        'x-cisco-meraki-api-key': _apikey,
        'Content-Type': 'application/json'
    }

    r = requests.get(url, headers=headers, timeout=10)
    r.raise_for_status()
    return r
```

```
[{"name": "Access Point MR33", "serial": "Q2PD-V2DQ-2JX6", "mac": "0c:8d:db:95:79:dd", "publicIp": "190.111.211.47", "networkId": "L_591660401045801690", "status": "online", "lanIp": "10.1.10.3"}, {"name": "Sec_MX64W_IC", "serial": "Q2MN-QWHR-GJ9M", "mac": "e0:55:3d:8b:7f:a0", "publicIp": "190.111.211.47", "networkId": "L_591660401045801690", "status": "online", "usingCellularFailover": false, "wan1Ip": "10.54.152.14", "wan2Ip": null}, {"name": "Cam_MV21_IC", "serial": "Q2BV-KVX9-PM9T", "mac": "e0:55:3d:84:8c:95", "publicIp": "190.111.211.47", "networkId": "L_591660401045801690", "status": "online", "lanIp": "10.1.10.2"}]
```

The Value-Proposition for Programmability

Coding is the process of writing down instructions, in a language a computer can understand, to complete a specific task.

Q: What task?

A: Your task.

```
for switch in my_network:  
    for interface in switch:  
        if interface.is_down() and interface.last_change() > thirty_days:  
            interface.shutdown()  
            interface.set_description("Interface disabled per Policy")
```

And who are you?

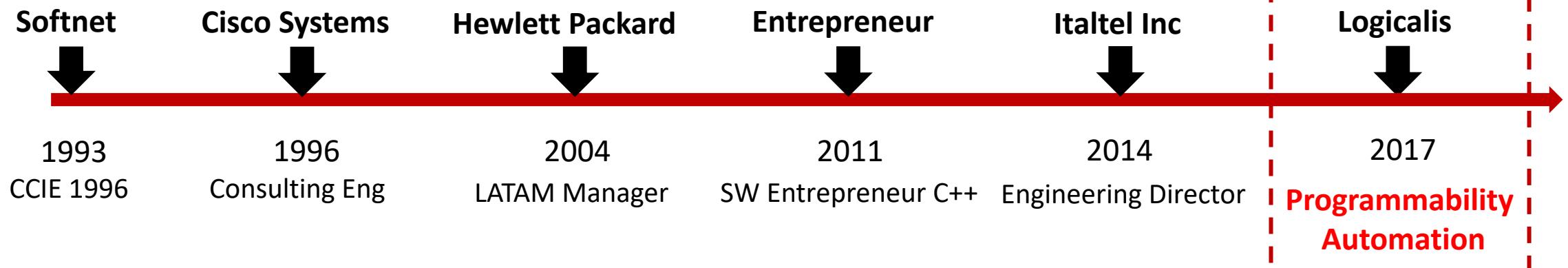
60% Network Engineer or Architect

14% Very familiar with Python and programming basics

41% OK with Python and programming basics

45% Not familiar with Python and programming basics

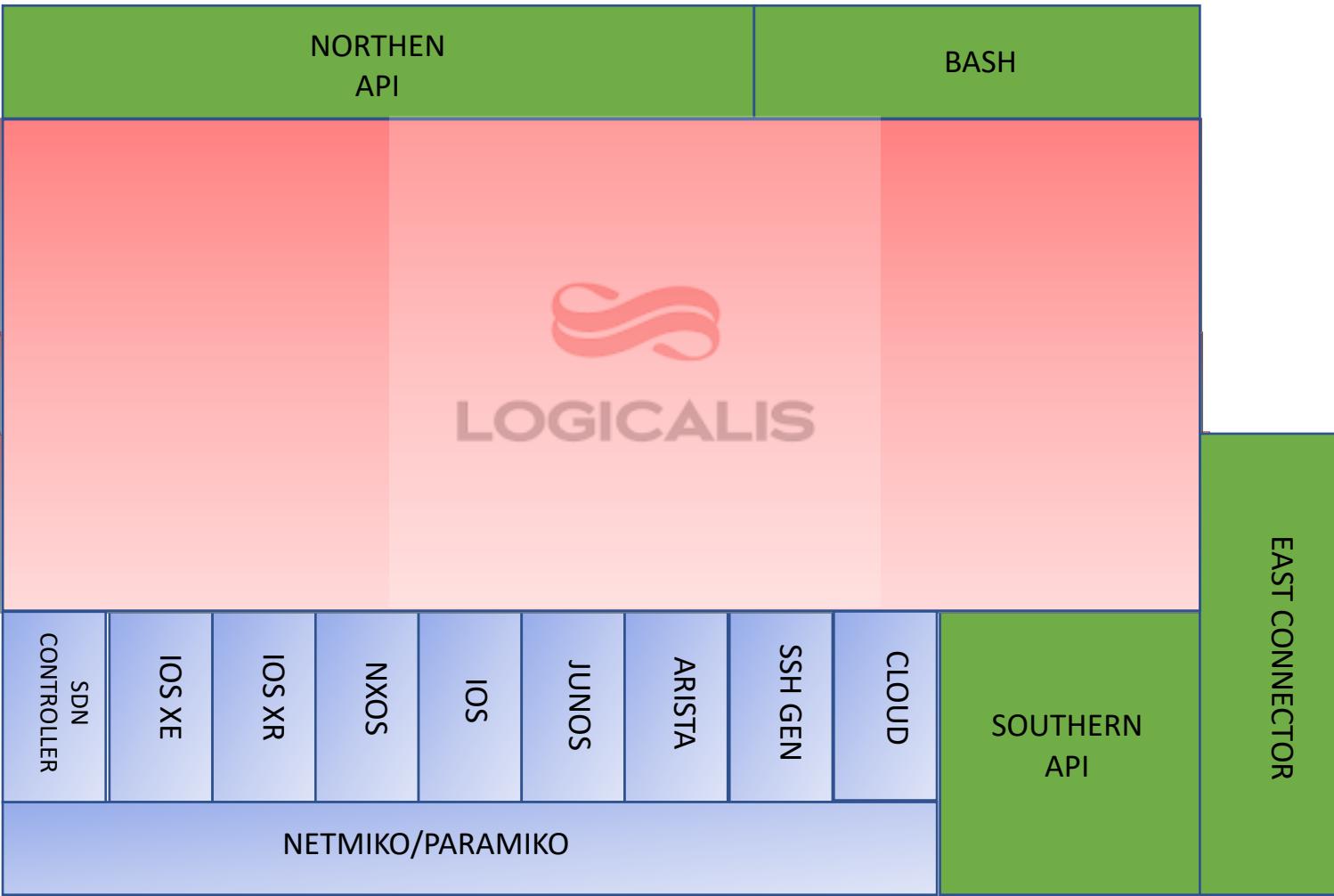
My own profile:



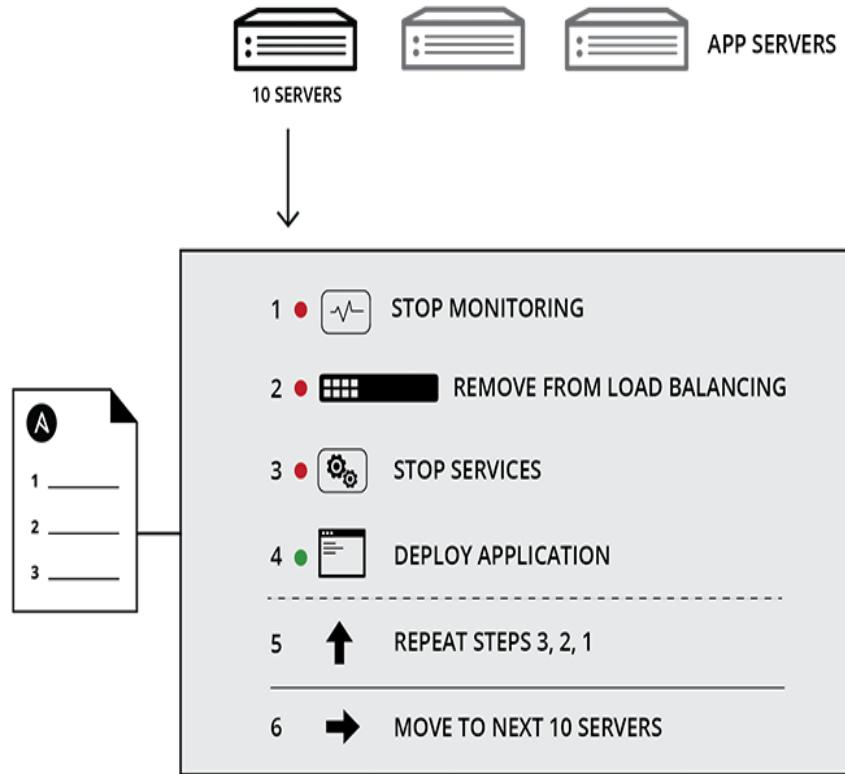
NEP@U



NEP@L – Automation Framework at Logicalis



NEP@L – Configuring & On-Boarding



```
# By Ed Scrimaglia
# Playbook Ansible 2.7
- name: CONFIGURACION SWITCHES NEXUS 9K SPINE
  hosts: dc1
  connection: network_cli
  gather_facts: no
```

tasks:

```
- name: CONFIGURACION VPC NEXUS SPINE
  nxos_interface:
    name: {{ item }}
    descripcion: 'Interfaces VPC to Leaf'
    mode: layer2
    hosts: {{ inventory_hosts }}
```

loop:

- Ethernet 1/1
- Ethernet 1/2

Inventory File

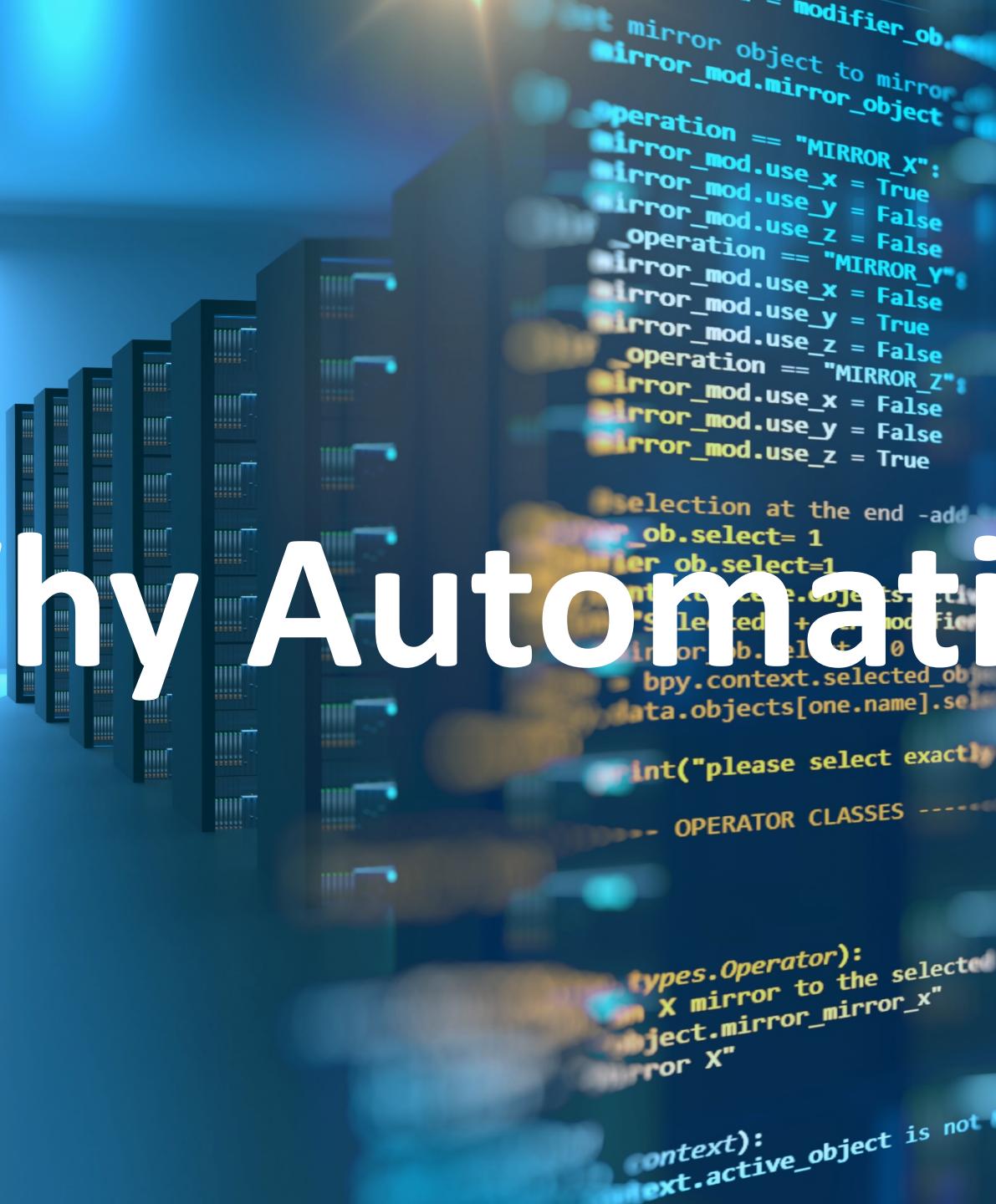
```
- name: CREACION VLANS
  nxos_vlan:
    vlan_id: {{ item.id }}
    name: {{ item.name }}
    host: {{ inventory_hosts }}
    state: present
```

[dc1]
Rtr1 ansible_host=10.1.1.1
Rtr2 ansible_host=10.1.1.2
Leaf ansible_host=10.1.2.[1:30]

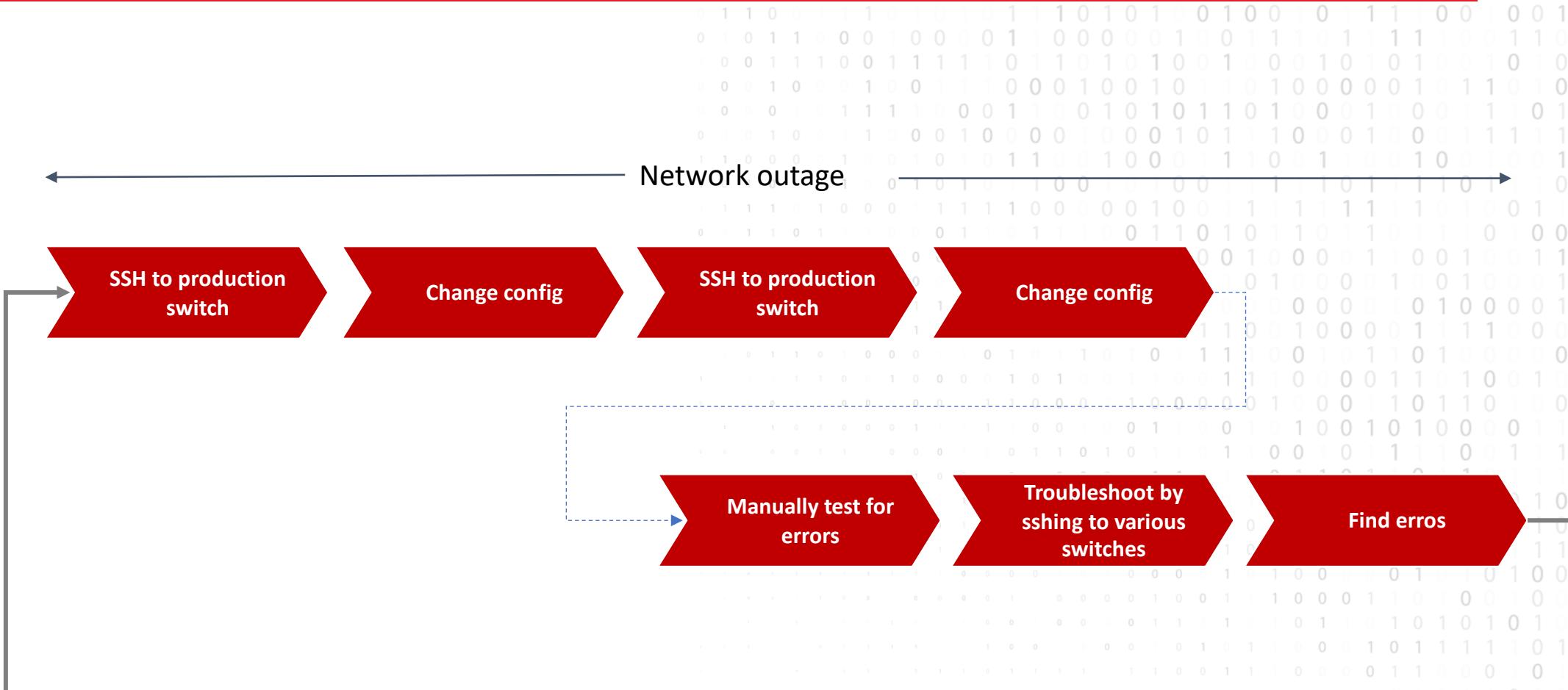
loop:

- id: 20
 name: Administracion
- id: 30
 name: Ingenieria

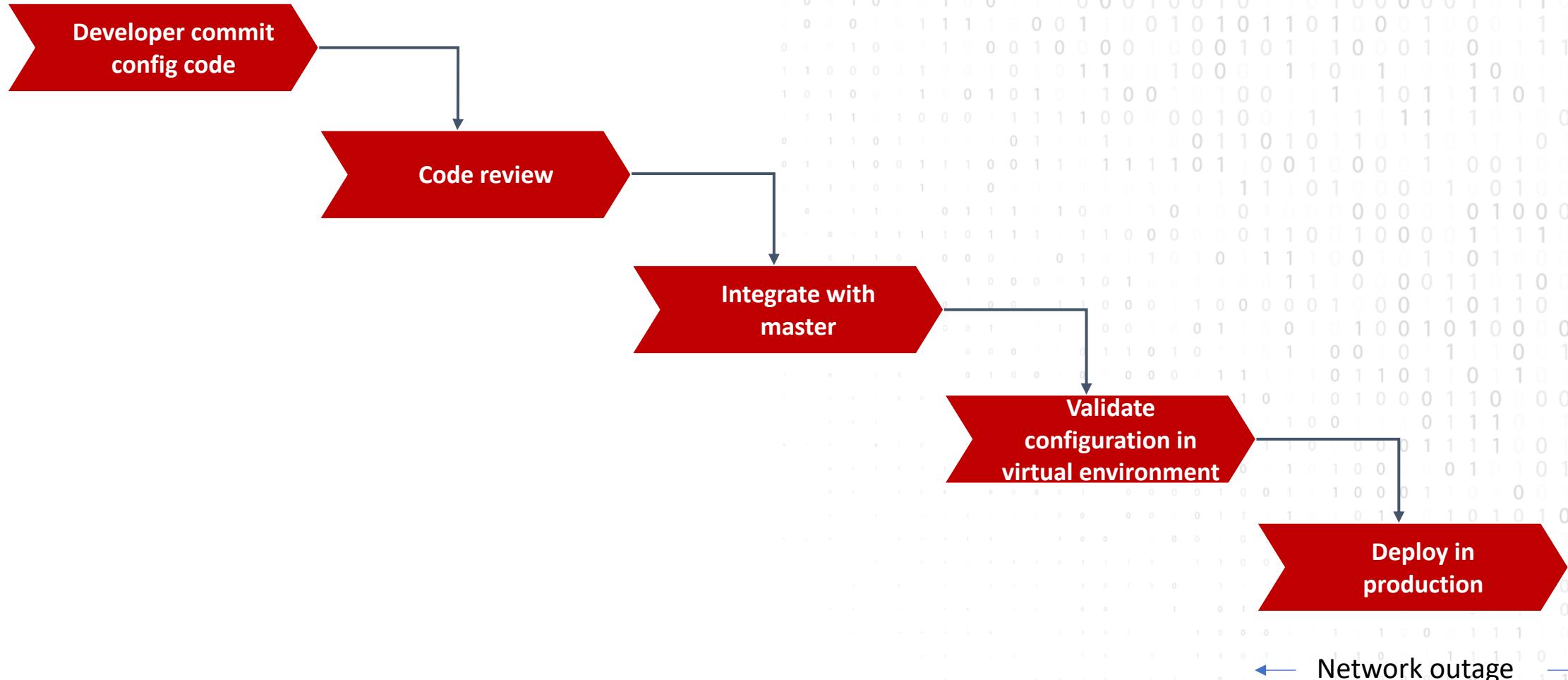
Why Automation



Manually configuration process – Network outage



Automated provisioning – Network outage



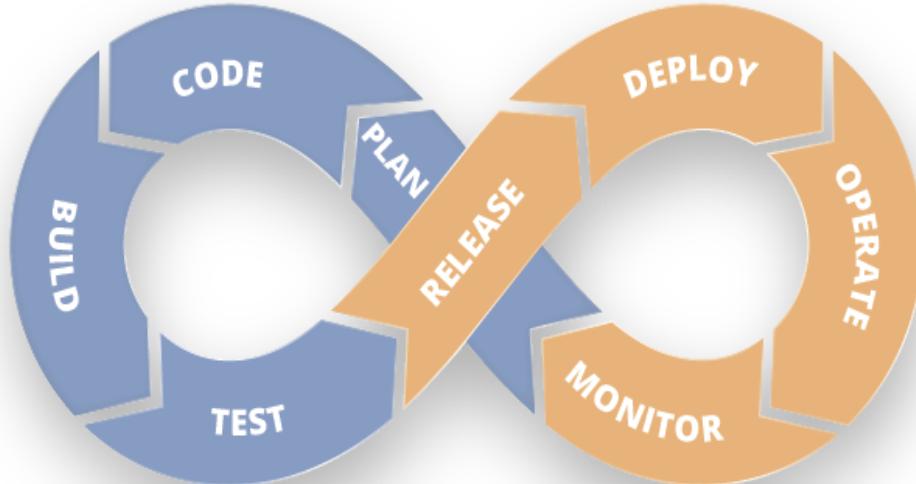
* Only as long as it takes to run code

CI/CD Continuous Development in the network context

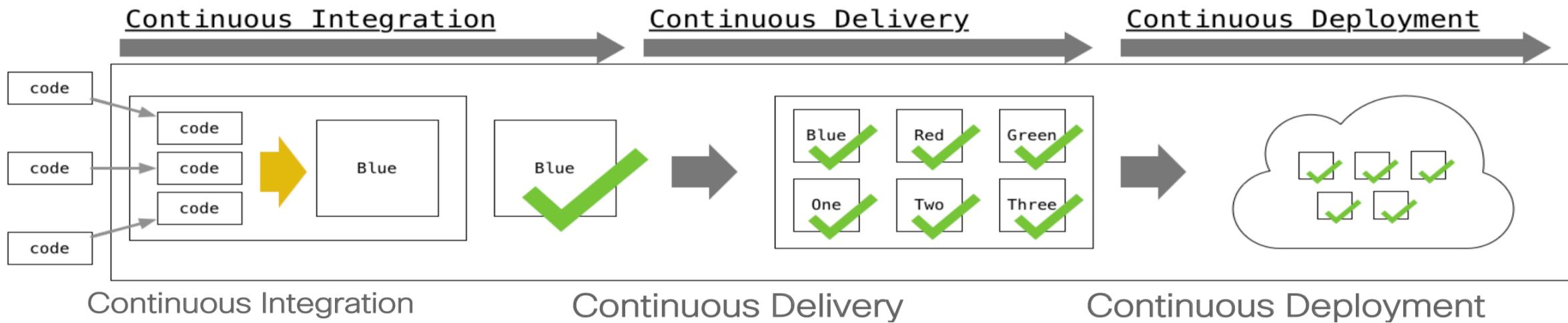
```
    modifier_ob = modifier_object_to_mirror(modifier, mirror_mod.mirror_object)
    if operation == "MIRROR_X":
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    elif operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    elif operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True
    else:
        raise ValueError("selection at the end - add more cases")
    ob.select= 1
    mirror_mod.select=1
    bpy.context.scene.objects.active = selected
    mirror_mod.select = 0
    bpy.context.selected_objects.append(mirror_mod)
    data.objects[one.name].select = 1
    int('please select exactly one object')
-- OPERATOR CLASSES --
<operator>
  types.Operator):
    X mirror to the selected
    object.mirror_mirror_x"
    mirror X"
    context):
      ext.active_object is not
      ext.active_object is not
```

Continuos Development

Continuous Development (CD) is a continuous process in which developers and testers collaboratively create, validate and delivery new code

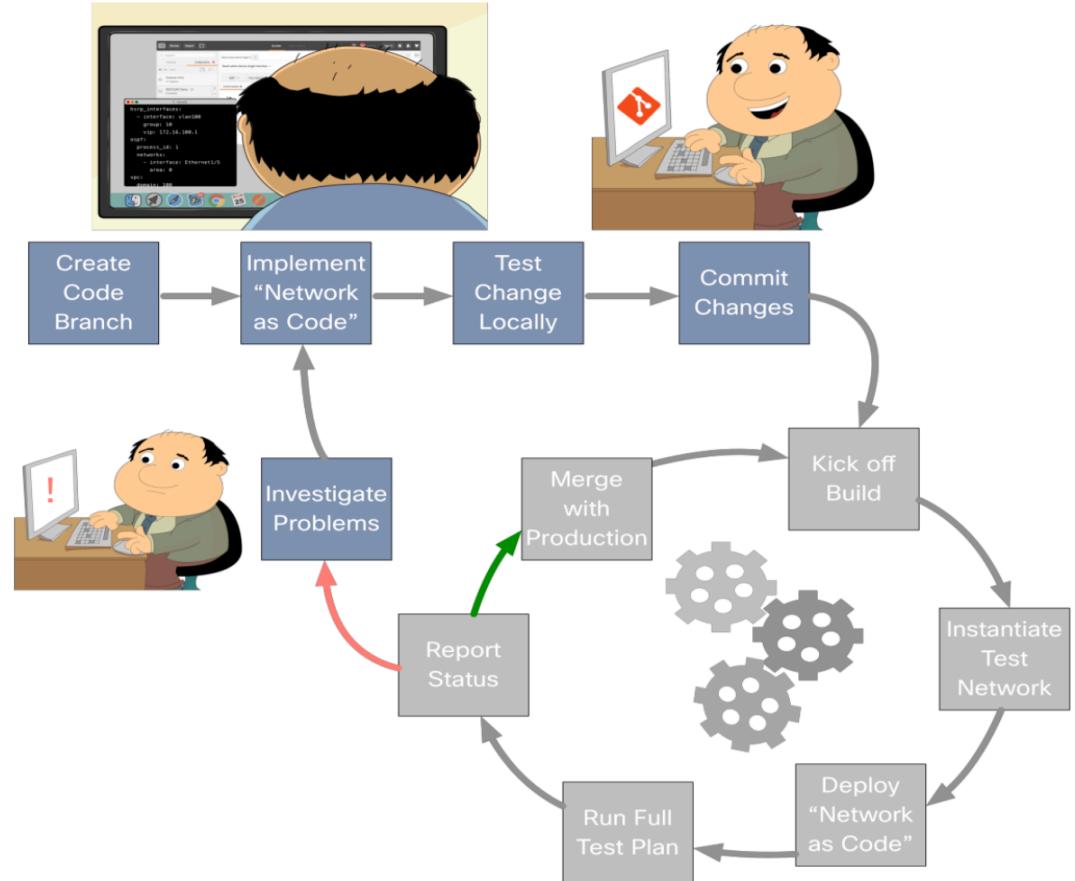


Continuos Development Process Flow– CI/CD



NetDevOps Pipeline— CI/CD

- “Treating the Network as Code”
- Network Configuration stored in Source Control
- Changes are proposed in code “branches”
- CICD Build Servers deploy and test proposed configurations
- Successful configurations ready for deployment to “Production”



“Networks are no longer
configured
.... they are programmed”

By Ed Scrimaglia

BILL PRICE & DAVID JAFFE

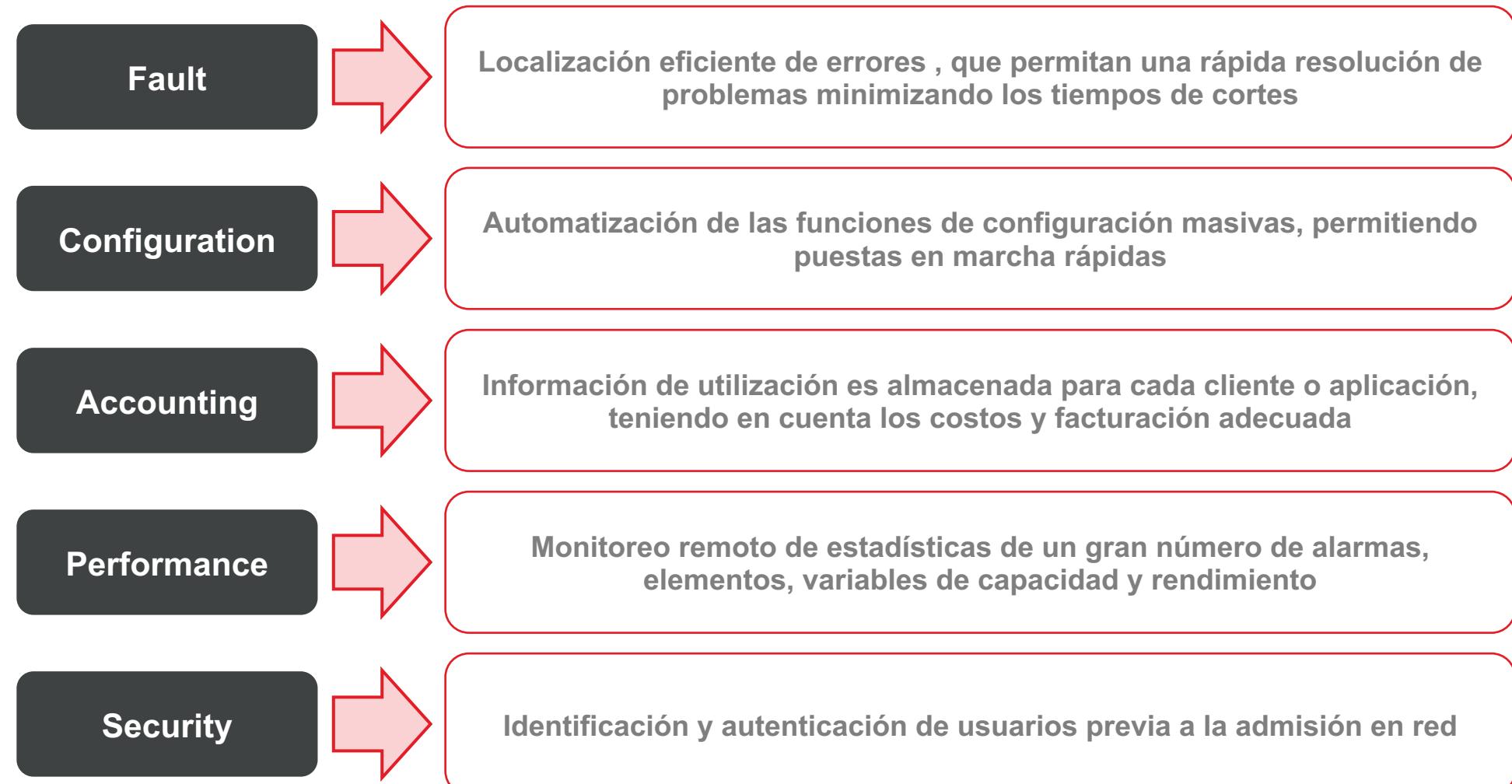
THE BEST SERVICE IS NO SERVICE

HOW TO LIBERATE YOUR CUSTOMERS
FROM CUSTOMER SERVICE,
KEEP THEM HAPPY & CONTROL COSTS

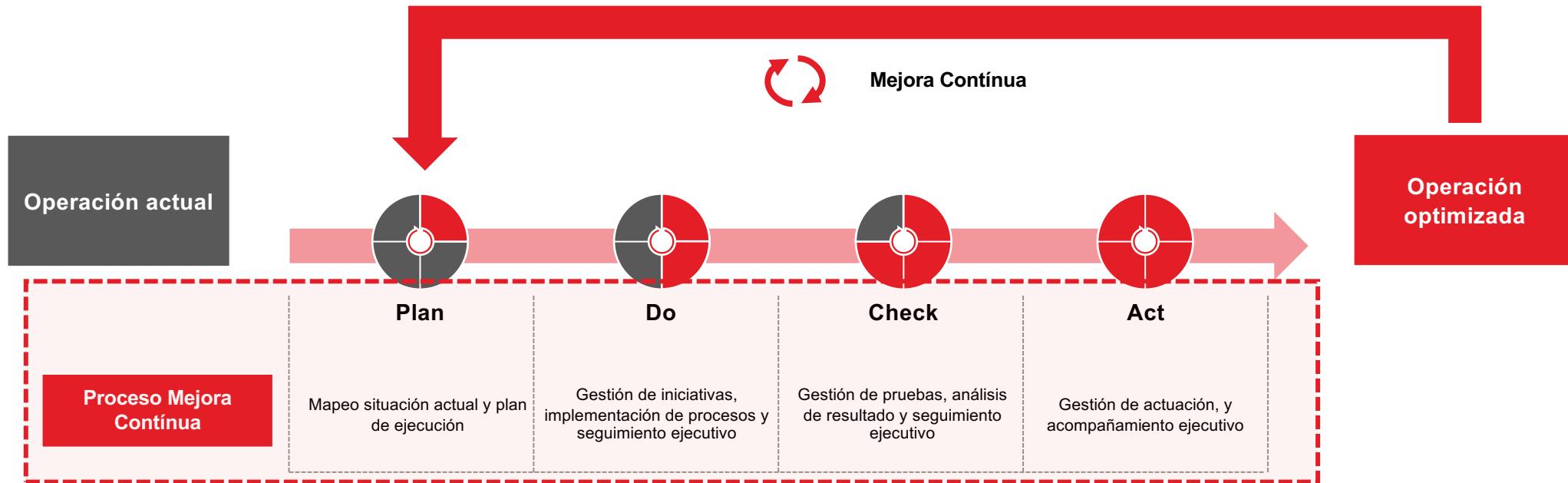


Managed Services - Methodology

FCAPS (Fault, Configuration, Accounting, Performance, Security)
son las categorías en las cuales el modelo ISO define las tareas de Gestión de TI



Managed Services - Methodology



- ✓ **Business alignment**
- ✓ **Best Practices**
- ✓ **Strategic Vision**
- ✓ **Business enabling**

Managed Services as a Continue Services Evolution



Estadio

- Fallas sirven como gatillo para acciones correctivas



- Monitoreo de Items

- Acciones preventivas basadas en experiencias anteriores

- Mejores Prácticas
- Estudios de software en comparación regional

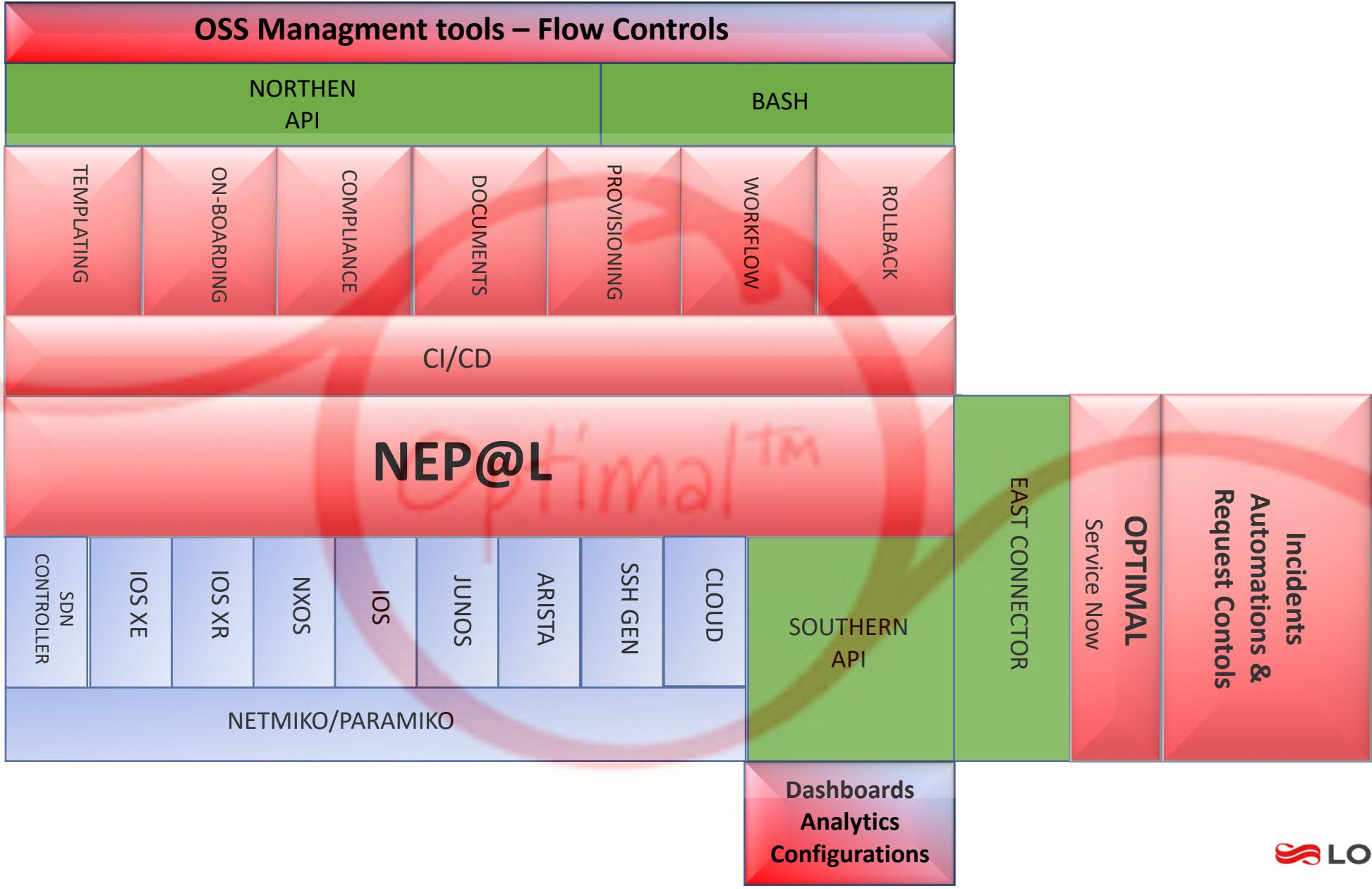
- Identificación temprana de posibles fallas basado en el aprendizaje de maquina

- Monitoreo de subitems
- Correlación de alarmas

- La estructura se adapta de forma automática por medio de una identificación de necesidad futura

- Generación de tickets automáticos por monitoreo

Services – Automation Framework at Logicalis





Stone Age

Spanning Tree
VLANs



Bronze Age

Routing Protocols
WAN Design
IP-magedon



The Renaissance

SDN
OpenFlow
Controllers
Overlays
MP-BGP
VXLAN
Micro-Segmentation
White Box



Programmable Age

Cloud
Python
REST / APIs
NETCONF / YANG
“Fabrics”
Network Function
Virtualization (NFV)
DevOps
Containers

The Four Ages of Networking.....



DevNet



Christian Hisas
Senior Technology Manager at Logicalis
4mo • Edited

Aceptando el desafío de transformarnos!!

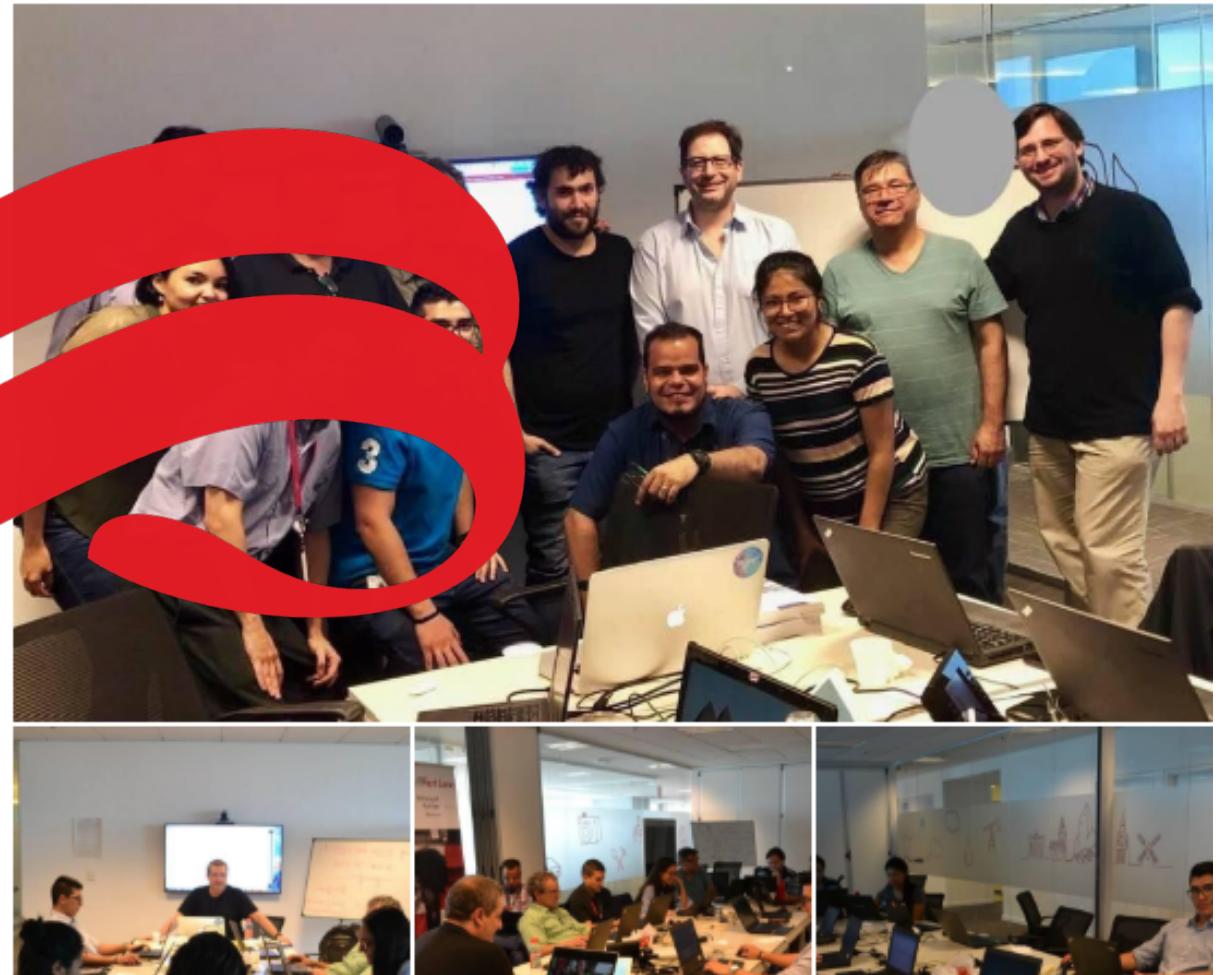
Gran trabajo **Edgardo Scrimaglia** y **Sebastian Troncoso**



Edgardo Scrimaglia • 1st
Network Programmability & Automation
2mo

#Logicalis #NetworkProgrammability #Automation enseñar, lo que más me gusta.

See translation



**“we are ready...
Network Programmability &
Automation is
NOW”**

By leo



TIC FORUM
IN MOTION 2019

 LOGICALIS

Thanks!

#Ed Scrimaglia
#Leo Malvar