

Juniper Automation Awareness Anuta ATOM Bootcamp (Day-2)



Sahil Katira

TME



Nishant Agarwal

Sr. Software Engineer



Atmesh Agarwal

Sr. Solution Architect

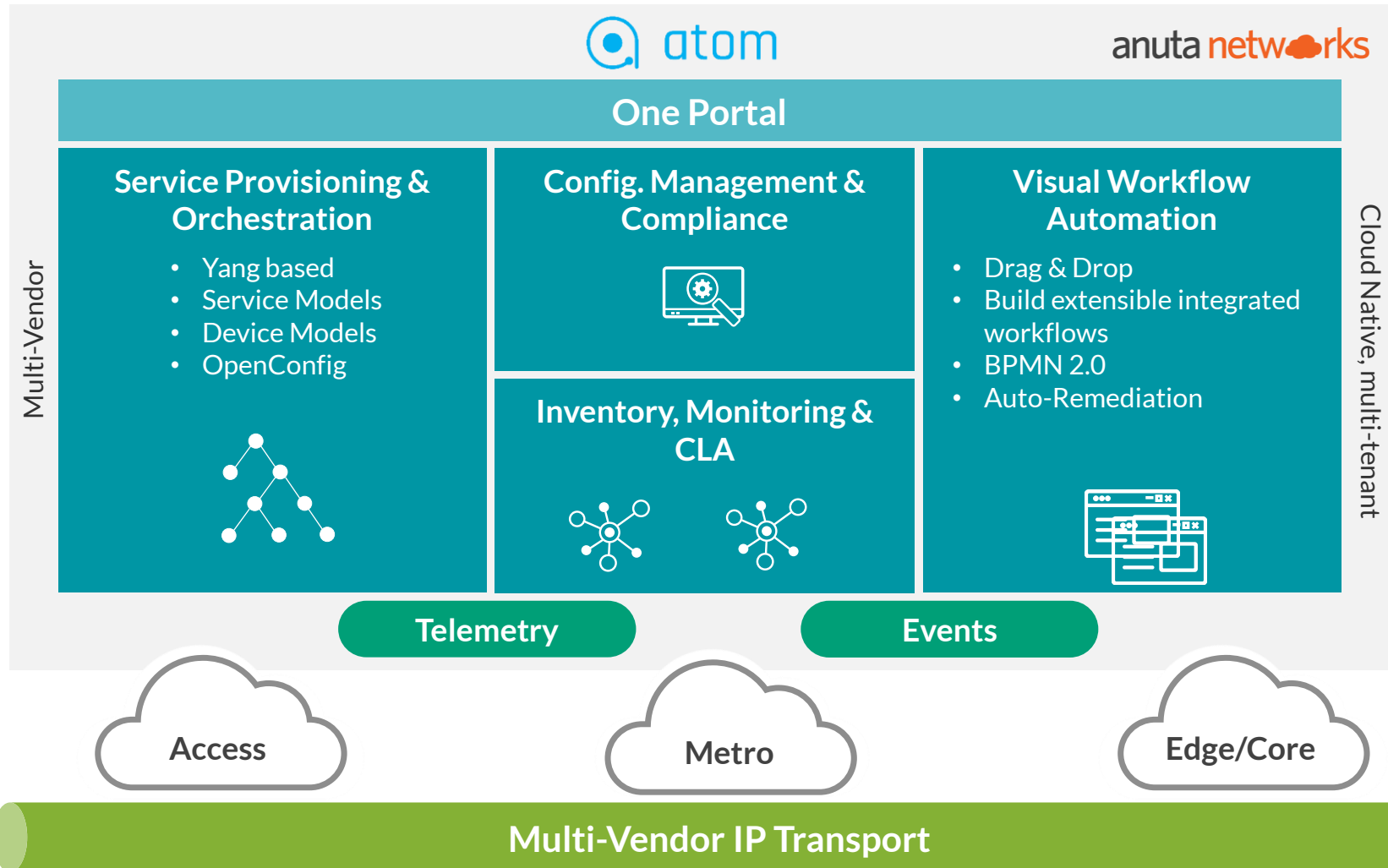


AGENDA

- Day – 1: Compliance Management
- Day – 2: Workflow Hands-On
- Why Low-Code Automation ?
- Workflow Automation in ATOM
- Hands-On Lab
- Advanced Concepts in Workflow Automation
- Day – 3: Service Model Hands-On
- Day – 4: CLA, Scale, Licensing, Administration.

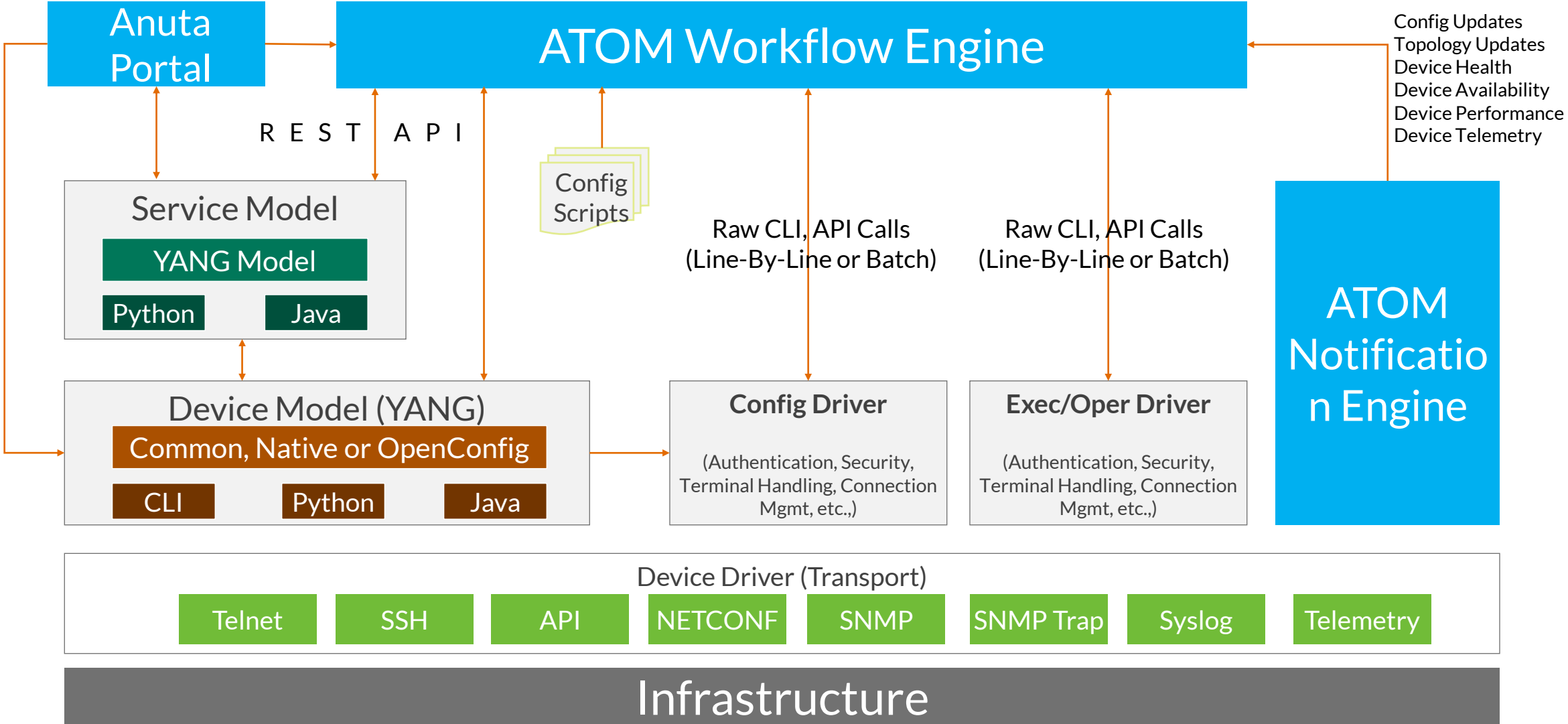
Anuta ATOM

Service orchestration, Workflow and Compliance with Closed loop Automation



- 3rd Party Resale agreement
- On Juniper pricelist
- JTAC
- Juniper Pro services

Anuta ATOM – stateful and stateless automation



Workflow Automation with Anuta ATOM

Network Operations - Today's Reality

Paper-based Methods-of-Procedures
often interpreted differently and need extra efforts to maintain them



Manual Operations
Delays process execution



Siloed Tasks in the manual workflow
Less visibility into tasks, leading to operational bottlenecks and process inefficiencies



Repetitive Tasks
Affects networking teams' productivity



Disjoint Processes affecting the flow
Low participation in automation



An Existing Method-of-Procedure

Gather Requirements

Create candidate config

Raise ServiceNow Ticket
[Change Request]

Pre-Checks

Update ServiceNow Ticket
[Implementation]

Reserve IP block on IPAM & Approvals

Manual Device Configs

Post-Checks

Update ServiceNow Ticket
[Review]

Close ServiceNow Ticket
[Closed]

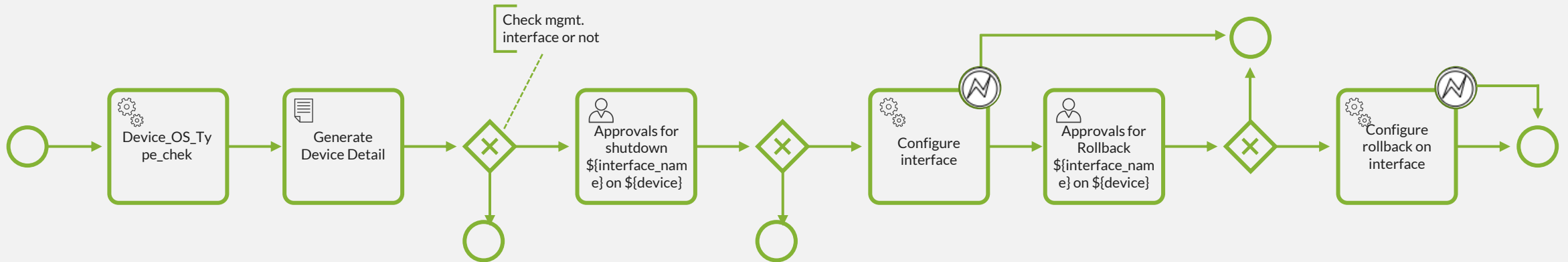
- Manual preparation of configurations
- Multiple approval requests
- Ticket lifecycle management
- Manual collaboration with other teams

Anuta ATOM

Visualize Workflows with Low Code



Visual Workflow Automation leverages Low Code to break down a higher-level activity into subtasks and ties them together with network events, provisioning actions, show-commands, pre-checks, post-checks, user forms and approvals, timed background tasks, inventory checks, alerts and others.



Examples : Device onboarding, SW Upgrades, Migrations, LAG management, Maintenance Procedures

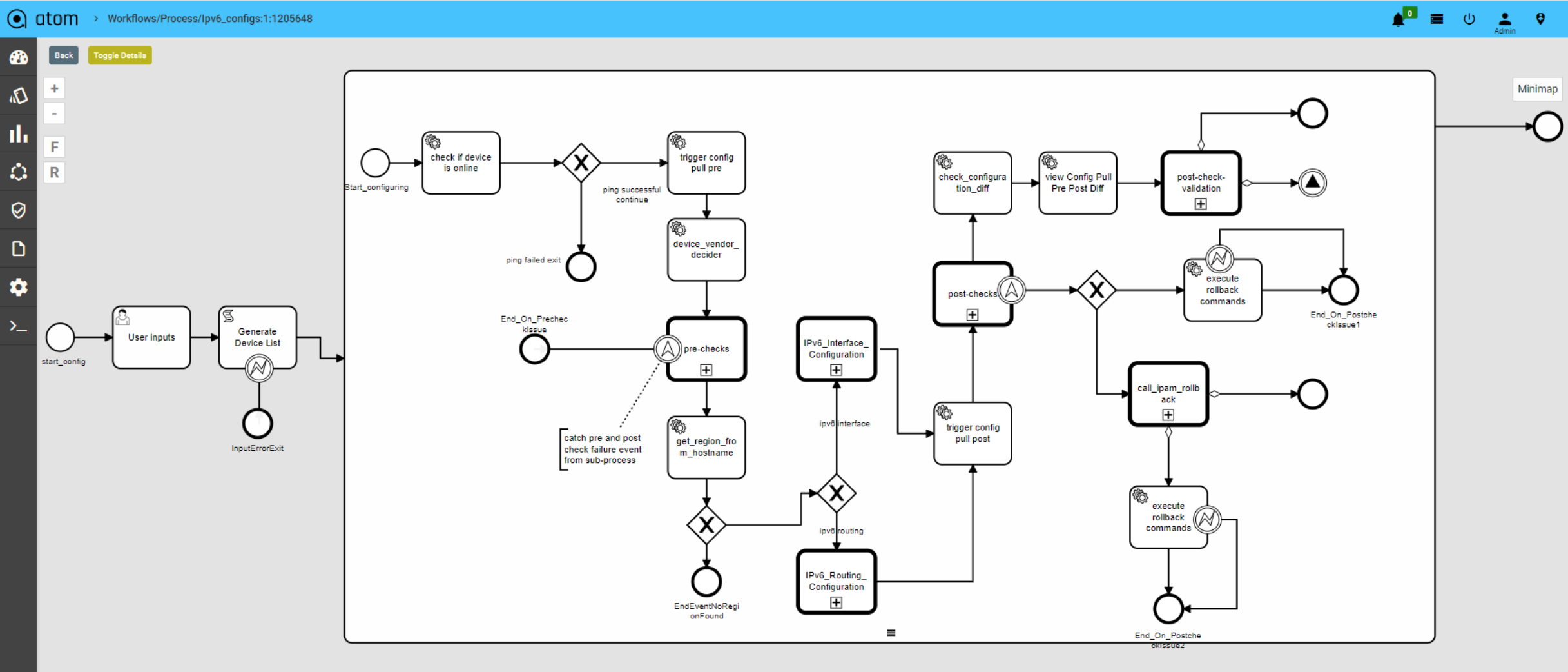
- Key capabilities:

- Reusable sub-routines
- Integrates with OSS, IPAM and ITSM tools
- Multivendor support

- Benefits

- Easy to deploy, manage
- Eliminate, reduce risk
- Improve operational efficiency

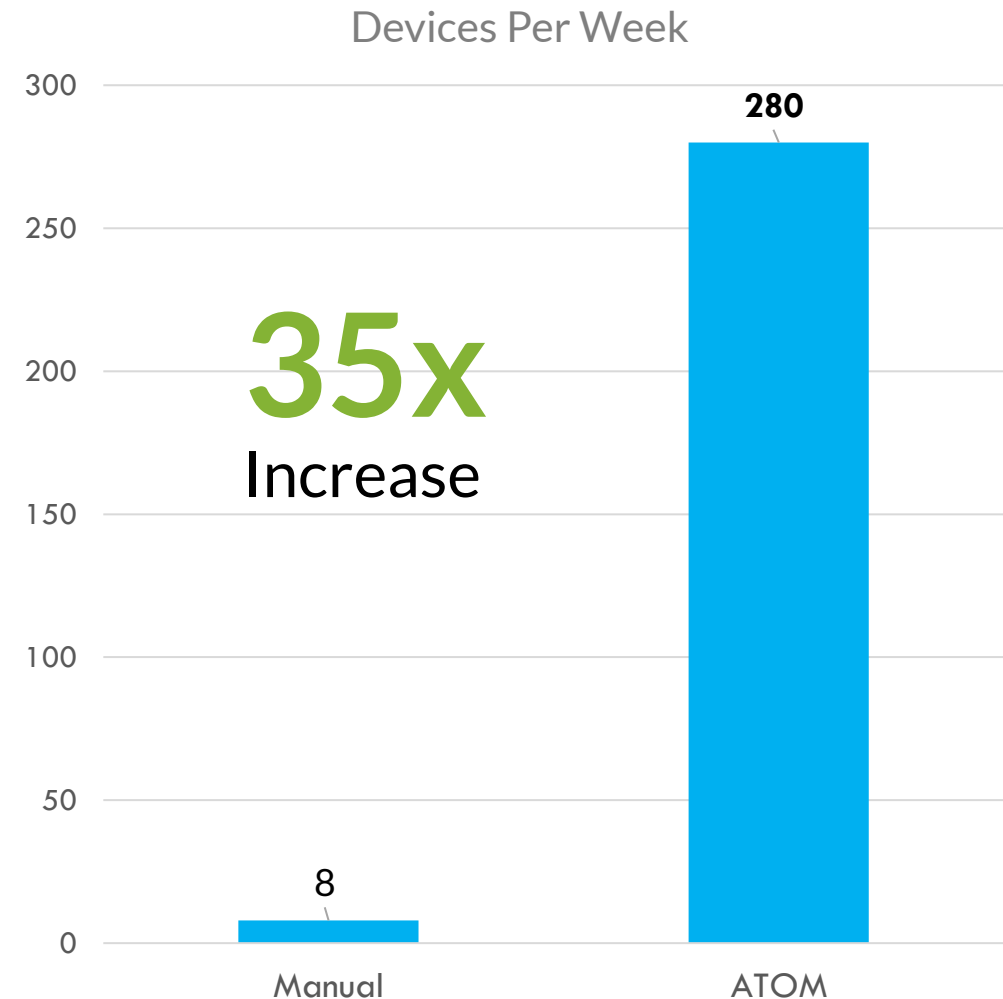
Visual, Programmable, Workflows



Automation Efficiency Gain – IPv6 Migration

Efficiency gain of ~95 minutes per device

Activity	Manual (sec)	ATOM (sec)
Identify Components	900	10
Config Preparation	900	10
Pre-Check Captures	600	90
Change Execution	900	30
Post-Check Captures & Validation	2400	90
IPv6 Address Management	900	10
Rollback (Automated & On-Demand)	1500	45
Overall (per Device)	110 Minutes	< 5 Minutes



Why Workflow Automation?

Reduce time, costs and errors

Helps in accelerating processes through MOP standardization

Increases efficiency

Drives more silos into network automation through Integrations

Offers more visibility into operational processes

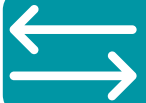
Where to use Workflow Automation ?



Software Upgrades



Pre & Post validations for Service Orchestration



Network Migrations



Alerting & Closed-Loop Automation

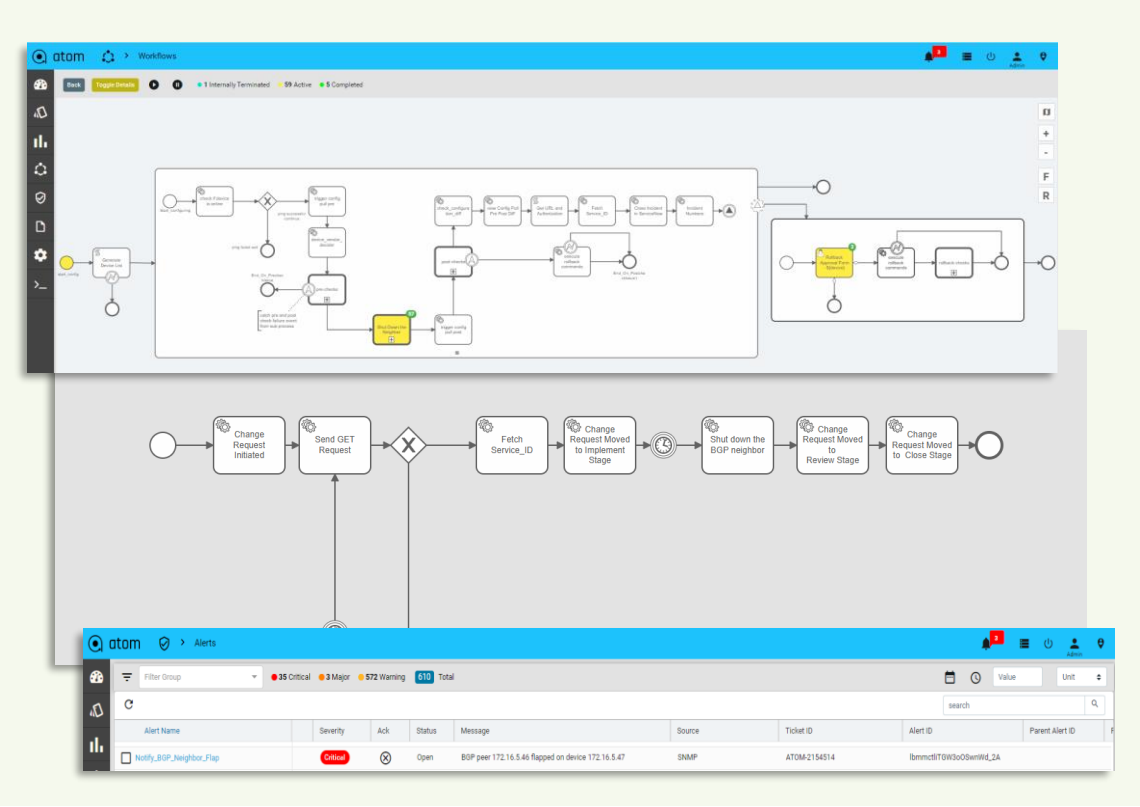


Bulk configuration changes such as SNMP config



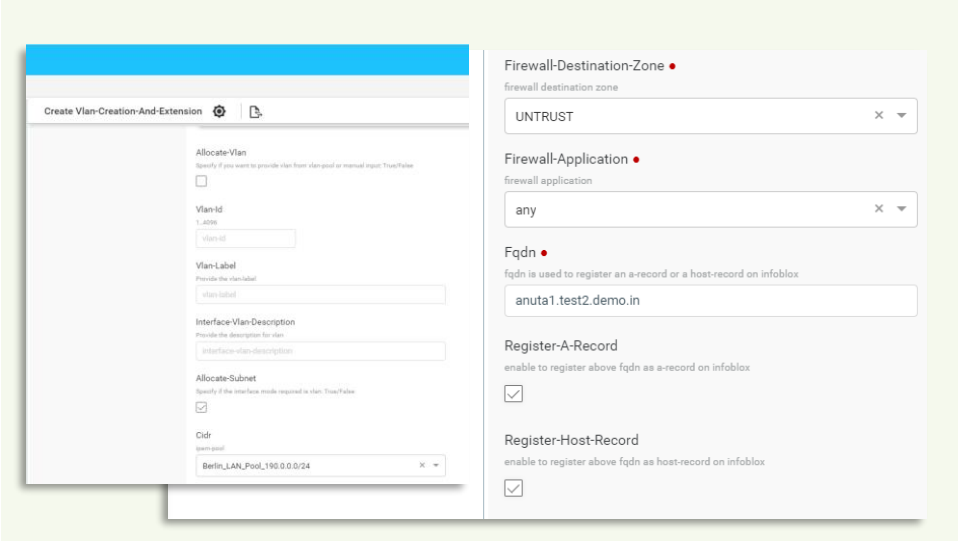
Provisioning, troubleshooting MOPs involving 3rd party integrations

Integrations with ITSM & IPAM



ServiceNow/Jira

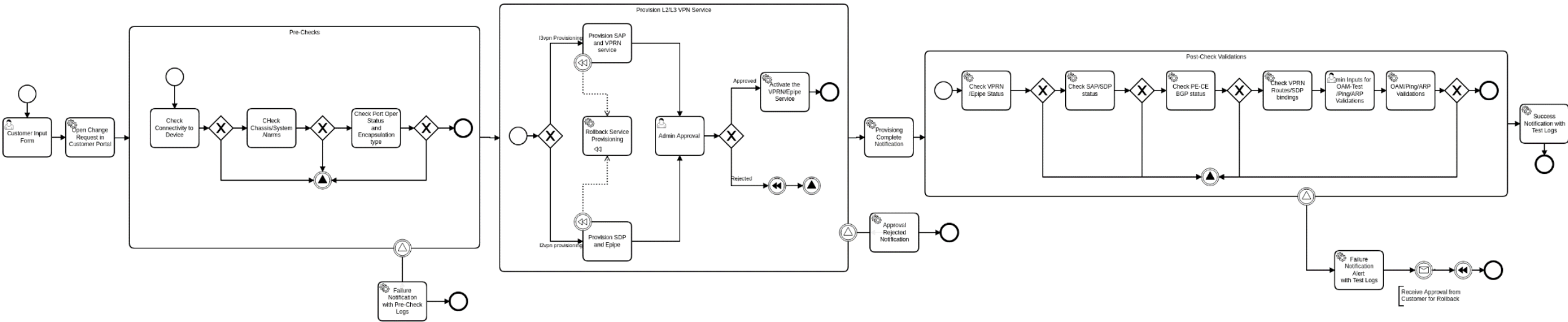
- Integration into Alerting & Workflows
- Ticket Lifecycle Management
- Incident Management
- Review & Approvals



Infoblox

- IPAM
- DNS
- API level integrations
- Available across services & workflows

Add Pre & Post Checks to Services using Workflows



Pre-Checks

- Device connectivity from ATOM.
- Check for any red flags from the Hardware.
- Check target port operation status and encapsulation type

Service Invocation

- Provision the L3VPN/L2VPN service on the hardware.
- On approval from Network Admin activate the service on the router.

Post-Checks

- Check for L3VPN/L2VPN service state.
- Check for target interface state.
- Log the route table summary for the service provisioned.
- Take admin inputs for OAM Validations/Ping Test etc.
- Perform the above checks on the router.

Anuta ATOM - Out of Box Low-Code Workflows

Out of Box Low-Code Workflows	
<ol style="list-style-type: none"> 1. Basic software upgrade 2. Advanced Software Upgrade 3. Basic Routing engine switchover 4. RMA between similar devices 5. Config Restoration 6. Multi-vendor Network Migration from IPV4 to IPV6 7. Add a member to existing LAG and validate increase in bandwidth 8. Provision on-demand configuration commands to set of devices or device groups 9. Replace or Update SNMP Configuration 10. Execute Compliance Profiles and remediate non-compliance 	<ol style="list-style-type: none"> 11. DHCP Based Zero touch provisioning (ZTP) of greenfield network 12. Invoking Ansible scripts from workflow 13. Provision EVPN E-Line services with prechecks and post checks 14. Discover existing L2VPN Services. Create L2vpn service if missing 15. Integration with Service Now 16. Integration with Jira 17. Ping devices to determine availability 18. Execute Traceroute on devices 19. Collect and store inventory information such as chassis ID, sfp etc.

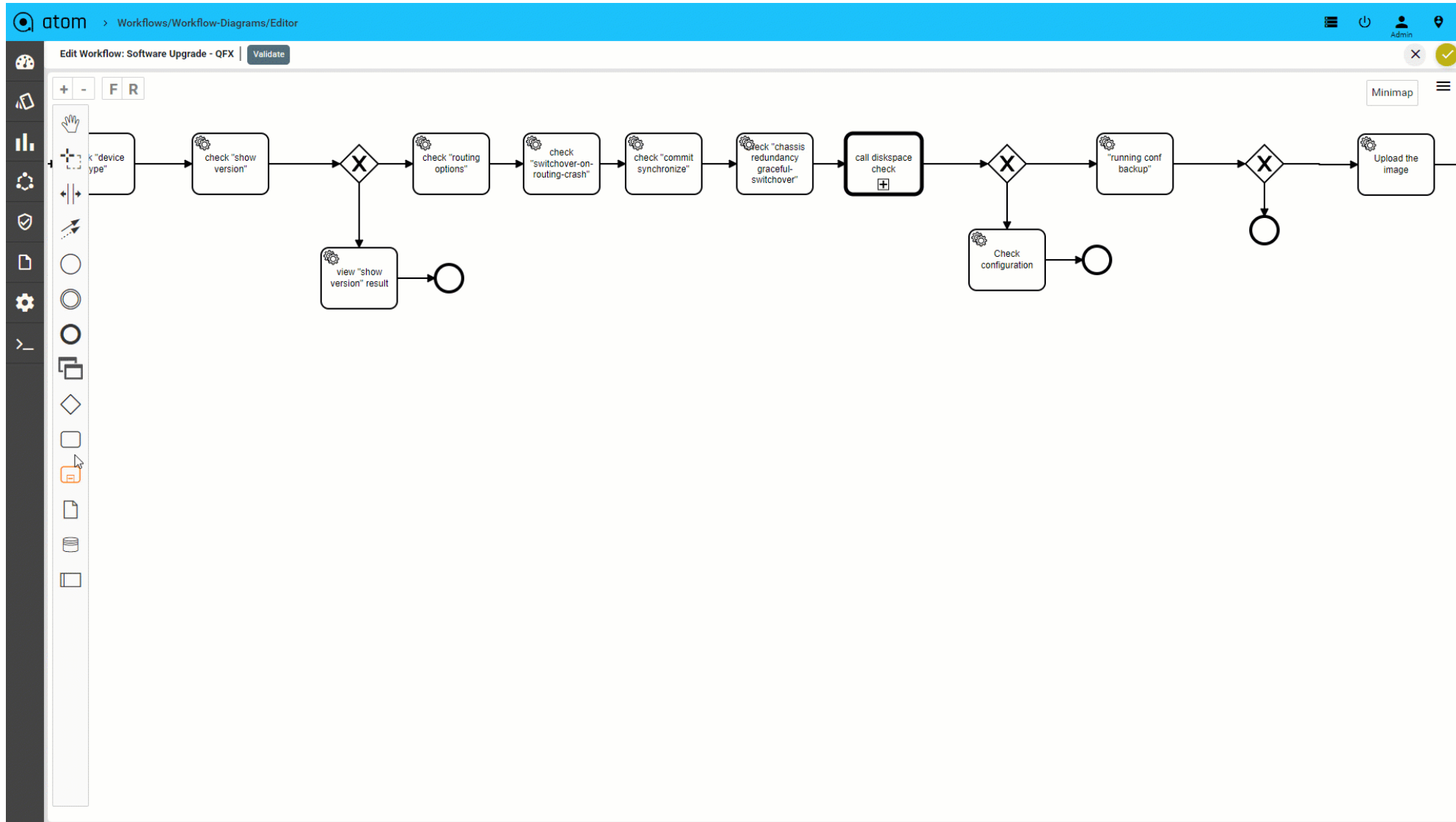
Support for 45+ Vendors
150+ Platforms

- ✓ JUNOS
- ✓ MX
- ✓ PTX
- ✓ ACX
- ✓ QFX
- ✓ EX
- ✓ SRX

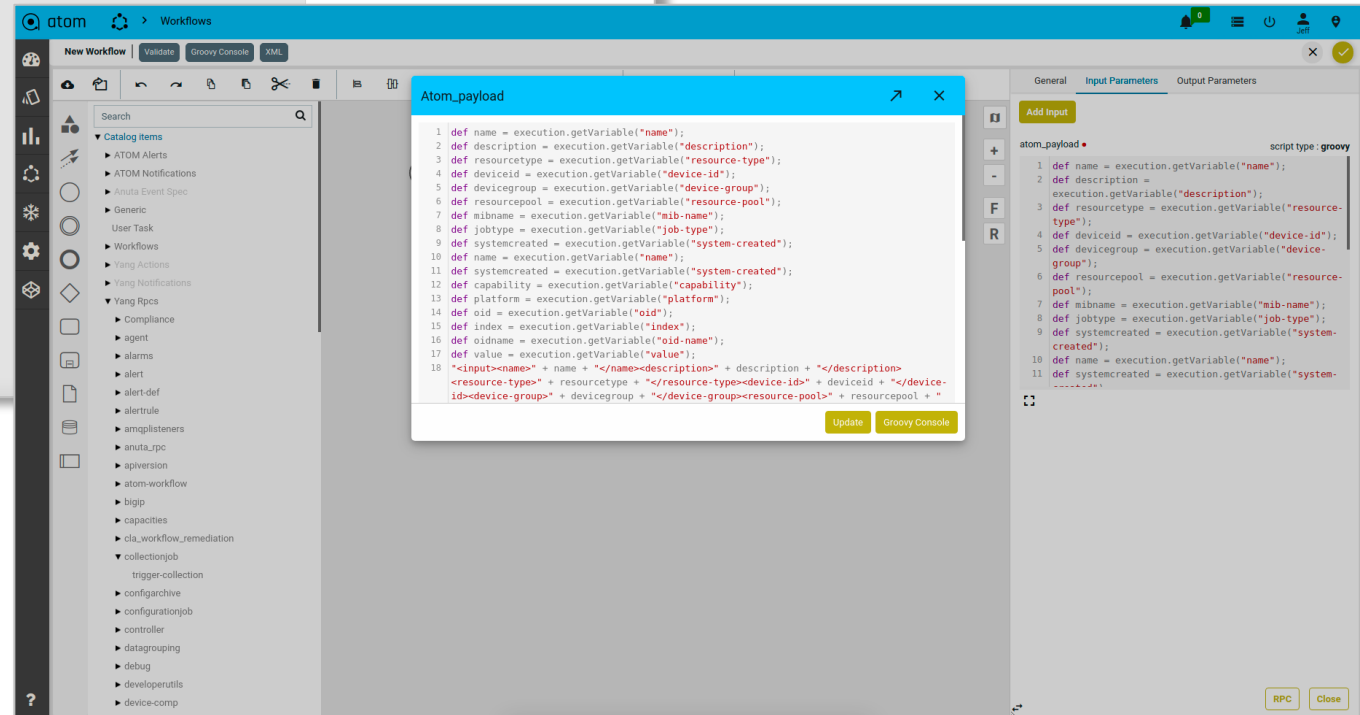
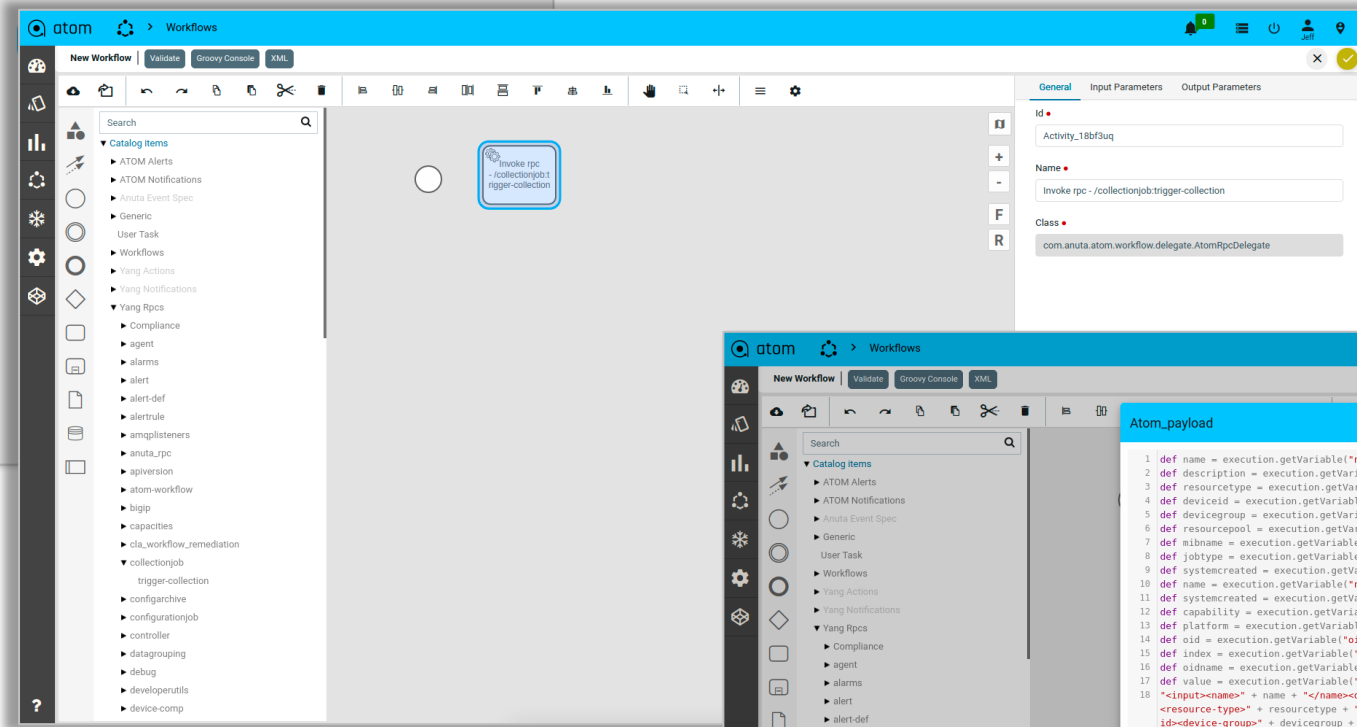
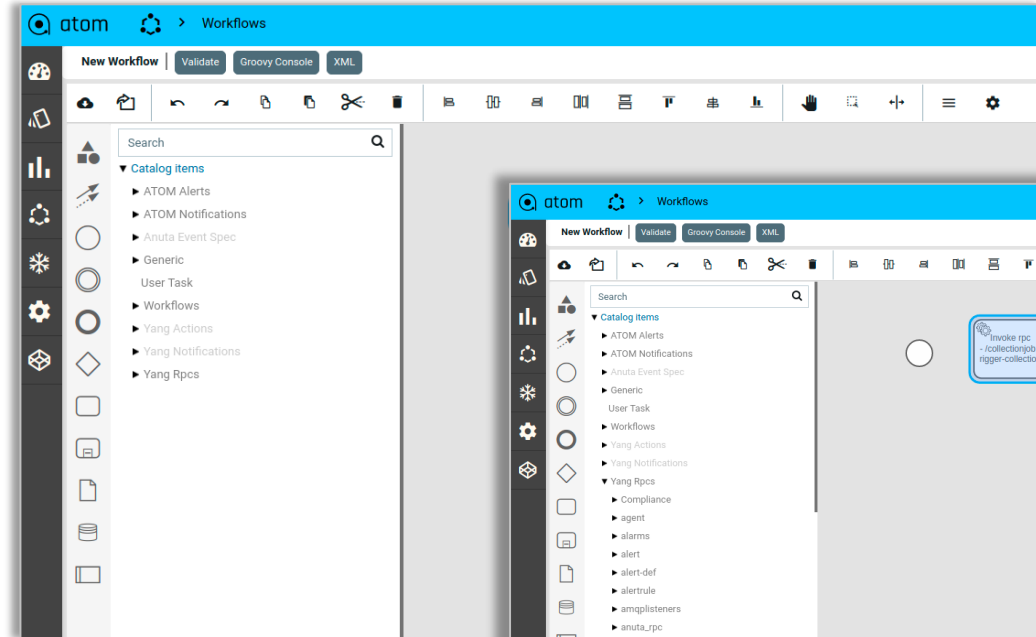
- ✓ IOS-XR
- ✓ ASR
- ✓ N9K
- ✓ ISR
- ✓ NCS
- ✓ Cat 9K*
- ✓ Firepower*

* - Roadmap CY21

ATOM – Extend existing workflows

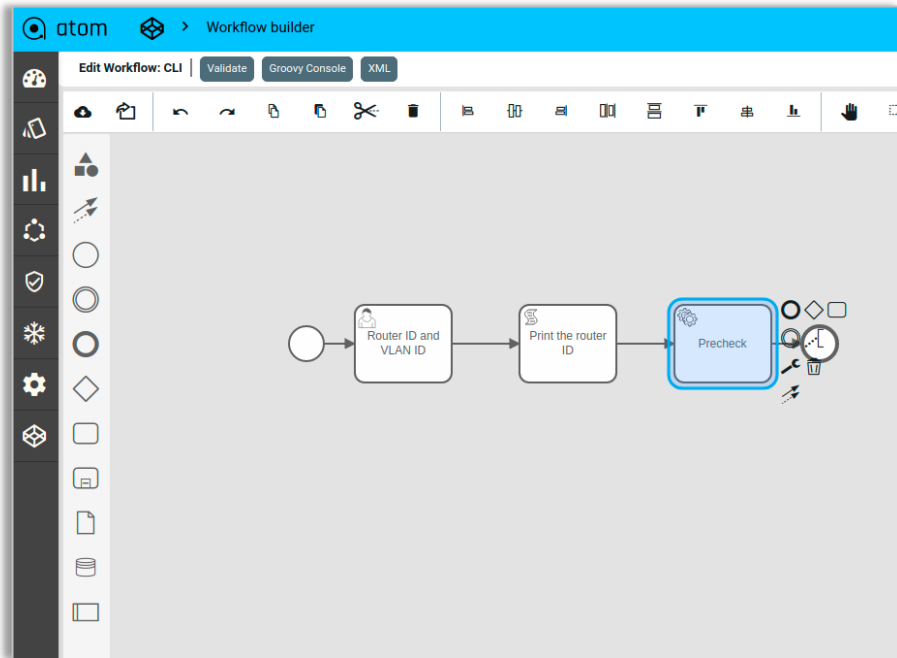


Workflow – Pre-Built Library



Drag-n-Drop Builder
Pre-Created Scripts - Customizable

Workflow – Integration with CLI



1. Create Service Task with RPC Delegate

The screenshot shows the 'Input Parameters' configuration window for the 'Precheck' task. The window has tabs for 'General', 'Input Parameters', and 'Output Parameters'. The 'Input Parameters' tab is active, showing an 'Add Input' button and three input fields: 'atom_url' (value: /developerutils:execute-cli-command), 'atom_action' (value: POST), and 'atom_payload' (script type: groovy). The Groovy script is as follows:

```
1 def RID = execution.getVariable("Router ID")
2 def Intf = execution.getVariable("Interface")
3
4 '<input><device-id>'+RID+'</device-id><command>show
configuration interfaces '+Intf+'</command></input>'
```

2. Edit the Command

The screenshot shows the details page for 'Workflow Instance 3810387 - Precheck'. It displays the following information:

- Task ID: J_yrI60ed-QF2bGjgAo01sAg
- Parent ID: HvpH3mXqEQX-upM8U1TwKjA
- User Name: admin
- Time Taken: 11/08/2021, 18:47:51 - 11/08/2021, 18:47:59 (8 seconds)

Below this information are tabs for 'Summary' and 'Logs'. The 'Summary' tab is active, showing the following details:

developerutils:execute-cli-command	
Input	<input><device-id>172.16.4.167</device-id><command>show configuration interfaces ge-0/0/1</command></input>
Output	<output><response>show configuration interfaces ge-0/0/1 unit 0 { family inet { address 31.1.2.3/24; } family mpls; } admin@mx66gt; </response></output>

Workflow – Script Integration

The screenshot shows the Atom workflow editor interface. On the left, a workflow diagram is displayed within a 'MultInstance' container. The workflow starts with a start node, followed by a task 'Router ID and VLAN ID', then a script task 'Print the router ID' (highlighted with a blue border), and finally a 'Precheck' task. On the right, the configuration panel for the selected script task is visible. It includes fields for 'Id' (Activity_1bng399), 'Name' (Print the router ID), and 'Script Format' (Groovy). The 'Script Type' is set to 'INLINE SCRIPT'. The script content is as follows:

```
1 def RID = execution.getVariable("Router ID")
2 println "ID of the router to be provisioned: " + RID
+ RID
```

The 'Result Variable' field is currently empty.

1. Create a Script Task and Edit the Script

The screenshot shows the Groovy Console window. The script content is the same as in the previous screenshot. The console output shows the result of the script execution:

```
1. def RID = "172.16.23.12"
2. console.log("ID of the router to be provisioned: " + RID)

ID of the router to be provisioned: 172.16.23.12
```
















The console output is displayed under the 'Console' tab, with 'Results' and 'Error' tabs also visible.

2. Validate Script

ATOM Workflow - Customizable Reports

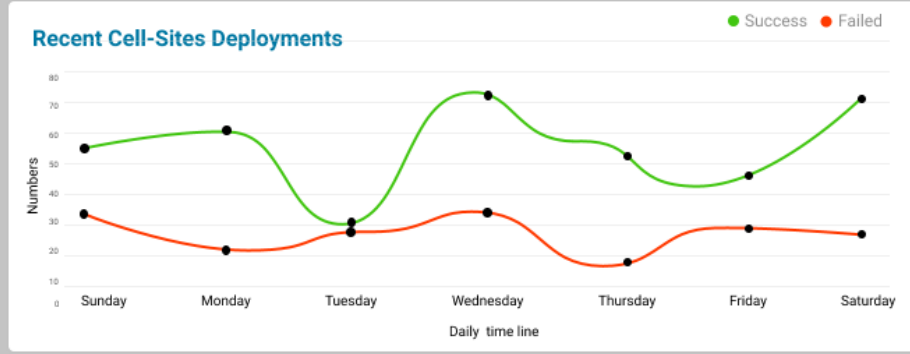
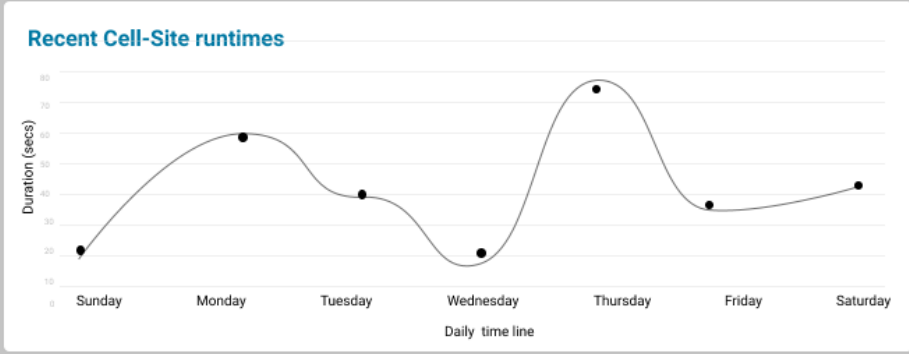
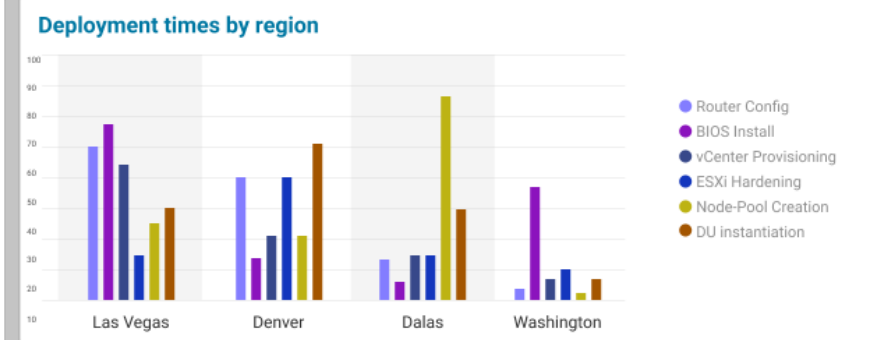
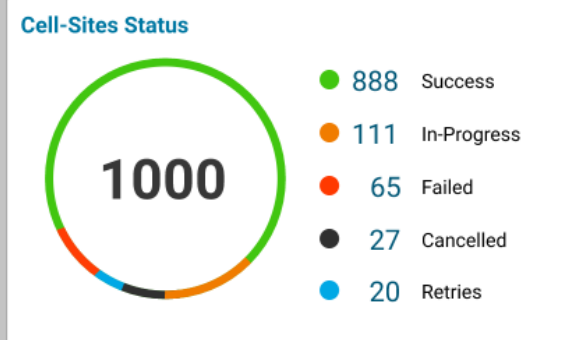
ZTPM Dashboard Report

Filter Group ⌵ 🕒 Week ⌵

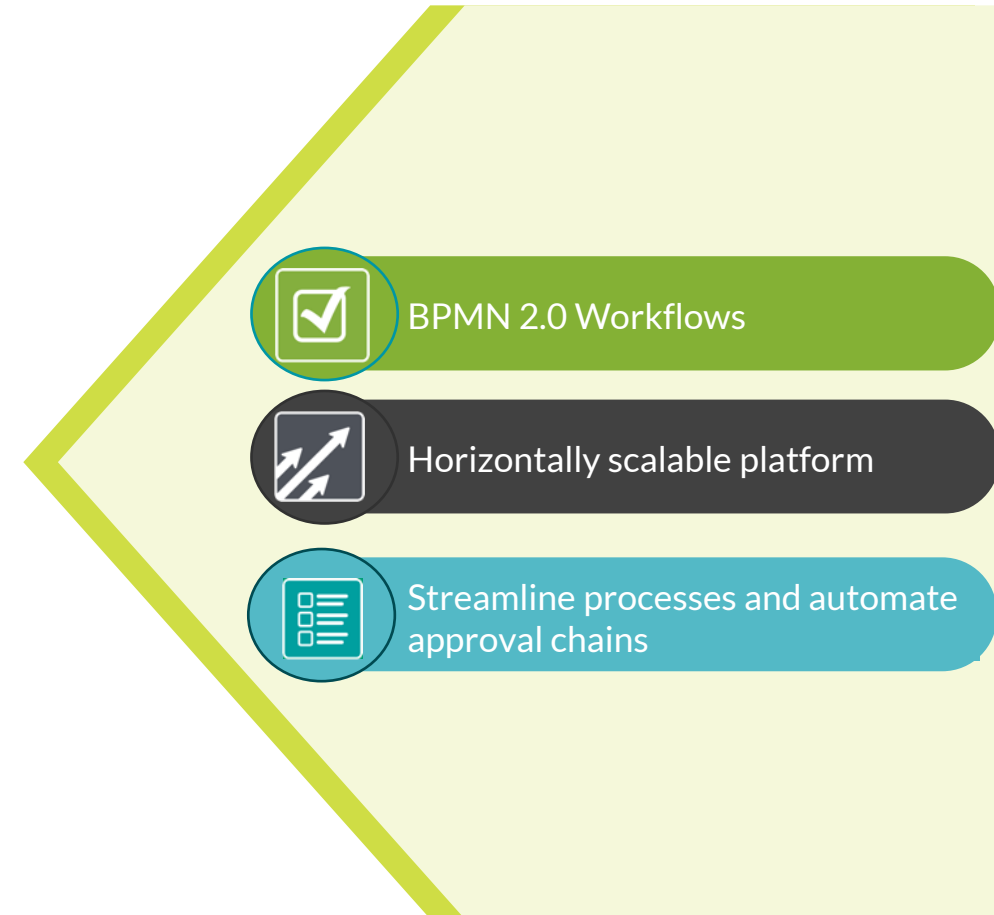
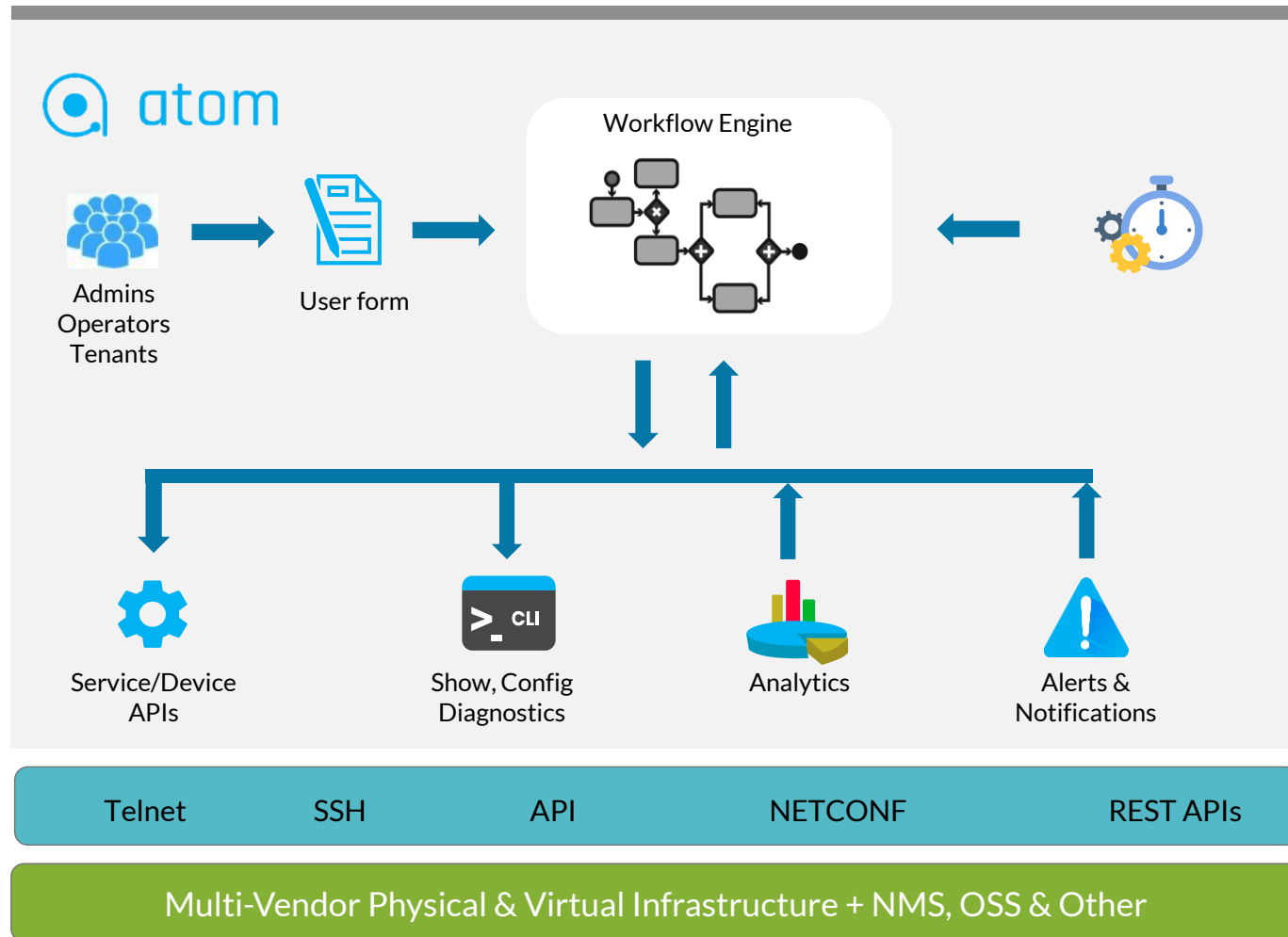
Average Deployment times per Cell-Site  00:02:11 Total	00:01:34 Automation 00:00:58 Manual	SLA Cell-Site  90% 00:01:54 Time saved Failed 65 Passed 900	Failed Activities  22 Cell-Sites 43 Failed Stages	Router Config SLAs  90%  00:02:11 Average deployment time	BIOS Install SLAs  88%  00:02:22 Average deployment time	vCenter Provisioning SLAs  89%  00:03:11 Average deployment time	ESXi Hardening SLAs  86%  00:01:11 Average deployment time	Node-Pool Creation SLAs  92%  00:00:11 Average deployment time	DU instantiation SLAs  90%  00:00:11 Average deployment time
---	--	--	---	---	--	--	--	--	--

Cell-Site Stages

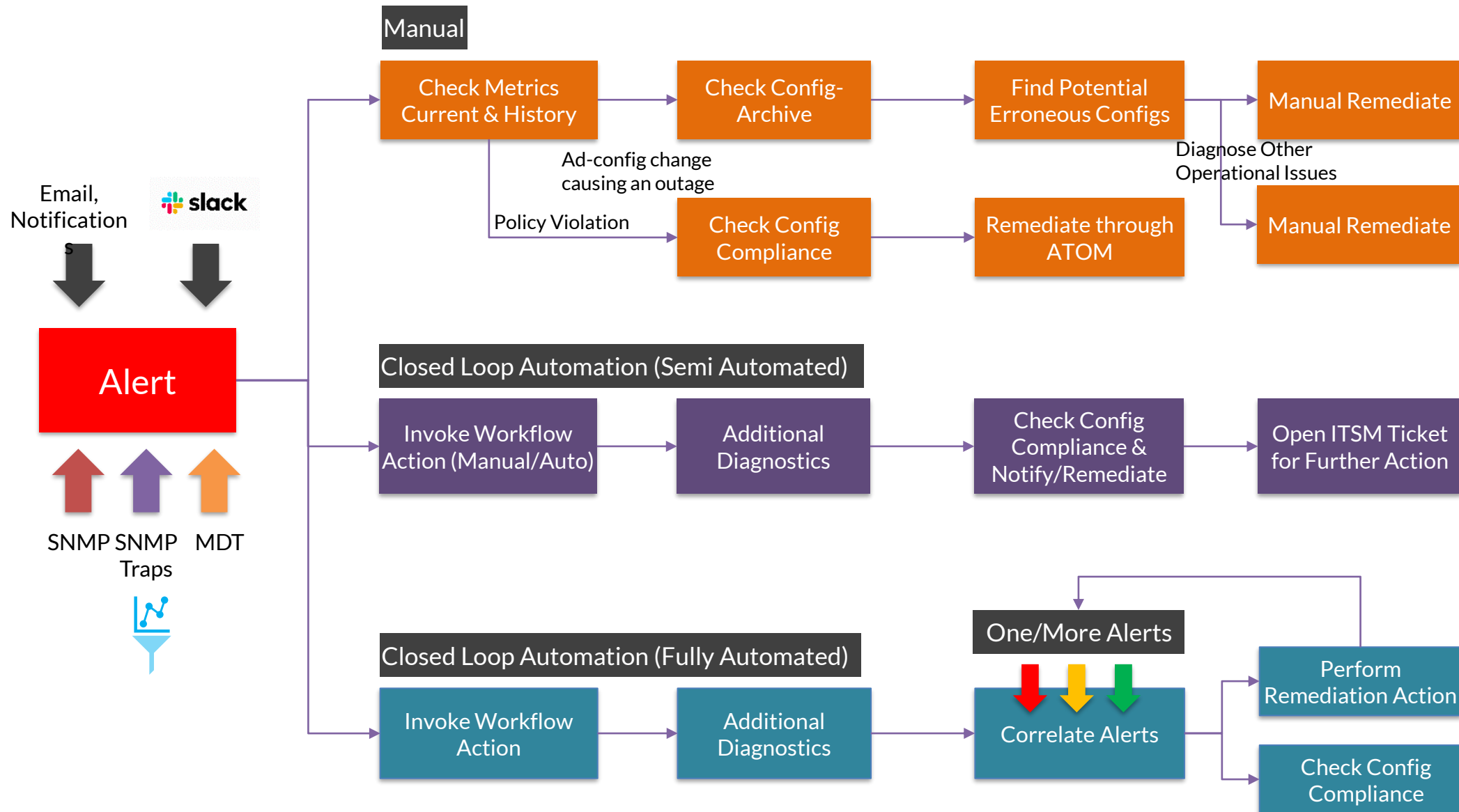
	Success	In-Progress	Failed	Cancelled	Retries
Router Config	148	18	22	3	12
BIOS Install	148	36	0	2	0
vCenter Provisioning	156	0	23	1	0
ESXi Hardening	140	21	0	0	0
Node-Pool Creation	108	28	0	10	6
DU instantiation	188	8	20	11	2



Bridging the gaps with low-code automation from Anuta ATOM



Closed Loop Automation Scenarios



The background features a night cityscape with a green overlay. A network diagram is overlaid on the city, consisting of numerous vertical lines with glowing green dots at their bases, and a complex web of white lines at the bottom representing a network structure. The text 'Thank you' is centered in the green area.

Thank you

JUNIPER | Engineering
NETWORKS | Simplicity