itential

HEAVY READING
WHITE PAPER

End-to-End Network Automation:

How to Develop Your Network Automation Strategy

A Heavy Reading white paper produced for Itential

Author: James Crawshaw, Senior Analyst, Heavy Reading

INTRODUCTION

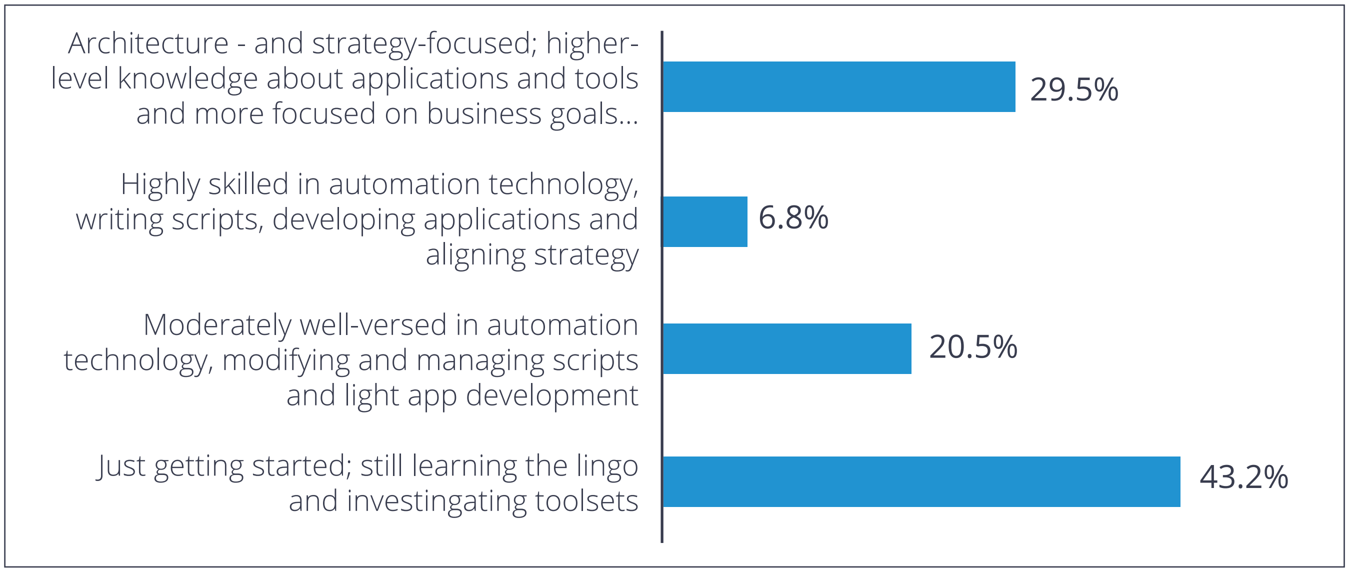
As Jason Edelman, et al., describe in Network Programmability and Automation: "Network automation is about simplifying the tasks involved in configuring, managing, and operating network equipment, network topologies, network services, and network connectivity."

Network automation use cases include:

- Device provisioning – using templates (such as Ansible or Salt) with configuration parameter files (written in languages such as YAML) to generate consistent config files
- Data collection – using open source tools to pull the data you want from network devices, not just what your existing monitoring tools allow
- Configuration validation – checking security compliance and automatically triggering remediation if there is a violation
- Troubleshooting – automating the analysis of fault and performance data to find root causes more quickly and reliably, and to predict future issues

There are many different components involved in network automation, including application programming interfaces (APIs), operating systems, scripting languages, data modeling languages, templating languages, system admin tools, and new DevOps methodologies. As noted by David Barroso, creator of NAPALM (Network Automation and Programmability Abstraction Layer with Multivendor support – a Python library for managing network devices): "Network automation is not just a single discipline; it is a collection of protocols, tools, and processes that can be overwhelming to the uninitiated." Unsurprisingly, as Figure 1 suggests, few network professionals today claim to be highly skilled in automation technologies.

Figure 1: How Do You Rate Your Own Skillset Re Network Automation Technologies?



Source: Light Reading webinar survey Mar 22, 2018 (n=44)

Automation is, of course, not an entirely new concept in networking. Any operations engineer worth their salt has been doing some form of automation, such as writing scripts, for years. While this type of narrow-scope automation is worthwhile, it is insufficient for complex networks.

Automating a discrete task has a marginal benefit, and may even introduce some static behavior that could have a negative impact on a dynamic system.

As service providers and enterprises start to expand their automation strategies, they rapidly move from scripting to orchestration. But this too can have its limitations, and more importantly can entail significant investment. This paper outlines an alternative approach to developing a network automation strategy, one that is devised top-down, based on user stories and workflows, rather than bottom-up, based on the tools and systems needed to automate such workflows.

TOP-DOWN VS. BOTTOM-UP APPROACH TO AUTOMATION

Top-Down Beats Bottom-Up

Top-down and bottom-up are contrasting methods for designing systems for the management of people, networks and countless other fields. With a top-down approach, we first construct an overview of the system and then specify the first layer of subsystems. Each subsystem is then defined in more detail, with lower-level components specified until we reach a useful level of atomicity. Essentially, the top-down approach starts with the big picture and then deconstructs it into finer elements.

In contrast, a bottom-up approach starts with subsystems and bolts them together to form more complex systems. Nature tends to take a bottom-up approach to system design, as it has the luxury of long time scales and natural selection to form good systems. Conversely, engineering tends to take a top-down approach. A top-down approach will often lead to a better outcome than bottom-up, as it optimizes the overall solution to meet a global objective, rather than individual components, which only optimize at a local level.

Although network teams are steeped in engineering talent, when it comes to developing a network automation strategy they often inadvertently end up taking a bottom-up approach, because they may have already made choices about subsystems such as network devices, controllers and orchestrators before they begin to think about automation more holistically. Network teams often start their engineering projects by selecting networking equipment; then add some management and automation systems on top (often from the equipment vendor); and then take those automated systems and look for ways to stitch them together.

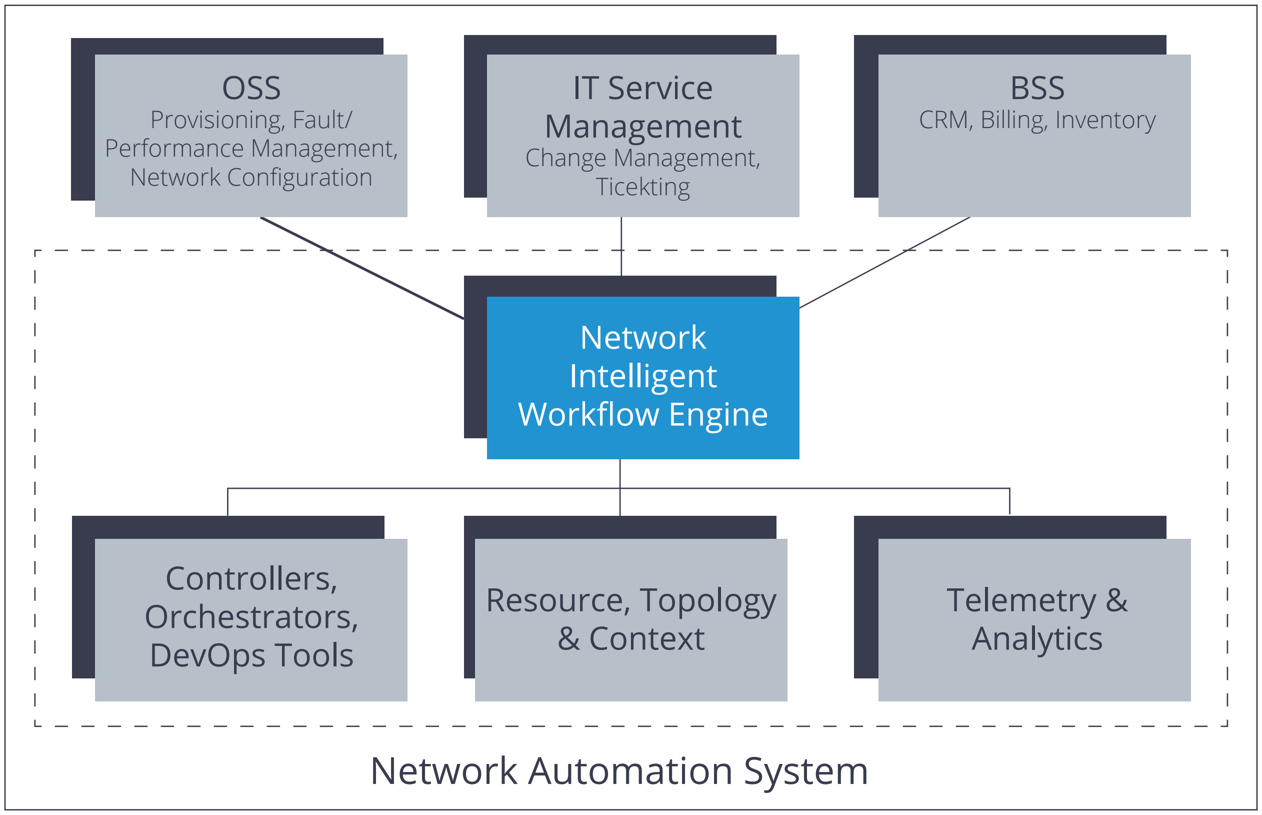
Deferring the more important strategic decisions about the automation strategy can often come back to haunt automation projects, leading to compromises having to be made or projects simply getting bogged down in analysis paralysis. By taking a bottom-up componentized view, you don't understand what you are automating until you have finished. As a result, you may end up regretting decisions made earlier (at the instrumentation level) in your project because you end up with suboptimal automation.

Top-Down Approach Requires an Intelligent Workflow Engine

The top-down approach requires an intelligent workflow engine that allows engineers and designers to visualize the workflows and user stories they are trying to automate. An intelligent workflow engine could be a whiteboard, a Visio document or a domain-specific software application. While generic workflow tools exist, a domain-specific tool will understand industry-specific concepts such as topology and technologies such as YANG.

Figure 2 shows where an intelligent workflow engine would sit within a network automation framework.

Figure 2: Network Automation Framework


OSS
Provisioning, Fault/ Performance Management, Network Configuration
IT Service Management
Change Management, Ticekting
BSS
CRM, Billing, Inventory
Network Intelligent Workflow Engine
Controllers, Orchestrators, DevOps Tools
Resource, Topology & Context
Telemetry & Analytics
Network Automation System

Source: Heavy Reading

Northbound integrations are key to providing the benefits of automation to the entire company, not just operations and engineering. The workflow engine connects northbound to:

- IT Service Management (ITSM) systems (i.e., business process management tools that \ automate human processes)
  - Ticketing and change management systems such as ServiceNow, Remedy and ManageEngine
  - Inventory and configuration management databases, including IP address management
- Operations support systems (OSS) such as provisioning, fault/performance management and network configuration
- Business support systems (BSS) such as CRM, billing and inventory

Southbound, the workflow engine connects to:

- Controllers (SDN, SD-WAN, etc.)
- Orchestrators
- Telemetry, monitoring and analytics systems

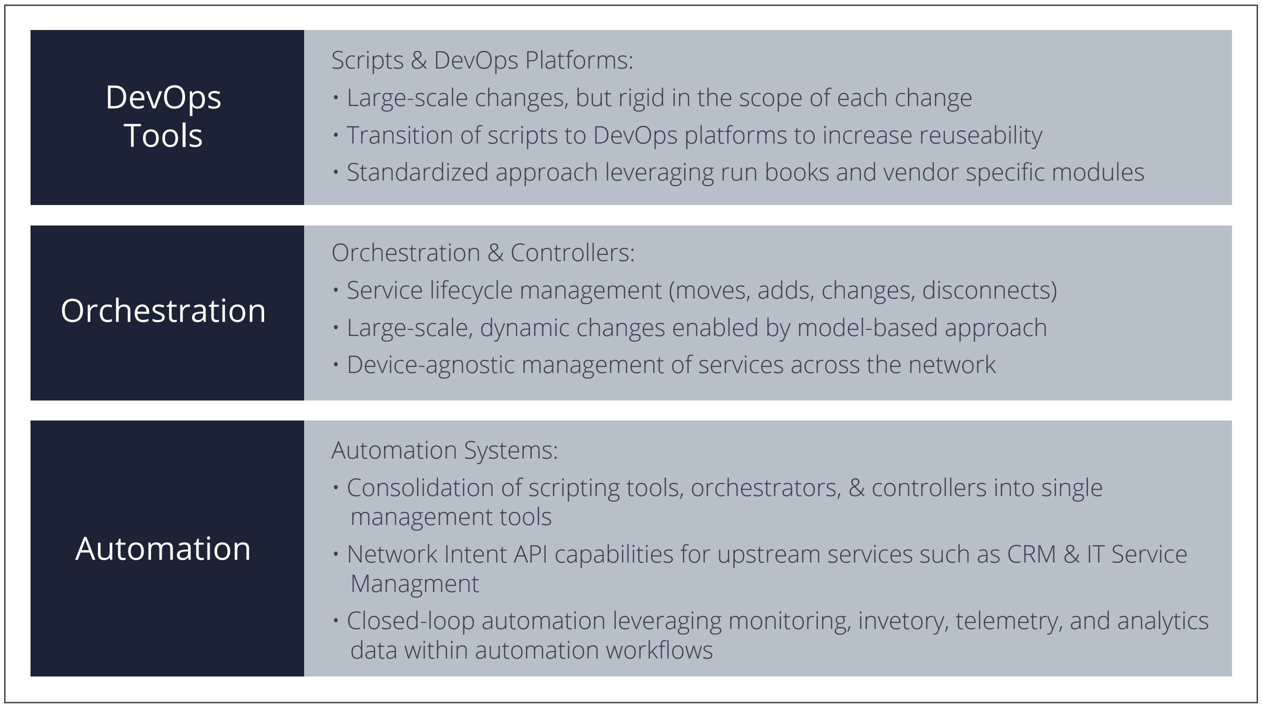
Naturally, the links shown in the diagram above are not the only ones; orchestrators will take feeds from telemetry and analytics applications and read/write to OSS, BSS and ITSM. However, domain-specific orchestrators may not have sufficient oversight to manage the automation of services that span multiple domains. For example, to automate a service turn-up might require the use of separate controllers for a firewall, a router, and an SD-WAN system, all of which need to be integrated.

Even multi-domain orchestrators may fall short in their ability to automate processes end-to-end. As network automation evolves, it should not be tied to a specific application, controller or orchestrator, but should become a platform that is deployed horizontally across the network, end to end. As this platform accesses multiple data sources, it can become a powerful agent of change, breaking traditional boundaries inside operators between networking and IT, and fostering collaboration. Initial efforts may target simple use cases such as TCP optimization, backhaul management or customer care. Once the operator is confident of the new approach, the platform can be expanded to more complex use cases that leverage the same data sources.

Automation Is More Than Orchestration

For a long time, network teams have used scripting for device-level management (see Figure 3). Over time, they progressed from device-level to service-level management with orchestration tools that handle any state-dependent use case in the service lifecycle. But to truly achieve automation, we need to reach end-to-end closed-loop management of services, from the opening to the closure of a ticket, across multiple different devices. In some cases, such end-to-end automation might be overkill, and device-level management with scripting will suffice. There is no one-size-fits-all answer. Network teams must look for the right technology to address their particular environment and challenges.

Figure 3: DevOps, Orchestration & Automation Use Cases



DevOps Tools	Scripts & DevOps Platforms: • Large-scale changes, but rigid in the scope of each change • Transition of scripts to DevOps platforms to increase reuseability • Standardized approach leveraging run books and vendor specific modules
Orchestration	Orchestration & Controllers: • Service lifecycle management (moves, adds, changes, disconnects) • Large-scale, dynamic changes enabled by model-based approach • Device-agnostic management of services across the network
Automation	Automation Systems: • Consolidation of scripting tools, orchestrators, & controllers into single management tools • Network Intent API capabilities for upstream services such as CRM & IT Service Managment • Closed-loop automation leveraging monitoring, invetory, telemetry, and analytics data within automation workflows

Source: Heavy Reading

Often organizations will start an automation project by implementing a comprehensive network orchestration platform. Sometimes, once they have automated and instrumented the various workflows, they realize that they could have achieved 80 percent of the functionality with an open source tool such as Ansible for a lot less money. Only 20 percent of the use cases required the functionality of the networking-specific orchestration platform.

Orchestration facilitates network service lifecycle management, but does not include all the operational components that contribute to operational cost, such as finding ports, cards and racks, or coordinating a change control to upgrade a branch. Equally, ITSM tools can push change requests to the network, but there are many other processes that need to be executed. An Intelligent Workflow Engine automates activities outside the scope of orchestration and ITSM, such as cross-domain (network/IT) coordination activities through API federation. A lot of the cost associated with automation projects relates to integration with OSS and IT. With a federated layer of APIs, this integration can be done once and reused multiple times.

In practice, organizations often end up with a multiplicity of orchestrators and controllers in their networks driven by distinct use cases, e.g., IP networking, SD-WAN and data center networking. If these tools are from different vendors, this can lead to a significant integration tax for the IT department. By using an intelligent workflow engine that sits above the orchestration layer, organizations can avoid this manual integration effort.

Workflow Documentation Is Key

Network teams should start their automation projects by planning their workflows and processes first, and then selecting the tools (including orchestration) that are required to achieve these goals, rather than the other way around. For example:

- If you need to maintain a service lifecycle, then yes, you need a robust tool such as a controller
- If you are doing something simple, such as software provisioning, configuration management or application deployment, then use Ansible
- If you have previously written code, then a Python script may suffice

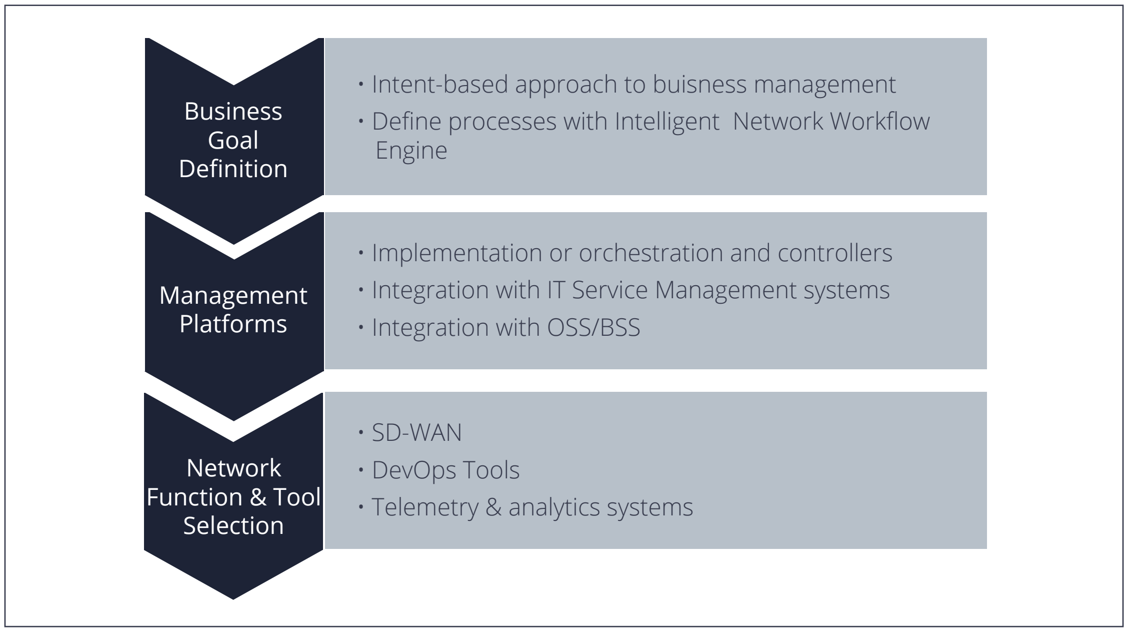
User stories should be the guide as to which tools are selected downstream of the workflow engine. Organizations must understand the existing, often manual, workflows before attempting to automate. As such, engineers must document these workflows properly, identify interdependencies before making changes with inadvertent consequences, and redesign them to take advantage of programmability such that the new workflow is better than the traditional, manual process.

Automation projects often start by looking at the existing way of doing things manually and then trying to automate each step. That avoids the cultural challenge of changing procedures and allows the engineers to maintain control. The next step is to look for new ways to add value or functionality that wasn't possible with manual processes.

Intent-Driven Automation

The workflow design should start with high-level business "intent" and then work its way down to more granular specifications. Usually that is how automation projects start out, with the executive sponsor (budget holder) asking to reduce network operations costs or enable a new business model of dynamic enterprise services (on-demand, self-serve, pay as you go, etc.). Often, however, engineering then starts building the solution from the bottom up (or perhaps the middle), based on an orchestration platform they have chosen or are evaluating. Instead, they should continue the top-down process step by step, getting closer to the network gradually.

Figure 4: Intent-Driven, Top-Down Approach to Automation

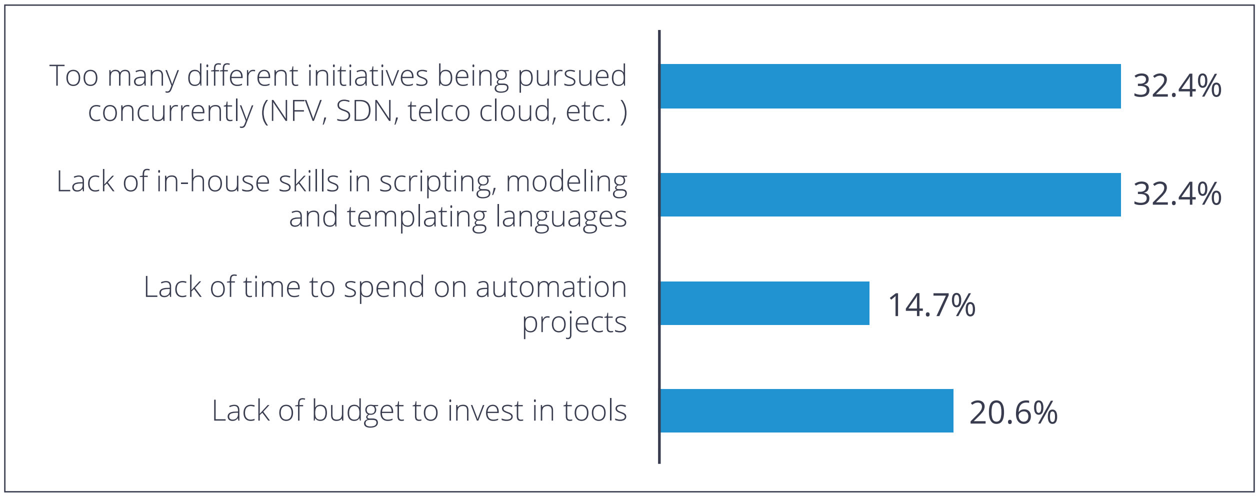

Business Goal Definition
• Intent-based approach to buisness management
• Define processes with Intelligent Network Workflow Engine
Management Platforms
• Implementation or orchestration and controllers
• Integration with IT Service Management systems
• Integration with OSS/BSS
Network Function & Tool Selection
• SD-WAN
• DevOps Tools
• Telemetry & analytics systems

Source: Heavy Reading

AUTOMATION JOURNEY

As **Figure 5** suggests, several factors are hampering network professionals' automation initiatives today, including a lack of time, skills and budget.

Figure 5: What Is the Key Thing Holding Back Your Network Automation Today?



Source: Light Reading webinar survey Mar 22, 2018 (n=44)

Automation is not binary; it is a journey. But unless network engineers document their workflows, how can they evaluate their degree of automation? An operator might have automated all of its command-line interface (CLI) inputs, only to realize that these repre-sent just 10 percent of a multi-step configuration change process. For example, a Layer 2 VPN provisioning process could be automated with an orchestrator, but engineers may still need to manually open a firewall port and update an inventory system. Once we have a holistic understanding of our end-to-end processes, we must decide what degree of automation is appropriate. For example, the last 25 percent to reach closed-loop or zero-touch automation may not be cost-effective or reliable.

Service provider networks are complex. With so many variables, zero-touch automation presents a significant challenge. As Murphy's law states: "Anything that can go wrong will go wrong." Humans are on hand to deal with unexpected exceptions and keep the networks running while trying to minimize the impact on customer experience.

Automating manual back-office tasks within a telecom operation with a technology such as robotic process automation (RPA) is straightforward. And automating low-level functionality in the network is simple with scripts. But when you try to automate a more complex service lifecycle, you get complex feedback loops with many interactions, which makes simple, rules-based systems hard to implement. Today we still require a human that can oversee the service lifecycle and guide the system down the right path when it gets stuck or confused. Perhaps we can reliably automate 80 percent of the service lifecycle, but in 20 percent of cases human intervention is required.

With the introduction of machine learning, every time the human intervenes to guide or override the automation system, this should be fed back into the algorithm to improve it. Learning is a continuous process, both to optimize existing workflows, and to keep up with changes in the network. The number of process variations is so large, it will likely take many years before we converge on an automation level of 99 percent. Zero-touch automation is perhaps an asymptotic goal.

Benefits of Automation

The benefits of network automation are clear: accelerating operations and reducing errors. Automation reduces the time needed for deployments and configuration changes, so that service providers can be more responsive to customers. Automation is also about reducing errors that impact customer experience.

Around a quarter of customer-care issues end up being directed to engineering and operations staff (the bulk of enquiries are billing- and plan-related). These technical teams spend a lot of time trying to resolve issues, using multiple tools trying to find the root cause, and work out how to fix it. Such manual troubleshooting is time-consuming, which is frustrating for the customer. By automating many of the data-gathering elements of such troubleshooting procedures, service providers and enterprises can shorten the time to resolution and achieve greater consistency.

Different engineers working on the same network will often implement changes via CLI in slightly different ways, and if they are copying and pasting changes, it is easy to introduce an error. By automating, we can make configuration changes more consistent and reduce the scope for human errors. Fewer errors translate to lower operating cost, but automation can also lead to lower capital expense, as it can increase the reliability of the network, so network planners don't have to rely on massively overprovisioning to compensate for uncertainty.

CONCLUSION

The telecom industry has been on a journey from manual operations to automation for more than a century, from electromechanical to electronic, digital and software-based switching. What's changed recently is the increasingly dynamic nature of the network that comes with the move to virtualization. To support NFV, service providers' automation efforts need to move beyond the initial deployment phase to cover the entire lifecycle of network technology, including configuration changes, port changes, security, maintenance, testing, performance monitoring and, ultimately, decommissioning.

The key to developing a good automation strategy is to take a top-down approach. Invest time in documenting existing workflows with an intelligent tool, and then let those automation stories guide which technology components (controllers, orchestrators, etc.) to select, not vice versa. Focus on use cases rather than technology components:

1. Which groups (engineering/operations) will be impacted?
2. What systems are involved?
3. What is the service or goal?
4. How many integrations are required?

Start with your business intent (deliver a service, be more secure/compliant, reduce maintenance spending, etc.) and then build the workflows and technology components to support it. Do not obsess with automating one particular part of the process (e.g., CLI input) while forgetting to automate other, more onerous manual tasks that hold back agility and time to market.

Automation is an iterative and incremental process, not a monolithic solution. You will only learn the lessons of automation in your network environment through doing. Start with manageable automation projects, rather than trying to boil the ocean. Create an automation "on-ramp" with measurable benefits that leverages an integrated automation ecosystem and DevOps practices and tools. Treat your infrastructure as code and make a plan to automate it:

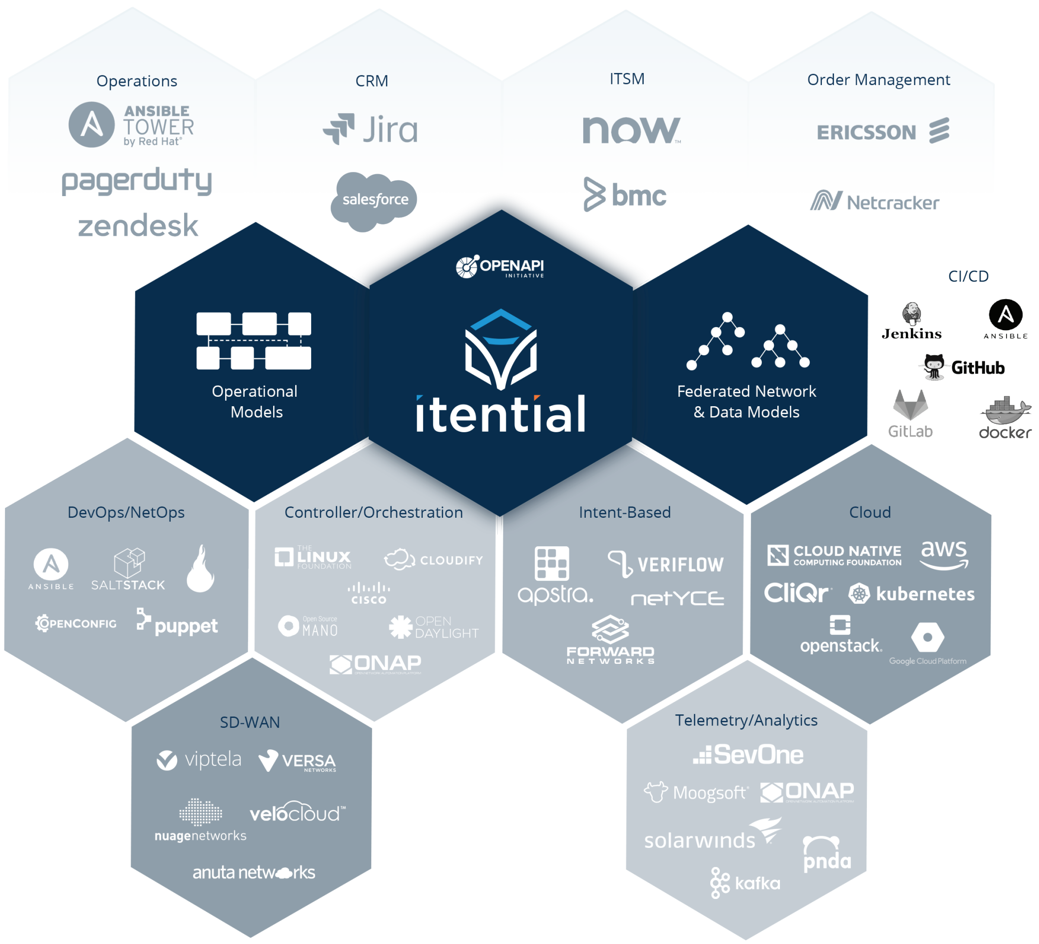
1. Where do we store configuration/metric data?
2. Where do we store code/models/scripts?
3. How do we manage code/configuration lifecycle?
4. How do we test?
5. How fast can we deploy updates?

ABOUT ITENTIAL

Itential simplifies and automates the journey toward the modern network and bridges the gap between IT and networking teams by enabling users to easily build, execute and visualize Network Intelligent Workflows. Itential's low-code environment provides a vendor agnostic, turnkey solution, connecting network orchestrators and controllers with IT Service Management applications and configuration tools to accomplish closed-loop network automation. Itential's products are in use today within some of the world's largest networks, including many of the top service provider and financial services companies throughout the U.S. and abroad.

Find out more at www.itential.com.

Figure 6: Itential's Network Automation Ecosystem



Source: Itential