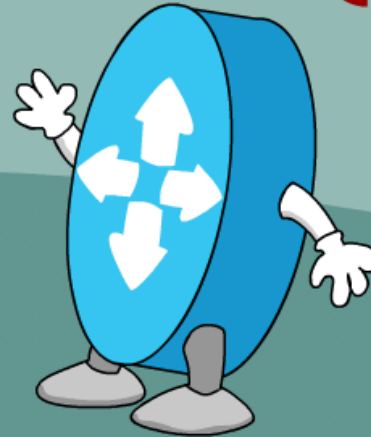




NETDEVOPS {LIVE!}



DEVNET

Is Your Network Working? Guess No More! with Python Testing Powered by pyATS & Genie

Simon Hart, @NetOpDevGuy

Technical Solution Architect, pyATS Genius, NetDevOps Evangelist

Siming Yuan @simingy

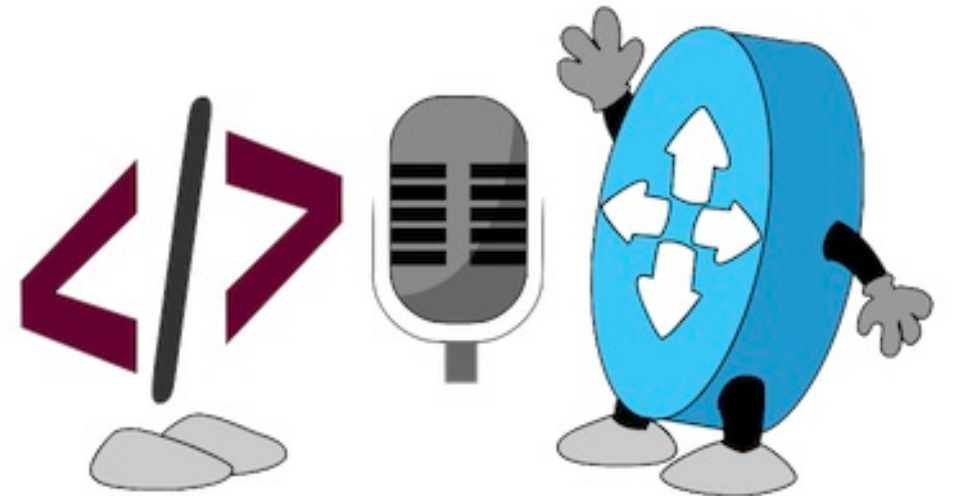
Technical Leader, pyATS Architect

Season 2, Talk 8

<https://developer.cisco.com/netdevops/live>

What are we going to talk about

- Recap: What the heck is pyATS & Genie?
- pyATS/Genie vs. <insert tool>
- Out-of-box feature models & platform support
- Stateful Network Validation:
 - CLI: No programming experience required
 - RobotFramework: English-like testcases
 - Raw, unadulterated Python



Recap: pyATS & Genie

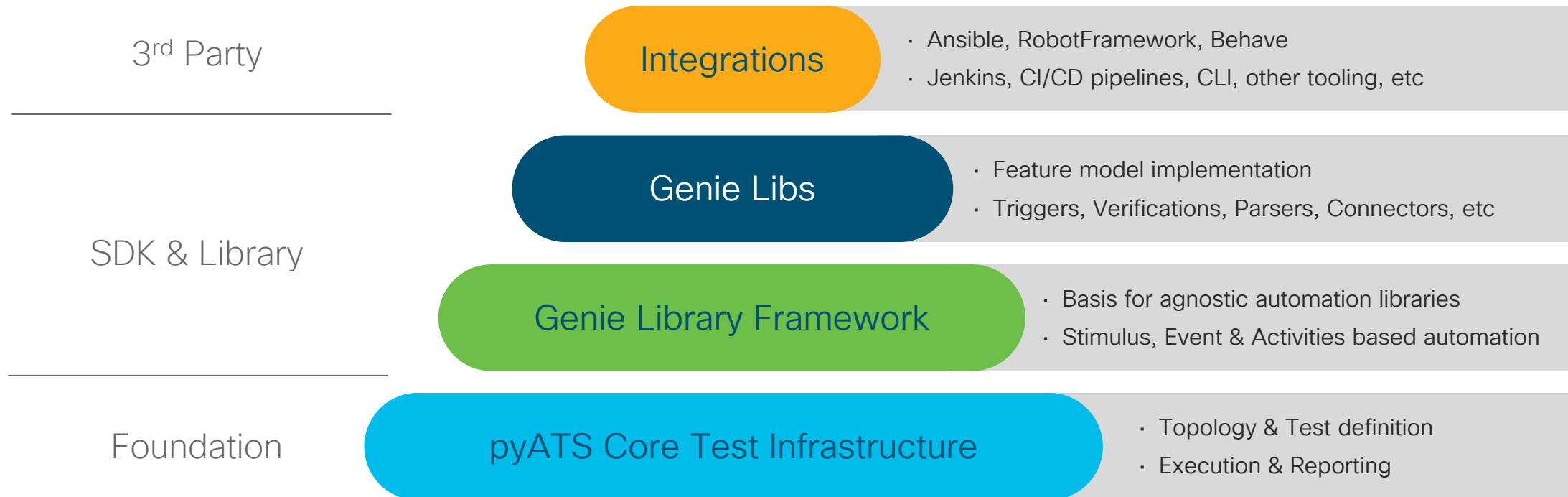


- De-facto Cisco engineering test infrastructure since 2014
 - CI/CD test automation across various platforms running IOS, IOSXE, IOSXR, NXOS, NGFW, Cable...
 - 3000+ Engineering developers, 5mil+ LoC, 2mil+ test runs per month
 - Active development & support by the same team that built the infrastructure
- Released to DevNet in 2017, all libraries open-source under Apache 2.0 License
- Full coverage of the infrastructure in NetDevOps Live Season 1 Episode 8
- The core of pyATS & Genie is platform, feature and management protocol agnostic
- Specific platform, device, and feature support are done via:
 - Plugins & extensions
 - Library implementations – loaded via an abstraction layer

Cisco Test Automation Ecosystem



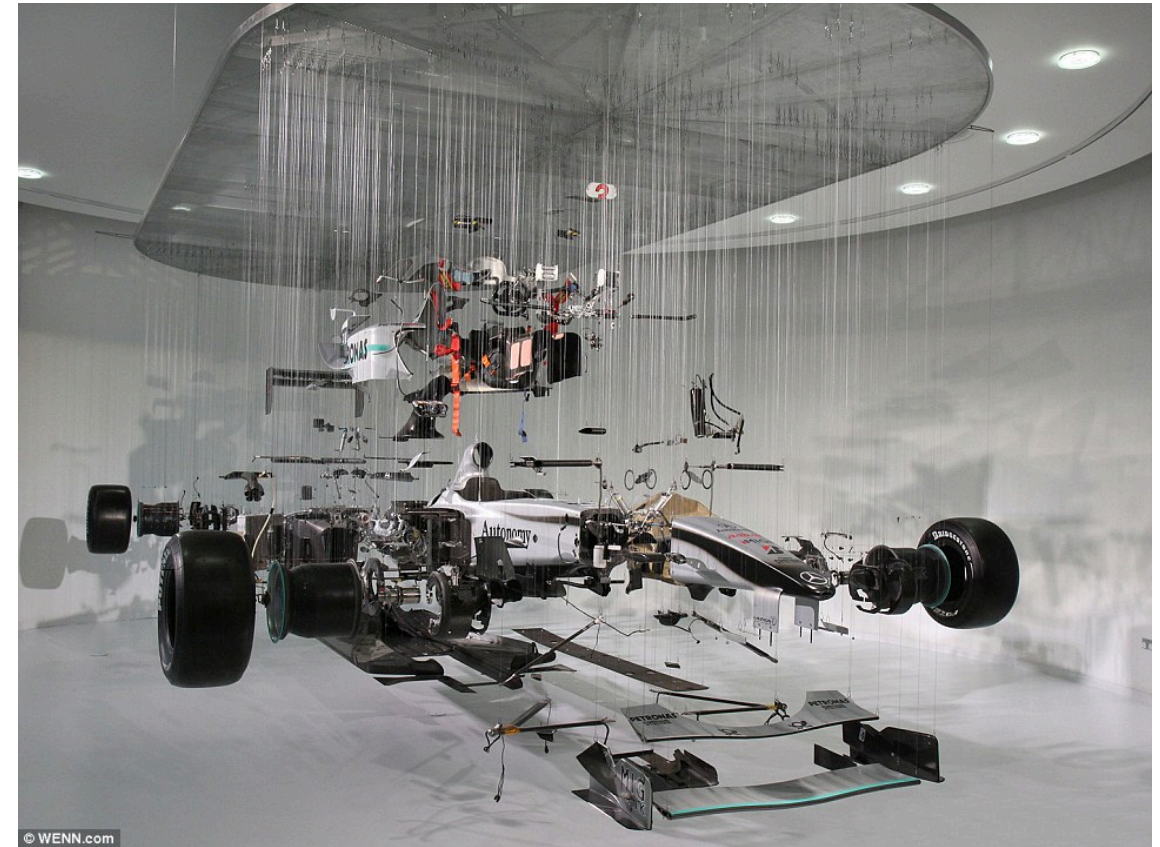
Solution Layout



Why not <insert-tool-here>?

- **unittest, pytest & nose**: test framework
 - Testcase definition & execution
 - Geared more towards unit testing (small, quick tests with independent setup/teardown)
- **Ansible, Chef, Puppet**: provisioning, deployment & management
- **RobotFramework, Behave**: ATDD/BDD specific solution, English-like statements
- **Netmiko, Paramiko, Ncclient**: connection protocol implementations
- **Textfsm**: CLI scraping infra

“Specialized Components”
vs.
“End-to-End Solution”



© WENN.com

Infrastructure “Mapping”

Function	pyATS Genie Component	Equivalent
Test Definition, Execution	AEtest	pytest, unittest, nose
Topology/Device Descriptor	Topology YAML	-
Connection + Services	Unicon, Netconf Connector	Paramiko, Netmiko, Pexpect (connection only)
Archive, Reporting	Easypy	-
Python Library	Genie Libs: ready-to use parser, models	Chef/Ansible/Puppet (cfg mgmt) NAPALM
Parser Framework	Genie.Parsergen, Genie.Metaparser	Textfsm Regex
Event-Driven, Data-Driven Reusable automation	Genie Triggers, Verifications	-
For Non-Programmers	Genie CLI	-

Open-Source Libraries

- The core of pyATS & Genie is light weight & platform, feature and protocol agnostic
- All platform, device, and feature implementations are open-source:
- Connection plugins & extensions
- Libraries – feature models, parsers, data-driven testcases are loaded via an abstracted library layer (eg: <https://github.com/CiscoTestAutomation/genieparser>)
- Example:
 - IOSXE (eg asr1k, cat9k)
 - Junos
 - NXOS (eg n9k, n7k)
 - NSO
 - IOSXR (eg asr9k)
 - NGFW (eg, asa)
 - ...

Environment

- Linux/macOS/WSL
- Python 3.4-3.7 (3.8 soon)
- Generates HTML+Text and Email Reports



```
bash$ python3 -m venv ~/pyats
bash$ source ~/pyats/bin/activate
(pyats) bash$ pip install pyats genie
(pyats) bash$ pyats --help
(pyats) bash$ genie --help
```

Out-of-Box Features

https://pubhub.devnetcloud.com/media/pyats-packages/docs/genie/genie_libs/#/

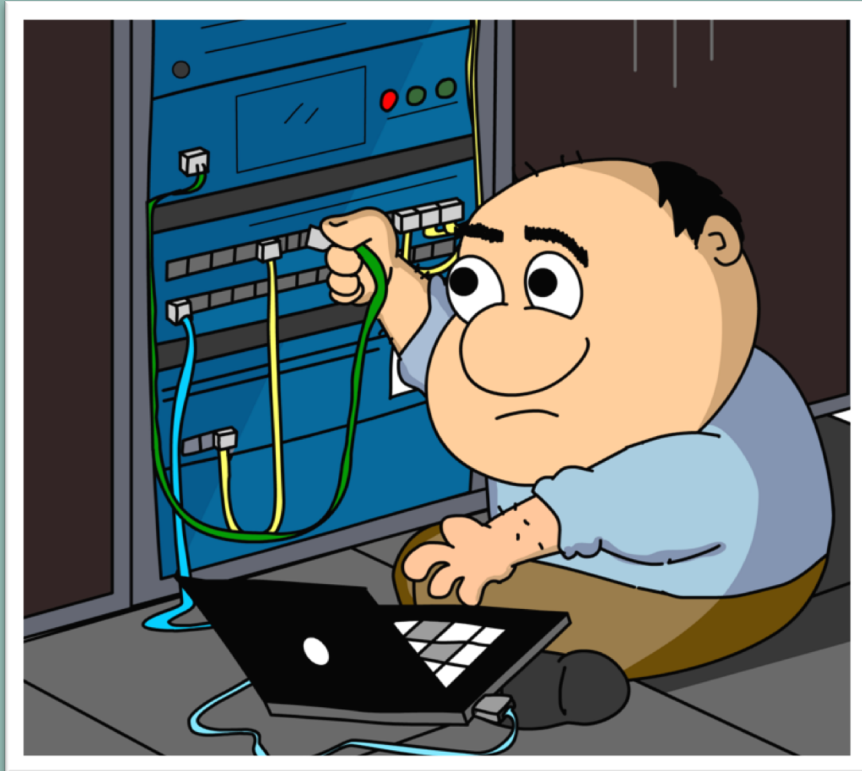
The screenshot displays the Genie web application interface. The top navigation bar is dark blue with the 'Genie' logo and a hamburger menu icon. The main content area is divided into three sections: a left sidebar, a central list of triggers, and a right sidebar for models.

Left Sidebar: Contains a hamburger menu icon and the 'Genie' logo. Below the logo, there are four menu items: TRIGGERS, VERIFICATIONS, MODELS, and PARSERS. The 'PARSERS' item is highlighted in blue.

Central List: A grid of 24 items, each starting with a green 'IOS' icon followed by a command. The commands are:
show ip interface
show ip interface brief {interface}
show ip mroute
show ip mroute vrf {vrf} static
show ip multicast vrf {vrf}
show ip ospf database external
show ip ospf database opaque-area
show ip ospf database summary
show ip ospf mpls ldp interface
show ip ospf neighbor detail
show ip ospf virtual-links
show ip pim interface
show ip interface brief
show ip interface brief | inclu
show ip mroute static
show ip multicast
show ip ospf
show ip ospf database netw
show ip ospf database route
show ip ospf interface
show ip ospf mpls traffic-enc
show ip ospf sham-links
show ip pim bsr-router
show ip pim interface detail

Right Sidebar: Titled 'Models List', it features a search bar with a magnifying glass icon. Below the search bar, the word 'all' is displayed in blue. A grid of 24 model names is shown, each with a small dark square icon containing the letter 'M'. The models are:
acl, arp, bgp, dot1x, fdb, hsrp, igmp, interface, isis, l2vpn, lag, lisp, lldp, mcast, mld, msdp, nd, ntp, ospf, pim, platform, prefix_list, rip, route_policy, routing, segment_routing, static_routing, stp, vlan, vrf, vxlan.

Demo



1. CLI:
No programming experience required
2. RobotFramework:
English-like testcases
3. Raw, unadulterated Python

Content available at:

https://github.com/RunSi/pyATS_GENIE_NetDevOpsLive

Summary

pyATS & Genie ...

- 1) provides an ecosystem for Network Engineers to perform test automation, DevOps and validation
- 2) is platform agnostic, free to use, and comes out of the box with ready-to-use libraries and network models, parsers and testcases
- 3) is suitable for Network Engineers with a variety of programming abilities and skillsets

Webinar Resource List

- pyATS & Genie landing page: <https://cs.co/pyats>
- Documentation: <https://developer.cisco.com/docs/pyats/>
 - Features: https://pubhub.devnetcloud.com/media/pyats-packages/docs/genie/genie_libs/index.html
 - Cookbook: <https://pubhub.devnetcloud.com/media/pyats-packages/docs/genie/cookbooks/index.html>
- Demo Content:
https://github.com/RunSi/pyATS_GENIE_NetDevOpsLive

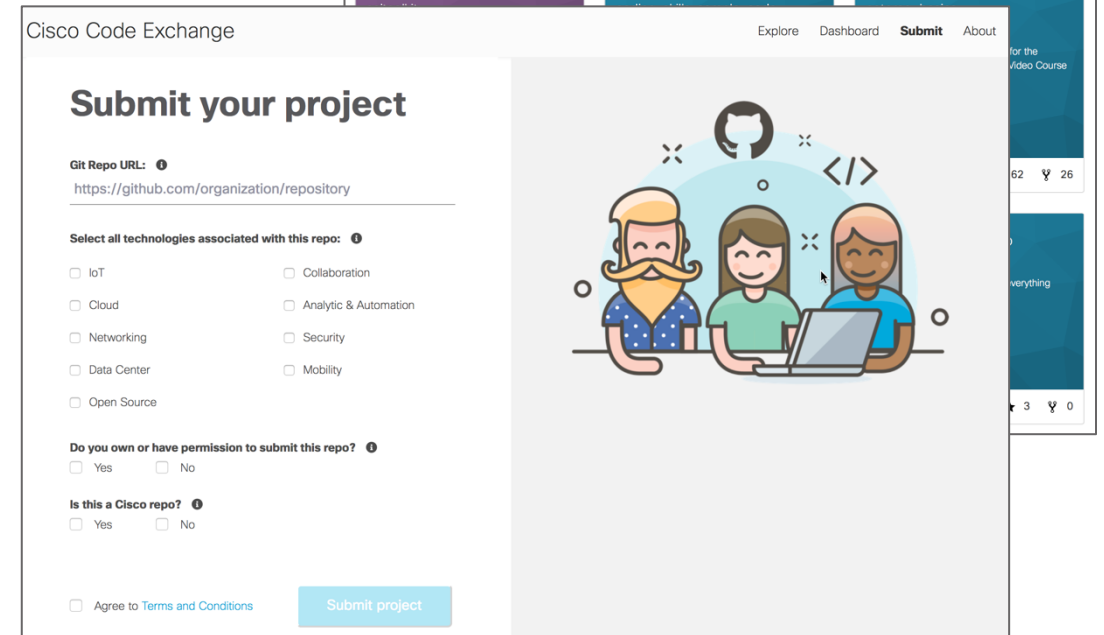
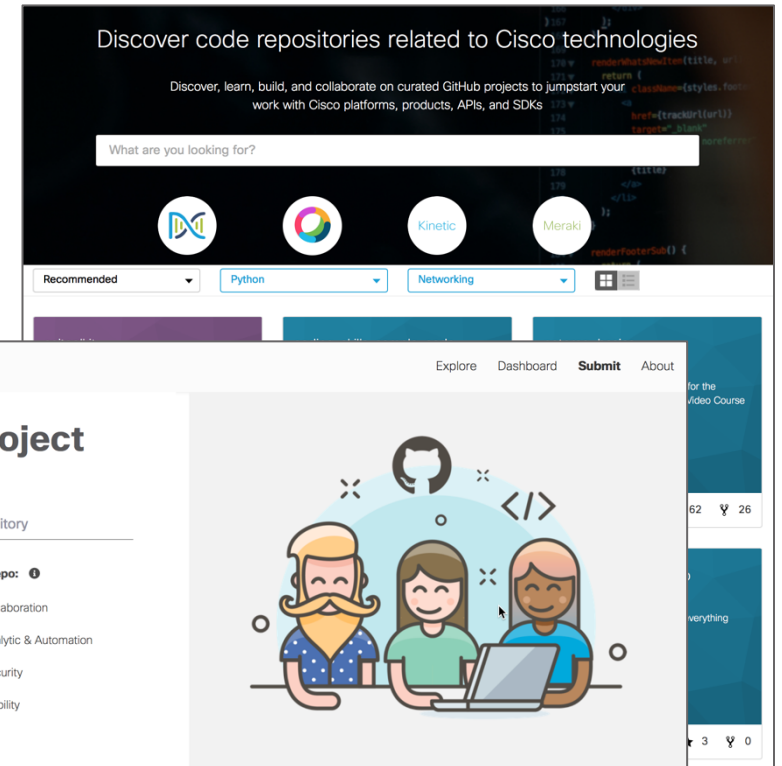
NetDevOps Live! Code Exchange Challenge

developer.cisco.com/codeexchange

Write your own Network Verifications

Examples:

- *Reachability Tests*
- *HSRP Status*
- *BGP/OSPF States*
- *Interface Counts/Counters*
- *Checks for critical routes*



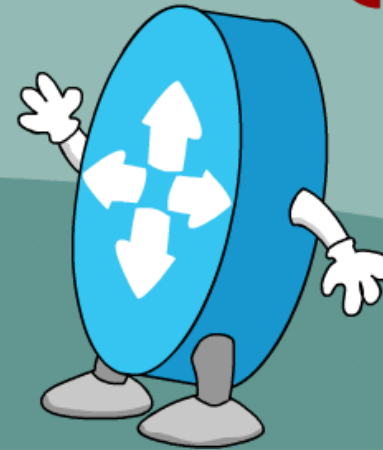
Looking for more about NetDevOps?

- NetDevOps on DevNet
developer.cisco.com/netdevops
- NetDevOps Live!
developer.cisco.com/netdevops/live
- NetDevOps Blogs
blogs.cisco.com/tag/netdevops
- Network Programmability Basics Video Course
developer.cisco.com/video/net-prog-basics/





NETDEVOPS {LIVE!}



DEVNET

<https://developer.cisco.com/netdevops/live>

@netdevopslive 