Exploring Practical AIOps Use Cases for Enterprise Networks with Splunk

Jason Shoemaker
Customer Delivery Architect, CX

Scott Lake
High Touch Engineering Technical Leader, CX

BRKOPS-2256

Cisco Webex App

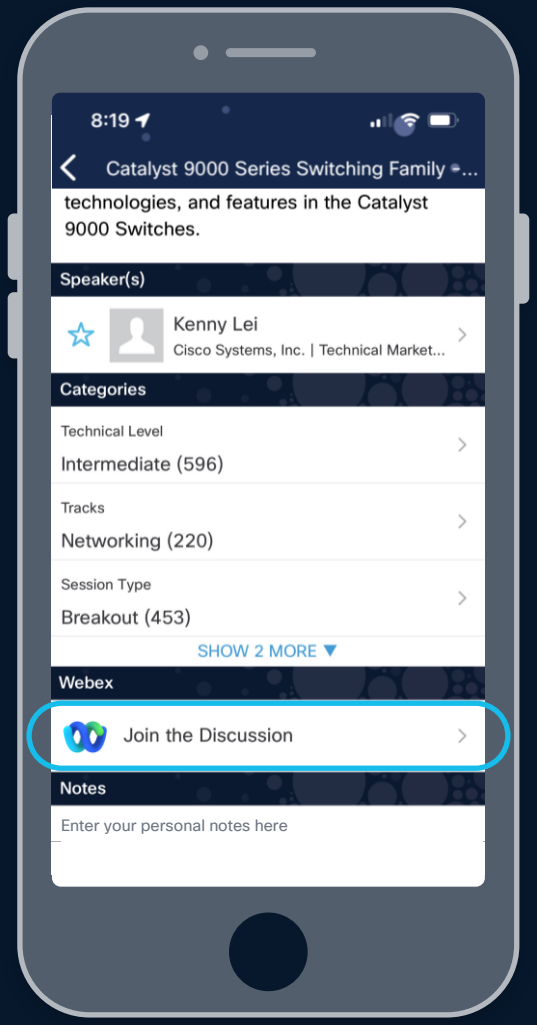
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

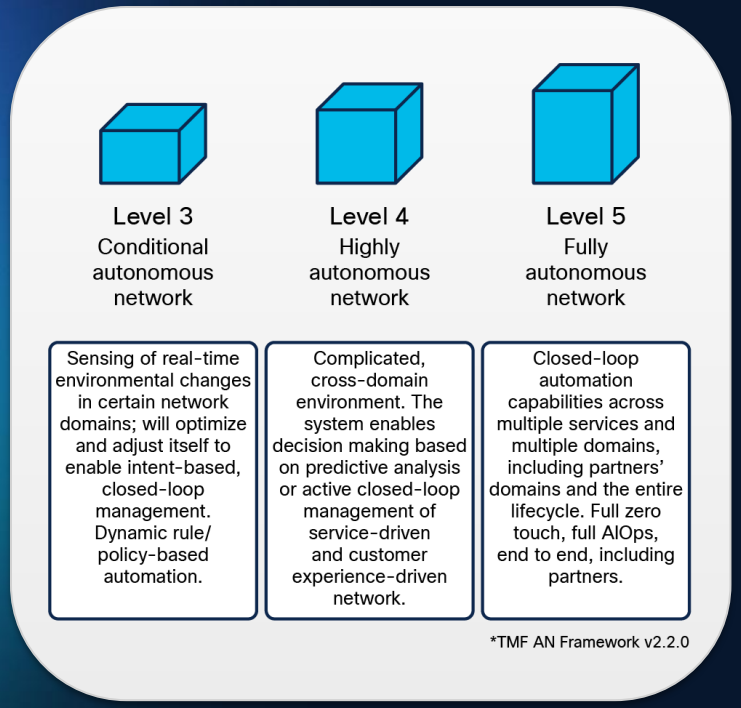
Webex spaces will be moderated by the speaker until June 13, 2025.



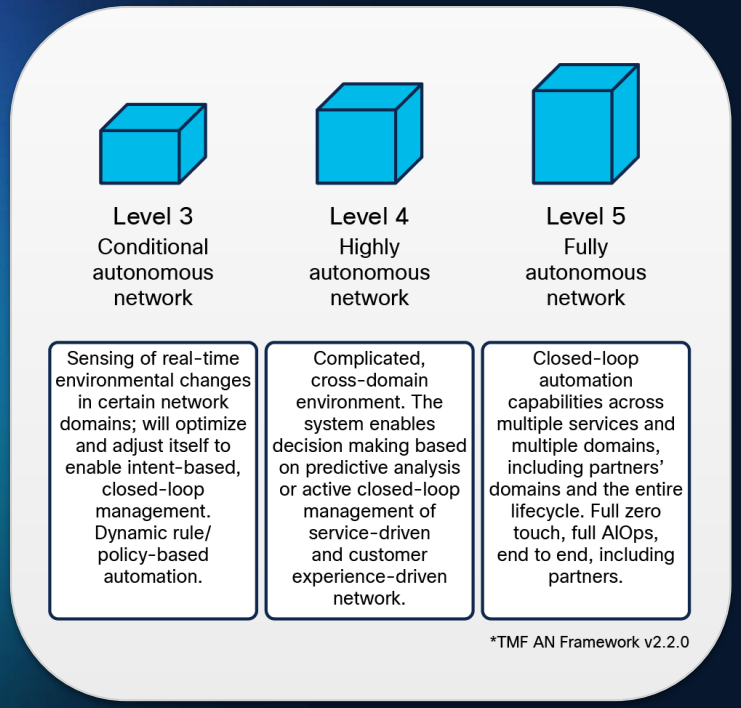
https://ciscolive.ciscoevents.com/
ciscolivebot/#BRKOPS-2256

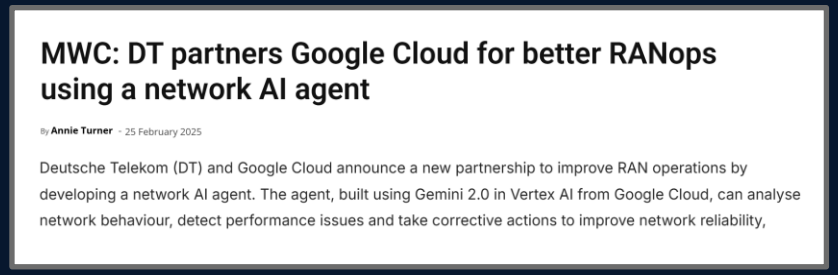
AIOps is driving the next generation of autonomous networks...

AIOps is driving the next generation of autonomous networks...

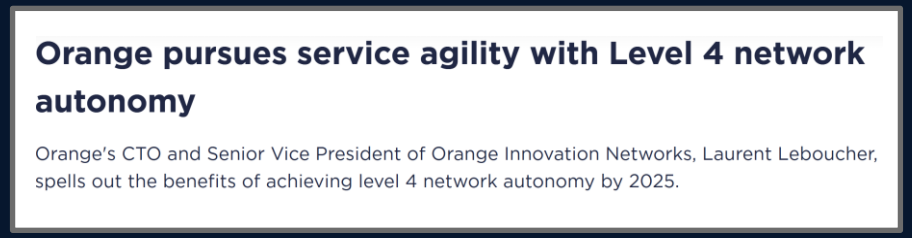


AIOps is driving the next generation of autonomous networks...

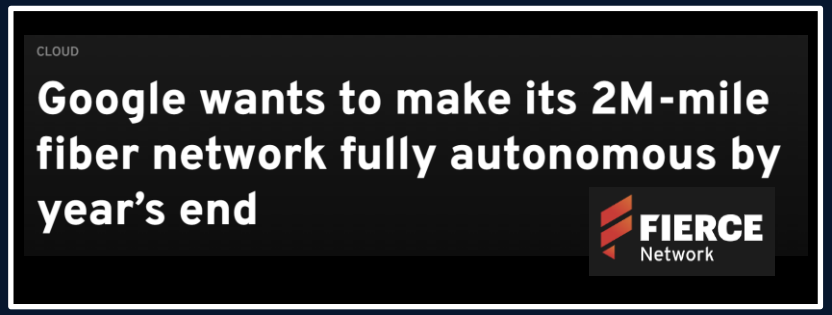




Source: Mobile Europe



Source: TMForum Inform



Source: Fierce Network

We are going from automation to autonomous.
We will have AI agents that run the network with no manual intervention.
We will be equal to TM Forum [Level] 5
by the end of the year.

Muninder Sambi, VP/GM Networking & Security, Google Cloud

AIOps is driving the next generation of autonomous networks...

AIOps is driving the next generation of autonomous networks...

...but what ultimately drives AIOps?

AIOps is driving the next generation of autonomous networks...

...but what ultimately drives AIOps?

Is it the data?

AIOps is driving the next generation of autonomous networks...

...but what ultimately drives AIOps?

~~Is it the data?~~

Is it the algorithms and models?

AIOps is driving the next generation of autonomous networks...

...but what ultimately drives AIOps?

~~Is it the data?~~

~~Is it the algorithms and models?~~

Is it the AI agents (our future robot overlords)?

AIOps is driving the next generation of autonomous networks...

...but what ultimately drives AIOps?

~~Is it the data?~~

~~Is it the algorithms and models?~~

~~Is it the AI agents (our future robot overlords)?~~

AIOps is driven by Y-O-U

Your **expertise**

Your **intuition**

Your **experience**

Agenda

01 A practical approach to AIOps

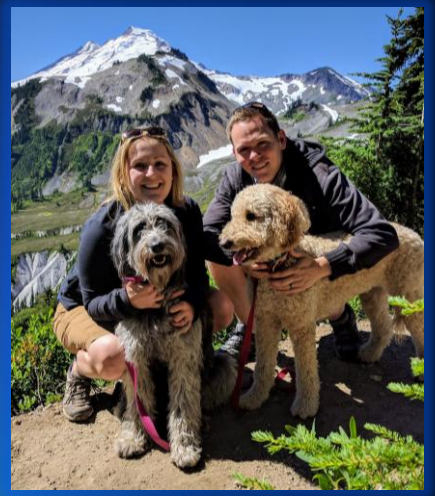
02 Phase 1: Model-driven telemetry

03 Phase 2: ML-driven insights

04 Phase 3: AI-driven decisions

05 Phase 4: YOU-driven innovation

About your speakers



Jason Shoemaker

- 14 years architecting solutions for Web Hyperscale companies
- DC Networking -> Network Automation -> Reliability Engineering
- Founder of CX Innovation Edge customer co-development initiative

Fun fact: Born the same year that Cisco was founded as a company

Scott Lake

- 27 years campus / data center experience (10 at Cisco)
- Hybrid background of network automation and reactive support
- Author of multiple CiscoLive Walk-in-Labs for MDT and ML w/Splunk

Fun fact: Piloted a tugboat once!

What to expect from this session

By network engineers.
For network engineers.

Emphasis on model-driven telemetry (no SNMP, syslog, etc)

A “take-home” package to experiment with your use cases

What NOT to expect from this session

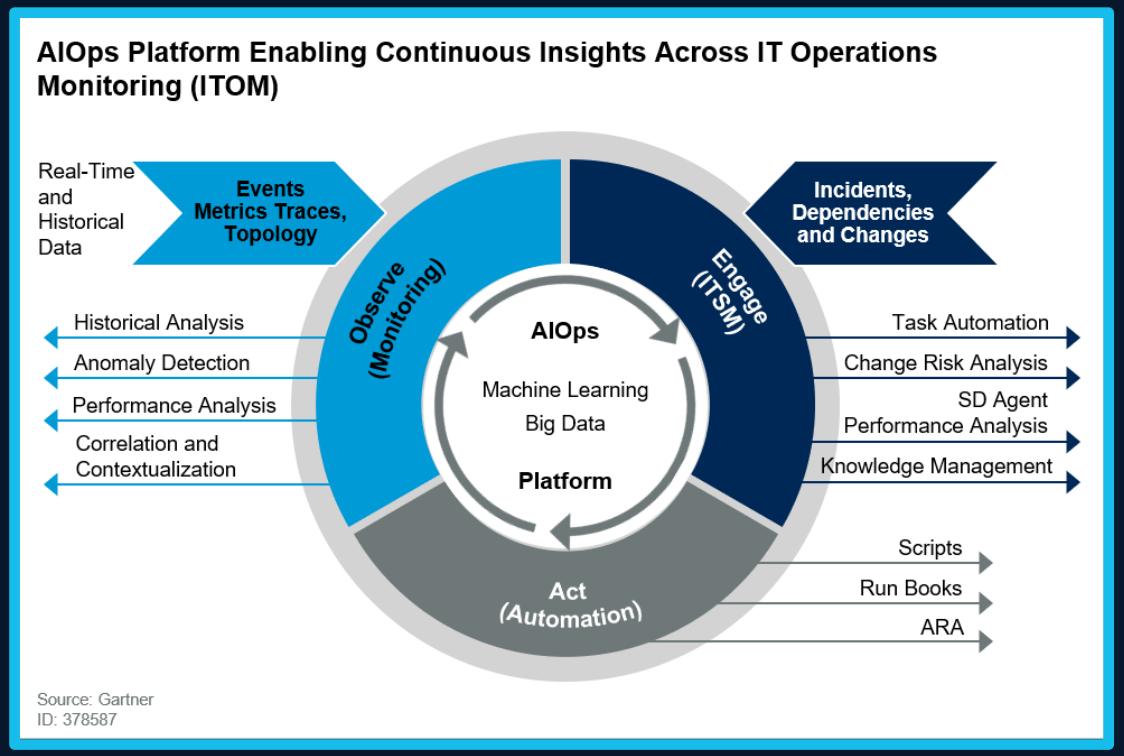
Production-ready solutions

Premium capabilities of Splunk

Emphasis on Cisco-specific tools

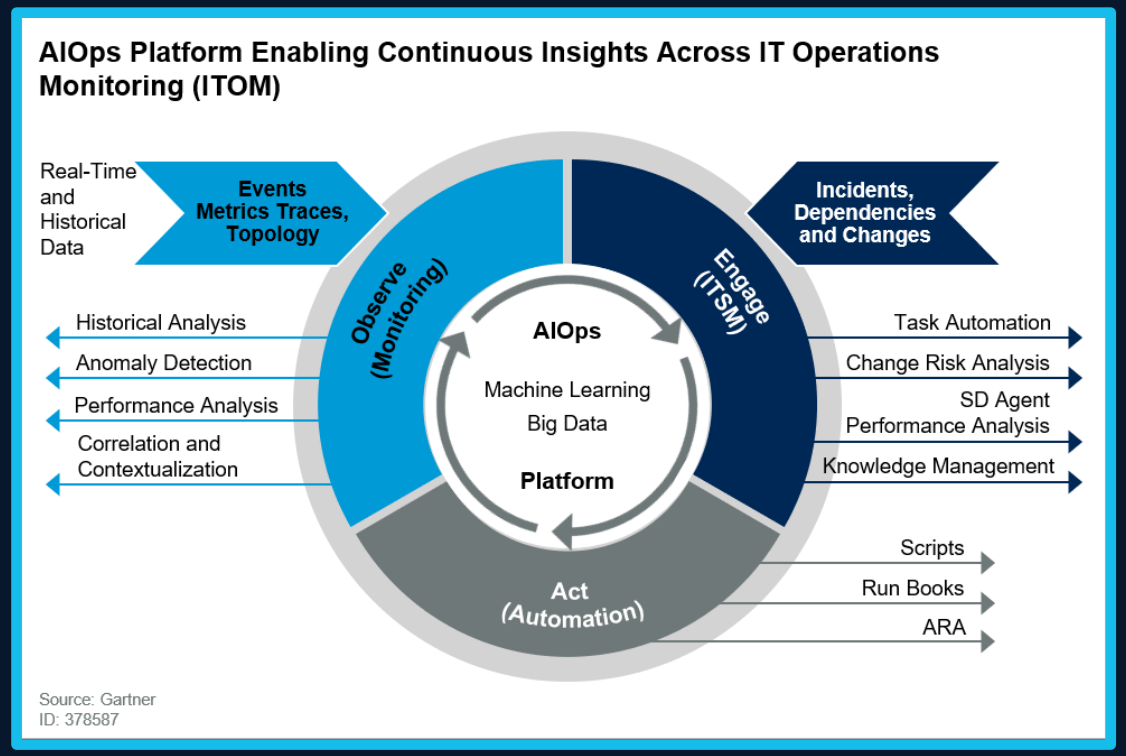
A practical approach to AIOps

What do we mean when we talk about “AIOps”?



Gartner defines AIOps as combining **big data and machine learning** to **automate IT operations** processes, including event correlation, anomaly detection and causality determination.

What do we mean when we talk about “AIOps”?



Gartner defines AIOps as combining **big data and machine learning** to **automate IT operations** processes, including event correlation, anomaly detection and causality determination.

Jason and Scott’s much simpler definition:

“AIOps is the pursuit of autonomous network operations.”

Getting practical with AIOps

"of or concerned with the actual doing or use of something rather than with theory and ideas."

Definition of "practical", Oxford English Dictionary

Lessons from our AIOps journey

Tools that you can use TODAY

Bring-your-own use case

Exploring the art of possible with the “AIOps Playground”

- Virtual network devices running on Cisco Modeling Labs
- YANG Suite for experimenting with YANG models
- gNMI-based streaming telemetry
- Telegraf, InfluxDB, Grafana (TIG) stack for collection and visualization
- Splunk and MLTK for telemetry data analysis, ML-based insights, and alerting
- AI-driven multi-agent solution for closed-loop automation

IOS XR IOS XE CISCO CML Cisco Modeling Labs

gNMI YANGSuite

telegraf influxdb Grafana

splunk> a cisco company Splunk Machine Learning Toolkit

AI Agents

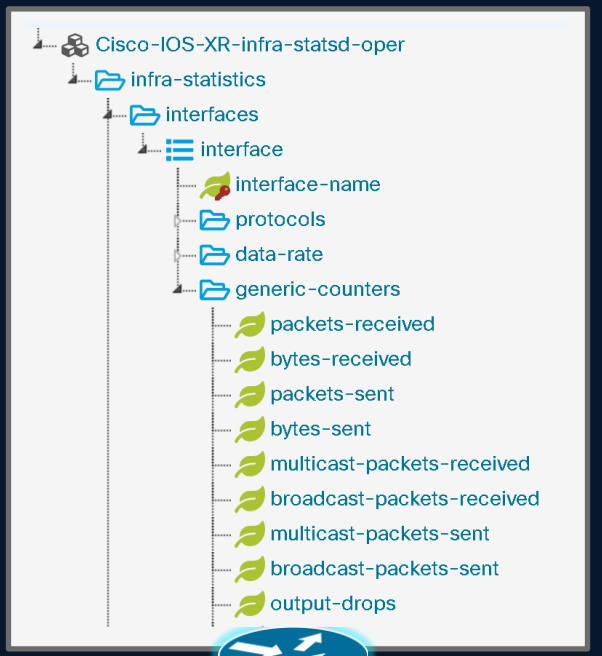
PydanticAI LangGraph

“Weeeeeee!”

Phase 1:
Model-driven telemetry

Delivering real-time visibility with model-driven telemetry

- Model-driven telemetry (MDT) is a modern, streaming-based approach to extracting operational data from network devices
- Supplements or replaces traditional polling (e.g., SNMP, CLI scraping) with push-based model using YANG data models
- Transported to collectors via gRPC or gNMI protocols, formatted using JSON_IETF or more efficient GPB-KV encoding
- **Significant advantages** over SNMP-based polling
  - More efficient delivery model: push instead of pull
  - High-frequency delivery interval: seconds instead of minutes
  - Fine-grained control: subscribe to what you need
  - Modern data modeling language: object-oriented automation



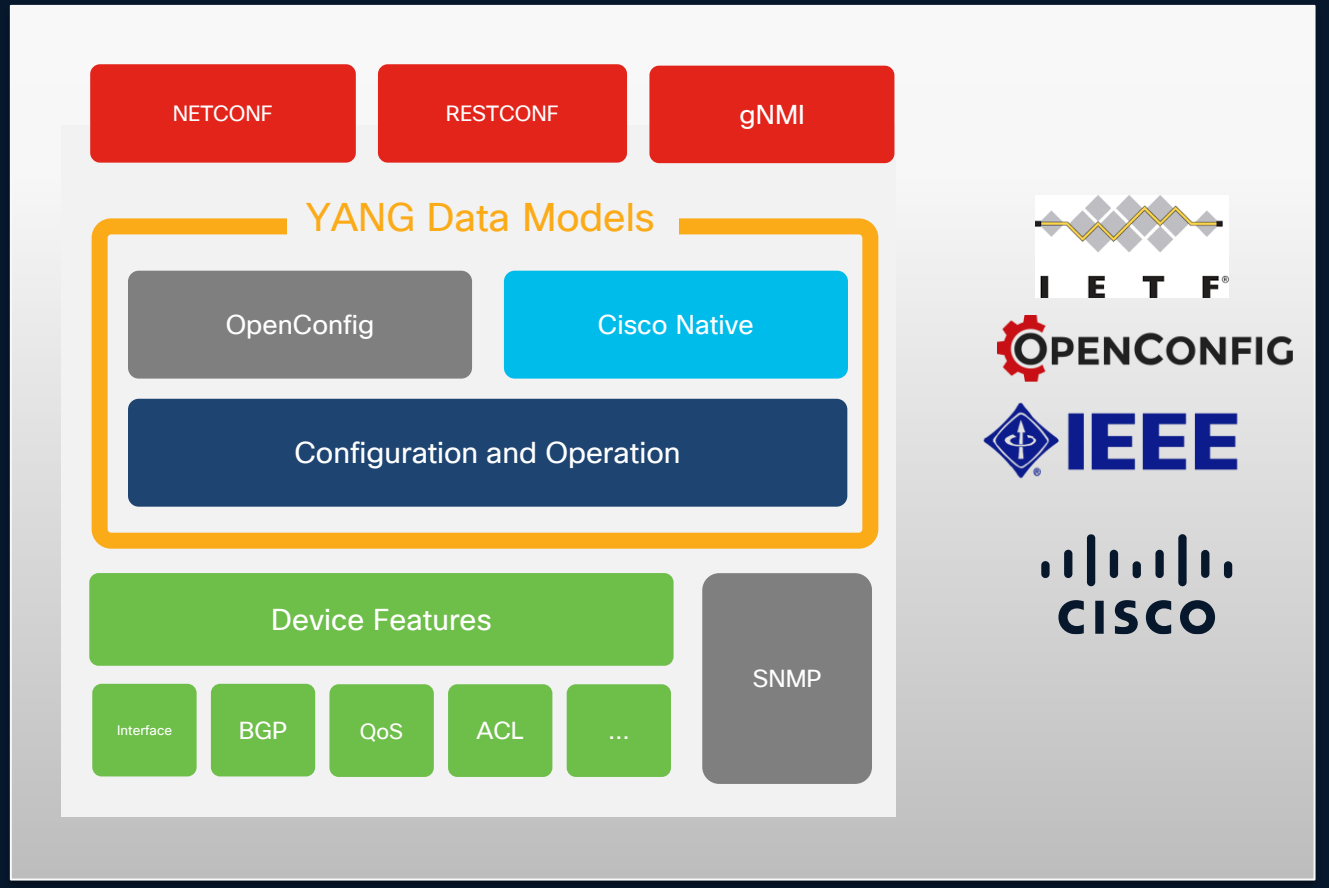

```
{
 "subscribe": {
  "subscription": [
   {
    "path": {
     "origin": "rfc7951",
     "elem": [
      {
       "name": "Cisco-IOS-XR-infra-statsd-oper:infra-statistics"
      }
     ]
    },
    "mode": "SAMPLE",
    "sampleInterval": "60000000000"
   }
  ],
  "encoding": "JSON_IETF"
 }
}
```

```
"generic-counters": {
  "packets-received": "307727",
  "bytes-received": "63184104",
  "packets-sent": "272729",
  "bytes-sent": "32413130",
  "multicast-packets-received": "0",
  "broadcast-packets-received": "3",
  "multicast-packets-sent": "0",
  "broadcast-packets-sent": "1",
  "output-drops": 0,
  "output-queue-drops": 0,
  "input-drops": 0,
  "input-queue-drops": 0,
```

Model-driven telemetry starts with YANG models

YANG data models define the operational data that is available for model-driven telemetry

- Native vs. OpenConfig
- Configuration vs. Operation
- Dial-In vs. Dial-Out

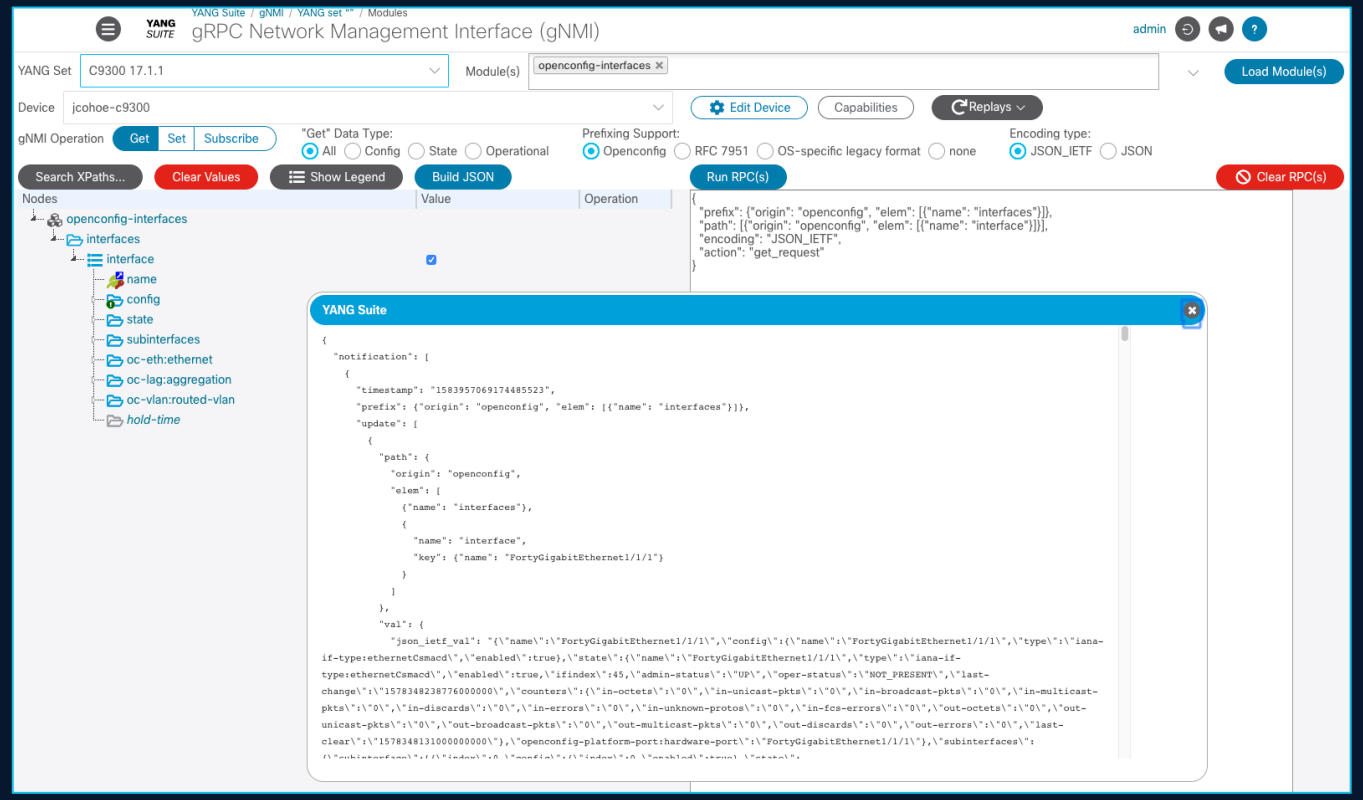


https://github.com/YangModels/yang

Finding the right YANG models with YANG Suite

Cisco YANG Suite provides a set of tools and plugins to learn, test, and adopt YANG programmable interfaces such as NETCONF, RESTCONF, and gNMI

- Search and explore YANG models
- Generate telemetry subscription RPCs
- Test MDT subscriptions against live devices

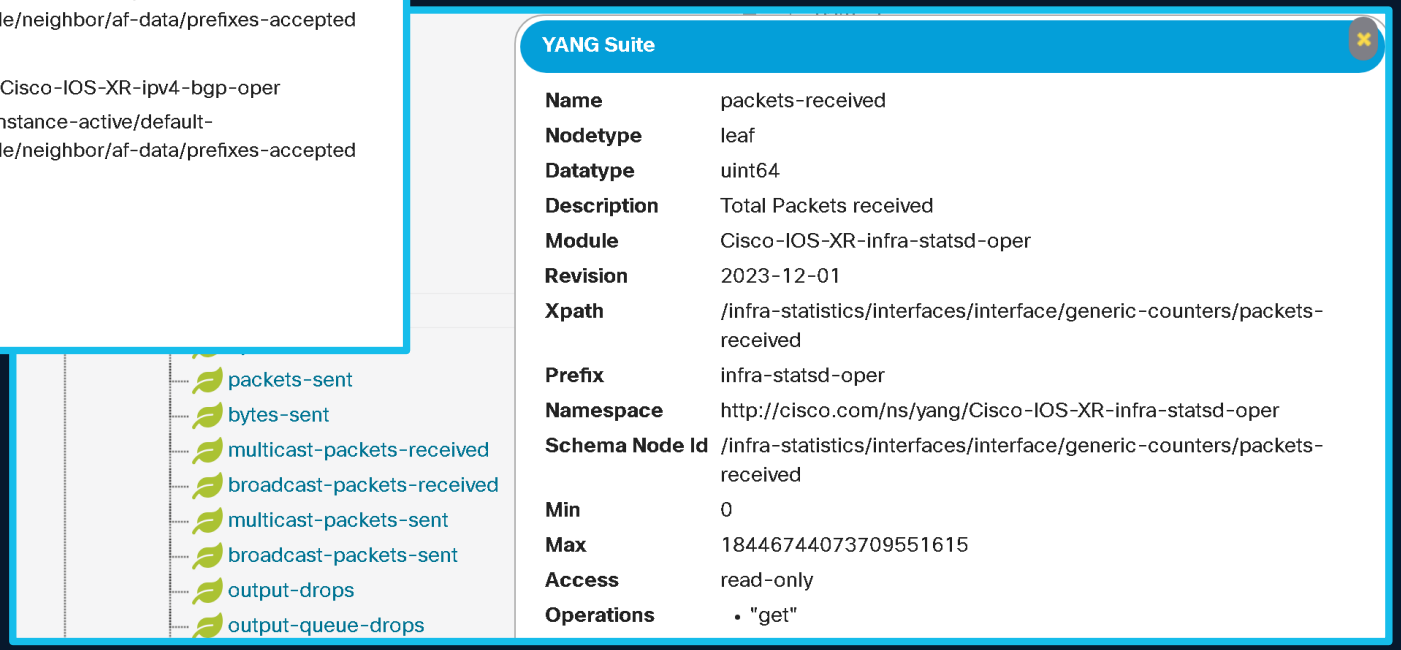


https://developer.cisco.com/yangsuite/

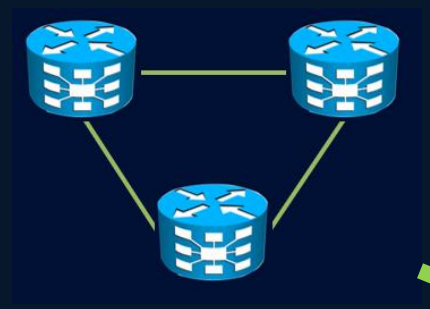
YANG paths for today’s use cases

BGP prefixes accepted

Interface packets received



How telemetry data is collected and forwarded to Splunk



Subscribe Request

Telemetry

gNMI dial-in

telegraf

HTTP Event Collector

"HEC"

splunk> a CISCO company

Splunk Machine Learning Toolkit

YANGSuite YANG Model Research

Configure desired telemetry streams

telegraf.conf

Configuring network devices and Telegraf for MDT

```
[[outputs.http]]
  url = "http://198.18.133.23:8088/services/collector"
  data_format="splunkmetric"
  splunkmetric_hec_routing=true
  [outputs.http.headers]
    Content-Type = "application/json"
    Authorization = "Splunk 1e3133fa-8b71-40d9-a9bb-0f5c154b8917"

[[inputs.gnmi.subscription]]
    name = "instanceSpecificBGPData-prefixes-accepted"
    origin = "Cisco-IOS-XR-ipv4-bgp-oper"
    path = "bgp/instances/instance/instance-active/
            default-vrf/afs/af/neighbor-af-table/neighbor/
            af-data/prefixes-accepted"
    subscription_mode = "sample"
    sample_interval = "30s"
```


```
grpc
 vrf mgmt
 port 57400
 no-tls
tpa
 vrf mgmt
  address-family ipv4
   default-route mgmt
   update-source dataports MgmtEth0/RP0/CPU0/0
```

```
gnxi
gnxi server
```

Configuring Splunk to receive telemetry data

Basic HTTP Event Collector (HEC) Setup Process

Create Custom App (Optional)

Apps > Manage Apps >

Create app

Folder Name {app_name}

Create new index

Settings > Indexes >

New Index

Index Name {index_name}
Index Data Type: Metrics

Save

Create HEC

Settings > Data Inputs >
HTTP Event Collector >
Add new >
Name {hec_name}

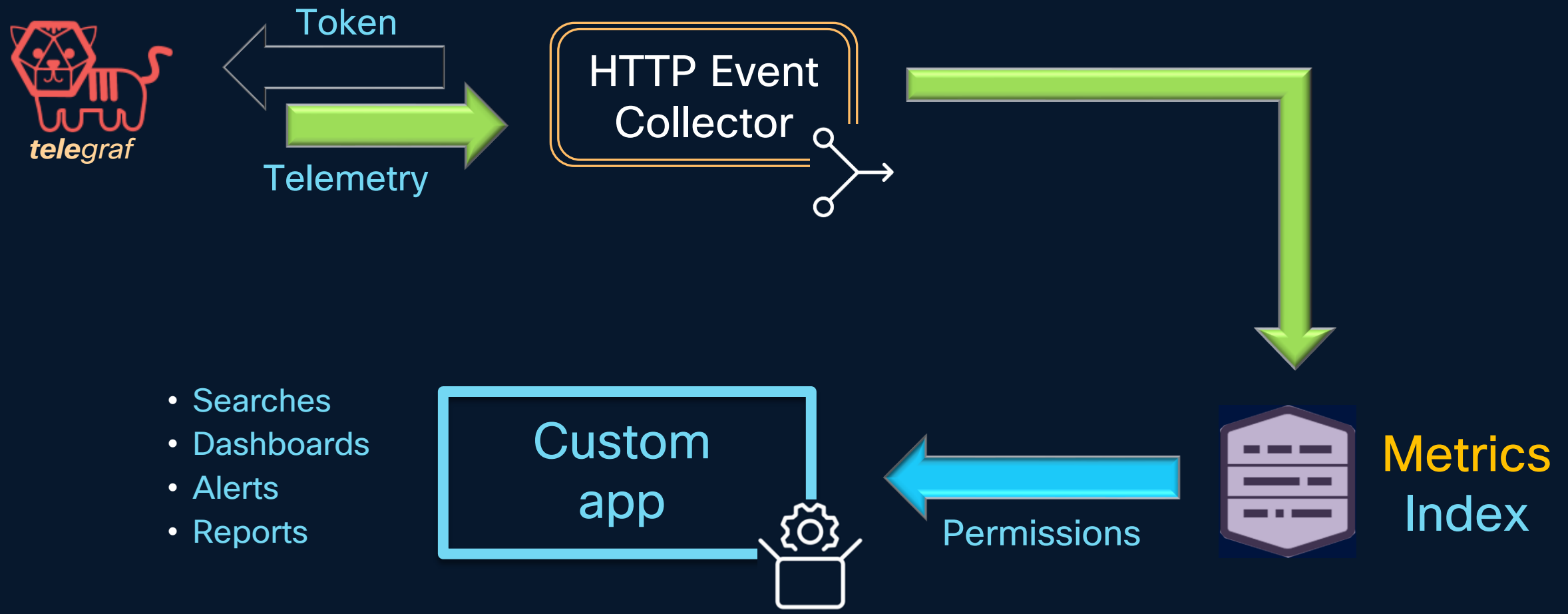
Next >

1. Source type >
Metrics > Telegraf
2. Select Allowed Indexes
{choose target index}

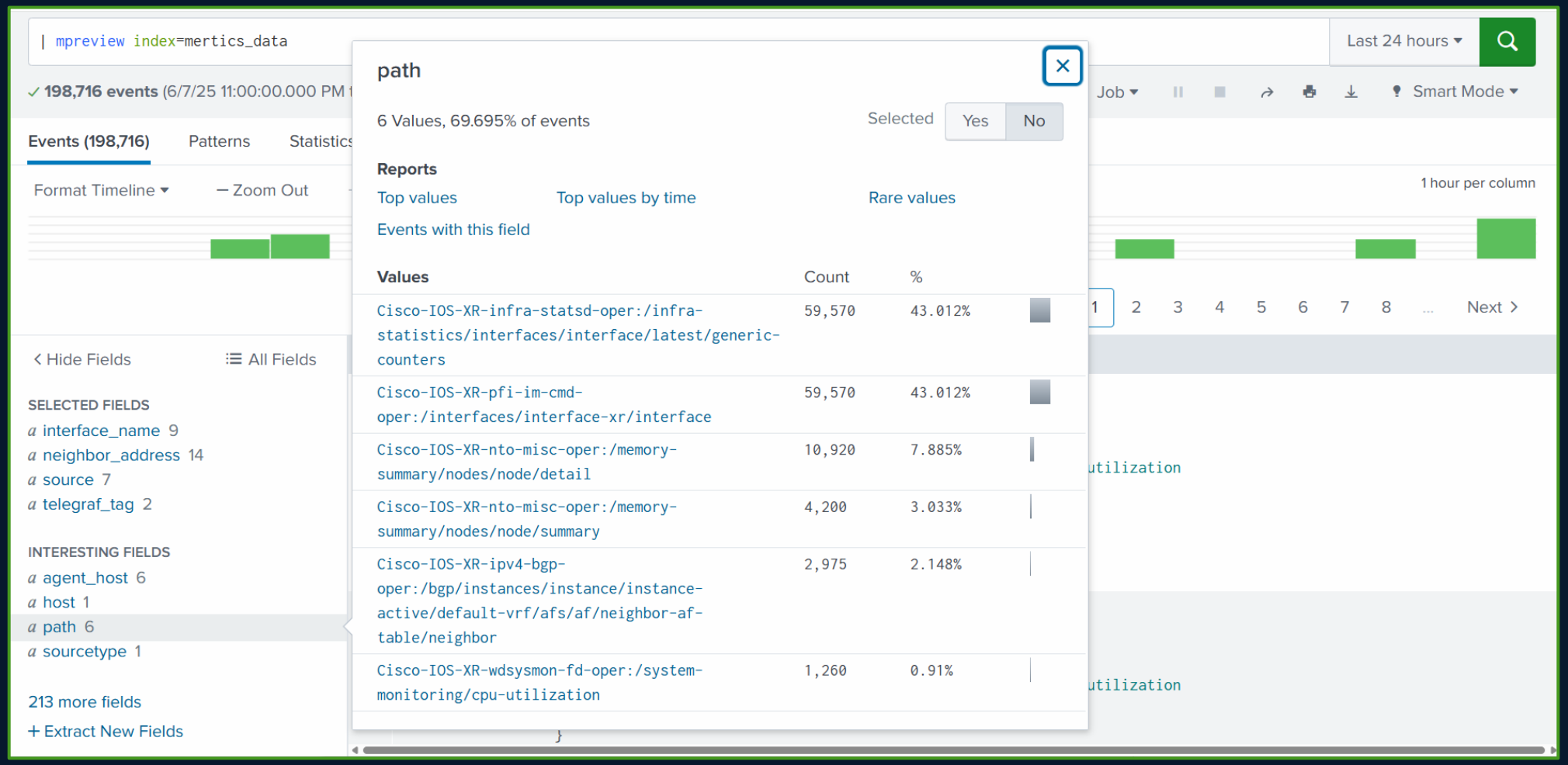
Review > → Submit >

...now copy HEC token over to telegraf.conf file

Bringing it all together



Verifying telemetry data is received in Splunk



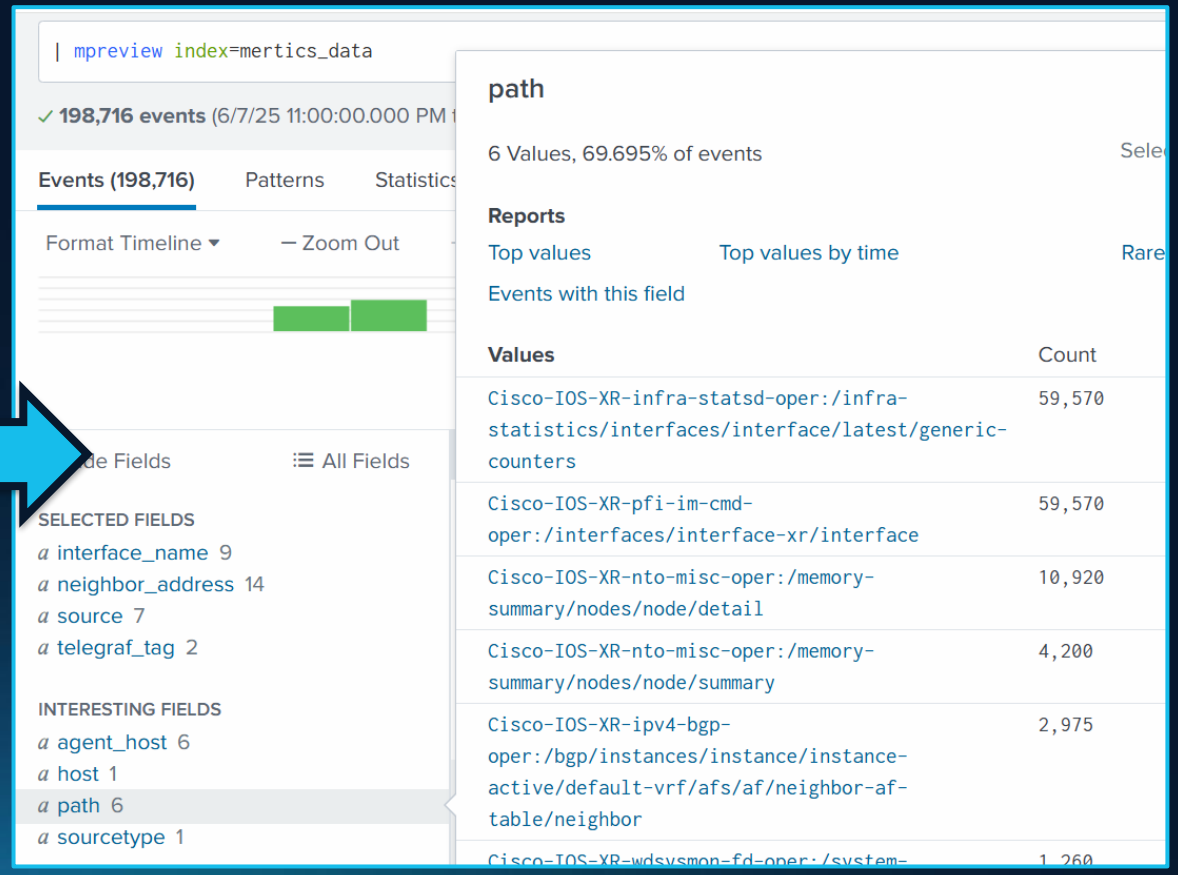
SPL Query: `| mpreview index={index_name}`

Watch the end-to-end setup process...

```
[[outputs.http]]
  url = "http://198.18.133.23:8088/services/collector"
  data_format="splunkmetric"
  splunkmetric_hec_routing=true
  [outputs.http.headers]
    Content-Type = "application/json"
    Authorization = "Splunk 1e3133fa-8b71-40d9-a9bb-0f5c154b8917"

[[inputs.gnmi.subscription]]
    name = "instanceSpecificBGPData-prefixes-accepted"
    origin = "Cisco-IOS-XR-ipv4-bgp-oper"
    path = "bgp/instances/instance/instance-active/
            default-vrf/afs/af/neighbor-af-table/neighbor/
            af-data/prefixes-accepted"
    subscription_mode = "sample"
    sample_interval = "30s"
```

telegraf

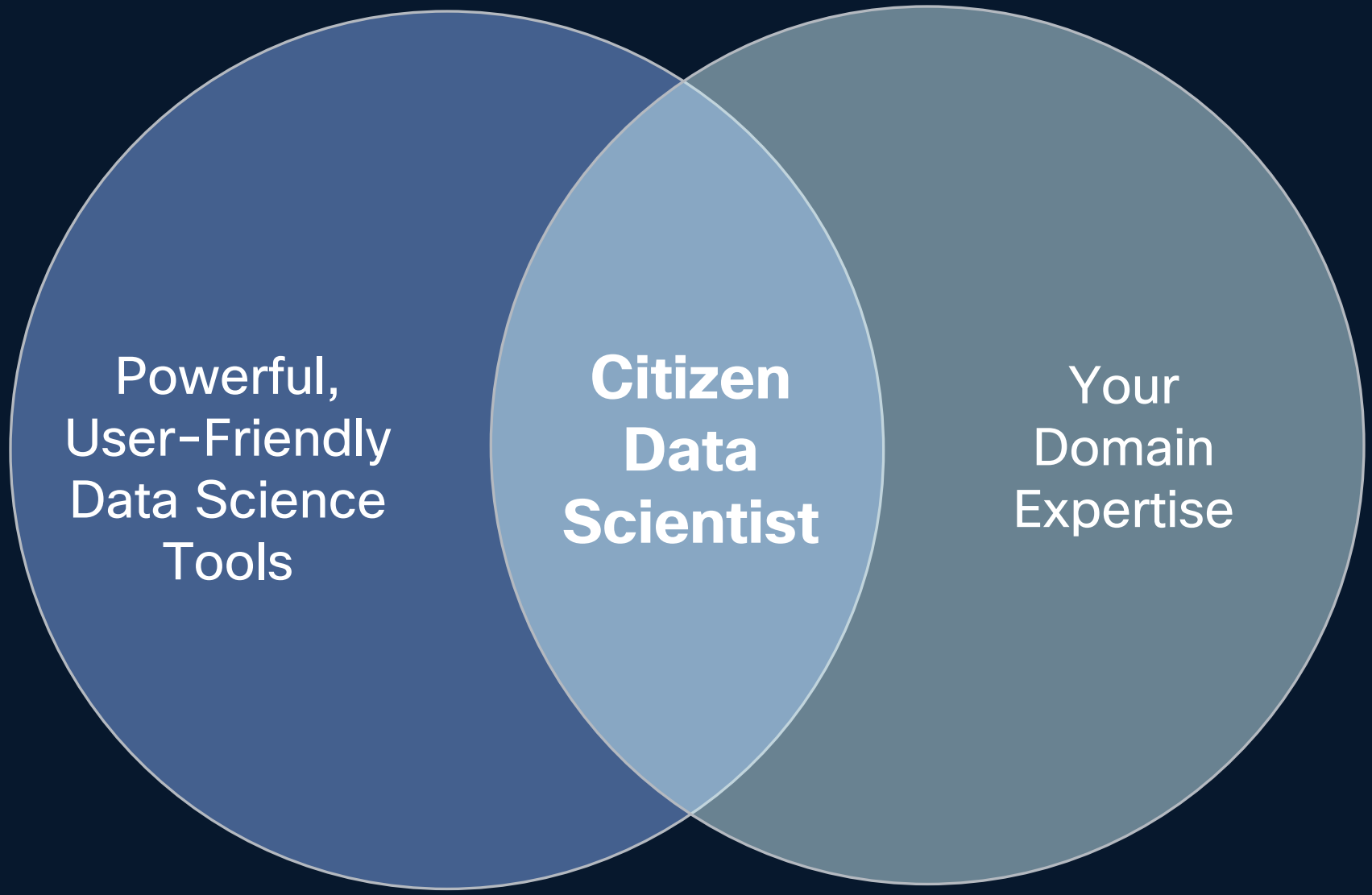


cs.co/practical-aiops

Phase 2:
ML-driven insights

Swimming in data...Starving for insights

Leveraging your domain expertise with machine learning



We want YOU to become a Citizen Data Scientist!

A Citizen Data Scientist...

- Understands the data AND the problem
- Leverages powerful analytics to extract actionable insights from telemetry

Maximize the value of your domain expertise in a changing world

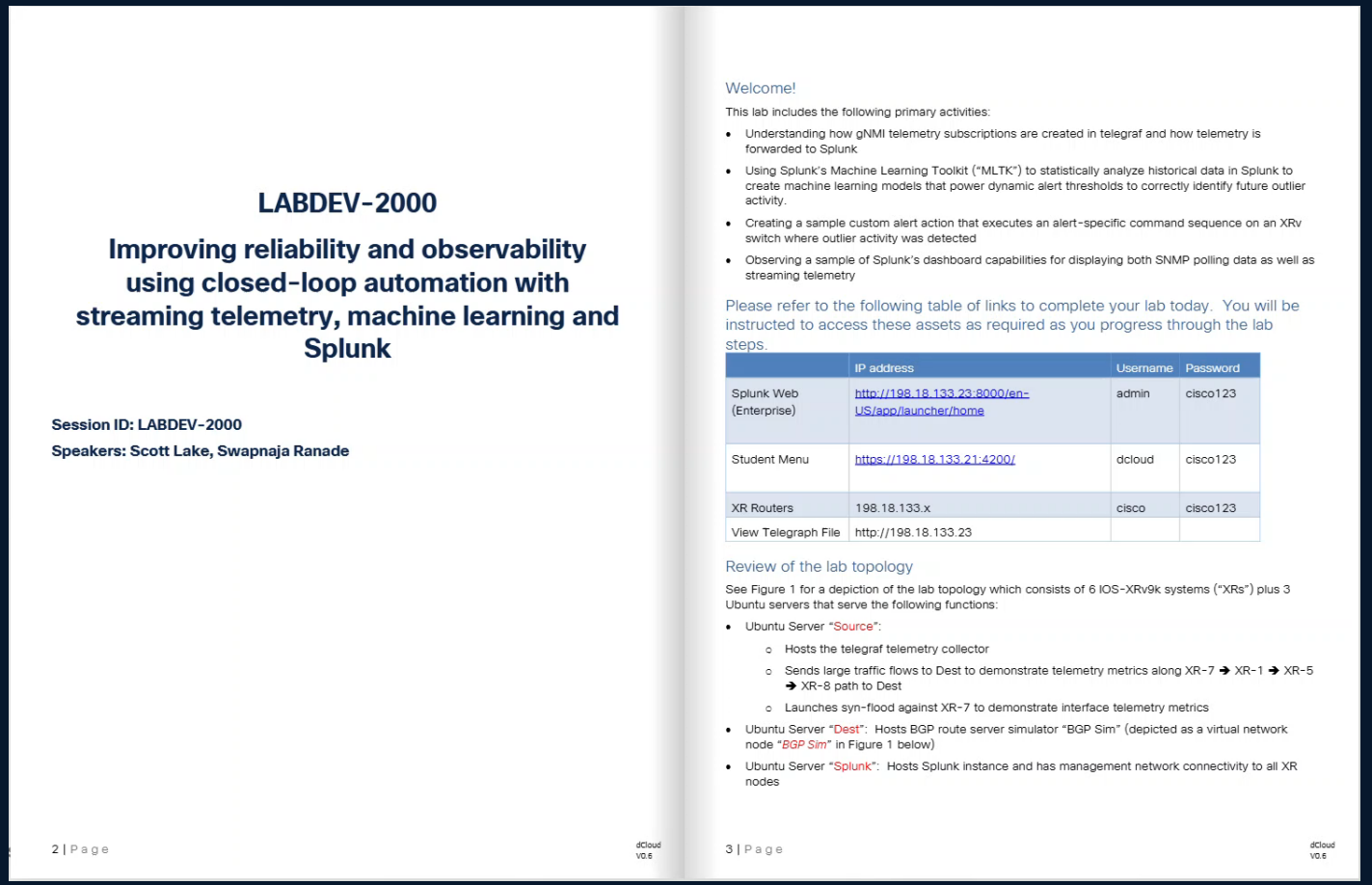
"It's NOT about becoming a *machine-learning engineer*, it's about becoming a better *network engineer* "
– Scott Lake, CDS

Zero-to-"Citizen Data Scientist" *Let's go!*

- ✓ Demystifying the Machine Learning Toolkit: "What is it really?"
- ✓ How to Model Your Streaming Telemetry – Step by Step
  - ➔ Time Processing
  - ➔ Feature Engineering
  - ➔ Algorithm Selection
  - ➔ Threshold Tuning
- ✓ Deploying Your Models In Alerts with Automated Actions

Before we get started...

- Pre-built lab environment available now for learning and POCs
- Includes comprehensive training guide ➔ ➔ ➔
- All components are installed, configured and telemetry is flowing
- Immediately start building and testing data models
- Includes pre-configured network event scenarios



LABDEV-2000

Improving reliability and observability using closed-loop automation with streaming telemetry, machine learning and Splunk

Session ID: LABDEV-2000

Speakers: Scott Lake, Swapnaja Ranade

2 | Page

dCloud v0.6

Welcome!

This lab includes the following primary activities:

- Understanding how gNMI telemetry subscriptions are created in telegraf and how telemetry is forwarded to Splunk
- Using Splunk's Machine Learning Toolkit ("MLTK") to statistically analyze historical data in Splunk to create machine learning models that power dynamic alert thresholds to correctly identify future outlier activity.
- Creating a sample custom alert action that executes an alert-specific command sequence on an XRv switch where outlier activity was detected
- Observing a sample of Splunk's dashboard capabilities for displaying both SNMP polling data as well as streaming telemetry

Please refer to the following table of links to complete your lab today. You will be instructed to access these assets as required as you progress through the lab steps.

	IP address	Username	Password
Splunk Web (Enterprise)	http://198.18.133.23:8000/en-US/app/launcher/home	admin	cisco123
Student Menu	https://198.18.133.21:4200/	dcloud	cisco123
XR Routers	198.18.133.x	cisco	cisco123
View Telegraph File	http://198.18.133.23		

Review of the lab topology

See Figure 1 for a depiction of the lab topology which consists of 6 IOS-XRv9k systems ("XRs") plus 3 Ubuntu servers that serve the following functions:

- Ubuntu Server "Source":
  - Hosts the telegraf telemetry collector
  - Sends large traffic flows to Dest to demonstrate telemetry metrics along XR-7 ➔ XR-1 ➔ XR-5 ➔ XR-8 path to Dest
  - Launches syn-flood against XR-7 to demonstrate interface telemetry metrics
- Ubuntu Server "Dest": Hosts BGP route server simulator "BGP Sim" (depicted as a virtual network node "*BGP Sim*" in Figure 1 below)
- Ubuntu Server "Splunk": Hosts Splunk instance and has management network connectivity to all XR nodes

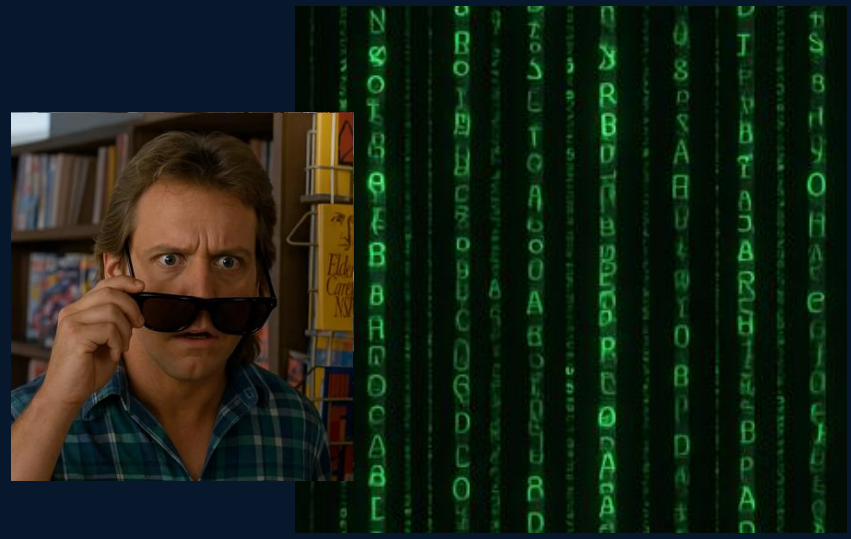
3 | Page

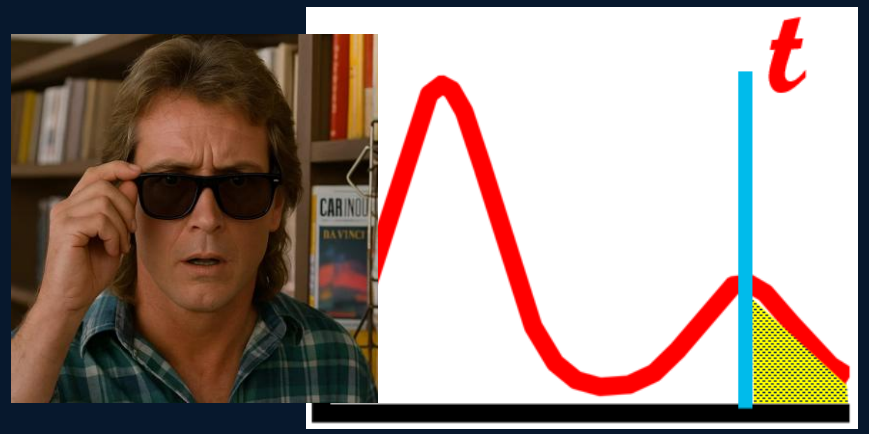
dCloud v0.6

cs.co/practical-aiops

What is machine learning?

- Finds patterns and relationships in past data that humans might miss or find too complex to identify manually
- Makes predictions or decisions on new, unseen data based on what it learned from its analysis
- It allows systems to autonomously learn and adapt their decision-making from that data, rather than relying on rigid, pre-set logic.





Demo:

Importing apps for machine learning
Viewing incoming telemetry

Under the hood: Splunk's Machine Learning Toolkit (MLTK)

Data science and machine learning: What does it take?

- Math and statistics
- Programming
- Data Science
- Data Modeling

$$\min_{a,b} \sum_{i=1}^{n} (y_i - (ax_i + b))^2$$

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

- Calculus
- Linear Algebra
- Probability/Statistics
- Analytic Geometry

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

Mathematics

Python

Programming

NumPy

matplotlib

pandas

TensorFlow

Keras

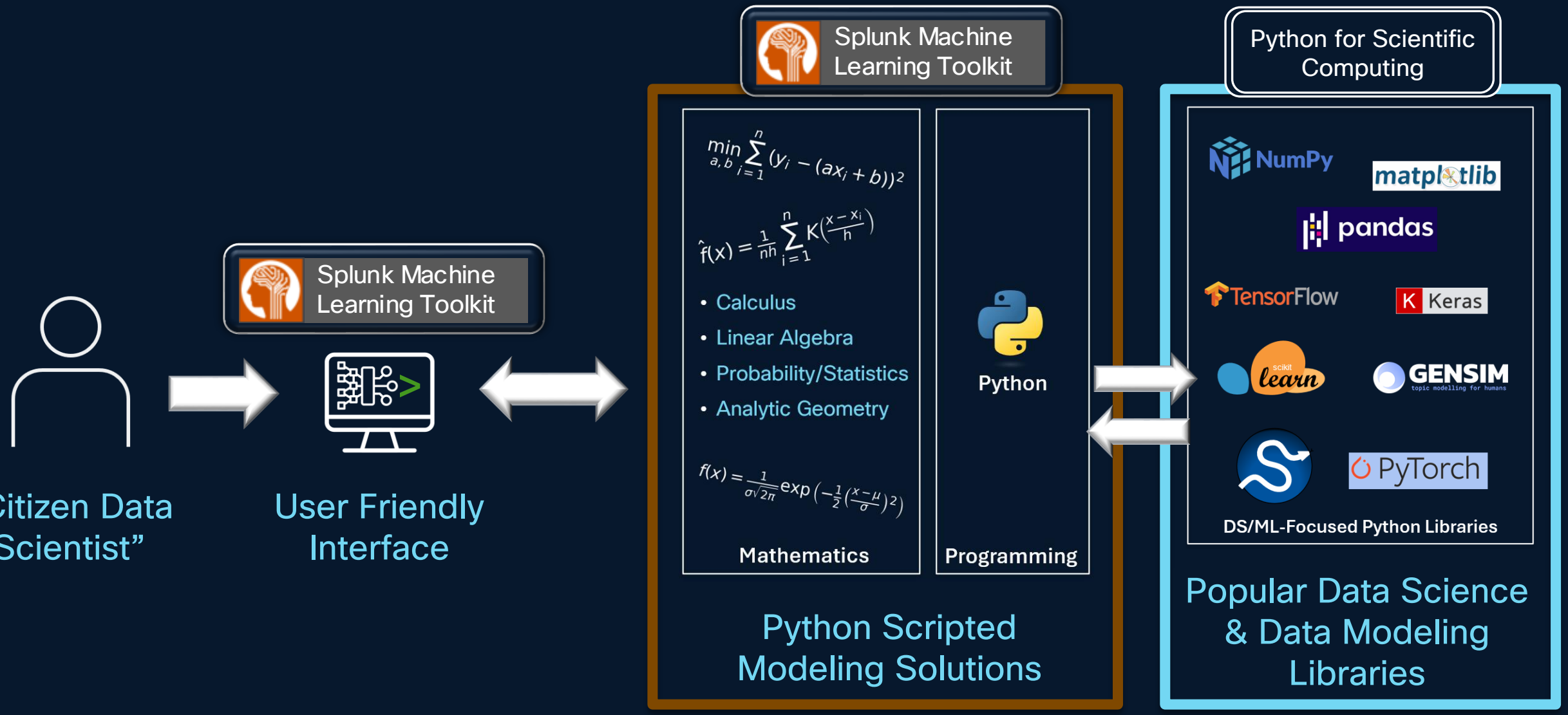
scikit learn

GENSIM
topic modelling for humans

PyTorch

DS/ML-Focused Python Libraries

Non-data science professionals achieving ML results



Splunk MLTK immediate take-aways…

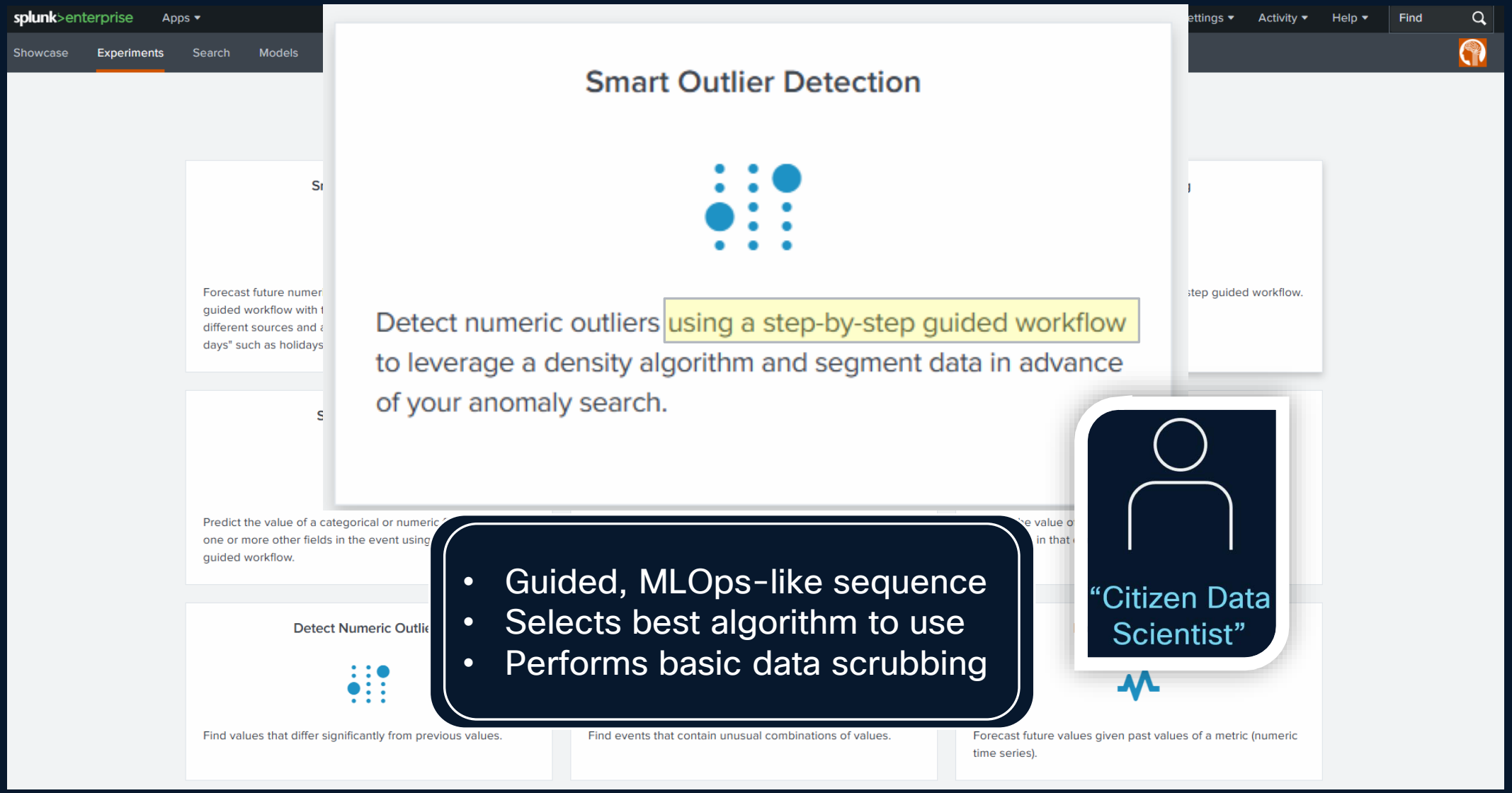
- **Not a black box:** All models originate from popular python libraries used by data science professionals, completely transparent

```
/opt/splunk/etc/apps/Splunk_ML_Toolkit/bin/algos/*
/opt/splunk/etc/apps/Splunk_ML_Toolkit/bin/algos_support/*
```

- **Adaptable:** Easy for network engineers to get started, yet powerful enough for data science pros to deeply analyze network data.
- **Great learning tool:** Learn and customize complex ML-related SPL queries, empowering you for more advanced use cases as you grow!
- **No additional cost:** Available on Splunkbase app store as a free download

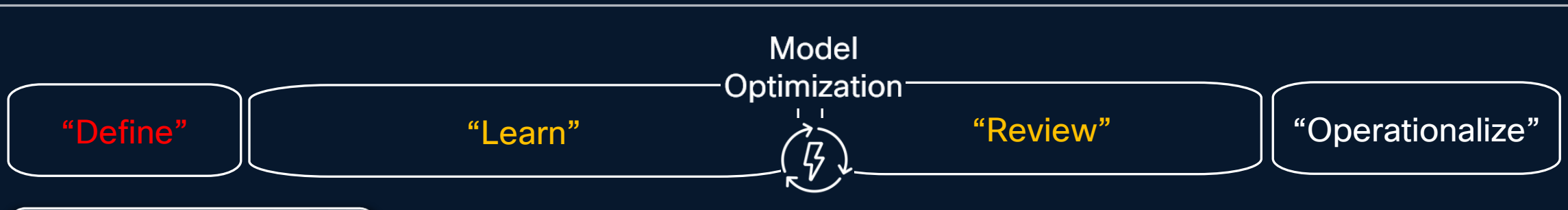
Enabling the Citizen Data Scientist: Smart Assistants

MLTK Preview – The Smart Outlier Detection Assistant



MLTK Guided Workflow Sequence

"Typical Machine Learning Operations ("MLOps") Sequence"



"Guided Configuration Sequence (Smart Assistants Only)

Business problem:

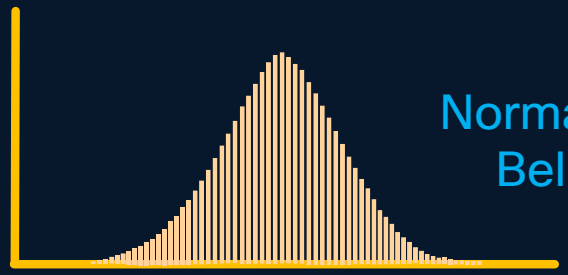
"Can we build better alerts to detect deviations from normal traffic flows in our network?"

Demo: MLTK first look

- Identify data to study
- Add time attributes to model
- Add more context to model

Algorithm selection

Real World Data Shapes

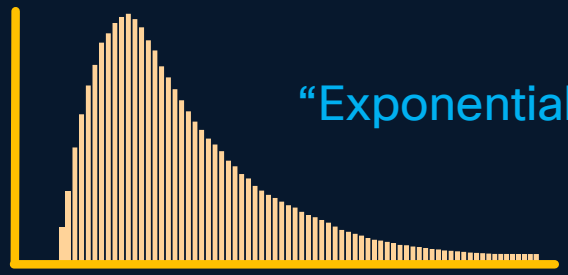


Normal / Bell

"Central Tendency, Symmetrical"

Data Behavior Implication
(e.g., "packets_received/second")

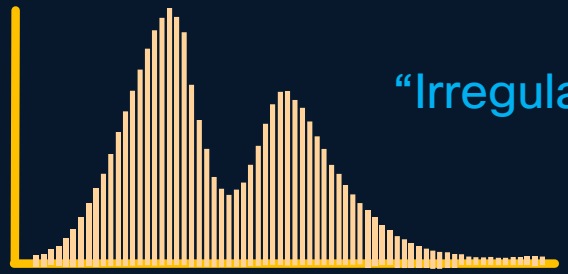
"Steady stream of traffic"



"Exponential"

"Skewed"

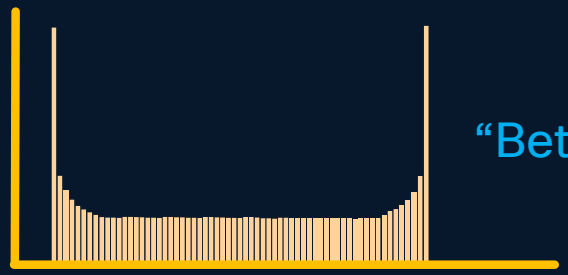
"Normally quiet, subject to infrequent, large flows"



"Irregular"

"Shape not assumed or known"

"Two distinct traffic rates observed"



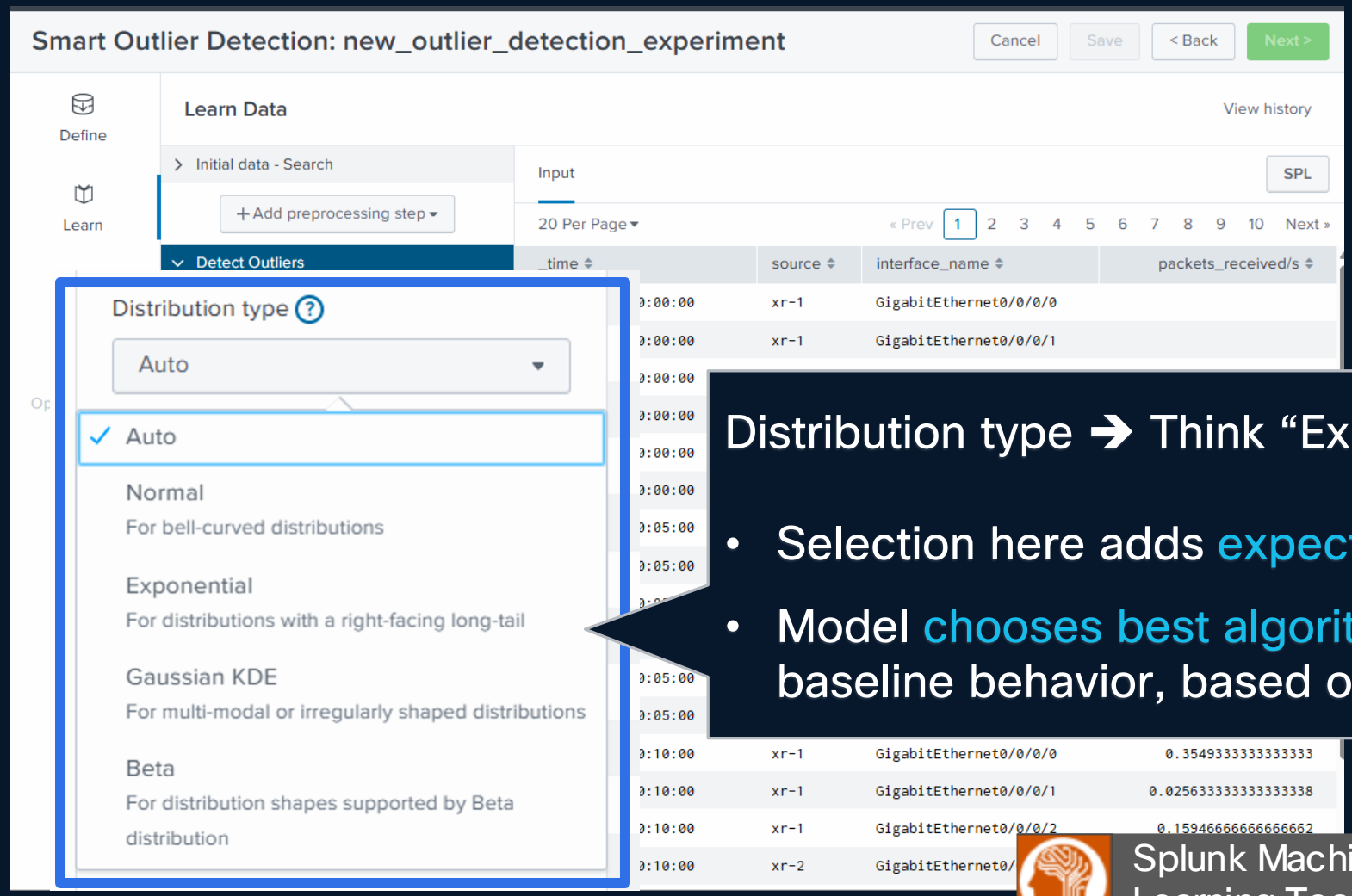
"Beta"

"Bounded, various shapes"

"Standby interface (e.g. HSRP)"

MLTK Preview – Distribution Type (“Expected Data Shape”)

Choose the assumed data shape to drive optimal algorithm selection for generating PDFs



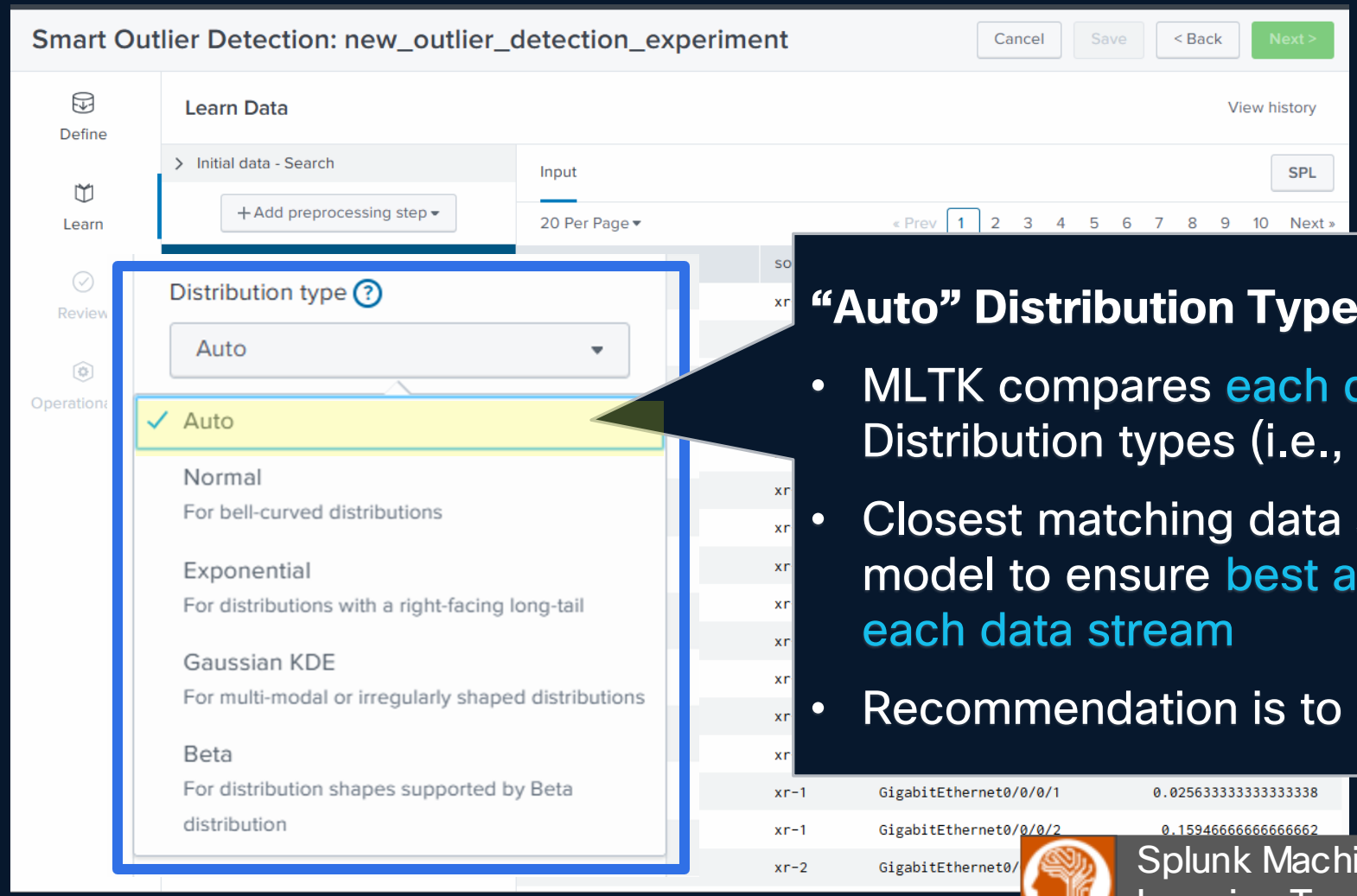
Distribution type ➔ Think “Expected shape of data”

- Selection here adds expected shape to model
- Model chooses best algorithm to understand the baseline behavior, based on the expected shape

Splunk Machine Learning Toolkit

MLTK Preview – Distribution Type (“Expected Data Shape”)

Choose the assumed data shape to drive optimal algorithm selection for generating PDFs



“Auto” Distribution Type

- MLTK compares each data stream to all 4 Distribution types (i.e., expected data shapes)
- Closest matching data shape is included in the model to ensure best algorithm selection for each data stream
- Recommendation is to always choose “Auto” ✓

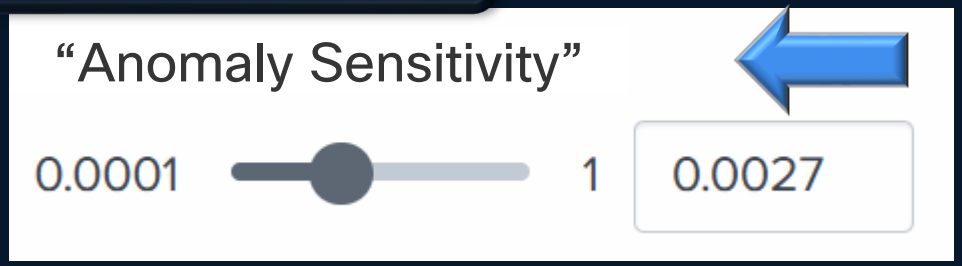
Splunk Machine Learning Toolkit

Threshold Selection

What's an anomaly?
What's not?

MLTK Preview: Threshold Tuning Sequence

Splunk Machine Learning Toolkit



Think “anomaly sensitivity”

Setting to .27% means we only want to flag anomalies for values that have been seen less than .27% of the time historically.

- Identifies more unusual events (anomalies)
- Focuses on minimizing *false negative* errors (model is more sensitive to anomaly activity)

- Identifies fewer unusual events (anomalies)
- Focuses on minimizing *false positive* errors (model is less sensitive to anomaly activity)

Smart Outlier Detection: sample_smart_outlier_detection_experiment Draft

Detect outliers in ***packets_received/s***, split by ***2 fields*** , using ***Auto*** distribution with a threshold of ***0.01***

Define

Learn

Review

Operationalize

Learn Data

Field to analyze

packets_received/s

Split by fields

interface_name, source

Distribution type

Auto

"anomaly sensitivity"

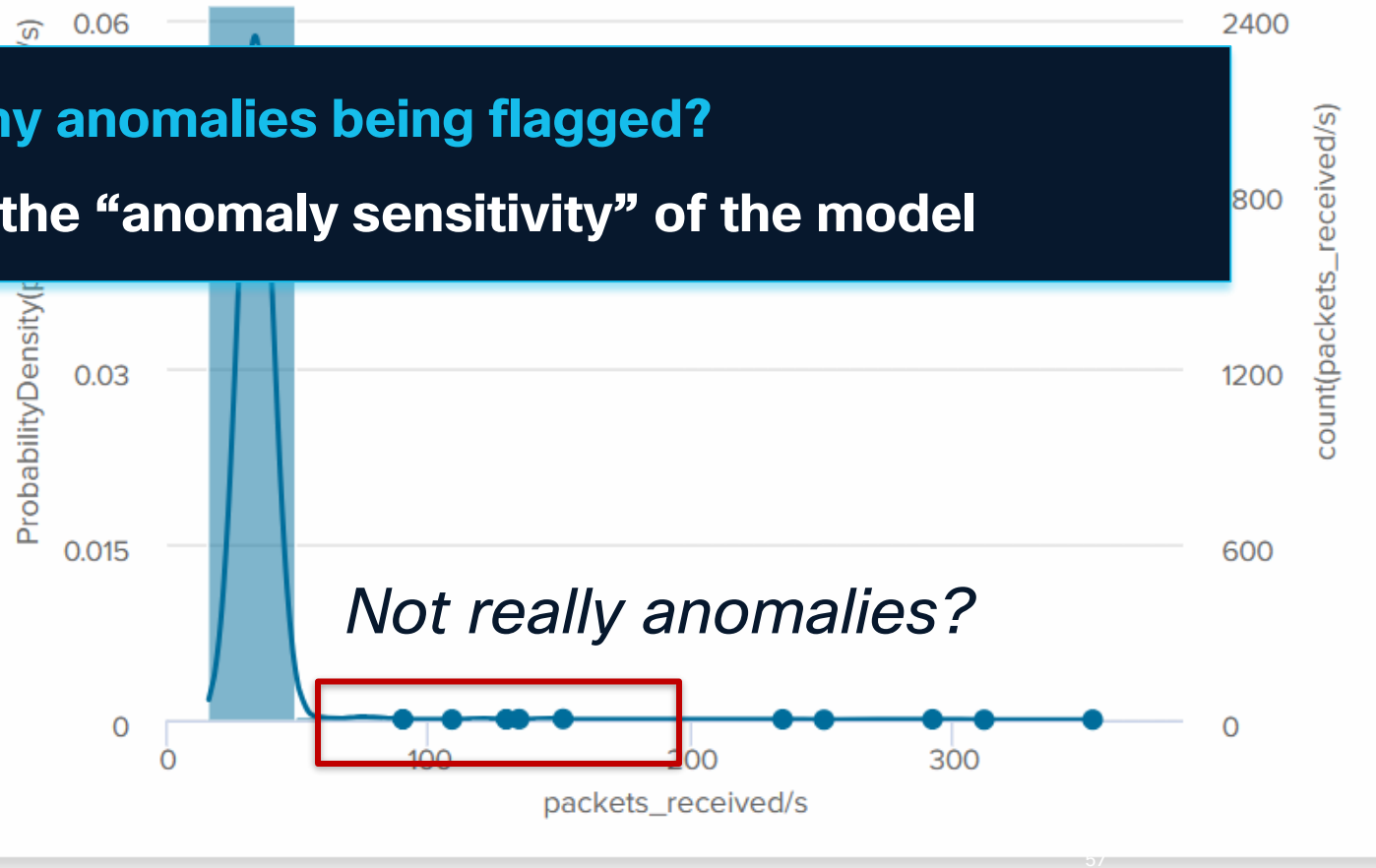
0.0001 — 1 — 0.01

Reduce Sensitivity

Notes

Too many anomalies being flagged?

Reduce the "anomaly sensitivity" of the model



Smart Outlier Detection: sample_smart_outlier_detection_experiment Draft

Detect outliers in ***packets_received/s***, split by ***2 fields*** , using ***Auto*** distribution with a threshold of ***0.006***

Define

Learn

Review

Operationalize

Learn Data

Detect Outliers

Field to analyze

packets_received/s

Split by fields

interface_name, source (2)

Distribution type

Auto

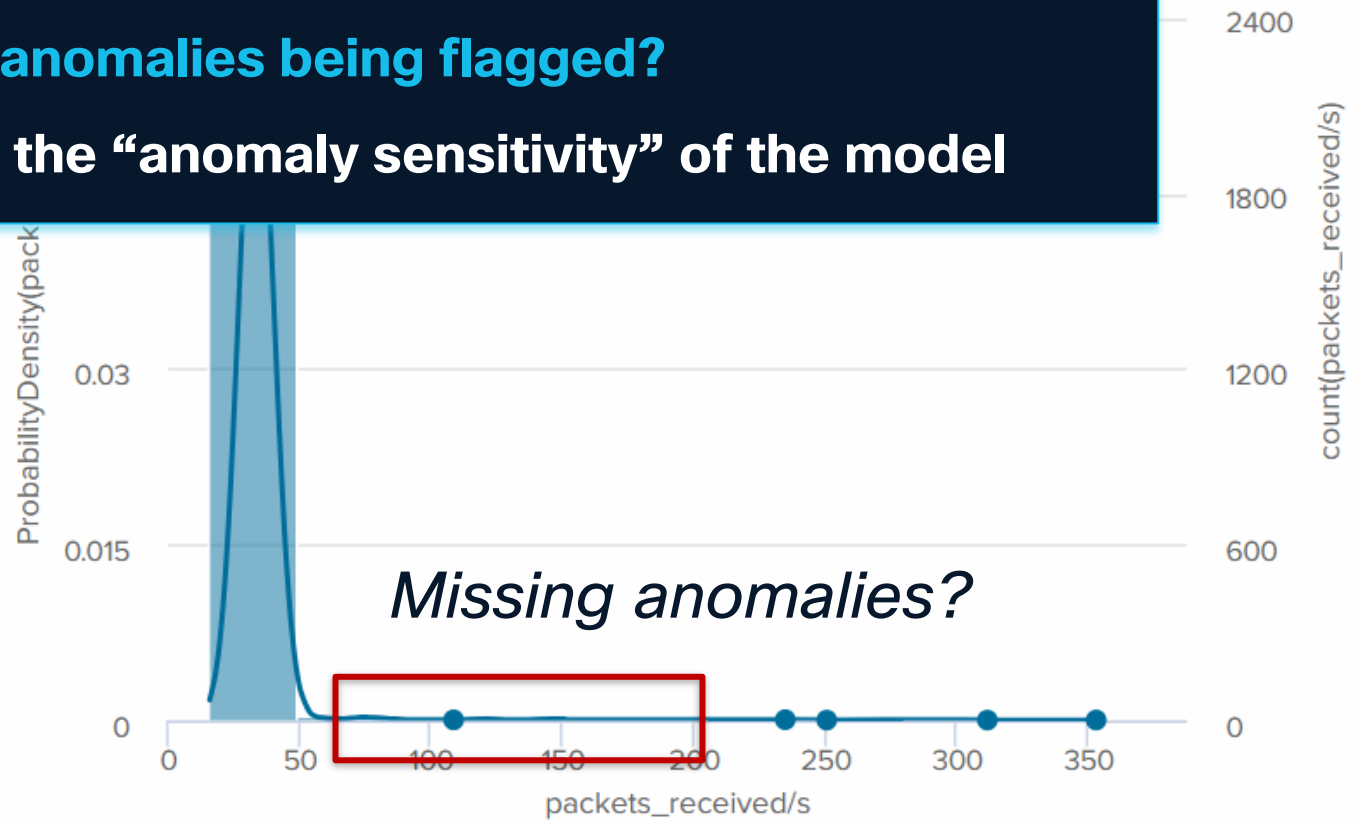
“anomaly sensitivity”

0.0001 1 0.006

Increase Sensitivity

Too few anomalies being flagged?

Increase the “anomaly sensitivity” of the model



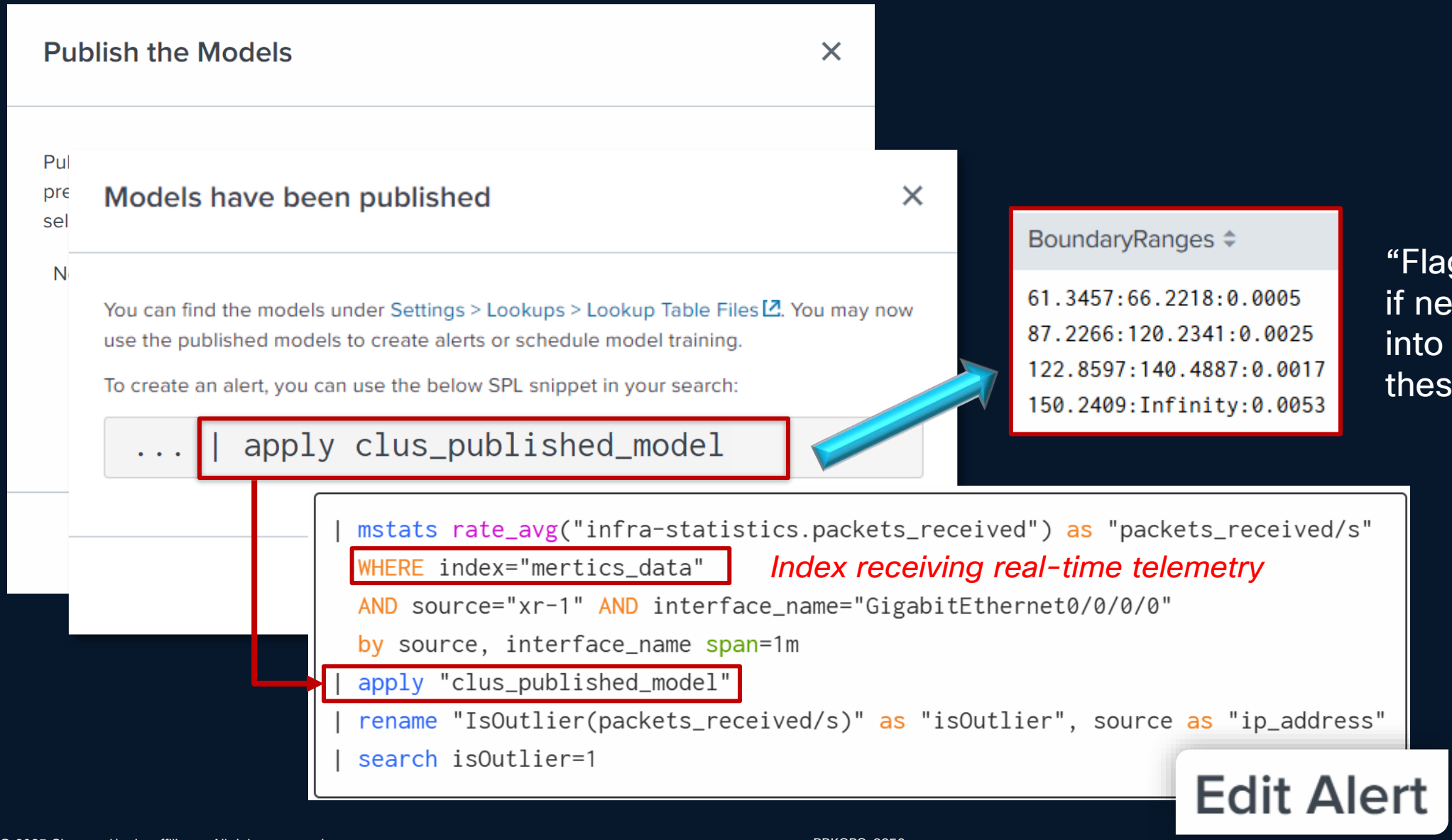
Demo:

- Algorithm selection
- Threshold tuning
- Reviewing model output

Deploying Your Models

Creating Machine Learning-Enabled Alerts

MLTK Preview – Formulate SPL Query For Alert



"Flag as anomaly if new value falls into any one of these ranges"

Demo: Enabling Alerts

Publishing a tuned data model
Creating SPL to deploy in alert
Specifying trigger action for an alert

Phase 3:
AI-driven decisions

The elusiveness of closed-loop automation

We all want closed-loop automation...

- Faster fault detection and diagnosis
- Automated remediation
- Dynamic resource allocation
- More reliability, less downtime

...but we probably don't have it.

- Fragmented data sources
- Lack of real-time data for decisions
- Limitations of rule-based automation
- Multi-vendor complexity

AI agents pave the way for practical closed-loop automation

- Summarize and normalize data across multiple data sources
- Dynamic reasoning and decision making with human-in-the-loop fallback
- "Tools" architecture for interfacing with a distributed, multi-vendor ecosystem

Prototyping an AI-driven multi-agent troubleshooting system for closed-loop automation

Introducing Project “ADAPT”

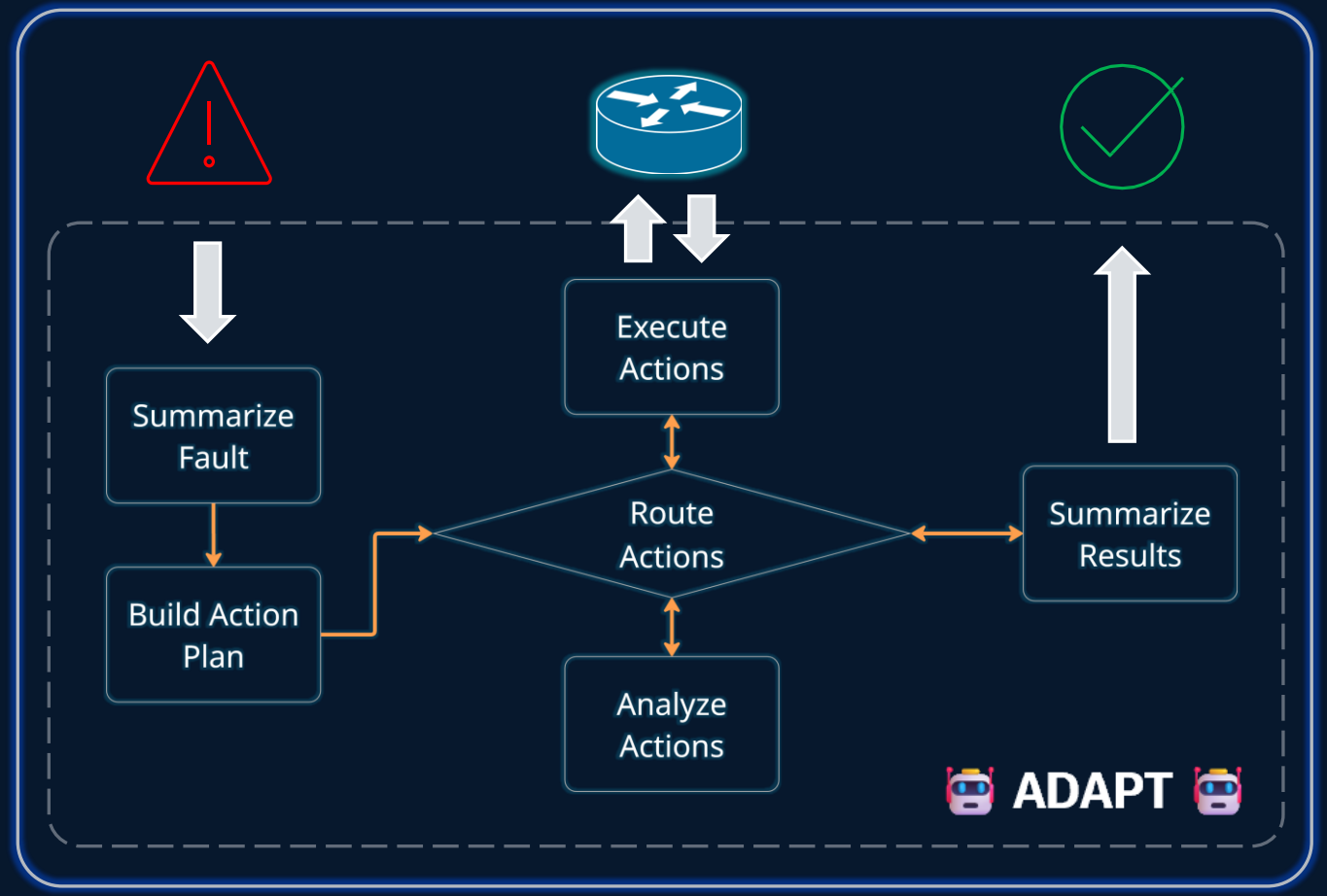
AI-**D**riven **A**ction **P**lanner and **T**roubleshooter

An autonomous troubleshooting system that adjusts its actions based on real-time findings.

Project Goals:

- Auto-diagnose and resolve network faults using a multi-agent system
- Demonstrate use of AI agent frameworks
- Explore the potential of multi-agent systems for complex adaptive workflows
- Experiment with your own use cases via our open-source repo

Coming soon: cs.co/practical-aiops

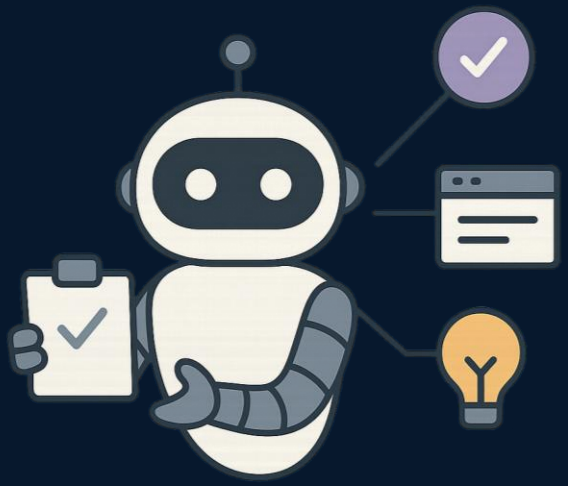


(Re)introduction to AI agents

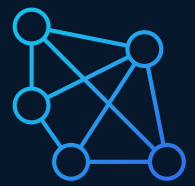
An AI agent is an intelligent system that makes decisions and takes actions in pursuit of a goal using a language model as its reasoning engine

Key characteristics of an AI agent:

- **Autonomous:** It can make decisions without human intervention.
- **Tool-using:** It can call functions, access APIs, or query databases as needed.
- **Goal-driven:** It operates toward achieving a specific objective, like answering a question, retrieving information, or solving a problem.



Design considerations for a multi-agent system



What is my system's purpose and scope?

Problem domain, desired outcomes, boundaries, success criteria

Diagnose/remediate network fault alerts by interfacing with network devices, analyzing diagnostic data, executing repair actions, and reporting results

What are the roles and responsibilities of agents?

Task modularity, system/user prompts, tools needed

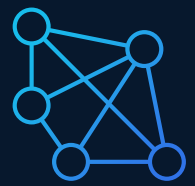
1. Summarize network fault details
2. Build action plan
3. Execute next action with tools
4. Analyze action result
5. Decide what to do next
6. Summarize results

How will my workflow be structured?

Step logic, decision points, external interactions

A step-based action plan with dynamic decisions based upon analysis of results; can continue to next step, escalate, resolve, or modify action plan.

Design considerations for a multi-agent system



What state data will flow through this system? Context, I/O, dependencies, schemas, memory, RAG	Network alert data, context-specific instructions, golden rules, settings, action plan, device facts, diagnostic data, action analysis, test data
What guardrails will be in place to restrict unintended actions? Reflection, human-in-the-loop, control limits, error-handling	“Human-in-the-loop” for potentially impactful changes, LLM-driven post-check validation actions for any changes, action count limits, error handling
How will the system be evaluated for continuous improvement? Metrics, feedback loops, testing, LLM-based evaluation	On-demand LLM-based generation and execution of test scenarios with synthetic data, LLM-based evaluation of test results

The toolbox for our multi-agent system

Agent Framework

PydanticAI

- Simple, clean, model-agnostic agentic framework architected for production use cases
- Benefits: Schema validation, dependency management, and structured outputs

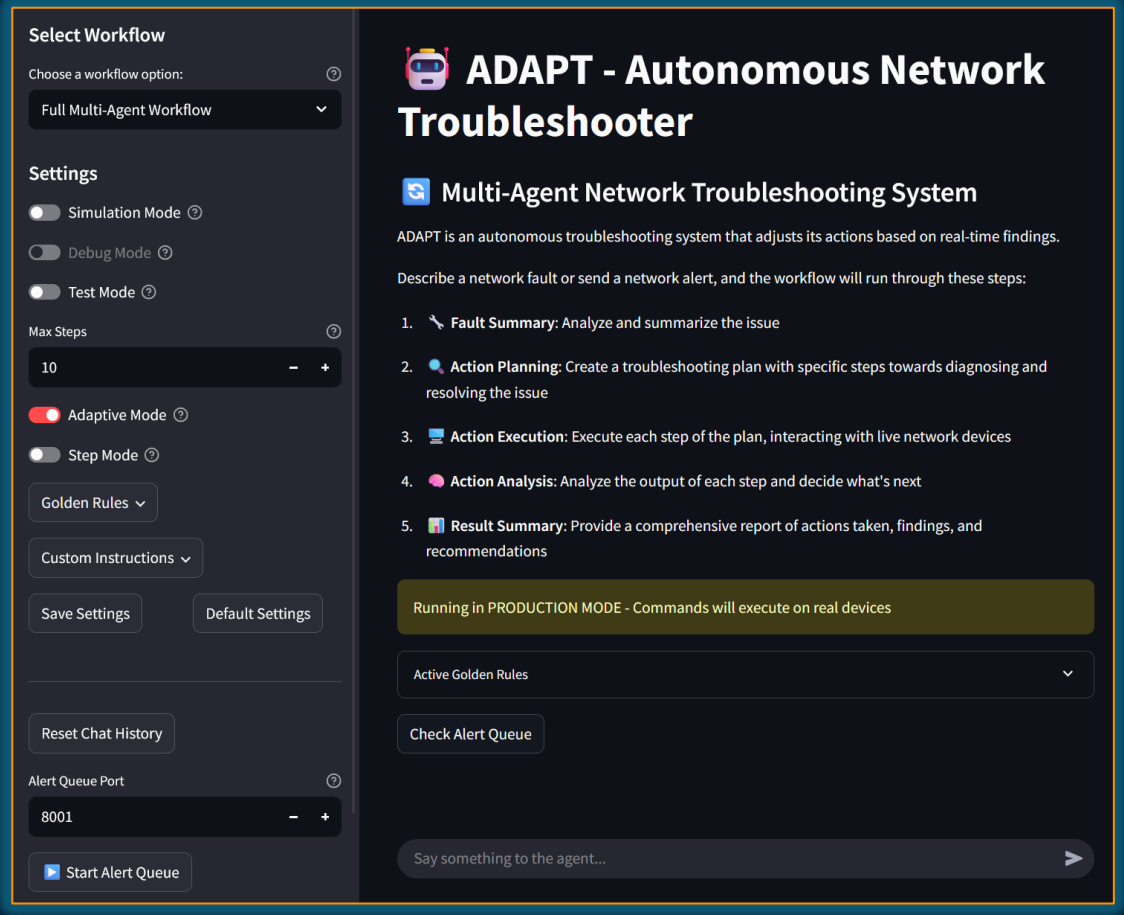
Agent Orchestrator

LangGraph

- Graph-based architecture for multi-agent, stateful, and reactive workflows
- Benefits: Stateful workflows, human-in-the-loop, conditional branching logic, parallel agent interactions

Additional Components

- **LLMs:** GPT-4.1 / GPT-o4-mini
- **UI:** Streamlit
- **Alert Queue API:** FastAPI
- **Observability:** Pydantic Logfire
- **Vector DB:** Coming soon!

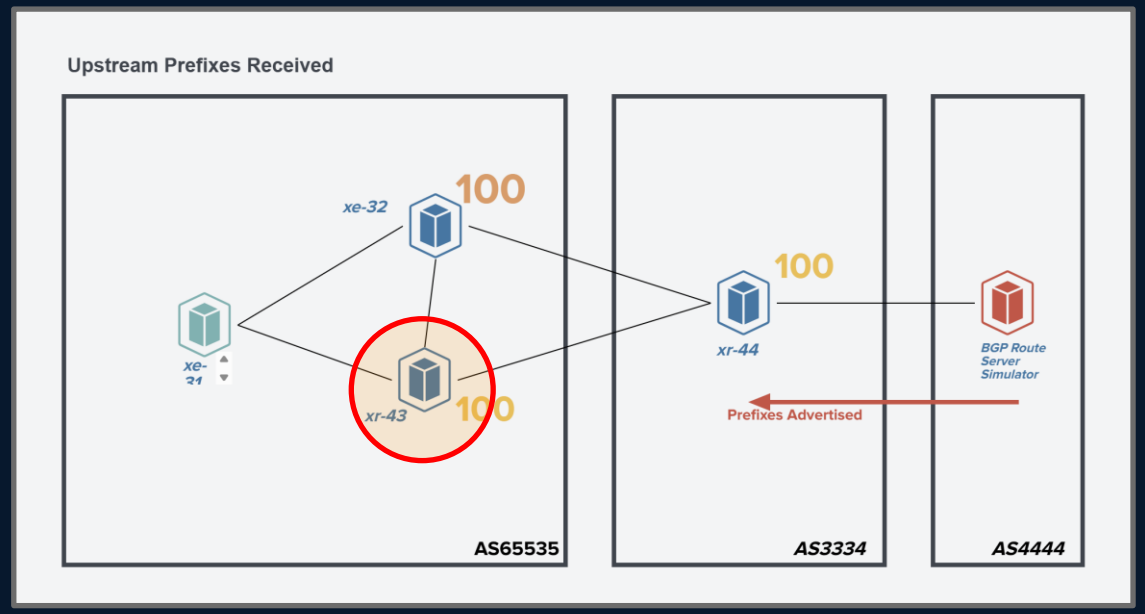


*The AI agent landscape is **constantly** changing*
*Prioritize understanding **capabilities**; don't over-rotate on tools*

Closed-loop automation in action: BGP prefix anomaly

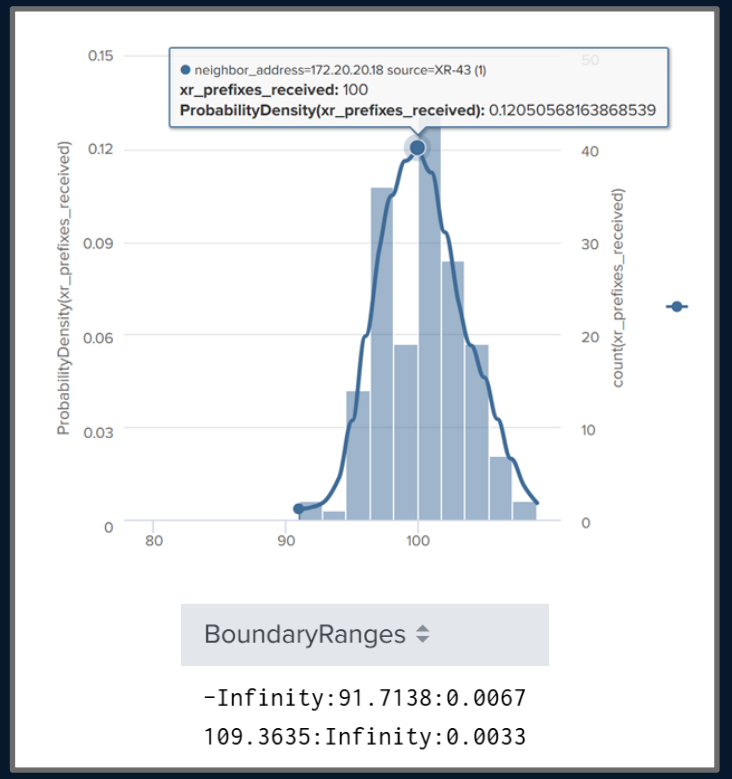
YANG path monitored:

Cisco-IOS-XR-ipv4-bgp-oper:bgp/instances/instance/instance-active/default-vrf/afs/af/neighbor-af-table/neighbor/af-data/prefixes-accepted



Failure scenario:

BGP prefix accepted anomaly for XR-43 -> Investigate RCA and remediate



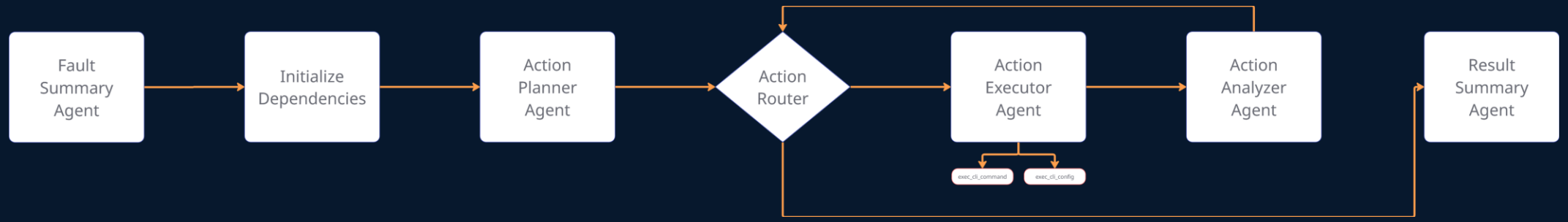
Model for Anomaly Detection:

BGP prefixes accepted count per neighbor

DEMO:
Closed-loop automation for BGP prefix anomaly

Walkthrough of multi-agent control flow

LangGraph

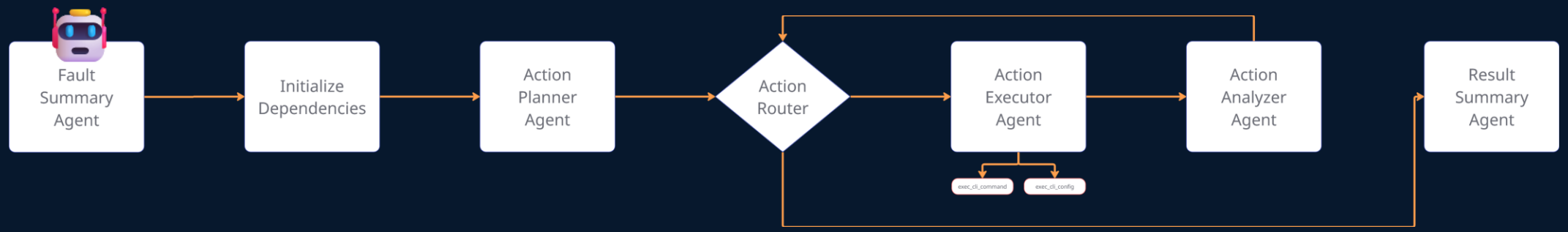


Input State

Objectives

Output State

Walkthrough of multi-agent control flow



Input State

- Raw Alert Data
- Golden Rules

Objectives

LangGraph Node

- Invoke PydanticAI agent

PydanticAI Agent

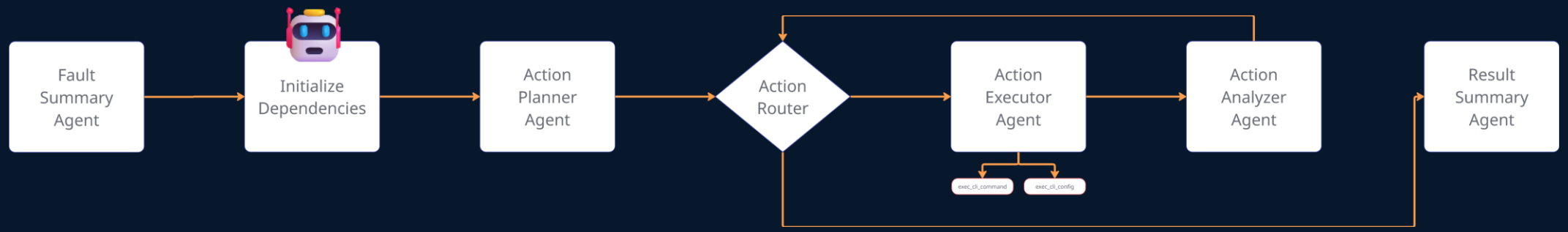
- Translates raw alert details (typically JSON) into a well-defined summary of the fault
- Extracts metadata from alert that can be used for future troubleshooting

Fault Summary Agent

Output State

- Fault Summary

Walkthrough of multi-agent control flow



Input State

- Settings
- Fault Summary

Objectives

LangGraph Node

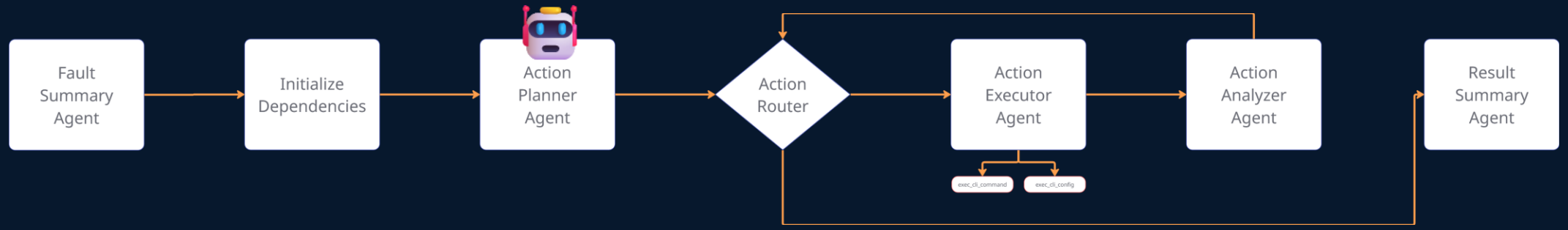
- Load network device inventory data
- Initialize Netmiko device driver
- Retrieve basic device facts (e.g. OS Ver, Interfaces, etc)
- Load Test Data when in Test Mode

Initialize Dependencies

Output State

- Device Inventory
- Device Facts
- Device Driver
- Test Data (Optional)

Walkthrough of multi-agent control flow



Input State

- Settings
- Fault Summary
- Device Facts
- Golden Rules
- Custom Instructions
- Test Data

Objectives

LangGraph Node

- Invoke PydanticAI Agent

PydanticAI Agent

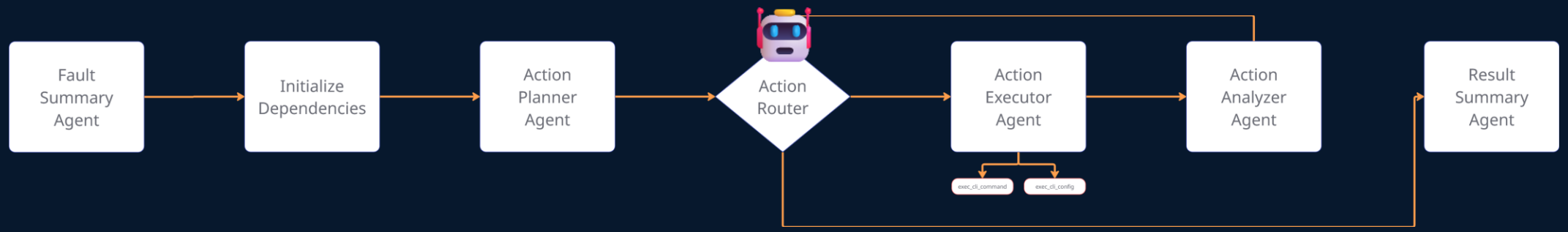
- Builds a step-by-step action plan for diagnosing and remediating the fault
- Each step includes description, action type, commands, and expected output
- Incorporates Custom Instructions into plan

Action Planner Agent

Output State

- Action Plan (List of Steps)
- Action Plan History
- Action Plan Remaining
- Current Step Index

Walkthrough of multi-agent control flow



Input State

- Settings
- Action Plan History
- Action Plan Remaining

Objectives

LangGraph Node

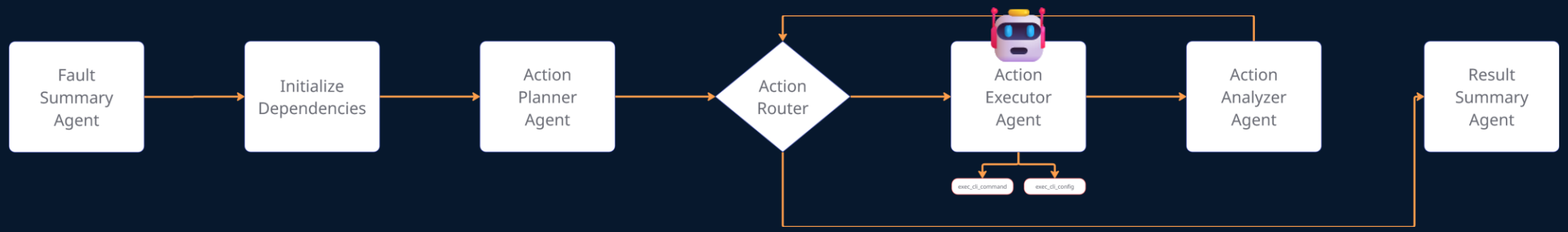
- Routes to the next LangGraph node based upon analysis of the latest action results
- Updates state information to set the “Current Step” details for other agents
- Invokes “human-in-the-loop” when action type is “config” or “exec” to request approval
- When receiving first step, routes to Action Executor Agent

Action Router (Part 1)

Output State

- Current Step
- Current Step Index
- Action Plan History
- Action Plan Remaining

Walkthrough of multi-agent control flow



Input State

- Current Step
- Device Driver
- Device Facts
- Test Data
- Golden Rules

Objectives

LangGraph Node

- Invoke PydanticAI Agent

PydanticAI Agent

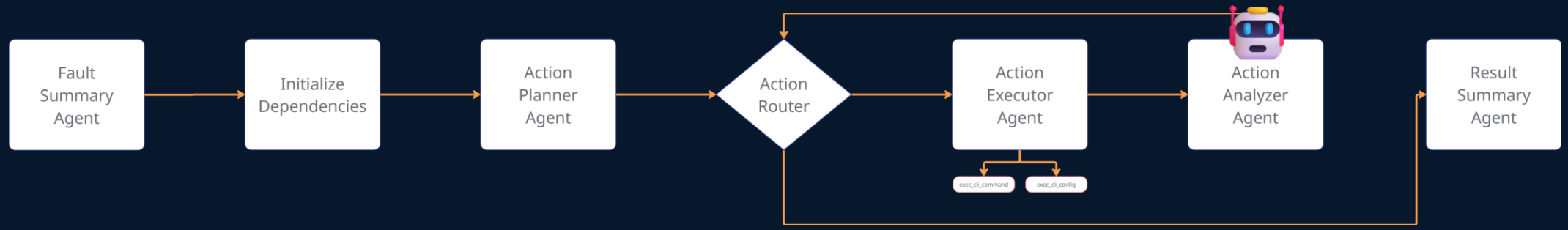
- Execute action via available tools and return the tool output (e.g. CLI “show” cmd output)
- If Simulation Mode enabled, use LLM to simulate output of a network device
- If Test Mode enabled, inject CLI test data

Action Executor Agent

Output State

- Execution Result
- Action Executor History

Walkthrough of multi-agent control flow



Input State

- Execution Result
- Fault Summary
- Current Step
- Device Facts
- Action Plan History
- Action Plan Remaining
- Custom Instructions
- Settings
- Golden Rules

Objectives

LangGraph Node

- Invoke PydanticAI Agent
- Appends Analysis Report to Current Step

PydanticAI Agent

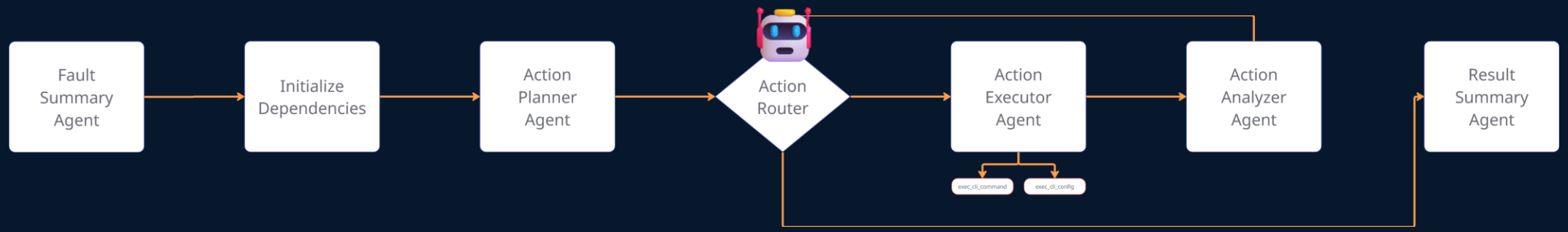
- Analyzes results of most recent action
- Determines next workflow action: continue, escalate, resolve, or revise the plan
- Revises action plan as needed

Action Analyzer Agent

Output State

- Current Step
- Action Plan Remaining

Walkthrough of multi-agent control flow



Input State

- Settings
- Action Plan History
- Action Plan Remaining

Objectives

LangGraph Node

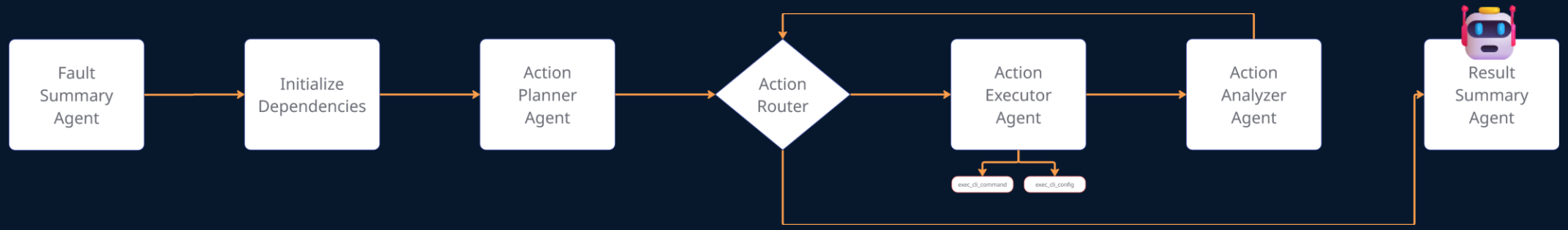
- Routes to the next LangGraph node based upon analysis of the latest action results
- Updates state information to set the “Current Step” details for other agents
- Invokes “human-in-the-loop” when action type is “config” or “exec” to request approval
- When analysis indicates “escalate” or “resolved”, routes to Result Summary Agent

Action Router (Part 2)

Output State

- Current Step
- Current Step Index
- Action Plan History
- Action Plan Remaining

Walkthrough of multi-agent control flow



Input State

- Execution Result
- Fault Summary
- Current Step
- Device Facts
- Action Plan History
- Action Plan Remaining
- Custom Instructions
- Settings
- Golden Rules

Objectives

LangGraph Node

- Invoke PydanticAI Agent
- Packages human readable and JSON reports

PydanticAI Agent

- Summarizes root cause, successful and failed actions, key findings, and resolution status
- Recommends next steps based on remaining issues or escalation needs

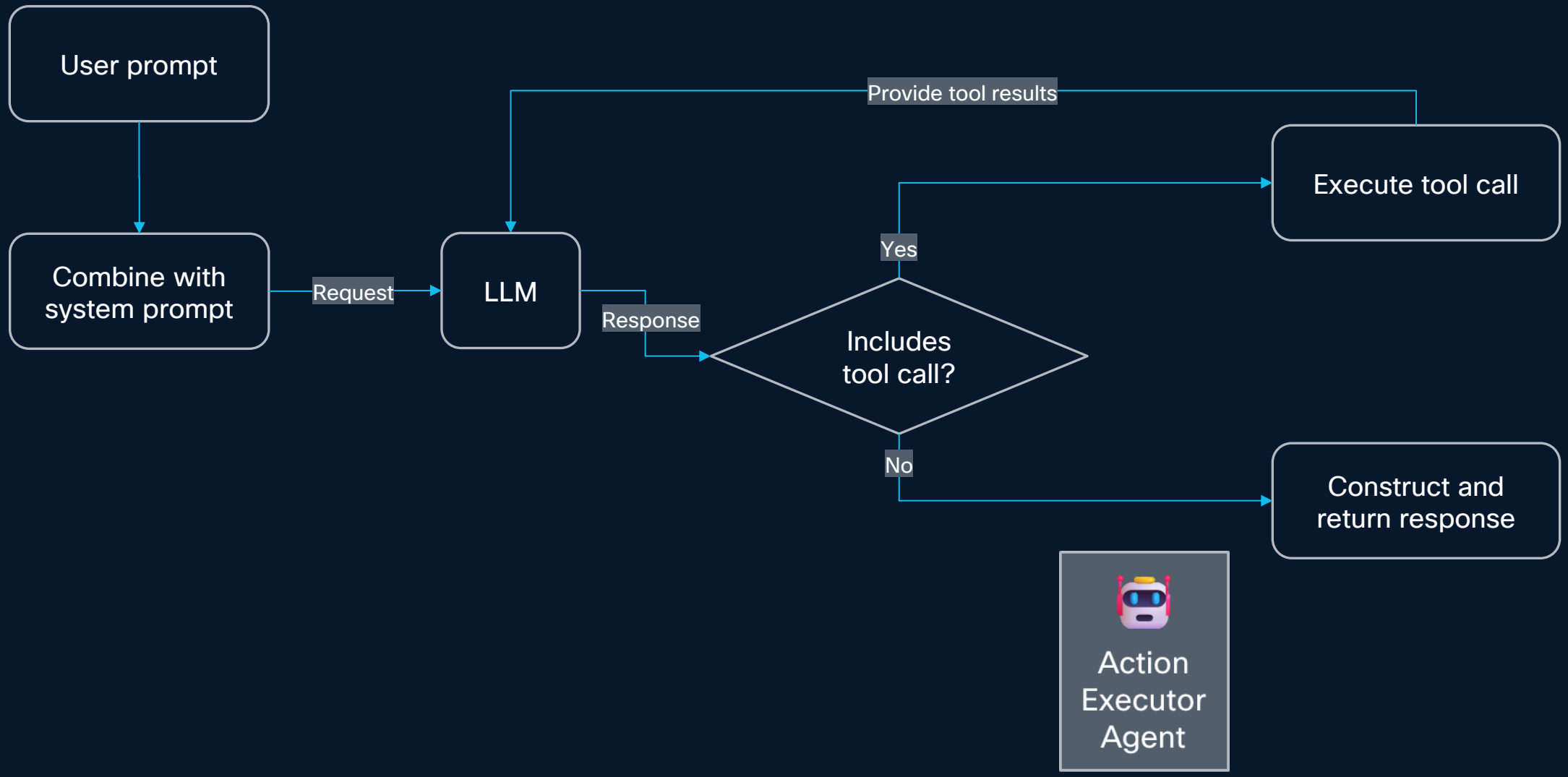
Result Summary Agent

Output State

- Result Summary

The building blocks of an AI agent

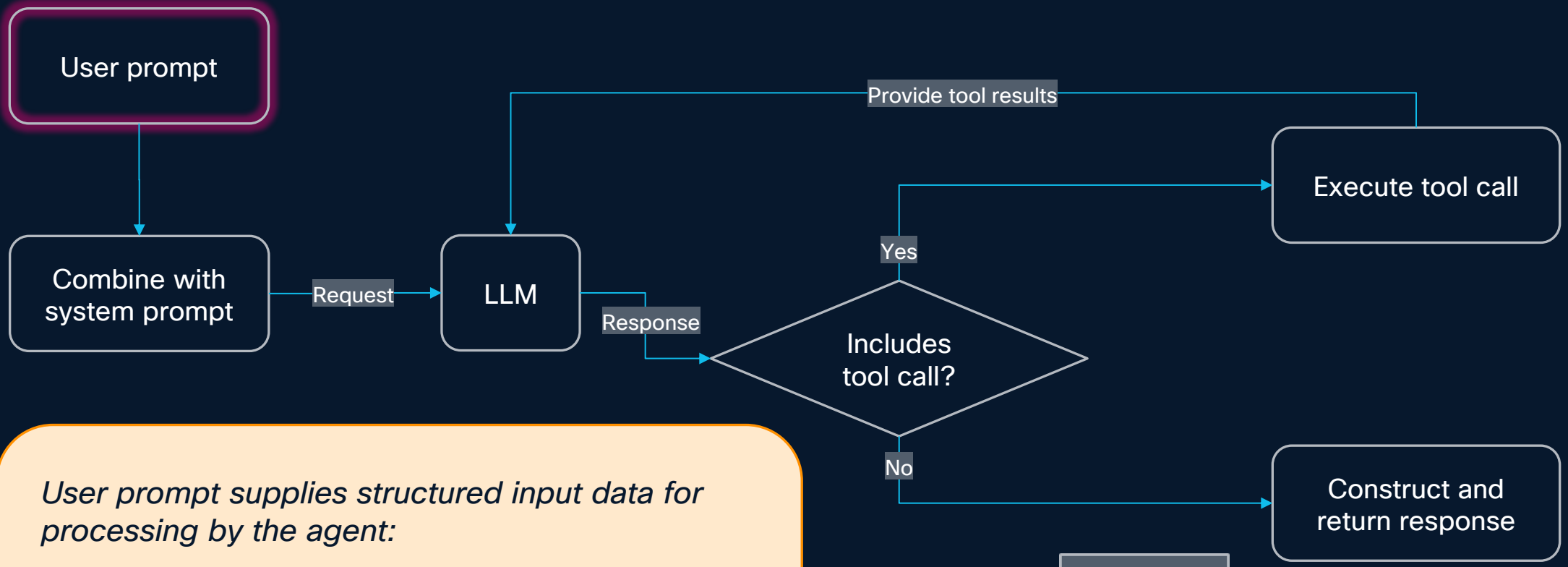
PydanticAI



cisco

The building blocks of an AI agent

PydanticAI



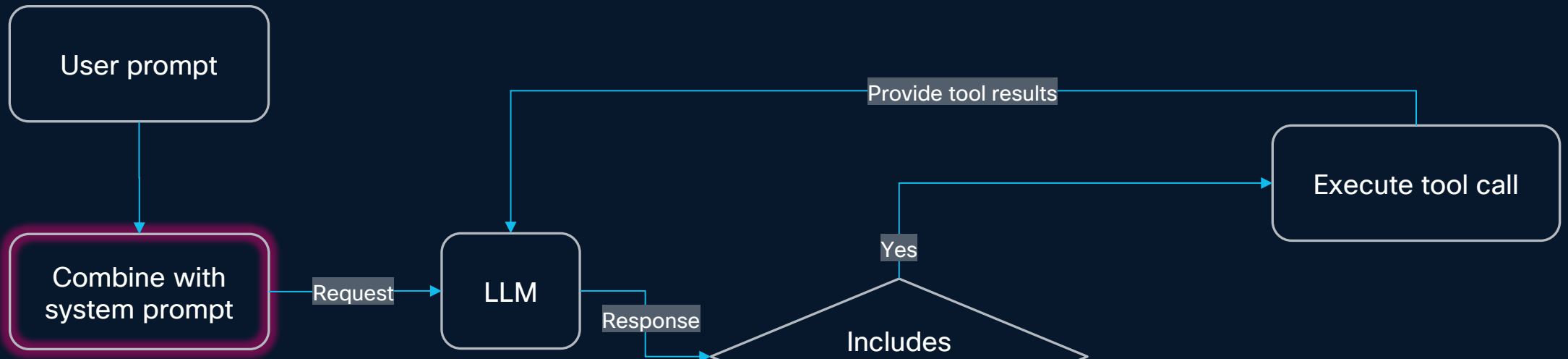
User prompt supplies structured input data for processing by the agent:

```
device_facts: {<details of target device>}
current_step: {<current step being executed,
including CLI commands}
simulation_mode: <true|false>
```

Action Executor Agent

The building blocks of an AI agent

PydanticAI

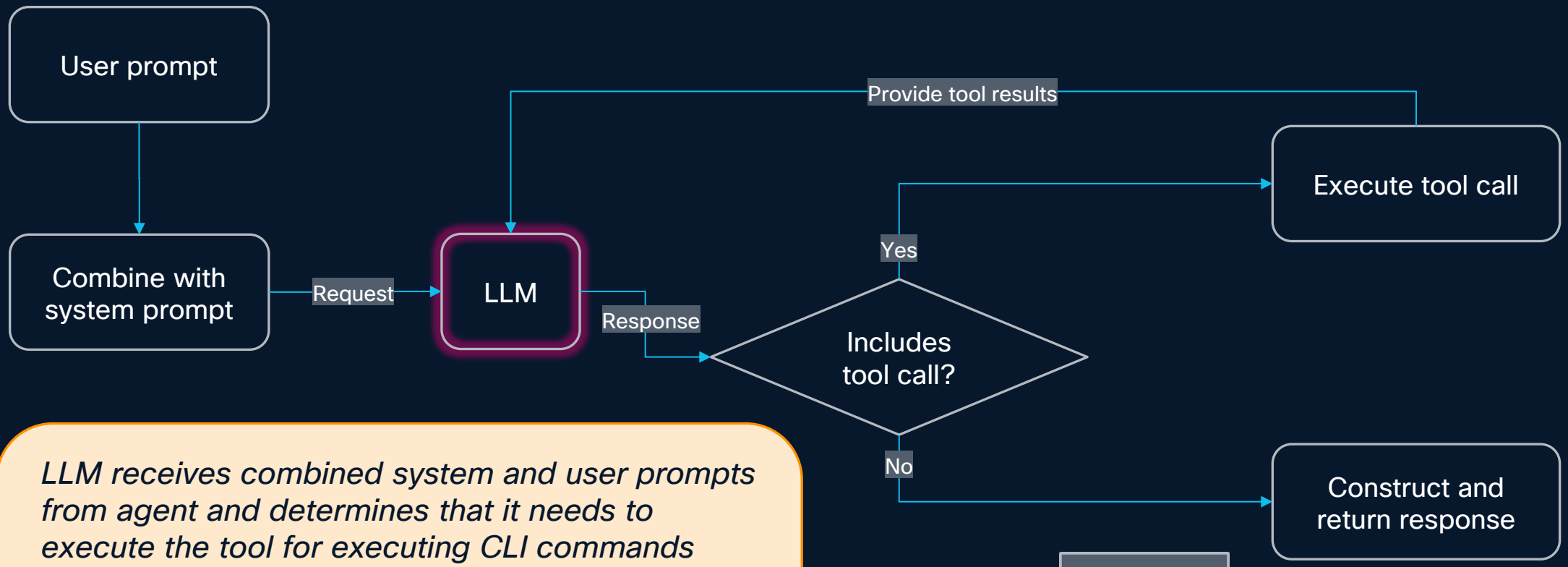


System prompt sets role and scope of the agent:

```
You are Action Executor Agent, responsible
**only** for running (or simulating) CLI
commands that appear in a single
troubleshooting action-plan step.
You have the following tools available to you…
Your input and output should be structured as
follows…
```

The building blocks of an AI agent

PydanticAI



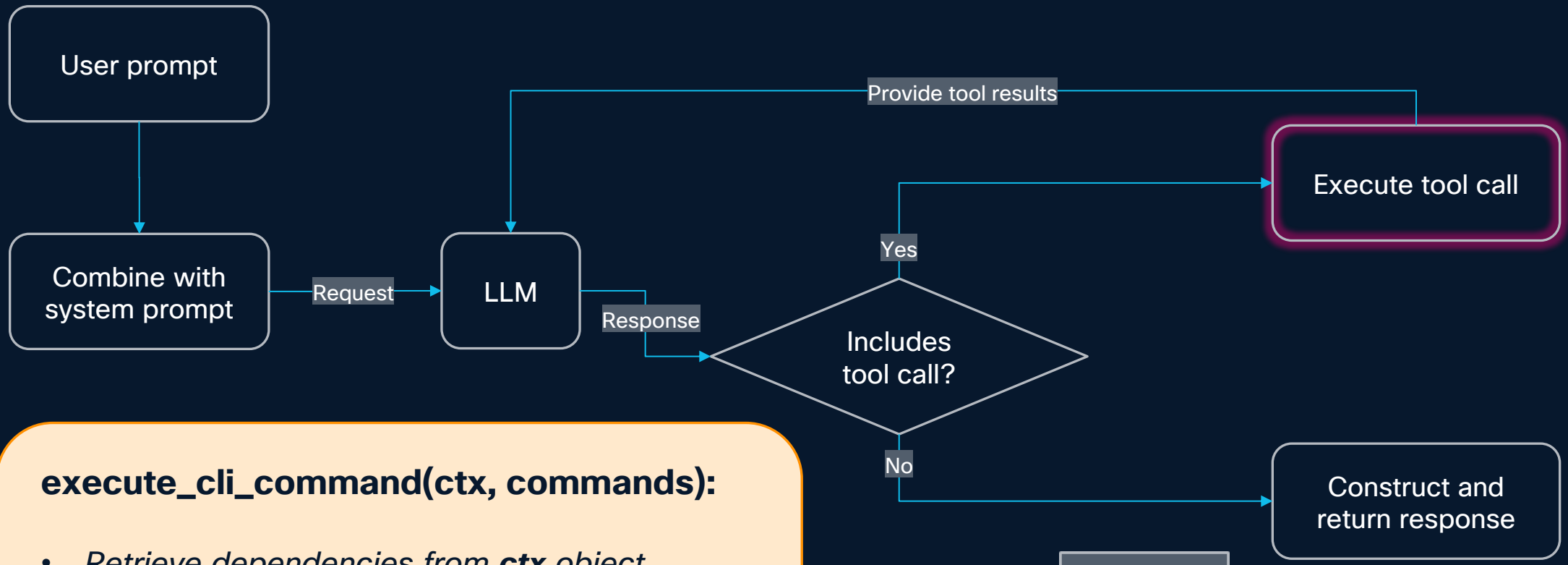
LLM receives combined system and user prompts from agent and determines that it needs to execute the tool for executing CLI commands

LLM Thinking: **current_step** includes an **action_type** of "diagnostic" with associated commands present. Therefore, I will call tool **execute_cli_commands(commands)**

cisco

The building blocks of an AI agent

PydanticAI



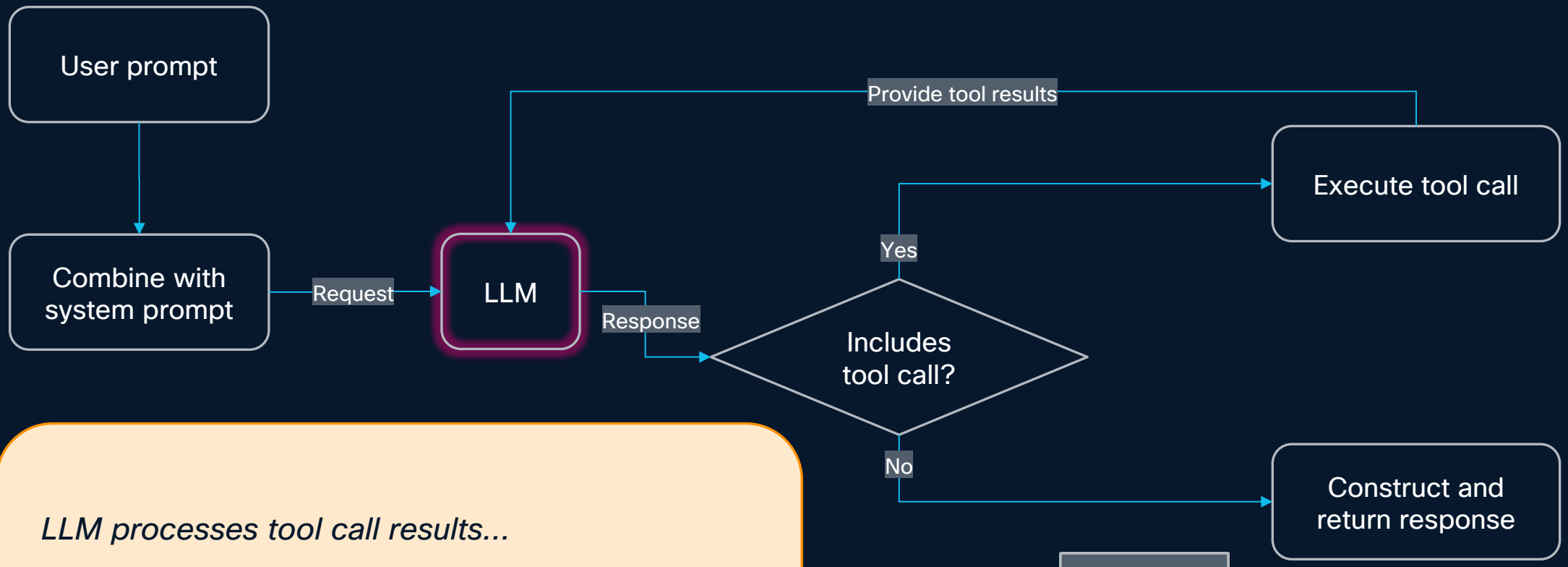
execute_cli_command(ctx, commands):

- *Retrieve dependencies from **ctx** object, including Netmiko driver for target device.*
- *Execute CLI diagnostic commands against target device and return output.*

Action Executor Agent

The building blocks of an AI agent

PydanticAI



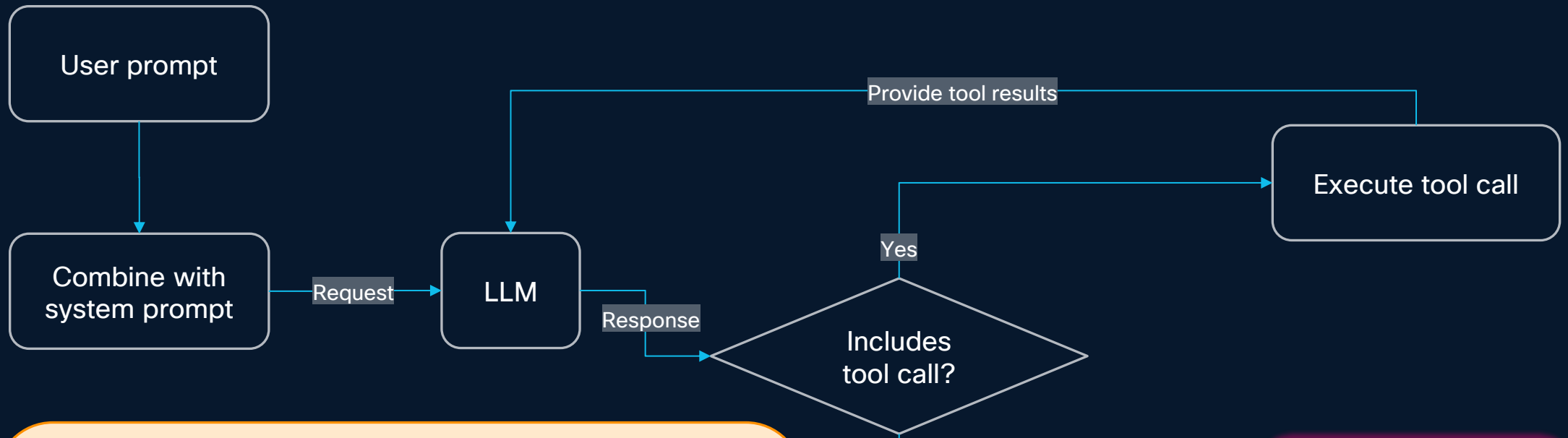
LLM processes tool call results...

LLM Thinking: "I have the CLI command output that I need! Let's return a response to the user."

Action Executor Agent

The building blocks of an AI agent

PydanticAI



Agent generates and returns structured output per system prompt and Pydantic AI specs:

```
description: <description of actions taken>
command_outputs: [<list of cmds with output>]
errors: [<list of errors encountered>]
```

Action Executor Agent

Lessons learned from our multi-agent system prototype

GenAI is like a box of chocolates. You never know what you’re gonna get.

Inspired by the great philosopher of our time, Forrest Gump

“You never know what you’re gonna get…”

Step 1: Check current BGP prefix count for the affected neighbor

- 🔄 **Action Type:** diagnostic
- 💻 **Commands:**
  - `show bgp neighbors 172.20.20.18`
- 🔍 **Expected Output:** Output shows the number of prefixes received from neighbor 172.20.20.18; verify if it remains at or near 84
- ⚠️ **Requires Approval:** No

Step 1: Check current BGP prefix count for neighbor 172.20.20.18 to verify if the outlier condition persists

- 🔄 **Action Type:** diagnostic
- 💻 **Commands:**
  - `show bgp neighbor 172.20.20.18`
- 🔍 **Expected Output:** Output lists the number of prefixes received from 172.20.20.18; verify if it remains at or near the anomalous count of 84
- ⚠️ **Requires Approval:** No

📋 Analysis of Step 4 Results

Commands Analyzed:

- `show configuration commit changes 1000000078`

📊 Analysis:

The configuration diff for commit 1000000078 shows the creation of a route-policy 'accept_from_AS_3334' affecting prefix handling, confirming a policy change that likely impacted BGP prefixes from neighbor 172.20.20.18. The next step is to rollback this commit as per the remediation plan.

📋 Analysis of Step 4 Results

Commands Analyzed:

- `show configuration commit changes 1000000078`

📊 Analysis:

The commit diff for ID 1000000078 only shows the definition of a route-policy (accept_from_AS_3334) with no neighbor 172.20.20.18 statements added, removed, or modified, leaving ambiguity whether the anomaly-triggering policy was applied to the neighbor.

Lessons learned from our multi-agent system prototype

Context is King

Design with state data in mind

Vibe coding is overrated...

... but agent-assisted coding with MCP tools isn't

It's all in the prompts: iterate and test

There's still hope for the human race! (for now) 

Phase 4:
YOU-driven innovation

Now it’s your turn!

- Walk through our use cases or experiment with your own
  - End-to-end MDT setup tutorials with configurations
  - CML topology and sample configurations for XE, NX and XR
  - Recordings and documentation to setup your own AIOps Playground
- dCloud Labs
  - YANG Suite
  - Model-driven Telemetry with Machine Learning
  - AIOps Playground
- Experiment with and contribute to Project “ADAPT”
- Helpful resources that we used on our journey

cs.co/practical-aiops

practical-aiops@cisco.com

Coming in late 2025 for early trial

AIOps-based consultative services for automated network assurance outcomes

Powered by Cisco Workflows → **Rule-based response** + **Cognitive agentic response**

To be available through Cisco Lifecycle Services

Cisco **Customer Experience**

Complete Your Session Evaluations

Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to win 1 of 5 full conference passes to Cisco Live 2026.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn exclusive prizes!

Complete your surveys in the Cisco Live mobile app.

Continue your education

Visit the Cisco Showcase for related demos

Book your one-on-one Meet the Engineer meeting

Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs

Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact us at: practical-aiops@cisco.com

Thank you


CISCO

REFERENCE SLIDES

Understanding AI agent behavior

① What's the weather in London right now?

User prompt

Combine with system prompt

Request

LLM

③

Response

Includes tool call?

Yes

Execute tool call

④ Use weather API to retrieve current weather in London

⑤

```
{"location": "London, UK", "temperature": "15°C", "humidity": "80%",
"wind_speed": "10 km/h", "description": "Cloudy with occasional light rain"}
```

Provide tool results

⑥

"I have the current weather in London!"

Call tool get_current_weather("London")

No

Construct and return response

⑦ "The current weather in London, UK is 15°C, with 80% humidity and a light rain. The wind speed is 10 km/h."

② You are a helpful and knowledgeable AI weather assistant. Your primary goal is to provide accurate and up-to-date weather information to users based on their queries.

You have access to the following tools:

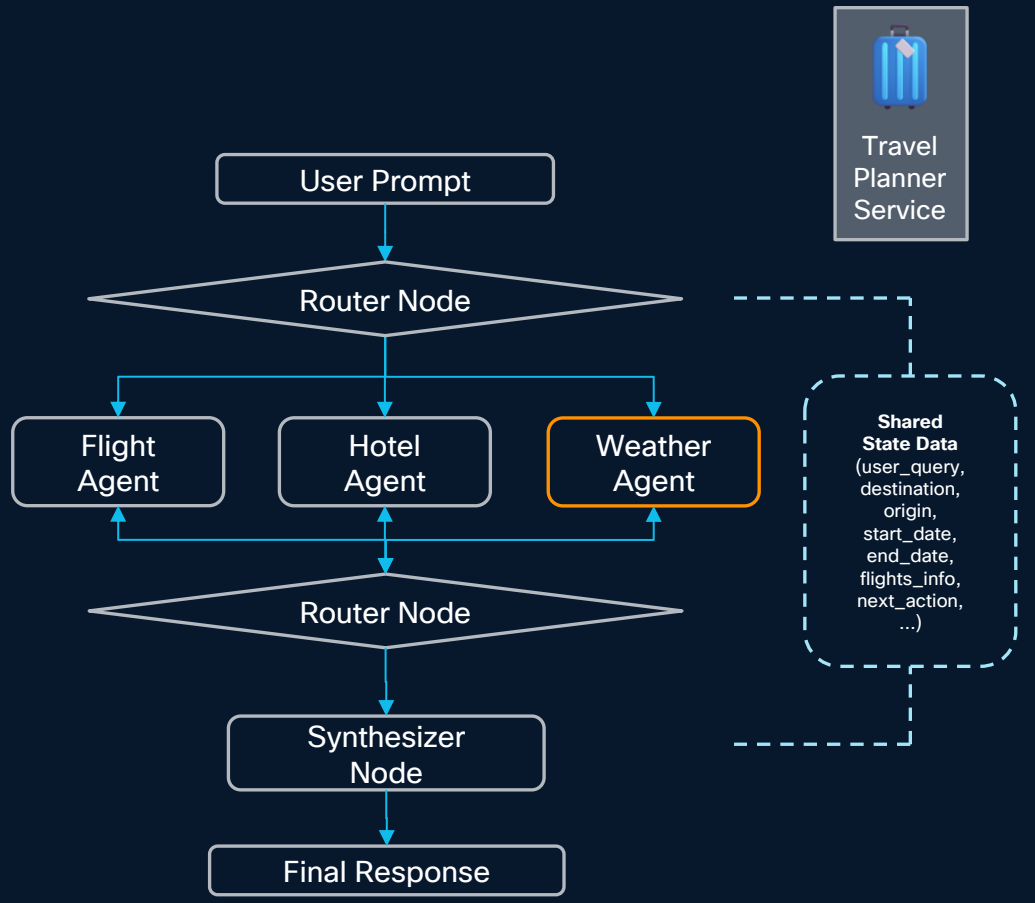
- `get_current_weather(location: str)`: Fetches the current weather conditions (temperature, humidity, wind speed, general description) for a specified city.
- `get_five_day_forecast(location: str)`: Provides a 5-day weather forecast (daily high/low temperatures, precipitation chance, general description) for a specified city.

When a user asks for weather information, you must use the appropriate tool(s) to retrieve the data.
- If the user asks for current weather, use `get_current_weather`.
- If the user asks for a forecast, use `get_five_day_forecast`.
- If the user's request is ambiguous or lacks a specific location, ask for clarification.
- After obtaining the weather data, summarize it clearly and concisely for the user.
- If you are unable to get weather information for a location, inform the user.

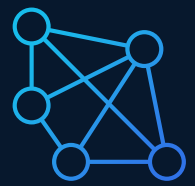
Dynamic workflows with multi-agent systems

Benefits of Multi-Agent Systems

- **Modularity:** Each agent is specialized, making the system easier to develop, debug, and scale.
- **Scalability:** New agents can be added for new domains without rehauling the entire system.
- **Robustness:** If one agent fails or an API is down, the system can potentially still provide partial information or gracefully handle the error.
- **Complexity Handling:** Breaks down complex user requests into manageable sub-problems.
- **Maintainability:** Easier to update or improve specific functionalities by focusing on a single agent.



Design considerations for a multi-agent system



What is my system's purpose and scope? Problem domain, desired outcomes, boundaries, success criteria	Diagnose/remediate network fault alerts by interfacing with network devices, analyzing diagnostic data, executing repair actions, and reporting results
What are the roles and responsibilities of agents? Task modularity, system/user prompts, tools needed	1. Summarize network fault details 2. Build action plan 3. Execute next action with CLI tools 4. Analyze action result 5. Decide what to do next 6. Summarize results
How will my workflow be structured? Step logic, decision points, external interactions	A step-based action plan with dynamic decisions based upon analysis of results; can continue to next step, escalate, resolve, or modify action plan.
What state data will flow through this system? Context, I/O, dependencies, schemas, memory, RAG	Network alert data, context-specific instructions, golden rules, settings, action plan, device facts, diagnostic data, action analysis, test data
What guardrails will be in place to restrict unintended actions? Reflection, human-in-the-loop, control limits, error-handling	"Human-in-the-loop" for potentially impactful changes, LLM-driven post-check validation actions for any changes, action count limits, error handling
How will the system be evaluated for continuous improvement? Metrics, feedback loops, testing, LLM-based evaluation	On-demand LLM-based generation and execution of test scenarios with synthetic data, LLM-based evaluation of test results

Types of algorithms for outlier detection models

Splunk's Machine Learning Toolkit Focuses on Density-Based Methods

Statistical, Rule-Based Thresholding Methods

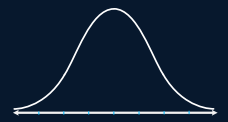
"Set fixed alarms for data points outside a normal range, based on historical averages and spread."

Time Series Forecasting-Based Methods

"Predict future values based on past patterns; anomalies are when actuals stray from predictions."

Reconstruction-Based Methods (Deep Learning)

"Learn to 'recreate' normal data; unusual data points can't be accurately reconstructed, signaling anomalies."



Density-Based Methods

"Analyze historical data to create a probability distribution curve, then set thresholds by specifying areas under the curve that represent anomalous behavior."

Splunk Machine Learning Toolkit

Types of algorithms for outlier detection models

Splunk's Machine Learning Toolkit Focuses on Density-Based Methods

Statistical, Rule-Based Thresholding Methods

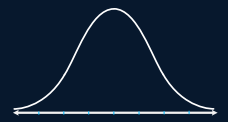
"Set fixed alarms for data points outside a normal range, based on historical averages and spread."

Time Series Forecasting-Based Methods

"Predict future values based on past patterns; anomalies are when actuals stray from predictions."

Reconstruction-Based Methods (Deep Learning)

"Learn to 'recreate' normal data; unusual data points can't be accurately reconstructed, signaling anomalies."



Density-Based Methods

"Analyze historical data to create a probability distribution curve, then set thresholds by specifying areas under the curve that represent anomalous behavior."

Splunk Machine Learning Toolkit

Types of algorithms for outlier detection models

Splunk's Machine Learning Toolkit Focuses on Density-Based Methods

Statistical, Rule-Based Thresholding Methods

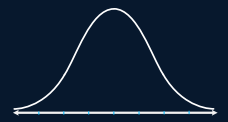
"Set fixed alarms for data points outside a normal range, based on historical averages and spread."

Time Series Forecasting-Based Methods

"Predict future values based on past patterns; anomalies are when actuals stray from predictions."

Reconstruction-Based Methods (Deep Learning)

"Learn to 'recreate' normal data; unusual data points can't be accurately reconstructed, signaling anomalies."



Density-Based Methods

"Analyze historical data to create a probability distribution curve, then set thresholds by specifying areas under the curve that represent anomalous behavior."

Splunk Machine Learning Toolkit

How machine learning solves problems for network engineers

Solving network engineering problems with ML

Forecasting

Predict future values based on recent trends

- Estimate future bandwidth requirements
- Estimate future platform resource demands (CPU, memory, etc)

Clustering

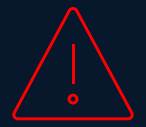
Identify common patterns or groups within data

- Create device health check rules directly from historical data
- Find patterns in log streams to support root cause analysis

Prediction

Identify relationships between data streams

- Routing table size & complexity vs memory utilization
- ARP rate impact on CPU load
- ACL size and complexity impact on TCAM



Outlier Detection

Identify activity that deviates from expected behavior

- Flag unusual spikes in traffic volume
- Identify unusual CPU load or memory usage
- Monitor BGP prefix counts

Why *Outlier Detection* is a great place to start with ML

Direct Impact on SLA and MTTx metrics

- Detect anomalies early, before SLA thresholds are violated
- Improve MTTD and MTTR with faster, smarter anomaly detection + automation

Dynamic, data-driven thresholding

- Replace time consuming creation and maintenance of static alert thresholds with adaptive, learning-based models

Reduced noise, smarter triage

- Eliminate alert fatigue and needless troubleshooting of false positive alerts
- Eliminate needless troubleshooting when alerts for key issues are missing

Catch issues early

- Identify early warning signs of potentially broader failures

SPL query format to capture training data

Convert counter metric to interval

Metric

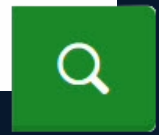
Metric Alias

```
| mstats rate_avg("infra-statistics.packets_received") AS "packets_received/s"
  WHERE index="mertics_data"
  earliest=-14d@d  latest=-1d@d
  BY source, interface_name
  span=5m
```

Only return data from this time slice

Also return these fields

SPL> “Search Processing Language”

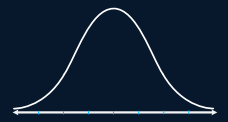


Analysis Granularity

_time	source	interface_name	packets-received/s
2025-05-08 00:09:00	xr-2	GigabitEthernet0/0/0/1	0.1311475409836065 6
2025-05-08 00:09:00	xr-2	GigabitEthernet0/0/0/2	0.147540983606557 37

How density algorithms enable outlier detection models

Important questions we will cover



Density-Based Methods

"Analyze historical data to create a probability distribution curve, then set thresholds by specifying areas under the curve that represent anomalous behavior."

Splunk Machine Learning Toolkit

- What is "*density*"?
- Why are *data shapes* important?
- What is a *probability distribution function* ("PDF")
- How we use PDFs to create outlier detection rules

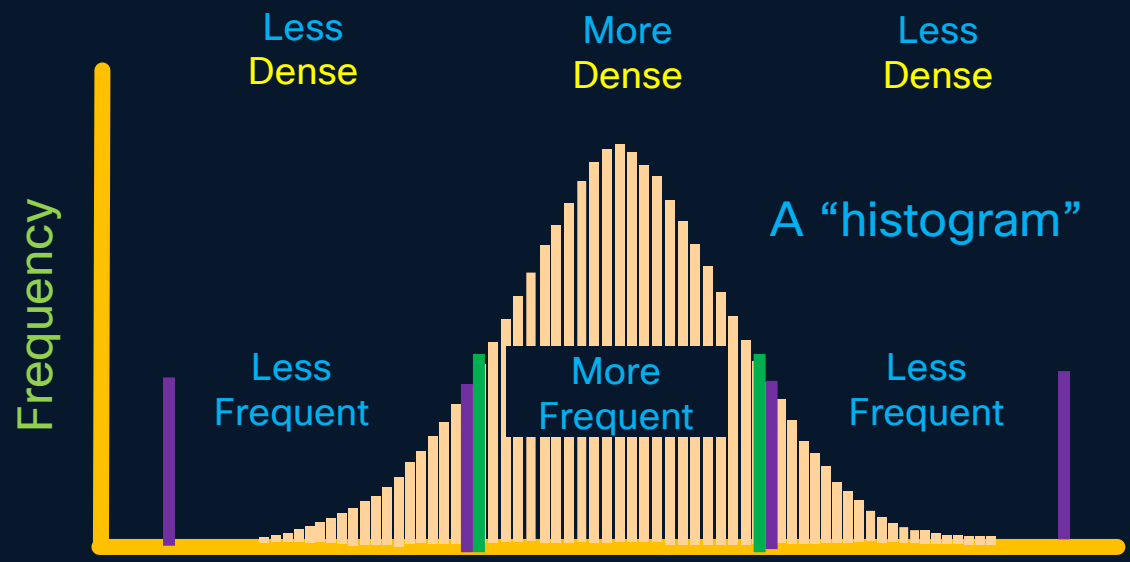
Producing a Data Shape - *Visually*

"Histograms Show Density: Where Your Data Is Common or Rare"

Historical Data

Sort Observations Into Size Buckets

Sort by Bucket Values

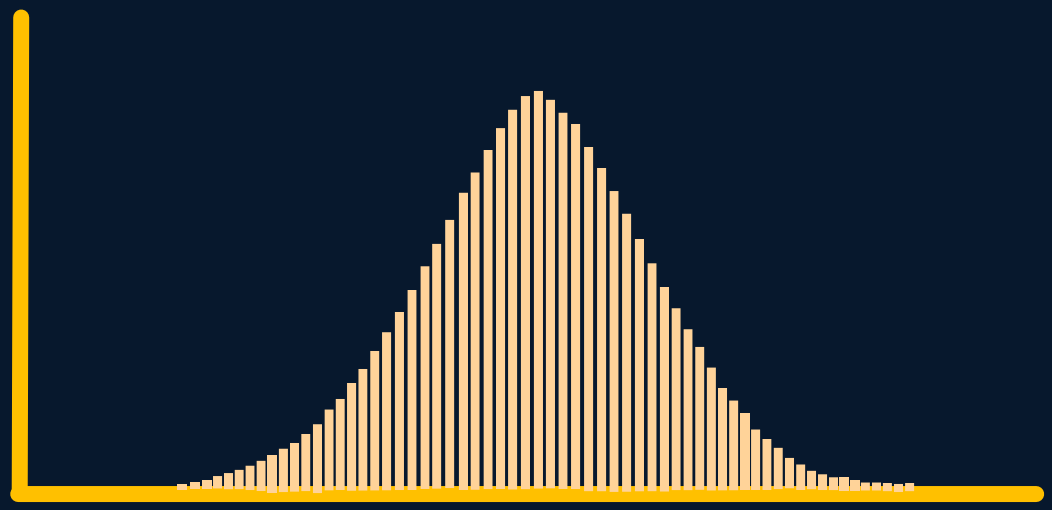


Observations Sorted into Buckets

Now we *visually* understand the data shape.

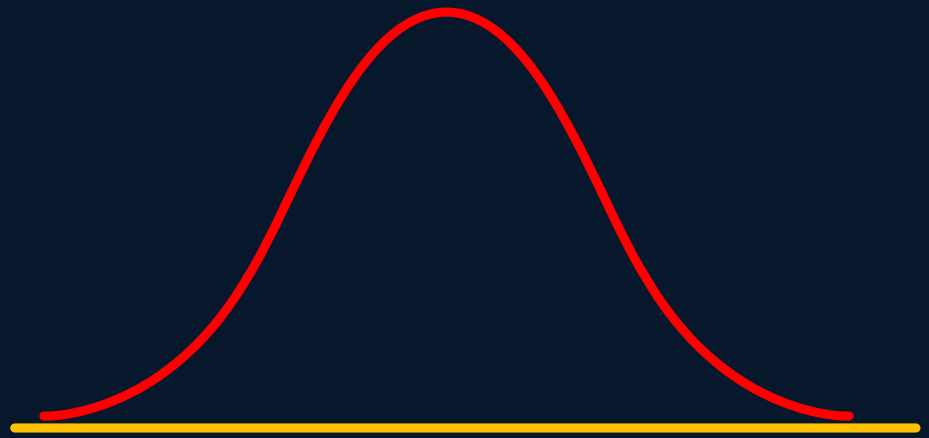
Understanding Data Shape – *Mathematically*

Histograms produce great visuals, but outlier detection models require *functions* to understand data shape



A provides a visual understanding of the shape of the data.

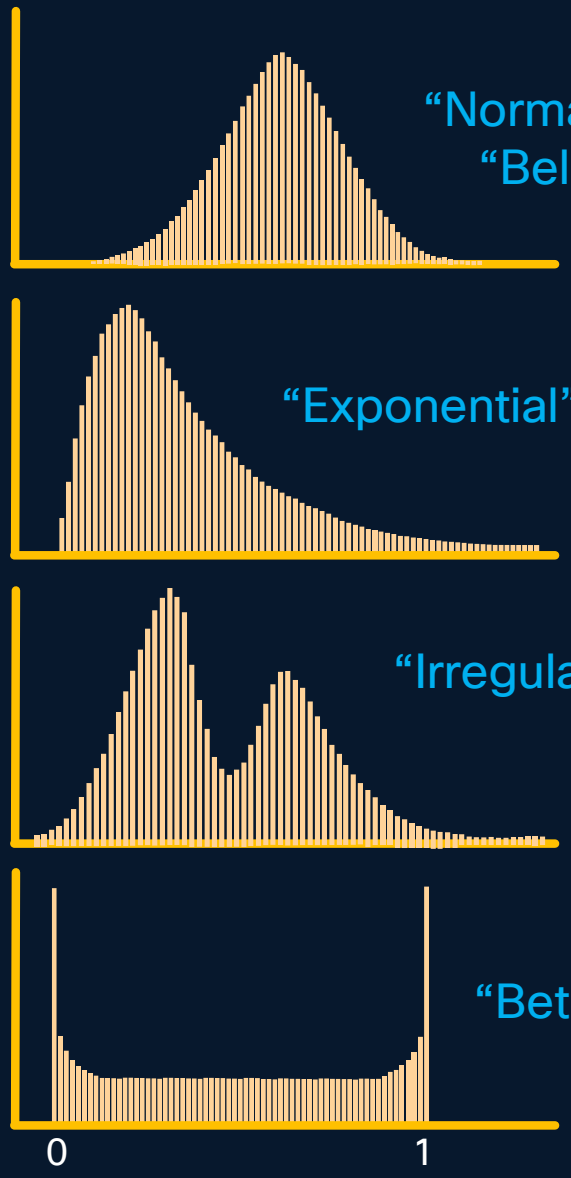
“Histogram”



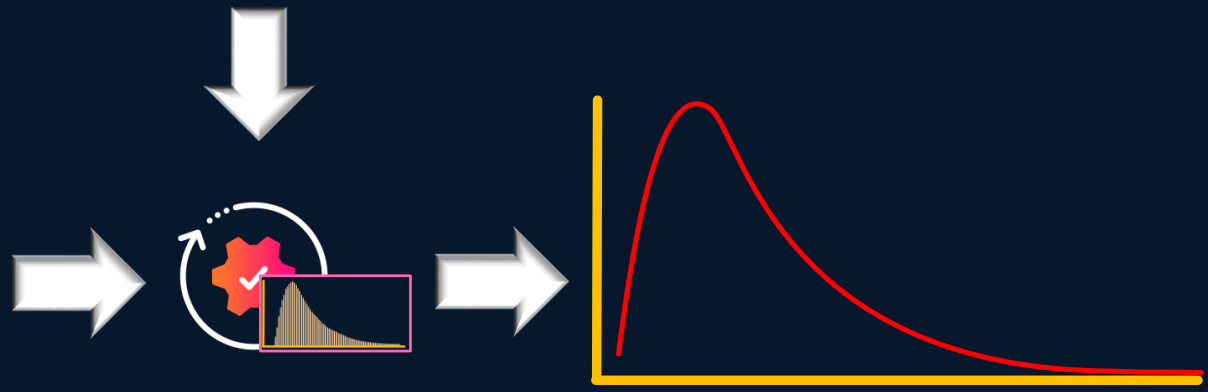
A mathematical model of the shape of the data.

“Probability Density Function”

So how do we calculate the Probability Density Function?



Historical Data



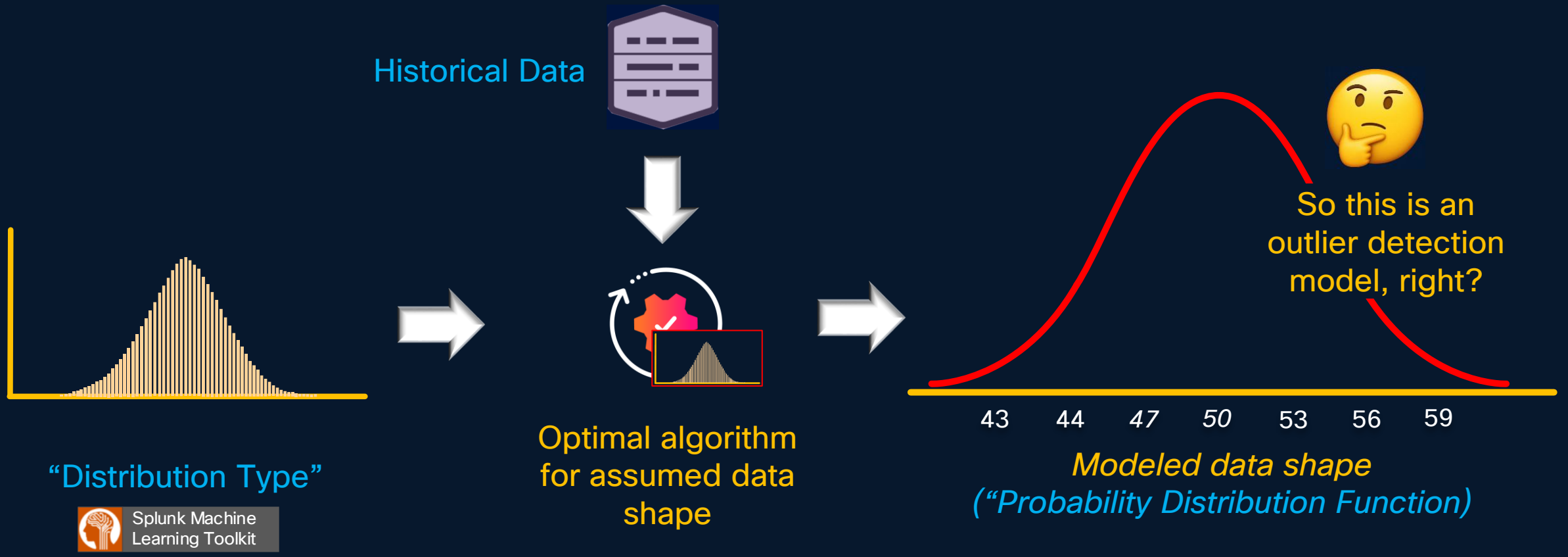
“Assumed data shape”

“Exponential”

Train model using *optimal* algorithm for this “expected” data shape

Modeled data shape (“Probability Distribution Function)

Why are probability distribution functions key to outlier detection models?

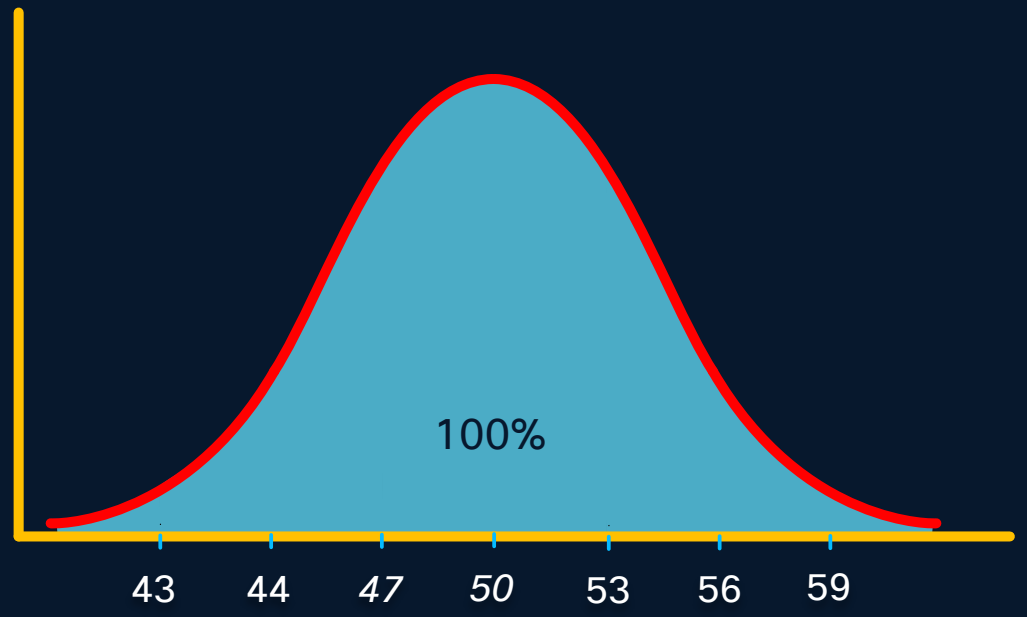


How functions enable mathematically-derived thresholds

Question: For the data set used to generate this probability density function, where do 99.7% of the most common observations occur?

The answer is in the probability density function:

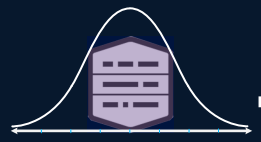
Probability Density Functions provide a way to translate a percentage of area under the curve to a range of observations



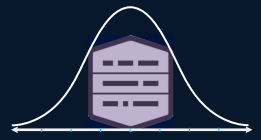
How functions enable mathematically-derived thresholds

Question: For the data set used to generate this probability density function, where do 99.7% of the most common observations occur?

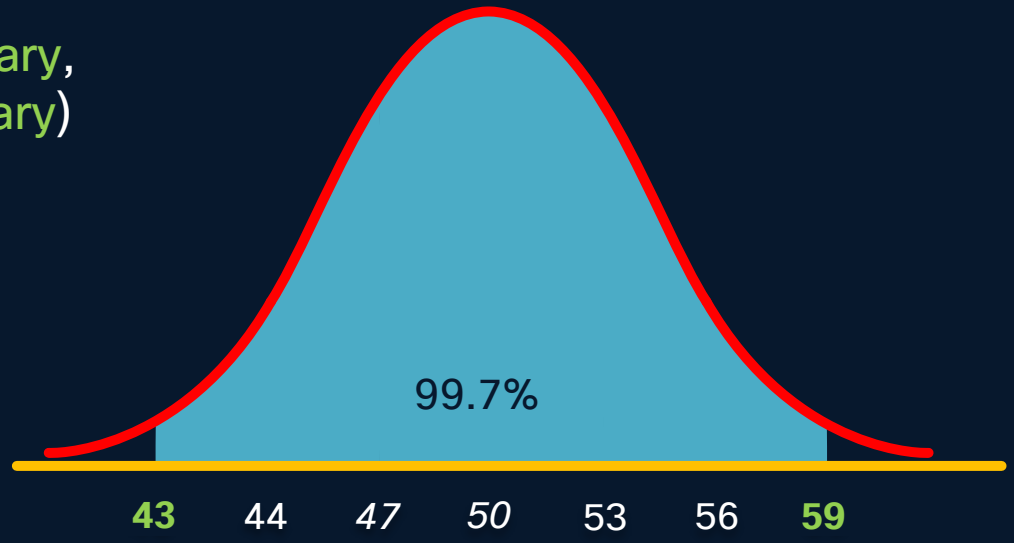
A *pseudo math* formula:



+ Typical_Range(**%_area**) = (lower_boundary, upper_boundary)



+ Typical_Range(**99.7%**) = (43 to 59)

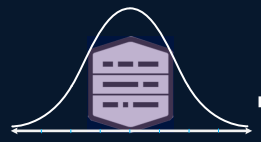


Answer: About 99.7% of the observations in the data set used to build the probability density function are between 43 and 59

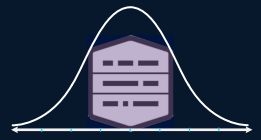
How functions enable mathematically derived thresholds

Question: For the data set used to this probability density function, where do .27% of the least likely observations occur? (i.e. the “outliers”)

A *pseudo* math formula:



+ Outlier_Range(**%_area**) = (lower_threshold, upper_threshold)



+ Outlier_Range(**.27%**) = (<43, >59)

Answer: For the specified percentage, observations are least likely to be observed greater than 59 and lower than 43

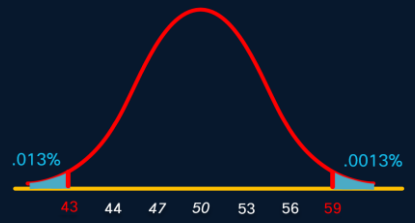
How functions enable mathematically derived thresholds

Question: For the data set used to this probability density function, where do .27% of the least likely observations occur? (i.e. the “outliers”)

“How rare should an observed value be before we are concerned about it?”

CDS

“Less than .27% of the time, please”



Derive thresholds values using function math

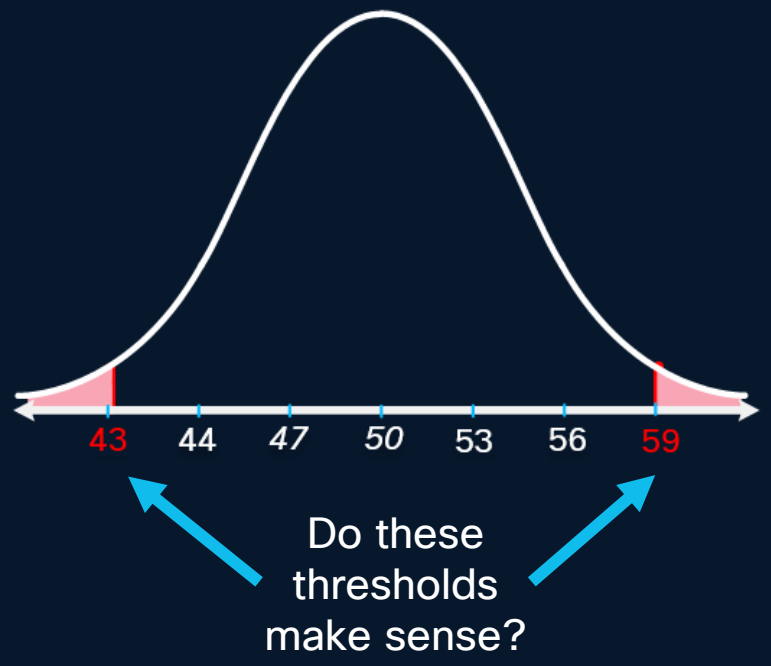
“Using the PDF, the threshold values are 43 and 59”

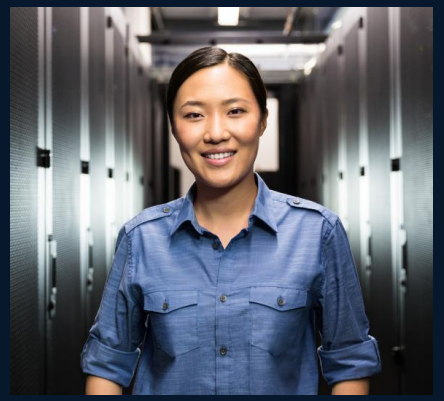
Outlier if < 43 or > 59

This is your outlier detection model!

So *now* we have our final outlier detection model, right?

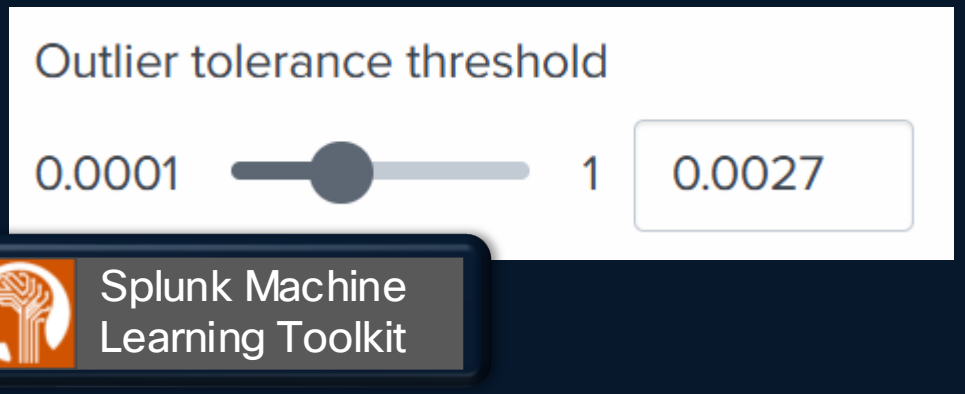
- *Not exactly…*
- Specifying outlier zones using probability percentages is how domain experts communicate with the model tuning process
- Outlier detection models are optimized when the model yields outliers which are also considered to be anomalies by the domain expert





Ms. D. Expert

MLTK Preview: Threshold tuning sequence

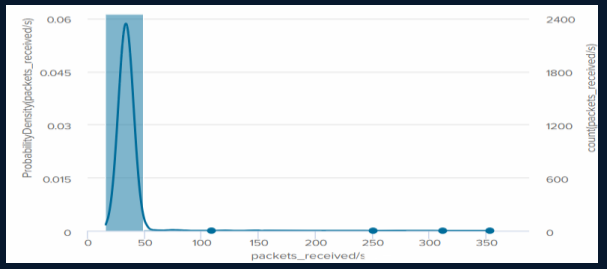


Specify Anomaly Sensitivity

Run Model

Detect Outliers

Examine flagged anomalies in training tata



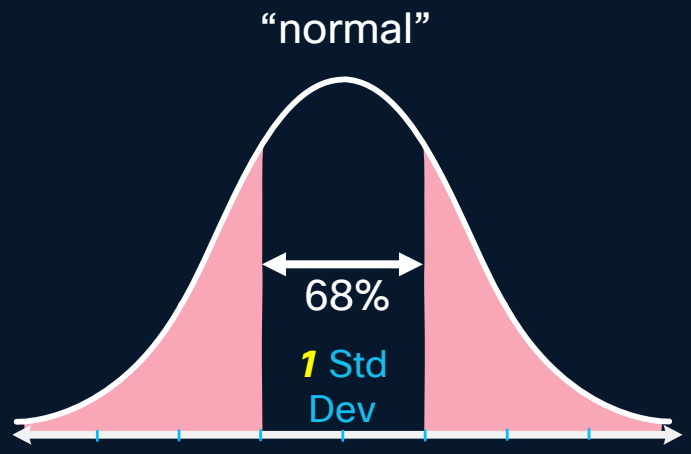
Correct Sensitivity?

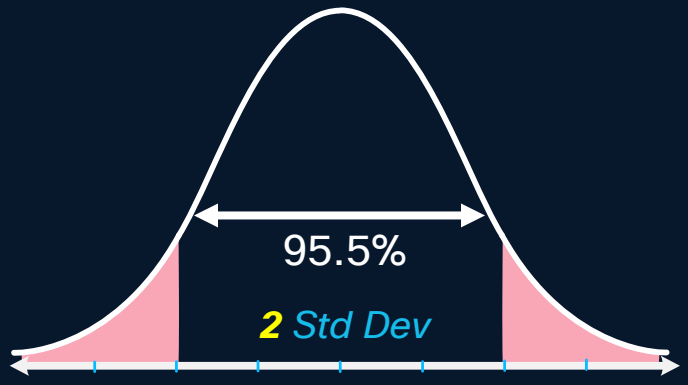
Not yet...

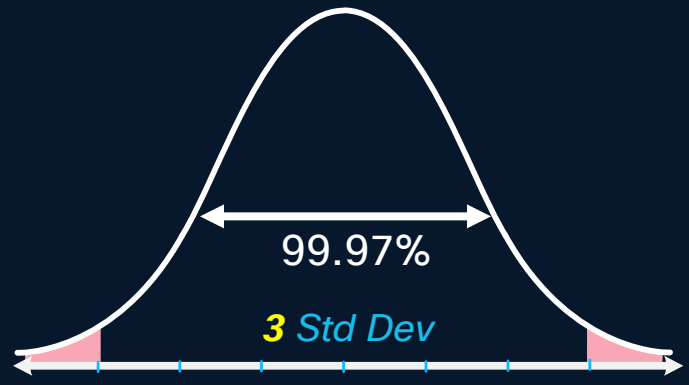
Yes!

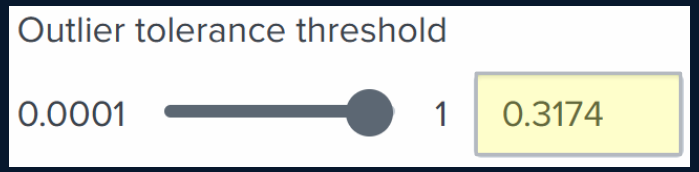
Tuned Model

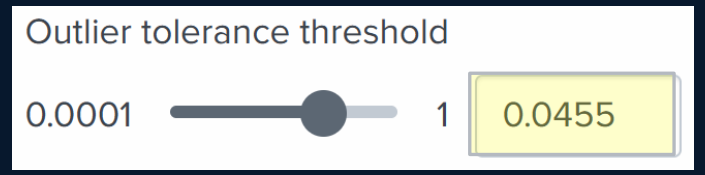
Turning outlier tolerance to find familiar rarity levels

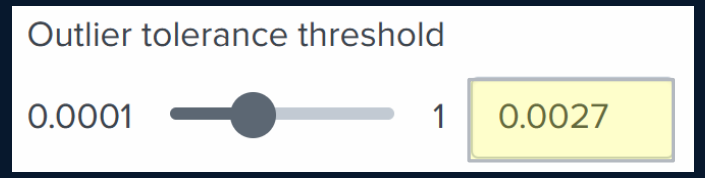












Outliers defined as more than 1 standard deviation from the average.

Let's set the threshold to identify the rarest ~32% of the observations

Outliers defined as more than 2 standard deviations from the average.

Let's set the threshold to identify the rarest ~4.55% of the observations

Outliers defined as more than 3 standard deviations from the average.

Let's set the threshold to identify the rarest .27% of the observations