Red Hat Summit

Connect

Event Driven Ansible: the next level of automation

Come utilizzare eventi per eseguire processi di automazione e self-remediation

Gianni Salinetti

Senior Account Solution
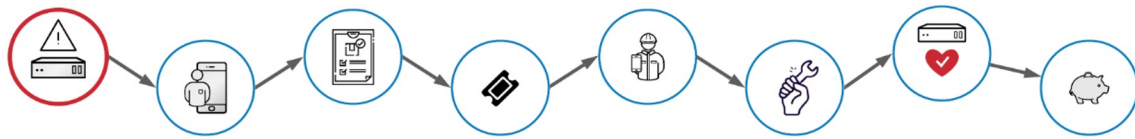Architect
Red Hat

Alessandro Rossi

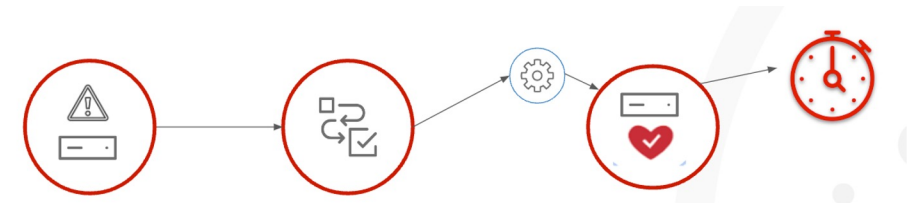Senior Specialist Solution
Architect
Red Hat

Valentino Uberti

Specialist Solution Architect
Red Hat

A brief history of IT Operations

The Old Way: Toil and Churn

The new way: Event-Driven Automated Ops

Outage resolution:
Follow a people-intensive multi-step manual process including opening tickets and multiple handoffs.

▶

Event-Driven outage resolution:
Receive event, matching to rule, respond and act automatically

Security risk resolution:
Monitor to identify risk, notify and open a ticket, manually apply a patch or manually initiate automation job.

▶

Event-Driven security risk resolution: Receive risk event, match to rule, automatically apply patch to impacted inventory.

Red Hat

Introducing Event-Driven Ansible

Achieve new milestones in IT service efficiency

Implement event-driven automation within and across multiple IT use cases

Automate decision making

Leverage numerous sources of events

Achieve new milestones in efficiency, service delivery excellence and cost savings

One subscription. One integrated platform.

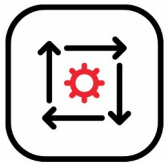**Automation controller**
Automation control plane

Automation execution environments
Scalable packaging and runtime execution plane

Automation mesh
Connectivity across diverse enterprise automation environments

NEW

Event-Driven Ansible
Automatic response to environment changes based on environment intelligence

Ansible-builder
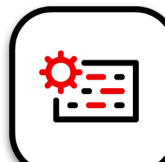Ansible containerized execution environment builder

Automation analytics & Red Hat Insights
Visibility, predictive analytics, and more

Ansible Content Collections
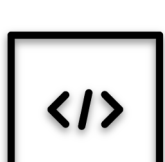100+ certified content collections

Automation hub
Hosted certified content repository.

Ansible-navigator
Execution environment orchestration tooling

Ansible Platform Operator
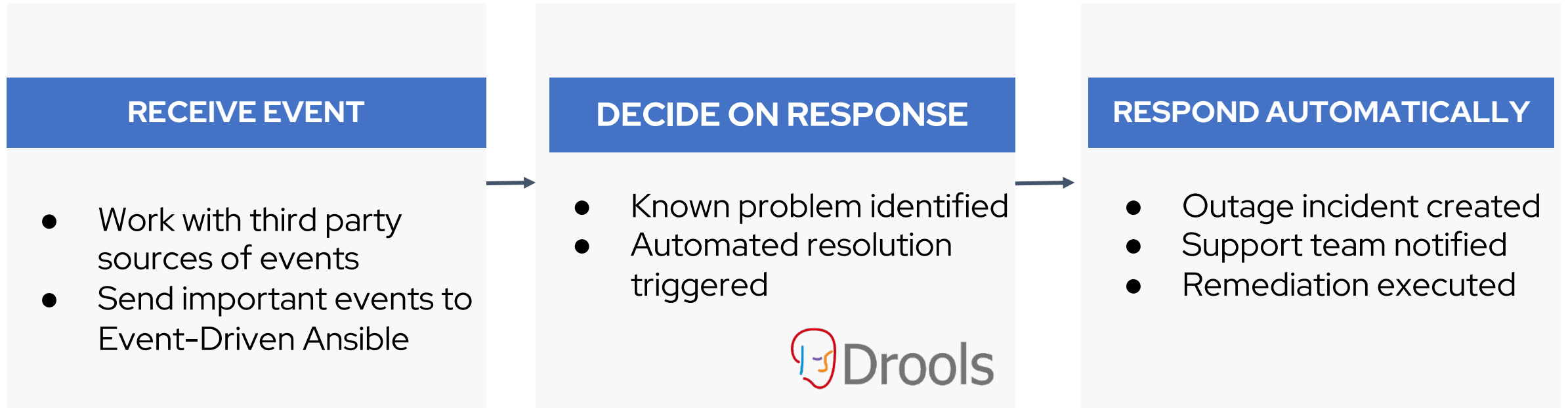Package, deploy and manage this platform on Red Hat OpenShift

Microsoft VS code plugin
Write and manage Ansible code with Visual Studio
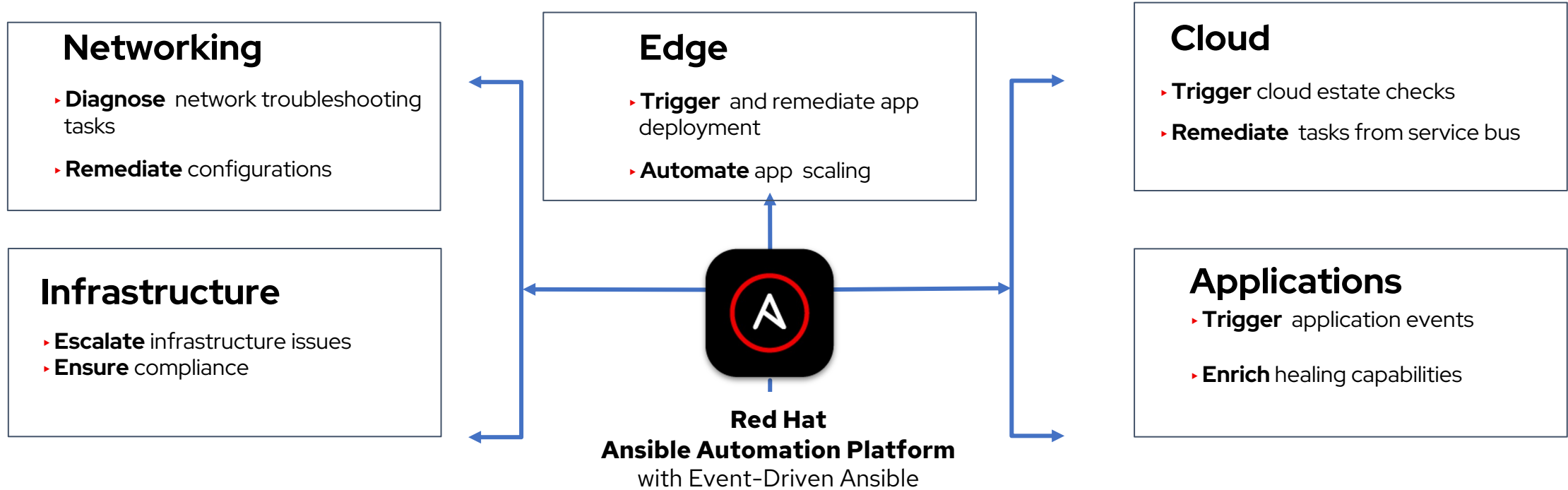
**Red Hat
Ansible Automation
Platform**

Red Hat

Event Driven Ansible

A typical event driven automation process

RECEIVE EVENT

- Work with third party sources of events
- Send important events to Event-Driven Ansible

DECIDE ON RESPONSE

- Known problem identified
- Automated resolution triggered

Drools

RESPOND AUTOMATICALLY

- Outage incident created
- Support team notified
- Remediation executed

WORK ACROSS MULTI-VENDOR IT OPERATIONS

Work flexibly and well with multi-vendor monitoring and other solutions across the event driven architecture with appropriate approvals, controls and awareness

Red Hat

Event-Driven Ansible is use case-friendly.

Networking
- **Diagnose** network troubleshooting tasks
- **Remediate** configurations

Edge
- **Trigger** and remediate app deployment
- **Automate** app scaling

Cloud
- **Trigger** cloud estate checks
- **Remediate** tasks from service bus

Infrastructure
- **Escalate** infrastructure issues
- **Ensure** compliance

Applications
- **Trigger** application events
- **Enrich** healing capabilities

**Red Hat
Ansible Automation Platform**
with Event-Driven Ansible

Security
- **Automate** log enrichment / response
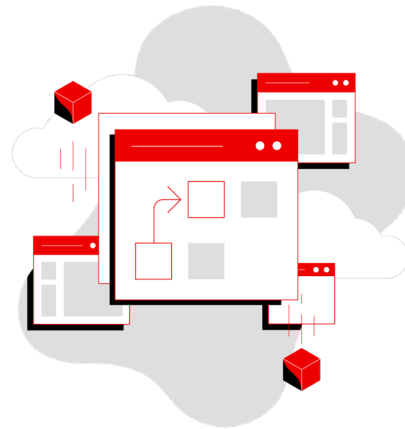- **Escalate** events

Key building blocks in Event-Driven Ansible

Simple, powerful, agentless
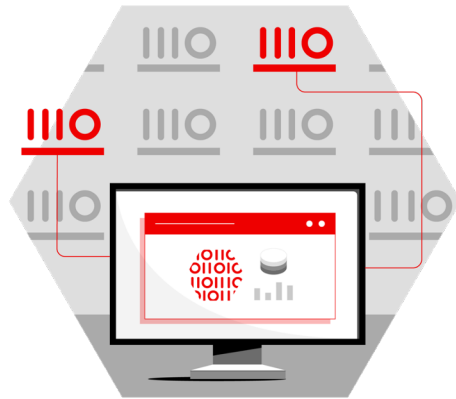
Sources

All the sources of event data you want to use

Rules

What you will create using Event-Driven Ansible®

Actions

When a condition or event is met, the Ansible Rulebook executes

Ansible Rulebooks contain the source of the event, as well as the instructions on what steps to perform when a certain condition is met–and it is all very flexible.

A broad choice of Source plugins.

**PARTNER SOURCE PLUGINS
@ LAUNCH**

- Arista (validated)
- Dynatrace
- F5
- Instana
- Red Hat Insights
- Turbonomic
- Zabbix

**CERTIFIED COMMUNITY
PLUGINS @ LAUNCH**

- AWS SQS
- Azure Service Bus
- GCP Pub/Sub
- Kafka (AMQ Streams)
- Kubernetes (Openshift)
- Prometheus/Alertmanager
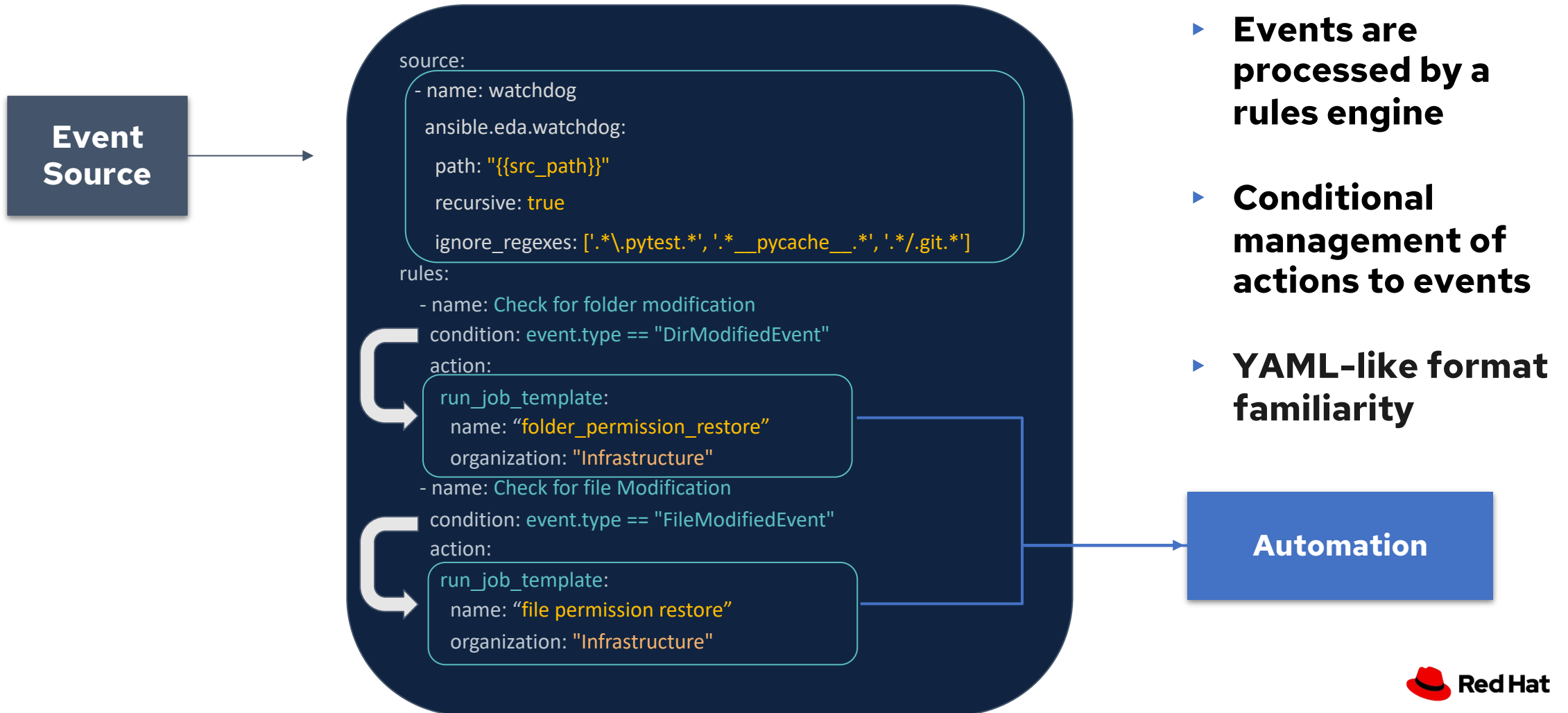- Webhooks

**COMMUNITY PLUGINS @
LAUNCH**

- watchdog (file system watcher)
- url_check (url status check)
- range (event generation plugin)
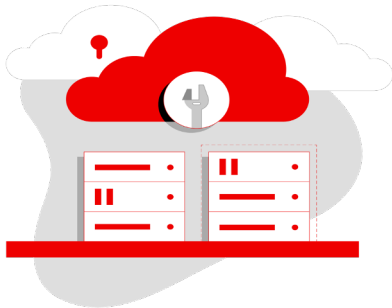- file (loading facts from yaml)

**ROADMAP FOR
INTEGRATIONS**

- AppDynamics/FSO
- Cyberark
- Palo Alto Networks
- Sensu
- Splunk
- ThousandEyes

Red Hat

Anatomy of an Ansible Rulebook

Smart automation from conditional rules

Event Source

```
source:
 - name: watchdog
   ansible.eda.watchdog:
     path: "{{src_path}}"
     recursive: true
     ignore_regexes: ['.*\.pytest.*', '.*__pycache__.*', '.*/.git.*']
rules:
 - name: Check for folder modification
   condition: event.type == "DirModifiedEvent"
   action:
     run_job_template:
       name: "folder_permission_restore"
       organization: "Infrastructure"
 - name: Check for file Modification
   condition: event.type == "FileModifiedEvent"
   action:
     run_job_template:
       name: "file permission restore"
       organization: "Infrastructure"
```

Automation

- ▸ **Events are processed by a rules engine**

- ▸ **Conditional management of actions to events**

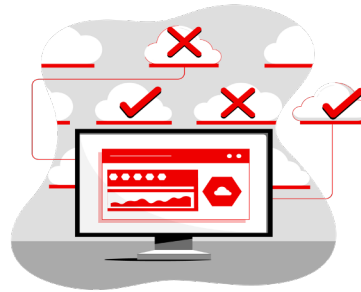- ▸ **YAML-like format familiarity**

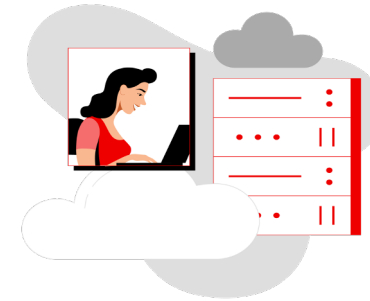# Suggested use cases for getting started

Service ticket enhancement

Automate fact gathering

Network administration

Edge device management

Remediation

Drift

Slow performance
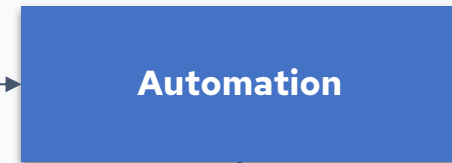
Outages

User management

User authentication and access

Login issues

Group and role access

Use Case:
Infrastructure Remediation

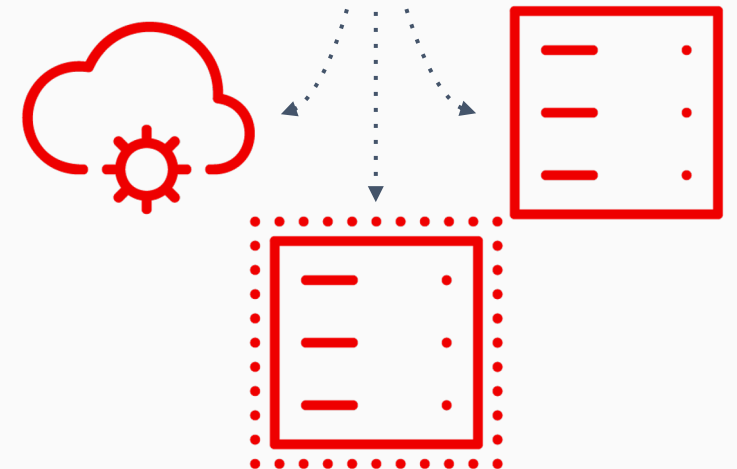# Event-Driven Ansible and Infrastructure remediation

EDA

Event Source

Decision Making and Workflows

Automation

▸ **Listen for event notifications**
  ▹ Proactive advisories coming from Red Hat Insights
  ▹ Webhooks triggered by heterogeneous platforms

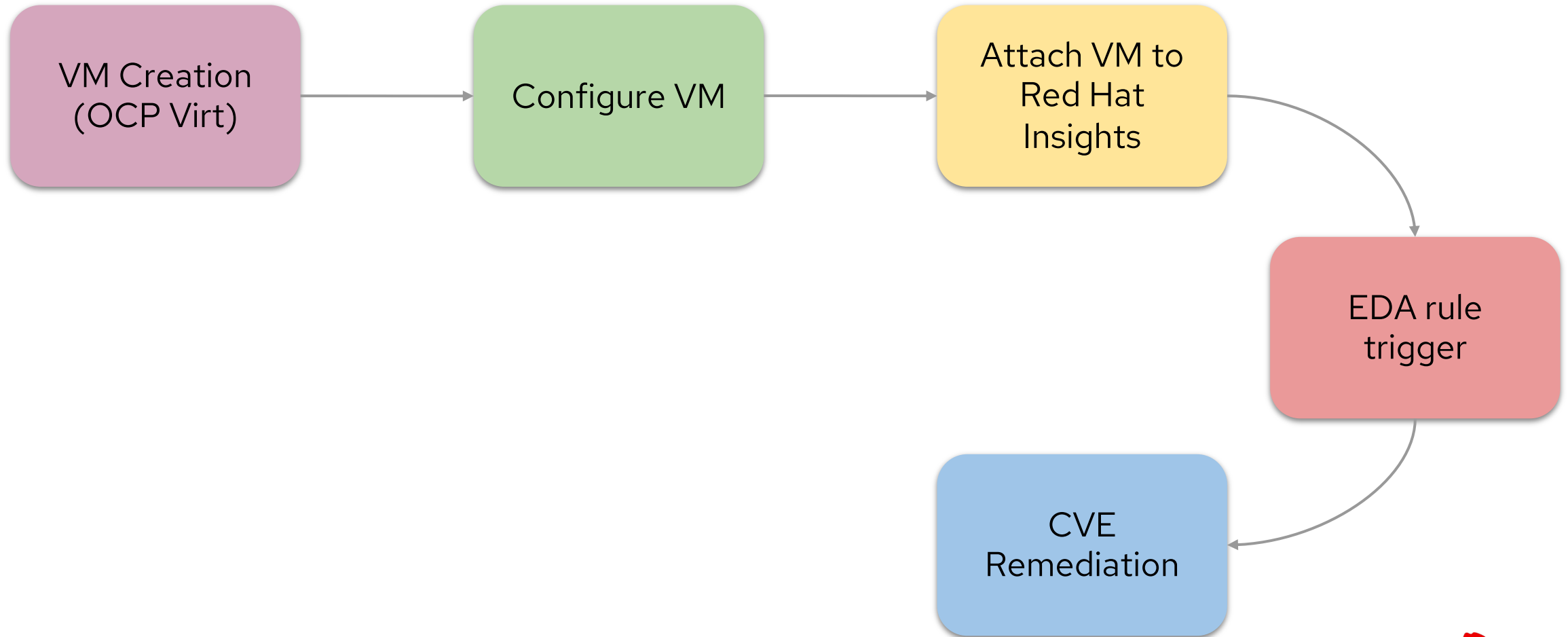▸ **React promptly**
  ▹ Perform remediations following best practices
  ▹ Streamline provisioning and deployment

Demo Workflow
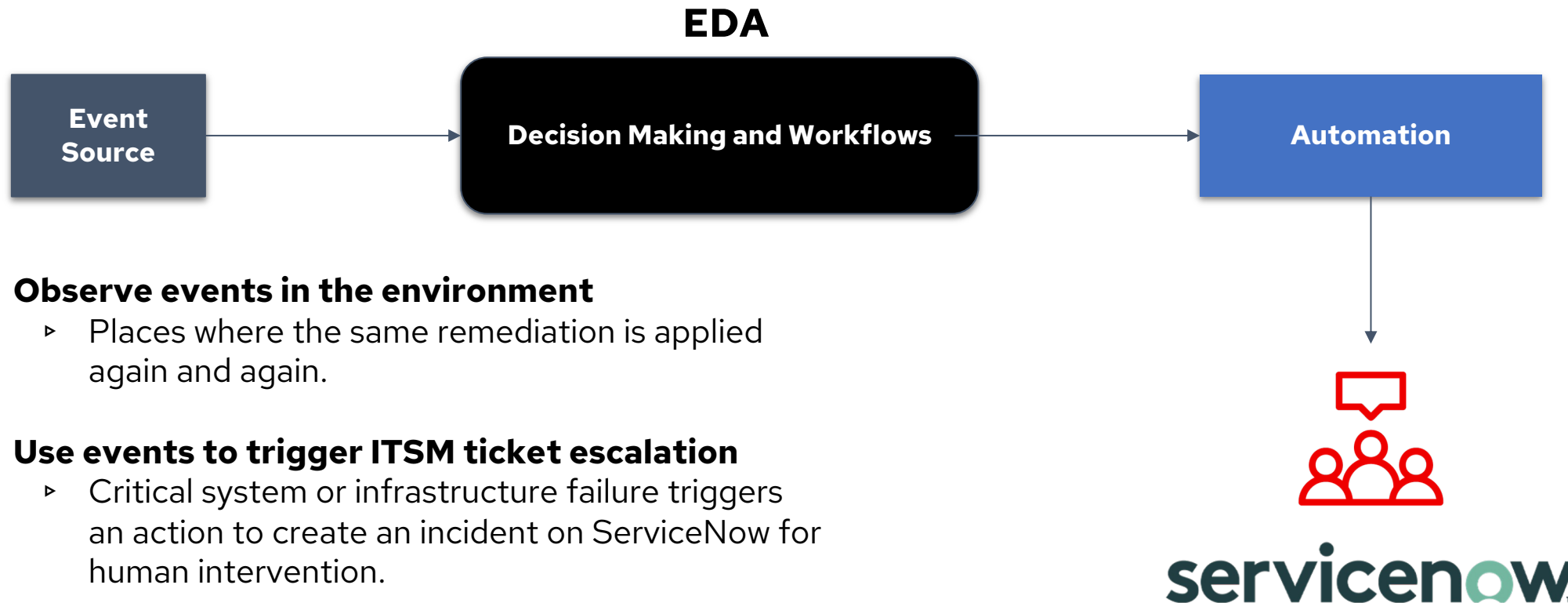
Infrastructure Remediation

Use Case:
ITSM Integration

Event-Driven Ansible and ServiceNow ITSM integration

EDA

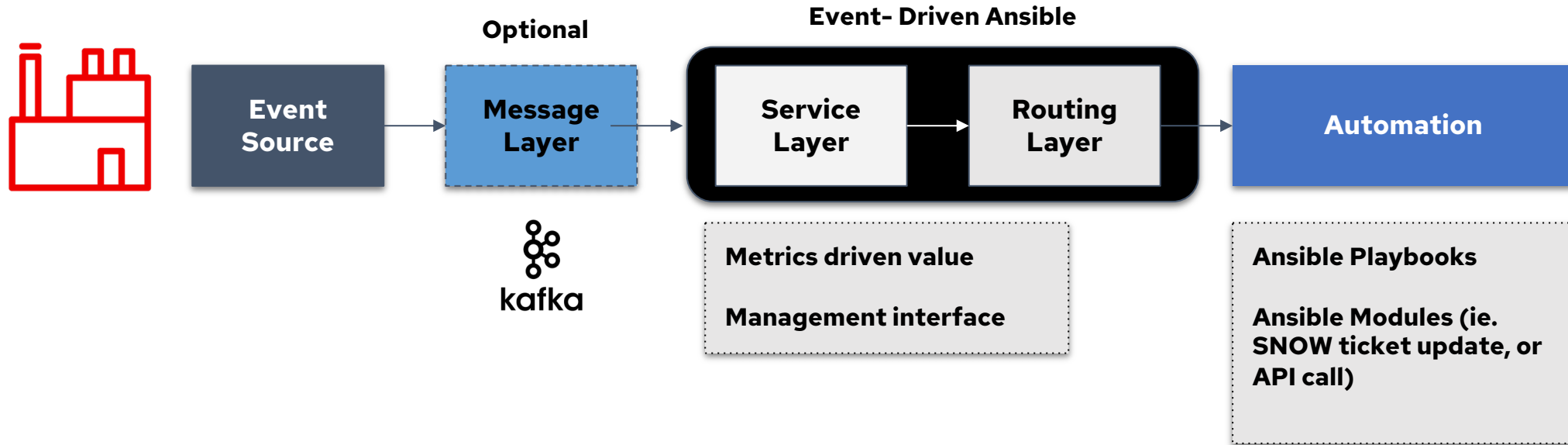| Event Source	→	Decision Making and Workflows	→	Automation

- **Observe events in the environment**
  - ▹ Places where the same remediation is applied again and again.

- **Use events to trigger ITSM ticket escalation**
  - ▹ Critical system or infrastructure failure triggers an action to create an incident on ServiceNow for human intervention.

- **Update ServiceNOW CMDB**
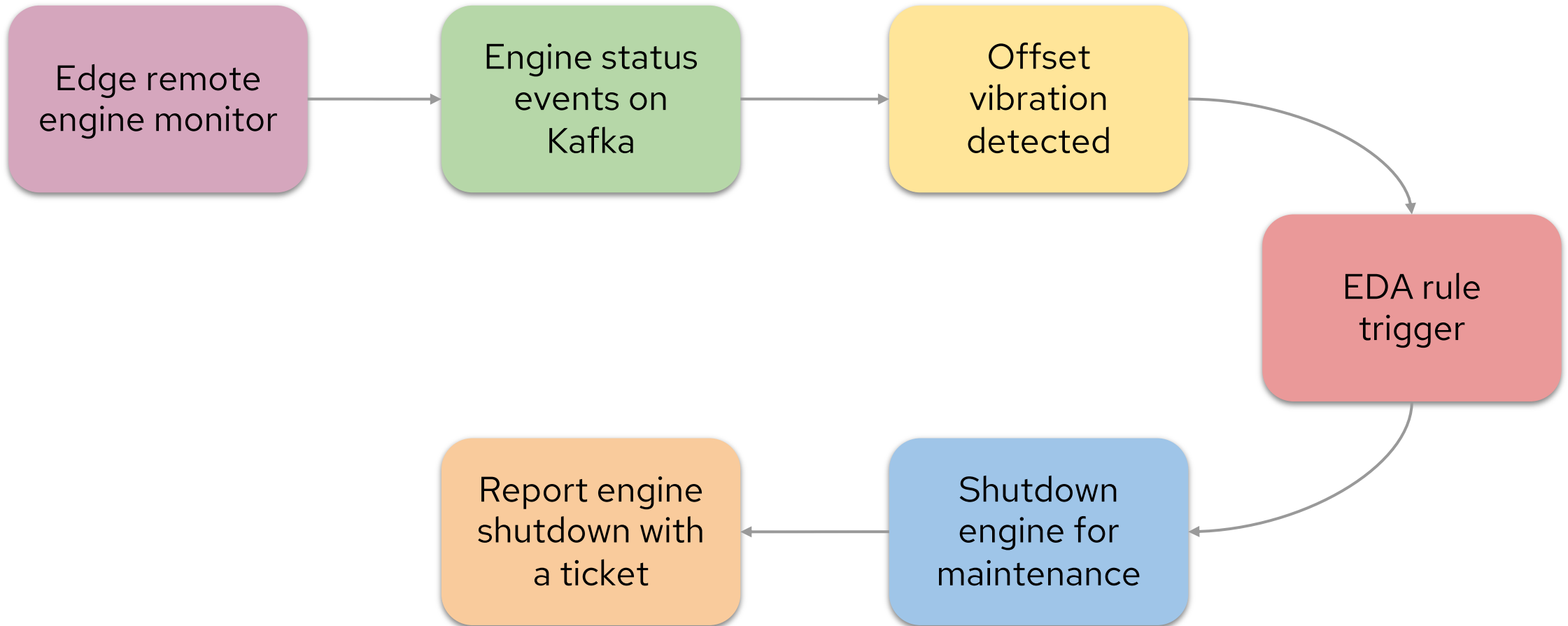 - ▹ Infrastructure changes can be observed and used to trigger ServiceNow to update its inventory

servicenow

Red Hat
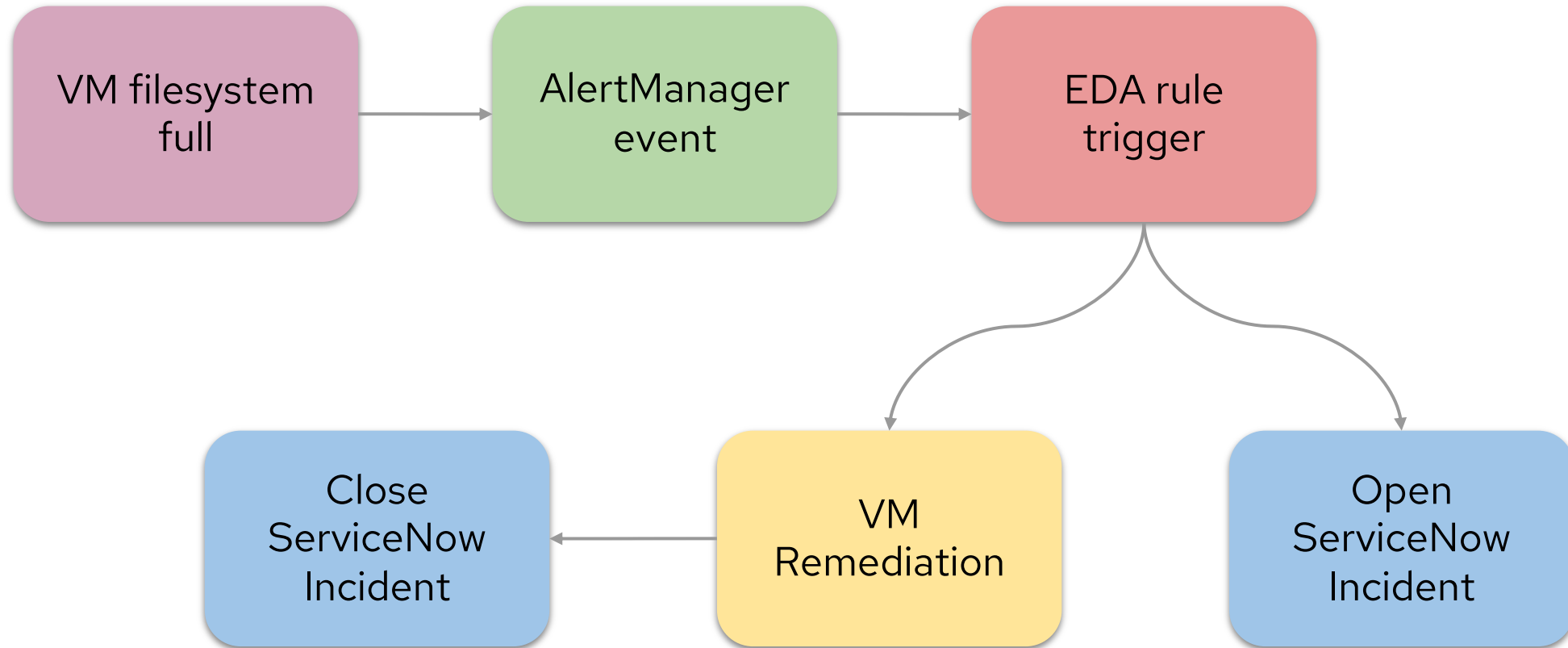
Demo Workflow

Use Case:
Edge monitoring and remediation

Execution layers of event driven automation

Optional

Event- Driven Ansible

Event Source → **Message Layer** → [**Service Layer** → **Routing Layer**] → **Automation**

kafka

Metrics driven value

Management interface

Ansible Playbooks

Ansible Modules (ie. SNOW ticket update, or API call)

Sources   **Rules**   **Actions**

Red Hat

Demo Workflow

Try it yourself!

All the materials in this session are available in the following Github repository:

https://github.com/redhat-italy/redhat-sc23-eda

Everybody is welcome to use, fork and suggest improvements.

Global Learning Resource: https://www.ansible.com/resources

Drools   Mario Fusco

Senior Principal Software Engineer @Red Hat

mfusco@redhat.com

Red Hat