LF NETWORKING

LFN Developer & Testing Forum

Open Policy Agent Service Assurance in Telecom Edge

Presented by

Sriram Rupanagunta, Aarna Networks

Jacob Philip, Aarna Networks

Sundar Nadathur, Intel

Agenda

- Policy Introduction
- Open Policy Agent
- Temporal Workflow
- EMCO Temporal Integration
- EMCO Policy Controller

Policy Introduction

- Policies are declarative business rules
- Policy is a set of rules that governs behaviour of software services
- Terminologies defined in RFC 2904/2753/3198:
  - Policy Decision Point (PDP) - Evaluates request against rules
  - Policy Enforcement Point (PEP) - Acts on policy decision
  - Policy Retrieval Point (PRP) - Policy Storage
  - Policy Administration Point (PAP) - Manages Policies

Importance of policy in Telecom Edge

Policy play an important role in Telecommunication in many functionalities like

- Closed loop Automation
- Achieving QoS and SLA
- Automation based on data analytics
- Policy driven orchestration of Network Functions & MEC applications
- Optimized usage of cloud & edge infrastructure
- Security of Core & Edge clusters

Policy Decision Point (Policy Engine)

- Policy Engine or PDP is core of any policy system
- Policy Engines like ONAP Apex, JBoss Drools, OPA, etc are evaluated during this project
- Following points are considered for evaluation
    - Simplicity
    - Fit for different use case like edge orchestration, service assurance, edge cluster security
    - Memory Footprint and Performance
    - Support & Community Adoption

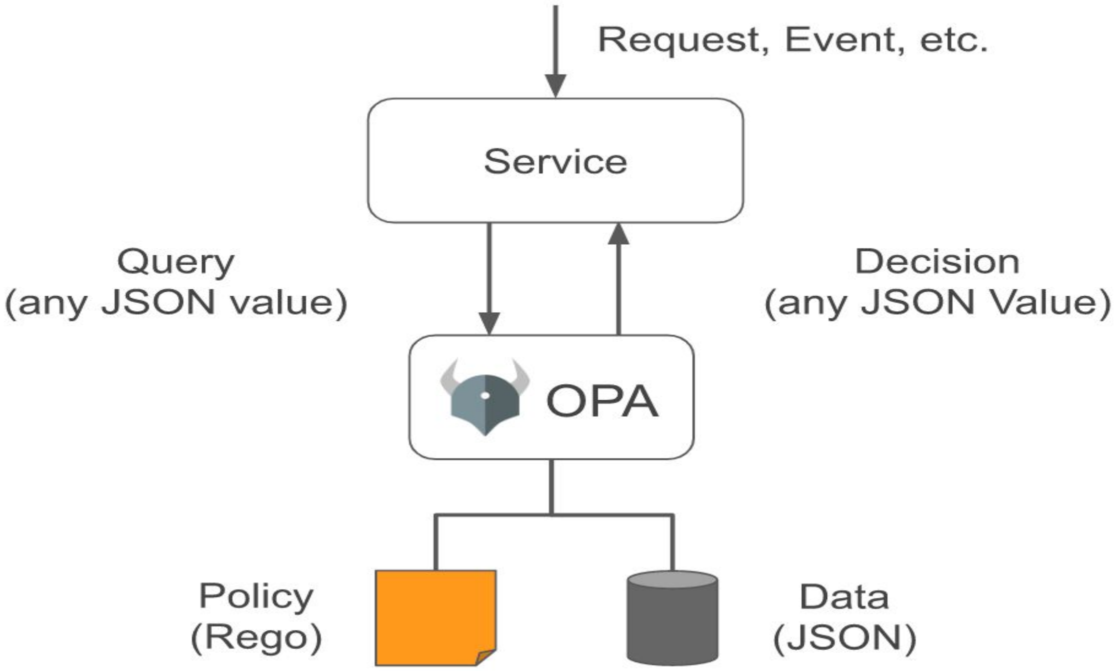
Open Policy Agent: Introduction

- CNCF Graduate Project
- Cloud Native
- De facto policy engine for Kubernetes environments
- Supported by Styra, Netflix, Microsoft, Google etc
- Can expect good community support
- High Performance and low memory footprint

Open Policy Agent: Introduction (Cont..)

- Implemented in Go
- Policy language, Rego, is a Go like declarative language inspired by Datalog
- Plugins for Kubernetes, Istio, Terraform, SQL, SSH, Linux etc
- Very good documentation

Open Policy Agent: Overview



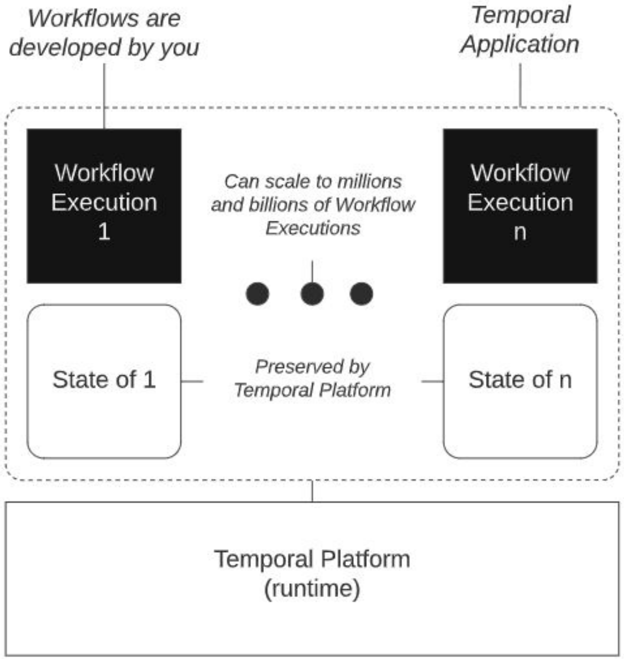
Temporal: Introduction

- Temporal is a scalable and reliable runtime for ”workflow executions”.
  - Distributed processes that can interact via messages.
  - Multi-step, stateful, long-running.
- Temporal is:
  - A workflow engine
  - A framework to develop and deploy distributed apps
  - A framework to implement microservices, with reliability and observability.

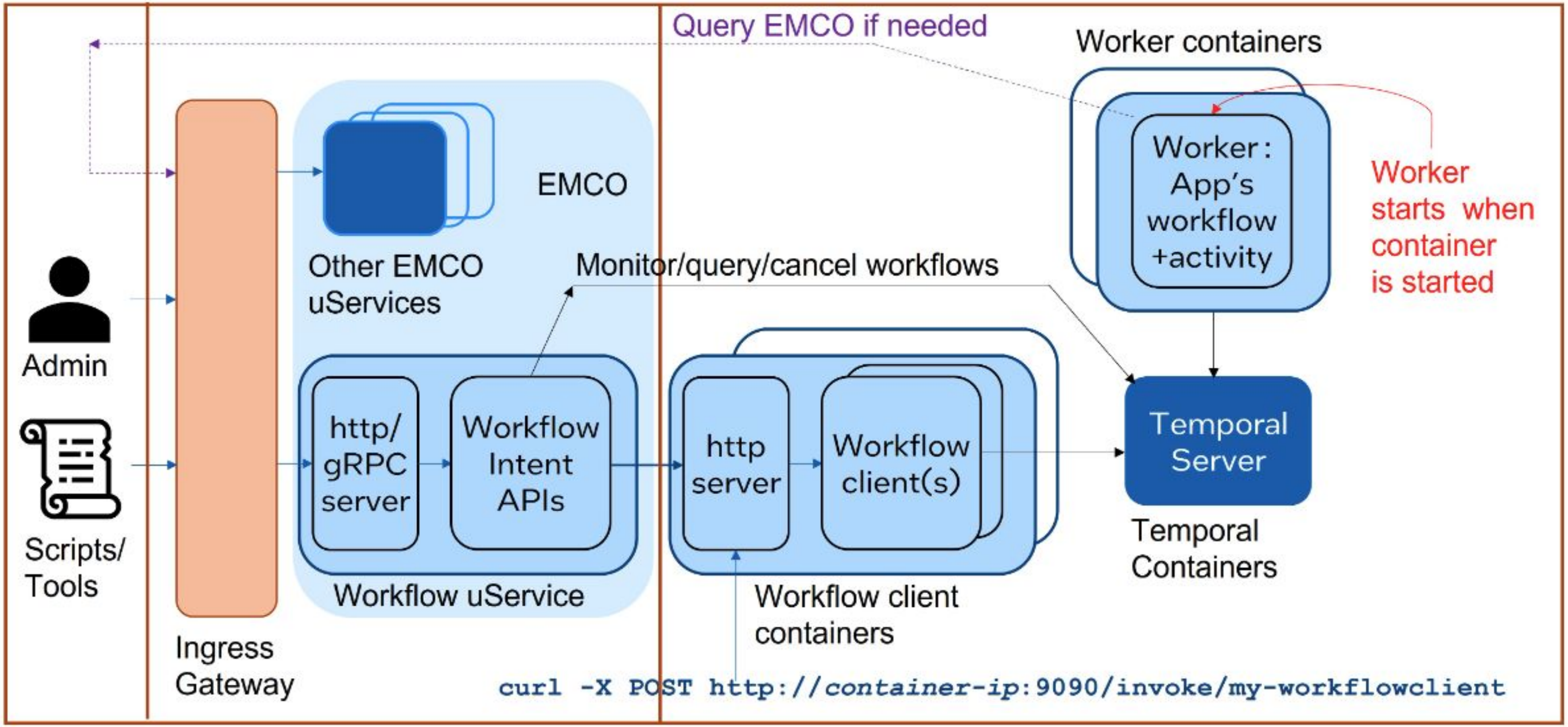
Temporal: Introduction (Cont..)

- Temporal server and workflows/ activities communicate via task queues.
- User provided workflow client, for managing workflow execution

Temporal: Overview


Workflows are developed by you
Temporal Application
Workflow Execution 1
Can scale to millions and billions of Workflow Executions
Workflow Execution n
State of 1
Preserved by Temporal Platform
State of n
Temporal Platform (runtime)

EMCO Temporal integration



```
curl -X POST http://container-ip:9090/invoke/my-workflowclient
```

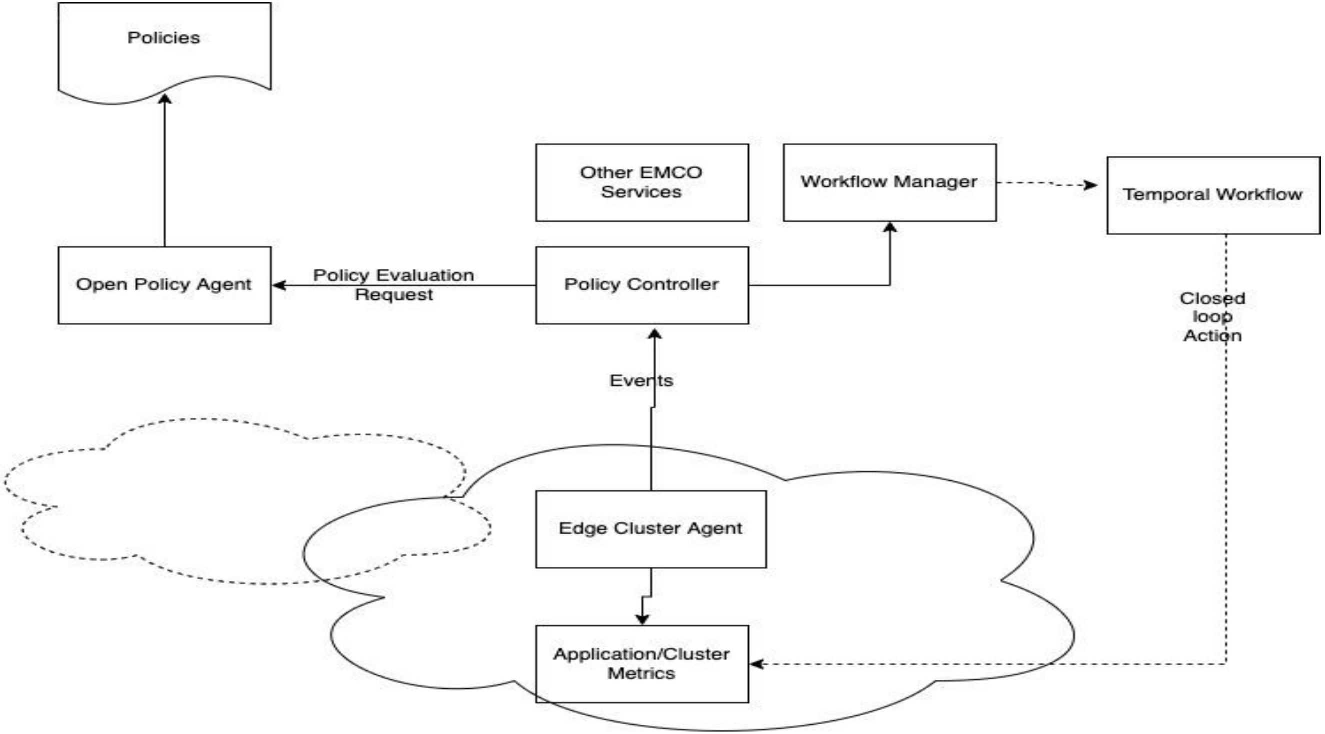
EMCO Policy Controller

- A new microservice in EMCO which act as a PEP (Policy Enforcement Point)
- Uses OPA as policy engine.
- Agent Microservices runs every edge cluster which collects metrics/events from different resources and pass to Policy Controller
- Agent adds additional information like emco clusterid, composite app name etc, to the metrics/events, that policy can use.
- Agents will consume KPIs from k8s Custom Resources
- Policy intents establish relation between Policy, Events(metrics) and Applications.
- Policy Controller doesn’t manage policy documents
- Plan to provide different actor plugin like Temporal workflow, CDS etc.

Policy Controller - Temporal Workflow Integration

- Plugin for converting policy evaluation result as an trigger for starting workflow.
- Actions, if required, will be a call to workflow manager with necessary arguments.
- Both policy intent and workflow intent should be part of same Deployment Intent Group. This allows same policy & workflow to be applied to any relevant composite application

Policy Controller - OPA- Temporal: Closed Loop Flow



Policy Intent

- Policy intents creates a relationship between DIG, Policy, Event and Actor
- Policy Controller sends policy evaluation request to OPA, for each events based on the Intents
- Key Parameters that defines a policy intent
  - **Policy** - Provide OPA endpoint and policy name
  - **Actor** - Actor details (Only Workflow in initial release)
  - **ActorArg** - Arguments required for calling Actor
  - **Event** - Event (From cluster Agents) related to this policy
  - **Supporting Events** (Optional) If Policy is dependent on multiple events/Metics, a list of such events can be provided
  - **Project**
  - **CompositeApp**
  - **CompositeAppVersion**
  - **DeploymentIntentGroup**
  - **PolicyIntentID**

Identifies Policy Intent

Example Use cases

1. Scaleout a composite application to another edge when load on container/cluster crosses a threshold
2. Policy driven edge relocation
3. Analytics and policy driven orchestration of edge systems
4. Event driven orchestration of edge system

Summary

- EMCO Policy controller, a new microservice in EMCO, provides a convenient way to enforce policies in edge clusters
- Customizable design, with plugin model for Events and Actions
- Create closed loop on edge infrastructure with developing only policies and KPI adaptors
- Initial release with Temporal workflow plugin (Integrated with EMCO Workflow Manager)
- Future releases will support ONAP CDS

LF NETWORKING

LFN Developer & Testing Forum

THANK YOU

Sriram Rupanagunta

srupanagunta@aarnanetworks.com

http://www.aarnanetworks.com