Model driven Control Loop Design

Overview

This functional requirement is intended to further advance the ease and ability of creating control loops in Dublin by addressing the following:

1) Utilize the TOSCA-Lab and DCAE-DS applications that were ingested by SDC during Casablanca Release.

2) Make it easier for the community to design and on-board DCAE applications based on re-usable policy models.

3) Enhance the Policy Lifecycle API and newly created Policy SDC Service Distribution component to dynamically make DCAE Services Policy Models available to CLAMP when Service Designers are creating Control Loops.

Problem Statement

In Casablanca, DCAE services were manually on-boarded and the deployment artifacts were Cloudify Blueprints that were created manually. In parallel, SDC did the work to ingest the TOSCA-Lab and the DCAE-DS (Design Studio) applications that were built to support the automation of both on-boarding DCAE services and creation of Cloudify Blueprints.

In Casablanca, CLAMP currently assumes only the DCAE TCA analytic component when designing Control Loops. While Policy supports multiple DCAE services policy models, the only way to on-board them is via the Policy GUI because there is no API in Policy to do this. Policy Framework also has no API for CLAMP (or other applications) to use to discover these models. In addition, the current DCAE service policy models do not quite conform to TOSCA specifications (minor fix).

Business Requirement

Utilize newly introduced Casablanca applications (TOSCA-Lab, DCAE-DS, Policy SDC Service Distribution) and build the Policy Lifecycle API to support auto-ingestion of re-usable policy models and discoverability of these models by CLAMP during Control Loop creation.

This work should evolve the ONAP Platform to make it easier to design and re-use any desired Control Loops in any ONAP service.

- Reduce development cycle when introducing new analytics to Control Loop.
- Reduce development cycle when introducing new Policy Models as part of Control Loop

Participating Companies

AT&T, Nokia

Assumptions

Policy - Will retain our current API and GUI for uploading DCAE micro service models, creation of policies for those models, etc.

DCAE - Deployment/integration of standalone services (non-control loop related) service will follow existing process followed in R3

CLAMP - Will move directly to the new Policy Lifecycle API and not retain backward compatibility to the R3 Policy API.

SDC - TBD

Dublin Goals

In Dublin we will describe two sets of goals:

Contributions

Ease of creating analytic components and on-boarding DCAE micro services

Paris Developer Event Control Loop meeting:

- 2019-01-09 Control Loop Sub Committee Weekly Meeting - Paris F2F
- The final PPT: Model_Driven_CL_Design_Dublin_v3.pptx

Example of TCA Model that would be generated by TOSCA-LAB - export_policies-v1806.yml

CL design R4.docx - Marco Platania

Impacts

The Closed Loop projects for Dublin will primarily impact CLAMP and Policy. It will also have impacts on DCAE and SDC.

Please see the Impacts Page for a more detailed list of requirements on each project.

Project Commitments

Model Driven Control Loop - Status

COMMITTED

Project	PTL	Commitment	Notes
CLAMP	Gervais-Martial Ngueko	COMMITTED	Goal #7 is a stretch goal but is not required for Dublin
DCAE	Vijay Venkatesh Kumar	COMMITTED	Based on AT&T commitment to find the resources
Policy	Pamela Dragosh	COMMITTED	

Self Serve Control Loop - Status STRETCH GOAL

This project will only be completed in Dublin if additional resources are assigned to it early in the release cycle

Project	PTL	Commitment	Notes
DCAE	Vijay Venkatesh Kumar	STRETCH GOAL	
Policy	Pamela Dragosh	STRETCH GOAL	Descoping Goal 16
SDC	Ofir Sonsino	STRETCH GOAL	

- The Model Driven Control Loop will outline the ability to create new control loops and new types of control loops without the need for development cycles
- The Self Serve Control Loop will create the ability for service designers to attach control loops and DCAE service components at service design time using SDC

In Dublin only the Model Driven Control Loop is being committed, but we will attempt to do as much of Self Serve Control Loop as possible based on the availability of resources.

⚡ **CLSC-6** - Model Driven Control Loop Design `CLOSED`

Model Driven Control Loop

1) The specification for the Policy Model for DCAE service component is enhanced to conform to TOSCA standards.

2) DCAE service component developers will manually create and then on-board their Policy Models into Policy. (No Development)

3) DCAE service component developers will manually create the Cloudify Blueprint (includes the list of Model ID per microservice?

Onboarding steps for DCAE MS through SDC/Policy/CLAMP (Dublin)

4) DCAE service component developers will load the blueprint into SDC manually. (Current Casablanca process) (No Development)

5) CLAMP will be modified to automatically render the UI for the DCAE service component Policy Models to configure the specific parameters for that service component in a specific Control Loop.

6) CLAMP will be able call the Policy Lifecycle API to find the Policy Model for the Analytic and create a concrete Policy for the analytic for the Control Loop.

7) STRETCH GOAL: Additionally, with the use of the new Policy Lifecycle API CLAMP will also be able to call the Policy Llifecycle API to query for available Operational Policy Types, Guard Policy Types, Control Loop Coordination Types, etc. for the Designer to choose from as desired.

8) When Control Loops are deployed, CLAMP will be able to call the Policy Lifecycle API to deploy/undeploy all the Policies for the Control Loop. NOTE: CLAMP deploys/undeploys the blueprints for the DCAE Controller at this time - no changes necessary.

9) While the DCAE Controller is deploying DCAE service component instances. The DCAE Policy Handler will call the Policy Lifecycle API to retrieve the policies for the service component.

10) Enhance Policy Lifecycle API to allow automated onboarding of DCAE service component policy models, querying of policy models, creation of concrete policies from these models, deploy/undeploy API for the concrete policies and decision API for DCAE service components to query for a decision on the appropriate configuration policy

11) Create method for automating tests of generic control loops

Self Serve Control Loop

In Dublin, the intention is do the following:

12) The specification for the Policy Model for DCAE service component is enhanced to conform to TOSCA standards. (Duplicate)

13) SDC will make modifications in TOSCA-Lab to conform to the new Policy DCAE service component Model.

14) SDC will need to make changes to the blueprint to generate new cloudify blueprints in the format required by DCAE that will support K8S deployment vs docker deployment (current). ONAP will be eliminating docker deployment and focusing solely on K8S deployment.

15) SDC will need to make changes to the blueprint to reference the Policy DCAE service component Policy Type from #1

16) *(Descoped in Dublin) DCAE service component developers will on-board their service component using a JSON micro service specification in TOSCA-Lab that will auto-generate their own instance of the Policy DCAE service component Model for use in future Control Loops. This step will also result in a simple Cloudify Blueprint generation for DCAE services.*

- *Deferred to El Alto: SDC building an API to enable Integration team to automate the call to load the JSON specification during testing.*

17) Simplifying the onboarding steps for MS not specifically binded to SDC service

18) If necessary, DCAE service component developers can also now use DCAE-DS to create more complex Cloudify Blueprints that can combine more than one service component together. NOTE: DCAE-DS is a consumer of the TOSCA-Lab tool. No development or testing will be done for this requirement via the Control Loop subcommittee, use case owners will be responsible for this functionality if it is a part of their Use Case flow.

19) When a service designer designs a new service, they will add to the service CSAR any DCAE service component Models and blueprints they wish to make available for Control Loops. SDC will distribute the CSAR as it does in Casablanca with these artifacts contained within the CSAR. REPEAT

20) In Casablanca, the Policy SDC Service Distribution application was created. This application will now be utilized to support auto-creation of the newly on-boarded DCAE service component Policy Model when a service is distributed. Thus, when SDC distributes a new CSAR, Policy will look for new DCAE service component Policy Models not current loaded into the framework and utilize the Policy Lifecycle API to ingest these Models. NOTE: Policy will only need to do this the first time they see this service component model version.

TESTING

Integration team