TOSCA based Control Loops with the existing CLAMP code in the Policy Framework

Initial Comments

1. The long term aim is to integrate CLAMP into the Policy Framework and to align the structure and technologies of CLAMP and the Policy Framework in general
    a. Is the Policy Framework structure correct?
    b. Should the Policy Framework shift to a framework?
    c. Module structure of Policy, in SDC an integration test is not possible? Integration test in SDC is not feasible
    d. In CLAMP, everything is in one module so integration test is done as part of the build
    e. Problem with Jacoco, coverage is taken using XML rather than a binary, so currently we can't report the coverage from the integration tests, by having a single module, we can get the coverage out of the integration test.
2. We should have a picture of what the long term vision for CLAMP in the Policy Framework
3. How the TOSCA based Control Loop features are implemented should be in line with this long term vision

Policy Framework and CLAMP

Architecture

Architecturally, the Policy Framework and CLAMP are complimentary as separate systems. The Policy Framework is part of control loops, and CLAMP is a control loop management system.

Technologies

No.	Policy Framework	CLAMP	Recommendation	Comment
~~POLICY-3209~~ - CLAMP Component Lifecycle Management using Spring Framework CLOSED	Policy Common	Spring Framework	Spring for new (All participants including DCAE /K8S) ~~POLICY-3168~~ - DCAE interaction for Helm service deployment CLOSED Migrate if doing something else in existing PF code in master (Spring in policy-common?)	
~~POLICY-3210~~ - REST Endpoint Implementation CLOSED	Policy Common, using JAX-RS annotations	Camel	Camel the Commissining /Instantiation Spring for Supervision /Monitoring Use Camel where we need flexibility.	
~~POLICY-3211~~ - Parameter Handling and Parameter Validation CLOSED	Built in parameter validation in policy common	Spring properties	Let's investigate if the policy-common parameter handling can be got to work in Spring (javax validation)	
~~POLICY-3212~~ - Common TOSCA Handling CLOSED	Policy Models, integrated serialization and persistence for most TOSCA entities	CLAMP TOSCA handling (more info)		Separate study ongoing in the Policy Framework on this We should try and get this framework on Spring, which would enable further merging
~~POLICY-3213~~ - Persistence Policy Models using JPA/JDBC/Eclipselink /MariaDB CLOSED	Policy Models using JPA/JDBC/Eclipselink /MariaDB	Spring using JPA /JDBC /Hibernate /MariaDB		To be investigated. Should also consider using the policy DB to store TOSCA rather than caching it in a separate CLAMP-specific DB
~~POLICY-3214~~ - Change Monitoring UI implementation to use React CLOSED	None (Angular in TOSCA PoC, APEX policy editor)	React	React	Angular (Security issues raised), new version did not solve the issues. React is flexible and easier to understand, we moved in an earlier release from Angular to React. Used Jsoneditor (library), easier with React. Develop the Monitoring GUI as a new tab in the CLAMP UI.

Code Structure, Build, and Test

	Policy Framework	CLAMP	Recommendation	Comment
~~POLICY-3215~~ - Update CLAMP Module structure to Multi Module Maven approach CLOSED	Maven multi module project	Single module project, builds everything	Multi Module	Price to pay is that we could have some issues with getting integration coverage
~~POLICY-3216~~ - Update Docker Build to handle new components CLOSED	Common approach for current components and repos using a "packages" maven module	Part of Single module		Add TOSCA components to the Docker build, also see if or how we use the Policy Framework approach
~~POLICY-3217~~ - Integration Tests to handle clamp components CLOSED	CSITs done per component, separate to build	Comprehensive Integration test, part of build	The ONAP recommendation is that Integration tests should be a part of the build.	
No Jira	All docs are in policy parent	docs in subdirectory in clamp repo	Move to policy parent	
~~POLICY-3218~~ - Integrate CLAMP GUIs (Instantiation /Monitoring) in the policy-gui repo CLOSED	Separate "policy gui" repo	ui-react subdirectories in clamp repo	Let's think about it.	We should do this
~~POLICY-3219~~ - Integrate Participant Simulator into Policy Framework CLOSED	DMaaP Simulator A&AI, SDNC, CDS, APPC, and others	Emulator for CLAMP external interfaces, TOSCA POC we have a participant simulator		CLAMP should use the real Policy components in the integration tests within the build (stretch goal)

Other Considerations

- CLAMP planned improvements (Can we add?)
- Caching policies in CLAMP vs accessing PF database (Mentioned in the demo video)
- DCAE is evolving, how to work towards the K8S based DCAE

Needs for TOSCA Control Loop

Participant components at run time, docker etc

Add the features from the PoC:

- Metadata and generic handling of definitions and instances of TOSCA control loop models (in policy models)
- Monitoring and supervision of instances of control loops
- Be able to handle arbitrary control loops on the fly (Arbitrary participants, maybe not DCAE or Policy Framework or ONAP controllers)
- Be able to parameterize the TOSCA directly (irrespective of the participant type)
- Participants (Intermediary, simulator, DCAE, Policy, and K8S) as separate executing components

Meeting notes

1. As complexity increases we ill need to move to some sort of multi maven project whilst preserving the power of the current approach.