1 Objective

This experiment aims at evaluating the usefulness of the detection and recommendation support provided by our detection tool to identify REST/OCCI Patterns and Anti-patterns.
Objects and materials:

- A user guide video : show how to use the detection tool.
- TestApi.owl: The file to be uploaded and tested during the experiment. It contains a semantic definition of patterns and anti-patterns as well as a set of REST operations we selected from three REST API designed to manage cloud services: Rackspace, COPAS, OOI
- Excel-RESToperations.xlsx: an excel file contains 7 REST operations from the selected REST APIs.
- Detection tool: is our detection tool deployed in AWS.
- Questionnaire: consists of a set of questions to evaluate the usefulness of the detection and recommendation support provided by our detection tool.

2 Fundamentals

Rest Patterns and Anti-patterns: represent the good and bad practices in the REST APIs regardless of any cloud standard. In this experiment we focus on two REST patterns and two REST anti-patterns:

- **REST Patterns**
  -**Correct use of POST pattern**: POST must be used to create a new resource or to execute an action.
  -**Tidy URLs:** appears when URIs use lower resource naming and does not contain trailing slashes and underscores.

- **REST Anti-Patterns**:
  -**Amorphous URIs anti-pattern**: appears when URIs contains symbols, capital letters, underscores, etc., making them hard to read and employ.
  -**Forgetting Hypermedia anti-pattern**: appears when links (i.e., hrefs and rels) within the resource representations provided by server are absent.

OCCI Patterns and Anti-patterns: represent the good and bad practices in the OCCI RESTful Protocol that is provided by OCCI cloud standard. In this experiment we focus on two OCCI patterns and two OCCI anti-patterns:

- **OCCI Patterns**
  - **Compliant Delete pattern**: HTTP DELETE must be used and only the URI identifying the resource that will be removed must be provided.
  - **Compliant URL pattern**: URL must be either a string or as defined in RFC6570.

- **OCCI Anti-patterns**:
  -Non-Compliant Create anti-pattern: appears when other HTTP method instead of POST or PUT is used, or the resource definition is not complete (for instance, the Category defining a particular resource Kind is missing).
  -Non-Compliant Trigger Action anti-pattern: appears when other HTTP method instead of POST is used, or the action definition is not complete (for instance, the Category defining a particular action is missing).

For more details about OCCI and REST patterns and anti-patterns, you can visit this link.

3 Experiment steps

The following steps are going to show how to use the detection tool:

1. **Step 1:** Browse TestApi.owl (that contains the semantic instantiation of a set of REST API operations from 3 cloud REST management APIs (OOI, COAPS, RackSpace)

2. **Step 2:** Upload this file by clicking on Uploadfile button

3. **Step 3:** Start the detection:

   - ***Regarding REST Patterns:*** In "Check compliance for REST principles" menu, click on *"Indentify RESTPatterns"*.
     See the detection results and answer the questions related to REST Patterns in the questionnaire.
   - ***Regarding REST Anti-Patterns:*** In "Check compliance for REST principles" menu, click on *"Indentify RESTAnti-Patterns"*.
     See the detection results and answer the questions related to REST Anti-Patterns in the questionnaire.
   - ***Regarding OCCI Patterns:*** In "Check compliance for OCCI principles" menu, click on *"Indentify OCCI Patterns"*.
     See the detection results and answer the questions related to OCCI Patterns in the questionnaire.
   - **Regarding OCCI Anti-Patterns:** In "Check compliance for OCCI principles", click on *"Indentify OCCIAnti-Patterns"*.

     See the detection results and answer the questions related to OCCI Anti-Patterns in the questionnaire.