Risk Assessment Pipeline

Table of Contents

[1. Introduction 3](#_Toc43503537)

[1.1 Risk Assessment Pipeline 3](#_Toc43503538)

[1.2 XML Parser 3](#_Toc43503539)

[1.3 Business Preprocessing 3](#_Toc43503540)

[1.4 Business Features 3](#_Toc43503541)

[1.5 Business Enrichment 4](#_Toc43503542)

[1.6 Fraud Detection 4](#_Toc43503543)

[2 Tools, techniques and methodologies 4](#_Toc43503544)

[2.1 Application 4](#_Toc43503545)

[2.2 Database 4](#_Toc43503546)

[2.3 Deployment 4](#_Toc43503547)

[3 Architecture 5](#_Toc43503548)

[3.1 Connectivity 5](#_Toc43503549)

[3.2 Risk Assessment Cluster 6](#_Toc43503550)

[3.3 Data Scientist - Configuration 6](#_Toc43503551)

[3.4 Risk Manager – Configuration 6](#_Toc43503552)

[3.5 Starting the pipeline 7](#_Toc43503553)

[4 Database 8](#_Toc43503554)

[4.1 EndpointRegistry 8](#_Toc43503555)

[4.2 Rules 8](#_Toc43503556)

[4.3 Comparison 8](#_Toc43503557)

[4.4 Scope 8](#_Toc43503558)

[4.5 MachineLearning 8](#_Toc43503559)

[4.6 MachineLearningType 8](#_Toc43503560)

[4.7 ModelLocation 8](#_Toc43503561)

[4.8 ModelLocationType 8](#_Toc43503562)

[5 Deployment Instructions 9](#_Toc43503563)

[5.1 Installation on Google Cloud 9](#_Toc43503564)

[5.2 Run the components 9](#_Toc43503565)

[6 Example flow 10](#_Toc43503566)

[6.1 Line of package 10](#_Toc43503567)

[Step 1: XML Parser – XML to JSON 10](#_Toc43503568)

[Step 2: Risk Assessment Pipeline – Index entities 10](#_Toc43503569)

[Step 3: Business Preprocessing – Preprocess 10](#_Toc43503570)

[Step 4: Business Features – Iterate through dynamic pipeline 11](#_Toc43503571)

[Step 5: Business Enrichment – Enrich 12](#_Toc43503572)

[Step 6: Fraud Detection – Detect fraud 13](#_Toc43503573)

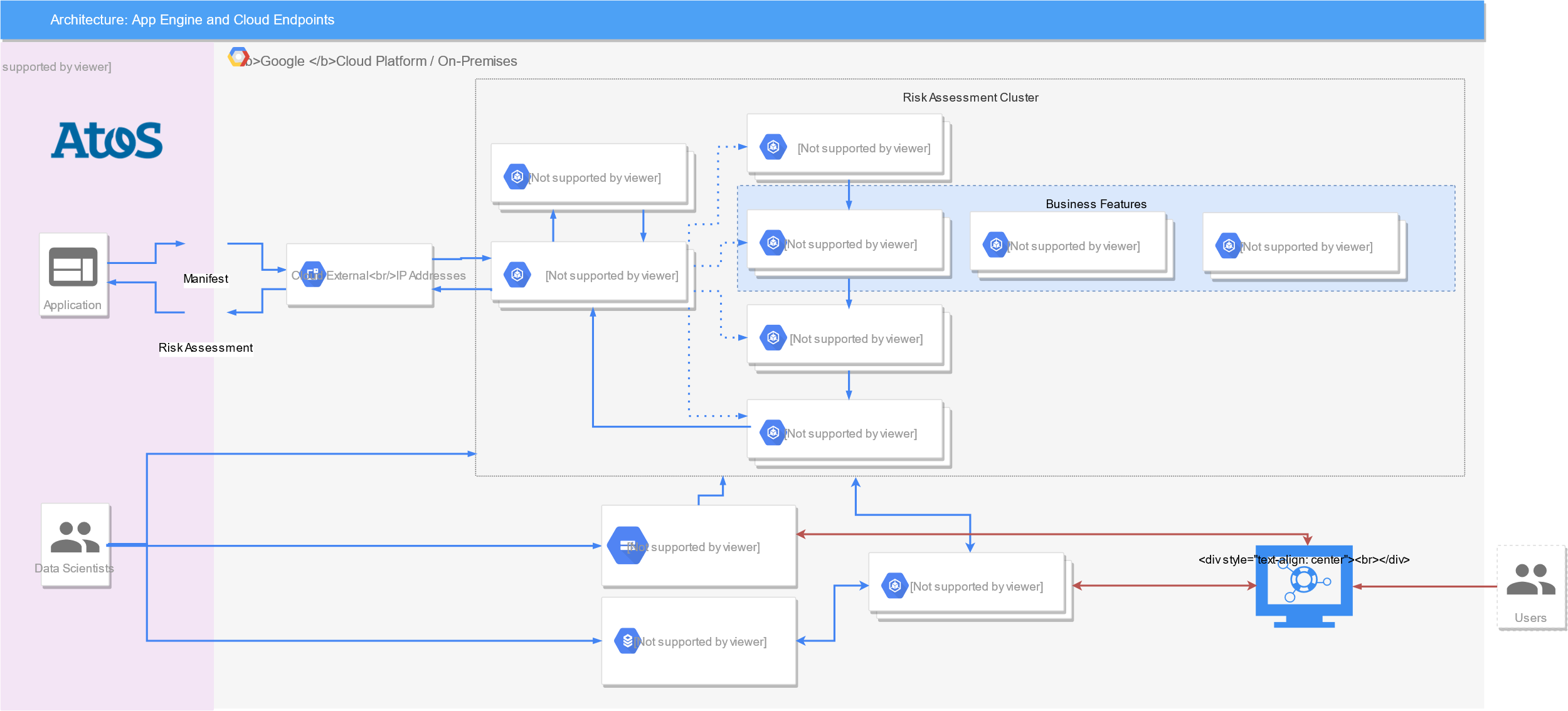
# Introduction

The risk assessment pipeline is created as a pipeline to assess the risk of a manifest. Each manifest that is given to the pipeline, goes through the whole process. As a manifest has several levels that need to be assessed, the pipeline can iterate through the following entities of a manifest:

* Summary Declaration
* Transport Documents
* Line of packages
* … <can be extended>

The risk assessment pipeline is designed as a microservice application that can use a combination of static and dynamic flow. The static flow, which is configurable within each microservice, is described underneath.

The relations between components:

**

## Risk Assessment Pipeline

Handles all microservices. **Minor note: the output for now is JSON instead of XML.**

## XML Parser

The XML Parser is the first step. It parses the Manifest XML into JSON format, which will be used in the pipeline.

## Business Preprocessing

The Business Preprocessing is the second step. It can do some preprocessing on each entity of the manifest. It takes in the JSON structure of a particular entity and will output JSON results.

## Business Features

The Business Features are the third (to <n>) step and can extract multiple features from each entity of the manifest. It takes in the JSON structure of a particular entity and will output JSON results. *The Business Features is a dynamic flow. If there is a new feature that needs to be introduced to the risk assessment pipeline it will be here. E.g. besides tariff classification we can also use cargo scanning. It is also possible to change the order of the dynamic flow.*

## Business Enrichment

The Business Enrichment is the fourth step and is able to do some business enrichment on each entity of the manifest. It takes in the JSON structure of a particular entity, which has been preprocessed and includes the outcomes of the business features and will output JSON results.

## Fraud Detection

The Fraud Detection is the last step and is configurable by the client. It takes in the JSON structure of a particular entity, which has all the information gathered in the previous steps and it will output JSON results.

# Tools, techniques and methodologies

## Application

The application is programmed in Python 3.6 and is implemented as a microservice application. Each component within the risk assessment pipeline is created with Swagger, an open-source software framework backed by a large ecosystem of tools that helps developers design, build, document, and consume RESTful Web services. To create a consistent code base across each of the components, the principle of Model-View-Controller (MVC) is used.

## Database

To create a Single Source of Truth (SSOT), we use a MySQL database, which is implemented in the 5NF, to store all configuration and share the data across the components within the risk assessment pipeline.

As the different types of entities and the entities of the same type do not have a direct relation, the risk assessment pipeline applies multithreading and multiprocessing throughout the components of the risk assessment pipeline.

## Deployment

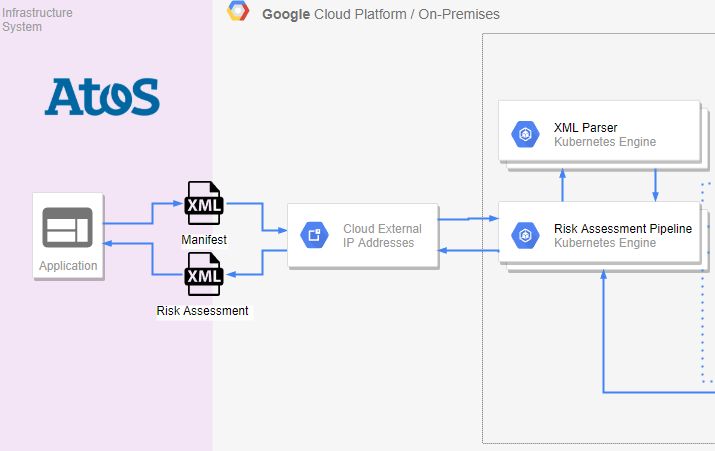
All of the components can be deployed as containers via a *Dockerfile* setup.

# Architecture

The full architecture is described in the first chapter*.* In this chapter we zoom in to several parts of the architecture. We want to emphasize that all components in these architecture are not restricted to cloud and can be also deployed on-premise.

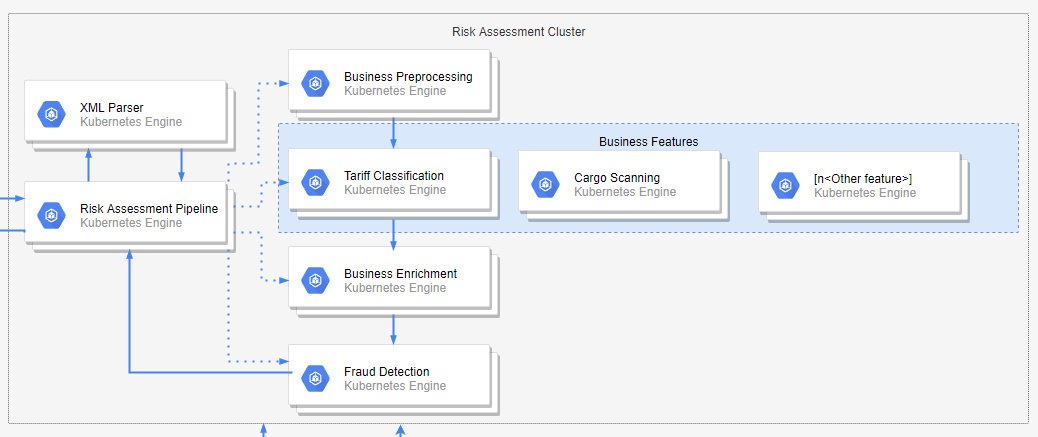
## Connectivity

In *figure 1: Connectivity*, we see that only the Risk Assessment Pipeline endpoint is available from the outside. All the other endpoints can only be reached from within the network.



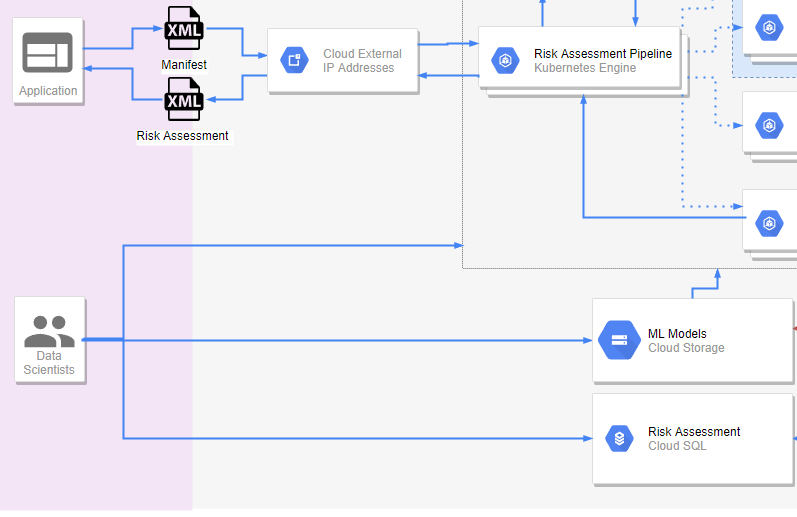
## Risk Assessment Cluster

The following scheme shows the order of the risk assessment pipeline, which is described in the introduction.



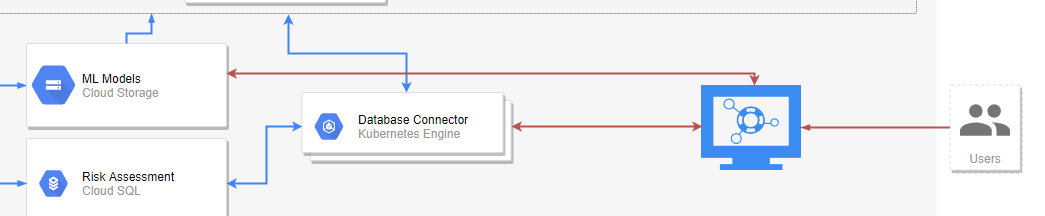
## Data Scientist - Configuration

The data scientists can change the risk assessment pipeline and its components, are able to modify the risk assessment SQL database and can create, update and delete models from the Storage. Possibly with a UI.



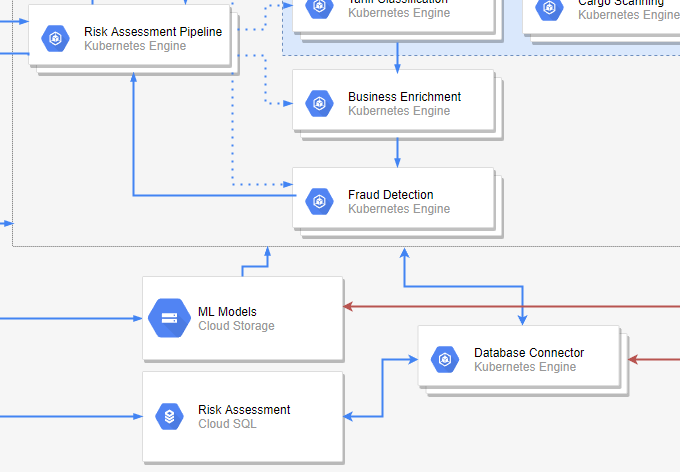
## Risk Manager – Configuration

Via a UI, risk manager is able to adjust a limited number of things in the database, e.g. which Rules and Models they use within the Fraud Detection component. Besides, the risk manager can create and change Machine Learning models configuration in the database.



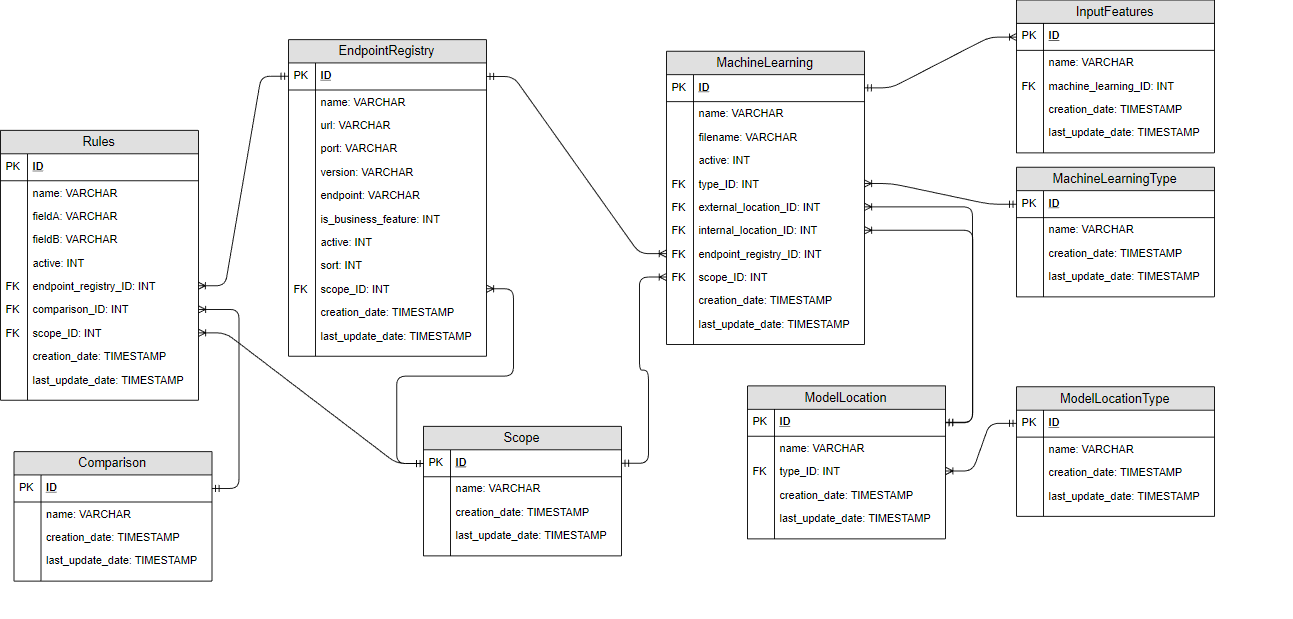
## Starting the pipeline

The key components of the risk assessment pipeline are the database connector and the storage of the machine learning models. These should be available for the risk assessment pipeline in order to retrieve the needed information. Before the risk assessment pipeline is created, make sure that the database connector works.



# Database

In the Entity Relationship Diagram (ERD) underneath, the structure of the database is shown. In this chapter we briefly explain what is in each table.



## EndpointRegistry

Contains all the endpoints that are part of the risk assessment pipeline.

## Rules

Contains all the rules that are used in each endpoint. E.g. if fieldA <comparison> fieldB, then do something with fieldA within a certain scope.

## Comparison

Contains all types of comparisons. E.g. greater than, equals and smaller than.

## Scope

Contains all types of scopes for entities. E.g., global (a.k.a. all), summary\_declaration, transport\_document, line\_of\_package.

## MachineLearning

Contains all the machine learning models that are used in a certain endpoint and apply to a particular scope.

## MachineLearningType

Contains all the machine learning types. E.g., sklearn, keras.

## ModelLocation

Contains all the model locations. E.g., folder name ‘/src/' or bucket name ‘risk-assessment-models'

## ModelLocationType

Contains all the model location types. E.g., local, google.

# Deployment Instructions

In the instructions, we navigate through the whole deployment process.

## Installation on Google Cloud

1. Create a SQL database that is accessible from the source where the risk assessment pipeline runs
2. Import the [initial SQL database](https://atos365-my.sharepoint.com/:u:/r/personal/remon_kruizinga_atos_net/Documents/Veritas/Risk%20Assessment/initial_sql_database?csf=1&e=4puQS6).
3. <for now> Create a bucket on Google Cloud Storage
4. Add the created bucket to the ModelLocation Table (bucket name = ModelLocation.name).
5. Import the model which you would like to use in a step of the pipeline on Google Cloud Storage.
6. Add the model to the MachineLearning Table (model name, the name which will be used within the pipeline = MachineLearning.name, filename of the model in the bucket = MachineLearning.filename), with the correct references from the ModelLocation of step 4.
7. Create Client Credentials in GCP which has Google Cloud Admin rights (might be possible with less rights).
8. In the components ‘fraud-detection’ and ‘risk-assessment-pipeline’, replace the file: ‘src/credentials/gcp-storage-read.json’ with the file created in the previous step.
9. In the component ‘database-connector’ replace the information in the ‘config/development.cfg’, ‘config/local.cfg’ and ‘config/production.cfg’ with the information of the database.

## Run the components

1. Run database-connection on f.e. port 8050
2. Run other endpoints
   1. Risk-assessment-pipeline, port 8000
   2. Xml-parser, port 8100
   3. Business-preprocessing, port 8200
   4. Business-features [tariff-classification, 8400, …]
   5. Business-enrichment, port 8500
   6. Tariff-classification, port 8600
3. Test [example](https://atos365-my.sharepoint.com/:t:/r/personal/remon_kruizinga_atos_net/Documents/Veritas/Risk%20Assessment/curl_request_example.txt?csf=1&e=hnLGx2)

# Example flow

## Line of package

Underneath there is an example of the flow of one *line of package* entity throughout the pipeline. In the output of each step there is **bold text. This is the output of each component.** The output of each component is as follows:

{ “results”: **bold text** }

### Step 1: XML Parser – XML to JSON

Transform XML to JSON.

**Input**

<manifest>data</manifest>

**Output**

{ “manifest”: data }

### Step 2: Risk Assessment Pipeline – Index entities

Index all existing entities as a Model

### Step 3: Business Preprocessing – Preprocess

Preprocesses a line of package according to some preprocessing steps.

**Input**

{ "id":"1",

"input":{

"additional\_desc":"31243",

"commodity":"0101210000",

"container\_reference":"43",

"cus\_code":"OTH",

"danger\_goods":"1702",

"decl\_gross\_wgt":"10000.0",

"decl\_gross\_wgt\_h":"10000.000000",

"decl\_qty":"100",

"description":"iPhone spareparts",

"goods\_type":"02",

"goods\_value":"10000.000000",

"goods\_value\_currency":"USD",

"lp\_marks":"SP00",

"lp\_ref":"1",

"lp\_remarks":null,

"lp\_type":"BG",

"net\_weight":"10000.000000",

"origin\_country":"AF",

"post\_charge":"10000.000000",

"post\_charge\_currency":"USD",

"receptacle":"4321",

"temperature":"4321",

"volume":"10000.000000",

"volume\_unit":"BX",

"weight\_unit":"KGM"

}}

**Output**

{ "business\_preprocessing":**{**

**"unit\_value":1**

**},**

"id":"1",

"input":{

"additional\_desc":"31243",

"commodity":"0101210000",

"container\_reference":"43",

"cus\_code":"OTH",

"danger\_goods":"1702",

"decl\_gross\_wgt":"10000.0",

"decl\_gross\_wgt\_h":"10000.000000",

"decl\_qty":"100",

"description":"iPhone spareparts",

"goods\_type":"02",

"goods\_value":"10000.000000",

"goods\_value\_currency":"USD",

"lp\_marks":"SP00",

"lp\_ref":"1",

"lp\_remarks":null,

"lp\_type":"BG",

"net\_weight":"10000.000000",

"origin\_country":"AF",

"post\_charge":"10000.000000",

"post\_charge\_currency":"USD",

"receptacle":"4321",

"temperature":"4321",

"volume":"10000.000000",

"volume\_unit":"BX",

"weight\_unit":"KGM"

}}

### Step 4: Business Features – Iterate through dynamic pipeline

Iterate through each step of the dynamic pipeline

#### Step 4.1 Tariff Classification – Determine tariff

Determine the tariff based on a *description* from the line of package.

**Input**

{ "business\_preprocessing":{

"unit\_value":1

},

"id":"1",

"input":{

"additional\_desc":"31243",

"commodity":"0101210000",

"container\_reference":"43",

"cus\_code":"OTH",

"danger\_goods":"1702",

"decl\_gross\_wgt":"10000.0",

"decl\_gross\_wgt\_h":"10000.000000",

"decl\_qty":"100",

"description":"iPhone spareparts",

"goods\_type":"02",

"goods\_value":"10000.000000",

"goods\_value\_currency":"USD",

"lp\_marks":"SP00",

"lp\_ref":"1",

"lp\_remarks":null,

"lp\_type":"BG",

"net\_weight":"10000.000000",

"origin\_country":"AF",

"post\_charge":"10000.000000",

"post\_charge\_currency":"USD",

"receptacle":"4321",

"temperature":"4321",

"volume":"10000.000000",

"volume\_unit":"BX",

"weight\_unit":"KGM"

}}

**Output**

{ "business\_features":{

"tariff\_classification":**{**

**"tfidf\_nb":{**

**"alternatives":[**

**{**

**"label":"01",**

**"score":0.19504498295044545**

**},**

**{**

**"label":"04",**

**"score":0.09247975503725943**

**},**

**{**

**"label":"02",**

**"score":0.07258668824119932**

**},**

**{**

**"label":"05",**

**"score":0.06428124903205008**

**}**

**],**

**"prediction":{**

**"label":"03",**

**"score":0.5755652870653161**

**}**

**}**

**}**

},

"business\_preprocessing":{

"unit\_value":1

},

"id":"1",

"input":{

"additional\_desc":"31243",

"commodity":"0101210000",

"container\_reference":"43",

"cus\_code":"OTH",

"danger\_goods":"1702",

"decl\_gross\_wgt":"10000.0",

"decl\_gross\_wgt\_h":"10000.000000",

"decl\_qty":"100",

"description":"iPhone spareparts",

"goods\_type":"02",

"goods\_value":"10000.000000",

"goods\_value\_currency":"USD",

"lp\_marks":"SP00",

"lp\_ref":"1",

"lp\_remarks":null,

"lp\_type":"BG",

"net\_weight":"10000.000000",

"origin\_country":"AF",

"post\_charge":"10000.000000",

"post\_charge\_currency":"USD",

"receptacle":"4321",

"temperature":"4321",

"volume":"10000.000000",

"volume\_unit":"BX",

"weight\_unit":"KGM"

}

}

### Step 5: Business Enrichment – Enrich

Enriches the line of package based on the results of the dynamic flow.

**Input**

{ "business\_features":{

"tariff\_classification":{

"tfidf\_nb":{

"alternatives":[

{

"label":"01",

"score":0.19504498295044545

},

{

"label":"04",

"score":0.09247975503725943

},

{

"label":"02",

"score":0.07258668824119932

},

{

"label":"05",

"score":0.06428124903205008

}

],

"prediction":{

"label":"03",

"score":0.5755652870653161

}

}

}

},

"business\_preprocessing":{

"unit\_value":1

},

"id":"1",

"input":{

"additional\_desc":"31243",

"commodity":"0101210000",

"container\_reference":"43",

"cus\_code":"OTH",

"danger\_goods":"1702",

"decl\_gross\_wgt":"10000.0",

"decl\_gross\_wgt\_h":"10000.000000",

"decl\_qty":"100",

"description":"iPhone spareparts",

"goods\_type":"02",

"goods\_value":"10000.000000",

"goods\_value\_currency":"USD",

"lp\_marks":"SP00",

"lp\_ref":"1",

"lp\_remarks":null,

"lp\_type":"BG",

"net\_weight":"10000.000000",

"origin\_country":"AF",

"post\_charge":"10000.000000",

"post\_charge\_currency":"USD",

"receptacle":"4321",

"temperature":"4321",

"volume":"10000.000000",

"volume\_unit":"BX",

"weight\_unit":"KGM"

}

}

**Output**

{

"business\_enrichment":**{**

**"vague\_description\_suspicion":0.3805203041148706,**

**"wrong\_classification\_suspicion":0.5755652870653161**

**},**

"business\_features":{

"tariff\_classification":{

"tfidf\_nb":{

"alternatives":[

{

"label":"01",

"score":0.19504498295044545

},

{

"label":"04",

"score":0.09247975503725943

},

{

"label":"02",

"score":0.07258668824119932

},

{

"label":"05",

"score":0.06428124903205008

}

],

"prediction":{

"label":"03",

"score":0.5755652870653161

}

}

}

},

"business\_preprocessing":{

"unit\_value":1

},

"id":"1",

"input":{

"additional\_desc":"31243",

"commodity":"0101210000",

"container\_reference":"43",

"cus\_code":"OTH",

"danger\_goods":"1702",

"decl\_gross\_wgt":"10000.0",

"decl\_gross\_wgt\_h":"10000.000000",

"decl\_qty":"100",

"description":"iPhone spareparts",

"goods\_type":"02",

"goods\_value":"10000.000000",

"goods\_value\_currency":"USD",

"lp\_marks":"SP00",

"lp\_ref":"1",

"lp\_remarks":null,

"lp\_type":"BG",

"net\_weight":"10000.000000",

"origin\_country":"AF",

"post\_charge":"10000.000000",

"post\_charge\_currency":"USD",

"receptacle":"4321",

"temperature":"4321",

"volume":"10000.000000",

"volume\_unit":"BX",

"weight\_unit":"KGM"

}

}

### Step 6: Fraud Detection – Detect fraud

Detect fraud in a line of package.

**Input**

{

"business\_enrichment":{

"vague\_description\_suspicion":0.3805203041148706,

"wrong\_classification\_suspicion":0.5755652870653161

},

"business\_features":{

"tariff\_classification":{

"tfidf\_nb":{

"alternatives":[

{

"label":"01",

"score":0.19504498295044545

},

{

"label":"04",

"score":0.09247975503725943

},

{

"label":"02",

"score":0.07258668824119932

},

{

"label":"05",

"score":0.06428124903205008

}

],

"prediction":{

"label":"03",

"score":0.5755652870653161

}

}

}

},

"business\_preprocessing":{

"unit\_value":1

},

"id":"1",

"input":{

"additional\_desc":"31243",

"commodity":"0101210000",

"container\_reference":"43",

"cus\_code":"OTH",

"danger\_goods":"1702",

"decl\_gross\_wgt":"10000.0",

"decl\_gross\_wgt\_h":"10000.000000",

"decl\_qty":"100",

"description":"iPhone spareparts",

"goods\_type":"02",

"goods\_value":"10000.000000",

"goods\_value\_currency":"USD",

"lp\_marks":"SP00",

"lp\_ref":"1",

"lp\_remarks":null,

"lp\_type":"BG",

"net\_weight":"10000.000000",

"origin\_country":"AF",

"post\_charge":"10000.000000",

"post\_charge\_currency":"USD",

"receptacle":"4321",

"temperature":"4321",

"volume":"10000.000000",

"volume\_unit":"BX",

"weight\_unit":"KGM"

}

}

**Output**

{

"business\_enrichment":{

"vague\_description\_suspicion":0.3805203041148706,

"wrong\_classification\_suspicion":0.5755652870653161

},

"business\_features":{

"tariff\_classification":{

"tfidf\_nb":{

"alternatives":[

{

"label":"01",

"score":0.19504498295044545

},

{

"label":"04",

"score":0.09247975503725943

},

{

"label":"02",

"score":0.07258668824119932

},

{

"label":"05",

"score":0.06428124903205008

}

],

"prediction":{

"label":"03",

"score":0.5755652870653161

}

}

}

},

"business\_preprocessing":{

"unit\_value":1

},

"fraud\_detection":**{**

**"production\_model":0.576,**

**"vague\_description\_suspicion":0.576**

**}**,

"id":"1",

"input":{

"additional\_desc":"31243",

"commodity":"0101210000",

"container\_reference":"43",

"cus\_code":"OTH",

"danger\_goods":"1702",

"decl\_gross\_wgt":"10000.0",

"decl\_gross\_wgt\_h":"10000.000000",

"decl\_qty":"100",

"description":"iPhone spareparts",

"goods\_type":"02",

"goods\_value":"10000.000000",

"goods\_value\_currency":"USD",

"lp\_marks":"SP00",

"lp\_ref":"1",

"lp\_remarks":null,

"lp\_type":"BG",

"net\_weight":"10000.000000",

"origin\_country":"AF",

"post\_charge":"10000.000000",

"post\_charge\_currency":"USD",

"receptacle":"4321",

"temperature":"4321",

"volume":"10000.000000",

"volume\_unit":"BX",

"weight\_unit":"KGM"

}

}

