Executive summary

AML **caseManager** is a Case Management tool to enhance the effectiveness of AML Investigation units.

Purpose

This document tries to gather/define the technical requirements for various activities needed to create this tool. Highlevel steps are given below

1. Data ingestion (Batch) from RDBMS to Hadoop Lake.
2. Data preparation/cleansing process using Hive and Pig.
3. Split the data into current and historical based on case id’s/timestamp.
4. Create the different facts for customers,cases etc.
5. Load the hive data into elasticsearch.
6. Visualize the data using Kibana.

Scope

This document defines technical requirements in terms of:

1. Data Ingestion: Different sources and feeds which needs to be ingested into the Hadoop.Sqoop tool will be used to ingest the data from sources to hadoop
2. Data Preparation/cleansing: Rules used to create a filtered list of cases, alerts, transactions, customers, accounts,Alert Similarity, CPR and facts.
3. Non-functional requirements to be taken care of as part of the project.

Highlevel Process Flow

Source Data

Kibana UI

Hadoop Lake

Environments / tools required

Any **Data generator** tool to create the test data for all the source tables.

**MySQL / Oracle** database to store the RDBMS tables.

**Sqoop** tool to extract the data from RDBMS.If there is any other Capgemini ingestion tool is already there in the cluster then we can use that also

[**HDFS**](http://hadoop.apache.org/docs/r0.19.0/hdfs_shell.html) - Hadoop’s distributed file system

[**Hive**](http://hive.apache.org/) - enables querying data held in various places including HDFS (and Elasticsearch, and MongoDB) with a SQL-like query language

[**Beeline**](https://cwiki.apache.org/confluence/display/Hive/HiveServer2+Clients#HiveServer2Clients-Beeline–NewCommandLineShell) - Hive command line interface

[**elasticsearch-hadoop**](https://github.com/elasticsearch/elasticsearch-hadoop) - a connector for Elasticsearch to Hadoop including Hive

[**Elasticsearch**](http://www.elasticsearch.org/overview/elasticsearch) - data store & analytics / search engine

[**Kibana**](http://www.elasticsearch.org/overview/kibana) - data visualization tool for Elasticsearch

Source tables

cases,alert\_case, alerts,transactions,customers,transaction\_types,country ,country\_risk etc.

Data Ingestion Process

Create a generic Sqoop script to read the tablename,metadata,connection parameters and create a file/hive table in Hadoop Lake.

Data Processing Process

There will be a source to target mapping flow created which identifies the key columns in target.

Data cleanup rules will be to clean up the special characters like ‘\n\r’ from the source systems.

Remove the records with invalid/blank data.

Split the data into current month data / Historical based on date.

Prepare the facts like counterparty risk score, Similar alerts, Counterparty Facts.

Enrich the address / industry info for each counterparty from external sources.

Create all the required Facts tables as Hive tables in Hadoop.

Data Visualization Process

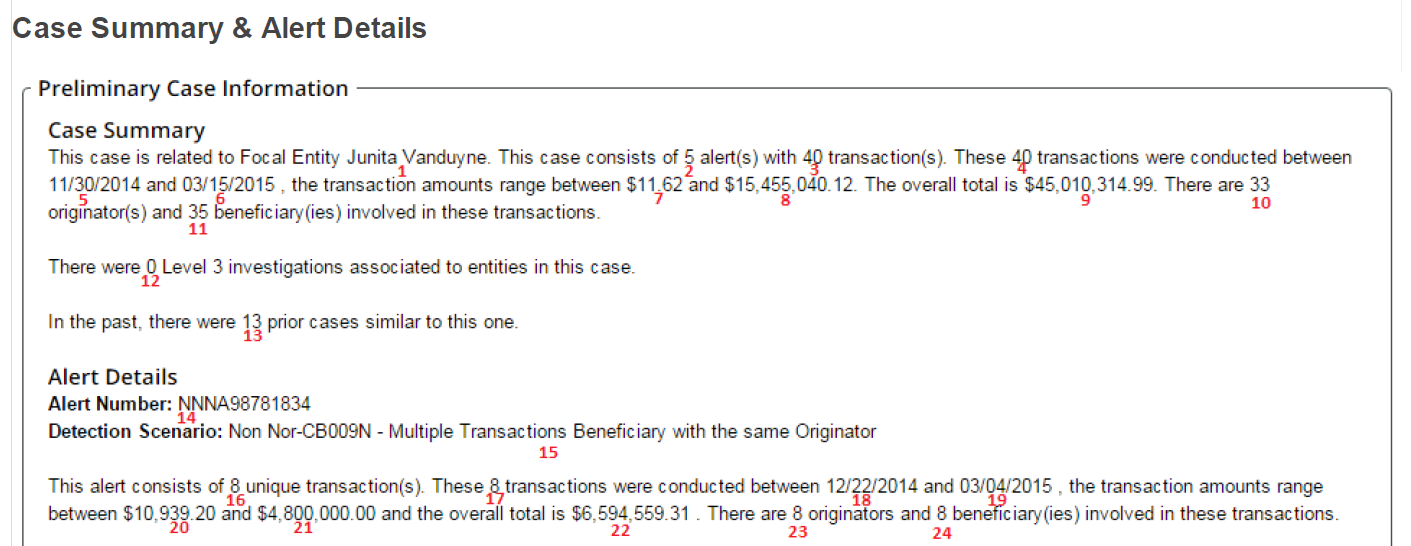
Load the data from Hive to Elasticsearch using the elasticsearch-hadoop connector.

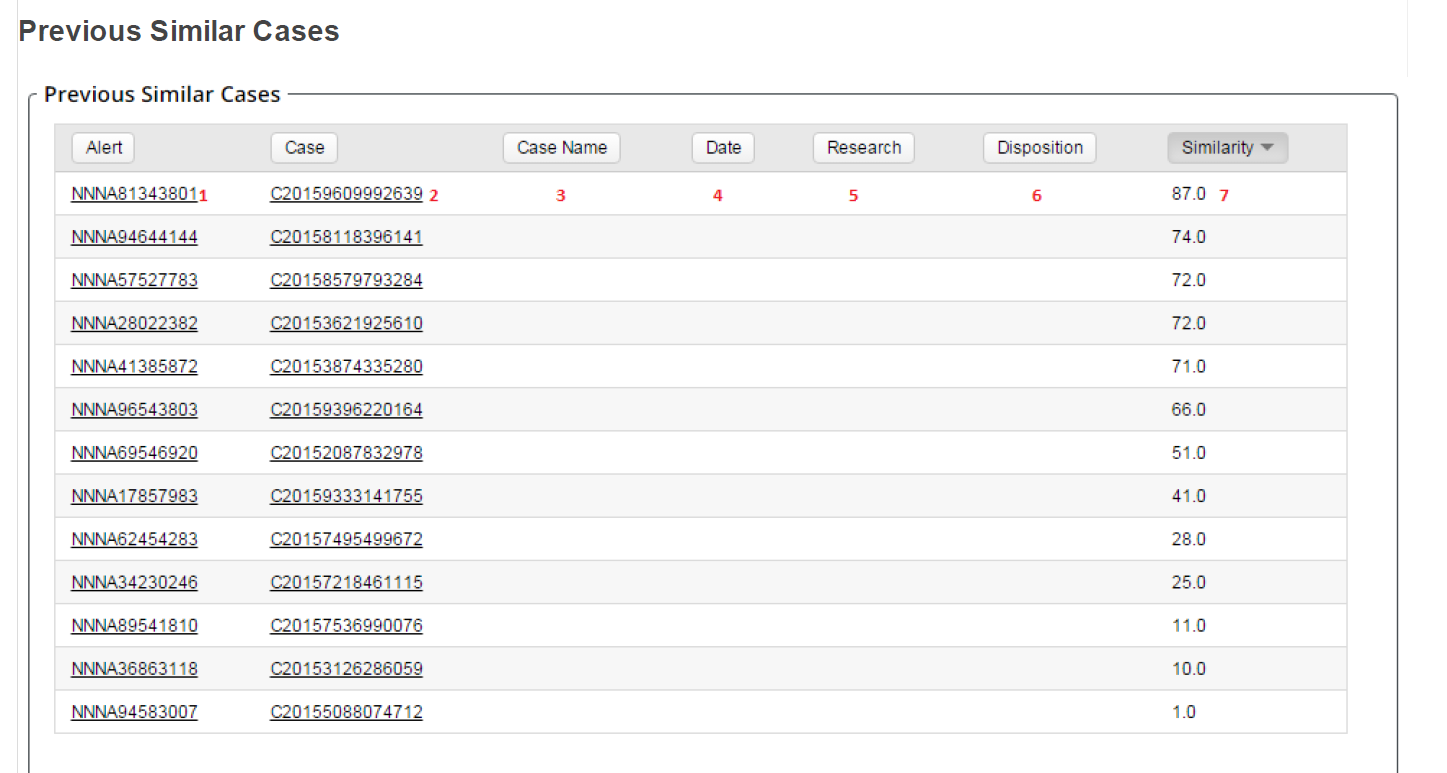
Elasticserach contains index within which data is stored,held under a schema known as mapping.Each index can have multiple mappings.

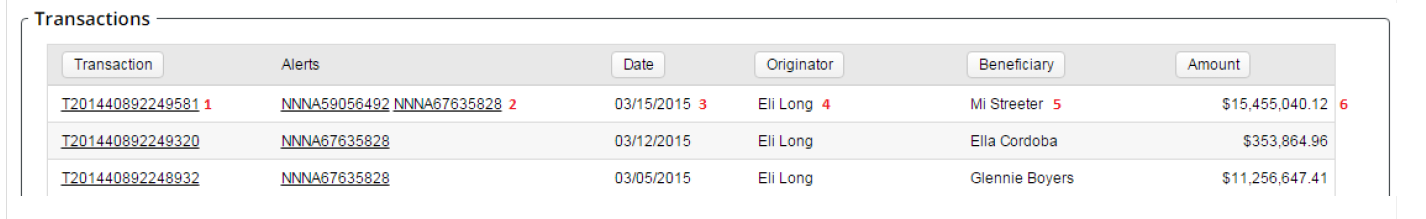
Kibana queries the data held in Elasticsearch and visualize the various case facts.

We need custom UI coding to visualize the case data.

UI Screen for a sample Case







**Counterparty Prioritization model**

**This model scripts can be run by using Hive fact tables created in Hadoop.**

**Input sources/tables:**

Database:casemanager.

Tables:

mon\_alert\_merge

transactions\_merge

transaction\_alert\_tag\_merge

customers\_merge

cpr\_ent\_ind\_risk\_score

customer\_addresses

country

customer\_account\_link

alert\_summary

cp\_alert\_rating

cases\_merge

mon\_base\_alert\_merge

cases\_merge