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

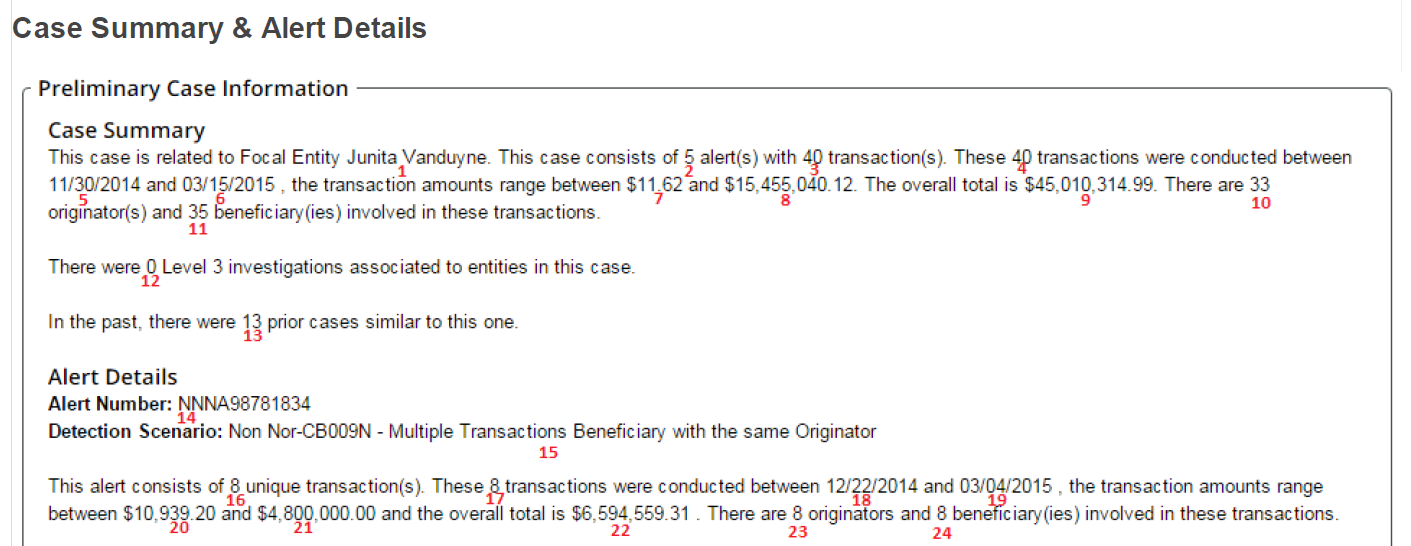
Load the data from Hive to Elasticsearch using the elasticsearch-hadoop connector.

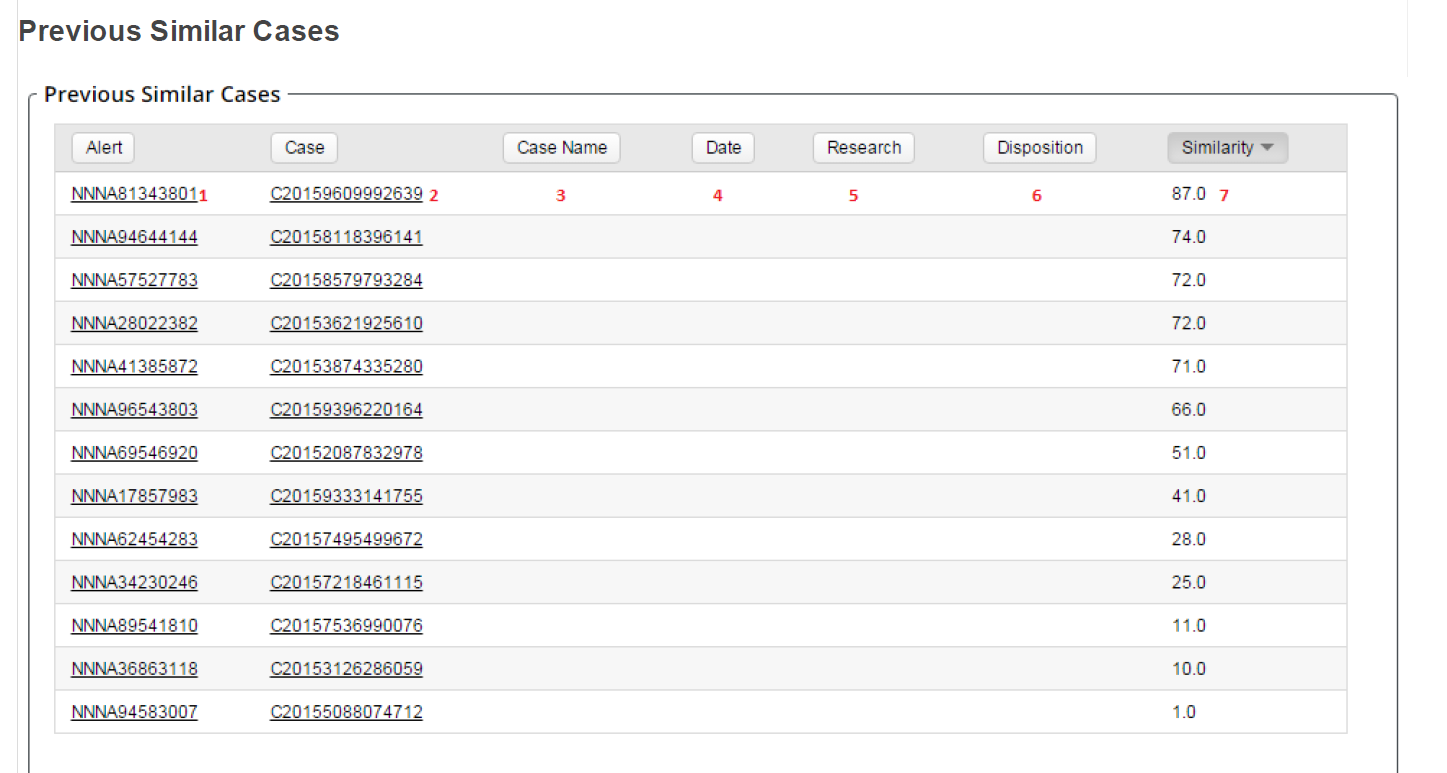
Elasticserach contains index within which data is stored,held under a schema known as mapping.Each index can have multiple mappings.

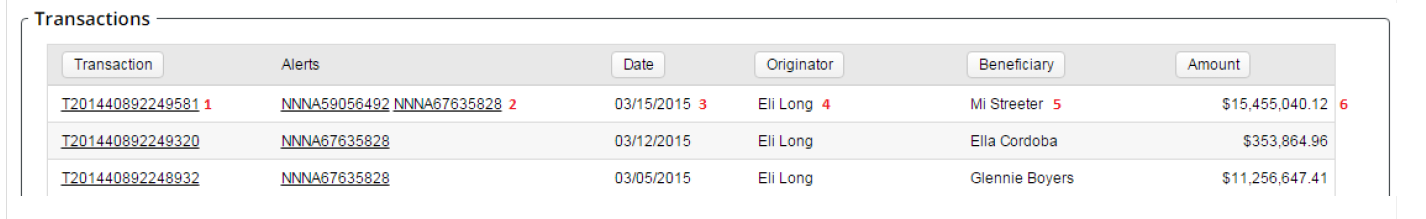
Kibana queries the data held in Elasticsearch and visualize the various case facts.

We need custom UI coding to visualize the case data.

UI Screen for a sample Case

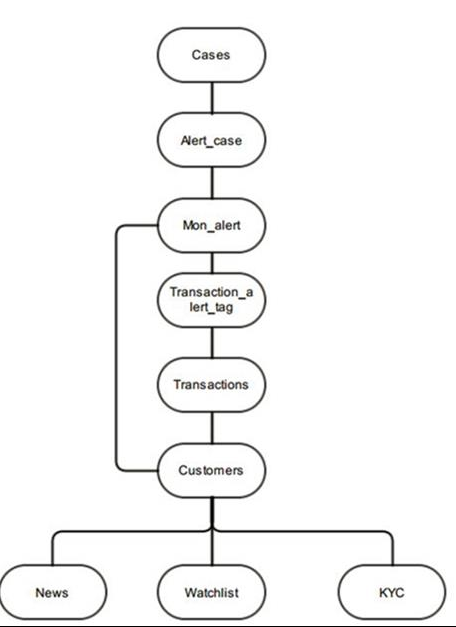






**Source Data flow:**

**Transactions** between **customers** in which one is an originator and other is beneficary.AML Transacton monitoring systems split the transactions into alerted and non alerted transactions.Alerted transactions are transactions which matches some the aml suspected rules and non alerted transactions are which doesn’t match the aml suspected rules.An **alert** is created by grouping 1 or more transactions based on the dectection scenario.A **Case** is created by grouping 1 ore more alerts.



**Counterparty Prioritization model**

Problem Statement

An alert contains multiple counter parties (cp) and multiple transactions. Investigator needs to research on an alert they need to manually go thru all the counterparties and transactions in the alert and then prioritize it

Prioritization Solution

The solutions offers a model where counterparties and transactions are prioritized. This rating will present an objective priority rank values for the investigators based on set of factors which will be consistent across all the reviewers. It is a one counter party prioritization model per LOB and scores are between 1-10.

Counterparty Prioritization

For each alert, the participating counterparties needs to be prioritized within the scope of alert. This means looking at counter party level information (CRR for customers)

The following elements are the potential inputs for model consideration.

Different factors are given below. Each of this variables needs to profiles to assess the data integrity with respect to missing data, invalid data and business rules

Round Dollar - % of round amounts as opposed to non-roundEg: Many fund transfers are sent in large, round dollar, hundred dollar or thousand dollar amounts Deposit of multiple cashier’s checks for foreign drafts in large denominations

Adverse media

Amount % per alert – Count of transactions

CRR (Customer Risk Rating) –Risk rating of counterparty. This is derived from the entity, industry and country data.CRR Values for clients are calculated by client itself.CRR for pseudo customers are calculated as a weighted average of Entity, Industry and Country Risk

Past Alert count – Past alerts for the counterparty (as a focal entity)

# of Relationships – Number of unique Originator and Beneficiary Relationships

Counter Party score determination logic

**Sources**

Cases,Cases alert,Mon alert,Transactions,Customers,Country,Customer address,Customer account link,Transaction alert tag

Rules for identifying the counter party

Retrieve all the Transaction and alert details from the Transaction and alert tables.

Trxn details: Account id,txn amount, orig unit code,orig country,bene country, orig bank country,bene bank country,originator,beneficiary

Alert details:id, mon base alert id,check name,event\_date,customer id

If customer = originator and beneficiary is not blank, then beneficiary is the counter party

If customer = beneficiary and originator is not blank, then originator is the counter party

If customer != originator and customer != beneficiary and originator is not blank, then originator is the counter party

Same logic for Counter party country and Counter party bank country

Filter the alerted transactions alone from the Transaction list(Eg: excluding the monthly total domestic ACH debits )

Identify the entity type and industry for the counterparty

(Eg: Gov,Corporation for Entity, Travel,Tech for Industry)

Identify the country\_score from country risk(High -5,Medium -3 ,Standard -1) and use finally the maximum country score for each unique counterparty key.(counter party key is counter party id+ alert identifier)

Create lookback transaction table for PCM

Create the below attributes based on the above rules

**Alert count** – Distinct mon alert id’s from mon alert grouped by Customer id.

**Relationship count**: Distinct count of beneficiary or originator for unique counterparty, alert

**Scenario Count**: Distinct count of check\_names

**CRR**:Maximum of CRR group on counterparty, alert

Adverse media score ?

**Total Trxn Amount**: Aggregation of trxn amount per counterparty,alert

**Total Trxn count**: Count of Unique transaction id per counterparty,alert

**Total Trxn Rounded amount**: Aggregation of txn\_amount\_base per countrerparty,key for the below filter

**Total Trxn Rounded count**: count (Unique transaction id) per counterparty,key for the below filter

Filter logic for identifying the rounded amounts:

*If txn\_amount\_base >=1 and txn\_amount\_base < 10000 and pMOD(txn\_amount\_base,100) in (0,90,85)) OR*

*If txn\_amount\_base >=10000 and txn\_amount\_base <100000 and pMOD(txn\_amount\_base,1000) in (0,990,985))*

*OR*

*If txn\_amount\_base >=100000 and txn\_amount\_base <1000000 and pMOD(txn\_amount\_base,10000) in (0,9990,9985))*

*OR*

*If txn\_amount\_base >=1000000 and txn\_amount\_base <10000000 and pMOD(txn\_amount\_base,100000) in (0,99990,99985))*

Round amount percentage: (Total Trxn Rounded amount/Total trxn amount)

Round count percentage: (Total Trxn Rounded count/Total Trxn count)

Normalize this variables (BLOM METHOD) and compute the total score. Standardize the scores