HIGH LEVEL DESIGN

FINANCIAL CRIME ANALYSIS

Name: Jay Ganesh Charole

Email: jaycharole@gmail.com

TABLE OF CONTENTS

ABSTRACT

Financial institutions around the world are turning to data science to combat crime and manage compliance due to the changing nature of crime and a quickly expanding regulatory landscape.

The global financial crisis of 2008 altered the course of history. It had an impact not only on the financial industry, but also on other industries and enterprises around the world. The crisis exposed ineffective policies that resulted in severe fractures that threatened to bring the global financial system to its knees.

1. INTRODUCTION

1.1 What is High Level Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

• Present all of the design aspects and define them in detail

• Describe the user interface being implemented

• Describe the hardware and software interfaces

• Describe the performance requirements

• Include design features and the architecture of the project

• List and describe the non-functional attributes like:

- Security
- Reliability
- Maintainability
- Portability
- Reusability
- Application compatibility
- Resource utilization
- Serviceability

1.2. Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

2. GENERAL DESCRIPTION

2.1 Problem Statement

Financial institutions around the world are turning to data science to combat crime and manage compliance due to the changing nature of crime and a quickly expanding regulatory landscape.

The global financial crisis of 2008 altered the course of history. It had an impact not only on the financial industry, but also on other industries and enterprises around the world. The crisis exposed ineffective policies that resulted in severe fractures that threatened to bring the global financial system to its knees.

Technological advancements, and new capabilities to understand enormous volumes of data can help to analyze and formulate the best approach to identify flaws and appropriate interventions techniques to reduce financial crime.

AI, machine learning, and automation, among other advanced analytics and cognitive techniques, can help to filter out false positives and improve inefficiencies in existing investigation processes. Data and analytics have the potential to not only improve efficiencies and save operating costs, but also help identify intelligence-led and data-driven approaches to combating financial crime.

2.2 Proposed Solution

For the problem statement we have decided to make a dashboard which will give us the detail insights of the data we have received. Also it will make the insights better understood by everyone and in an easy manner.

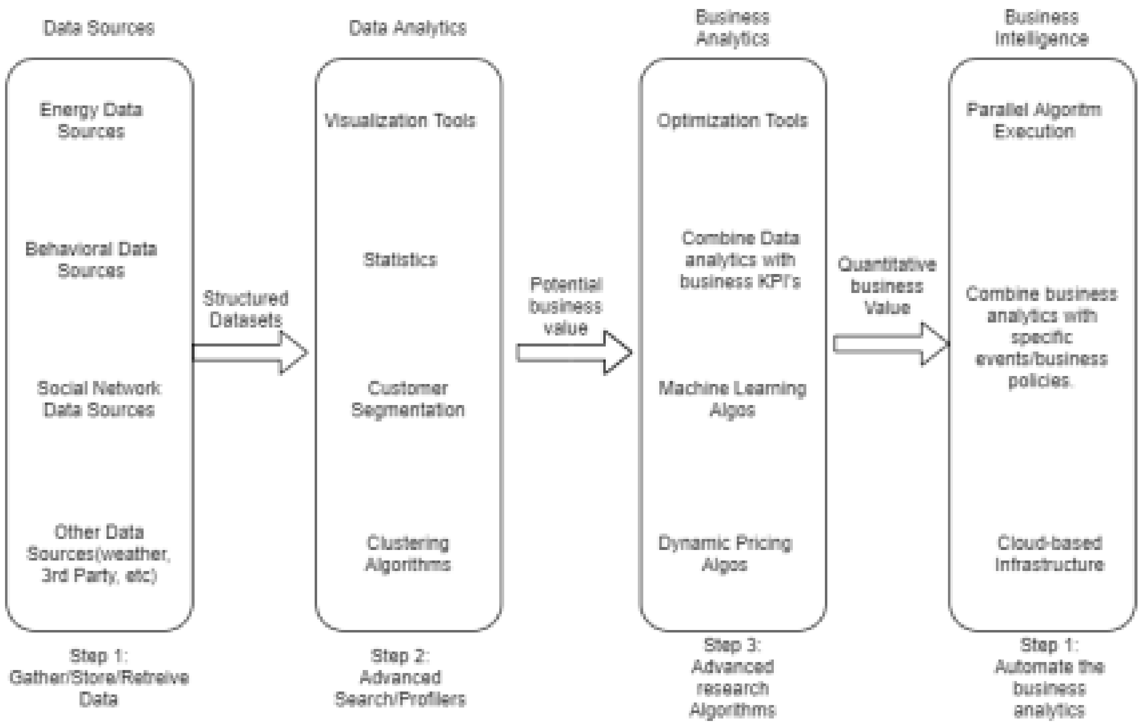
We can also perform EDA on datasets given using Python so that more insights can be gathered for the same.

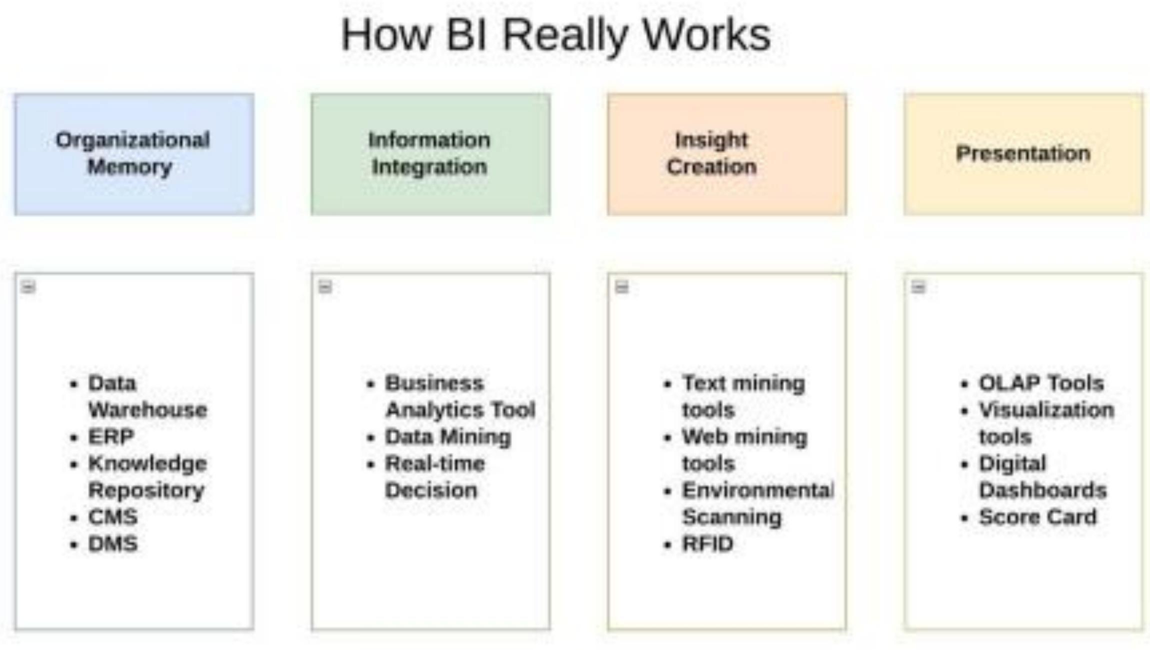
2.3 Tool Used

Business Intelligence tools and libraries works such as Numpy, Pandas, Excel, R, Tableau, Power BI are used to build the whole framework.

3. DESIGN DETAIL

3.1 Functional Diagram


Data Sources
Energy Data Sources
Behavioral Data Sources
Social Network Data Sources
Other Data Sources(weather, 3rd Party, etc)
Step 1: Gather/Store/Retreive Data
Structured Datasets
Data Analytics
Visualization Tools
Statistics
Customer Segmentation
Clustering Algorithms
Step 2: Advanced Search/Profilers
Potential business value
Business Analytics
Optimization Tools
Combine Data analytics with business KPI's
Machine Learning Algos
Dynamic Pricing Algos
Step 3: Advanced research Algorithms
Quantitative business Value
Business Intelligence
Parallel Algoritm Execution
Combine business analytics with specific events/business policies.
Cloud-based Infrastructure
Step 1: Automate the business analytics



3.2 Optimization

Your data strategy drives performance

• Minimize the number of fields

• Minimize the number of records

• Optimize extracts to speed up future queries by materializing calculations, removing columns and the use of accelerated views

Reduce the marks (data points) in your view

• Practice guided analytics. There's no need to fit everything you plan to show in a single view. Compile related views and connect them with action filters to travel from overview to highly-granular views at the speed of thought.

• Remove unneeded dimensions from the detail shelf.

• Explore. Try displaying your data in different types of views.

Limit your filters by number and type

• Reduce the number of filters in use. Excessive filters on a view will create a more complex query, which takes longer to return results. Double-check your filters and remove any that aren't necessary.

• Use an include filter. Exclude filters load the entire domain of a dimension, while include filters do not. An include filter runs much faster than an exclude filter, especially for dimensions with many members.

• Use a continuous date filter. Continuous date filters (relative and range-of-date filters) can take advantage of the indexing properties in your database and are faster than discrete date filters.

• Use Boolean or numeric filters. Computers process integers and Booleans (t/f) much faster than strings.

• Use parameters and action filters. These reduce the query load (and work across data sources).

Optimize and materialize your calculations

• Perform calculations in the database

• Reduce the number of nested calculations.

• Reduce the granularity of LOD or table calculations in the view. The more granular the calculation, the longer it takes.

- LODs - Look at the number of unique dimension members in the calculation.
- Table Calculations - the more marks in the view, the longer it will take to calculate.

• Where possible, use MIN or MAX instead of AVG. AVG requires more processing than MIN or MAX. Often rows will be duplicated and display the same result with MIN, MAX, or AVG. Make groups with calculations. Like include filters, calculated groups load only named members of the domain, whereas Tableau's group function loads the entire domain.

• Use Booleans or numeric calculations instead of string calculations. Computers can process integers and Booleans (t/f) much faster than strings. Boolean>Int>Float>Date>DateTime>String

4. KPI (Key Performance Indicators)

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the dashboard.

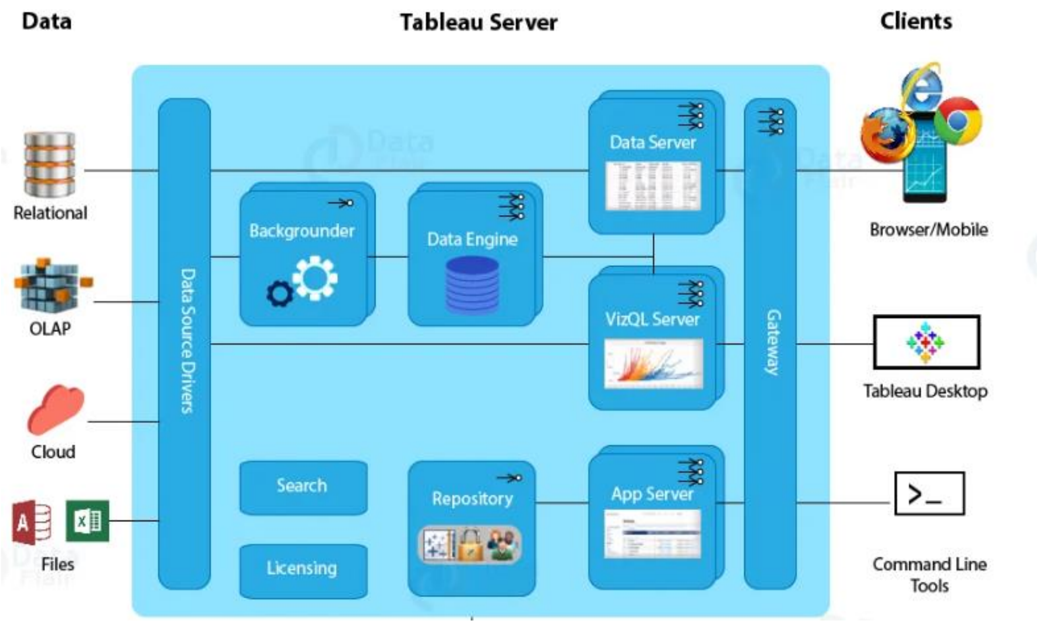
As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors

4.1 Key Performance Indicators

1) Percentage of frauds
2) Alerts used
3) TX amount per id

5. DEPLOYMENT

Tableau Server is an important component in the Tableau architecture as it is thoughtfully designed to manage and execute crucial processes. It is important for us to understand what's under the hood of Tableau Server as it is a core component and helps to understand Tableau better.



1. Gateway

Gateway is a kind of web-server that helps clients communicate to the server via HTTP or https. The server receives incoming client requests and directs them to the appropriate server for action. A gateway handles processes such as load balancing, traffic routing, URL rewriting, serving static files to clients, serving multi-thread processes etc. The gateway server used by Tableau is Apache Tomcat.

2. Application Server

Gateway interacts primarily with the application server, also known as a workgroup server. The application server deals with login processes, domain authentication, data authorization, user or group permission management, content searches, etc. It works in close association with the server's repository and handles the data access operations. Also, it renders processes related to the user interface other than the visualizations which are the work of the VizQL server. The user interface is the page that gives the users an overview of the workbooks and projects they have access to.

3. Repository

The repository in Tableau Server stores server metadata related to users, permissions, assignments, groups, and projects. Along with the metadata, it stores visualizations in flat files (TWS, TDS), and performance data for auditing. Whenever a server service or component demands for metadata, it is provided from the repository. Also, it cooperates with the active directory to provide useful information to the app server for login verification processes.

4. VizQL Server

It is an important component of Tableau Server as it is responsible for loading all the visualizations that you see and work with on Tableau. It has an in-built caching for performance improvement and editing tasks. Whenever a user requests a visualization or wishes to update an existing one, the request received by VizQL is first converted into an SQL statement and sent down to the data sources via respective data source drivers. The requested data sent back from the data source comes to the VizQL server again, where it is processed with some final touches of additional calculations and sent to the user. Any new visualization coming from the data source is cached in the VizQL for further use.

5. Data engine

The data engine is another efficient component which handles processes related to Tableau data extract (TDE). It is invoked only when a query is shot, which involves data from TDE. The data engine stores multiple TDEs and can run on multiple servers maximum 2). It also attends multiple requests parallelly at a given point of time. The data engine hosts the piece of data in-memory extracted from the TDE upon getting a request from the user.

6. Backgrounder

Backgrounder is an essential multi-process, a multi-process element that manages schedules for information refreshing and ensures proper functioning of the Tableau Server and Data Engine.

7. Data Server

The data server helps in centralizing metadata management, driver deployment, and extract management. It also contributes to access control and serves as a proxy to the data sources. It hosts user queries and requests to prevent users from directly accessing the data source.

8. Search and License

Two other important components are search and license. The search component manages the search indexing for the data in the repository. Whereas, the license component is responsible for the licensing and configuration of the Tableau server. Both these services run on the primary server of the Tableau’s server cluster.