

# High Level Design (HLD)

## Analyze International Debt Statistics

Revision Number: 1.0

Last date of revision: 02/02/2023

Abhilash Amrutkar

## Document Version Control

Date Issued	Version	Description	Author
2/2/2023	1.0	First Version of Complete HLD	Abhilash Amrutkar

## Contents

Document Version Control .....	2
Abstract .....	4
1 Introduction .....	5
1.1 Why this High-Level Design Document? .....	5
1.2 Scope .....	5
2 General Description .....	6
2.1 Product Perspective & Problem Statement .....	6
2.2 Tools used .....	6
3 Design Details .....	7
3.1 Functional Architecture .....	7
3.2 Optimization .....	8
4 KPIs .....	9
4.1 KPIs (Key Performance Indicators) .....	9
5 Deployment .....	10

## Abstract

This study provides an analysis of international debt statistics compiled by the World Bank. The data provides insights into the trends and patterns of debt across various countries and regions. The study aims to evaluate the extent and distribution of debt among countries and to identify any potential implications for financial stability and economic growth. The findings indicate that the global debt has increased significantly in recent years, particularly in low-income countries. The results also suggest that there is a need for more effective debt management strategies to mitigate the risks associated with high levels of debt and to promote sustainable economic development.

# 1 Introduction

## 1.1 Why this High-Level Design 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 Product Perspective & Problem Statement

It's not that we humans only take debts to manage our necessities. A country may also take debt to manage its economy. For example, infrastructure spending is one costly ingredient required for a country's citizens to lead comfortable lives. The World Bank is the organization that provides debt to countries. In this project, you are going to analyze international debt data collected by The World Bank. The dataset contains information about the amount of debt (in USD) owed by developing countries across several categories. You are going to find the answers to questions like:

- What is the total amount of debt that is owed by the countries listed in the dataset?
- Which country owns the maximum amount of debt and what does that amount look like?
- What is the average amount of debt owed by countries across different debt indicators?

The data used in this project is provided by The World Bank. It contains both national and regional debt statistics for several countries across the globe as recorded from 1970 to 2015

### 2.2 Tools used

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



### 3 Design Details

#### 3.1 Functional Architecture

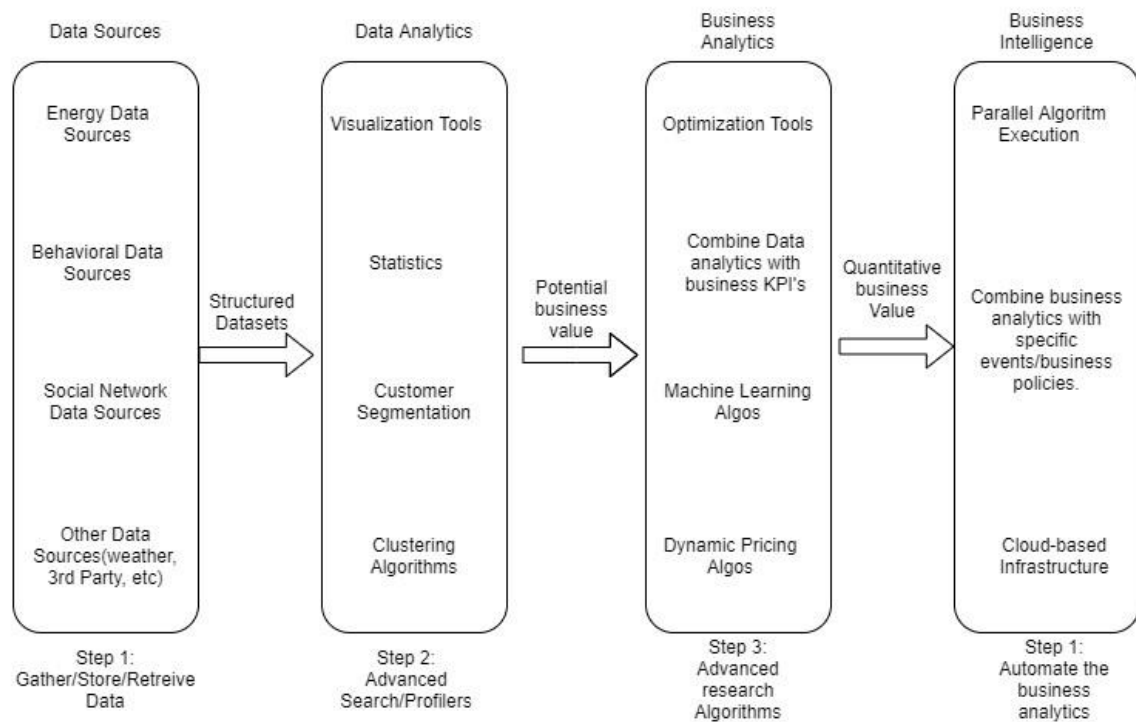


Figure 1: Functional Architecture of Business Intelligence

### How BI Really Works



## 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 KPIs

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



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 KPIs (Key Performance Indicators)

Key indicators displaying a summary of Analysis of international debt statistics.

1. . The World Bank's international debt data
2. Finding the number of distinct countries
3. Finding out the distinct debt indicators
4. Totaling the amount of debt owed by the countries
5. Country with the highest debt
6. Average amount of debt across indicators
7. The highest amount of principal repayments
8. The most common debt indicator

## 5 Deployment

To publish a Power BI report or dashboard to the Power BI service, follow these steps:

1. Open the Power BI Desktop application and sign in with your Power BI account.
2. Load and prepare the data for your report or dashboard.
3. Create the report or dashboard using the Power BI visualizations and tools.
4. Save the report or dashboard to the Power BI service by clicking the "Publish" button in the Home tab.
5. Select the destination workspace in the Power BI service where you want to publish the report or dashboard.
6. Click "Publish" to upload the report or dashboard to the Power BI service.

Once the report or dashboard has been published, users can access it by logging into the Power BI service or by embedding it into another application or website using the Power BI API or embedding code.