Low Level Design

Financial Analytics

Written By	Abhilash Amrutkar
Document Version	1.0
Last Revised Date	

DOCUMENT CONTROL

Change Record:

VERSION	DATE	AUTHOR	COMMENTS
1.0	29- Dec - 2022	Abhilash Amrutkar	

Reviews:

VERSION	DATE	REVIEWER	COMMENTS

Approval Status:

VERSION	REVIEW DATE	REVIEWED BY		APPROVED BY	COMMENTS

Contents

1. Introduction

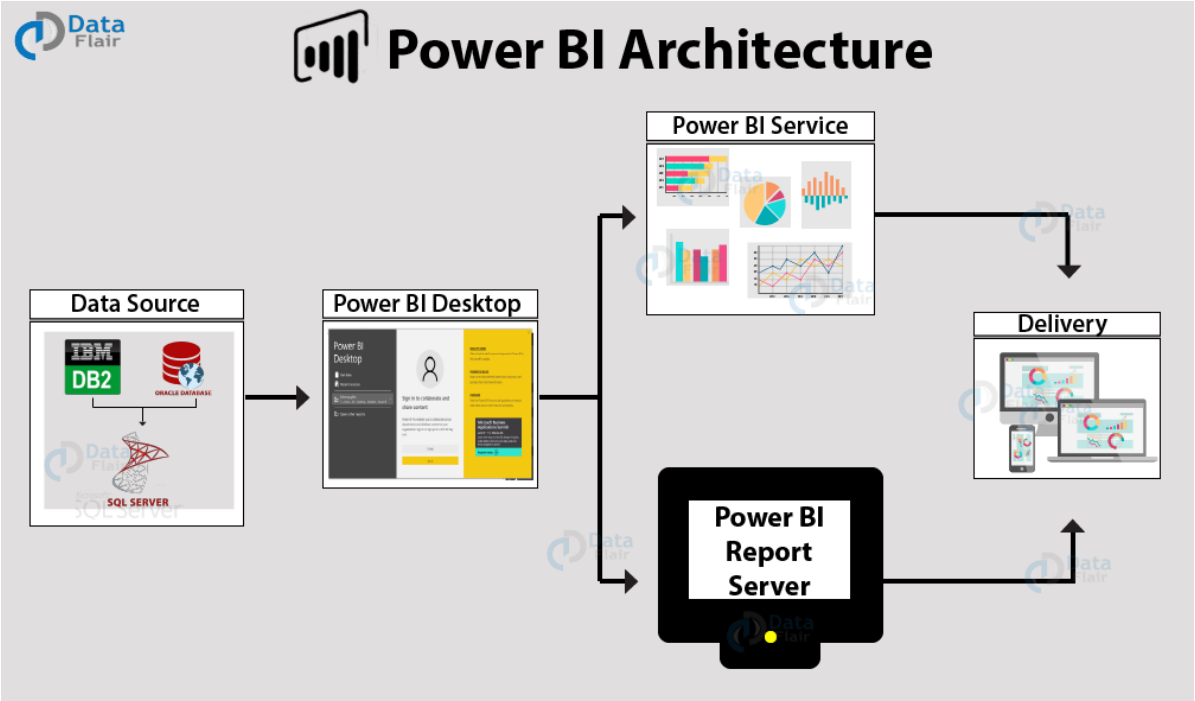
1.1 What is Low-Level design document?

The goal of the LDD or Low-level design document (LLDD) is to give the internal logic design of the actual program code for the Financial Analytics dashboard. LDD describes the class diagrams with the methods and relations between classes and programs specs. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

2. Architecture



Power BI is a business suite that includes several technologies that work together. To deliver outstanding business intelligence solutions, Microsoft Power BI technology consists of a group of components such as:

- Power Query (for data mash-up and transformation)
- Power BI Desktop (a companion development tool)
- Power BI Mobile (for Android, iOS, Windows phones)
- Power Pivot (for in-memory tabular data modeling)
- Power View (for viewing data visualizations)
- Power Map (for visualizing 3D geo-spatial data)
- Power Q&A (for natural language Q&A)

In simple terms, a Power BI user takes data from various data sources such as files, Azure source, online services, DirectQuery or gateway sources. Then, they work with that data on a client development tool such as Power BI Desktop. Here, the imported data is cleaned and transformed according to the user's needs.

Once the data is transformed and formatted, it is ready to use in making visualizations in a report. A report is a collection of visualizations like graphs, charts, tables, filters, and slicers.

Moving on to the chain of processes, you can publish the reports created in Power BI desktop on two kinds of platforms; Power BI Service and Power BI Report Server.

Power BI Service is a cloud-based public platform whereas Power BI Report Server is an on-premise platform protected by firewall security.

You can create dashboards on these platforms by pinning visualizations from your published reports. Lastly, share your dashboards and reports and collaborate with other users from your organization or outside, using delivery options like a web-browser, Power BI on iPad, tablets, laptops, phones, etc.

Components of Power BI Architecture

Let us learn about the components of Power BI architecture in detail.

1. Data Sources

An important component of Power BI is its vast range of data sources. You can import data from files in your system, cloud-based online data sources or connect directly to live connections. If you import from data on-premise or online services there is a limit of 1 GB. Some commonly used data sources in Power BI are:

Excel
Text/CSV
XML
JSON
Oracle Database
IBM DB2 Database
MySQL Database
PostgreSQL Database
Sybase Database
Teradata Database
SAP HANA Database
SAP Business Warehouse server
Amazon Redshift
Impala
Google BigQuery (Beta)
Azure SQL Database
Salesforce Reports
Google Analytics
Facebook
GitHub

You must learn about Power BI Data Sources **thoroughly**

2. Power BI Desktop

Power BI Desktop is a client-side tool known as a companion development and authoring tool. This desktop-based software is loaded with tools and functionalities to *connect to data sources, transform data, data modeling and creating reports.*

You can download and install Power BI Desktop in your system for free. Using Power BI Desktop features, one can do *data cleansing, create business metrics and data models, define the relationship between data, define hierarchies, create visuals and publish reports.*

3. Power BI Service

Power BI Service is a web-based platform from where you can *share reports made on Power BI Desktop, collaborate with other users, and create dashboards.*

It is available in three versions:

Free version
Pro version

Premium version
Power BI Service is also known as, **"Power BI.com"**, **"Power BI Workspace", "Power BI Site"** and **"Power BI Web Portal"**. This component also offers advanced features like *natural language Q&A* and *alerts*.

4. Power BI Report Server

The Power BI Report Server is similar to the Power BI Service. The only difference between these two is that Power BI Report Server is an on-premise platform. It is used by organizations who do not want to publish their reports on the cloud and are concerned about the security of their data.
Power BI Report Server enables you to create dashboards and share your reports with other users following proper security protocols. To use this service, you need to have a Power BI Premium license.
Want to learn more about it? Check out the Power BI Report Server Tutorial

5. Power BI Gateway

This component is used to connect and access on-premise data in secured networks. Power BI Gateways are generally used in organizations where data is kept in security and watch. Gateways help to extract out such data through secure channels to Power BI platforms for analysis and reporting.
Wait! Have you checked our Tutorial on Power BI Gateway

6. Power BI Mobile

Power BI Mobile is a native Power BI application that runs on iOS, Android, and Windows mobile devices. For viewing reports and dashboards, these applications are used.

7. Power BI Embedded

Power BI Embedded offers APIs which are used to embed visuals into custom applications.

3. Architecture Description

3.1. Data Description

The Dataset contains top 500 companies in India along with their Market Capitalization value & Quarter Sales.

1. Mar Cap – It is market capitalization value of company (in Crores)

2. Sales Qtr – It is quarterly sales of company (in Crores)

3.2 Data Transformation

In the Transformation Process, we will find Null values after finding null values those are replaced by 0. In financial analytics Price to sale ratio is important, that will be found by dividing Mar Cap (P) by Sales Qtr. (S) In general, less P/S ratio is good for doing investment in company.

3.3 Deployment

Deploy content to an existing workspace

Deploying content in a working production pipeline, to a stage that has an existing workspace, includes the following:

- Deploying new content as an addition, to a stage that already contains content.
- New content deployed to replace old content, in a current working stage.

Deployment process

Content from the current stage is copied over to the target stage. Power BI identifies existing content in the target stage and overwrites it. To identify which content item needs to be overwritten, deployment pipelines uses the connection between the parent item and its clones. This connection is kept when new content is created. The overwrite operation only overwrites the content of the item. The item's ID, URL, and permissions remain unchanged.

In the target stage, item properties that aren't copied, remain as they were before deployment. New content and new items are copied from the current stage to the target stage.

Auto-binding

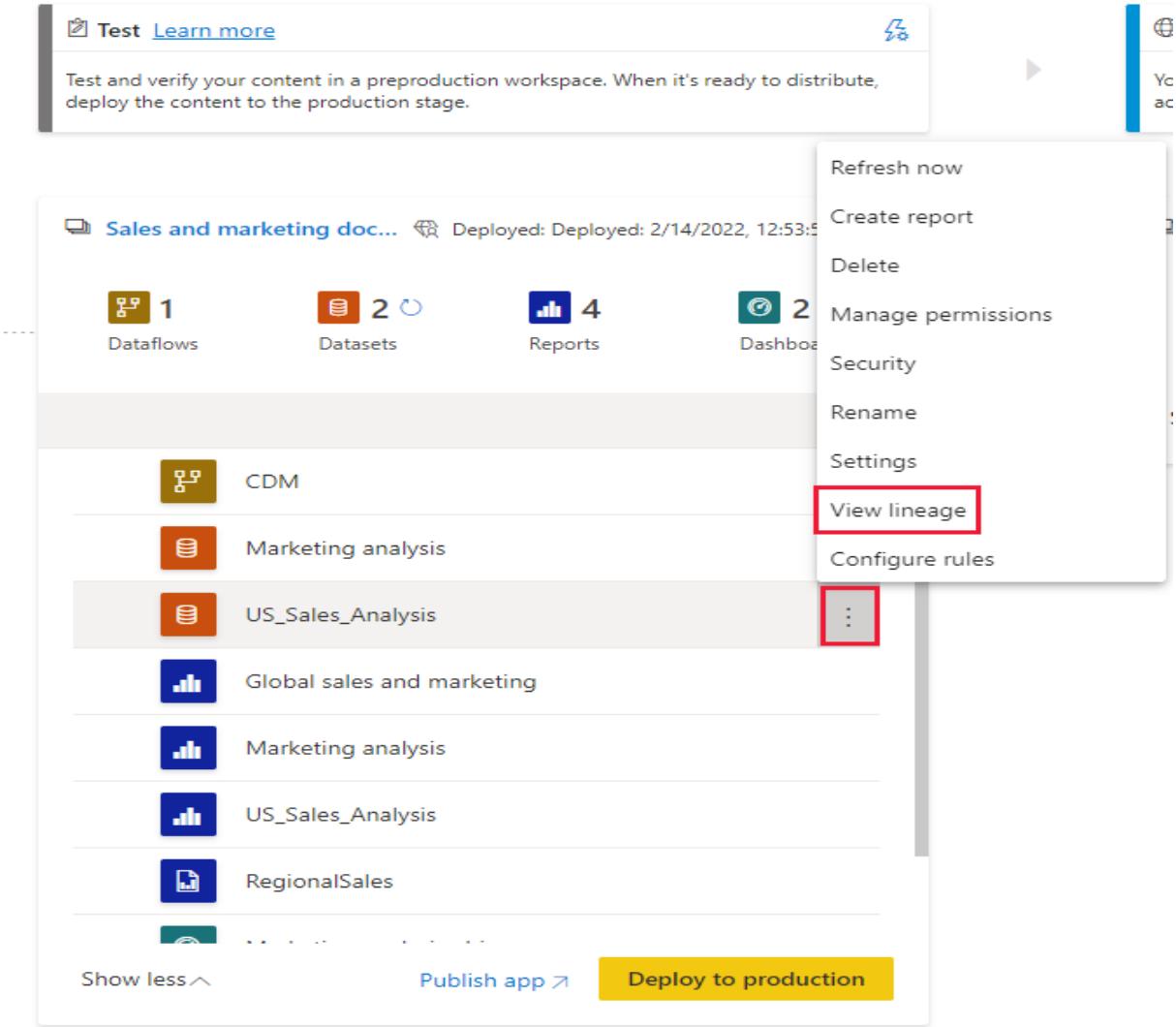
In Power BI, when items are connected, one of the items depends on the other. For example, a report will always depend on the dataset it's connected to. A dataset can depend on another dataset, and can also be connected to several reports that depend on it. If there's a connection between two Power BI items, deployment pipelines will always try to maintain this connection.

During deployment, deployment pipelines checks for dependencies. The deployment will either succeed or fail, depending on the location of the item that provides the data that the deployed item depends on.

- **Linked item exists in the target stage** - Deployment pipelines will automatically connect (auto-bind) the deployed item, to the item it depends on in the deployed stage. For example, if you deploy a paginated report from development to test, and it's connected to a Power BI dataset that was previously deployed to the test stage, it will be automatically connected to that dataset.

- **Linked item doesn't exist in the target stage** - Deployment pipelines will fail a deployment if an item has a dependency on another item, and the item providing the data isn't deployed and doesn't reside in the target stage. For example, if you deploy a report from development to test, and the test stage doesn't contain its Power BI dataset, the deployment will fail. To avoid failed deployments due to dependent items not being deployed, use the *Select related* button. *Select related* automatically selects all the related items that provide dependencies to the items you're about to deploy.

Auto-binding works only with Power BI items that are supported by deployment pipelines and reside within Power BI. To view the dependencies of a Power BI item, from the item's *More options* menu, select *View lineage*.



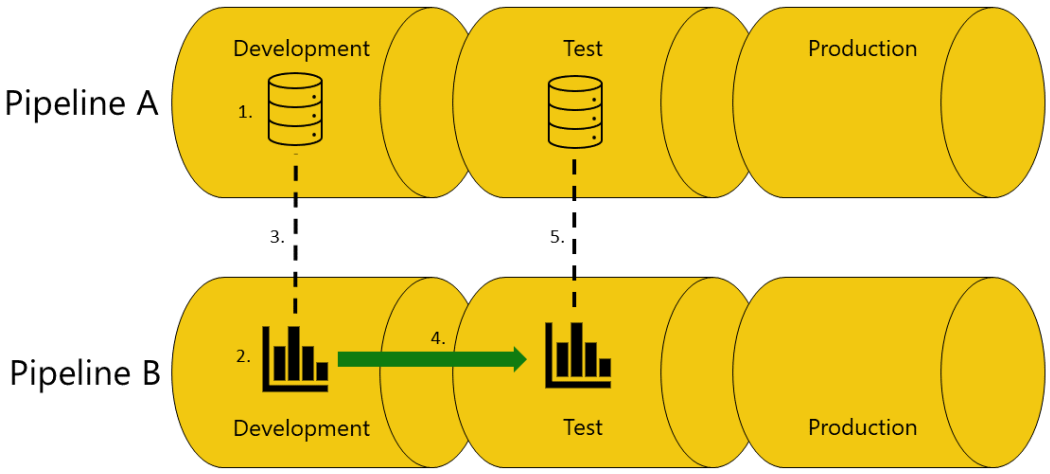
Auto-binding across pipelines

Deployment pipelines automatically binds Power BI items that are connected across pipelines, if they're in the same pipeline stage. When you deploy such items, deployment pipelines will attempt to establish a new connection between the deployed item and the item it's connected to in the other pipeline. For example, if you have a report in the test stage of pipeline A that's connected to a dataset in the test stage of pipeline B, deployment pipelines will recognize this connection.

Here's an example with illustrations that will help demonstrate how auto-binding across pipelines works:

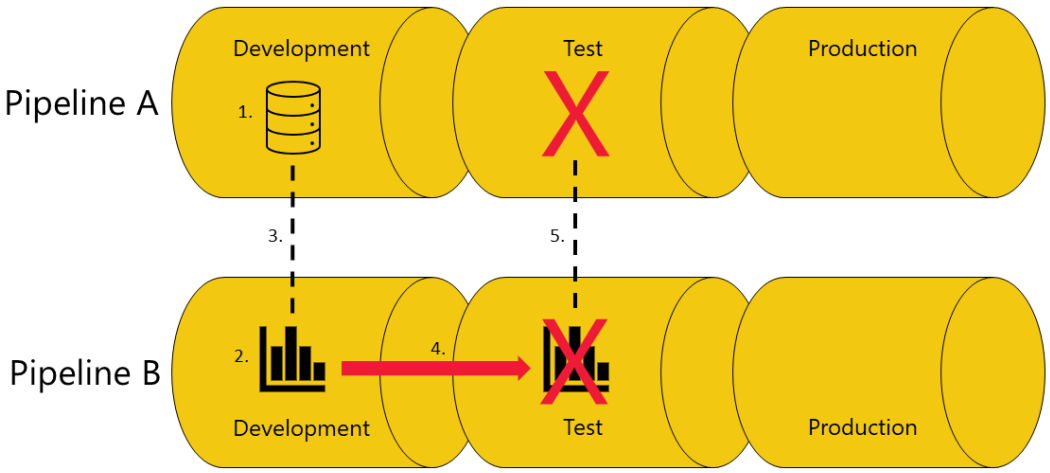
1. You have a dataset in the development stage of pipeline A.
2. You also have a report in the development stage of pipeline B.
3. Your report in pipeline B is connected to your dataset in pipeline A. Your report depends on this dataset.
4. You deploy the report in pipeline B from the development stage to the test stage.
5. The deployment will succeed or fail, depending on whether or not you have a copy of the dataset it depends on in the test stage of pipeline A:
    - *You have a copy of the dataset the report depends on in the test stage of pipeline A*

      The deployment will succeed, and deployment pipelines will connect (auto-bind) the report in the test stage of pipeline B, to the dataset in the test stage of pipeline A.



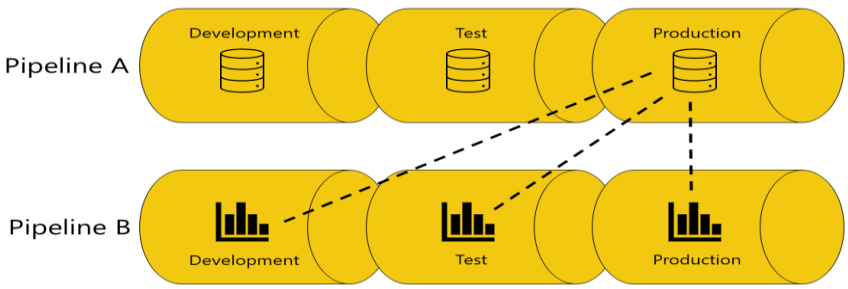
- *You don't have a copy of the dataset the report depends on in the test stage of pipeline A*

  The deployment will fail because deployment pipelines can't connect (auto-bind) the report in the test stage in pipeline B, to the dataset it depends on in the test stage of pipeline A.



Avoid using auto-binding

In some cases, you might not want to use auto-binding. For example, if you have one pipeline for developing organizational datasets, and another for creating reports. In this case, you might want all the reports to always be connected to datasets in the production stage of the pipeline they belong to. To accomplish this, you'll need to avoid using the auto-binding feature



There are three methods you can use to avoid using auto-binding:

- Don't connect the Power BI item to corresponding stages. When the items aren't connected in the same stage, deployment pipelines keeps the original connection. For example, if you have a report in the development stage of pipeline B that's connected to a dataset in the production stage of pipeline A. When you deploy the report to the test stage of pipeline B, it will remain connected to the dataset in the production stage of pipeline A.
- Define a parameter rule. This option isn't available for reports, you can only use it with datasets and dataflows.
- Connect your reports dashboards and tiles to a proxy dataset or dataflow, that isn't connected to a pipeline.

The deployment process lets you clone content from one stage in the pipeline to another, typically from development to test, and from test to production.

During deployment, Power BI copies the content from the current stage, into the target one. The connections between the copied items are kept during the copy process. Power BI also applies the configured deployment rules to the updated content in the target stage. Deploying content may take a while, depending on the number of items being deployed. During this time, you can navigate to other pages in the Power BI portal, but you can't use the content in the target stage.

.