Architecture Design

Financial Data Analysis

Written By	Prateek Lapalikar, Palkesh Gupta
Document Version	0.3
Last Revised Date	

DOCUMENT CONTROL

Change Record:

VERSION	DATE	AUTHOR	COMMENTS
0.1	19- May - 2021	Prateek Lapalikar	Introduction and architecture defined
0.2	20 - May - 2021	Palkesh Gupta	Architecture & Architecture description appended and updated.

Reviews:

VERSION	DATE	REVIEWER	COMMENTS
0.2			

Approval Status:

VERSION	REVIEW DATE	REVIEWED BY		APPROVED BY	COMMENTS

Contents

1. Introduction 04

1.1 What is Architecture Design Document? 04

1.2 Scope 04

2. Architecture 05

2.1 Power Bi Architecture 05

2.2 Power Bi Server Architecture 05

2.3 On-Premise 06

2.4 On-Cloud 06

2.5 Power BI Service 07

2.6 Front End cluster 07

2.7 Working of Power BI Service 08

3 Snowflake Architecture 09

3.1 1 Data Platform as a Cloud Service 09

3.2 Database Storage 10

3.3 Query Processing 10

3.4 Cloud Services 10

3.5 Connecting to Snowflake 11

1. Introduction

1.1 What is Architecture design document?

Any software needs the architectural design to represents the design of software. IEEE defines architectural design as “the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.” The software that is built for computer-based systems can exhibit one of these many architectures.

Each style will describe a system category that consists of :

- A set of components (eg: a database, computational modules) that will perform a function required by the system.
- The set of connectors will help in coordination, communication, and cooperation between the components.
- Conditions that how components can be integrated to form the system.
- Semantic models that help the designer to understand the overall properties of the system.

1.2 Scope

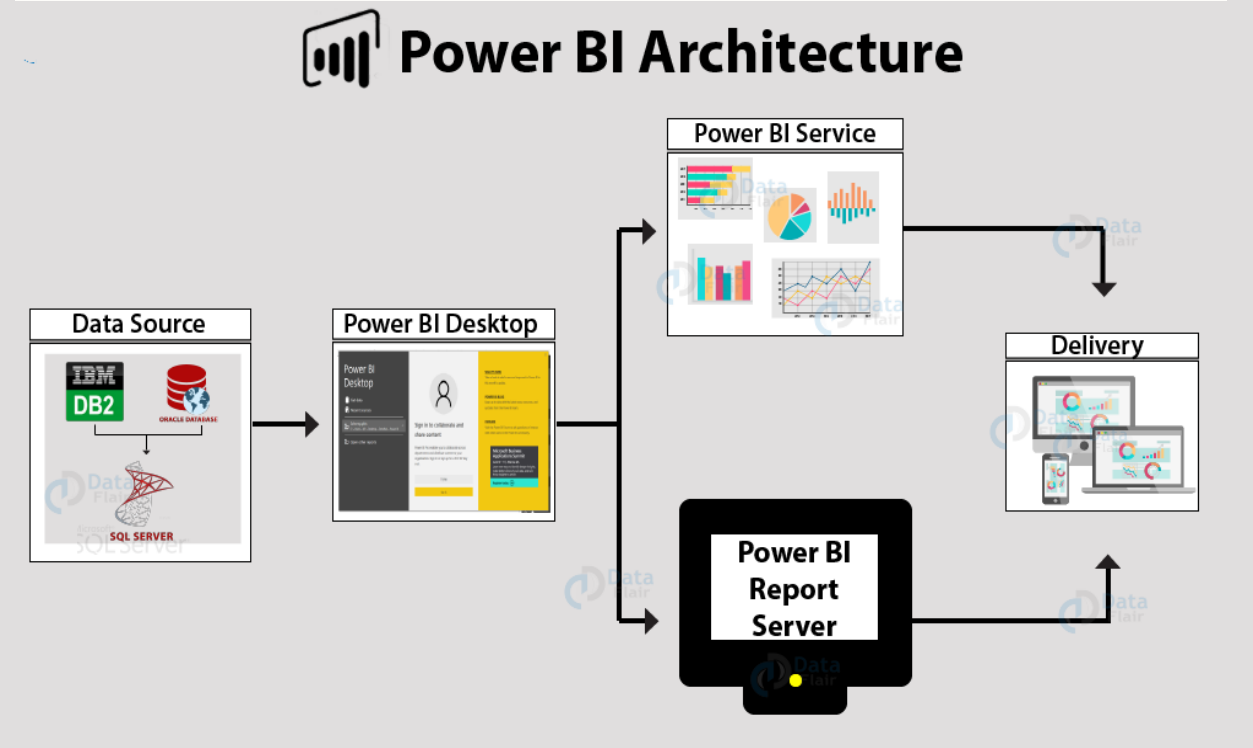
Architecture Design Document (ADD) is an architecture design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the design principles may be defined during requirement analysis and then refined during architectural design work.

2. Architecture

2.1 Power Bi Architecture

2.2 Power Bi Server 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



2.3 On-Premise

Power BI Desktop is a companion development, authoring, and publishing tool. You can import data from data sources to Power BI Desktop and use it to create reports and then publish them on a Power BI Service or Power BI Report Server.
You can also publish Excel workbooks directly using Power BI Publisher for Excel to the Power BI Report Server. The SQL Server Data tools and Report Publisher help in *creating datasets, KPIs, mobile reports, paginated reports,* etc. The reports from all kinds of reports are published to the Power BI Report Server from where they are distributed to the end-users.

2.4 On-Cloud

An important component in Power BI architecture is the Power BI Gateway. The Power BI Gateway acts as a secure channel to transport data from on-premise data sources to on-cloud apps or sites.

On the cloud side of the architecture, resides a lot of components. Like a complete Power BI suite having *dataflows, datasets, dashboards, reports, Power BI Embedded, Power BI Premium,* etc. You can embed your reports and dashboards into *Teams, SharePoint, custom applications,* etc. There are on-cloud data sources as well that connects to Power BI tools via direct connections.

At last, there is a layer of authenticated users who share the published reports and dashboard and collaborate with one another to make educated decisions based on the insights. There are different kinds of users who consume Power BI reports and dashboards and connect through *web browsers, Excel, third-party tools, and mobile devices* (iOS, Windows, Android apps).

Power BI Service

As we have learned in the earlier sections, all the reports that you create in Power BI Desktop are published on a cloud platform known as Power BI Service.

Users can access the reports and dashboards from Power BI Service using client platforms like websites, mobile devices, etc. This means that every client who wants to access content created on Power BI needs to interact with Power BI Service. And so, we must take a look under the hood and learn how Power BI Service works.

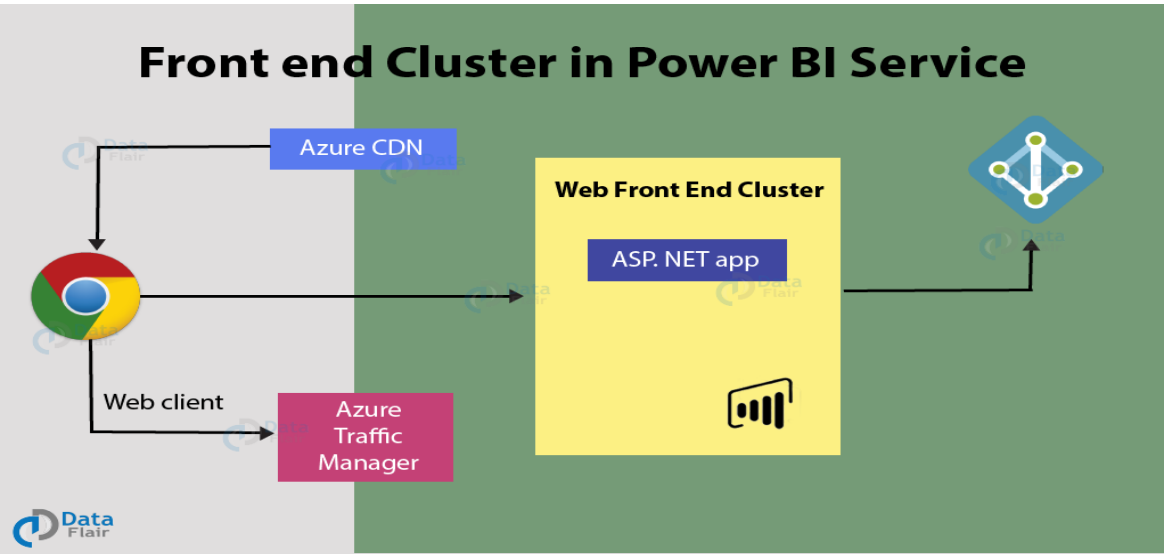
Power BI Service's architecture consists of two parts:

- **A front end**
- **A back end**

Front End cluster

The front end also called the web front-end cluster acts as an intermediary between clients and the back end. The front end services are used for establishing an initial connection and authenticating clients using Azure Active Directory. The Azure Active Directory stores user identities.

Along with this, Azure Traffic Manager is used to direct user requests to the nearest data center after authentication. Once a client/user is authenticated, the Azure Content Delivery Network (CDN) distributes static Power BI content/files to users.



Back End Cluster

The Power BI services at the back end take care of *visualizations, datasets, storage, reports, data connections, data refreshing, and other interactions* with Power BI. At the back-end, a web client has only two direct points of interaction, **Azure API Management**, and **Gateway Role**. These two components are responsible for *load balancing, authentication, authorization, routing,* etc.

Working of Power BI Service

- Power BI stores its data in two main repositories; **Azure block storage** and **Azure SQL database**. Azure block storage stores the datasets uploaded by users and all the metadata and system-related data is stored in the Azure SQL database.
- After Azure API Management authenticates a user request, it is sent to the Gateway Role. The Gateway Role processes the requests and directs them to suitable components like *Presentation Role, Background Job Processing Role, Data Role, and Data Movement Role.*
- For instance, the Presentation Role handles all the visualization related queries like for dashboards and reports.
- For all the data related queries, the request is sent by the Gateway Role to the Data Role or Data Movement Role.
- Power BI Service back end uses Azure Service Bus to connect on-premise datasources with the cloud. Azure Service Bus receives all the requests to fetch data from the on-premise data source. Then it processes the request and executes the query on the on-premise data source to retrieve data from it to the cloud service.
- The Azure Service Fabric manages all the microservices and components associated with running Power BI.
- Azure AD Cache helps in real-time reporting using the data stored in the in-memory of the Power BI system.

3 Snowflake Architecture

Snowflake's architecture is a hybrid of traditional shared-disk and shared-nothing database architectures. Similar to shared-disk architectures, Snowflake uses a central data repository for persisted data that is accessible from all compute nodes in the platform. But similar to shared-nothing architectures, Snowflake processes queries using MPP (massively parallel processing) compute clusters where each node in the cluster stores a portion of the entire data set locally. This approach offers the data management simplicity of a shared-disk architecture, but with the performance and scale-out benefits of a shared-nothing architecture.

3.1 Data Platform as a Cloud Service

Snowflake is a true SaaS offering. More specifically:

- There is no hardware (virtual or physical) to select, install, configure, or manage.
- There is virtually no software to install, configure, or manage.
- Ongoing maintenance, management, upgrades, and tuning are handled by Snowflake.

Snowflake runs completely on cloud infrastructure. All components of Snowflake's service (other than optional command line clients, drivers, and connectors), run in public cloud infrastructures.

Snowflake uses virtual compute instances for its compute needs and a storage service for persistent storage of data. Snowflake cannot be run on private cloud infrastructures (on-premises or hosted).

Snowflake is not a packaged software offering that can be installed by a user. Snowflake manages all aspects of software installation and updates.

3.2 Database Storage

When data is loaded into Snowflake, Snowflake reorganizes that data into its internal optimized, compressed, columnar format. Snowflake stores this optimized data in cloud storage.

Snowflake manages all aspects of how this data is stored — the organization, file size, structure, compression, metadata, statistics, and other aspects of data storage are handled by Snowflake. The data objects stored by Snowflake are not directly visible nor accessible by customers; they are only accessible through SQL query operations run using Snowflake.

3.3 Query Processing

Query execution is performed in the processing layer. Snowflake processes queries using “virtual warehouses”. Each virtual warehouse is an MPP compute cluster composed of multiple compute nodes allocated by Snowflake from a cloud provider.

Each virtual warehouse is an independent compute cluster that does not share compute resources with other virtual warehouses. As a result, each virtual warehouse has no impact on the performance of other virtual warehouses.

3.4 Cloud Services

The cloud services layer is a collection of services that coordinate activities across Snowflake. These services tie together all of the different components of Snowflake in order to process user requests, from login to query dispatch. The cloud services layer also runs on compute instances provisioned by Snowflake from the cloud provider.

Services managed in this layer include:

- Authentication
- Infrastructure management
- Metadata management
- Query parsing and optimization
- Access control

3.5 Connecting to Snowflake

Snowflake supports multiple ways of connecting to the service:

- A web-based user interface from which all aspects of managing and using Snowflake can be accessed.
- Command line clients (e.g. SnowSQL) which can also access all aspects of managing and using Snowflake.
- ODBC and JDBC drivers that can be used by other applications (e.g. Tableau) to connect to Snowflake.
- Native connectors (e.g. Python, Spark) that can be used to develop applications for connecting to Snowflake.
- Third-party connectors that can be used to connect applications such as ETL tools (e.g. Informatica) and BI tools (e.g. ThoughtSpot) to Snowflake.