

Data Architect and the Implementation of an Enterprise Data and Analytics Environment

The Imagnet Approach

Imagnet's Databases and Business Intelligence practice has adopted the principles of Agile Analytics, as best described in *Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing*, by Ken Collier. We value excellence, velocity, and the ability to adapt to change in our development practices. We ensure that our practices allow us to change whatever we need to, at any point of a project, with confidence and high quality.

Imagnet has had broad experience and a high level of success with various clients and projects, implementing the best patterns and practices, and continually refining and improving our approach. We consistently deliver business value in our projects early and often, and help our clients improve their data analytics environments with clear and concise data models that are easily consumable by end-users and other systems.

Our solutions are constructed so that each layer is well-defined, self-contained, and has minimal dependencies. We believe in establishing clear separation of concerns, so that distinct types of development and authoring can take place by different groups of users, but that teams can collaborate well and react quickly and easily to changes. This means that data engineering work is done separately from data modelling, and report authoring is done independently from the data model.

While Power BI can be used for data engineering (Power Query), data modelling (tables and measures), and data visualizations (reports and dashboards), this approach tends to work well only for small, short-lived data analytic efforts, and becomes unwieldy for long-term, enterprise efforts. Many of our client successes have resulted because of our ability to transform a complex and heavyweight Power BI model, by industrializing processes by splitting them out of Power BI, and then giving the report authors a more generalized data model that they can build many standard reports from. This still allows power users to have their own data mashups, by extending the enterprise data models with their own supplementary data.

Data Architect and the Implementation of an Enterprise Data and Analytics Environment

Data Architecture Patterns for Standard Solutions

Imagnet has implemented data engineering, data warehouse, data model, and report authoring solutions using a set of source code projects and code patterns, for many of its clients. We have refined these patterns and practices for each engagement to have a continuously improving process, while allowing us to customize each project for the specific data needs of the client. For example, we have projects for retailers, wholesale distributors, mortgage lenders, academic institutions, and investment and asset management companies, but each database project uses similar patterns for staging data, handling slowly changing dimensions, processing data from structured files, such as JSON or CSV, and providing appropriate interfaces to the consumers of the data (such as Power BI data models, tabular models, or ad-hoc SQL users).

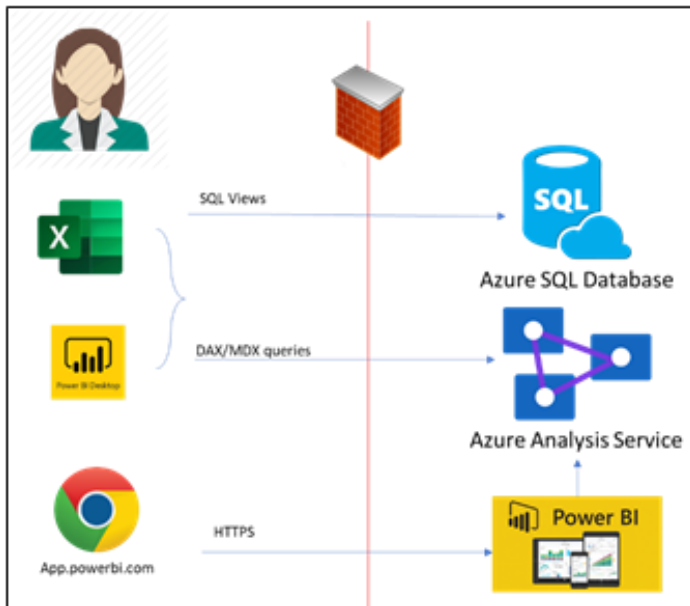


Figure 1: Sample end-user data analytics architecture

Our typical architecture maximizes the use of Microsoft Azure cloud resources, including Azure SQL, Azure Synapse, and Azure Data Lakes for data storage, Azure Data Factory for ETL processes, and various other types of Azure resources for security and functionality purposes, such as Azure Key Vault, Azure Functions, and Azure Analysis Services. Azure is a highly secure cloud service, and all communication is encrypted, as well as all data at rest, which is encrypted by default (or can be configured to be encrypted). Azure resources are highly scalable and elastic, so they can be configured to be as small or large as needed, when needed; they can scale up to handle increases in workloads and scale down again as the workloads decrease. Much of the management requirements for similar on-premises resources, such as SQL Server, are no longer applicable to Azure-based resources. While performance, tuning, and optimization of SQL databases is still preferred, Azure-based SQL databases

Data Architect and the Implementation of an Enterprise Data and Analytics Environment

can be much more easily tuned and scaled to the actual workloads they are experiencing. In comparison, on-premises SQL Servers, do not easily or quickly allow you to add memory, processors, or disk space, and you lack the flexibility to reallocate resources that you no longer need, which would result in an overall cost savings.

In the figure below, the data architecture uses an on-premises SQL Server instance for hosting the data warehouse database and uses Azure Data Factory and its on-premises agent, the self-hosted integration runtime (IR), to extract data from on-premises (and external) resources, such as SQL databases or file systems, and to transform it into the target database, a data warehouse. In this case, the on-premises IR performs all the data queries within the college network (reads and writes) and the cloud-based Azure Data Factory instance is only orchestrating the tasks within the pipeline. Power BI datasets are connected to the internal data warehouse via a Power BI data gateway that performs all data queries against the internal data sources locally, and the cached data is stored in the Power BI service. To use a client's on-premises SQL Server instance for the data warehouse, this architecture would be used.

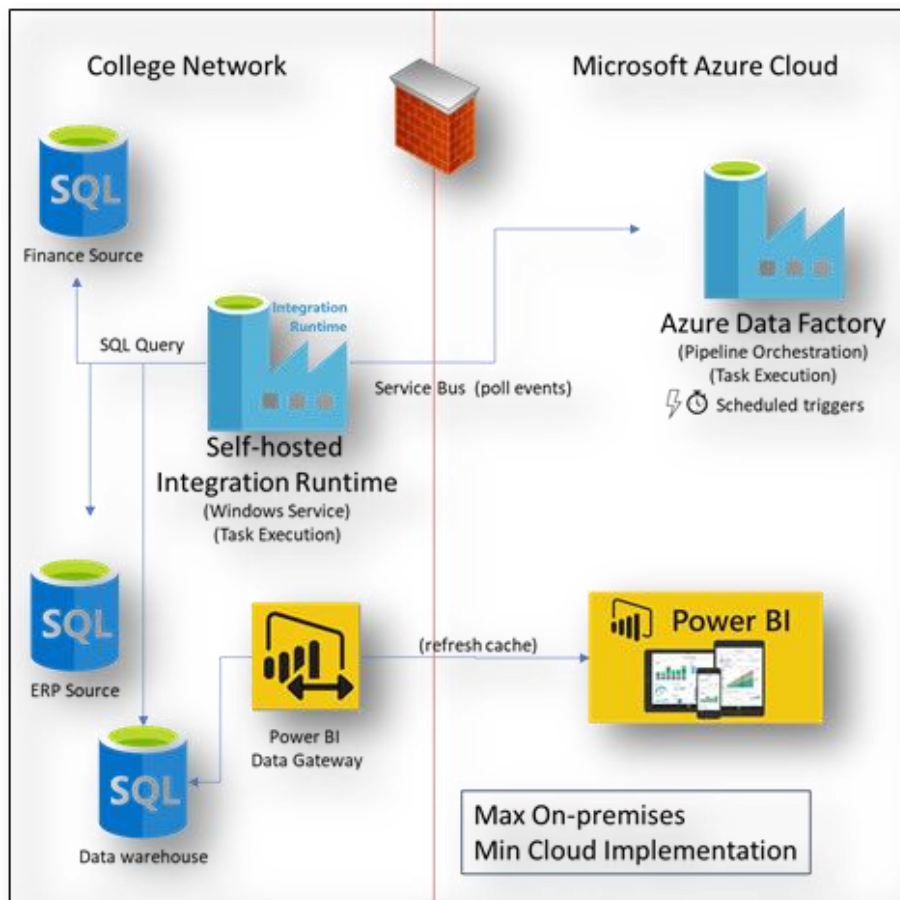


Figure 1: Sample data flow architecture, minimal cloud implementation

Data Architect and the Implementation of an Enterprise Data and Analytics Environment

The architecture below, Imagnet's preferred approach, uses an Azure SQL Database to host the data warehouse. The data flow between the data sources on the corporate network is still handled by the on-premises IR, but the target is now an Azure SQL Database. These databases are limited to an empty IP whitelist by default, so the college network public Internet IP address (or range) would need to be configured on the Azure SQL virtual server instance for the databases to be accessible to the IR. Power BI datasets that use the data warehouse would not need to use a Power BI data gateway, and likely the Azure SQL Database and the Power BI dataset would be in the same Azure data center, leading to extremely high performance for refreshing Power BI.

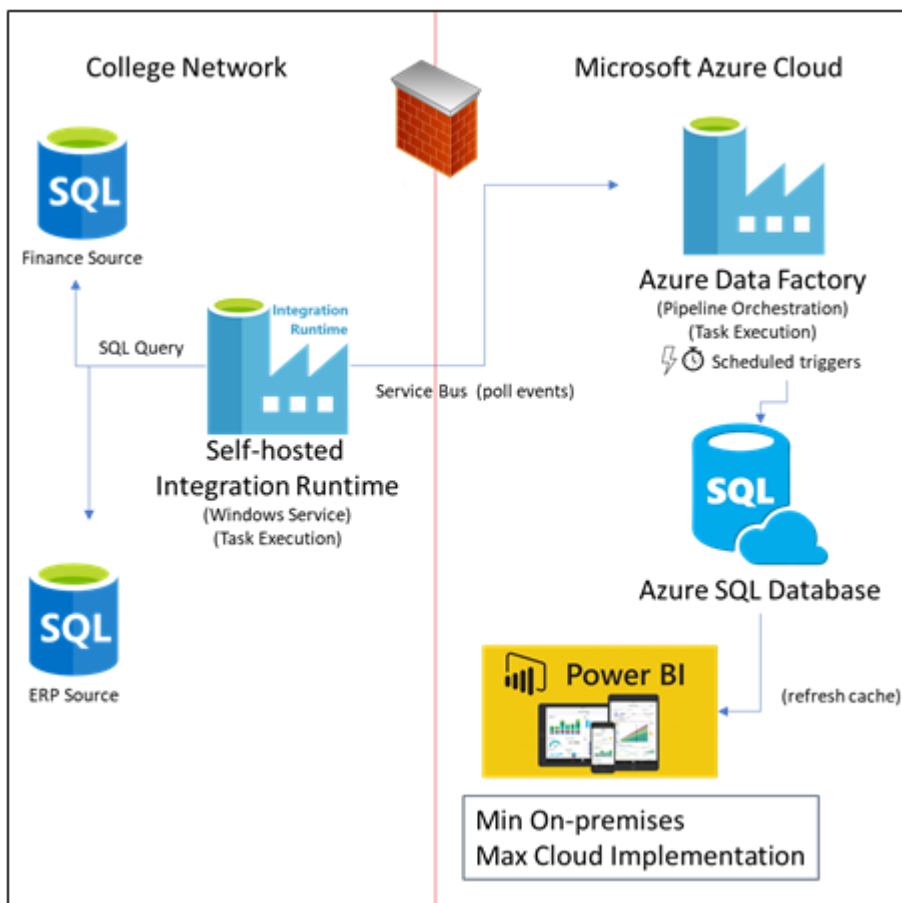


Figure 2: Sample data flow architecture, maximum cloud implementation

Database Projects and Refactoring

We use database projects in Visual Studio as the repository for data warehouse objects and use the deployment tools for those projects to make changes to existing data warehouses. Database projects

Data Architect and the Implementation of an Enterprise Data and Analytics Environment

support various refactoring operations, which allows us to respond to database schema changes very quickly and with a high level of confidence and quality. We can add a new table, view, stored procedure, or other object into the database project, or we can make changes (such as renaming a table or a column) and ensure that all references to that table or column are also hanged, and that the project compiles successfully before an attempt to deploy. Renaming operations are also managed by the deployment tools. For example, a column in a table in the project can be renamed from Postdate to PostingDate, and when that source code is deployed to the database, the deployment process will rename the column PostDate, rather than drop the PostDate column and add a new column PostingDate.

Using database projects brings the value of source code control, including historical changes and rollback processes, to databases. When combined with DevOps processes, to automate database deployments, database schemas can be changed quickly and easily, with a full understanding of the dependencies of the changes, and with confidence that the changes will not break existing assets unexpectedly.

Metadata-Driven Extract/Load/Transform (ELT) Pipelines

Our Azure Data Factory pipelines are driven by metadata stored in the data warehouse, so that once a pipeline is built to support loading from one source into the data warehouse, any additional data that needs to be loaded from that same source is now a matter of adding a row to a table in the data warehouse that contains the metadata. No further pipeline development is needed. Once a metadata-driven pipeline is built for one source, it can handle many data extraction tasks from that source.

On-premises ETL development using SQL Server Integration Services (SSIS) packages, by comparison, requires development changes and package deployment for every data transformation or metadata change, such as adding a new source query or table, or the addition or removal of columns from the source system to an existing transformation. Multiple developers using SSIS packages in Visual Studio concurrently often causes problems integrating changes between team members that can cause quality issues, breakage of ETL processes, loss of development time, and additional deployment time.

Data Architect and the Implementation of an Enterprise Data and Analytics Environment

Using a metadata-driven approach to ETL, with Azure Data Factory, increases developer velocity significantly.

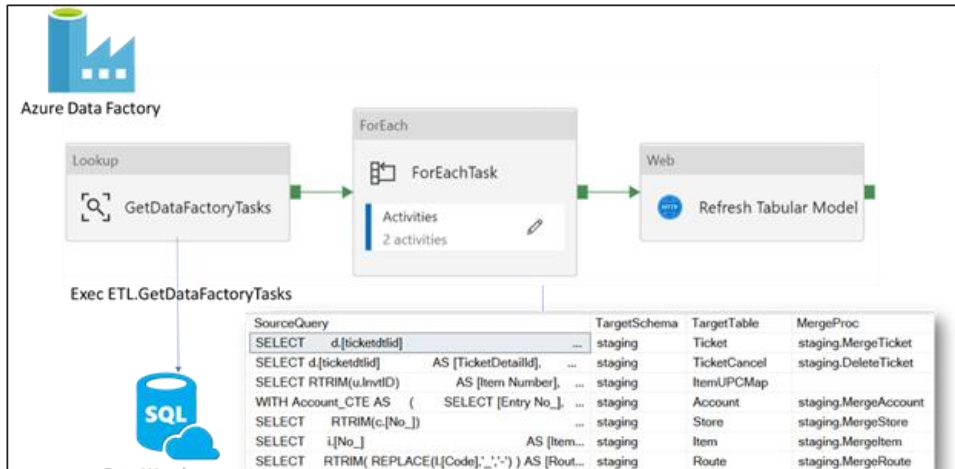


Figure 3: Metadata-driven Azure Data Factory pipeline pattern

Azure DevOps

Imagnet uses Microsoft Azure DevOps for source code repositories, work item (task) tracking, and build and release pipelines. Azure DevOps has five free basic licences and unlimited stakeholder licences, and Visual Studio MSDN licences also include access to Azure DevOps. Imagnet's consultants all have their own MSDN licence. Basic licences include access to the source code repositories and are intended for developer use. Stakeholder licences provide access to the work items and build and release pipelines and are intended for roles such as project managers and users who wish to see the work items and progress boards, such as project stakeholders. Additional basic licences can be purchased for approximately \$6 CAD per month.

Imagnet will work with each client to create a new account in Azure DevOps and configure a new Agile project in that account, associated with that client's Azure Active Directory tenant. If the client already uses Azure DevOps, we can use that account for this project.

Imagnet's deep experience with Azure DevOps brings significant benefits to this project; primarily, it supports an Agile methodology and DevOps processes to increase team velocity and quality. This means that the development lifecycle for our client's data analytics in this project is highly optimized, to deliver real business value as soon as possible.

Continuous Integration/Continuous Deployment (CI/CD)

We have standard build and release pipelines that we construct in Azure DevOps for deployment of data analytics assets, such as data warehouses, Azure Data Factory pipelines, and tabular data analytics

Data Architect and the Implementation of an Enterprise Data and Analytics Environment

models. Once code changes to the database or pipeline, or other assets have been reviewed and successfully compiled through a pull request, they can be deployed immediately to an initial environment (we call this Beta, to ensure that a deployment is successful) and then subsequently to a test environment (we call it UAT, short for User Acceptance Testing) for full data validation. After acceptance testing has passed the UAT build, the Production deployment can be executed with confidence that the approved assets will be reliably sent to the production environment. This is often referred to as CI/CD (Continuous Integration/Continuous Deployment).

Imagnet's team velocity for DevOps is fastest when an Azure-maximized architecture is used. However, we have also implemented the same architecture when using on-premises SQL Server instances for the data warehouse. Azure DevOps release pipelines can use an on-premises agent to deploy to resources (such as SQL Databases) that are only available on the internal (private) client network, given the appropriate permissions in the target database, or on the target server. Usually, the primary constraint for automated deployment of databases to an on-premises SQL Server is the reluctance of DBAs to allow such deployments to occur automatically.

Azure release pipelines can be configured to require approval of deployments by various people, prior to and/or after deployments have occurred in an environment (such as Beta, UAT, or Production).

Work Item Tracking and Source Code Repositories

We use work items in Azure DevOps to track functional and system requirements, and tasks, for design, development, and validation of work. When developers check in code changes to the source code repositories, a policy requires them to associate one or more work items, that the check in gets reviewed by someone else (most often the Imagnet senior data engineer assigned to the project), and that the code compiles successfully. When these policies are enforced, they are a gate that ensures only high-quality code is accepted.

When checked in code is deployed with Azure DevOps release pipelines, the associations of work items made during each check in is maintained, so each release has a list of work items that were implemented in the changed code. This provides auditing capabilities to determine who made what code changes, and when and where the code has been deployed.

Data Architect and the Implementation of an Enterprise Data and Analytics Environment

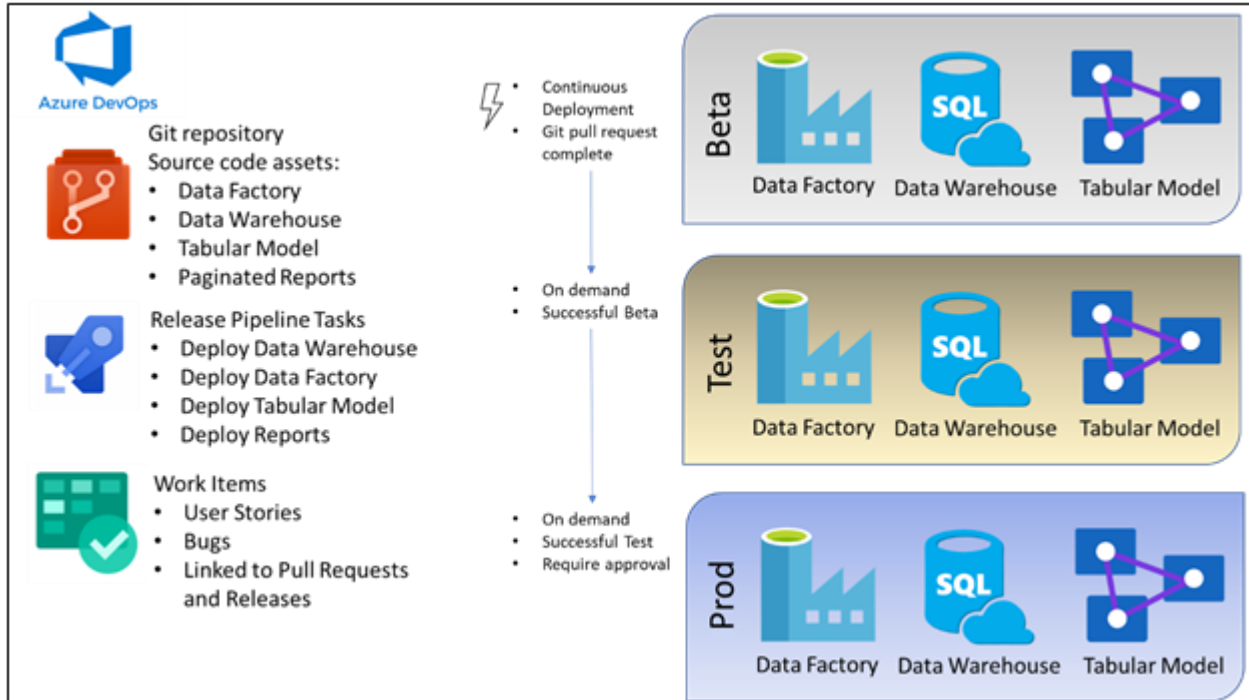


Figure 4: Azure DevOps CI/CD processes

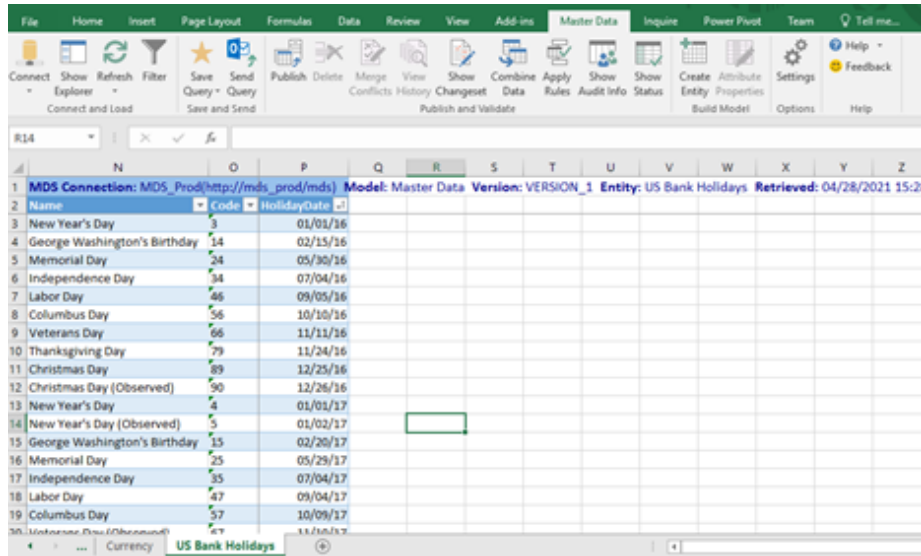
SQL Server Master Data Services

Imagnet has experience with SQL Server Master Data Services, which may benefit certain clients to use in their solution.

Our largest client is using SQL Master Data Services (MDS) to handle its master data and other data attributes that have no system of record. For example, their corporate finance department wants to produce organization-wide, standardized, monthly financial reports and needs to filter and aggregate the expense and employee task hours in a very specific way. They need additional attributes on dimensional data like GL Account or Cost Center, for which the base attributes of those objects are managed by the ERP accounting database. We have implemented a mechanism to populate an entity in MDS called Cost Center, with base attributes from the accounting database and supplemental attributes from a project management database, and leave additional attributes empty for manual updating from a data steward, the vice president of corporate finance. The beauty of MDS is that the end-user interface is through an Excel add-in, the most frequently used tool for data of this nature. The data steward, a business user, can make changes to the MDS data as they wish, and publish those changes (with comments and approval workflow if required), while other systems, such as ADF, can use SQL views to read the data, or load staging tables, and execute stored procedures to programmatically add, update, or delete master data.

Data Architect and the Implementation of an Enterprise Data and Analytics Environment

SQL Server Master Data Services is an Enterprise edition feature and is not available in Azure but is a very powerful tool for providing good data stewardship of master data, and a central repository for integration of master data between systems (including Azure and Power BI).



Name	Code	HolidayDate
New Year's Day	3	01/01/16
George Washington's Birthday	14	02/15/16
Memorial Day	24	05/30/16
Independence Day	34	07/04/16
Labor Day	46	09/05/16
Columbus Day	56	10/10/16
Veterans Day	66	11/11/16
Thanksgiving Day	79	11/24/16
Christmas Day	89	12/25/16
Christmas Day (Observed)	90	12/26/16
New Year's Day	4	01/01/17
New Year's Day (Observed)	5	01/02/17
George Washington's Birthday	15	02/20/17
Memorial Day	25	05/29/17
Independence Day	35	07/04/17
Labor Day	47	09/04/17
Columbus Day	57	10/09/17

Figure 5: Master Date add-in for Excel

Power BI

Imagnet has worked with Power BI since its inception. We have established best practices for dataset, and report and dashboard development, and focus especially on providing clear and concise datasets that are easily understood by end-users, to foster adoption of a self-service data analytics environment.

In addition to Power BI Pro, we have worked with our clients to use the advanced features of Power BI Premium workspaces. We have leveraged XMLA Endpoints in Premium workspaces to deploy tabular models (SSAS Tabular databases), to separate data analytical models from reporting and visualization (these two layers are combined by default in a Power BI Desktop data model). We have also deployed these same models to Azure Analysis Services, the cloud version of SQL Server Analysis Services. We have also deployed RDL report definitions (SQL Reporting Services and Power BI Paginated Reports) to Premium workspaces, using Azure DevOps release pipelines. We have also had experience with monitoring Premium capacity workloads across multiple Power BI workspaces, to identify data models that are consuming the most and least resources.

A key function that Power BI Premium enables is an XMLA endpoint, so that a Tabular data model (SSAS Tabular project) can be deployed to a Premium-enabled workspace. This type of project is functionally equivalent to the data model portion of a Power BI Desktop file, so using this project

Data Architect and the Implementation of an Enterprise Data and Analytics Environment

establishes a clear separation of the data model from the reporting and visualization of data, built in Power BI Desktop. Secondly, data models in the Tabular data project (and deployed to XMLA) have greater control for data refresh operations and may be desirable for incrementally refreshing Power BI data. For example, a large data model may not need to have the entire dataset refreshed, but only the most recent week or day of data.

Power BI Premium capacities also support Paginated Reports, essentially a Power BI-hosted instance of SQL Server Reporting Services, which may be useful for shared operational reporting.

Paginated Reports are designed for printing, as opposed to Power BI reports and dashboards, which are intended to be interactive on screen and not printed.

Premium capacities remove the user licencing requirement for Power BI Pro licences for viewer access to Power BI workspaces. Those users who publish in Power BI workspaces or require Contributor, Member, or Admin role membership in a Power BI workspace still require a Pro licence.

The decision to use Power BI Premium capacities, or Power BI Premium Per User, or Power BI Pro licencing models, is determined by cost and use of features. Typically, if there are about 250 or more users in the organization that would only require viewer access to Power BI assets, then a Power BI Premium capacity is typically a cost savings. Even when using Premium capacity licencing, there is still a need for Power BI Pro licences for anyone who will be authoring or managing Power BI assets.