



Planning a Power BI Enterprise Deployment

Whitepaper

Summary: This is a technical whitepaper outlining considerations for a well-performing and secure organizational Power BI deployment.

Writers: Chris Webb (Crossjoin Consulting), Melissa Coates (SentryOne)

Technical Reviewers: Meagan Longoria (BlueGranite), Adam Wilson, Sirui Sun, Ajay Anandan, Riccardo Muti, Kasper de Jonge, Sergei Gundorov, Jacob Grimm, Dimah Zaidalkilani, Eyal Carmel, Adam Saxton, Kim Manis, Tzvi Keisar, Lukasz Pawlowski

Published: June 2017

Contents

Section 1. Introduction.....	7
Purpose of this Whitepaper.....	7
Scope of This Whitepaper.....	7
High Level Overview of Power BI.....	7
Section 2. Deployment Options and Recommendations.....	9
Scenarios for Power BI.....	9
Key Decision Points.....	10
Who is Going to Consume the Reports?	11
Where Do Consumers Want to View Reports?	12
How Is Power BI Being Utilized?	13
Usage Scenario: Self-Service BI - Teams and Small Groups	13
Usage Scenario: Self-Service BI - Personal.....	14
Usage Scenario: Organizational Reporting	15
Usage Scenario: Prototyping Activities.....	17
Usage Scenario: Operational Reporting / Embedded Analytics.....	18
Usage Scenario: Streaming / Near-Real-Time Reporting	19
Is an On-Premises (Non-Cloud-Based) Solution Required?.....	19
Section 3. Licensing Options.....	20
Power BI Free	21
Power BI Pro	22
Power BI Premium.....	23
Premium Capacity.....	24
Power BI Report Server.....	25
Comparison of Power BI Service vs. Power BI Report Server vs. SQL Server Reporting Services	26
Power BI Embedded.....	28
Comparison of Features Available Per Tier.....	29
Section 4. Source Data Considerations.....	30
Taking an Inventory of Data Sources	31
Data Source Limitations.....	31
Cleansing, Filtering, Transforming and Integrating Data.....	35

Section 5. Power BI Data Storage Options.....	37
Data Storage Modes	37
Option 1. Importing Data.....	37
Option 2. Live Connections to SQL Server Analysis Services.....	38
Option 3. DirectQuery Mode.....	39
Option 4. Real-Time Streaming Datasets.....	40
Choosing a Storage Mode	42
Question 1. Is Analysis Services Already in Use?.....	42
Question 2. How Much Data is Needed by the Report?	44
Question 3. How Frequently does Data Need to be Refreshed?	45
Question 4. How Complex is the Dataset? Are any Complex Calculations Necessary?	46
Question 5. How Many Concurrent Users Need to be Supported?.....	47
Question 6. Are there Legal or Regulatory Reasons for Storing Data in a Specific Location?	48
Question 7. What Technical Skills are Present In-House?	48
Question 8. What Budgetary Constraints Need to be Considered?	49
Summary of Data Storage Recommendations.....	50
Section 6. Data Refresh and the On-Premises Data Gateway	51
Types of Data, File, and Cache Refresh in Power BI	52
Dataset Refresh.....	52
Package Refresh (File Level).....	52
Cached Tile Refresh.....	52
Cached Visual Container Refresh.....	53
Introduction to the On-Premises Data Gateway	53
Gateway Terminology	54
Mode 1: On-Premises Data Gateway	55
Mode 2: Personal mode	56
Summary of Features Per Mode.....	57
When a Gateway is Required.....	58
Process for Using the On-premises data gateway	58
Current On-premises data gateway Limitations	58
Considerations for Using the On-premises data gateway	59

Data Gateway Installation File.....	59
Naming Conventions.....	59
Identities and Credentials Used by the Data Gateway	61
Associating Datasets with a Gateway.....	62
User Interactions with the Gateway	63
Sharing of Data Gateway with Other Applications.....	64
Using One or More Data Gateways	64
Performance Impact on the Server Running the On-premises data gateway.....	65
Where to Install the Data Gateway	66
Firewall Ports.....	66
Monitoring and Managing the Data Gateway.....	67
Data Refresh Duration.....	67
Pausing or Disabling of a Data Refresh.....	67
Updates to the Data Gateway	67
Monitoring Data Refresh Operations	67
Data Refresh Limitations Based on License Type	69
Gateway Disaster Recovery	69
Section 7. Report Development Considerations	69
Choosing the Right Tool for Report Development.....	69
Power BI Desktop.....	70
Power BI Service	70
Excel	71
Best Practices for Report Design in Power BI Desktop	72
Managing Report Authorship	72
Version Control for Reports	73
Separating Dataset Development from Report Authorship	74
Use Templates to Speed Up and Standardize Report Development.....	75
Centralize Data Source Connection Information	76
Minimizing the Amount of Data Loaded	76
Reducing Dataset Size.....	77
Minimizing the Amount of Data Displayed on a Page	77
Custom Visuals.....	78

Custom Data Connectors.....	78
Creating Dashboards to Help Navigation	78
Section 8. Collaboration and Sharing.....	79
Using Apps and App Workspaces for Collaboration and Sharing.....	79
Two Types of Workspaces	79
What Should an App Workspace Contain?.....	80
App Workspaces and Apps	81
Naming an App Workspace.....	82
Permissions for an App Workspace	82
Moving from Development to Testing to Production	83
Using Organizational Content Packs for Collaboration and Sharing.....	83
End User Personalization of Reports and Dashboards	83
Distribution of Shared Datasets and Starter Reports	84
Apps vs Organizational Content Packs Summary	84
Sharing Dashboards	84
Publishing Reports to the Web.....	85
Options for Viewing Reports and Dashboards.....	85
Viewing Reports and Dashboards in the Power BI Service in a Browser	86
Power BI Report Server	86
The Power BI Mobile App for Windows 10.....	86
The Power BI Mobile Apps for Phones and Tablets.....	86
Analyze in Excel	87
Microsoft Teams.....	87
Quick Insights.....	87
Embedded Reports in SharePoint Online and SharePoint On-Premises	87
Visio.....	88
Viewing Data from Power BI in a Custom Application	88
Q&A	88
Cortana	89
Options for Subscriptions and Alerting	89
Subscriptions	89
Alerts.....	89

Integration with Microsoft Flow	90
Section 9. Administration, Security, and Compliance.....	90
Power BI Administration	90
Administrator Roles Related to Power BI.....	90
Office 365 Administrators and The Power BI Administrator Role	91
Initial Tenant Setup	91
Choosing an Azure Data Center for Power BI.....	92
Managing Users and Licenses.....	93
Managing Power BI Pro Trials	93
Managing Power BI Premium Capacities.....	93
Assigning Workspaces to Power BI Premium Capacity.....	94
Deploying Power BI Desktop.....	95
Managing Power BI Mobile App Deployment.....	95
Compliance and Auditing	96
Monitoring Usage.....	96
Data Privacy and Security.....	97
Controlling Access to Power BI.....	97
Managing Access to App Workspaces, Apps, and Content Packs	98
Row-Level Security	98
Securing Data at the Data Source Level.....	99
Restricting Sharing and Publishing	100
Restricting Export and Printing.....	100
Data Privacy Levels in M Queries	100
Data Classification	101
Security Keys.....	101
Section 10. Getting Support from Microsoft and the Community.....	101
Support and Help	102
Learning More About Power BI.....	103
Additional Resources and Links	104

Section 1. Introduction

Purpose of this Whitepaper

Deploying Power BI in a large enterprise is a complex task, and one that requires a lot of thought and planning. The purpose of this white paper is to help you make your Power BI deployment a success: it covers key considerations, decisions which will be necessary, and potential issues you may encounter. Best practices and suggestions are offered when possible.

The target audience for this white paper are technology professionals. Some knowledge of Power BI and general Business Intelligence concepts will be assumed.

Scope of This Whitepaper

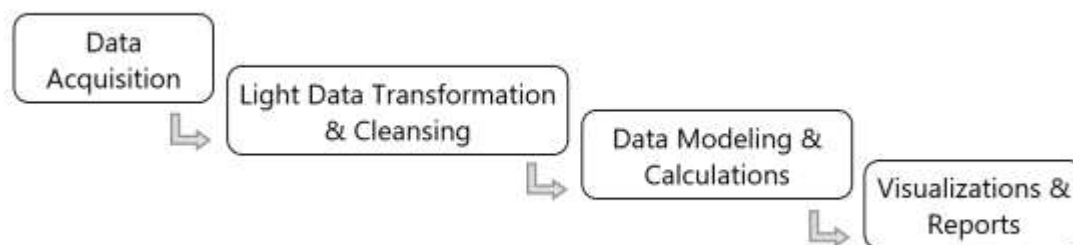
Unless specifically noted, all features mentioned in this whitepaper are available as of June 12, 2017.

The following topics are out of scope for this whitepaper:

- Power BI fundamentals (such as the meaning of terms such as dataset, report and dashboard, or how to create reports and dashboards).
- ISV deployment scenarios, which are handled differently from enterprise deployment scenarios.
- Performance tuning of DAX or M syntax (though that is a very important topic and deserves a whitepaper of its own).
- SQL Server Reporting Services, beyond direct comparisons to Power BI Report Server.
- Third party solutions which integrate with Power BI.
- Power BI security in-depth, as it is covered in a [separate whitepaper](#).

High Level Overview of Power BI

Power BI is both a SaaS (software-as-a-service) and PaaS (platform-as-a-service) platform which facilitates:



Version 2 of Power BI, which is [based on Azure infrastructure](#), has grown rapidly. With over 5 million users, Power BI has been ranked a leader on Gartner's Magic Quadrant for BI & Analytics.

Power BI is still a relatively young product which is maturing and growing rapidly. New features and functionality are introduced every month based on customer feedback. Due to the frequent pace of change related to Power BI features, be cautious about online guidance because information very easily becomes inaccurate as Power BI grows beyond serving small self-service scenarios into an enterprise-level toolset.

Best practices are evolving as integration points, system monitoring and auditing, security, scalability, and report lifecycle management capabilities mature and deepen.

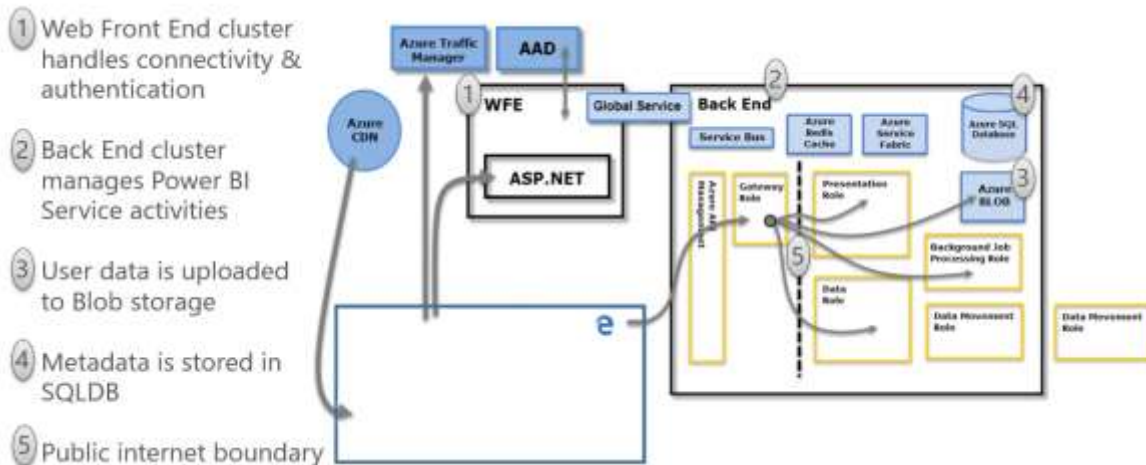


There is an abundance of information about Power BI online, due to a vibrant user and partner community. However, the frequent pace of change represents a challenge for accuracy. Make it a practice to stay current, and verify the information found online since it can quickly become out of date.

The term Power BI, as used in this whitepaper, is meant to refer to the ecosystem as a whole. The most prominent components of the Power BI suite of tools include (not an exhaustive list):

Power BI Desktop	A desktop-based authoring tool of choice for connecting, transforming, and modeling data for creating interactive reports.
Power BI Service	A web-based service for collaboration, sharing, creation of dashboards, and additional functionality such as Q&A and alerts. It may also be referred to as the "Power BI Site" or "PowerBI.com" or perhaps the "Power BI Web Portal."
Power BI Mobile Apps	Native applications for iOS, Android, and Windows for viewing reports and dashboards from anywhere.
On-Premises Gateway	An agent installed within the corporate network to allow secure access to organizational data stored on-premises. There are two options: Enterprise, and Personal.
Power BI Report Server	An alternative to the Power BI Service for deploying Power BI reports within an on-premises data center, as opposed to the cloud-based Power BI Service. Power BI Report Server requires a Power BI Premium license.
Power BI Embedded	APIs for embedding visuals into a custom application.

The Power BI Service is built on Microsoft's Azure environment:



More information about Power BI architecture can be found [here](#).

Section 2. Deployment Options and Recommendations

We face a combination of business requirements and technical requirements when making decisions related to technology deployments. This section presents options to investigate to help you make optimal architectural decisions which fit your situation.

Scenarios for Power BI

Power BI is flexible and can be utilized in several different types of scenarios:

- 1 **Self-service data analysis (ad hoc analysis)**
With self-service scenarios, users import data into Power BI and create their own dataset (aka data model) and calculations, followed by reports and dashboards. The roots of Power BI are here, so some of the strongest functionality lies in its data acquisition capabilities (the functionality found in the Query Editor) and data modeling and calculation (DAX) capabilities. In this scenario, value is typically placed on providing freedom and flexibility to the report author.
- 2 **Organizational reporting (traditional enterprise reporting / corporate BI)**
In this scenario, Power BI can work like any other reporting tool in which it issues live queries to retrieve source data when a report is viewed. Although data could be imported, this scenario can also be handled with DirectQuery functionality (for a relational data source) or a Live Connection (for an Analysis Services data source). In

this scenario, value is placed on having data remain in a centralized system (such as a data warehouse or semantic layer).

3 Embedded analytics (operational reporting)

Power BI reports and dashboards can be embedded into a custom application using custom APIs. This type of integration can greatly improve workflow efficiency for users performing specific functions.

4 Streaming (near real-time monitoring)

Internet of Things (IoT) and streaming solutions are becoming more prevalent. Power BI can be used effectively for a streaming dashboard solution, in conjunction with Azure Stream Analytics, the Power BI APIs, or PubNub.

Key Decision Points

As you will infer from the above scenarios, there are some key questions to ask when planning a deployment:

Where is the source data stored?	In a web-based cloud service, in a proprietary on-premises database, etc
Who is going to be responsible for dataset and/or report development?	Segment of dataset authors, report authors, and consumers
Who is going to consume the reports?	Select number of internal users only, a wide variety of internal users, and/or external users such as customers or business partners
What are the usage patterns?	Dashboards, reports, interactive filtering and/or downloads of data
How do users want to consume these reports?	In a web browser, via mobile devices, embedded within an application, etc
How much data latency (delay) is acceptable?	Importing the data generally offers superior performance, but requires a data refresh which introduces latency
Do corporate security standards permit publishing data to a cloud-based service?	The Power BI Service is cloud-based and offers the most functionality; on-premises deployments can be done with Power BI Report Server integration
What is the complexity level of the solution?	Model size/complexity, query complexity, row-level security needs

Who is Going to Consume the Reports?

The following are options, discussed in further depth in [section 8](#) of this whitepaper, to consider for different ways to share content with consumers:

Type of Report Consumer	Sharing-Related Features	Type of License Required for Content Creator
Individual user scenario	No sharing or collaboration	Power BI Desktop + Power BI Free
Small group within the organization	Dashboard sharing <i>(best suited for a small # of viewers)</i> or Power BI App <i>(better suited for a large # of viewers)</i>	Power BI Desktop + Power BI Pro
Wide audience within the organization	Power BI App or Power BI Embedded <i>(embedded analytics)</i>	Power BI Desktop + Power BI Pro + Power BI Premium
Specific third party user outside the organization	Sharing of URL	Power BI Desktop + Power BI Pro
General public	Publish To Web	Power BI Desktop + Power BI Free or Power BI Pro

Where Do Consumers Want to View Reports?

Several options are available for viewing of reports. These options are discussed in more detail in [Section 8](#) of this whitepaper, including:

Report Viewing Preference	Options to Consider
Web browser	Power BI Service
Mobile device	Power BI Mobile App (iOS, Android, Windows)
Power BI Report Server	Power BI Premium + Power BI Report Server
Custom application	Power BI Premium + Power BI Embedded
Excel	Analyze in Excel and/or Excel native functionality for queries and the built-in Excel data model
SharePoint	SharePoint Online SharePoint Web Parts SharePoint Live Tiles

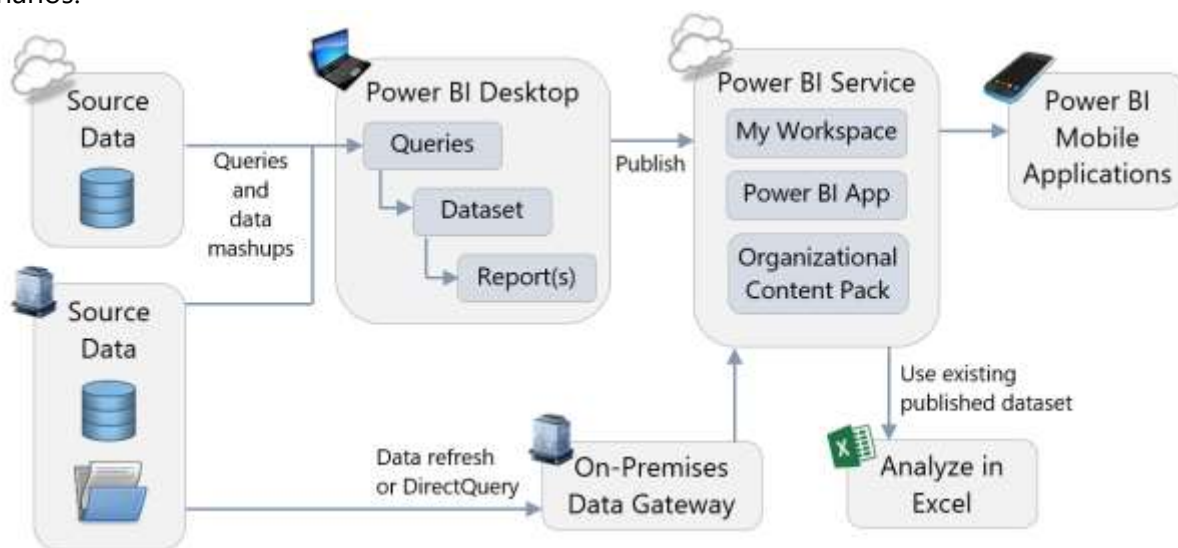
There are additional possibilities, such as Cortana, discussed in [Section 8](#).

How Is Power BI Being Utilized?

In this section, the most prevalent components or features in the Power BI ecosystem are identified based on how Power BI is intended to be used. The objective here is not to call out every possible option – just the primary features to investigate for each scenario.

Usage Scenario: Self-Service BI - Teams and Small Groups

The use of Power BI in self-service scenarios is a very common use case. Following is a highly simplified representation of the prominent Power BI components to support self-service BI scenarios:



Potential Solution Components:

Power BI Desktop. Most commonly, Power BI Desktop is utilized in “import mode” in which data can be acquired from multiple data sources (i.e., mashups) and Power BI Desktop contains a dataset. Imported datasets will need to be refreshed using the On-Premises Data Gateway, whereas DirectQuery or Live Connection will not require a refresh.

OR

Excel. The built-in Excel Data Model and Queries functionality (formerly referred to as the add-ins of Power Query, Power Pivot and Power View) offer similar functionality to Power BI Desktop. These features are updated on a much slower release cycle than Power BI Desktop.

Power BI Service with Power BI Pro. For collaboration and sharing purposes, the Power BI Service is the best option. Sharing and collaboration features require a Pro license for both the creator and the receiver. Although Power BI Desktop files can be shared among users via the file system, that is not the recommended approach because it is highly limiting. The Power BI Service opens up additional features as well, such as dashboards, Q&A, and data alerts.

Power BI App. A Power BI App serves as a container in the Power BI Service for related reports, dashboards, and datasets. This is useful for organizing content by subject area and/or team, particularly helpful for broadly delivering content to a large number of users. Apps are always read-only for viewers, so it is not currently possible to utilize them for personalization scenarios.

Organizational Content Pack. Although Power BI Apps are intended to replace Organizational Content Packs, there is still one scenario where the use of Organizational Content Packs is recommended: to facilitate self-service scenarios such as the distribution of datasets and/or starter reports. Sharing using Organizational Content Packs in this way can help because it allows users to make their own changes to reports and dashboards. Because Apps will supersede content packs, it is recommended to use Apps if they satisfy your requirements.

Analyze in Excel. This feature permits a live connection from Excel to a published dataset in the Power BI Service. This technique is recommended for use whenever possible because:

- ✓ It reuses an existing (centralized) dataset which greatly reduces the number of redundant datasets, and
- ✓ Reduces the risk of datasets and calculations being out of sync, and
- ✓ Allows the user to utilize all slice, dice, and pivoting capabilities within Excel

Power BI Service Live Connection. Usage of a Power BI Service Live Connection, available in the Get Data area of Power BI Desktop, is similar to Analyze in Excel in that it permits connectivity to a published dataset. However, a Power BI Service Live Connection currently requires edit privileges on the dataset, so it's not as flexible for use with self-service scenarios (i.e., with numerous read-only users).

On-Premises Data Gateway. For refreshing datasets in the Power BI Service, a data gateway is required for secure access to data within the corporate network. The data gateway also supports Live Connection and DirectQuery operations. A gateway is generally not required when accessing cloud data. The On-Premises Data Gateway is strongly recommended over the Personal Gateway for team/group scenarios.

Usage Scenario: Self-Service BI - Personal

Personal self-service BI scenarios are slightly different than team-based situations.

Potential Solution Components:

Power BI Desktop. Same as above.

OR

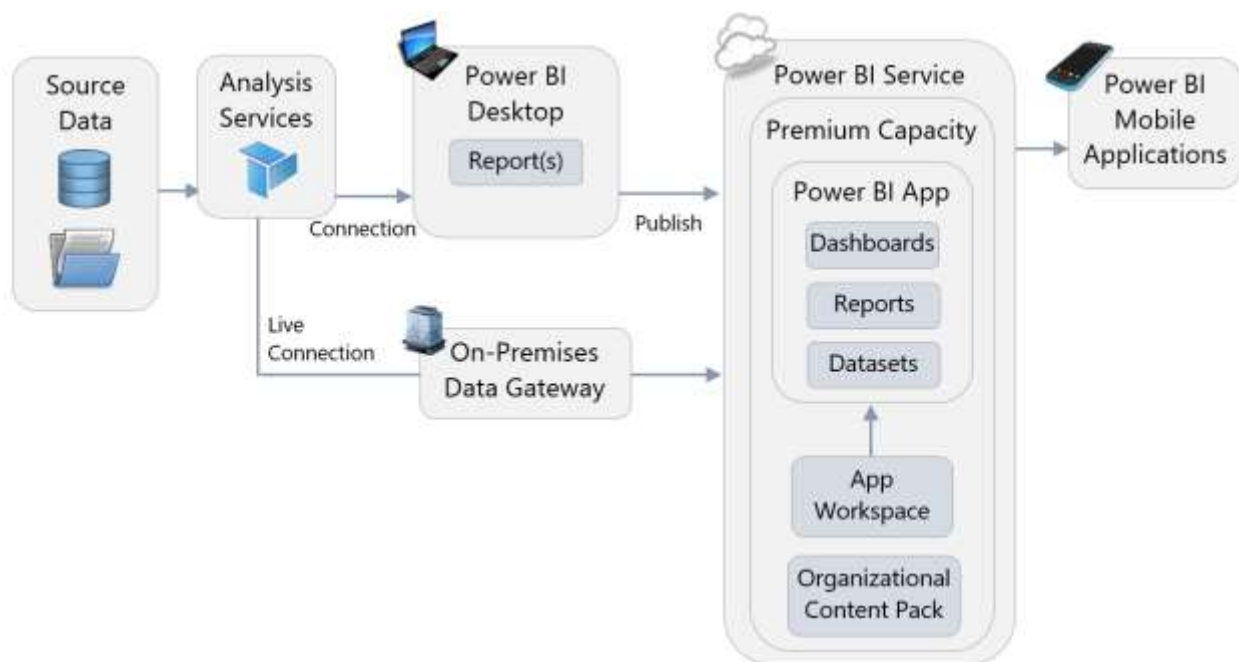
Excel. Same as above.

Power BI Service with Power BI Free. For personal use (i.e., no sharing or collaboration), the Power BI Service can be used with a Power BI Free license. Other features such as Analyze In Excel and Power BI Service Live Connection are omitted from this Personal scenario because they require a Pro license.

On-Premises Data Gateway: Personal Gateway. For refreshing datasets imported to the Power BI Service, a data gateway is required for secure access to data within the corporate network. The data gateway also supports Live Connection and DirectQuery operations. A gateway is generally not required when accessing cloud data. The Personal Gateway (which is typically installed on the user machine) is appropriate in personal usage scenarios.

Usage Scenario: Organizational Reporting

Corporate BI, or more traditional reporting scenarios, are commonly supported by the following components:



Potential Solution Components:

Power BI Desktop. For Corporate BI scenarios, it is more likely that common data storage such as a data warehouse or a semantic layer (such as SSAS) is available. In order to avoid replicating data into a new dataset, DirectQuery or Live Connection may be utilized. This behaves like other standard reporting tools which sends a query when reports are executed. Import mode is still an option as well, particularly when you need to combine data sources.

Power BI Service with Power BI Premium Capacity. Some larger, demanding workloads may justify the use of dedicated capacity, especially when there are a large number of readers or view-only consumers of reports and dashboards.

Power BI App Workspace. A Power BI App Workspace serves as a container in the Power BI Service for related reports, dashboards, and datasets. This is useful for organizing content by subject area and/or team, particularly helpful for delivering content to a large number of users.

Power BI App. A Power BI App is utilized to deliver published content to consumers.

Organizational Content Pack. Power BI Apps will ultimately supersede organizational content packs. In the interim, if personalization is a requirement, organizational content packs are an option. More detail of when to choose Apps vs. Organizational Content Packs is discussed in [Section 8](#).

Q&A Natural Language Querying. The Q&A feature allows a user to render a report by inputting verbiage such as: "Sales by salesperson for this quarter." This capability reduces the need for publishing many permutations of similar reports.

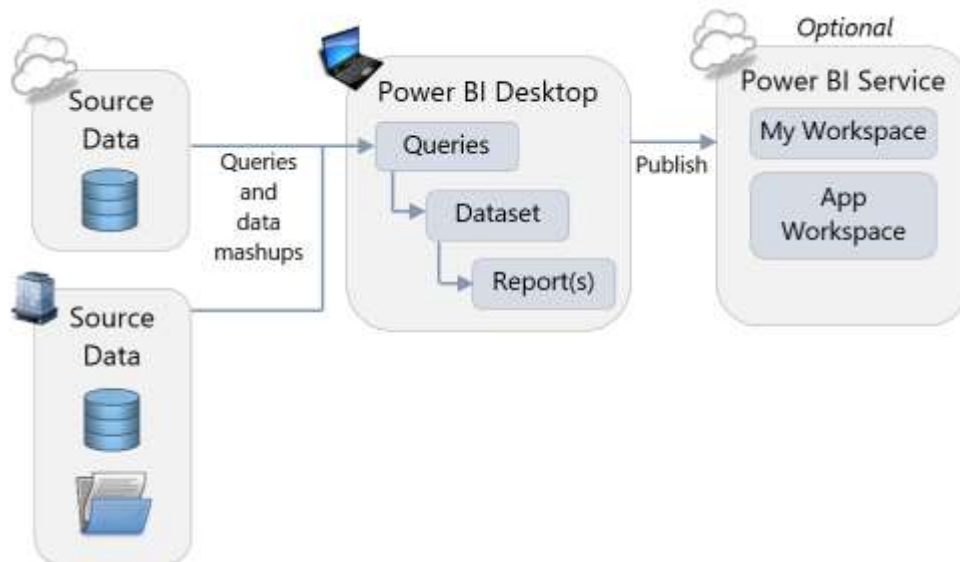
SQL Server Analysis Services or Azure Analysis Services. For Corporate BI scenarios, there is significant value in creating a semantic layer such as SSAS or AAS. Building out a semantic layer is useful for organizations which utilize multiple reporting tools beyond just Power BI.

Power BI Mobile Applications. Reporting 'on the go' via mobile apps is available for the reports and dashboards which reside in the Power BI Service.

Deployment of Development > Test > Production. App Workspaces do not yet support the ability to copy content from one workspace to another. However, it is a future feature which is coming to the product which will improve the ability to manage the lifecycle of Power BI solutions.

Usage Scenario: Prototyping Activities

Power BI can be used very effectively as a prototyping tool.



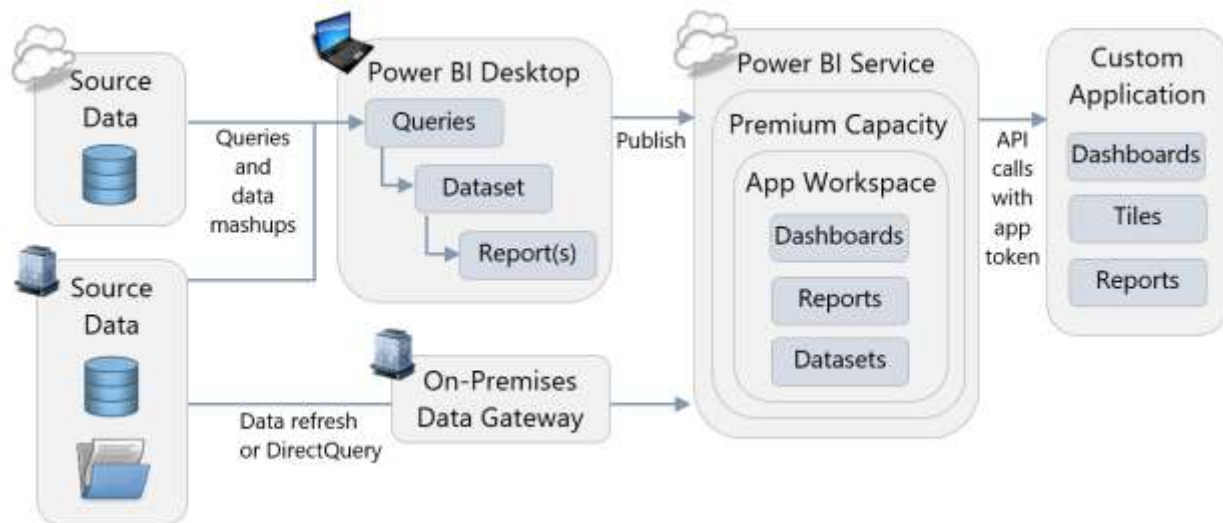
Potential Solution Components:

Power BI Desktop. The authoring tool can be used during interactive prototyping sessions. It is very useful for validating calculation rules, how the dataset should be designed, and how the reports should be displayed.

Power BI Service. Optionally, the Power BI Service can be useful when you need to share the results for feedback and decision-making purposes. Be sure to segregate prototyping solutions from other solutions, so its use is very clear. Power BI Apps are useful for this purpose.

Usage Scenario: Operational Reporting / Embedded Analytics

[APIs](#) can be used for embedding Power BI dashboards and reports into custom applications (such as a sales or customer application which was developed in-house). Embedded analytics like this can improve the workflow for functional users by not requiring them to navigate to another application to view reports.



Potential Solution Components:

Power BI Desktop. The authoring experience is the same in Power BI Desktop, even if you intend to deploy the reports within a custom application.

Power BI APIs with Power BI Premium. A Premium subscription is required in order to take advantage of read-only users. Otherwise, a Pro subscription is sufficient for users of the custom application.

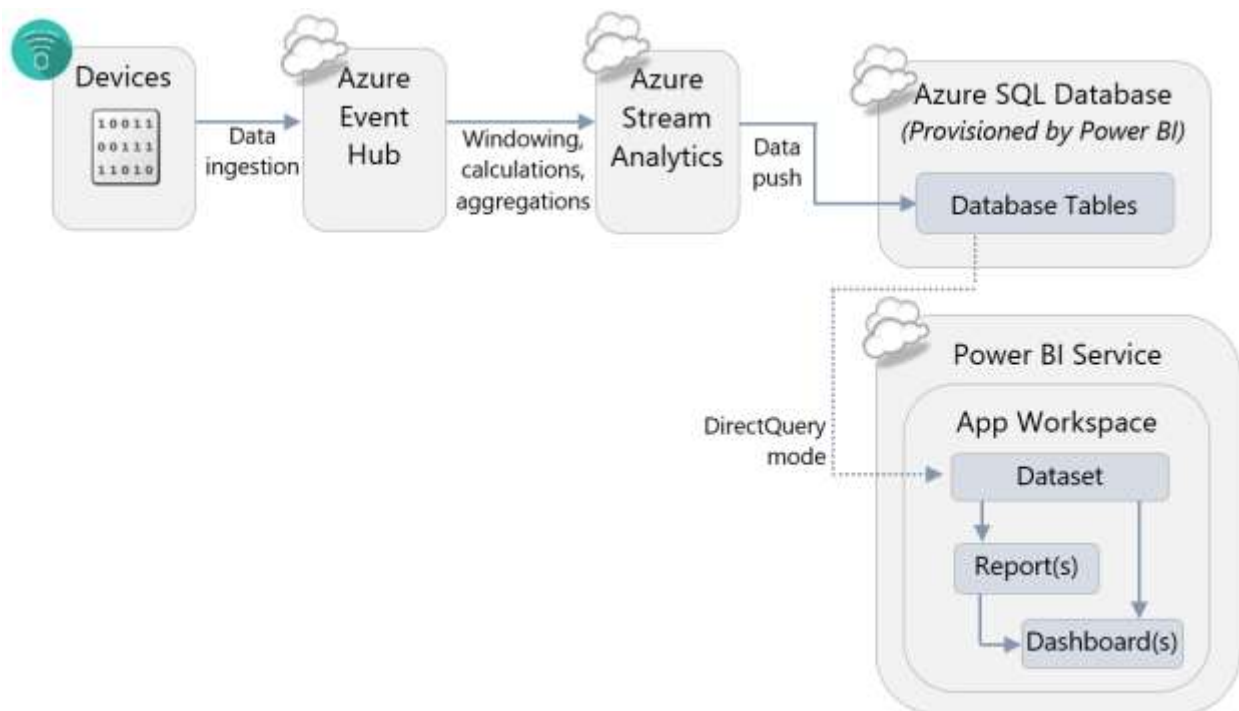
Power BI App Workspace. The objects do need to be published to the Power BI Service so the APIs can retrieve them and render in the custom application. The reports, dashboards, and datasets need to be in a Power BI App Workspace to serve as a container. (Note that publishing to an App, however, is not needed for this scenario; the App Workspace is sufficient since users will render visuals in the custom application rather than in the Power BI Service.)

On-Premises Data Gateway. For refreshing datasets imported to the Power BI Service, a data gateway is required for secure access to data that resides within the corporate network. The data gateway also supports Live Connection and DirectQuery operations. A gateway is not required when accessing cloud data. The On-Premises Data Gateway is strongly recommended over the Personal gateway for supporting APIs.

Usage Scenario: Streaming / Near-Real-Time Reporting

Streaming data solutions have been gaining in popularity because they allow the business to react faster. Streaming solutions are fundamentally different from other types of solutions in that data is typically pushed into Power BI. See [Section 5](#) for a discussion of streaming datasets, push datasets, and hybrid datasets.

The components required to support a streaming solution will vary depending on if you choose to use the Power BI APIs, Azure Stream Analytics, or PubNub. The following components are supported in a streaming solution which utilizes Stream Analytics:



Is an On-Premises (Non-Cloud-Based) Solution Required?

One feature of Power BI Premium is the ability to publish PBIX reports from Power BI Desktop to a Power BI Report Server. The following are reasons why you may want to consider utilizing a Power BI Report Server:

Complement the PBI Service

The Power BI Report Server is useful for customers who want to have an additional choice for where to publish certain content. For instance, highly confidential reports which must remain within the corporate firewall could be deployed to the Power BI Report Server whereas less sensitive solutions can continue to be published to the Power BI Service in order to utilize its broad feature set.

- ✓ Pro: Allows customers to have flexibility and choice.

- ✖ Con: Users may have to visit multiple places to access content.

PBIX distribution flexibility for highly regulated customers

For highly regulated customers who like using Power BI Desktop, but are not allowed to use the cloud-based Power BI Service, the Power BI Report Server offers an option for publishing and securing PBIX reports.

- ✓ Pro: Makes available an option for customers who must remain on-premises.

Build on existing investment in Reporting Services

There are many thousands of customers who have SQL Server Reporting Services implementations. Existing content can be migrated to a Power BI Report Server without any loss of functionality for existing RDL and Mobile reports.

- ✓ Pro: Expand usage of existing system to also include PBIX reports.
- ✓ Con: Power BI Pro licenses are required for those who publish PBIX reports (there is no such requirement for publishing RDL or Mobile reports). This requirement exists so that licensing requirements for publishing of Power BI content is consistent regardless of destination in the cloud or on-premises.

Section 3. Licensing Options

The licensing model your organization selects for Power BI significantly impacts deployment options and scalability possibilities.

Large-scale deployments introduce additional concerns around scale, performance, security, governance, control, and monitoring: all things typically associated with an enterprise business intelligence rollout. The Power BI licensing model is based on three options to simplify choices and better meet the varying needs of individuals, small groups, and large organizations.

With respect to terminology: a single organization will have a single Power BI [tenant](#), and to use the Power BI Service one or more subscriptions must be obtained for that tenant. A Power BI subscription may be free or it may require payment. A subscription has a start and end date, and a certain number of licenses are available within the subscription. Licenses are in turn assigned to individual users in the Power BI tenant. Once a license has been assigned to a user the user can access the Power BI Service. Subscriptions are purchased and licenses are assigned in the Office 365 Admin Portal.

Following is a summary of the three Power BI licensing options:

Power BI Free	Includes the ability to use the Power BI Service for personal use. No collaboration or sharing options.
Power BI Pro	Includes Power BI Free features + ability to share and collaborate with colleagues in the Power BI Service. Includes additional features such as usage metrics.
Power BI Premium	Complements Power BI Pro features by supporting higher data limits, enhanced performance, and ability to distribute to report consumers with Power BI Free licensing. On-premises report publishing. APIs for custom application integration.

These three options can be mixed and matched within an organization, based on the workload for individual users and groups.

Power BI Free

Power BI Free makes the web-based Power BI Service available for personal use. The Power BI Service is a multi-tenant online service which provides features such as scheduled data refresh, Q&A, and data alerts.

Power BI Free can be used one of two ways:

- A. As a standalone implementation of the Power BI Service for individual use. No sharing or collaboration features are available as part of Power BI Free. Neither dashboard sharing, nor App Workspaces, are available in the Power BI Free tier. A user in this tier will only be able to see 'My Workspace' in the Power BI Service.
- B. In conjunction with Power BI Premium. When content is deployed to Premium capacity, an unlimited number of Free tier users ("consumers") may view the content. Power BI Premium is discussed further on in this section.



Power BI Free is intended for individual personal use only.

For an enterprise deployment, Power BI Premium becomes more cost-effective when there are a large number of read-only users.

All data sources are available with the Power BI Free tier. The size of an imported dataset, also referred to sometimes as a cached dataset, is limited to 1GB per individual dataset (after the in-memory columnar compression has been applied). An overall storage quota of 10GB, per user, is applicable to Power BI Free. A user with a Power BI Free license can utilize the Personal Gateway for refreshing data up to 8 times per day.

The '[Publish to Web](#)' functionality, which can be used for publishing reports to a public website with no authentication, is also available with a Free license.

Power BI Free (as well as Pro and Premium discussed next) requires the user to sign in using a work or school e-mail address, also referred to as an organizational account. Because a tenant is

created for each organization behind the scenes, Power BI does not support consumer e-mail accounts such as outlook.com, gmail.com, yahoo.com.

Power BI Free, as the name suggests, costs nothing to use but nonetheless requires a subscription.

Power BI Pro

Power BI Pro introduces the following collaboration features within the Power BI Service:

- **Apps** – An App is a collection of related datasets, reports, and dashboards. Apps are very useful for organizing content by subject area, or segregating by security boundaries. Apps retain their identity by keeping content grouped together within an App Workspace.
- **Organizational Content Packs** – An Organizational Content Pack is similar to an App in that it is a collection of related datasets, reports, and dashboards. However, when a user connects to a content pack, the objects are extracted into My Workspace and lose their original identity. This facilitates the ability to personalize (save as).
- **Sharing** – Also known as peer-to-peer sharing, sharing capability allows a dashboard to be shared with particular individuals; the dashboard will be read-only to the recipient of the share. Sharing is most suitable for sharing with a very small number of individual users.



An App is the replacement for organizational content packs, which will be phased out over time. If personalization ability is not a requirement, use Apps instead of Organizational Content Packs if an App meets your requirements.

Don't confuse Power BI Apps with PowerApps, or with the Mobile Apps. Though similarly named, they are very different.

Note that a Pro license is required for any form of sharing. Put another way: both the person doing the sharing of content, as well as the person receiving the shared content, require a Pro license (with the exception of Premium read-only users discussed next).

In addition to Apps and sharing, the Pro tier also makes available the following capabilities:

- Analyze in Excel
- Power BI Service Live Connection
- Access to report usage metrics

Like the Free tier, the size of a dataset which contains imported data (aka cached data) is limited to 1GB per individual dataset. This 1GB limit is after the in-memory columnar compression has been applied (which frequently reduces data sizes between 3X and 10X). A data refresh schedule for an imported dataset is limited to a maximum of 8 times per day with a Power BI Pro license.

Like the Free tier, the Power BI Pro service tier operates as a multi-tenant service. In this type of shared capacity mode, Microsoft monitors performance and implements controls to ensure

against “noisy neighbor” scenarios in which any one customer can dominate too many system resources.

The monthly cost of a Power BI Pro subscription can be found [here](#). Power BI Pro licenses are also included with other Microsoft subscription packages, including Office 365 E5 subscriptions and Visual Studio Enterprise subscriptions.

The Pro licensing model becomes difficult cost-wise for companies when there are a large number of report consumers (who only view reports), and only a small number of report authors (who create content for others to consume). For that scenario, Power BI Premium may be a viable option.



If a Pro user adds a Free user to an App, or shares a dashboard with a Free user, there is no immediate validation of licenses, nor is an error displayed. When the Free user attempts to view the content, the Free user will be prompted to upgrade to a Pro trial at that time.

Need Power BI Pro for Groups

To access groups in Power BI, try Power BI Pro for 60 days, or contact your Office 365 tenant administrator to purchase Power BI Pro. [Learn more](#)

Try Pro for free

Upgrade account

Cancel

Power BI Premium

Power BI Premium is a new offering as of June, 2017. The Premium uses a new service architecture designed to address the workload demands of larger enterprise deployments.


Power BI Premium includes all Pro features, as well as:

- Support for **thousands of users**, allowing for a cost-effective way to deploy reporting to a large number of consumers.
- **Isolated, dedicated hardware capacity**, operating in single tenancy mode rather than multi-tenancy.
- **Unlimited number of free consumer user licenses**. Usage is bound by capacity level rather than the number of individual users.
- **Scale performance as the solution grows**, handled by a Power BI administrator who is designated as a capacity admin. Premium capacity can be scaled based on actual usage and performance levels.
- Ability to move content in and out of Premium capacity as **demand** dictates.
- **Power BI Report Server**, offering an additional choice for where to deploy Power BI reports, integrated with other report types (paginated and mobile reports).
- Use of APIs for **custom application integration**.

Power BI Premium is targeted at organizations which have a large number of users who consume content only (i.e., “readers”), and who have a high volume of query and refresh activity. Though some (or all) of the resources in a Premium tenant will operate in exclusive/dedicated hardware mode, end users will not be aware of nor see any difference in functionality.

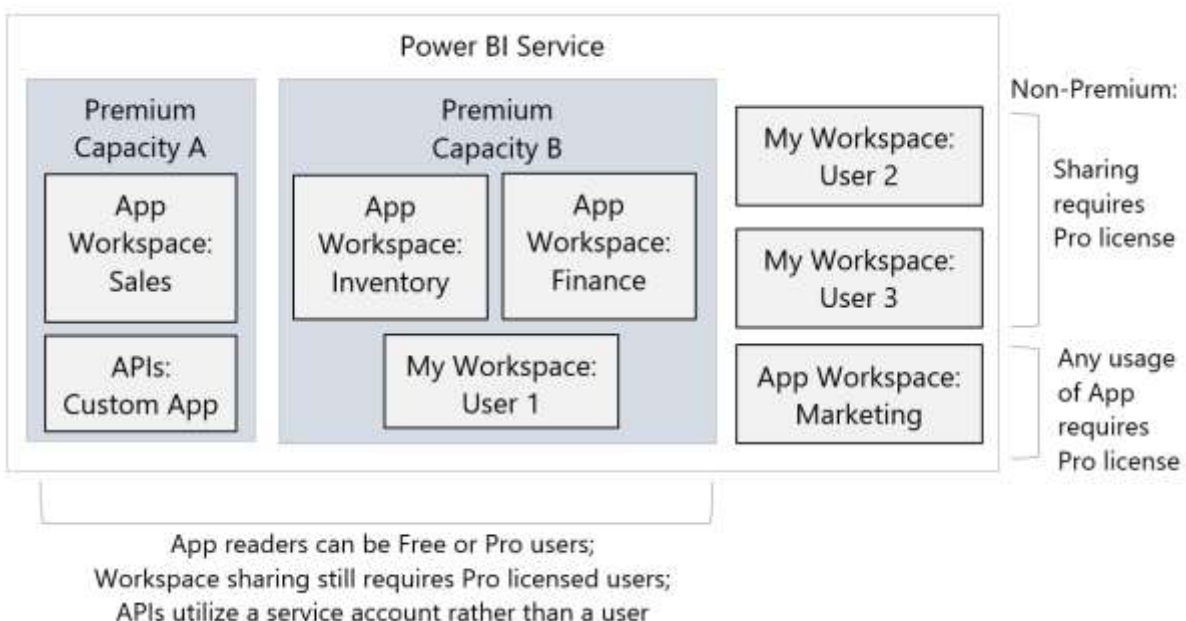
Premium Capacity

Premium capacity (capacity = virtual cores) works by assigning a workspace (either a user’s personal workspace, or an App Workspace, or even the entire tenant) to dedicated capacity in the Power BI Admin site. Alternatively, a user can be “whitelisted” so any workspace created by that user automatically goes into Premium capacity. Once a workspace has been assigned to dedicated capacity, an App published from it can then support an unlimited number of consumers without requiring a Pro license for each individual user. Pro licenses are still required for report developers.

 Read-only report consumers require, at minimum, a Power BI Free license so that the user can be authenticated to the system.

Capacity settings can be managed in the [Power BI Admin Portal](#).

Premium capacity can be applied to the entire tenant, or for strategically isolating particular App Workspaces which are deemed mission-critical, or which have specific performance requirements.



In the above depiction, Capacity A contains a Power BI App called Sales which has a demanding workload; Capacity A also supports a custom application which contains embedded reports

rendered via the Power BI APIs. The Inventory App, Finance App, and workspace for User 1 are in another capacity, whereas User 2, User 3, and the Marketing App remain in the standard multi-tenant service.

The capacity administrator can scale performance for the P series levels as the workload grows. Check the online [Power BI Premium Calculator](#) to help estimate the capacity level needed to support your usage patterns and to see the associated costs. Key considerations for capacity include:

- Number of concurrent users
- Query volume generated by users interacting with reports and dashboards
- Query complexity (particularly complex DAX calculations in measures)
- Number of concurrent data refresh schedules
- Competing, concurrent query and data refresh operations
- Data size of refreshed datasets, including efficiency of filtering and query folding operations
- Use of import (vs. DirectQuery or SSAS Live Connection which delegates much of the query workload to the source system)
- Volume of streaming datasets coming in continually
- Number of datasets which are actively queried (dynamic memory management offloads idle datasets from memory)
- Frequency of potentially system intensive operations (such as Quick Insights)



Monitoring of a large environment with varying usage levels can be challenging. Expect it to be an iterative process. Be sure to reserve a capacity level which will support peak load times, such as quarter end. Ongoing monitoring of Premium capacity is necessary.

Power BI Report Server

Power BI Premium also includes a Power BI Report Server subscription. Power BI Report Server is a server application designed to run on-premises that supports a subset of the functionality of the cloud-based Power BI Service. It is also closely related to, and offers all the functionality of, SQL Server 2017 Reporting Services. It provides the ability to deploy PBIX reports (authored in Power BI Desktop) to a Power BI Report Server alongside the paginated (.RDL) supported by SQL Server Reporting Services. This offers flexibility for customers whose security guidelines (or preferences) do not permit them to utilize the cloud-based Power BI Service.

Customers can acquire Power BI Report Server one of two ways:

1. Purchase of a Power BI Premium subscription. With Power BI Premium, you are permitted to also install Power BI Report Server on-premises with the same number of cores as specified in Premium capacity (ex: 8 v-Cores Premium capacity in the Power BI Service)

equates to an 8 core Power BI Report Server license). With this method, customers can implement a hybrid approach, and are permitted to publish to either the Power BI Service or the Power BI Report Server.

2. Via SQL Server Enterprise Software Assurance (SA). Customers with an existing SQL Server Reporting Services (SSRS) implementation may elect to migrate an SSRS server to become a Power BI Report Server for the benefit of integrating Power BI (.PBIX) reports alongside existing report (.RDL and .RSMobile) types. This migration is a one-way, irreversible, migration. With this method, customers may publish to the on-premises Power BI Report Server only (i.e., a corresponding Power BI Service subscription is not authorized).

Power BI Report Server does not require a separate SQL Server license unless you intend to use other components such as the SQL Server database engine or SQL Server Integration Services. A limited-use rights database engine comes with the Power BI Report Server for hosting the Report Server catalog and temporary databases.



The release cadence for updates to the Power BI Report Server will be much faster than historically experienced on-premises. This is considered a modern lifecycle policy, which other services support as well. The updates will not be auto-installed; however, you should plan to keep current as per the [support timeline for Power BI Report Server](#).

Power BI Report Server releases, approximately 4 times per year, will coincide with a specific Power BI Desktop release which will be optimized for the Power BI Report Server. PBIX file support in Power BI Report Server will be backward-compatible but not forward-compatible. For users who publish content to **both** the Power BI Service and the Power BI Report Server, there are two ways of handling this:

1. Stay on the version of Power BI Desktop which is compatible for both the Power BI Service and Power BI Report Server. New features will be available to users approximately every 4 months in this scenario. This is the recommended approach.
2. Run two versions of Power BI Desktop side by side. This is not recommended since it introduces additional complexity for the user to manage, but is possible for those who require new features in the Power BI Service to sync up with Power BI Desktop. If a PBIX from the newer version is inadvertently published to Power BI Report Server, it may not be able to render properly.

Comparison of Power BI Service vs. Power BI Report Server vs. SQL Server Reporting Services

Following is a summary of the key feature differences available when deploying to the cloud-based Power BI Service vs. the on-premises report deployment options (this is not an exhaustive

list of all features). SQL Server Reporting Services (SSRS) is included only for comparison to the Power BI Report Server, as SSRS is beyond the scope of this whitepaper.

Feature/Capability	Power BI Service	Power BI Report Server June 2017	SQL Server 2017 Reporting Services
<i>Publish and View Reports:</i>			
Publish and view RDL (Reporting Services paginated report)		✓	✓
Publish and view PBIX (Power BI report)	✓	✓	
Publish and view XLSX (Excel workbook)	✓		
Rendering of custom visuals utilized in a PBIX	✓	✓	
Use of native mobile applications (iOS, Android, Windows)	✓	✓	✓
APIs for embedding reports in a custom application	✓		
<i>Data Sources and Data Refresh:</i>			
Power BI report rendering from imported data	✓		
Power BI report rendering from SSAS Live Connections	✓	✓	
Power BI report rendering from DirectQuery	✓		
Data refresh schedule for Power BI imported datasets	✓		
Maximum size of an individual imported dataset (*=future size increases over time within Power BI Premium capacity)	1GB*		
Reuse of imported Power BI datasets (Analyze in Excel)	✓		
Reuse of shared datasets (Reporting Services paginated reports only)		✓	✓
<i>Additional Capabilities:</i>			

Feature/Capability	Power BI Service	Power BI Report Server June 2017	SQL Server 2017 Reporting Services
Create dashboard (compilation of visuals from one or more reports)	✓		
Pinning of certain visuals to a dashboard in the Power BI Service	✓	✓	✓
Report editing in the Power BI Site	✓		
Q&A natural language queries	✓		
Data alerts	✓		
E-mail subscriptions ("subscribe") for Power BI reports	✓		
E-mail subscriptions for RDL reports (Report server paginated reports)		✓	✓
Apps (associated with AppSource)	✓		
Organizational content packs	✓		
R integration	✓		
Get Insights (machine learning)	✓		

Power BI Embedded

Power BI Premium also makes APIs available for organizations who wish to embed Power BI reports, tiles, and dashboards into a custom application accessed by external users. This is referred to as Power BI Embedded. The existing, separate Power BI Embedded service will be deprecated at a future date; all new custom application integration should use the Power BI APIs within the main Power BI Service. The [functionality has converged](#) in order to make a more consistent and unified set of features available.

By associating these capabilities with Power BI Premium, the cost of consumption becomes a predictable fixed price rather than one based on usage, which can be very difficult to budget. ISVs should refer to separate guidance about pricing for Power BI Embedded, which is outside of the scope of this whitepaper.

Note that initial development of an embedded solution can occur with Pro licensing. The Premium tier will be required for the production deployment of the solution.

Comparison of Features Available Per Tier

Following is a summary of the key feature differences per tier (this is not intended to be an exhaustive list of all features).

Feature/Capability	Power BI Free	Power BI Pro	Power BI Premium
Create datasets and reports in Power BI Desktop	✓	✓	✓
Publish datasets and reports to Power BI Service	✓	✓	✓
Publish datasets and reports to Power BI Report Server		✓	✓
Create dashboard (compilation of visuals from one or more reports)	✓	✓	✓
Q&A natural language queries	✓	✓	✓
Data alerts	✓	✓	✓
E-mail subscriptions ("subscribe")		✓	✓
Export to CSV, Excel	✓	✓	✓
Export to PowerPoint	✓	✓	✓
Analyze in Excel (connectivity to a published dataset from Excel for creating reports based on an existing dataset)		✓	✓
Power BI Service Live Connection (connectivity to a published dataset from Power BI Desktop for creating reports based on an existing dataset- requires)		✓	✓
Access to all data sources (unrestricted)	✓	✓	✓
Scheduled data refresh via On-Premises Gateway in Personal Mode	Up to 8x/day	On-Premises Data Gateway Recommended	
Scheduled data refresh via On-Premises Data Gateway	Up to 8x/day	Up to 8x/day	Up to 48x/day
DirectQuery and Analysis Services Live Connection via On-Premises Data Gateway	✓	✓	✓
Use of streaming datasets	✓	✓	✓
Use of custom visuals from Office Store	✓	✓	✓

Feature/Capability	Power BI Free	Power BI Pro	Power BI Premium
Sharing read-only dashboard with individual colleagues (peer-to-peer)		✓	✓
Sharing read-only dashboard with an external user		✓	✓
Create and publish Apps		✓	✓
Create and publish Organizational Content Packs		✓	✓
Publish to Web (public report distribution)	✓	✓	✓
Power BI Mobile Apps	✓	✓	✓
Cortana/Windows integration for report rendering	✓	✓	✓
Maximum size of an imported dataset (*= <i>future size increases over time</i>)	1GB per file	1GB per file	1GB per file*
Maximum data storage quota	10GB per user	10GB per user	100TB per capacity
Multi-tenant service	✓	✓	
Isolated capacity			✓
Unlimited # of consumers (view-only users) of Apps			✓
Dedicated hardware capacity with scalability on-demand			✓
Use of Power BI Report Server for on-premises deployment scenarios			✓
APIs for embedding reports in an internal custom application		✓	✓
Power BI Embedded for embedding reports in an external custom application			✓
Third party SaaS Apps (from AppSource)	✓	✓	✓
Share queries in Azure Data Catalog		✓	✓
Integration with Office 365 Groups		✓	✓
Integration with Azure Active Directory	✓	✓	✓

Section 4. Source Data Considerations

All the data presented in reports and dashboards has to come from somewhere, and the nature of the data sources can have a significant impact on the rest of your Power BI project. In this section we look at which data sources can be used by Power BI and how they can affect the amount of time needed to develop reports, the reliability of your system, and how long it takes for reports to refresh.

Taking an Inventory of Data Sources

One of the first tasks on a Power BI project is to take an inventory of all the data sources that will be used by Power BI. It's likely that, when you speak to the various people in the organization who create reports, you'll be surprised at the number and diversity of the data sources currently being utilized. You will likely encounter 'shadow IT' as part of this exercise: systems and data sources that are not under the control of the official IT department and, in many cases, not even known to IT. These can range from Excel spreadsheets and Access databases, to accountancy systems running on a local PC, or cloud-based CRM services.



One mistake often made by Power BI projects run by IT departments is to ignore these 'shadow IT' systems and data sources, or to try to force their users to use officially sanctioned data sources instead. Doing so increases the risk of alienating users and potentially even project failure.

One of the advantages of Power BI as a self-service BI platform is that it makes it easy for users to build reports from 'shadow IT' data sources. Since users are already likely to be well aware of the problems associated with their current reporting methods (such as the amount of time it takes them to prepare reports, and the frequency with which these reports break or show incorrect figures), focus instead on showing them how Power BI can solve some of these problems. Once the user base has begun using Power BI, you will have a much greater visibility about how these reports are used and by whom, and it will be much easier to work with users to migrate their reports to utilize more 'official', centralized sources of data when appropriate.

The inventory of data sources created as a result of this exercise is useful not only when planning a Power BI deployment; it will be useful when that project goes into production. Having an enterprise-wide list of all of the data sources used for reporting and analysis and making that list available to everyone using Power BI will help people discover, gain access to and understand data sources they may not have known existed, and in turn increase the value of the reports they create. Consider using [Azure Data Catalog](#), a cloud-based service for cataloguing all the data assets in your organization, to store your inventory of data sources.

Data Source Limitations

While Power BI can connect to many different types of data sources, some types of data sources should be preferred over others. Even if Power BI can connect to all of the data sources in your

inventory it is important to assess how well Power BI works with these data sources. Some things to consider include:

Minimize use of Excel, csv, and text files when possible	<p>Excel workbooks, csv files and other text files where there is the chance that the format can change regularly as a result of human editing or mistakes often cause the most problems for Power BI. Excel workbooks in particular are frequently used as a data source for Power BI because of the flexibility and the ease with which users can edit the data in them.</p> <p>However, this flexibility also means that there is a greater chance that users will add new columns to tables, rename columns, delete columns and enter data in the wrong place (for example entering text in a column that should contain numbers). When unplanned changes happen, it is highly probable that Power BI reports will fail to refresh or that they show incorrect data. The extra time spent dealing with these issues and the damage they can do to the credibility of the project must be taken into account.</p> <p>Using tables or named ranges in Excel to identify different data sets within a single workbook can reduce the frequency with which these issues occur but these issues can never be completely avoided when working with files.</p>
Store files in a central location, accessible by all users of the Power BI solution	<p>If Excel workbooks, csv files or other text files are used as data sources, consider storing them in a place where they can be accessed easily by all users such as OneDrive for Business, a SharePoint folder, or a network file share. Specifically, avoid referring to a path such as c:\Users\UserName1\ because that path will not be valid for User2. This is an important consideration for any user who will need to be able to refresh the dataset.</p>

<p>Be aware of API limits</p>	<p>Many Software-as-a-Service (SaaS) solutions provide access to data via APIs. In many cases Power BI can connect to these APIs to retrieve data. However, it cannot do so in all cases: for example, certain types of authentication are not supported by Power BI.</p> <p>Even when Power BI can connect to an API, it may involve writing code in the M language, something with which end users and even experienced IT professionals may struggle. In the future it will be possible to develop Custom Data Connectors for Power BI in M. When this functionality is released, likely in late 2017, it will increase the number of data sources that can be accessed by Power BI.</p> <p>There may also be limits on how often the APIs can be called and how much data can be retrieved from them, and downloading data from them may potentially be slow due to throttling imposed by the service provider. The API owner typically documents limits on their website.</p>
<p>Know how to support SaaS solutions from AppSource</p>	<p>Datasets, reports, and dashboards from several popular SaaS solutions, such as Salesforce and Google Analytics, can be accessed using the AppSource website. They make it easy to get started quickly with analyzing data. However, these services use APIs behind the scenes and may suffer from the same restrictions on the frequency and volume of data which are discussed in the previous point. Also, the datasets which are created as part these SaaS solutions cannot be modified or added to; this can make it difficult to create the reports and dashboards you need.</p> <p>Expect that users who begin to get value from the SaaS solutions will often request the data to be integrated with other corporate data. This is a good thing as it informs the Corporate IT team what users find useful, but may add some new requests to the backlog.</p>

Minimize the load on source systems	<p>Querying data from source systems may, in some cases, cause problems for the data source itself. For example, many line-of-business applications use databases to store their data and it is possible to connect to these databases using Power BI. However, if users frequently connect to these databases and retrieve large amounts of data, or run complex queries (particularly with DirectQuery), this can result in the line-of-business applications becoming slow or unresponsive because they are experiencing contention or locking conflicts.</p> <p>Because it can be challenging for an OLTP to satisfy both transactional and analytical reporting needs, Power BI users may be prevented from accessing certain databases at certain times, such as during working hours, or even altogether, by the administrators of these systems. Techniques such as nonclustered columnstore indexes may help satisfy HTAP (hybrid transactional analytical processing) scenarios.</p>
Expect data refresh operations to take some time	<p>If Power BI reports are deployed to the cloud and the data sources they use are on premises, or vice versa, then bear in mind that moving large amounts of data into Power BI over the public internet for data refresh may take some time.</p> <p>The On-Premises Data Gateway, discussed in detail in Section 6, will need to be installed within the corporate firewall to allow Power BI to connect to on-premises data sources (personal mode is not recommended for use in enterprise scenarios). The installation of the On-Premises Data Gateway on a dedicated server will result in additional expense in terms of hardware and ongoing maintenance.</p>

Test data refresh in the Power BI Service regularly during development	<p>At the time of writing, there are limitations within the On-Premises Data Gateway that prevent certain types of complex M queries (used to load data into Power BI) from running successfully, even if these M queries succeed when the dataset is refreshed within Power BI Desktop. Scenarios where this happens include combining data from online and on-premises data sources, and using M functions to make multiple calls to web services.</p> <p>If reports use complex M code in their queries, it is strongly recommended to test whether data refresh works once the file has been published to the Power BI Service – a successful data refresh test in the Power BI Service should be verified on a regular basis during development.</p>
Utilize relational database sources when practical	<p>Relational databases, such as SQL Server, work best as data sources for Power BI. Typically a well-tuned database can handle large data volumes and deliver data very quickly. Power BI is able to generate SQL to request just the data that it needs when a dataset is refreshed (a process called “query folding”) which can make data refresh much faster.</p> <p>In addition, relational databases have many other features that can be used to avoid the kind of problems described above with using files as data sources; even the fact that the columns in a table can only contain one data type eliminates one of the most common problems associated with using Excel as a data source.</p>
Assume referential integrity when possible to improve performance	<p>When using a trustworthy centralized data source such as a data warehouse, you may be able to select “assume referential integrity” when working in DirectQuery mode. This will utilize inner joins, rather than outer joins, which may result in a more efficient query plan.</p>

Cleansing, Filtering, Transforming and Integrating Data

It is inevitable that the data used by reports will need some extra work on the part of the report designer to put it into the right format for Power BI. Examples of the kind of changes needed include:

- Filtering
- Handling dirty data, incorrect data, or errors
- Integrating data from multiple, inconsistent data sources



In some cases the amount of effort required to do this might be trivial. In other cases it might represent as much as 80% or 90% of the effort involved in building the report. If this is the case, consider where and how this work is done.

Power BI has some very powerful functionality for cleansing, filtering, transforming and integrating data in its Query Editor so this is not a question of whether Power BI is able to do the job. Instead, ask:

- Do users want to spend so much time and effort preparing data?
- Will cleansing, transformation, and integration of data be done consistently across all datasets?
- Should cleansing and transformation be handled closer to the source?

Creating a centralized data repository to store data that has already been cleansed is one way of reducing the amount of data preparation work that end users need to do in Power BI. If that repository is a relational database, such as SQL Server, then it can also solve some of the problems described earlier in this section to do with loading data from Excel files, web services, and line-of-business systems.



This is work that is best undertaken by someone with strong data management skills. This is a good example of where the IT department can effectively collaborate with the rest of the business in a self-service BI project.

A centralized repository like this is often referred to as data warehouse, although this has become a loaded term in that there are many competing methodologies describing how a data warehouse should be designed. Data warehousing has also gained a bad reputation in some organizations after previous BI projects spent years building them at great expense without delivering enough value to the business, even if these failures were typically the result of poor project management more than anything else. Whatever the name you use, though, there are obvious benefits in creating this kind of centralized repository, however simple or sophisticated it is. A centralized data warehouse can evolve and mature as the output of self-service activities.

Section 5. Power BI Data Storage Options

Once you have identified the data sources you intend to use in your Power BI reports, the next major decision you need to make is how Power BI should store the data it needs for its reports. In this section we will discuss the available options and the factors to consider when making this decision.

Data Storage Modes

There are four different ways that Power BI can connect to the data that is needed for display on a report:

- 1 Importing data into a dataset
- 2 Live Connection to Analysis Services
- 3 DirectQuery mode
- 4 Real-time streaming dataset

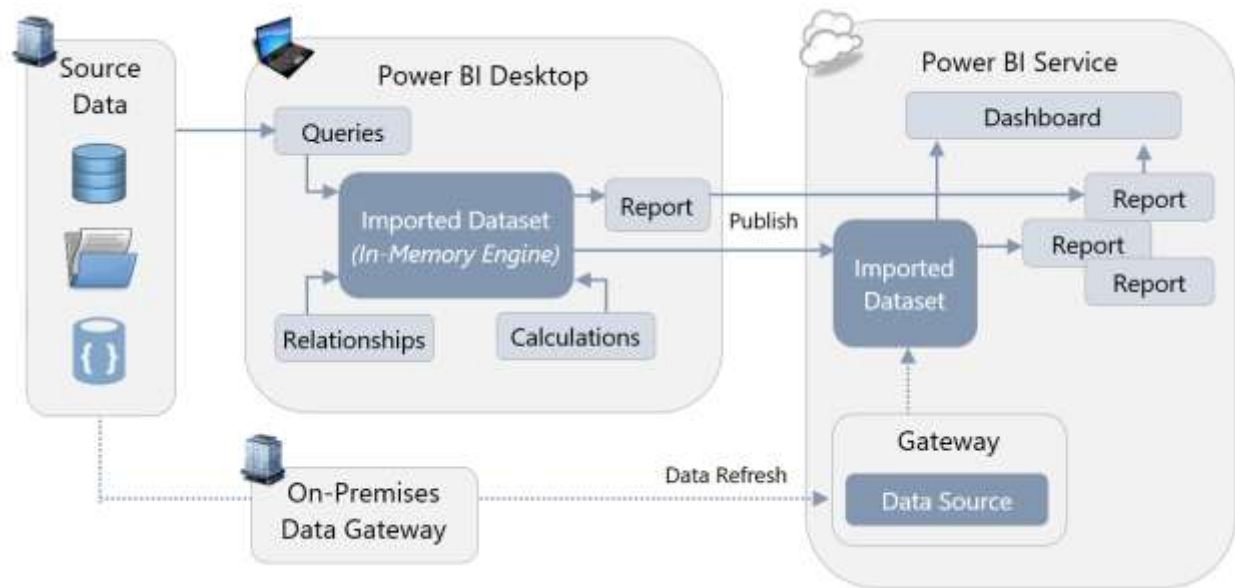
A report can only use one of these storage modes. Each of these storage modes has different advantages and disadvantages, and you may choose to use different storage modes for different reports. The following is a short summary of the technical details of each storage mode:

Option 1. Importing Data

The most common way of working with data in Power BI is to have it connect to your data sources and then import a copy of that data into a dataset stored within Power BI. Imported data is also sometimes referred to as being cached or embedded in the dataset.

In this scenario, Power BI uses its own high-performance, in-memory columnstore database engine to store the data. When reports and dashboards are viewed, Power BI queries its copy of the data stored within the dataset instead of going back to the original data source(s). The widest range of Power BI functionality is available when you work with imported datasets – when you use any of the other option listed below, some Power BI functionality is unavailable.

Below is a depiction of a dataset imported into Power BI:



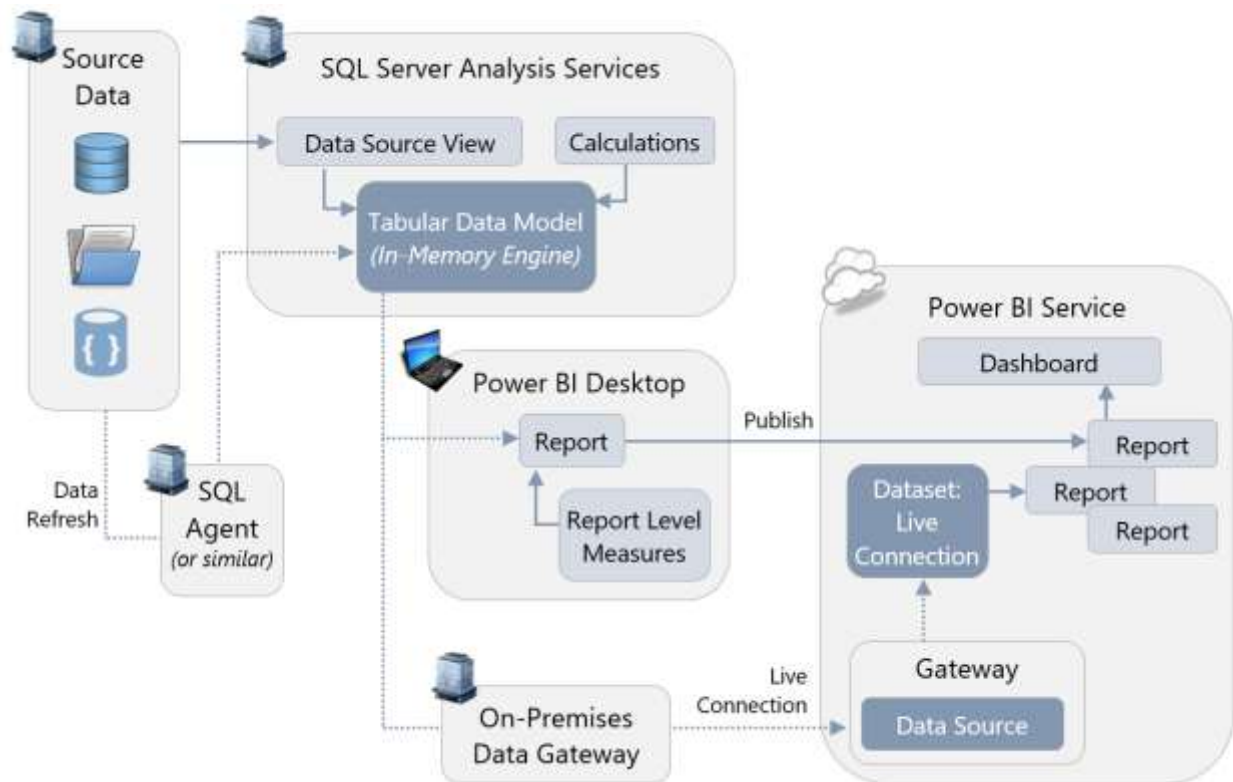
Refer to [Section 6](#) for more details on when the On-Premises Data Gateway is required for data refresh. [Section 6](#) also discusses when and where credentials are stored.

Option 2. Live Connections to SQL Server Analysis Services

The database engine that Power BI uses to store imported datasets is, essentially, the same database engine that runs inside SQL Server Analysis Services Tabular models. As a result Power BI can use SQL Server Analysis Services, or Azure Analysis Services, as a place to store the data needed for reports instead of using imported datasets. Any version and edition of SQL Server Analysis Services Tabular, and any tier of Azure Analysis Services, can be used.

SQL Server Analysis Services Multidimensional 2012 and 2014 is supported as well, provided the database is SQL Server 2012 SP1 CU4 or greater, and is either Enterprise or Business Intelligence Edition. With Analysis Services 2016 Multidimensional, Standard Edition may also be used.

Below is a depiction of a Power BI dataset which retrieves data from an SSAS Tabular model in Live Connection mode:



Note that Azure Analysis Services can also be used in place of SQL Server Analysis Services in the above depiction. In that scenario, an On-Premises Data Gateway is not required.

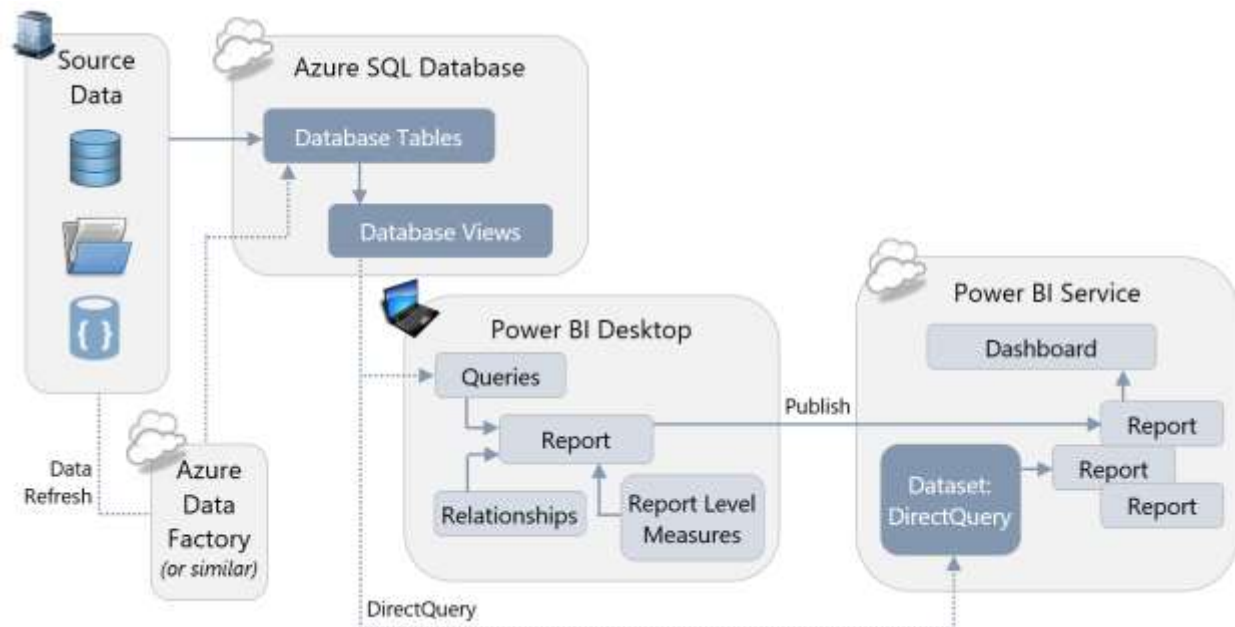
The above image indicates usage of [report level measures](#). This feature is available for SSAS Tabular Live Connections, or Power BI Service Live Connections.

Refer to [Section 6](#) for more details on when the On-Premises Data Gateway is required to support a Live Connection. [Section 6](#) also discusses when and where credentials are stored.

Option 3. DirectQuery Mode

The third option is to use a Power BI dataset running in DirectQuery mode. In this scenario, Power BI creates an internal dataset but this dataset does not store any data. Instead, when your Power BI reports and dashboards query the dataset, the dataset in turn runs queries against your original data sources. Only certain types of data source, such as relational databases, can be used in DirectQuery mode. See [here](#) for a full list of data sources supported by DirectQuery.

The following diagram shows a Power BI dataset which retrieves data from an Azure SQL Database using DirectQuery mode:



Refer to [Section 6](#) for more details on when the On-Premises Data Gateway is required to support DirectQuery. Section 6 also discusses when and where credentials are stored.

Note that DirectQuery and Live Connections to Analysis Services both involve sending a query to an external data source. However, the key difference is that with a Live Connection no dataset is stored inside Power BI – the Analysis Services Tabular model or Multidimensional cube to which you connect does the job instead. In DirectQuery mode there is a dataset created inside Power BI, it just doesn't store any data. DirectQuery mode cannot be used with Analysis Services data sources.

Option 4. Real-Time Streaming Datasets

Using streaming datasets is similar to importing data in many respects: in both cases data is stored within the Power BI dataset. However, when you import data into the dataset you *pull* data from a data source, whereas with streaming datasets an external application has to *push* data into the dataset. When data is pushed, the data may only stay in the dataset for a limited period of time. There are three varieties of real-time streaming datasets:

- 1 **Streaming datasets**
Streaming datasets store data in a transient cache, so the amount of time between data arriving in the dataset and any dashboard visuals that display it is kept to a minimum.
- 2 **Push datasets**
Push datasets actually store data in an Azure SQL Database instance that is provisioned by Power BI; when data is pushed into the dataset it is stored in this

database and the Power BI dataset connects to the database in DirectQuery mode. The amount of data that is retained in the underlying database depends on how the defaultRetentionPolicy property on the dataset is set: if it is set to None then data is loaded into the dataset until it reaches the 1GB limit for an import dataset; if it is set to basicFIFO the dataset can hold a maximum of 200,000 rows of data. When that limit is reached older rows are removed.

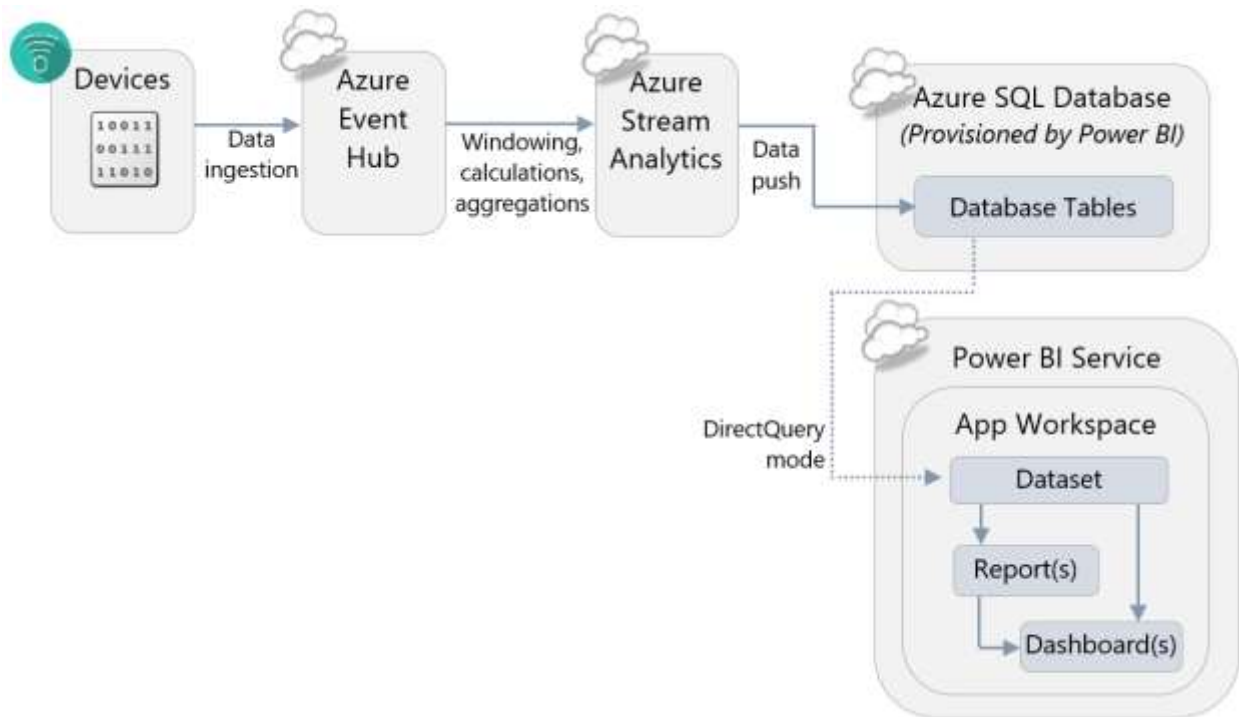
3 Hybrid datasets

Hybrid datasets are a combination of the two options above. Since a hybrid dataset consists of a Streaming dataset and a Push dataset you get the advantages of both, at the expense of duplicate storage.

Some of the key differences between a push dataset and a streaming dataset include:

	Push Dataset	Streaming Dataset
Destination for data imported to the Power BI Service	Dataset	Temporary Cache
Data history is retained	Yes (within dataset size limits in accordance with license)	1 hour only
Reports can be created	Yes	No
Dashboard tiles can be created	Yes	Yes
Can be used with Power BI APIs	Yes	Yes
Can be used with Azure Stream Analytics	Yes	Yes
Can be used with PubNub	Indirectly via Stream Analytics	Yes

Below is a depiction of a push dataset in Power BI:



Choosing a Storage Mode

The remainder of this section focuses on which storage option to use in which scenario, and is structured as a series of questions to ask about your project, your report, and your requirements. In many cases it may not be easy to choose the right storage mode for a particular report. If there is any uncertainty, a proof-of-concept involving creating the same report using different storage modes may be useful.

Question 1. Is Analysis Services Already in Use?

If you already have an investment in SQL Server Analysis Services and you have built cubes or Tabular models which contain the data that you wish to use in your Power BI reports, there is a strong argument to be made in favor of using Live Connections to those cubes or Tabular models in Power BI. There are two primary reasons why:

- ✓ It will greatly reduce the amount of development work needed for your Power BI reports because that investment has been made in Analysis Services.
- ✓ Data structure, naming, calculated values, and derived columns shown in your Power BI reports will match the data and calculated values displayed in other tools (such as Excel or SQL Server Reporting Services reports) that also use the same Analysis Services databases as their data source.

In this scenario, Power BI becomes an extra data visualization tool in your existing corporate BI deployment. There could be several reasons why you might choose not to use the existing Analysis Services cubes or Tabular models with Live Connections:

- ✖ Analysis Services cubes and Tabular models are usually designed and maintained by the IT department and not by the typical Power BI user, although it is possible to convert a dataset created in Power BI into an Azure Analysis Services model. Power BI users who want to have a significant level of control over the way that data is modelled may not wish to route requests for changes through the IT department. A centralized source such as Analysis Services is excellent for corporate BI scenarios. However, utilizing a centralized source may result in a slower pace to introduce changes as well as limitations to what customizations can be made to satisfy individual team needs. When connecting to an Analysis Services Tabular model (though not an Analysis Services Multidimensional cube) users can define their own report level DAX measures, but no other changes to the way the data is modelled are possible. It may therefore be appealing to create your own imported dataset in Power BI. However, there is always risk of introducing discrepancies and differences. We recommend carefully considering the trade-offs.
- ✖ If you are using a version of Analysis Services Multidimensional that is older than SQL Server 2012 SP1 CU4, then Live Connections with Power BI are unavailable. If you cannot upgrade SQL Server, perhaps for cost or other reasons, then you will have to look into other options.
- ✖ Live Connections perform the best with Analysis Services 2016+ Tabular instances. With older instances of Analysis Services Tabular, and especially with Analysis Services Multidimensional instances, you may find that query performance is not as good as using an imported Power BI dataset. We recommend doing a proof-of-concept to test performance early in the project.
- ✖ If you are using Analysis Services on-premises and Power BI is connecting to it through the On-Premises Data Gateway, the latency of querying data from Analysis Services to Power BI could cause unsatisfactory performance. This is less likely to be a major issue if you are using Live Connections to Azure Analysis Services. You may also want to consider using Azure ExpressRoute to create private connections between your on-premises data sources and the Azure data center where your Power BI tenant is stored to minimize the impact of latency.
- ✖ With Analysis Services Multidimensional instances, certain features in your cube are not supported fully or at all with Live Connections from Power BI. Examples include dynamic format strings on measures and calculated members on parent/child hierarchies. A full list can be found [here](#).

Question 2. How Much Data is Needed by the Report?

An individual Power BI dataset stored in import mode may only contain a maximum of 1GB of data after in-memory columnar compression is applied. Pro license users can only store a maximum of 10GB of data across all the Power BI datasets that they publish to the Power BI Service. If the amount of (compressed) data you have exceeds those limits, or will grow to exceed those limits in time, then you have three options:

- Purchase Power BI Premium and use it to store some or all of your imported datasets. This allows significantly more storage capacity, as high as 100 TB with the [P series](#). At introduction of Premium, the 1GB limit will still be in effect. However, in the near future there will be fewer limitations on the size of an individual dataset.
- Consider using Analysis Services Tabular on-premises or Azure Analysis Services to store your data instead, and then use Live Connections to Analysis Services from Power BI. The amount of data that can be stored in Analysis Services is constrained by the specifications of the server the instance is running on; at the time of writing, the largest Azure Analysis Services instance can store 400GB of data, four times more than the largest Power BI Premium SKU.
- Use a DirectQuery connection back to a relational database such as SQL Server or a cloud-based service such as Azure SQL Data Warehouse. The pros and cons of using DirectQuery are discussed later on in this section. DirectQuery may be the only feasible option for extremely large data volumes that are too large for Analysis Services to handle.



All three options may be more expensive than purchasing Power BI Pro licenses on their own, and for some small and medium-sized organizations this cost may be unacceptable. If this is the case, you may need to look at your requirements again and try to reduce the volume of data needed for your reports, for example, reducing the number of years or months of data that you import. Additional techniques are discussed in the Report Development section of this whitepaper.

Another performance aspect to consider is that if you have imported datasets that are used infrequently, Power BI may evict them from memory to make way for more frequently queried datasets. If evicted, the next time a user views a report that dataset will have to be loaded back into memory, making report performance slower for the user. In the future, datasets stored in Power BI Premium capacity will have the option to be pinned in memory so that they always reside in memory which will provide the best possible performance for datasets which are important yet less frequently used.

Question 3. How Frequently does Data Need to be Refreshed?

The data in the sources used by your Power BI reports will change at different rates. On one hand, the data in a corporate data warehouse may only be updated once a day, whereas sensors connected to machines in a factory may return thousands of readings every minute. As a result, you need be aware how often the data in your Power BI reports needs to be updated to enable timely decision-making.

For streaming data, which arrives at the highest velocities and needs to be analyzed in near real-time, you will need to use one of the varieties of real-time streaming dataset. If your data source is Microsoft's Azure Stream Analytics service you must use a Push dataset; if your data source is PubNub then you must use a Streaming dataset; if you are sending data to Power BI via the API then you can use either a Push dataset, a Streaming dataset, or a Hybrid dataset. These types of dataset cannot be created in Power BI Desktop, they can only be created programmatically via the API or via the Power BI web interface, making development more complex.



Most reports do not need to be updated this frequently. However, it is still important to determine how often the data does need to be refreshed because this can be an important factor in choosing between the remaining data storage options.

If you are using a Power BI Pro license without Power BI Premium, imported datasets can only be refreshed at most eight times a day. When you create a scheduled refresh in the Power BI web interface you can specify a time for the data refresh to take place, but there is no guarantee that that data refresh will take place at exactly this time – it may take place as much as 20-30 minutes afterwards. Microsoft has announced that, in the future, it will be possible to refresh imported datasets stored in Premium storage up to forty-eight times per day, and incremental refresh will be available as well. Refreshing only the changed portions of a dataset will significantly help refresh performance for larger datasets.

The refresh-related features that have been promised for Power BI Premium in the future are available today if data is stored in SQL Server Analysis Services and a Live Connection is used. Since you have full administrative control over an Analysis Services instance, you have very granular control over how and when data refresh takes place. Incremental refresh is also currently available in Analysis Services if you use partitioning (which is available for tables in Analysis Services Tabular and measure groups in Analysis Services Multidimensional on premises, but in both cases only in (a) Enterprise or BI Edition with older versions of SQL Server and (b) Standard tier of Azure Analysis Services). Configuring and optimizing data refresh in Analysis Services can become a complex task that Analysis Services developers spend a lot of time working on, but it is also well-documented and well-understood.

DirectQuery mode offers a compromise between real-time streaming datasets, imported datasets, and Live Connections. If your data resides in a data source which is supported by

DirectQuery mode, such as SQL Server, then the fact that Power BI queries this data when a user interacts with a report means there is no need to schedule data refresh: when the data changes in your data source the reports will reflect that change almost immediately. If your team is more comfortable with relational database technology versus Analysis Services, you may consider loading your data into a relational database such as SQL Server that is supported by Direct Query.

With both Live and DirectQuery connections, after you have published your reports, Power BI will cache some data inside the service to improve the performance of your dashboards. This cache updates every hour, so every hour you will notice the data in your dashboard tiles get refreshed. The frequency of cache refresh may be changed in the dataset settings (for DirectQuery or Live Connection datasets).

Latency between Power BI and the On-Premises Data Gateway, and between the gateway and the ultimate data source, can also have an adverse effect on performance if you are using DirectQuery against an on-premises data source or a Live Connection to an on-premises Analysis Services database. To minimize latency, try to co-locate your On-Premises Data Gateway and the data source in the same region as your Power BI tenant.



If you ask your end users how often they would like the data in their reports refreshed, they will inevitably say “as often as possible.” If you take them at their word this can make your life as a developer much more difficult than it needs to be. If you do some research into the actual business requirements, you might find that less frequent data refresh is perfectly acceptable.

Question 4. How Complex is the Dataset? Are any Complex Calculations Necessary?

A dataset is composed of a series of tables linked together by relationships, DAX (Data Analysis eXpressions) calculations and other metadata – this is also referred to as the data model. The complexity of the dataset, and the number and complexity of any measures used in it, will have a significant impact on the performance of your reports and therefore their usability.

Tuning the DAX used in measures and optimizing the data model of imported datasets can make a big difference to performance. If you are working with Live Connections to Analysis Services or DirectQuery you may also be able to tune the performance of those data sources, but there are some general remarks that can be made about complexity and the different options for storing data:

- As complexity of a dataset and calculations increase, including the use of complex row-level security, performance of a DirectQuery dataset will usually degenerate faster than the other types of datasets. This is because it becomes increasingly difficult for Power BI

to generate efficient SQL to retrieve the data needed by a report. By default, Power BI prevents you from using many DAX functions in your calculations when you are in DirectQuery mode because it cannot generate efficient SQL for them. This behavior can be changed by going to the Options dialog: on the DirectQuery pane check the “Allow unrestricted measures in DirectQuery mode” box to enable the use of more DAX functions inside measures and calculated columns. Be aware that doing this greatly increases the risk of poor report performance.

- If you are using a Live Connection to Analysis Services 2016 Tabular or newer, the version of the database engine used to store the data in Analysis Services is more or less equivalent to the one used to store an imported dataset in Power BI. However, for older versions of Analysis Services Tabular and for Analysis Services Multidimensional, Power BI generates less efficient DAX queries which will perform worse because these versions do not support the latest additions to the DAX language (referred to as SuperDAX). This performance difference will be more noticeable the more complex the dataset becomes. Older versions of Analysis Services may not be able to evaluate certain types of calculations as efficiently as the version of the database engine inside Power BI, which does comprehend SuperDAX.
- A Live Connection to an on-premises Analysis Services database may perform better than an imported dataset depending on the hardware the Analysis Services instance is running on, how the instance has been configured, and how well-optimized the Analysis Services database is. Important factors include the speed of the CPUs, the amount of memory on the server, and, in the case of Analysis Services Multidimensional, disk performance.

Question 5. How Many Concurrent Users Need to be Supported?

The number of concurrent users who will be using reports and dashboards will also have a significant impact on performance. Power BI imported datasets that are not stored in Power BI Premium capacity are subject to throttling by Microsoft so that the shared hardware that they run on is not monopolized by any one tenant. Therefore, reports and dashboards that use imported datasets not stored in Power BI Premium capacity will be unable to support an extremely large number of concurrent users. Power BI Premium capacity can support a much larger number of concurrent users since it has no throttling because it runs on dedicated hardware not shared by other customers,

To handle a large number of concurrent users, you will need to consider which operations need to be scaled out. Power BI Premium allows for scale-out for both front-end and back-end operations:

- Front-end operations include everything relating to the user experience: the web service, dashboard and document management, access rights management, APIs, uploads and downloads.

- Back-end operations include query processing, cache management, running R servers, data refresh, natural language processing, real-time feeds, and server-side rendering of reports and images.

Using a Live Connection to Analysis Services provides scale-out for only a subset of the back-end operations that Power BI Premium allows: only those operations that are concerned with query processing and data refresh. As a result, Power BI Premium is essential for scenarios where you need to support a large number of concurrent users, and you may choose to use Power BI Premium in combination with Live Connections to Analysis Services to get the best results. The question of how much capacity to purchase in Power BI Premium, or how to specify your Analysis Services server, is much harder to answer. This will depend on a number of factors, including licensing and support costs which are discussed in questions (7) and (8) below.

Question 6. Are there Legal or Regulatory Reasons for Storing Data in a Specific Location?

In some industries or countries there may be legal restrictions on what data you can store in the cloud, forcing you to use on-premises data storage options instead. There may also be legal restrictions on which database technologies can be used to store data: for example there may be a corporate security mandate to store certain data in Always Encrypted columns in SQL Server, resulting in the use of DirectQuery datasets in Power BI. If you think that these restrictions might apply to your project, seek legal advice.

More details on how and where Power BI stores data when different storage modes are used is available in the Power BI Security white paper available [here](#). You should also look at the compliance information and list of certifications for Power BI available in the Microsoft Trust Center [here](#).

Question 7. What Technical Skills are Present In-House?

If you are making a choice between using a Live Connection to Analysis Services versus using an imported dataset stored inside Power BI, bear in mind that an instance of Analysis Services will have ongoing support costs associated with it that Power BI on its own does not.

An Analysis Services instance, whether in the cloud or on-premises, will need someone to configure it and to monitor it; the overhead with Azure Analysis Services will be considerably less than with an on-premises instance of Analysis Services, but it will still require oversight by an administrator. Power BI Premium, in contrast, requires less administration.

The flip side of the previous point is that Analysis Services is more flexible and configurable while Power BI Premium is less flexible but much easier to manage. If you already use Analysis Services in-house and have staff on your team who have experience administering Analysis Services, then it is unlikely that using it with Power BI will present any extra concerns as you have

people who can take advantage of the extra opportunities it offers. However, if you do not have existing Analysis Services experience in-house then it may be appealing to choose Power BI Premium for your datasets until the more advanced capabilities of Analysis Services become a need.

More advanced architectures are possible when Live Connections to Analysis Services are used and you have the resources to implement them. For example, the following diagram shows a reference architecture created by Microsoft as part of a proof-of-concept for an enterprise-level customer:

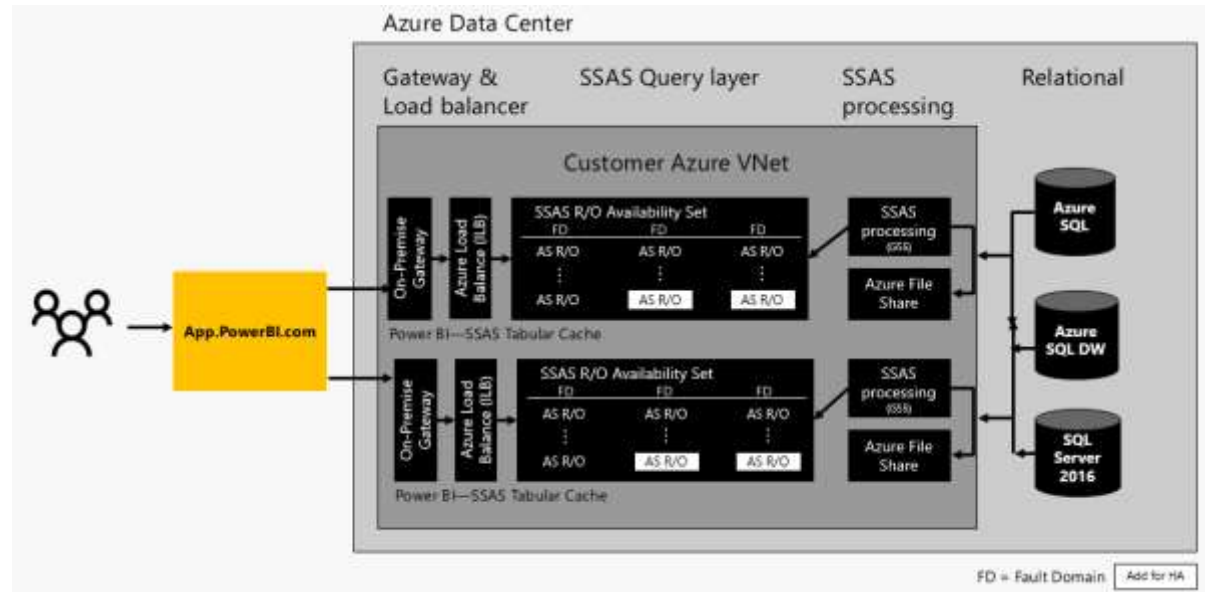


Image Source: Microsoft Power BI Team

In it, data is stored in a variety of cloud and on-premises relational data sources and then processed on dedicated instances of Analysis Services that are not used for querying, so processing does not affect the performance of reports. The queries run by reports are handled by a separate set of Analysis Services instances; a Fault Domain is used for fault tolerance and Azure Load Balancer is used to scale the queries generated by Power BI out over these Analysis Services instances.

Question 8. What Budgetary Constraints Need to be Considered?

No project has an unlimited budget, and you will need to understand what options are available for the funds you have available. Equally you will want to avoid over-specifying and paying for capacity and features that your user base does not need.

An important decision is when you should move from using a mixture of Power BI Free and Pro licenses to using either Power BI Premium, or Analysis Services, or both. This will depend on the number of users who need to create and publish reports and dashboards (who still require Pro licenses) and the number of users who only need to view those reports and dashboards (who

can then utilize Free licenses). You will find that there will be a point where it is most cost-effective to purchase Power BI Premium rather than continuing to purchase additional Power BI Pro subscriptions. Refer to the online [Power BI Premium calculator](#) to work out the exact threshold for your organization.

Typically, the recommendation is to start using the least expensive tier of either Power BI Premium or Azure Analysis Services, and then run a series of load tests on your reports to determine how well they will perform with your expected number of users. If you find that performance of the reports under load is unacceptable, then upgrade to the next tier and repeat the process until you achieve acceptable performance or you reach the limits of your budget. You should also pay close attention to usage once you have gone into production and be prepared to upgrade in the future as more and more reports are deployed to Power BI and reports are used more frequently.

Summary of Data Storage Recommendations

The recommendations in this section can be summarized as follows:

Only use real-time streaming datasets if:

- ✓ You need near real-time data to be displayed as soon as it is generated, and
- ✓ You are using a service such as Azure Stream Analytics or a custom application that can deliver data through the API to Power BI

Only use DirectQuery if:

- ✓ Your data needs to be refreshed more quickly than it can feasibly be loaded into a Power BI imported dataset or an Analysis Services database, or
- ✓ Regulatory concerns mean that your data cannot be stored in a Power BI imported dataset or an Analysis Services database, or
- ✓ Your data volumes are so large that the data you need cannot be loaded into a Power BI imported dataset or an Analysis Services database, and
- ✓ Your dataset is simple, consisting of relatively few tables and relationships, and
- ✓ You are using very few DAX measures or calculated columns in your reports, and
- ✓ You are using a data source such as SQL Server that supports DirectQuery mode

Use Live Connections to Analysis Services if:

- ✓ You already have all the data you need in Analysis Services Multidimensional cubes, or ideally, Analysis Services Tabular models, and
- ✓ Row-level security is required but managing RLS within individual Power BI datasets would introduce too much management overhead, and
- ✓ Your end users do not need to customize the datasets or work with data that is not currently stored in Analysis Services, and
- ✓ You have the resources in-house to support an instance of Analysis Services, and

- ✓ Your end users already use, or wish to use, Analysis Services with other tools such as Reporting Services or Excel, and
- ✓ You need a great degree of control over processing, monitoring, fault tolerance and other architectural features

Consider Azure Analysis Services over on-premises Analysis Services if:

- ✓ Some or all of your data sources already reside in the cloud, and
- ✓ Testing shows that the latency involved with using data from an on-premises instance of Analysis Services through the On-Premises Gateway in a Power BI report has a noticeable effect on report performance, and
- ✓ You have no legal requirements to store data on-premises, and
- ✓ You do not already have an on-premises instance of Analysis Services (or you purposely want a new Analysis Services instance), and
- ✓ You want to be able to scale up and down on demand

Use imported datasets stored in Power BI Premium storage instead of Analysis Services if:

- ✓ You prefer easy management over cost and fine control over configuration, and
- ✓ Your mix of report developers and read-only consumers means that it is more cost-effective to use Power BI Premium versus Analysis Services

Use Power BI Premium, regardless of whether you are using an imported dataset stored in Premium capacity or you are storing your data in Analysis Services, if:

- ✓ You have a large number of concurrent users and find that the user experience for operations like publishing reports, using the API and changing settings in the web interface is unacceptably slow with just Pro licenses, and
- ✓ Your mix of report developers and read-only consumers means that it is more cost-effective to use Power BI Premium versus just Power BI Pro licenses

Section 6. Data Refresh and the On-Premises Data Gateway

Data stored in Power BI must be refreshed for reports and dashboards to stay relevant. If you are importing data into a Power BI dataset then you must ensure that Power BI can connect back to its data sources for scheduled refresh to take place. Similarly if you are using a Live Connection to Analysis Services or DirectQuery mode then Power BI must be able to connect to the data source to run queries when reports are rendered. Since most enterprise data sources are on-premises and Power BI (unless you are using Power BI Report Server) is a cloud-based service, an On-premises data gateway must be used to allow Power BI to connect from the cloud to on-premises data sources. This section provides a brief overview of the different types of data refresh in Power BI and then looks in detail at how the On-premises data gateway can be configured and used.

Types of Data, File, and Cache Refresh in Power BI

There are several different scenarios where data stored in Power BI is refreshed:

Dataset Refresh

Description	A dataset refresh queries the underlying data source to perform a full refresh of data and metadata in an imported dataset. For DirectQuery datasets and Live Connections only metadata is refreshed.
How Refresh Is Handled	By the owner of the dataset: via Scheduled Refresh or Manually with the 'Refresh Now' menu item

Package Refresh (File Level)

Description	A package refresh synchronizes a .pbix or .xlsx file in the Power BI Service from OneDrive or SharePoint Online. It does not run a data refresh from the underlying source data—it only copies the dataset out of the file and into Power BI. It also reflects any changes made to the reports in the .pbix file.
How Refresh Is Handled	Automatic by the Power BI Service: approx. every 1 hour

Cached Tile Refresh

Description	Dashboard tiles display cached data, to ensure performance is optimal. The cached tile refresh updates the cached data displayed by dashboard tiles. This is useful after a data refresh has occurred to bring the dashboard tiles up to date.
How Refresh Is Handled	Automatic by the Power BI Service: after data refresh for imported datasets, or approx. every 1 hour (or as specified in the dataset properties) for DirectQuery and Live Connection datasets or Manually with the 'Refresh Dashboard Tiles' menu item

Cached Visual Container Refresh

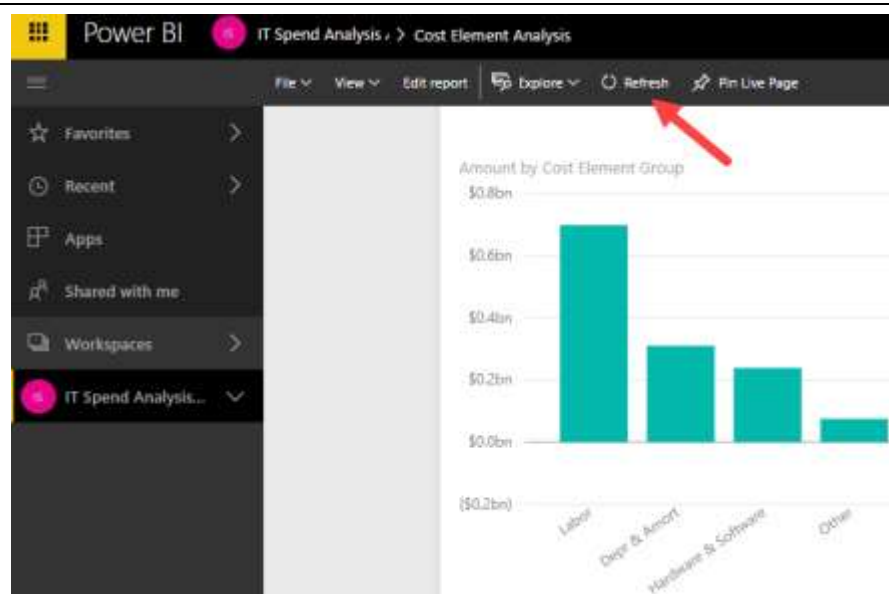
Description	The visual container refresh updates the charts, graphs, and tables which have been cached in reports for performance reasons. This is useful after a data refresh has occurred to bring the visuals up to date.
How Refresh Is Handled	Automatic by the Service

With respect to cached tiles and visuals, DirectQuery and Live Connection datasets will send queries back to the underlying source to refresh the cache. This can result in a lot of queries depending on the number of dashboards with pinned visuals.



Tip: The Refresh button when viewing Reports executes a cached visual container refresh as described above, which does not trigger a refresh of the underlying dataset.

Therefore, this Refresh button is generally only useful for DirectQuery or Live Connection datasets.



Introduction to the On-Premises Data Gateway

The **On-Premises data gateway** is an agent, installed within your corporate network, which securely handles data connectivity and data transfer between on-premises data and cloud services. A gateway is required for the following Power BI activities:

- Scheduled **data refresh** for on-premises datasets imported to the Power BI Service
- Use of **DirectQuery** to query data from the Power BI Service
- Use of Analysis Services **Live Connections** to query data from the Power BI Service

See [Section 5](#) of this whitepaper for a thorough discussion of the distinctions between imported datasets, DirectQuery, and Live Connections.

In addition to handling data communications, the gateway handles encryption/decryption, as well as compression of data transmissions.

At the time of this writing, Power BI content published to a Power BI Report Server supports Live Connections to Analysis Services only (both SSAS and AAS). This connectivity does not require implementation of a data gateway. Therefore, this section applies exclusively to the Power BI Service.



This section refers to LogicApps, Flow, and PowerApps. These are separate services which are considered part of the BAPI (Business Applications, Platform and Intelligence) organization at Microsoft. Although these services are beginning to introduce some integration points with Power BI, they are not directly related nor required for Power BI.

Gateway Terminology

The On-premises data gateway can be installed in one of two modes. One mode, formerly known as the Personal Gateway, is referred to as the **On-premises data gateway (personal mode)**. The other mode, formerly known as the Enterprise Gateway, is still referred to as the **On-premises data gateway**. The On-premises data Gateway functionality was previously known as the Data Management Gateway (DMG), but that term is no longer applicable to Power BI; it is, however still used by Azure Data Factory.

In this whitepaper, we will use the term 'On-premises data gateway' to refer to enterprise-level functionality. The term 'On-premises data gateway (personal mode)', or just 'personal mode', will be used to refer to that particular mode which offers a subset of functionality, intended for individual use.

Following is a summary of gateway support per service:

Connectivity for Services	On-premises data gateway	On-premises data gateway (personal mode)	Data Management Gateway
Power BI	✓	✓	
LogicApps	✓		
Flow	✓		
PowerApps	✓		
Azure Analysis Services*	✓ *		

Connectivity for Services	On-premises data gateway	On-premises data gateway (personal mode)	Data Management Gateway
Azure Data Factory			✓

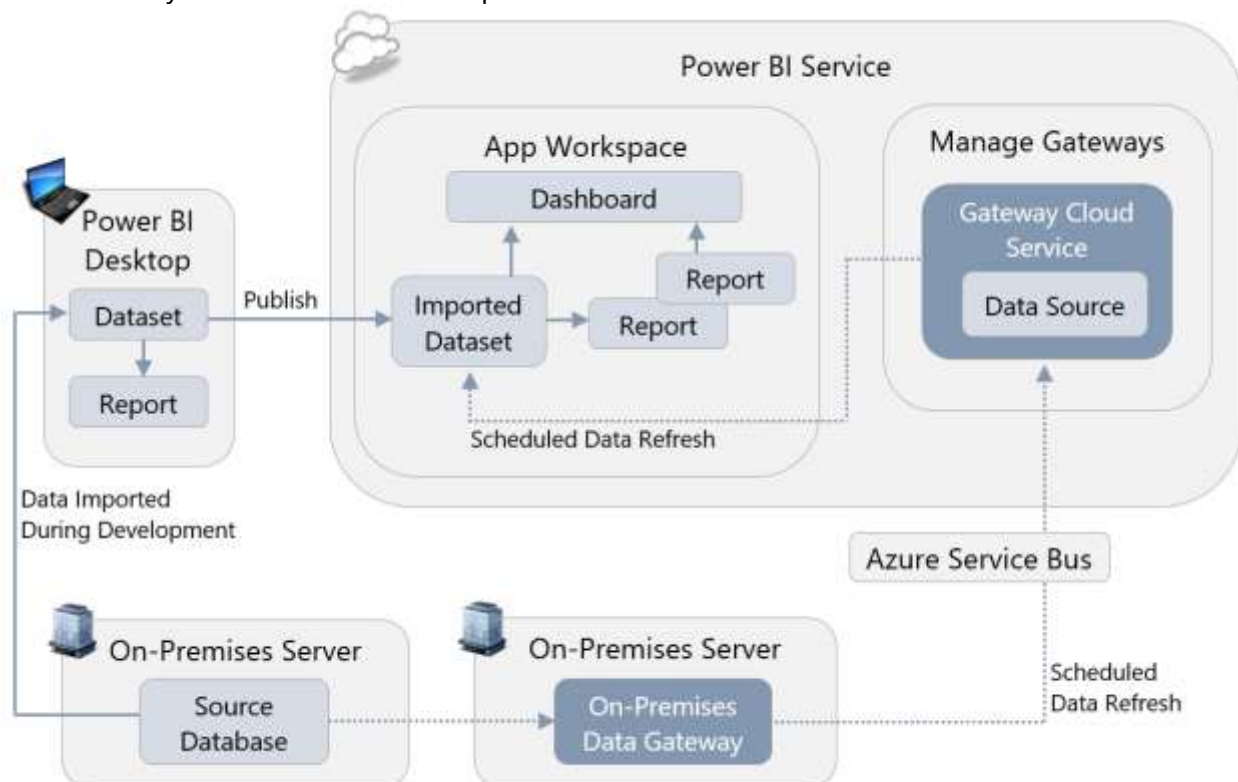
*The On-premises data gateway is targeted to support Azure Analysis Services (AAS) in a future release. At the time of this writing, the gateway for AAS is temporarily still a separate installation process.

Note that all the three types of gateways in the above table can co-exist on the same machine.

Mode 1: On-Premises Data Gateway

The On-premises data gateway is intended for enterprise-wide deployments. It may be installed on Windows 64-bit platforms only.

The following diagram depicts a scheduled data refresh which is executed via the On-Premises Data Gateway installed within the corporate network:



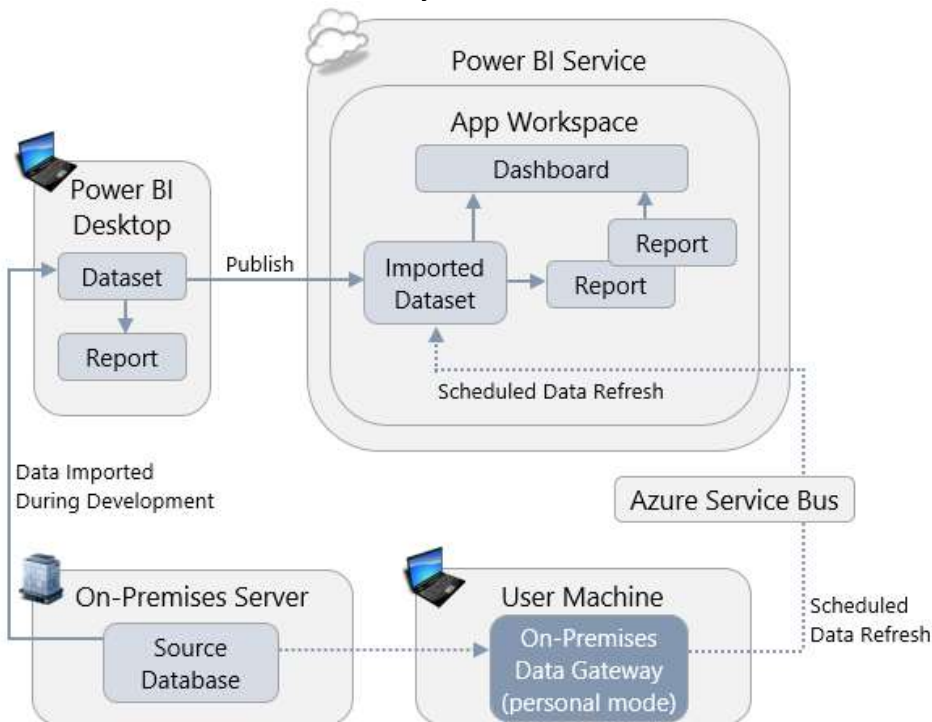
Advantages of the On-premises data gateway over personal mode include:

- ✓ The gateway and data source configurations are reusable for numerous datasets, across numerous users. If a source database is migrated to a new server, the configuration can be changed once in the Power BI portal (whereas every individual user who runs Personal mode will need to make this change).
- ✓ The gateway is intended to run on a server, rather than a user machine, which typically makes it more reliable for uptime and performance.
- ✓ The On-premises data gateway supports DirectQuery and SSAS Live Connections as well as data refresh for imported datasets (whereas personal mode only supports data refresh).
- ✓ The On-premises data gateway configuration is reusable among other BAPI applications, such as LogicApps, PowerApps, and Flow (personal mode supports Power BI only).
- ✓ Monitoring and tracking of usage is coming in a future release.
- ✓ Can set up privacy levels per data source

Mode 2: Personal mode

On-premises data gateway (personal mode) involves installing the On-Premises Data Gateway, in personal mode, on an individual user's machine, for purposes of scheduled data refresh. The Personal Gateway may be installed on 64-bit Windows PCs.

The following diagram depicts a scheduled data refresh of an imported dataset which is executed via the Personal Gateway:



Advantages of personal mode over the On-Premises Data Gateway include:

- ✓ It is fast and easy to set up, making it useful for prototyping and personal projects.
- ✓ It offers freedom for business users to get a data refresh accomplished if they do not have permissions to utilize or configure an On-Premises Data Gateway.
- ✓ Personal mode is capable of [ignoring data privacy levels](#) for the purpose of improving data refresh times (though it should be used with caution for sensitive data). See Section 9 for more details on data privacy levels.



An On-premises data gateway(personal mode) may be installed once per Power BI user account. Therefore, the same user is not permitted to install Personal mode once on their laptop and again on a server.

Summary of Features Per Mode

Feature/Capability	On-premises data gateway (personal mode)	On-Premises Data Gateway
Executes scheduled data refresh for imported data in PBIX	✓	✓
Executes scheduled data refresh for imported data in XLSX	✓	✓
Sends DirectQuery queries		✓
Sends Analysis Services Live Connection queries		✓
Integrates with other services: Azure Analysis Services, PowerApps, Flow, LogicApps		✓
Intended for use with enterprise-wide scenarios		✓
Intended for limited, personal use by one individual	✓	

On-premises data gateway (personal mode) is not intended for nor scalable for enterprise deployment scenarios. The remainder of this section focuses primarily on the On-premises data gateway.

When a Gateway is Required

A gateway is not required when working within Power BI Desktop. After a Power BI solution (.pbix or .xlsx) has been deployed to the Power BI Service, a gateway is required to access on-premises data sources for data refresh, DirectQuery, or SSAS Live Connection.

Typically, a gateway is not necessary to access cloud sources, with two exceptions:

1. Data sources which reside in an IaaS (Infrastructure-as-a-Service) virtual machine.
2. If Web.Page() is utilized in an M query.

PaaS (Platform-as-a-Service) services, such as Azure SQL Database, Azure SQL Data Warehouse, or Azure Analysis Services do not require a gateway for connectivity, nor do SaaS (Software-as-a-Service) solutions such as Salesforce or Google Analytics.

Process for Using the On-premises data gateway

Setting up the On-premises data gateway involves the following process:



**Periodic updates will be necessary for the gateway. There may also be multiple servers which act as a gateway in order to distribute the load.

Current On-premises data gateway Limitations

As Power BI continues to evolve and mature, additional enterprise capabilities are coming in future updates. At the time of this writing, following are known limitations with respect to the On-Premises Data Gateway:

End-to-End Monitoring	Visibility to all gateways, data refresh activity, and data sources in use is not currently available.
High Availability	There is no automatic or manual failover capability to a secondary gateway.
Load Balancing	Currently load balancing can be accomplished to a certain extent by manually distributing data sources across gateways.

Combination of On-Premises and Online Data Sources	If the queries for a dataset contain multiple sources, each of those sources must be contained in a single gateway for the data refresh to work. However, the On-premises data gateway can only refresh both online sources and on-premises data sources in the same operation for a limited number of data sources (any online data sources that can be added through the Manage gateways page in Power BI) . The current workaround for this is to temporarily use the Personal Gateway.
Unsupported Data Sources	Certain data sources, such as R, are not yet supported by the On-premises data gateway.
Migration of a PBIX or XLSX from Dev to Test to Prod	Currently there is not an easy way to change a data source for the purpose of migrating from development through a production environment.
Prevention of Gateway Creation	There is not currently a mechanism to prevent a gateway from being created, nor to blacklist servers.

Considerations for Using the On-premises data gateway

Data Gateway Installation File

There are two ways to obtain the installation file:

1 Power BI

The download link within the Power BI Service provides a single installer for both modes: the On-premises data gateway and the On-premises data gateway (personal mode).



2 Microsoft Downloads

The On-premises data gateway [available from Microsoft Downloads](#) is an installer that installs only the On-Premises Data Gateway (with no option for the Personal Gateway).

Naming Conventions

The following are suggestions for naming the gateway and the data sources.

Gateway Name

It is helpful to have some variation of "On-premises data gateway" in the name. This is useful with respect to logging, auditing, and troubleshooting on the server, particularly if the data

gateway resides on a multi-purpose server. You may also choose to include a specific purpose for the server in the gateway name, if applicable.

It is also useful to have the name of the server where the gateway is installed as part of the actual gateway name (though this is not necessarily where all of the related data sources are located).

When viewing the gateway configuration in the Power BI Service, including the server name in the naming convention will help you know exactly which of your on-premises servers this gateway relates to – otherwise it is very easy to lose track of those details. An alternative is to store the original server name in the description.

When setting a data refresh schedule for a dataset, the Power BI user does see the gateway name, department, description, and contact information. This presents an opportunity to add helpful information for the user when applicable (i.e., if the user should select a particular gateway for certain circumstances).

Data Source Name

The data source name should equate as closely as possible to the source itself, to avoid any ambiguity. For instance, the name of the database.

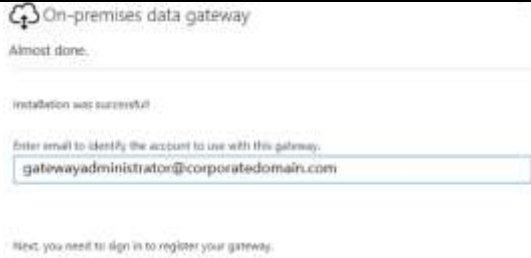

If applicable, other clarifications (such as Dev/Test/Prod) can be included in the data source name.

The name, type, server, and database cannot be changed after it is initially configured.

Learn more'. Below this are fields for 'Username' (BSQLVM1Dev_BI_Svc_PowerBI) and 'Password' (masked). An 'Advanced settings' section includes a 'Proxy (used only for the data source)' dropdown set to 'Organizational'. At the bottom are 'Add' and 'Discard' buttons."/>

Identities and Credentials Used by the Data Gateway

There are several types of identities associated with the data gateway:

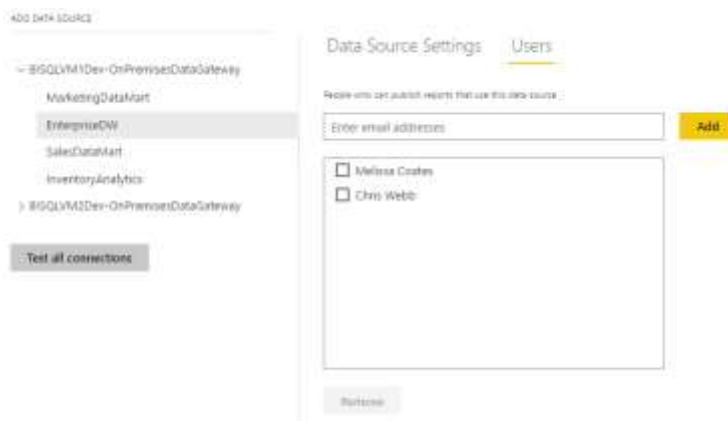
Gateway owner	<p>The gateway owner is specified when the gateway is initially installed on the corporate server. The e-mail account input during installation will be marked as the gateway owner, and will be specified as the contact in the Power BI Service.</p> <p>The extent to which you can utilize multiple BAPI services with a single gateway depends on binding to this organizational account. Therefore, it is recommended this be a service account rather than an account associated with an individual user.</p>											
Gateway service account	<p>When the On-Premises Data Gateway is installed, a PBIgwService local account is automatically set up on the local machine. The gateway service then runs in the context of this account, and is thus granted Log On As A Service permissions during installation.</p> <p>Some organizations may want to change this service account to run instead as a domain service account, especially if that is needed by a proxy server. This can be done in the On-premises data gateway server in the Service Settings pane.</p>	 <table><tr><th>Name</th><th>Description</th><th>Status</th><th>Startup Type</th><th>Log On As</th></tr><tr><td>On-premises data gateway service</td><td>The on-premises data gateway service</td><td>Running</td><td>Automatic</td><td>NT SERVICE\PBIgwService</td></tr></table>	Name	Description	Status	Startup Type	Log On As	On-premises data gateway service	The on-premises data gateway service	Running	Automatic	NT SERVICE\PBIgwService
Name	Description	Status	Startup Type	Log On As								
On-premises data gateway service	The on-premises data gateway service	Running	Automatic	NT SERVICE\PBIgwService								

Data source credentials	<p>Within the data source configuration in the Power BI Service, an authentication method, user name, and password need to be specified for access to the data source. These credentials are stored (using asymmetric encryption) in the On-premises data gateway, which also handles decrypting those credentials when the data source is accessed.</p> <p>A domain-based service account should be used for data source credentials rather than an individual user account which becomes problematic when password changes occur or upon terminations and transfers. The service account should have a password that does not expire (or is managed). Using a service account will also be helpful for logging and auditing purposes.</p> <p>This account typically requires only read permissions on the data source. One exception is when specifying an Analysis Services data source: SSAS requires the account to have administrator privileges on the SSAS instance. With SSAS, the account for the user interacting with the report is sent via the EffectiveUserName connection string property in the form of an e-mail address. The domain in this e-mail address must match what is in the Azure Active Directory associated with Power BI, or match a user principal name mapping.</p>
Administrators for the Gateway	Administrators of the data gateway are permitted to change configuration, and to add new data sources. It is strongly recommended there always be more than one administrator specified, in the event of employee terminations, transfers, or vacations.
Users in the Data Source	<p>The users specified for each data source are those who are permitted to:</p> <ul style="list-style-type: none"> • Schedule data refresh in the Power BI Service, or • Publish a dataset which uses DirectQuery, or • Publish a dataset which uses Analysis Services Live Connection <p>Consumers of the published content do not need to be listed in the Users pane.</p>

Associating Datasets with a Gateway

For a gateway to show up as an option when the user sets up data refresh, there are 3 criteria:

First, the user who is attempting to set up data refresh must be listed on the Users page of the data source within the gateway. This omission is a common source of confusion for users trying to set up data refresh.



Second, the server name and database name used in Power BI Desktop need to match what is configured in the Power BI Service. These names are not case-sensitive. For usability and readability, it is recommended that names be used instead of IP addresses. If SSL is configured for a source server, the server name for the gateway is required to be the fully-qualified domain name.

Third, each of the data sources referenced by a dataset need to be set up in the gateway. This is because a data refresh operation can refer to only one gateway.

User Interactions with the Gateway

The only awareness of a gateway that a typical business user (non-administrator) should have is when setting up data refresh. Ideally, data sources for commonly used data sources in the organization are pre-configured so that users working in DirectQuery or Live Connection mode have no issues. Suggestions to make this process as smooth as possible for the user include:

- ✓ Set up a data source only once. This is most common.
- or
- ✓ If the same dataset is purposely set up in multiple gateways (usually done to spread the load across gateway servers), configure the users for each data source to be distinct groups of users if possible. This will ensure each user sees only one data source, and will prevent the user from needing to guess which gateway to select.

If a user has set up personal mode for themselves, that is the first option listed for the dataset in the Power BI Service:

Settings for Product Analysis

This dataset has been configured by mncates@...com

[Refresh history](#)

Gateway connection

To use a data gateway, make sure the computer is online and the data source is added in [Manage Gateways](#). [Learn more](#)

☐ Use your data gateway (Power BI – personal)

← Personal Gateway

(offline, running on SS-240)

☐ Use a data gateway


← On-Premises Data Gateway - usually preferred when available

You have access to multiple data gateways. Choose the data gateway you'd like to use.

Status	Department	Gateway	Contact information	Description
<input checked="" type="radio"/> online	This gateway is...	BISQLVM1Dev...	gatewayadministrator@co...	This gateway is installed on BISQLVM1Dev. It provides connectivity to several data sources.

[Apply](#)

[Discard](#)

 The second radio button “Use a data gateway” refers to the On-premises data gateway and is usually the preferred option. This is a user training issue to encourage users to use the second listed option when available.

If the source is cloud-based (such as Azure SQL Database), the user can expect to see “Connect directly” instead since a gateway is not required:

Settings for Product Analysis

[Refresh history](#)

Gateway connection

To use a data gateway, make sure the computer is online and the data source is added in [Manage Gateways](#). [Learn more](#)

☒ Connect directly

← Cloud source which does not require a gateway

☐ Use a data gateway

[Apply](#)

[Discard](#)

Data source credentials

-sqlservers...database.windows.net

[Edit credentials](#)

When a dataset is using an existing On-premises data gateway, then data source credentials cannot be specified by the dataset owner. Rather, the credentials are inherited from the data source as configured in the data gateway.

Sharing of Data Gateway with Other Applications

The On-premises data gateway can be reused among Power BI and Azure Analysis Services, as well as other BAPI Applications (LogicApps, PowerApps, and Flow). This applies at the gateway level only; the data sources for each application can be independently configured. If one of the services begins to place more demand upon the gateway server, it may become necessary to introduce more than one gateway to support all operations.

Using One or More Data Gateways

One data gateway can communicate with multiple data sources. Depending on the volume of activity, a single gateway may be adequate.

However, if there are numerous types of data sources, and/or a lot of DirectQuery activity, and/or multiple BAPI services are involved, the load may become too much for a single gateway machine to handle effectively. In this situation, spreading data sources across multiple gateways is a useful scale-out technique.



If a Power BI Desktop file contains multiple data sources in order to perform a mashup, each of those data sources needs to be configured in a gateway for the scheduled refresh to succeed. Put another way, a scheduled refresh can only refer to one gateway. For this reason, a data source may end up being configured in multiple gateways. This scenario should be approached carefully so as not to be confusing for users.

You also may want to consider separating gateway which serve import models vs. DirectQuery models, in some circumstances.

Performance Impact on the Server Running the On-premises data gateway

Before discussing where to install the data gateway, it is important to understand the potential performance impact on a gateway server. The impact can vary greatly.

Generally, the gateway incurs minimal load in terms of CPU and memory when passing queries in DirectQuery or Live Connection mode. Network bandwidth is extremely important for passing volumes of data and/or handling frequent queries.

For simple queries which extract small amounts of data into imported datasets, the impact is typically minimal on the gateway server. When operations such as predicates, calculations, aliasing, joins, and type conversions can be pushed down to the source system using query folding, the gateway avoids incurring the load of performing those operations.

In all cases, strong network connectivity is important. There are situations when the gateway server will require more CPU and memory resources. When operations cannot be pushed down to the source system, then Power BI may need to perform filtering, matching, or other operations itself to prepare the dataset. The data privacy settings per source can also have a very significant impact on performance, as discussed in [this blog post, and can even prevent refresh from working completely](#). It is possible to stop the gateway from [checking data privacy levels completely](#) (applicable to the Personal Gateway only), but this option should be used with extreme caution.

Where to Install the Data Gateway

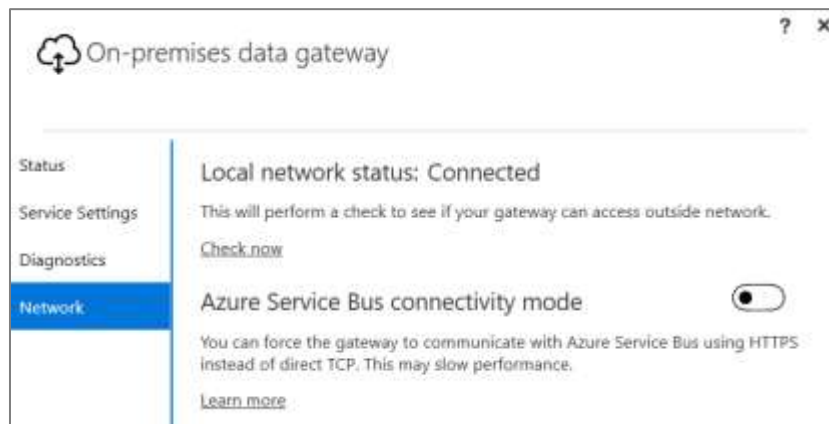
The following are some considerations for where to install the data gateway.

- Use of a machine dedicated to gateway operations vs. a multipurpose machine (such as a server with SQL Server already installed). If an existing server is currently underutilized, using an existing machine is often a good place to start.
- One gateway can be installed per machine. Gateways cannot co-exist.
- The machine should always remain powered on, with near full-time uptime.
- The gateway server should reside close to the data source to minimize network latency. As previously mentioned, the gateway can be on the same machine as the data source if that source machine can handle the workload. However, many organizations prefer a dedicated gateway machine, particularly if numerous data sources are involved.
- An Azure virtual machine can be a good option to install a gateway if on-premises equipment is not available. In this situation, the VM should reside in the same region as the Power BI tenant. It should also be as close to the data source(s) as possible to minimize latency and data egress charges.
- The gateway must reside in the same domain as the data source(s), or in another domain which is trusted.
- The gateway server requires strong network connectivity (preferably not wireless), high bandwidth, and consistent internet access. [ExpressRoute](#) is highly recommended.
- A potential starting point for a gateway server is 8 cores and 8GB RAM, as noted in [this blog post](#).

Firewall Ports

The On-Premises Data Gateway communicates with the Azure Service Bus via TCP on outbound ports 443 by default. It can also use 5671, 5672, or 9350 – 9354 as documented [here](#).

With TCP, the On-Premises Data Gateway communicates with the Azure Service Bus with a combination of IP addresses and domain names. Depending on your corporate network configuration, you may investigate [whitelisting Azure IP address ranges](#) or [usage of HTTPS mode](#) instead of TCP:



Changing the above setting can degrade performance, possibly severely, as HTTPS is not as fast as the default TCP communication with the Azure Service Bus. HTTPS communications will use fully qualified domain names (FQDN) only, rather than IP addresses.

Monitoring and Managing the Data Gateway

Data Refresh Duration

Scheduled data refresh jobs will time out if not completed within 2 hours. As discussed in [Section 4](#), it is very important to extract the smallest amount of data from the source that will meet your objectives. As data volumes grow over time, the duration of a data refresh will grow as well. Therefore, it is important to monitor job runtimes.

Pausing or Disabling of a Data Refresh

Scheduled refresh will be paused by the system if no users utilize any dashboard or report built on the dataset in 60 days. If any user revisits a dashboard or report built on the dataset, the scheduled refresh will resume.

After 5 failed scheduled refreshes, a scheduled refresh will be disabled by the system. When a data refresh schedule has been paused or disabled, an e-mail notification is sent to the owner of the refresh schedule.

Updates to the Data Gateway

Updates are not auto-installed for the On-premises data gateway. The gateway administrator will be soon notified in the Manage Gateways area of the Power BI Service when an update is available. It is highly encouraged to remain current with the latest data gateway version. The gateway is released on a monthly basis.

Monitoring Data Refresh Operations

There are various techniques for monitoring discussed in the documentation for [troubleshooting the On-premises data gateway](#). The following is a summary of commonly used techniques.

Available in the Power BI Service:

Data Refresh History

History is available on an individual dataset basis:

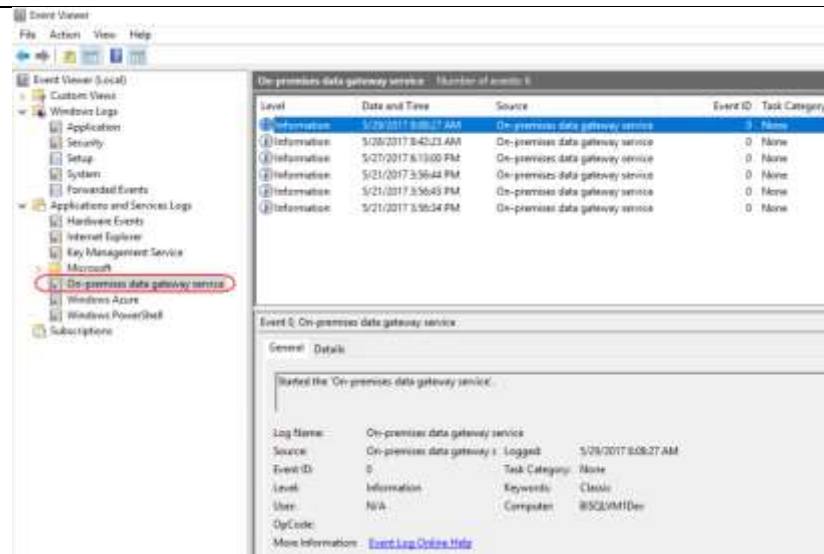
Refresh history

Scheduled OneDrive

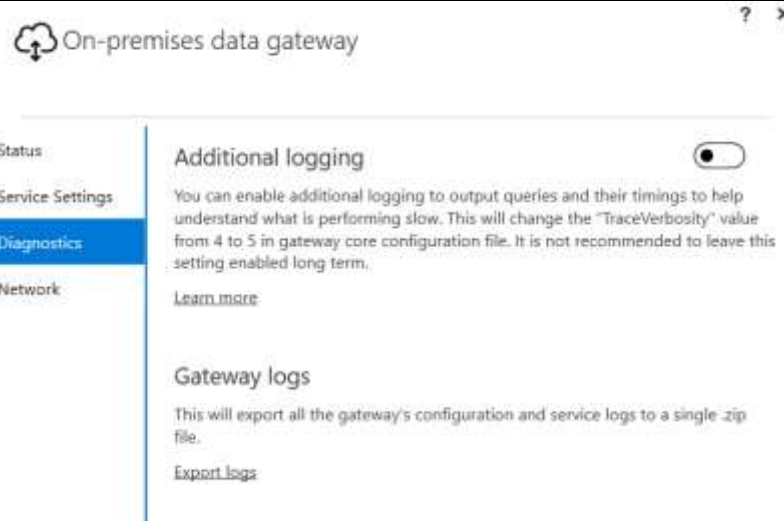
Details	Type	Start	End	Status	Message
	Scheduled	5/28/2017, 5:01:06 PM	5/28/2017, 5:14:12 PM	Completed	
	Scheduled	5/21/2017, 5:01:06 PM	5/21/2017, 5:13:04 PM	Completed	

Available on the Data Gateway Server:

Event Log Viewer



Verbose Logging and Log Exports



Logs

Gateway logs are typically available under:

C:\Users\PBIEgwService\AppData\Local\Microsoft\on-premises data gateway\Gateway*.log

Performance Counters	Relevant performance counters are documented here: https://powerbi.microsoft.com/en-us/documentation/powerbi-gateway-onprem-tshoot/#performance
----------------------	---

Data Refresh Limitations Based on License Type

The expectations for data refresh differ based on the type of Power BI license, as follows:

Feature/Capability	Power BI Free	Power BI Pro	Power BI Premium
Scheduled data refresh via Personal mode	Up to 8x/day	On-premises data gateway Recommended	
Scheduled data refresh via On-Premises Data Gateway		Up to 8x/day	Up to 48x/day
DirectQuery and SSAS Live Connection via On-Premises Data Gateway		✓	✓

With the dedicated capacity in Power BI Premium, the data refresh schedule should start at or very near the scheduled time. In the standard multi-tenant (non-dedicated) service, data refresh may start up to 60 minutes after the scheduled time due to balancing resources across customers.

Gateway Disaster Recovery

A recovery key is assigned (i.e., not auto-generated) by the administrator at the time the On-Premises Data Gateway is installed. The recovery key is required if the gateway is to be relocated to another machine, or if the gateway is to be restored. Therefore, the key should be retained where other system administrators can locate it if necessary. A corporate password keeper is very suitable for this purpose. More information about the data gateway recovery keys is available in the [Power BI Security whitepaper](#).

Section 7. Report Development Considerations

Choosing the Right Tool for Report Development

Power BI allows report development in three ways:



Using Power BI Desktop on a PC



Using the Power BI Service in a browser



Using Excel on a PC

It is likely that report developers will use all three methods at some point. To choose the right tool for the job, you should be aware of the strengths and weaknesses of each option.

Power BI Desktop

Power BI Desktop is the default choice for report development, and recommended in most scenarios. It allows you import data from over 70 types of data source, combine data from multiple data sources, clean and transform that data, create a dataset, add DAX calculations, and build reports all in one application. Factors to be aware of:

- Reports can be created in Power BI Desktop, whereas Power BI dashboards can only be created in the Power BI Service (because one dashboard can contain pinned visuals from numerous reports).
- Streaming datasets cannot be created in Power BI Desktop.
- It is available as a Windows desktop application. It does not run natively on other platforms such as Apple Mac.
- Power BI Desktop is updated every month. This is good news for users who wish to receive new features as quickly as possible. However, rolling out these updates for large number of desktops can require planning, and this is discussed in more detail in [Section 9](#).

Power BI Service

The Power BI web interface offers a subset of the functionality of Power BI Desktop for report development, in that it allows you to build reports and save them within a workspace. However, it does not allow you to import, clean or transform data, nor create nor edit datasets. Its strengths are the opposite of Power BI Desktop's weaknesses:

- Dashboards can be created in the Power BI Service, as well as reports.
- Streaming datasets can be created in the Power BI Service.
- Since it is a web-based application, it is available to all users regardless of what desktop operating system they use. A list of supported browsers can be found [here](#).
- As a web-based application, the Power BI Service always receives the latest updates and needs no specific maintenance or management by the customer. However, as a customer you cannot delay the introduction or control the timing of new features and updates.



There is one very important additional consideration: if a report has been created in the Power BI Service, rather than Power BI Desktop, you *cannot* download a copy of it for use with Power BI Desktop. This means that reports created in the Power BI Service must always be edited in the Power BI Service. Since it is not currently possible to use source control to store different versions of reports created in the Power BI Service this is a good reason to use Power BI Desktop for development instead.

Excel

Excel 2013 and Excel 2016 contain several features that are very closely related to Power BI: originally available as separately downloadable add-ins, Power Pivot became a native feature of Excel 2013 as the Excel Data Model, and Power Query became a native feature of Excel 2016 as the “Get & Transform” feature. The Excel Data Model uses the same database engine as Power BI for storing imported datasets; “Get & Transform” offers almost identical functionality to the Power BI Query Editor for loading data.

Using Excel in combination with these features to build reports provides a useful alternative to Power BI Desktop in certain circumstances. However, Excel does have several limitations compared to Power BI Desktop:

- ✦ The maximum size of an Excel dataset that can be published to Power BI is 250MB, compared to 1GB for Power BI Desktop. Therefore, you cannot work with large data volumes in Excel if you intend to publish them to the Power BI Service. Outside of the Power BI Service, the file size limit only depends on the individual user’s machine specifications.
- ✦ In many organizations users typically have the 32-bit version of Excel installed rather than the 64-bit version. The maximum size of dataset in the 32-bit version of Excel is around 1-2GB after compression. The 64-bit version of Excel can support much larger data volumes and generally performs much better than the 32-bit version.
- ✦ Some features that are present in Power BI Desktop, such as bi-directional cross-filtering on relationships between tables in the dataset, are not present in the Excel Data Model. Therefore, complex modelling requirements cannot be handled as easily in Excel.
- ✦ Excel reports that use connections to Analysis Services or published Power BI datasets (created either using the Analyze In Excel feature or via the Power BI Publisher Excel Add-In) are not interactive when they are published to Power BI.
- ✦ When opened in the Power BI mobile apps, Excel reports that use data from the Excel Data Model appear as static reports (they are fully interactive when viewed in a browser).
- ✦ The Power View add-in for Excel 2013, a very basic data visualization tool, is no longer being developed by Microsoft and is no longer visible in the Excel 2016 ribbon by default. The Power View add-in should not be used for report development. Rather, Excel-based report development should primarily focus on PivotTables, PivotCharts, and [cube formulas](#).

- ✖ Not all Excel functionality is supported when an Excel workbook is published to Power BI and viewed in a browser. For example, the 3D Maps functionality in Excel 2016 (previously known as Power Map in Excel 2013) is not available in a published workbook.

That said, there are many cases where Excel does offer a better experience for report design than Power BI Desktop:

- ✓ Excel has the full power of Excel formulas for creating calculations, in addition to what you can do in DAX, and of course a vast number of users are familiar with writing Excel formulas.
- ✓ Excel has a very mature (and still [expanding](#)) list of charts and graphs for visualizing data, and in most cases Excel charts have many more options for customization of visuals versus their equivalents in Power BI Desktop.
- ✓ Excel PivotTables are a much easier way of doing ad hoc data exploration than the Power BI matrix visualization.
- ✓ Excel [cube functions](#) allow for fine control over where values appear in a report layout, and have no equivalent in Power BI Desktop. Cube functions are extremely useful for creating financial-style reports, for example.
- ✓ If your users are creating reports for their own personal use, or are distributing and sharing their reports without publishing to Power BI (for example by saving them to a network file share), then the limitations on functionality for published reports mentioned above do not apply.

Many users will feel more at ease with Excel than with any other tool. This is an important factor that should not be ignored.

Best Practices for Report Design in Power BI Desktop

It is beyond the scope of this whitepaper to go into detail on best practices for report design in Power BI Desktop. Entire books have been written on the subject of creating and optimizing DAX calculations, for example. However, it is possible to offer some simple tips and tricks that will make your reports as responsive and easy to manage as possible.

Managing Report Authorship

Controls over who builds reports, where these reports are published, and how many reports are published, are essential if you are going to avoid thorny issues later on in your project. If you allow unrestricted report authorship you will find that duplicate reports and datasets are created, and this in turn means that:

- Maintenance of reports becomes time-consuming and difficult, for example when data sources change or calculations need to be altered.
- You reach the limits for how much data a user can publish using a Power BI Pro subscription, or even the available memory of your Power BI Premium capacity, more quickly.

- Scheduled refresh for large numbers of reports at the same time can put excess load on data sources, making both the data sources and the report refresh slow.
- Different data modeling decisions, variations in how calculations are defined, and even inconsistent column and measure names, mean that users have difficulty comparing the data displayed in different reports and are unsure which reports to trust.

This is not a recommendation to centralize all report authorship in the IT department because that diminishes many of the beneficial aspects of self-service BI. However, it is important to ensure that:

- ✓ All report authors are properly trained.
- ✓ Report authors follow your organization's documented best practices and standards when building reports.
- ✓ Duplicate reports and datasets are avoided.
- ✓ Reports are thoroughly tested before they are published.
- ✓ Reports always have owners who are responsible for maintaining them, and answering questions, after they have been published.



Many of these considerations relate more to the 'process' and 'people' side of the equation, rather than 'technology.' The [Power BI Governance and Deployment](#) whitepaper discusses many of these aspects.

Version Control for Reports

A lot of work can go into designing a report, and you should take care that this work is not lost by making sure that reports are safeguarded with a copy of each version when make change are made. The best way of doing this is through some form of version control. Although Power BI does not have any native features for version control or for interfacing with external version control systems such as Git, Microsoft Team Foundation Server (TFS) or Visual Studio Team Services (VSTS), most version control systems allow you to store files of any type in a repository. This includes the .pbix files which contain your queries, datasets, and reports. For example, if you are using TFS, you can create a folder in your Visual Studio solution, add your .pbix files to it and then check them in and out as changes occur.

Because some version control systems impose file size limits (ex: by default, GitHub places a limit of 100MB), you may also want to consider using [Power BI template](#) files (.pbit files) instead of .pbix files so that the files you are storing in version control are as small as possible. Power BI templates do not contain any data, and this makes them much smaller than .pbix files.

Business users may not feel comfortable using a fully-fledged version control system. In this situation, we recommend storing .pbix files in OneDrive For Business (or any similar file-sharing solution) because it saves historical copies of your files. The number of historical copies retained is typically configurable. This allows you to gain access to these previous versions, which is particularly helpful if a change needs to be reverted. Also, the file size limit for files in OneDrive For Business is currently 10GB which is more than adequate for most .pbix files.



Some users like to implement their own file versioning with file names (ex: Inventory Stock Prices 5-23-2017). This technique of changing the file name does not work well in Power BI because data refresh schedules, pinned tiles, and related reports are dependent upon a consistent file name.

Separating Dataset Development from Report Authorship

Creating a dataset requires a deep understanding of the data and how it should be modeled. Depending on the complexity level of data sources, calculations, cleansing, and relationships, this requires some level of technical skill.

In many cases business users just want to create their own reports and may not have the skill, time, or desire to build a dataset for that report. If users do create datasets associated with every report they build, this can result in the creation of many duplicate datasets which leads to the kind of [issues discussed earlier in this section](#). Therefore, it is important to separate the development of the datasets used by reports from the reports themselves wherever possible.

This can be achieved in three ways:

1

Use of Analysis Services

The first option is to utilize SQL Server Analysis Services to store your data. The pros and cons of this approach are discussed extensively in [Section 5](#) of this whitepaper.

2

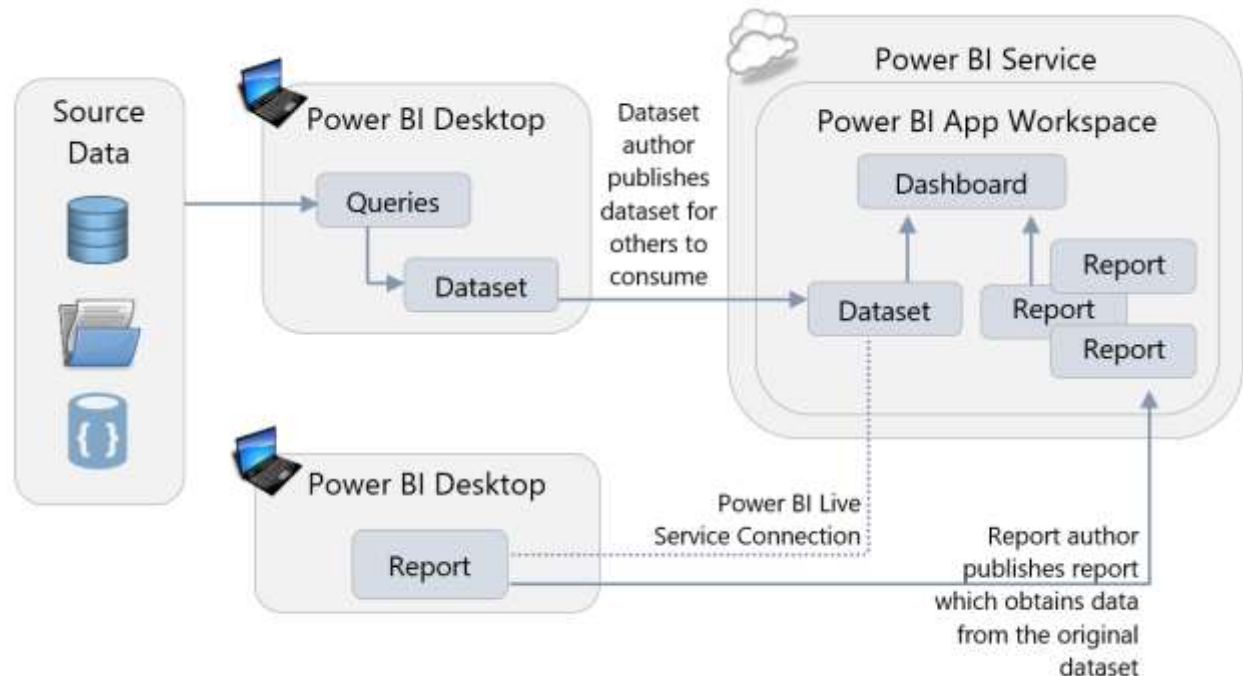
Use of Organizational Content Packs

Organizational Content Packs can be used to distribute shared datasets to report authors. This option is discussed in [Section 8](#) of this whitepaper.

3

PBIX File with Dataset Only

The final option is to create a dataset in a .pbix file without any reports in that file. This .pbix file can then be published to Power BI, creating a new dataset in the Power BI service, and then users can create new reports in Power BI Desktop that connect to the published dataset. This creates what is called a [Power BI Service Live Connection](#), and behaves in a very similar way to creating a Live Connection to Analysis Services – without, of course, the need for a separate server – and is more suitable for smaller workgroups. Alternatively, reports can be created directly in the Power BI Service using the web interface.



Datasets intended for reuse in this manner can be created by the IT department, or by subject matter experts. There are advantages to this approach, even in a pure self-service BI project:

- Different users with different skillsets can work on the dataset and reports independently.
- The overall number of datasets is reduced, meaning that less maintenance is needed, scheduled refresh will be faster, and less data is stored in Power BI.

Use Templates to Speed Up and Standardize Report Development

Building new reports from scratch can be time-consuming, and enforcing standards on those reports can be extremely difficult. The use of Power BI templates (.pbit files) can speed up report development significantly, and help with standardization. Instead of starting with an empty .pbix file, templates allow you to create new .pbix files which have:

- A corporate color scheme already in place (it is also possible to store a color scheme as a [Report Theme](#), a file that can be imported into Power BI Desktop).
- Corporate branding already applied to pages.
- Connections to commonly used data sources already created.
- [Parameterized queries](#) connected to a data source. This is particularly powerful because it allows your users to reuse queries that someone else has created yet customize the way these queries behave. For example, you can have a parameterized query that applies a date filter to a large table so only rows that fall in a given date range are imported. When a user opens a template containing the parameterized query, they are prompted to enter the date range for the data they wish to import.
- Commonly used DAX measures already created inside the dataset.

Centralize Data Source Connection Information

It is very likely that a report will contain queries that import multiple tables of data from the same data source; for example, you may need to import multiple tables or views from the same SQL Server database. The Query Editor in Power BI makes it easy to do this. However, each query that it creates contains the connection information (the server name and the database name in the case of SQL Server) for the data source. As a result, this makes it difficult to change the connection information in the future because it requires an edit to each query. An alternative is to utilize parameters in the Query Editor window containing the connection information before you connect to the data source. Then, when you do connect to the data source to import your data, use the parameters to supply the connection information to the queries instead of typing it in. With this in place, editing the connection information needs to be done only once, in the parameters, instead of having to edit each query.

Minimizing the Amount of Data Loaded

When you are importing data into Power BI (rather than using DirectQuery or a Live Connection) it is important to limit that data to only what is needed for the reports you are building. There are two ways to reduce the amount of data stored in a dataset:

1. Remove columns from source tables in the Query Editor.
2. Filter the rows in your source tables down to just the slice of data you need, either by using hard-coded filters in the Query Editor or by using query parameters.

Often the tables you are connecting to will have a lot of data in them that seems as though it *might* be useful at some point in the future. Indeed, if you ask your users, they will probably tell you to import it just in case it is needed. However, doing this can have some serious negative implications later on.

First of all, importing rows and especially columns of unnecessary data will increase the size of the dataset. This is particularly true for columns with a large number of unique values. Remember that there is an upper limit on the size of the dataset (with the exception of datasets in Premium capacity, though this is a future feature at the time of writing). In many cases importing unnecessary data can contribute to a dataset approaching its size limits. Also, the performance of a report is partly related to the size of the dataset so the larger the dataset the more chance that the report will perform poorly.

Secondly, the more data – and again, especially the more columns of data – that is imported, the harder it will be for report developers and users to find the data that is most important. This makes report development slower and more frustrating, and it increases the chances of mistakes where the wrong column is used in a visual or a calculation.

It is easy to add columns back to a table, or change a filter to load more data, at a later date. It is much harder to remove columns or add more restrictions to a filter after a report has been published because it is hard to know what data your users are using and what data they are not.

This is particularly true when using Analyze in Excel to browse the data in your dataset in an Excel PivotTable, or Power BI reports connect to a published dataset using a Power BI Service Live Connection. As a result, the best policy is to restrict the amount of data loaded as much as possible at an early stage and add more if and when it is necessary, rather than loading everything initially.

Reducing Dataset Size

In addition to minimizing the amount of data loaded, there are other ways to reduce the size of a Power BI dataset. If you are utilizing a data warehouse or similar data structure which is in dimensional format, then a date dimension table is likely to be readily available. In that case, you can disable the Auto Date/Time feature of Power BI Desktop which automatically creates date tables in the background for each date column in your dataset. Disabling this feature, which is done in the Data Load pane of the Options dialog in Power BI Desktop, can significantly reduce the size of a dataset.

Other techniques which can reduce dataset size include:

- Reduce the number of distinct values in a column. One way to do this is to eliminate the time element from date/time columns if time is not relevant. Or, if it is useful, extract time into its own individual column separate from the date.
- Round numeric values to reduce the number of distinct values in a column
- Avoid importing columns which contain values calculated from other columns in the same table.

Minimizing the Amount of Data Displayed on a Page

More is not necessarily better when it comes to report design – in fact the more visuals you put on a page, and the more data displayed in these visuals, the slower your report will be to render and the harder it will be for users to make sense of. Reports which are slow and difficult to interpret are unpleasant to use and will discourage people from using solutions created with Power BI.

As the number of visuals on a page increases, so does the number of DAX queries requesting data from the dataset, and inevitably each query that is run will increase the overall render time of the report. The more data that these DAX queries request, the longer they are likely to take to execute, and the longer it will take for Power BI to display the results in a visual. In some cases, such as certain map visuals, there is a maximum of 3,500 data points that can be displayed in a visual (after which Power BI will begin sampling points to ensure performance and interactivity of the visuals are maintained).



The performance of a report will depend significantly on storage of the dataset: imported, Analysis Services, or DirectQuery. This is discussed in detail in [Section 5](#).

As a result, you should always aim to minimize the number of visuals displayed on a page and minimize the amount of data displayed to maximize report performance. In particular, avoid “data dump”-style reports with tables containing hundreds of columns and thousands of rows, usually created when the user expects to export this data to Excel. If the user intends to export data to Excel, consider designing the report directly within Excel (using Get & Transform) to give them the data extraction they require. Alternatively, consider using slicers and filters in your reports to allow the user to select just the data that they need at any one time. Also, you may wish to set the slicer or filter at report publishing time, so the initial rendering of the report performs acceptably.

Custom Visuals

Custom visuals enable both third party developers, as well as the Microsoft Power BI team, to create many new visualization types for Power BI and enrich the available visualization stack. Developers can list their visuals in the [Office Store](#) to get wide public exposure both in the web-based store as well as in the integrated store in the Power BI service and Power BI Desktop.

While custom visuals provide rich ways for visualizing your data, you should be aware that they should be treated with caution. It is important to understand that custom visuals have access to the data that in Power BI datasets, and that custom visuals are not blocked from sending this data out of the Power BI service, so make sure that you are not using custom visuals from sources that you don’t trust.

Custom visuals created by Microsoft have been thoroughly tested and are safe to use. Custom visuals created by third party developers that are available in the Office Store have been through a validation process in Microsoft that ensures their quality and safety to some extent, but there is no guarantee that all code paths have been tested. Custom visuals downloaded from sources other than the Office Store have not undergone any validation process by Microsoft and you should be very cautious when using them – make sure you trust the developer and test the visual before allowing it to be used widely in your organization. It is recommended to have a policy for using and testing all custom visuals in your organization.

Custom Data Connectors

At the time of writing, Microsoft has just announced that in the future it will be possible to extend the number of data sources that Power BI can connect to by creating [custom data connectors](#). The same warnings that apply to using community visuals will apply to using custom data connectors: they may not be properly tested or supported, and as a result you should have a policy for using and testing them in your reports.

Creating Dashboards to Help Navigation

Dashboards have a role to play in helping users find metrics that would otherwise be buried in reports. Pinning visualizations to a dashboard allows developers to provide an “at-a-glance”

view of the most important metrics. A user can treat a dashboard as something like a start page or executive summary: if all looks well the user does not need to look at the underlying reports, but if necessary the user can click on a visualization and navigate to the report it comes from to see it in context.

Also, because you can pin visualizations from multiple reports, from Excel workbooks, and even from SQL Server Reporting Services reports and the new Power BI Report Server, you can use dashboards to create a unified view of metrics from different reports created in different tools.

Section 8. Collaboration and Sharing

Using Apps and App Workspaces for Collaboration and Sharing

Two Types of Workspaces

When a report author has finished developing a report in Power BI Desktop, he or she can publish it to the Power BI Service. When a report is published, it is published to a workspace. There are two types of workspaces in Power BI:

My Workspace	Every Power BI user has a workspace called My Workspace which is intended purely for personal use. Publishing a report here can be useful during development because it allows you to test what your report looks like in a browser or on a mobile app. Any dashboard sharing done from My Workspace should be minimal, and with a very small number of recipients, if it is done at all.
App Workspace (formerly known as Group Workspace)	App Workspaces are shared workspaces where multiple users can collaborate on report development. For sharing purposes, it is recommended to publish the relevant datasets, reports, and dashboards to an App Workspace. Reports and dashboards created in an App Workspace can then be published into a Power BI App when development is finished. The published App is where other users can then view these reports and dashboards by installing the App. When used in conjunction with Power BI Premium, the read-only users (who only need a Power BI Free license) don't need any permission to the App Workspace – just to the App.

The process of setting up an App Workspace and creating a Power BI App is [well documented](#) and outside the scope of this document. What is in scope is the question of how Power BI Apps should be used within an organization.

What Should an App Workspace Contain?

Different organizations will use App Workspaces and Apps in different ways depending on their requirements. What is important is that an organization should have a clear and consistent policy on how App Workspaces and Apps are used and what they contain. Following are some ideas on how Apps can be used:

Scope:	Per Department	Per Report	Per Business Question
Example:	Finance Reports	Sales Summary	Quarter-End Reports

Per Department

Create one App for each department in your organization, and put all the reports and dashboards needed by the people that work in that department inside the App.

- ✓ Makes it easy for users to find all the reports they need to access if they are all contained in a single App which serves as a report gallery.
- ✓ Controlling which users see which reports is easy if security is also defined along departmental lines.
- ✗ May result in Apps which contain a large number of reports and dashboards, making it more difficult for users to locate specific items.
- ✗ If users need to see reports and dashboards from departments they do not work in, then managing permissions can become more complex.

Per Report

Create one App for each report.

- ✓ Makes it easy to find individual reports and dashboards.
- ✗ Will result in a large number of Apps being created, making finding the right App difficult.
- ✗ Makes it difficult to share datasets between multiple reports, leading to duplication of datasets and increased maintenance across Apps which serve similar purposes.

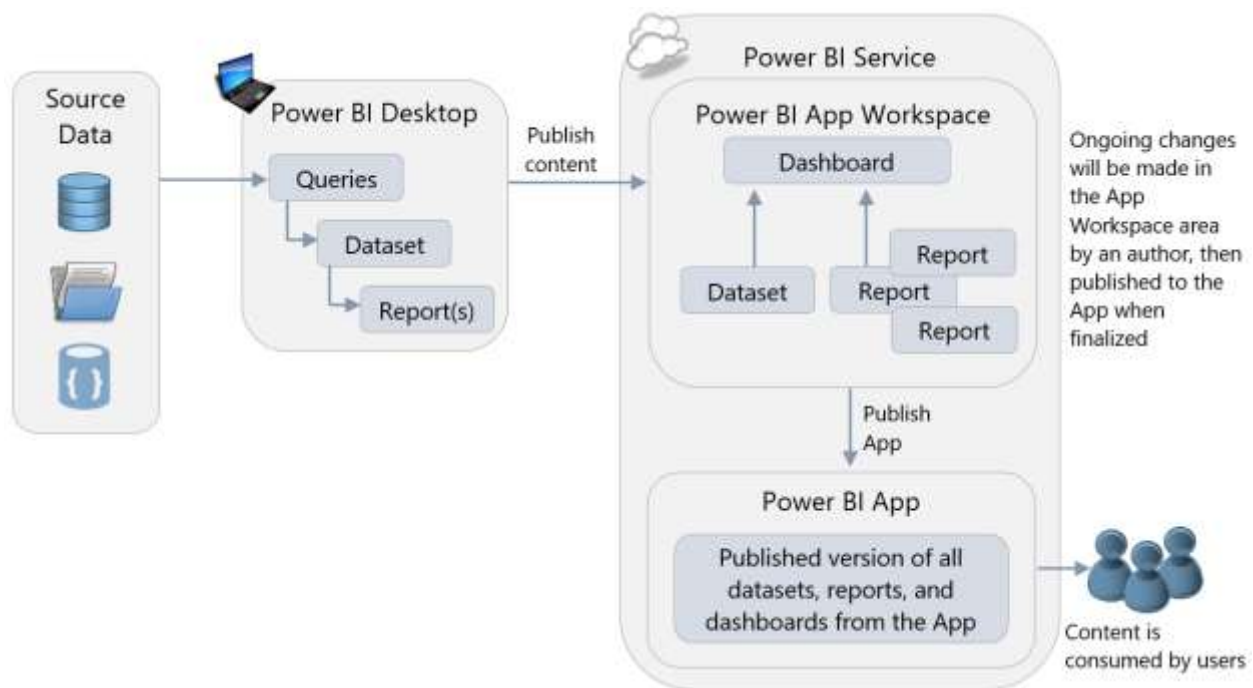
Per Business Question / Type of Analysis Conducted

Create one App for each business question that your organization needs to answer, and put all the reports and dashboards needed to answer that question inside the App.

- ✓ Makes it easy to find reports and dashboards across departmental boundaries.
- ✓ A good compromise between Apps that contain too much information and Apps that contain too little.
- ✗ Can still lead to too many Apps being created, making finding the right App difficult.
- ✗ One report may be needed to answer multiple business questions.

App Workspaces and Apps

A Power BI App Workspace is the publishing destination for datasets, reports, and dashboards. This is where collaboration occurs. As discussed in the following 'Permissions for an App Workspace' section, all members of an App Workspace should have edit permissions.




When content changes are finalized, such as an updated report, it is published from the App Workspace to the App. A Power BI App is the mechanism for delivering content to others.



Most consumers of the content will have permissions to the published App only, and not its underlying App Workspace.

Naming an App Workspace

When an App Workspace is created, various properties need to be specified which are more significant than they may first seem. One aspect is the name of the workspace. Since the name of the workspace becomes the name of the App once it has been published, it is important to choose a name that accurately reflects its contents such as “Finance Department Reports” or “Social Media Sentiment Analysis”. This will make it easier for users to find the correct content when searching for Apps in AppSource.



Because content is delivered via the App rather than the App Workspace, it is generally not recommended to include the word ‘workspace’ in its name.

The name of a workspace must be unique in the Power BI tenant. A workspace ID is created behind the name which is a concatenation of the name minus spaces and special characters. The user creating the new workspace will be told if the name is not available because it has already been used.

Permissions for an App Workspace

The following permissions must be set on an App Workspace:

- Visibility can be set to either Public or Private. The recommendation here is that App Workspaces, because they are development environments, should be set to Private so that only users specifically added to the workspace can see the content. Remember that publishing the App will allow a wider audience of users to view the reports and dashboards.
- All members of a workspace can be granted permission to either edit the contents or to only view the contents. Again, because an App Workspace should be thought of as a development environment, the recommendation is for group members to be able to edit its contents. At the time of this writing, users can only connect to a dataset in a workspace using a [Power BI Service Live Connection](#) from Power BI Desktop if users have permission to edit the contents of the workspace. If a broader audience needs to use Power BI Service Live Connections then Organizational Content Packs can be used to distribute datasets, as described below.
- Users can be added to the workspace. At present, it is only possible to add individual users and not group aliases here. In the near future, it will be possible to add multiple Active Directory security groups or Office 365 unified groups (aka modern groups). It is likely that only a few users will need to be added to a workspace in any case, because these are just the users who develop reports and dashboards – not the viewers.
- Users can be ordinary members or administrators of a workspace. Each workspace must have at least one administrator (by default it is the person who created the workspace). It is good practice to always specify more than one administrator for a group, in case one administrator goes on vacation or leaves the organization. This avoids the necessity of another Office 365 administrator manually adding additional administrators to the Office 365 group, just for the purposes of accessing Power BI content.

At the time of this writing, creation of a new Power BI App Workspace also creates an Office 365 modern group in the background. In the future, Power BI App Workspaces will be decoupled from Office 365 groups. At that point, none of the other artifacts associated with an Office 365 group (such as Calendars or OneNote notebooks) will be created automatically when you create an App Workspace.

Moving from Development to Testing to Production

You should consider having separate development, test, and production environments inside Power BI, and moving reports from development to testing, and from testing to production, only when it is appropriate to do so. Power BI does not explicitly support this kind of methodology yet: although the capability to copy contents from one App to another is a future feature coming to Power BI later this year, it is still important to plan for this from a process perspective.

At present, when you publish an App from an App Workspace this makes a copy of your reports and dashboards. Any further changes you make in an App Workspace are not reflected in the App until you republish the App. This is effectively the same thing as moving from a development environment (the App Workspace) directly to a production environment (the published App).

Using Organizational Content Packs for Collaboration and Sharing

Apps in Power BI are intended for distribution of reports and dashboards to consumers within organizations. Previously, [Organizational Content Packs](#) were a way of achieving this as well. With Apps now available, this is the preferred approach for distribution to consumers and not Organizational Content Packs.

However, there are still some scenarios where Organizational Content Packs are appropriate, as outlined below.

End User Personalization of Reports and Dashboards

Although there are many similarities between Apps and Organizational Content Packs, the main difference is that once a user has connected to an Organizational Content Pack he or she has the option to customize it by making a personal copy of it. This personal copy can be further customized according to the user's requirements and content from it can be pinned to the user's own dashboards. In contrast, users cannot edit the content of an App they have installed, nor can the content be personalized.

Since personalized copies of reports are not connected to the original report, and do not receive any updates made by the author of the original, using Organizational Content Packs in this way

can create serious problems with report maintenance and consistency. As a result it should only be used in very tightly-monitored scenarios and not used widely.

Distribution of Shared Datasets and Starter Reports

Similar to the previous scenario, Organizational Content Packs can be used as a way of distributing shared datasets through an organization from which users can build their own new reports. These datasets can be accompanied by near-empty template-like reports that users can treat as starting points for their own reports. This makes self-service report creation much easier. Something similar can be achieved using a combination of the Power BI Service Live Connection feature (to connect to published Power BI datasets) and Power BI Desktop template files, but a user must be a member of an App Workspace that contains a dataset in order to be able to connect to it from Power BI Desktop.

Apps vs Organizational Content Packs Summary

The following table summarizes up the scenarios where an App should be used and where an Organizational Content Pack is still necessary at the time of this writing:

Scenario	Best Suited
Broad distribution to end users	App
Reports and dashboards to be surfaced in a custom application via APIs	App
Personalization (save as) of reports	Organizational Content Pack
Distribute a shared dataset and/or starter reports to facilitate self-service BI	Organizational Content Pack

Sharing Dashboards

Along with Apps and Organizational Content Packs, the third method for making content available to other users is to [share dashboards](#). When a user shares a dashboard with other users, the supporting reports, workbooks, and datasets are made available as well.

Dashboard sharing is only appropriate for informal scenarios where content is to be shared with a small number of users. In most cases, using App Workspaces and Apps is preferred because:

- Apps achieve the same result with better control and visibility over what has been shared.
- Sharing from My Workspace results in a situation where there is only one owner who can manage the original content. This can become an issue if a user who shared numerous dashboards leaves the organization. This risk can be avoided by storing original content in an App Workspace instead.

- Dashboard sharing changes are viewable to others immediately, which may result in confusion for work-in-progress. Conversely, the timing of displaying changes to others can be controlled via the publish process from an App Workspace to an App.
- By default, users that have a dashboard shared with them can re-share the dashboard with other users and this presents the risk of content being shared with users that the original dashboard owner did not want to share it with. This can be prevented by unchecking the “Allow recipients to share your dashboard” option when sharing.

It is also possible to share dashboards with other users outside the organization that have Power BI Pro licenses.



A Power BI administrator can turn off external sharing, or limit it to specific users or groups, in the Power BI Admin Portal. Depending on security guidelines, it is strongly recommended this option be disabled as a precaution.

Publishing Reports to the Web

It is possible to make a report publicly visible to anyone on the public Internet with the [Publish To Web](#) feature. In this scenario, once a report has been published to the Power BI Service, when viewing the report in the browser, you can click on the Publish to Web option under the File menu. Once that has been done, a URL for the report and HTML code to embed the report in a public website will be generated. Anyone who knows the URL, whether they are inside or outside the organization, will be able to view the report.

This feature is useful if your organization needs to publish reports on a public-facing website. Typically, journalists, public agencies, corporate communications professionals or Power BI enthusiasts will use the Publish to Web feature when sharing data intended for broad public consumption, including on social media. However, this feature also presents obvious security risks. It should not be used for data that is confidential or for proprietary use only. In the future, reports published in this way will be added to a public gallery of reports.

Publish To Web is enabled by default but can be turned off by a Power BI administrator in the Power BI Admin Portal.

If you need to make Power BI reports available over the web to external users, another option is to use Power BI Embedded as discussed in [Section 3](#).

Options for Viewing Reports and Dashboards

There are a wide variety of ways that end users can view the reports and dashboards, so many that users may not be aware of all the options that are available. As always different options are

appropriate in different scenarios, and organizations should be guided to use the options that are right for them.

Viewing Reports and Dashboards in the Power BI Service in a Browser

The most common method for viewing published reports and dashboards is via a web browser. The majority of users will use the Power BI Service most of the time. The broadest set of features are available in the Power BI Service. It is also a prerequisite for using the mobile apps, or exposing content via APIs.

Power BI Report Server

For data and reports that need to be stored on-premises instead of in the Power BI Service in the cloud, Premium licensing provides the option of [Power BI Report Server](#) instead of the Power BI Service. Power BI Report Server provides a subset of the functionality of the Power BI Service. A comparison of functionality has been discussed in [Section 3](#) of this whitepaper, but it is important to restate that by opting to store data on-premises rather than in the cloud, you are also ruling out the use of features such as dashboards, apps and Q&A. These trade-offs should be considered when making the choice between cloud and on-premises.

The Power BI Mobile App for Windows 10

If users have PCs or Windows tablets running Windows 10 then the [Power BI Mobile App For Windows 10](#) can be a very effective alternative to viewing reports and dashboards in a browser, even in situations that you might not think of as “mobile”. The user interface closely mirrors what a user sees in a browser, and for less technical users may be easier to navigate. For those with touch-screens, the touch-enabled interface works better than the browser interface. Sharing screenshots of reports via email and other applications such as OneNote is built-in. Finally, the [Surface Hub Presentation Mode](#) feature not only makes presenting on a large screen much more striking, it also allows the user to make annotations on a report if they are using a Windows device with a stylus.

The Power BI Mobile Apps for Phones and Tablets

Many users prefer to view reports and dashboards on a mobile device such as a tablet or a phone, for example if they are at home, travelling for work, or just away from their desk. This is not just a matter of convenience: the [mobile apps for iPhone, iPad, Android phones and Android tablets](#) are designed for touch screens and smaller screens and make it easier to scan through reports and dashboards quickly. There is an option for creating alternative layouts for phones when you are designing reports and dashboards in Power BI Desktop and the Power BI Service. Other types of reports (paginated and mobile) published to SSRS or Power BI Report Server are integrated in the mobile apps as well.

Analyze in Excel

The [Analyze in Excel](#) feature of Power BI allows users to analyze data stored in datasets in Power BI using Excel. For users that want to explore data, Excel PivotTables provide a much faster and more intuitive way of slicing and dicing data than anything that is available in a Power BI report. They also have the advantage of familiarity: millions of Excel users know how to use PivotTables, and PivotTables connected to a Power BI dataset behave in a very similar way to regular Excel PivotTable. In addition [Excel cube functions](#) can be used to create complex and customized report layouts, for example for financial reporting, that are not possible in Power BI Desktop.

It is important to understand that many users will refuse to use anything other than Excel for reporting and analysis, however many amazing Power BI demos they see. Rather than trying to fight against this attitude, promoting the use of Analyze in Excel allows users to keep using Excel but with data centralized in Power BI datasets. This gives you the advantages of using Power BI for creating datasets (and avoiding the limitations of the Excel Data Model mentioned in [Section 7](#)) and the functionality of Excel for building reports. However, it does suffer from the serious drawback that when reports created in this way are published to Power BI they cease to be interactive, meaning that users can no longer drill down, change filters or slice and data does not change after a refresh - something that reports connecting to data stored in the Excel Data Model do not suffer from.

Microsoft Teams

[Microsoft Teams](#), a part of the Office 365 suite, is a group-chat application for teams of people working together on projects. Power BI reports can be added as tabs to a channel in Power BI, and this allows members of a team to discuss the contents of a report using threaded conversations. Since all members of a Team must have permission to view a report in an app workspace for this to work effectively, using Teams for collaboration is only useful for relatively small groups of users.

Quick Insights

The [Quick Insights](#) feature executes algorithms associated with Azure Machine Learning to search for interesting trends, correlations or outliers across an entire dataset or the data used by an individual tile pinned to a dashboard. If the algorithm happens to find something previously unknown to the user, or something with business value, a new visualization can be pinned to a dashboard directly from the Quick Insights results.

Embedded Reports in SharePoint Online and SharePoint On-Premises

Power BI reports can be embedded in a SharePoint Online Modern Page using the [SharePoint Online web part for Power BI](#). There is no equivalent web part for SharePoint on-premises but it is possible to use the Power BI API with some custom code to create a SharePoint App and make Power BI content visible in a page. There is also a third-party SharePoint add-in called

[Power BI Tiles](#) available in the Office Store that uses the API in the background and avoids the need for custom code.

Visio

Visio diagrams stored in Visio Online can be displayed in a Power BI custom visual, with data in a dataset mapped to the shapes in the diagram. Using Visio diagrams for data visualization can be a compelling alternative to standard charts and graphs, especially for scenarios where users need to see data overlaid on maps or layouts. For example Visio could be used to create a diagram of the seats in an airplane and that is then used to display data for seat profitability in Power BI.

Viewing Data from Power BI in a Custom Application

One of the features of the Power BI API is that it makes it possible to embed reports, dashboards and individual tiles from dashboards in your own applications. This has already been mentioned in the context of SharePoint on-premises above, although it can be used for any type of application.

For users inside your organization, embedding Power BI in a custom application can be beneficial in several scenarios, for example:

- Addition of reporting and analysis features to an existing business application.
- Providing more control over how users interact with a report (for example how they select items to filter by, or applying rules over what can be selected), or how they move between reports to perform a particular type of analysis.

Internal users accessing data stored in Power BI through a custom application must be licensed in the same way as users accessing data directly in Power BI reports and dashboards. It is only possible to allow external users to access data stored in Power BI through a custom application through the use of Power BI Embedded, as discussed in [Section 3](#).

Q&A

For users that are not able to build their own reports in Power BI Desktop or in the Power BI Service, [Q&A](#) makes it possible to automatically generate visualizations from English or Spanish natural language queries. These visualizations, once created, can then be further edited in the Power BI Service like any other report; they can also be saved by pinning them to a dashboard. Q&A is most effective at returning results for a dataset where care has been taken to name tables, columns and measures properly (synonyms can be defined in Power BI Desktop to supply alternative names for objects). Q&A can also be accessed from the Power BI mobile apps.

Q&A can be a very effective way of letting users with limited technical skills browse data and create simple visualizations. This in turn expands the number of users who are actively engaged with Power BI, and who otherwise would be relying on other users to build reports for them or

who would not be using data to make decisions at all. Within the settings for a dataset in the Power BI Service, featured Q&A questions can be specified which will be shown as suggestions when using Q&A.

Cortana

Cortana is the name of the digital assistant built into Windows 10. Cortana can include Power BI content in its search results if:

- You have published reports with pages in the Cortana Answer Card format in Power BI Desktop or the Power BI Service. This is nothing more than a small-scale page layout that will fit into a Cortana results pane.
- Cortana is allowed to access the dataset. This is enabled in the dataset settings in the Power BI Service.

If your organization has deployed Windows 10 to its desktop PCs and users are familiar with searching for enterprise content using Cortana, then this feature may make it much easier for users to discover Power BI reports.

Options for Subscriptions and Alerting

In addition to considering how users view Power BI content, it is important to consider how users will be brought to view that content. While many users will go to Power BI of their own accord, others may need to be prompted or reminded to view reports and dashboards daily or weekly. There may also be scenarios where users need to view a report when a particular event has taken place. This is where subscriptions and alerts are useful.

Subscriptions

For users who want to view the same report on a regular basis, [subscriptions](#) can be used to deliver email notifications. After a user subscribes to a report they will receive an email containing a snapshot of a report page when the dataset used by the report changes. Users can opt to receive emails at most either once a day or once a week. At the time of writing, to create a subscription users must go to the report and create the subscription themselves; in the future it will be possible to create subscriptions for other users.

Alerts

[Alerts](#) can be created for tiles on a dashboard that use card, gauge, or KPI visualizations. When the value displayed in the visualization changes and reaches a given threshold the alert will be fired. Alerts can be configured to fire at most once an hour or once a day. When an alert is fired, the user receives a notification that is visible in the Power BI Service in the browser and also in the Power BI mobile apps. Alerts can be effectively used to make users aware of important changes and events that require timely action on their part.

Integration with Microsoft Flow

[Microsoft Flow](#) is a cloud-based tool for creating automated workflows. A Flow can be triggered when a Power BI alert is fired, and Flow provides a wide range of integration options for actions that can happen after that. For example, emails can be sent to multiple users with customized content, data can be written to Excel files, or an issue can be created in services such as Jira and GitHub.

Section 9. Administration, Security, and Compliance

This section covers the topics of Power BI Administration, security and compliance. The Power BI documentation contains a lot of detailed information on [administering Power BI](#) and configuring security. Similarly, there is already a very detailed whitepaper on Power BI Security available [here](#) and we strongly recommend that you read it. It covers the following subjects:

- The architecture of the Power BI Service
- Power BI tenants, Azure data centers, and where data is stored
- How Power BI handles authentication when a user logs into Power BI
- How Power BI handles authentication when it connects to a data source
- How row-level security is applied
- Using Power BI and Azure ExpressRoute
- Power BI mobile apps and data storage

This whitepaper will not be duplicating content which is available in the online administration documentation, nor from the Power BI Security whitepaper. Instead the focus here will be on providing an overview of key functionality available and, more importantly, providing advice on how this functionality should be used.

Power BI Administration

Every organization using Power BI should specify at least two administrators to manage its tenant. Two administrators are necessary to cover vacations, sickness, and employees leaving the organization. The following is a description of the tasks involved with administering Power BI and the permissions required to do so.

Administrator Roles Related to Power BI

There are several relevant administrator roles which will be discussed in this section:

Type of Administrator	Scope
Office 365 Global Administrator	Office 365
Office 365 Billing Administrator	Office 365

Type of Administrator	Scope
Power BI Service Administrator	Power BI
Power BI Premium Capacity Administrator	A single Premium Capacity
Power BI App Workspace Administrator	A single App Workspace
Power BI Organizational Content Pack Owner	A single Organizational Content Pack

Office 365 Administrators and The Power BI Administrator Role

Global Administrators in Office 365 or Azure Active Directory will have administrator rights in Power BI. It is also possible for an Office 365 Global Administrator to assign other users to the [Power BI Service Administrator role](#), which grants administrative rights over Power BI features only. Most large organizations should have two, at a minimum, dedicated Power BI Service Administrators with BI experience and who can provide effective oversight. Power BI Service Administrators have access to the [Power BI Admin Portal](#) which includes various tenant-level settings regarding functionality, security, and monitoring such as:

- Share content with external users
- Publish to web
- Export data
- Export reports as PowerPoint presentations
- Publish content packs to the entire organization
- Create template organizational content packs
- Ask questions using Cortana
- Use Analyze in Excel with on-premises datasets
- Use global search for Power BI
- Create audit logs for external activity auditing and compliance
- Usage metrics for content creators
- Data classification for dashboards
- Embed content in apps

Certain tenant settings can be specified for users and groups, whereas other settings pertain to the entire organization.

Note that the Power BI Service Administrator role does not grant the ability to assign licenses to users or view [audit logs in Office 365](#). Therefore, the task of administering Power BI cannot currently be performed by users who are solely members of the Power BI Service Administrator role.

Initial Tenant Setup

Ideally, planning a Power BI deployment should start before anyone has even logged in to Power BI for the first time. The first task of the person charged with configuring a new Power BI tenant is to find whoever it is in the organization who administers Office 365 or Azure Active

Directory. As has already been noted, the Office 365 Global Administrator will be able to add users to the Power BI Service Administrator role and will also be able to control settings in Office 365 for how other users can sign up for Power BI. The Office 365 Global Administrator will also be needed on occasion for help and support after the Power BI tenant is up and running.

Unless the capability has been disabled in Office 365, anyone can sign up for Power BI Free or a Power BI Pro trial with a corporate email address (email addresses from providers such as Hotmail or Gmail cannot be used). The first user to sign up creates a new auto-generated Power BI tenant for the organization based on the e-mail address that was used. For an enterprise deployment of Power BI, it is likely you will want to control this more carefully. If a tenant has been auto-generated via user sign-ups, an Office 365 Global Administrator can take over the tenant and [become the administrator](#). If an auto-generated tenant was inadvertently created in the wrong data region, you may want to consider saving any content in it, deleting it, and re-provisioning the tenant again in the correct data region.

Self-service sign-ups can be disabled using PowerShell (full details are available [here](#)). In this scenario, the Office 365 Global Administrator or Office 365 Billing Administrator becomes responsible for purchasing Power BI subscriptions and assigning licenses as appropriate.

Choosing an Azure Data Center for Power BI

When a Power BI tenant is created, it is created in an Azure data center. Choosing the correct Azure data center to host your Power BI tenant is important for two reasons:

- It may be important for legal reasons that corporate data is stored in a specific jurisdiction.
- The performance of reports and dashboards depends, in part, on users being in close proximity to the Power BI tenant.

It is not possible to select a data center directly. Instead, Microsoft selects the data center closest to the country selected in the "Where's your company located" dropdown box when the tenant is first created (see this [article](#) for more details). The location of the data center chosen is displayed in the 'About Power BI' dialog, accessible from the Help & Support dropdown menu (which is displayed as a question mark icon) in the top right-hand corner of the screen when logged into the Power BI Service.

It is also vital that the right decision is made first time because it is currently not possible to move a Power BI tenant to another Azure data center. The ability to migrate a Power BI tenant to another Azure data center will be a future capability. In the interim, the only option to move to a different data center is to delete the Power BI content and recreate it in a new tenant.

Managing Users and Licenses

The purchase of subscriptions and assignment of user licenses takes place in the Office 365 admin center and is therefore a job for an Office 365 Administrator or an Office 365 Billing Administrator. Organizations that are not utilizing Power BI Premium will need to assign Pro licenses to all users who intend to do any sharing and collaborating. See [Section 3](#) for a detailed list of features per license type.

Organizations that are using Power BI Premium may have a more complex choice to make, depending on the breakdown of the user base:

- Free licenses can be assigned to read-only users who do not create any content that is to be shared. Users with a free license may view Apps published to Power BI Premium capacity, and may also publish content to their “My Workspace” workspace.
- Pro licenses must be assigned to all users who create and share content, and to users who want to view any shared content except Apps that are stored in Premium capacity.



If a workspace is moved from Premium capacity back to Shared capacity, read-only users who previously had a Free license will now require a Pro license.

Managing Power BI Pro Trials

In large organizations, it may be commonplace for users to accept a prompt within the Power BI Service to start a Power BI Pro trial. This can occur when content is shared with a user who only currently has a Free license (because there is no error or notification if a Pro user shares content with another user who does not have Pro). If this situation is common, [this video](#) describes how to use PowerShell to check for trial licenses and automate the assignment of a Pro license permanently to avoid the trial running out.

Managing Power BI Premium Capacities

Power BI workspaces are stored in capacities. When a Power BI tenant is created, initially it only has one capacity available: Power BI Shared Capacity. This is capacity shared by all Power BI customers. Buying Power BI Premium Capacity provides dedicated resources in the cloud for your workspaces, which in turn provides added benefits such as better performance and increased storage limits. See [Section 3](#) for more information about Premium licensing.

Power BI Premium capacity can be purchased by going to the Power BI Admin Portal, clicking on the Premium Settings pane, then clicking the “Buy” (or “Purchase More”) button. This redirects the administrator to the Office 365 Admin Portal where the purchase takes place. A single Power BI tenant can have multiple Power BI Premium capacities, each of which can be at different tiers. A capacity can be set up by a Power BI



Only an Office 365 Global Administrator or Billing Administrator can

Administrator after it has been purchased by an Office 365 administrator.

purchase Power BI Premium capacity.

Purchasing Power BI Premium capacity brings two more sets of permissions that may be granted to a user within a capacity: Capacity Administrators and assignment permissions. Power BI Premium Capacity Administrators can perform the following tasks within a capacity:

- Grant assignment permissions
- Grant workspace permissions
- Bulk assign workspaces to capacity
- Remove workspaces from capacity
- Monitor capacity usage

Assignment permissions allow users (who must have a Power BI Pro license) to:

- Assign workspaces to a capacity
- Allow other users with Power BI Pro licenses to access these workspaces

Assigning Workspaces to Power BI Premium Capacity

Once Power BI Premium capacity has been purchased, workspaces must be assigned to a capacity to take advantage of its benefits. This can happen in several ways:

- Any user with assignment permissions, whether they are a Power BI Capacity Admin or a regular Pro user with assignment permissions, can assign an individual workspace to a capacity.
- A Power BI Capacity Admin can supply a list of users or security groups and assign all workspaces of which these users are administrators to a capacity.
- A Power BI Capacity Admin can assign all workspaces to a capacity, making it the default storage location for all workspaces and giving all current and future users permission to reassign individual workspaces to this capacity.

Workspaces may be assigned from one Premium capacity to another, or they may be assigned out of Premium capacity back to Power BI Shared Capacity.

If the entire organization's Power BI content can fit into a Premium capacity then it likely makes sense to do so. If not, then workspaces should be prioritized for Premium capacity depending on:

Dataset Size	Any datasets that may soon exceed the size limits of Power BI Shared Capacity should be relocated to Premium Capacity to prevent data refresh failures. At the time of this writing, the size limit is 1GB-these limits will be increasing in the future within Premium Capacity.
--------------	---

Report Performance	Any workspaces containing slow reports should be relocated because doing so will likely improve performance.
Number of Passive (Read-Only) Viewers	Users who only need to view App content stored in Premium Capacity require a Free license rather than a Pro license. Therefore, moving these Apps into Premium Capacity may save the organization on licensing costs.

Deploying Power BI Desktop

Nearly all users who develop Power BI reports will need Power BI Desktop installed on a Windows PC, unless they intend to exclusively build reports in the Power BI Service using a browser. Power BI Desktop is available in 32-bit and 64-bit versions. However, we recommend installing the 64-bit version: Power BI uses an in-memory database engine and the 32-bit version cannot address enough memory to handle larger data volumes. The minimum requirements for a PC to run Power BI Desktop can be found [here](#). A PC with at least 8GB of RAM is recommended.

There is a new release of Power BI Desktop every month and for compatibility reasons it is important that all report developers use the same release of Power BI Desktop. It may not be possible to open .pbix files created with newer versions of Power BI Desktop, or downloaded from the Power BI Service, in older versions of Power BI Desktop. We recommend testing a new release with existing .pbix files, followed by deployment to developers' desktops as soon as possible.

No subscription or license is required to install or use Power BI Desktop.



Typically IT pushes Power BI Desktop updates using tools such as Microsoft System Center Configuration Manager to ensure version compatibility, particularly if users do not have rights to install software on their own machine.

Managing Power BI Mobile App Deployment

The Power BI mobile apps can be mass deployed to users' corporate mobile devices using [Microsoft Intune](#). We recommend installing the mobile apps for every user that has a Power BI license assigned to them at the beginning of a Power BI project because some users will prefer to consume Power BI content through the mobile apps rather than through a browser.

For users with Windows 10 PCs, especially for devices with touch screens such as Surface Pros, we also recommend installation of the Power BI Mobile App for Windows.

Compliance and Auditing

It has already been mentioned that there may be some types of data that may not be stored in Power BI cloud storage, or must be stored in data centers in a particular jurisdiction, for legal reasons. It is the job of the Power BI administrator to be aware of all the legal issues surrounding the data that is used in Power BI (we recommend taking advice from your organization's legal resources, if necessary) and to put in place measures to enforce organizational policies. More details on Power BI and the compliance and regulatory standards to which it adheres can be found in the [Microsoft Trust Center](#).

It may also be necessary to keep a record of the actions of individual users for regulatory reasons, for example which users have viewed a given report or dashboard. Power BI auditing can be enabled in the Power BI Admin Portal and the resulting audit logs can be accessed via the Office 365 Security & Compliance Center. More details on this can be found [here](#). If your organization has an Azure Active Directory Premium license then you can use Azure Active Directory's built-in reports to find out which users have signed in to Power BI and when they last signed in, as shown in this [video](#).

Monitoring Usage

It will be useful to monitor usage of Power BI for a variety of other reasons apart from compliance, for example:

- To see which users are making frequent use of Power BI and which users are not. If Power BI is not being used regularly by some or all users it will be necessary to find out why and take action. More training may be appropriate. Alternatively, it may be due to dashboards and reports not displaying the required data. Or, perhaps reports are slow to render or respond and need tuning to improve the user experience.
- To identify the most commonly used sources of data. Frequently a Corporate BI environment can be improved upon by learning what is occurring via Self-Service BI initiatives.
- To determine which workspaces should be assigned to Power BI Premium capacity to improve performance or to avoid limits on dataset size.
- To look for suspicious usage patterns that indicate a user is exporting data for unauthorized purposes.
- To identify what type of devices are being used to view reports, in case these reports need to be redesigned to suit those devices.
- To become aware of errors being experienced by users. This is particularly true for assisting with data refresh issues.

Usage information can be obtained in the following ways:

1. From the [Power BI audit logs](#) in Office 365 which were mentioned above. These logs are disabled by default, and can be enabled in the Power BI Admin portal.

2. The [Usage Metrics](#) report in the Power BI Admin Portal gives a basic tenant-wide view of the number of reports, dashboards, and datasets in an organization, as well as the number of authors and consumers of content.
3. More detailed [usage metrics reports for individual dashboards and reports](#) can be accessed on the Actions menu by anyone with Edit access to the dashboard or report.
4. Power BI Premium Capacity Admins can view [high-level usage metrics](#) for their capacities in the Manage Capacity pane of the Power BI Admin Portal.

Data Privacy and Security

The reports and dashboards in Power BI are likely to contain some of the most sensitive information your organization possesses. Managing who has access to this data and how it is shared around your organization is, as a result, one of the most important ongoing tasks in Power BI. The following security considerations will be discussed in this section:

- Access to the Power BI Service
- Access to specific content in Power BI (workspaces, etc.)
- Row-level security
- Source data security
- Rights to share and publish content
- Rights to export and print content
- Data privacy levels
- Data classifications

Controlling Access to Power BI

Since Power BI relies on Azure Active Directory (AAD) for authentication, it can use AAD's full range of functionality to control where and how users can log in to Power BI. For example it may be important to restrict access to Power BI to users within a specified IP range (this could be IP addresses inside the corporate network), to users of specific security groups, or to users of specific types of device – all of this is possible with [AAD Conditional Access policies](#). It may be difficult to find a balance between unrestricted access (which runs the risk of an unauthorized user obtaining someone else's credentials and seeing sensitive data) and overly restrictive access (which frustrates users who need to see Power BI content, perhaps when traveling or at home). The best strategy is to start off with a cautious access policy while remaining responsive to the needs of the business and loosening restrictions where appropriate.

It is also possible to use AAD Conditional Access policies to force the use of multi-factor authentication when signing in to Power BI, adding an extra layer of protection against stolen credentials. The Power BI Mobile Apps can be configured to require the user to enter a PIN code before they can be opened if they are deployed using [Microsoft Intune](#).

Managing Access to App Workspaces, Apps, and Content Packs

At the most basic level, whether a user can see a given piece of information is determined by whether he or she has access to an App Workspace, App or Organizational Content Pack that displays that data.

As already discussed in [Section 8](#), in most cases App Workspaces should be treated as development environments. Only a small number of users – the developers of the content inside the App Workspace – should be added as members. These members can only be added by email address.

After the App Workspace has been published to an App then it can be made available to a wider audience of users, and at this point you have three choices as to who can see the App:

- It can be made available to the entire organization. This is only likely to be appropriate in very small organizations or for data that is non-sensitive.
- Individual users can be granted access. We do not recommend this approach unless an App is to be restricted to one or two specific people. Managing individual users can become tedious and error-prone, and can only be done by an administrator of the App Workspace.
- Office 365 email distribution lists can be granted access. This is the best option for larger organizations for two reasons. First, it means that the same email distribution lists can be reused to control access to multiple Apps, which saves a lot of time setting up access rights for Apps and minimizes the chances of error. Second, it means that access to the App can be controlled centrally by someone who is responsible for managing security, which makes administration within Power BI that much easier.

Access to Organizational Content Packs can be controlled in a similar way, although with Organizational Content Packs Office 365 groups, distribution groups, and security groups may all be used. Since Apps are intended to supersede Organizational Content Packs, it is recommended to use Apps if they meet the requirements. More details are available in [Section 8](#).

Row-Level Security

For situations where different users should only see some of the data in a dataset, Power BI [row-level security](#) allows filters to be applied on the rows in any table in that dataset. For example, if a table in a dataset contains sales data for different countries, it may be a requirement that users in one country can only see the sales information for that country in any reports they view. Row-level security allows a report developer to create a single report that, when viewed by users in the USA, only shows sales information for the USA, when viewed by users in Russia only shows sales in Russia, and so on.

Row-level security can work in two ways. In the most common scenario a report developer creates multiple security roles, gives each role permission to see a different slice of data, and

then assigns users or groups of users to those roles. However, it is also possible to create a dynamic security role, where a single role can display different slices of data to different users using DAX expressions that can determine who is trying to connect to the report and filter the data appropriately. Dynamic security is more complex to set up but can avoid the need to create and maintain large numbers of roles with different permissions.

If row-level security is applied in a dataset it is crucial that it is properly tested, especially if complex DAX expressions are used in the role definitions. Moreover, if the same data is used in multiple datasets, row-level security must be applied consistently across them all. The use of Power BI Service Live Connections, where multiple reports can connect to a single shared dataset, can significantly reduce the overhead of managing row-level security.

It is also important to keep in mind that row-level security that is set up within Power BI Desktop applies to only that one solution. This increases the potential for inconsistencies, errors, and potentially maintenance. If there are numerous Power BI solutions all setting up row-level security, then specifying security at the data source level may be more effective.

Securing Data at the Data Source Level

Since creating, managing and maintaining row-level security can be difficult across multiple datasets it may make more sense to apply security at the data source level instead, if that is possible.

The use of Live Connections to Analysis Services instead of imported datasets means that data is already centralized, and Analysis Services Tabular models have the same row-level security functionality as Power BI datasets (Analysis Services Multidimensional also has similar functionality) so row-level security can be centralized in one place. Analysis Services Tabular 2017 and Azure Analysis Services can also apply security to entire tables and individual columns within tables, something that is not possible with Power BI's own security. If Power BI uses SQL Server as a data source in DirectQuery mode, it is also possible to use the SQL Server's own row-level security functionality to achieve the same result.



When connecting to a PaaS service such as Azure SQL Database or Azure SQL Data Warehouse, you will also need to address firewall port openings for users who connect from Power BI Desktop. This is frequently handled by opening the relevant IP address range for an office. In Azure, firewall ports can be opened at the server level via the Azure Portal, or the database level via T-SQL. The more secure practice is to open needed ports at the database level.

Restricting Sharing and Publishing

As well as restricting access to App Workspaces, Apps and Organizational Content Packs, you will likely also want to specify who may publish Apps and Organizational Content Packs. Only users with Power BI Pro licenses can publish Apps and Organizational Content Packs, but unless Apps are published to Power BI Premium capacity users will also need Power BI Pro licenses to view shared content so restricting the type of license assigned to a user is not a workable way of doing this. Instead, in the Tenant Settings pane of the Power BI Admin Portal it is possible to apply the following restrictions:

- Prevent the ability to publish Apps and Organizational Content Packs to the entire organization. Notice that this does not prevent the creation of Apps and Organizational Content Packs completely, just the ability to publish them so that the whole organization can see them.
- If publishing to the entire organization is enabled then it is also possible to restrict the ability to do this to specific security groups, or alternatively to exclude specific security groups from being able to do this.

These settings can be applied to regular Apps and Organizational Content Packs, separately from template Apps and template Organizational Content Packs.

The Power BI Admin Portal also contains settings to control sharing with external users and publishing reports to the web, both of which are enabled by default. It is strongly recommended to consider restricting both to prevent unauthorized data sharing, either by disabling entirely or only enabling them for specific security groups if there is a genuine need to share outside the organization.

Restricting Export and Printing

The Power BI Admin Portal also includes options to disable export to Excel, export to PowerPoint, and printing. It is rarely a good idea to disable these features because many users consider them to be the most useful features in Power BI, regardless of whether an administrator thinks they should be using them or not. It is better to give users the features they want than have them ignore Power BI and use other, less efficient reporting tools instead.

Data Privacy Levels in M Queries

One easy-to-miss feature around data security is the concept of data privacy levels for loading data. In certain scenarios, the M queries that Power BI uses to load data into datasets may be able to gain a significant performance boost by sending data from one data source to another.

For example, a user may have two data sources: an Excel workbook and a table in a SQL Server database, and want to be able to filter the data in the SQL Server table using a value found in the Excel workbook. The most efficient way to do this is for Power BI to generate a SQL query with a WHERE clause to perform the necessary filtering on the server (known as “query folding”).

However, to do this, the WHERE clause of this query will have to contain a value taken from the Excel workbook, and if this is the case a DBA monitoring the queries would be able to see this value. If the Excel workbook contains sensitive data this may represent a breach of security: the DBA is able to see values that he or she may not have permission to see. The alternative is for Power BI to download the entire contents of the SQL Server table and perform the filtering itself, something that is likely to be more inefficient and slow, but at the same time more secure.

Power BI developers can set data privacy levels for each of the data sources they use in the Query Editor of Power BI Desktop, and this controls whether data from one data source is sent to another. All report developers should be made aware of this feature and its implications if they are working with sensitive data. Report developers may be tempted to configure Power BI Desktop to always ignore data privacy levels but this can be an insecure practice in organizations that handle sensitive data.

Data Classification

One last feature to point out is the ability to assign [data classification tags](#) to dashboards to indicate to users the sensitivity of the data they are looking at. For this to work well, a Power BI Service Administrator needs to enable the feature (in the Power BI Admin portal), create the tags, and instruct all Power BI dashboard developers to use these tags consistently. Since data classification tags cannot be applied to reports, though, this feature will only be useful if users always navigate to reports via a dashboard rather than go to them directly.

Security Keys

On-Premises Data Gateway Recovery Key

As discussed in [Section 6](#), a recovery key is assigned at the time the On-Premises Data Gateway is installed. The recovery key is required if the gateway is to be relocated to another machine, or if the gateway is to be restored.

Section 10. Getting Support from Microsoft and the Community

It is inevitable that, at some point, you will encounter a problem that you do not know how to solve. The first thing you must do is determine the kind of problem you have, and once you have done that you can determine the appropriate course of action and who you can ask for help.

Support and Help

Power BI Support, Service Outages, and Degradations

Although the Power BI Service is extremely reliable there may be times when the service is unavailable or certain features do not work as expected. If you find that your published reports and dashboards are having problems and you have not changed anything, check the status of the Power BI Service at:

<https://powerbi.microsoft.com/en-us/support/>

This page will tell you about any outages or degradations that Microsoft is currently aware of and the timelines for getting them resolved.

You can also open a support ticket from this page, which is available to users with paid Power BI Pro licenses.

Reporting Bugs

If you suspect that something is wrong with the Power BI Service and that Microsoft is unaware of the problem, or if you find a bug in Power BI Desktop, the place to report the problem is the Power BI Issues forum:

<https://community.powerbi.com/t5/Issues/idb-p/Issues>

Before you post, make sure that you search the forum to check whether someone else has already posted the same issue. If not, when you post, make sure that you include full details of the steps needed to reproduce the problem and screenshots of any error messages that you receive.

Getting Help with Power BI Development

If you are having trouble with any aspect of Power BI development or configuration then the Power BI community forums are a great place to get free help and support from Microsoft and from other members of the Power BI community. You can find them here:

<https://community.powerbi.com/t5/Forums/ct-p/Forums>

Once again, you should search the forums to see whether anyone else has asked the same question before you post. You will increase the chances of getting your problem solved if you post in the right forum for your issue, you supply all the necessary technical details and examples of your data, and you are polite and courteous.

Issues, Feedback, and Suggestions for Improving Power BI

On the Power BI Ideas site (also known as User Voice), you can suggest improvements, add new ideas, vote for and/or comment on existing ideas, and view status of ideas from the product team (such as started, or under review):

<https://ideas.powerbi.com/forums/265200-power-bi-ideas/filters/top>

As always, check to see whether a similar idea has already been suggested before you post. You can also vote for issues that other people have posted. Microsoft takes the feedback on this forum very seriously, and the more votes an issue receives the more likely it is that it will be implemented.

Whitepaper feedback

We welcome any feedback you may have about this, or any other, whitepaper. To provide that feedback, you can head over to the Power BI Community site [whitepaper feedback forum](#).

Learning More About Power BI

Power BI Team Blog and Videos

Power BI is a complex product that changes quickly, so there is always more to learn about it. The Power BI blog has regular roundups of new updates and features, technical tips, and announcements. To keep your knowledge up-to-date, you should follow all posts on the official Power BI blog:

<https://powerbi.microsoft.com/en-us/blog/>

It will inform you of new features and events in the community, and also has posts on Power BI report development tips and tricks. There are many other useful resources available for you to learn about Power BI, such as the Guy In A Cube channel on YouTube (run by Microsoft staff):

<https://www.youtube.com/channel/UCFp1vaKzpfvoGai0vE5VJ0w>

Power BI Community Site

Community contributions including Forum Q&A, Themes Gallery, Data Stories Gallery, R Scripts Gallery, Quick Measures Gallery, among others.

<http://community.powerbi.com/>

Power BI Community Bloggers

The Power BI blog roll collects posts from many of the prominent bloggers in the Power BI community:

<https://community.powerbi.com/t5/custom/page/page-id/BlogRollExtracts>

Power BI User Groups

Power BI user groups have been established in many places around the world and are a great place to hear talks by Power BI experts, talk to other Power BI users and recruit new staff. To find a local Power BI user group see:

<https://www.pbusergroup.com>

Guided Learning

For introductory training on Power BI, the videos in the Guided Learning section of the Power BI site are useful place to start:

<https://powerbi.microsoft.com/en-us/guided-learning/>

Power BI Documentation

The official Power BI documentation site is very detailed and updated on a regular basis, and should be your first port of call when you need to learn about any feature of Power BI:

<https://powerbi.microsoft.com/en-us/documentation/>

Additional Resources and Links

Microsoft Power BI Premium Whitepaper

Published: May 2016.

<https://aka.ms/pbipremiumwhitepaper>

Power BI Premium Calculator

Online calculator for estimating pricing for usage of dedicated capacity in Power BI.

<https://powerbi.microsoft.com/en-us/calculator/>

Support Timeline for Power BI Report Server

Timeline for security and critical updates

<https://powerbi.microsoft.com/en-us/documentation/reportserver-support-timeline/>

Power BI Security Whitepaper

Published: September 2016.

<https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-power-bi-security/>

Power BI Governance and Deployment Whitepaper

Published: March 2016.

<https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-governance/>

Microsoft Trust Center

Status of compliance and certifications.

<https://www.microsoft.com/en-us/trustcenter/cloudservices/powerbi>

Microsoft Online Services Terms

Terms which govern Online Services, including the Power BI service.

<http://www.microsoftvolumelicensing.com/DocumentSearch.aspx?Mode=3&DocumentTypeId=31>

Microsoft Enterprise Privacy Statement

Privacy terms for Online Services, including the Power BI service.

<https://www.microsoft.com/EN-US/privacystatement/OnlineServices/Default.aspx>

Sovereign Clouds

Power BI Service availability for sovereign clouds.

<https://powerbi.microsoft.com/en-us/clouds/>

Using Tabular Models in a Large-Scale Commercial Solution

An Analysis Services case study

<https://msdn.microsoft.com/en-us/library/dn751533.aspx>

Performance Tuning of Tabular Models in SQL Server 2012 Analysis Services

Tuning documentation for Analysis Services (still applicable post 2012)

<https://msdn.microsoft.com/en-us/library/dn393915.aspx>