



Planning a Power BI Enterprise Deployment

Whitepaper

Summary: This is a technical whitepaper which outlines considerations and best practices for achieving an optimal, well-performing, enterprise level Power BI deployment.

Writers: Melissa Coates ([BlueGranite](#)), Chris Webb ([Crossjoin Consulting](#))

Technical Reviewers: Meagan Longoria ([BlueGranite](#)), Lukasz Pawlowski, Adam Wilson, Will Thompson, Chris Finlan, Orion Lee, Nimrod Shalit, Kay Unkroth, Adi Regev, Adam Saxton, Tamer Farag, Christian Wade, Youssef Shoukry, Amanda Cofsky, Michel Abd Elmalik, Maya Shenhav, Brandon Unger

Version: 2.0

Published: July 2018

Contents

Section 1. Introduction.....	9
Purpose of this Whitepaper	9
Scope of This Whitepaper.....	9
Future Features	10
High Level Overview of Power BI.....	11
Definitions and Terminology	11
Section 2. Power BI Usage Scenarios.....	13
Personal BI.....	13
Small Team Collaboration.....	15
Large Team Collaboration and Distribution.....	17
Enterprise Level Collaboration and Distribution.....	19
Centralized Corporate Reporting and Data Warehousing.....	20
Prototyping Activities and Sharing	22
Organizational Embedded Analytics	23
External Embedded Analytics	25
On-Premises (Non-Cloud) Reporting Portal	26
Workflow Automation + Updating Data from Within Power BI	27
Self-Service Data Preparation With Dataflows	29
Streaming and Near-Real-Time Reporting.....	31
Section 3. Power BI Architectural Choices.....	32
The Power BI Service	32
What is the Power BI Service?	32
Why Use the Power BI Service?	33
Power BI Service Release Cycle.....	35
Power BI Report Server.....	35
What is Power BI Report Server?.....	35
Why Use Power BI Report Server?	35
Power BI Report Server Release Cycle.....	36
Comparing Power BI Report Server Functionality with the Power BI Service.....	36

Power BI Premium.....	37
What is Power BI Premium?.....	37
Power BI Premium Capacities.....	37
Buying Power BI Premium Capacity Nodes	38
Premium Tiers and SKUs.....	39
Deciding Whether To Use Power BI Premium.....	41
Power BI Embedded	43
What is Power BI Embedded?	43
Deciding Whether to Use Power BI Embedded.....	43
Power BI Embedded vs Organizational Embedding.....	44
Supported Features in Power BI Embedded.....	44
Section 4. Power BI Licensing and User Management.....	45
Power BI Licensing Summary.....	46
How to Purchase Power BI Licenses	47
Ways to Assign a Power BI User License.....	49
Managing Power BI Trials and License Assignments.....	50
Types of Power BI User Licenses.....	51
Power BI Free.....	51
Power BI Pro	52
Other Power BI License Types	52
Power BI Premium	52
Power BI Report Server	53
Power BI Embedded.....	53
Section 5. Power BI Source Data Considerations.....	54
Taking an Inventory of Data Sources	54
Data Source Limitations	55
Cleansing, Filtering, Transforming and Integrating Data	58
Dataflows	59
Section 6. Power BI Data Storage Options.....	61
Data Storage Modes.....	61
Importing Data	61
Live Connection to Analysis Services	62
DirectQuery Mode	63

Composite Models	64
Real-Time Streaming Datasets.....	65
Choosing a Storage Mode.....	66
Is Analysis Services Already in Use?	66
How Much Data is Needed by the Report?	68
How Frequently Does Data Need to be Refreshed?	69
How Complex is the Dataset? Are any Complex Calculations Necessary?	71
How Many Concurrent Users Need to be Supported?	72
Are there Legal or Regulatory Reasons for Storing Data in a Specific Location?	72
What Technical Skills are Present In-House?	73
What Budgetary Constraints Need to be Considered?	73
Summary of Data Storage Recommendations	74
Section 7. Power BI Data Refresh and Data Gateway	75
Types of Data, File, and Cache Refresh in Power BI	76
Dataset Refresh.....	76
Package Refresh (File Level).....	76
Cached Tile Refresh.....	76
Cached Visual Container Refresh	77
Introduction to the On-Premises Data Gateway.....	77
Gateway Terminology.....	78
Mode 1: On-Premises Data Gateway	79
Mode 2: Personal Mode.....	80
Summary of Features Per Mode.....	81
When a Data Gateway is Required	82
Process for Using the On-Premises Data Gateway	82
Current On-Premises Data Gateway Limitations.....	82
Considerations for Using the On-Premises Data Gateway	83
Data Gateway Installation File.....	83
Naming Conventions	83
Identities and Credentials Used by the Data Gateway	85
Associating Datasets with a Gateway	87
User Interactions with the Data Gateway	87
Sharing of Data Gateway with Other Applications	88

Using One or More Data Gateways.....	89
Performance Impact on the Server Running the On-Premises Data Gateway.....	89
Where to Install the On-Premises Data Gateway.....	90
Firewall Ports.....	90
Monitoring and Managing the Data Gateway.....	91
Data Refresh Duration.....	91
Pausing or Disabling of a Data Refresh.....	91
Keeping the On-Premises Data Gateway Updated.....	91
Monitoring Data Refresh Operations	92
Data Refresh Limitations Based on License Type.....	93
Data Gateway Disaster Recovery.....	94
Section 8. Power BI Report Development Considerations.....	94
Choosing the Right Tool for Report Development.....	94
Power BI Desktop.....	94
Power BI Service	95
Excel	95
Best Practices for Report Design in Power BI Desktop	97
Managing Report Authorship	97
Version Control / Source Control for Power BI Content.....	97
Separating Dataset Development from Report Authorship	98
Use Templates to Speed Up and Standardize Report Development.....	100
Minimizing the Amount of Data Loaded.....	100
Reducing Dataset Size.....	101
Naming Objects and Q&A	101
Minimizing the Amount of Data Displayed on a Page.....	102
Accessibility.....	102
Custom Visuals	103
Custom Data Connectors	104
Creating Dashboards to Help Navigation	104
Developing for the Power BI Service and Power BI Report Server	104
Section 9. Power BI Collaboration, Sharing and Distribution.....	105
The Role of Power BI Desktop.....	105
Collaboration, Sharing, and Distribution in the Power BI Service	106

Workspaces in the Power BI Service	106
Options For Making Content Available To Users in the Power BI Service.....	111
Controlling the Timing of Publishing Content.....	117
Deployments To Development, Test, and Production.....	118
Collaboration, Sharing and Distribution with Power BI Report Server	121
Options for Consuming Content	122
Viewing Reports and Dashboards in a Web Browser	122
The Power BI Mobile App for Windows 10.....	122
The Power BI Mobile Apps for Phones and Tablets	122
Analyze in Excel	123
Embedded Reports in Microsoft Teams.....	123
Embedded Reports in SharePoint Online and SharePoint On-Premises	123
Viewing Data from Power BI in a Custom Application	124
Cortana.....	124
Options for Subscriptions and Alerting.....	125
Subscriptions.....	125
Alerts.....	125
Integration with Microsoft Flow	125
Section 10. Power BI Administration.....	126
Initial Power BI Tenant Setup.....	126
Creation of a Power BI Tenant	126
Geographic Data Region	126
Administrator Roles Related to Power BI	127
Power BI Admin Center	129
Tenant-Level Settings	129
Managing Distribution of Organizational Custom Visuals	131
Managing Embed Codes for Publicly Available Reports.....	133
App Workspace Settings.....	133
Monitoring Usage of Power BI	134
Power BI Service Usage.....	135
Power BI Report Server Usage	140
Automating and Managing Power BI Programmatically.....	140
PowerShell Cmdlets for Power BI.....	140

PowerShell Cmdlets for Power BI Report Server	141
PowerShell Cmdlets for On-Premises Data Gateway	141
Power BI REST APIs	142
Office 365, Azure Active Directory, and Microsoft Graph REST APIs	144
Managing Power BI Premium	145
Managing Power BI Premium Capacities	145
Managing Capacities in Power BI	146
Assigning Workspaces to Power BI Premium Capacity	146
Assigning Capacity Admins and Capacity Assignment Permissions	148
Monitoring Capacity Resources	149
Deploying and Managing Installers and Applications	149
Power BI Desktop	149
Power BI Mobile Apps	151
On-Premises Data Gateway	152
Power BI Publisher for Excel	152
Analyze In Excel	153
Firewall Settings	153
Managing Power BI Report Server	153
Managing Users, Licenses, and Power BI Trials	154
Monitoring Service Health	154
Power BI Support Site	154
Azure	155
Office 365	155
Section 11. Power BI Security	156
Integration with Azure Active Directory	157
Managing Access to the Power BI Service	157
Azure B2B	157
Managing Access to App Workspaces and Apps	158
Securing Data in Transit	159
Securing the Connection Between Data in Azure and Power BI	160
Securing the On-Premises Data Gateway	160
Data Transmitted Outside of the Power BI Service	160
Securing Data at Rest	160

Dataset Row-Level Security	161
Securing Data at the Data Source Level.....	162
Encrypted Data Storage for Imported and Push Datasets	163
Encrypted Storage for Metadata and Visual Cache.....	163
Encrypted Storage of Credentials to Connect to Source Data.....	163
Limiting Functionality in the Power BI Service	163
Restricting Sharing and Publishing.....	163
Restricting Export and Printing.....	164
Data Privacy and Classification.....	164
Data Privacy Levels in M Queries	164
Data Classification.....	165
Power BI Premium	165
Compliance Certifications.....	165
Securing Power BI Report Server.....	166
Managing Encryption Keys.....	166
Mobile User Access Control	167
Power BI in Sovereign Clouds.....	167
Private Connectivity with ExpressRoute	167
Section 12. Power BI Limits + Feature Comparisons	168
Individual File Size Limit (Per .pbix File).....	168
Power BI Workspace Object Limits	169
Scheduled Data Refreshes Per Day.....	169
V-Cores Per Premium Capacity.....	170
Concurrent Data Refreshes Per Premium Capacity.....	171
Streaming Dataset Limits	171
Push Dataset Limits.....	172
REST API Limits.....	172
Comparison of Power BI Free vs. Power BI Pro	173
Comparison of Power BI Service vs. Power BI Report Server vs. SQL Server Reporting Services	176
Comparison of Power BI Premium vs. Power BI Embedded	178
Section 13. Power BI Deprecated Items.....	181
Organizational Content Packs	181

V1 Workspaces in the Power BI Service	181
Get Data Experience in the Power BI Service	181
Azure Workspace Collections	181
Power BI CLI	181
Section 14. Getting Support from Microsoft and the Community	182
Support and Help	182
Learning More About Power BI.....	183
Community Tools	185
Additional Resources and Links	186

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 whitepaper is to help you make your Power BI deployment a success: it covers key considerations, the decisions which will be necessary throughout the process, and potential issues you may encounter. Best practices and suggestions are offered when possible.

The target audience for this whitepaper 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 July 18, 2018.

The following topics are out of scope, or limited scope, for this whitepaper:

- Detailed explanation of Power BI fundamentals, such how to create reports and dashboards or write DAX calculations.
- ISV deployment scenarios, which are handled differently from enterprise deployment scenarios. Some information about the Power BI Embedded service in Azure is included in this whitepaper for clarity.
- Performance tuning of reports, or of DAX or M expressions.

- SQL Server Reporting Services, beyond direct comparisons to Power BI Report Server.
- Third party solutions which integrate with Power BI.
- GDPR compliance in-depth, as it is covered in a [separate GDPR whitepaper](#).
- Power BI security in-depth, as it is covered in a [separate whitepaper](#).
- Features and functionality for the national clouds (US Govt, China, Germany). The guidance in this whitepaper is focused on the commercial cloud.



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 any information found online since it can quickly become out of date. Indeed, some information in this whitepaper may be out of date by the time you read it: this [What's New](#) page contains a reference list of major changes to features and functionality since this whitepaper was published.

Future Features

This whitepaper makes reference to future features, as a way to communicate meaningful changes to be aware of that are not available at the time of this writing.

This whitepaper purposely makes no estimation of release dates in conjunction with future features. Some of these features may be released very quickly after this whitepaper becomes available, and some may take longer to be released.

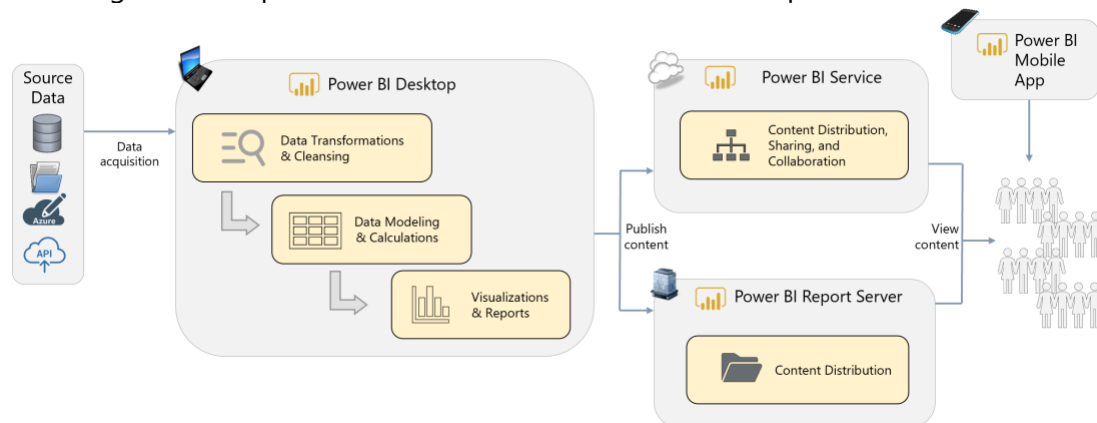


At the time of this writing, a transition is occurring from V1 workspaces (Office 365-backed workspaces) to a new workspace experience in the future. The concept of workspaces is a crucial to an enterprise deployment.

High Level Overview of Power BI

Power BI is collection of tools and services which facilitates the acquisition of data, data modeling, visualizations and reports, as well as the distribution of analytical solutions.

Following is a conceptual illustration of the most common capabilities:



[Section 2](#) reviews several Power BI usage scenarios which expand upon this high level illustration, and incorporate other components and features as well.

Power BI has grown rapidly to millions of users. Microsoft has been consistently ranked a leader on [Gartner's Magic Quadrant for BI & Analytics](#).

Definitions and Terminology

The most prominent components of the Power BI suite of tools are defined as follows:

Power BI	The ecosystem as a whole which includes a collection of software services, applications, and connectors that facilitate the creation of reports and dashboards and the visual analysis of information.
Power BI Desktop	Power BI Desktop is a desktop-based authoring tool of choice for connecting, transforming, and modeling data for creating interactive reports. There are two Power BI Desktop executables available: one which is updated monthly and aligns with the Power BI Service, and one which aligns with the capabilities of Power BI Report Server and is updated less frequently (approx. 3x/year). Files produced by both versions work with Power BI Service. Power BI Desktop saves reports as files in the .pbix format.

Power BI Service	<p>The Power BI Service is a cloud-based service that supports collaboration, sharing, enterprise distribution, creation of dashboards, and additional functionality such as alerts. Online resources may refer to it as:</p> <ul style="list-style-type: none"> • PowerBI.com • Power BI cloud service • Power BI web portal • Power BI web service • Power BI site
Power BI App Workspace	<p>A Power BI App Workspace is a collaboration area within the Power BI Service. An App Workspace is a distinct area, separate from My Workspace, which is dedicated to a specific team, subject area, or project. Power BI and Excel reports are published to an App Workspace so that colleagues can view the content, and contribute as appropriate.</p>
Power BI App	<p>A Power BI App is a set of packaged up content in the Power BI Service for the purpose of distributing related reports and dashboards to a larger base of consuming users.</p>
Power BI Mobile Apps	<p>Power BI Mobile Apps are native applications for iOS, Android, and Windows for viewing reports and dashboards on a mobile device.</p>
On-Premises Data Gateway	<p>The On-Premises Data Gateway is an agent installed within the corporate network to allow secure access to organizational data stored on-premises. There are two gateway modes: Enterprise mode and Personal mode.</p>
Power BI Premium	<p>Power BI Premium is an offering that provides dedicated resources for reports, dashboards, and datasets within the Power BI Service. This provides organizations more predictable performance, larger storage volumes, larger dataset sizes, higher refresh rates, and incremental data refresh. Premium also enables widespread distribution of content without requiring a Pro license for each read-only user. As discussed in the Licensing section, the purchase of a Premium P-series SKU also includes optional usage of the Power BI Report Server. Purchase of either the P-series or EM-series SKU also allows usage of the Power BI REST APIs targeted towards embedding scenarios.</p>
Power BI Report Server	<p>Power BI Report Server is 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 reports are deployed and delivered in an on-premises portal alongside Reporting Services reports, Excel reports, and Mobile reports.</p>

Power BI Embedded	Power BI Embedded is an Azure service which provides capacity resources for embedded analytics scenarios. Although the set of REST and JavaScript APIs are unified between the Power BI Embedded capabilities in Azure and the Power BI Service, they are separate and distinct offerings. Power BI Embedded (available via A-series SKUs in Azure) is for embedding visuals into custom applications.
-------------------	--

Each of the above components are discussed in much more detail throughout this whitepaper.

Section 2. Power BI Usage Scenarios

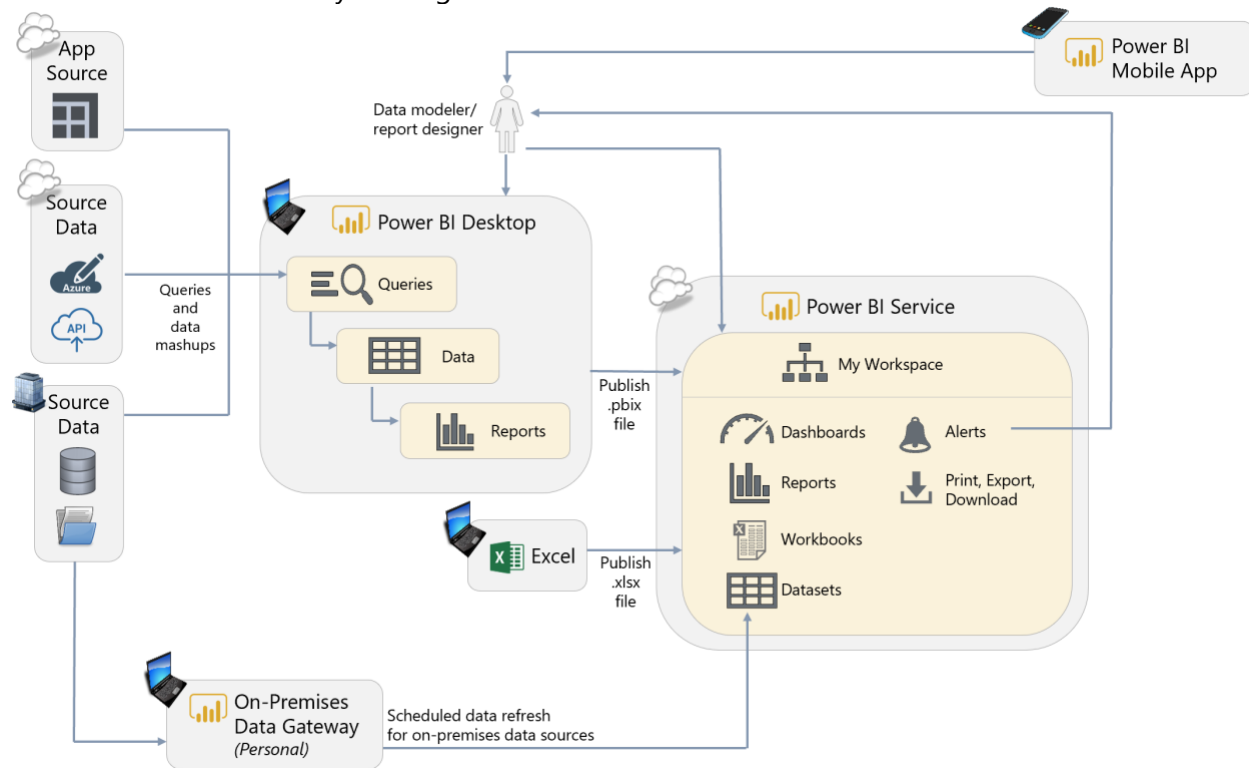
The Power BI ecosystem is very flexible and can be utilized in several different types of circumstances. This section presents high level usage scenarios to illustrate different ways you might deploy Power BI. After you have become familiar with the usage scenarios, the remainder of the whitepaper provides technical details you will need for a successful implementation.

In each of the following usage scenarios, the most prevalent components of Power BI are identified based on how Power BI is intended to be used. The objective here is not to call out every possible option for each usage scenario – rather, each diagram illustrates the primary features to investigate for each scenario. It is very possible you will “mix and match” ideas presented in this section to formulate a BI strategy that fits your situation.

Personal BI

With personal self-service BI scenarios, content is created for one’s own personal use with no sharing or collaboration involved. The user typically imports data into Power BI to create their own dataset and calculations, followed by reports and dashboards. In personal BI scenarios, the report author has a lot of freedom and flexibility. Many Power BI users begin using the product for personal purposes, then evolve to sharing their solutions with colleagues.

The following is a high level overview of the primary Power BI components to support personal self-service BI without any sharing or collaboration:



Power BI Desktop. Power BI Desktop is the authoring tool for queries, datasets, and reports.

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. With Power BI Free, all activities within the Power BI Service are constrained to within My Workspace. Features such as Analyze In Excel, Power BI Service live connections, and e-mail subscriptions are not available with a Power BI Free license. Although no sharing or collaboration is occurring within the Power BI Service in a Personal BI scenario, the Power BI Service is still very useful for functionality such as scheduled data refresh, dashboards, and data alerts.

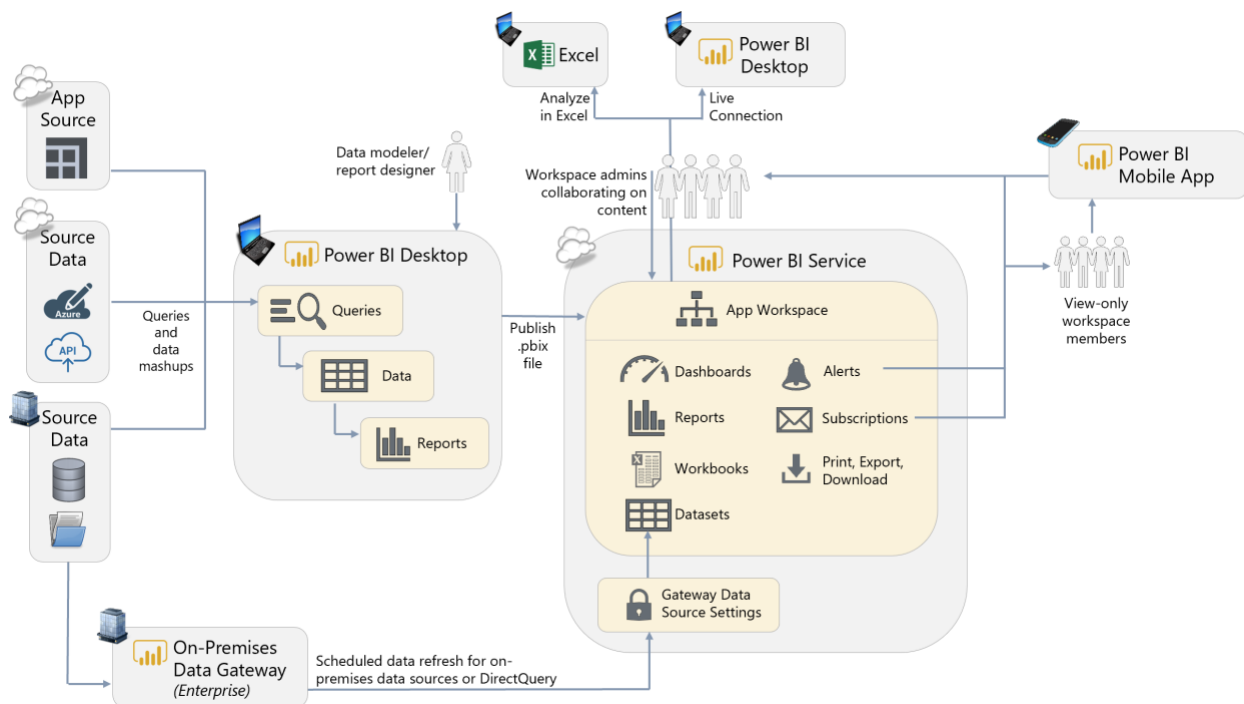
On-Premises Data Gateway: Personal. For refreshing datasets imported to the Power BI Service, an On-Premises Data Gateway is required for secure access to on-premises data sources which reside within the corporate network. The On-Premises Data Gateway in personal mode supports data refresh, but not live connection or DirectQuery operations. When in personal mode, the On-Premises Data Gateway is typically installed on an individual user's machine, and therefore is best-suited to personal usage scenarios only.

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

Small Team Collaboration

When a Power BI user has created a solution of value, it is time to collaborate with colleagues. This is sometimes thought of as a “bottom-up” approach wherein the subject matter experts in the functional business units drive the analytical initiatives for their team.

For small team scenarios with a group of users who all interact regularly, collaboration and distribution could be handled directly within an App Workspace as depicted in the following diagram:



Power BI Desktop. Power BI Desktop is the authoring tool for queries, datasets, and reports. For team-based scenarios, storing the original Power BI Desktop file in a location such as OneDrive for Business is recommended because it is easily accessible by others and has built-in history and versioning. The shared location also facilitates shared development among team members, for instance if one person works on the dataset and another works on the reports.

Power BI Service with Power BI Pro. For collaboration and distribution purposes, the Power BI Service is the option with the most flexibility and functionality. Although Power BI Desktop files can be shared among users via the file system or a document library, that is not the recommended approach because it is highly limiting. In addition to facilitating collaboration and distribution, the Power BI Service opens up additional features such as dashboards, scheduled refresh, data alerts, and e-mail subscriptions. When utilizing an App Workspace for collaboration (as shown above), all users require a Pro license even if they are only consuming reports.

App Workspace. A Power BI App Workspace serves as a container in the Power BI Service for related reports, dashboards, and datasets. The App Workspace is considered the development and collaboration area. In small team scenarios, it can be most practical to use the App Workspace for both authoring and collaboration, as well as the distribution and delivery of reports and dashboards. An App could optionally be used to deliver read-only content to users (as shown in the next scenario for large teams), but is not required.

Power BI Service Live Connection and Analyze in Excel. These two features are very important additions to a team collaboration scenario. These capabilities permit a live connection, from Power BI Desktop and Excel, respectively, to a dataset which has already been published to the Power BI Service. Usage of these capabilities is recommended (particularly over exporting to Excel) 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 while remaining pointed to the original data which resides in the Power BI Service

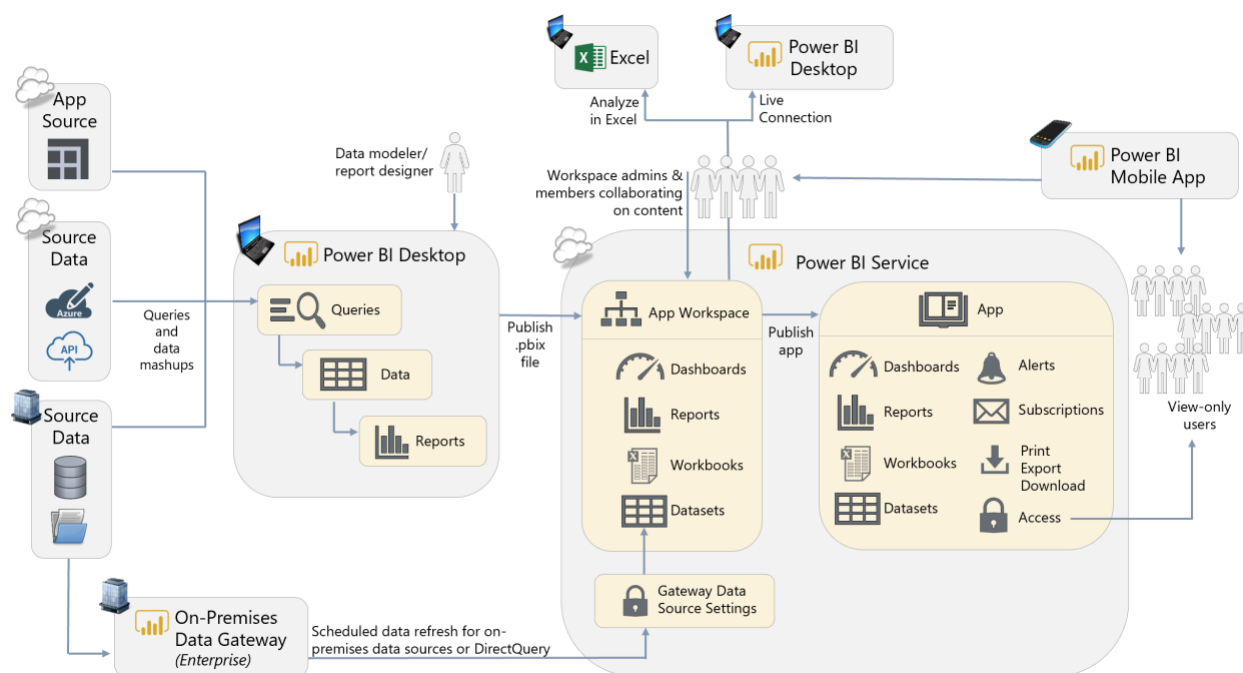
Currently, use of a Power BI Service live connection or Analyze in Excel requires the user to have edit privileges on the App Workspace where the dataset resides.

On-Premises Data Gateway: Enterprise. For refreshing datasets in the Power BI Service, an On-Premises Data Gateway is required for secure access to on-premises data sources which reside within the corporate network. A gateway is generally not required when accessing cloud data. The use of centralized On-Premises Data Gateways in enterprise mode is strongly recommended over gateways in personal mode for team/group scenarios. In enterprise mode, the On-Premises Data Gateway supports live connection and DirectQuery operations as well as data refresh.

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.

Large Team Collaboration and Distribution

When teams grow larger, it becomes impractical to use an App Workspace effectively for all report distribution (as was shown in the previous scenario). The best way to handle larger team-based scenarios is via the usage of App Workspaces for collaboration, and Apps for distributing the content to a larger group of people:



Power BI Desktop. Power BI Desktop remains the authoring tool for queries, datasets, and reports. As a team grows with multiple collaborators, processes should be developed to address the challenges of collaborating on the creation of a .pbix file. Storing the original Power BI Desktop file in a location such as OneDrive for Business which has built-in history and versioning continues to be recommended. As deployments grow larger, a source control system may also be put in place.

Power BI Service with Power BI Pro. For collaboration and distribution purposes, the Power BI Service is the option with the most flexibility and functionality. Although Power BI Desktop files can be shared among users via the file system or a document library, that is not the recommended approach because it is highly limiting. In addition to facilitating collaboration and distribution, the Power BI Service opens up additional features such as dashboards, scheduled refresh, data alerts, and e-mail subscriptions. When utilizing an App Workspace and Apps (as shown above), all users require a Pro license (the exception being read-only users of an App when its underlying App Workspace has been assigned to Premium capacity which is shown in the next scenario).

App Workspace + Power BI App. A Power BI App Workspace serves as a container in the Power BI Service for related reports, dashboards, and datasets. The App Workspace is the development and collaboration area, whereas the App is responsible for delivering content to users, as shown in the diagram above. For larger team situations, it is a best practice to limit access to the App Workspace to only the persons(s) handling authoring, development, and quality assurance activities – this is typically a small number of people. All read-only users then obtain access to the content via a Power BI App rather than directly via the App Workspace (which was shown previously in the small team scenario). An App becomes particularly useful in larger team situations where the user base is more diverse, and when you want to control explicitly when the content is redistributed to users (i.e., via updating of the App).

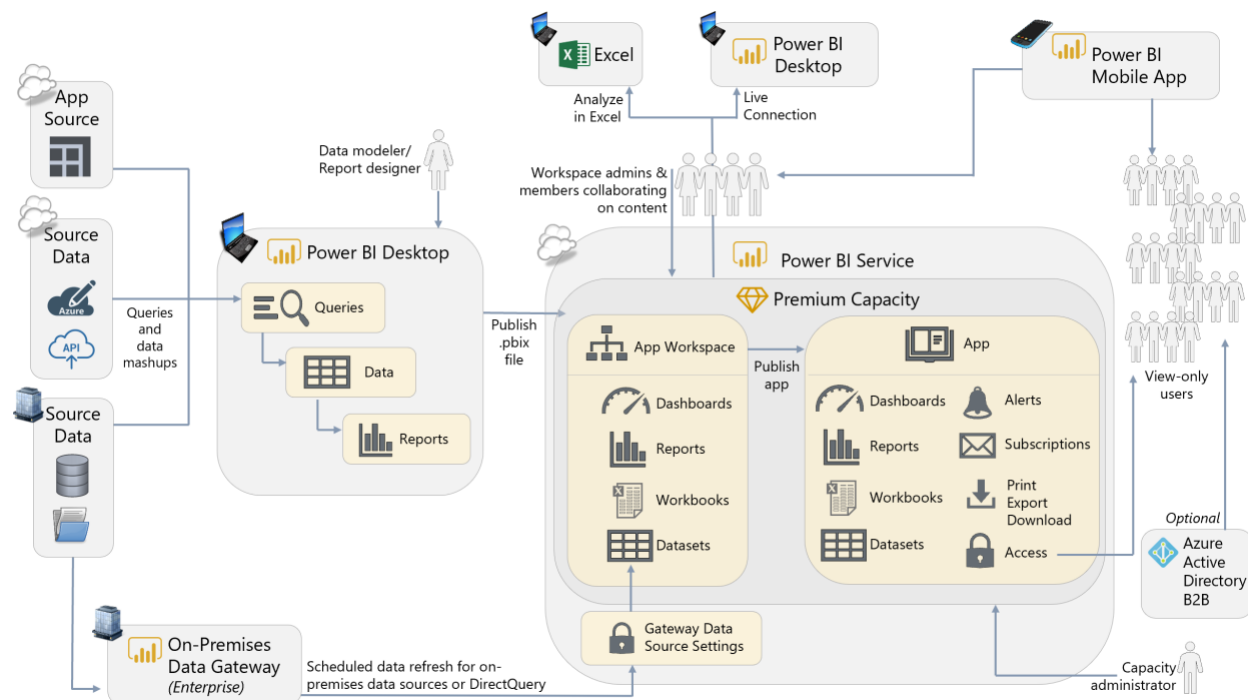
Power BI Service Live Connection and Analyze in Excel. As the team scenario continues to grow larger, the use of these capabilities (described in the previous section) is very important to minimize the duplication of datasets. However, since use of a Power BI Service live connection or Analyze in Excel requires the user to have edit privileges on the App Workspace where the dataset resides, these capabilities are only available for workspace admins and members with edit permissions (a future capability coming to the Power BI Service will address this limitation by permitting users to create a derivative report from an existing dataset).

On-Premises Data Gateway: Enterprise. For refreshing datasets in the Power BI Service, a data gateway is required for secure access to on-premises data sources which reside within the corporate network. A gateway is generally not required when accessing cloud data. The use of a centralized On-Premises Data Gateway in enterprise mode is strongly recommended over the use of gateways in personal mode for team/group scenarios. In enterprise mode, the On-Premises Data Gateway also supports live connection and DirectQuery operations as well as data refresh.

Enterprise Level Collaboration and Distribution

Enterprise-Level BI environments often employ a “top-down” approach because it is common for the Power BI content to be largely created and maintained by IT and/or a BI team.

Enterprise-Level reporting scenarios commonly have a significantly larger number of read-only users compared to a much smaller number of authors. A common way to cost-effectively support organizational/enterprise level reporting is via the use of Power BI Premium:



Power BI Desktop. Power BI Desktop remains the authoring tool for queries, datasets, and reports. For organizational-level reporting scenarios, usage of DirectQuery rather than importing the data into a dataset, or usage of live connections to a semantic layer such as Analysis Services, are common techniques (which are discussed in the Corporate Reporting scenario, next).

Premium Capacity in the Power BI Service. Some larger implementations may justify the use of dedicated Premium capacity in the Power BI Service. When there are a large number of readers or view-only consumers of reports and dashboards, the use of Free licenses afforded by Power BI Premium is a cost-effective alternative to purchasing Pro licenses for all users. Larger implementations may also find advantages from the larger storage limits, larger dataset sizes, higher refresh rates, and incremental data refresh.

App Workspace + Power BI App. A Power BI App Workspace serves as a container in the Power BI Service for related reports, dashboards, and datasets. The App Workspace is the development and collaboration area, whereas the App is responsible for delivering content to

users, as shown in the diagram above. For enterprise-level scenarios, it is a best practice to limit access to the App Workspace to only the persons(s) handling authoring and development – this is typically a very small number of people. All read-only users then obtain access to the content via a Power BI App rather than directly via the App Workspace. An App becomes particularly useful in enterprise-level situations where the user base is diverse, and when you want to control explicitly when the content is redistributed to users (i.e., via the re-publishing of the App).

Azure Active Directory B2B (Business-to-Business) or B2C (Business-to-Consumer).

(Optional.) If any users of your Power BI solutions are external to the organization, such as partners, vendors, and customers, they can be [invited to your AAD \(Azure Active Directory\) tenant as guest users](#). This includes not only business users with corporate account, but also consumer accounts such as *.outlook.com or *.gmail.com. Guest users will only be able to view content via a browser, as the Power BI mobile applications are not available to guest users.

Centralized Corporate Reporting and Data Warehousing

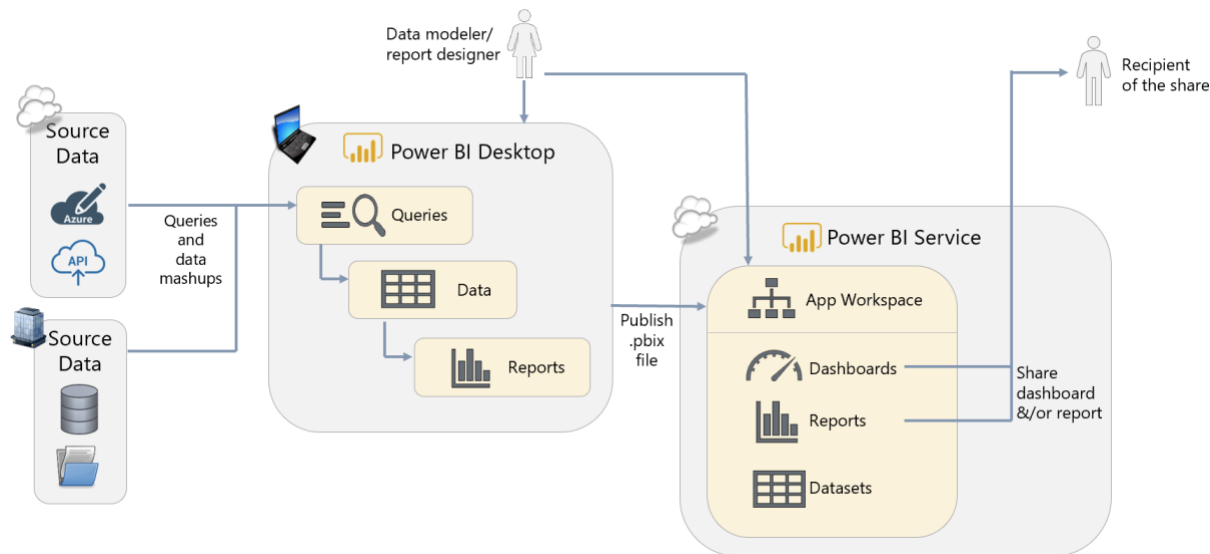
There are a number of options for delivering Corporate Reporting scenarios. This scenario illustrates one such option, in which Power BI can work like any other reporting tool which issues queries to retrieve data on-demand when a report is viewed. Centralized Corporate Reporting scenarios frequently occur in organizations where the data is centralized and maintained by a technical team, and the reporting and analysis capabilities are handled by the subject matter experts.

This Corporate Reporting scenario certainly has many characteristics in common with the Enterprise scenario discussed above. The intent of the previous Enterprise scenario was to focus on the distribution of a Power BI App to a large number of users via a Premium workspace, whereas this Corporate Reporting scenario focuses on how Power BI can reuse existing centralized data models stored in services like Analysis Services. The content can then be distributed broadly as described in the previous Enterprise scenario.

In this scenario, Power BI is being used as purely as a report development tool, in circumstances where the data model has been centralized (one option being Analysis Services, as shown in this diagram). Put another way, when Power BI does not contain a dataset, relationships, or calculations, the downstream report development side is significantly simpler. This approach facilitates delegating, or decentralizing, the report development activities via usage of centralized source data.

Prototyping Activities and Sharing

Power BI can be effectively used as a prototyping tool. Prototyping is very useful for confirming requirements for visuals and calculations with end users before formal requirements are created and/or development time is incurred to build the solution. Prototyping solutions are often temporary, short-lived solutions. Therefore, this scenario also illustrates the usage of sharing functionality for collaboration purposes (as opposed to App Workspaces and Apps covered in the previous scenarios).



Power BI Desktop. The authoring tool can be used during interactive prototyping sessions conducted with business users and subject matter experts. Prototyping sessions are 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 preliminary results for feedback and decision-making purposes. It is recommended that you segregate a prototyping solution into its own App Workspace, so it is clear that it's not production-ready.

App Workspace. An App Workspace is appropriate in this scenario since we are dealing with a small team-based collaboration scenario. With an App Workspace in place, once the solution has been finalized it can quickly be promoted to production, or operationalized, by publishing to the appropriate App Workspace, then creating an App for delivery of content to read-only users.

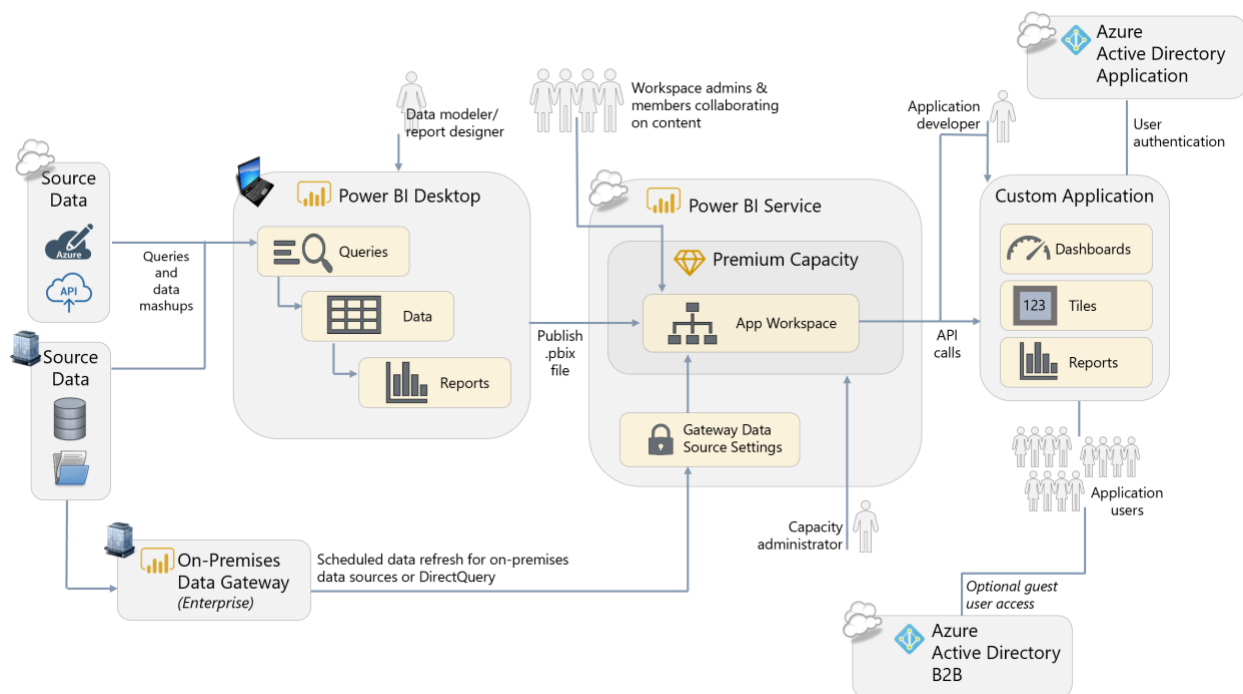
Sharing Reports & Dashboards. The above diagram depicts sharing directly to a recipient (without usage of an App). Sharing is appropriate for collaboration scenarios when colleagues work together in an ad hoc way, and typically works best when limited to a small number of

colleagues who primarily review content or need access to it. Another valid alternative, not depicted, is to specify the colleague as a member in the App Workspace (as was shown in the small team collaboration scenario). Adding colleagues to an App Workspace works best when those colleagues need access to many pieces of content in the App Workspace, need to provide new or updated content, and have reason to see all the content of the App Workspace.

Organizational Embedded Analytics

[The Power BI REST APIs](#) can be used for securely embedding dashboards and reports into custom applications, SharePoint Online, Microsoft Teams, or even in a third party application such as Salesforce. Making analytical data easier for business users to access, by integrating visuals in the tools and applications they use every day, allows data-driven decision-making to become more prevalent and improves workflow efficiency for functional users.

There are two main techniques for delivering embedded analytics with Power BI. This first technique describes the usage of Power BI Premium (P or EM SKUs) for internal/organizational users. In smaller organizations where all users have Pro licenses, or in organizations covered by Office 365's E5 license, organizational embedding works for all users who have Pro licenses.



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. This is beneficial because a business user who is the subject matter expert can change content and publish to the Power BI Service with no changes to their normal practices. If row-level security is required, it is defined in Power

BI Desktop using the same technique as if the reports were to be delivered in the Power BI Service.

Power BI Premium. A Premium subscription allows the usage of the Power BI REST APIs for embedding scenarios which are primarily organizational in nature. Specifically, this scenario refers to the P and EM SKUs which let you distribute embedded content to authenticated AAD users without the user having a Power BI Pro license.

Power BI App Workspace. The .pbix file does need to be published to the Power BI Service so the APIs can render report items in the custom application. The reports, dashboards, and datasets should be published to a specific Power BI App Workspace which has been designated for the embedded analytics project, or is designated for collaboration in this manner. Note that publishing to an App, however, is optional for this scenario - the App Workspace is sufficient since the visuals will be rendered in the custom application and accessed by the users in the application. However, if you prefer to embed from the App, that can ease permissions management for all related content (instead of individually sharing content).

Power BI Report Web Part for SharePoint Online. (*Not depicted in the above diagram.*)

Often organizations create team or topic-specific portals in SharePoint Online to enable collaboration. The Power BI Report web part for SharePoint Online makes it easy to use embedding without building a custom application.

Authentication and Authorization. When embedding content in an organizational scenario (as opposed to utilization of Power BI Embedded described in the next scenario), every user is authenticated to Power BI to view the data embedded in the custom application. Therefore, authentication to Azure Active Directory (AAD) is based on the user's individual AAD account. This allows the embedded content to leverage the same row-level-security configured for your analysts and report consumers in Power BI.

AAD Application. When embedding content in an organizational scenario (as opposed to utilization of Power BI Embedded described in the next scenario), in order to sign as in the user via the REST APIs, an Azure Active Directory application is required.

On-Premises Data Gateway. For refreshing datasets imported to the Power BI Service, a data gateway is required for secure access to data which resides within the corporate network. The use of a server-based On-Premises Data Gateway in enterprise mode is strongly recommended over the use of personal mode for supporting enterprise-level refresh scenarios.

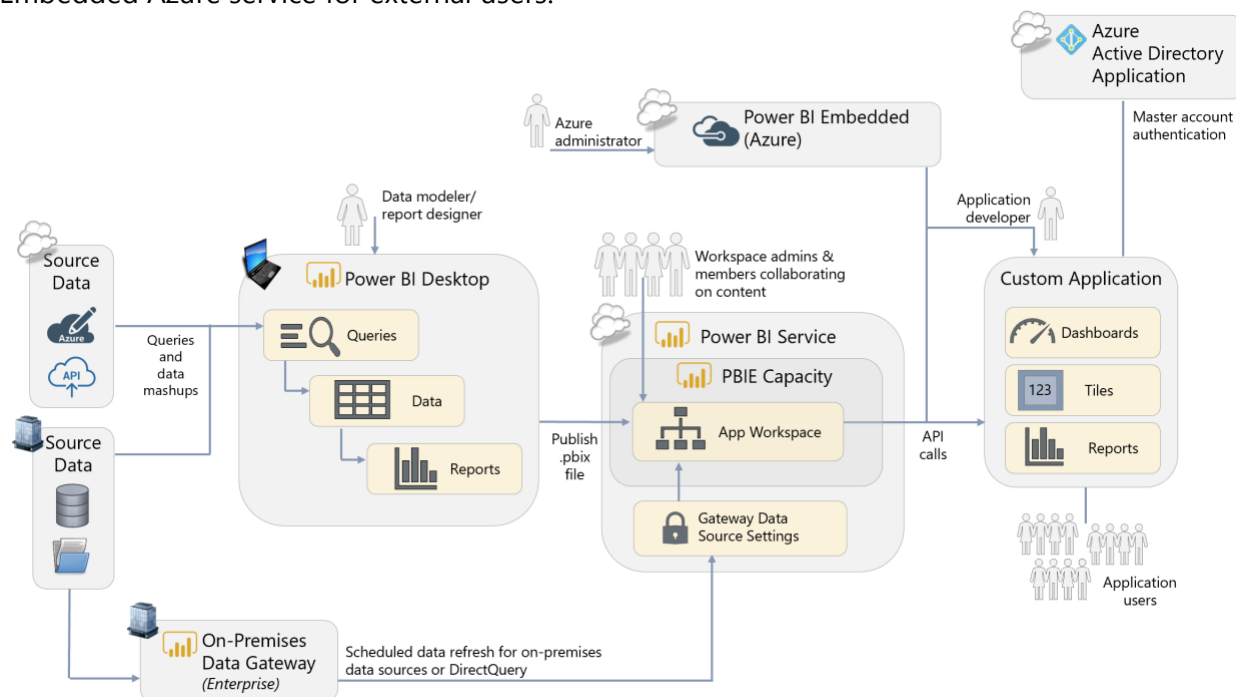
Azure Active Directory B2B (Business-to-Business) or B2C (Business-to-Consumer).

(Optional.) If any users of the application are external to the organization, they can be [invited to your AAD \(Azure Active Directory\) tenant as guest users](#). This includes not only business users with corporate account, but also consumer accounts such as *.outlook.com or *.gmail.com. Guest users will only be able to view content via a browser, as the Power BI mobile applications are not available to guest users.

External Embedded Analytics

[The Power BI REST APIs](#) can be used for securely embedding Power BI visuals, tiles, and Q&A into custom applications which are used externally by customers and partners.

There are two main techniques for delivering embedded analytics with Power BI. The first technique was discussed in the previous section. This second technique, commonly used with independent software vendors (ISVs), describes the usage of the A SKUs in the Power BI Embedded Azure service for external users:



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. This is beneficial because a business user who is the subject matter expert can change content and publish to the Power BI Service with no changes to their normal practices. If [row-level security](#) is required, it can be defined as well.

Power BI Embedded. A subscription in Azure to Power BI Embedded allows the usage of APIs for external customers embedding and ISV embedding scenarios. Specifically, this scenario refers to the Power BI Embedded A SKUs available in Azure. The Power BI Embedded service can be paused when not in use (during which no costs are incurred), which can be useful for dev/test scenarios. It can also be scaled up or down to handle the peaks and valleys of varying workload levels.

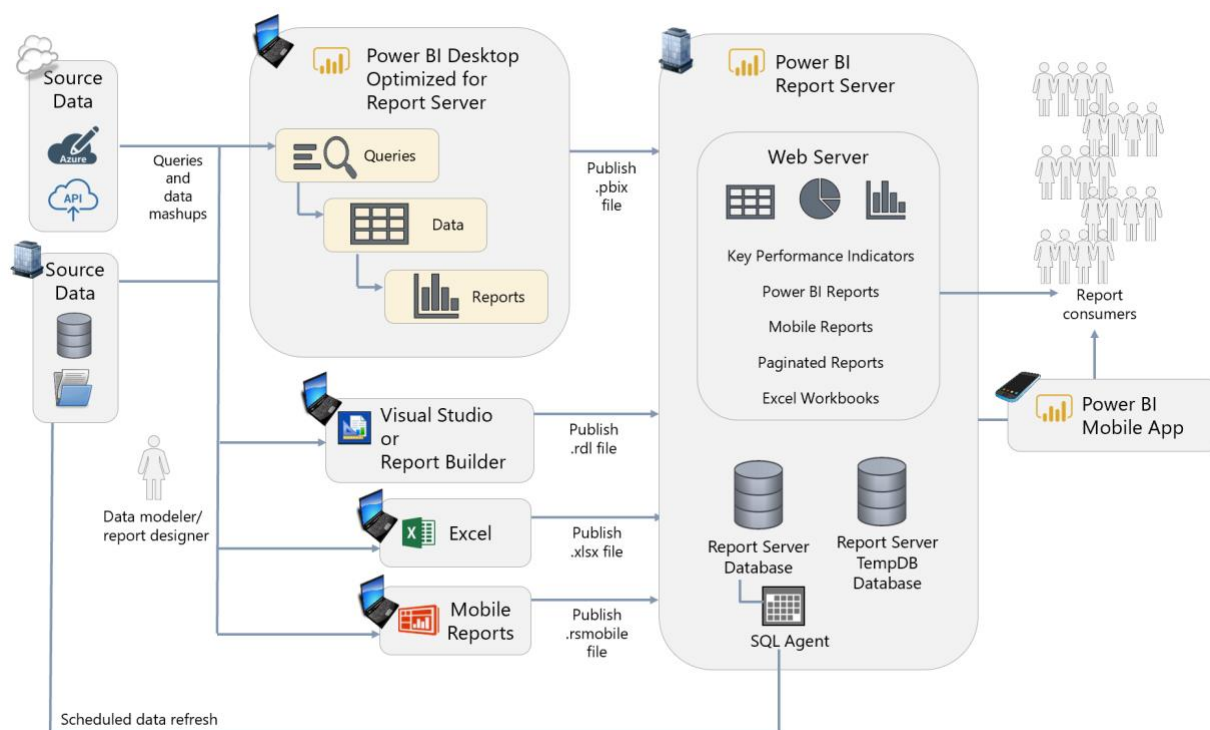
Power BI App Workspace. The .pbix file does need to be published to the Power BI Service so the APIs can render report items, dashboards, and tiles in the custom application. The reports,

dashboards, and datasets should be published to a specific Power BI App Workspace which has been set up for the embedded analytics project. Note that publishing to an App, however, is completely unnecessary for this scenario - the App Workspace is sufficient since the content will be rendered in the custom application and accessed by the users in the application.

Authentication and Authorization. When embedding content via use of Power BI Embedded, the application user is authenticated and authorized by the application using any means desired. The application end user does not have a Power BI account, nor is assigned a Power BI license. The custom application makes the connection to Power BI, which it does through Azure Active Directory based on the one master account configured for the application which has been assigned a Power BI Pro license (which differs from user authentication described in the previous scenario in conjunction with Power BI Premium). The custom application uses the Power BI REST API to generate an embed token for each object that is embedded and shown to the user.

On-Premises (Non-Cloud) Reporting Portal

One benefit of Power BI Premium (also available with Enterprise SQL Server licenses with Software Assurance) is the ability to install, and publish Power BI content to, a Power BI Report Server which resides within the corporate network. This capability is useful for customers who wish to deploy to an on-premises server within their corporate environment:

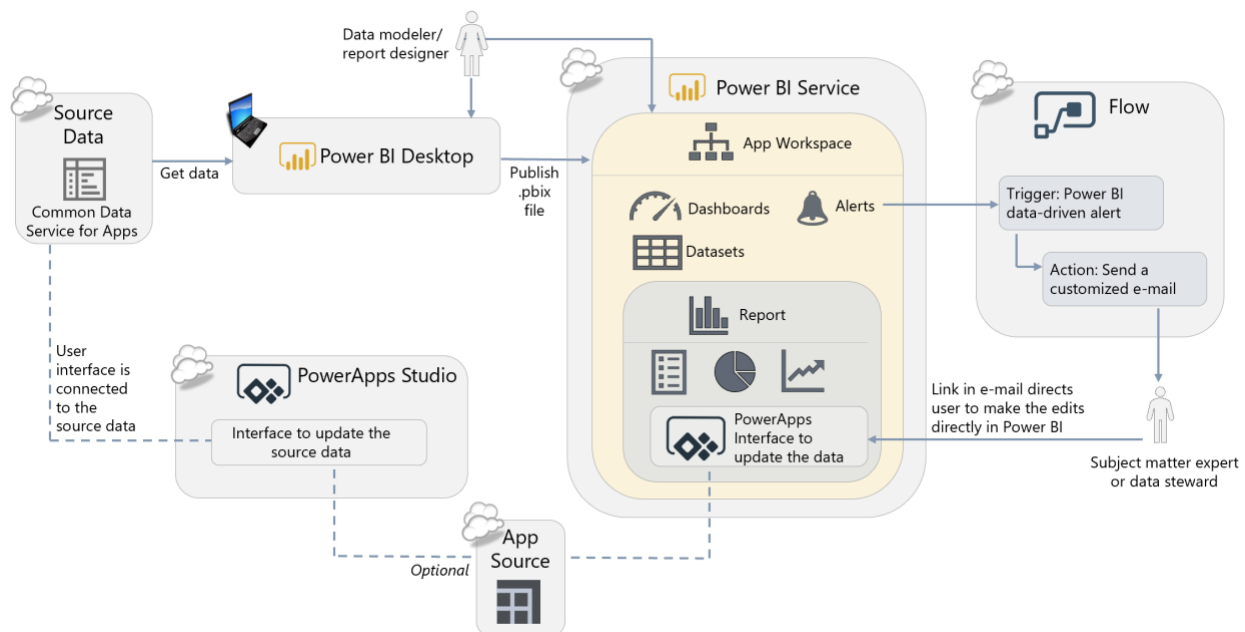


Power BI Desktop Optimized for Report Server. The authoring experience remains the same for authors. However, Power BI Desktop authors must utilize a version of Power BI Desktop which is designated as being compatible with the Power BI Report Server release cycle. Updates are released approximately 3 times per year.

Power BI Report Server. The Power BI Report Server offers a repository and user portal for delivering reports from not only Power BI, but also from Reporting Services, Excel, and Mobile Reports (previously known as Datazen).

Workflow Automation + Updating Data from Within Power BI

The family of tools that is comprised of Power BI, Flow, and PowerApps allows for many interesting use cases that involve integration between these three business applications. In the following scenario, envision that we are checking for data quality issues or missing data. Flow is used to send a customized e-mail alert which directs a user back to Power BI to update some source data:



Power BI Desktop. When obtaining data from the Common Data Service for Apps (which is where the data for PowerApps is stored in this scenario), the Get Data experience within Power BI Desktop is the same as usual. Creation of queries, calculations, datasets and reports all work as expected.

Alerts in the Power BI Service. Alerts in the Power BI Service can be used to identify when an issue exists. This could be many things, such as missing data, invalid data, presence of a critical situation, or exceeding a threshold. Alerts can be set up on dashboard tiles, and are fired when data is updated in Power BI. Because alerts are defined on a threshold, alerts can only be defined on certain types of dashboard tiles which display a singular value such as gauges, KPIs, and cards.

Flow. Microsoft Flow is an Office 365 service for workflow automation between services. Used in conjunction with alerts in Power BI, Flow can be utilized for many types of workflow automation, such as:

- Send a customized e-mail alert or text message
- Create task in tools such as GitHub, Trello, or a SharePoint list
- Post a message on forums such as Teams or Slack
- Request an approval
- Trigger another workflow

In the above diagram, Flow is being used to send a detailed, data-driven message to the subject matter expert once the Power BI alert has triggered the Flow to kick off.

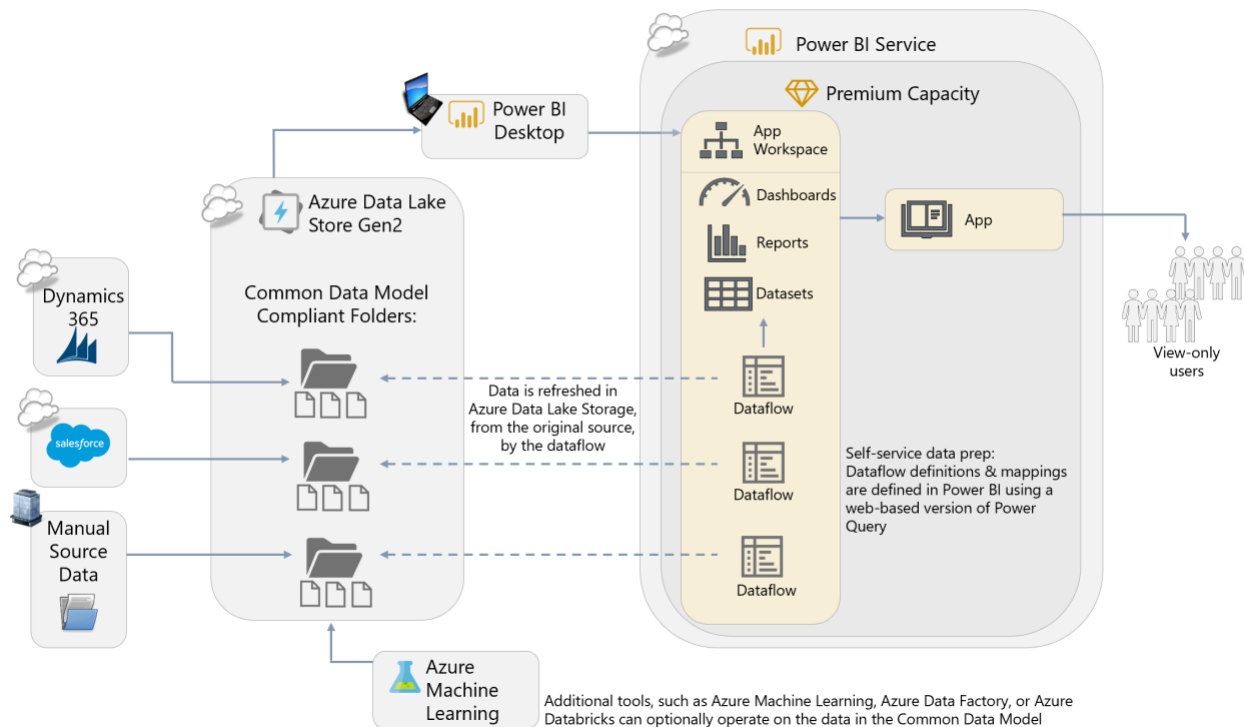
PowerApps. PowerApps is an Office 365 service for creating small line of business applications and user interfaces in a low-code/no-code way. PowerApps can be used for many things, such as:

- Task and issue tracking
- Budget input
- Cost estimations
- Surveys

Within the e-mail message shown in the above diagram, a URL is provided which directs the recipient back to Power BI to make the data updates – this is for user convenience since the user is frequently in Power BI anyway. The user interface being exposed in Power BI is backed by PowerApps, so the data updates could certainly be made directly in PowerApps if preferred. When updates are done via the interface in PowerApps, the source data is immediately updated; the updates will be reflected in Power BI the next time a data refresh operation is executed.

Self-Service Data Preparation With Dataflows

Dataflows are a future new capability. Their objective is to integrate data from disparate sources into a Common Data Model (CDM) which is easy for users to customize, and easy for Power BI to consume and analyze.



Power BI Desktop. To consume data output from dataflows, use the standard 'Get Data' experience within Power BI Desktop. Creation of queries, calculations, datasets, and reports all work as expected in order to create customized reports from data residing in the Common Data Model. Once published to the Power BI Service, data refresh works the same, as do publishing activities.

Dataflows. Data becomes inherently more valuable once it has been integrated with related data. Accordingly, dataflows are intended to make data integration from various sources easier. Dataflows are a new workspace artifact, similar to datasets, reports, and dashboards. Each dataflow is mapped to an App Workspace; an App Workspace can contain multiple dataflows. A pre-defined Common Data Model (CDM) is available for the common sources, such as Dynamics and Salesforce shown in the diagram above. Entities can be extended or new custom entities can be created as well, which enable the dataflow to adapt to an organization's unique needs, while preserving the built-in relationships that make analysis easier. A key advantage of dataflows is that the data integration work and mapping into the Common Data Model is done only once,

and then reusable among many Power BI solutions. Entities can be linked, to avoid loading the data multiple times.

Note that Dataflows were previously known as the Common Data Model for Analytics, or datapools, during the private preview period.

Azure Data Lake Storage Gen2. The output from a dataflow is mapped data to the Common Data Model, which resides in a CDM-compliant folder in Azure Data Lake Storage Gen2 (a folder in the Data Lake Storage is considered 'compliant' due to the presence of an additional file which contains schema and metadata information associated with the data file). The default state is usage of a built-in storage account provided to the customer; alternatively, a data lake account within an existing Azure subscription can be used if preferred. Most large organizations will prefer using their existing Azure subscription.

Power BI Premium (optional). Usage of dataflows in conjunction with Power BI Premium will open up capabilities such as refreshing computed entities, incremental data refresh, and usage of larger datasets. It is possible to get started using dataflows without Premium, and evolve into using Premium as the solution grows.

Power BI Service Tenant Settings (not depicted above). Within the Admin portal, the extensions related to connectivity to Common Data Model compliant folders in Azure Data Lake Storage Gen2 need to be enabled.

Power BI Apps. Optional pre-packaged apps contain reports to help users get a jump start on reporting from third party services such as Dynamics and Salesforce. The Power BI Apps utilize data from the Common Data Model in dataflows, and can be customized as needed (for instance, to integrate with additional data loaded via dataflows).

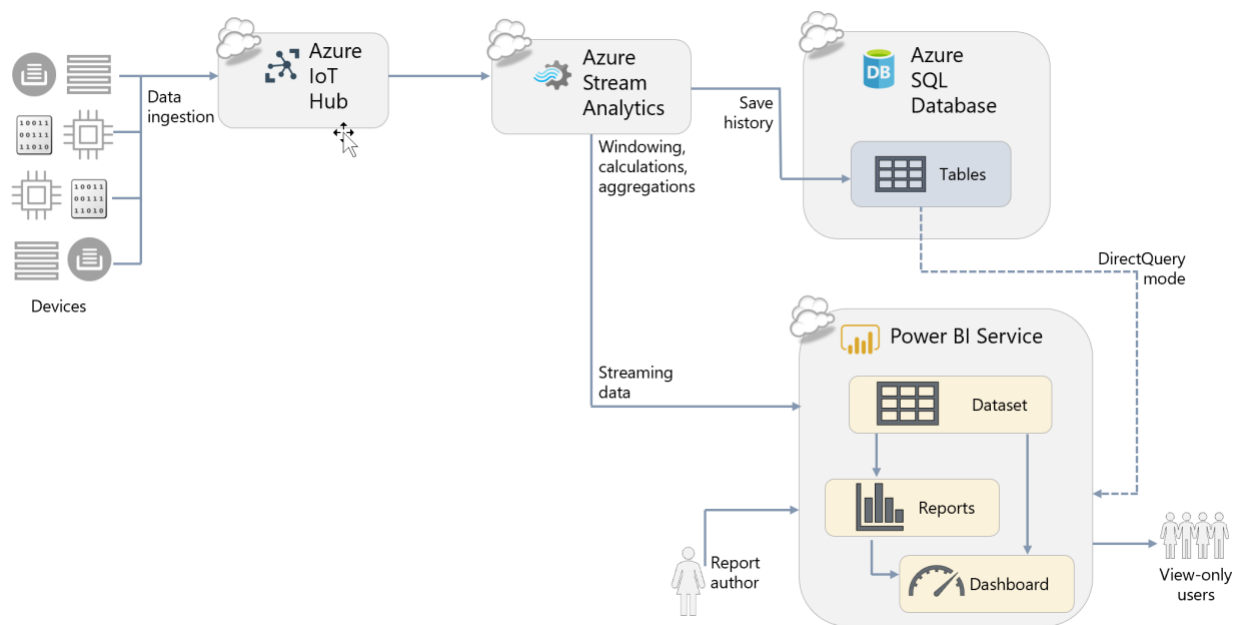
Azure Machine Learning. Because the data resides in Azure Data Lake Storage Gen2, other tools can also operate on the data including tools such as Azure Machine Learning, Azure Data Factory, or Azure Databricks.

Streaming and Near-Real-Time Reporting

Streaming data solutions have been gaining popularity. Access to near real-time data allows 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 6](#) for additional 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 REST APIs, Azure Stream Analytics, or PubNub.

There are numerous ways to accomplish a streaming solution. Following is one example which utilizes Stream Analytics:



Power BI Service. The capabilities for visuals and data retention available for streaming and near real-time solutions vary greatly based on which technique is utilized.

Azure IoT Hub (optional). In the above diagram, Azure IoT Hub serves as the ingestion point in Azure. There are numerous alternatives for handling streaming data solutions, including Azure Event Hub, Kafka, and Storm.

Azure Stream Analytics (optional). In the above diagram, Azure Stream Analytics is utilized as the event processor. It handles required data processing tasks in-flight such as aggregations, or windowing calculations. It also handles distribution of data to two downstream paths: one path to support a near-real-time streaming dashboard, and one path to persist the data.

Power BI Dataset. When storing near-real-time data in a Power BI dataset, if it is configured for Hybrid or using the push data API, the data is available for monitoring through dashboards or analysis through reports. The Power BI dataset will store the latest near-real-time data.

Azure SQL Database (optional). For longer term data retention and analysis capabilities, the streaming data should be mirrored to a database. Azure SQL Database is shown above for this purpose, though numerous options are available.

Power BI Dashboard. To view data in near-real-time, it is necessary to utilize dashboard tiles which reference the Power BI dataset. Depending on the frequency of near-real-time data, the visuals update up to once per second.

Power BI Reports. Near-real-time data can be used for analysis directly over top of the Power BI dataset. In reports, the data does not update live since changing data can make it difficult to complete analytical tasks. However, the data is updated whenever a user interacts with visuals on the report page or presses the refresh button.

Section 3. Power BI Architectural Choices

This section discusses three of the most important decisions you must make when planning a Power BI enterprise deployment:

- The decision whether to use the Power BI Service, to use Power BI Report Server on-premises, or to use both.
- The decision whether to use Power BI Premium.
- The decision whether to use Power BI Embedded.

To help you make an informed choice this section describes the Power BI Service, Power BI Report Server, Power BI Premium, and Power BI Embedded, including reasons why you might consider using them.

The Power BI Service

What is the Power BI Service?

The Power BI Service is a central repository for your organization where your users can:

- Publish data, reports, dashboards and other business intelligence-related content that they have created.
- Schedule the refresh of published data.
- Share and distribute this content securely with other people both inside and outside your organization.
- Discover and access data, reports, dashboards and other business intelligence-related content that has been shared with them.

The Power BI Service appears to users as a web-based portal, but it is much more than that. Among other things, it can:

- Distribute reports and dashboards to large numbers of users
- Store and manage the data used by reports
- Manage access to external data sources
- Connect to cloud or on-premises data sources
- Apply security both to reports and the underlying data
- Generate alerts
- Set up subscriptions to inform users that new content is ready to be viewed by email
- Enable access to analytics through mobile devices using rich device-specific applications
- Centralize administration and governance activities

The Power BI Service is organized into workspaces: when content is published to the Power BI Service it is published either to an App Workspace, or to a user's personal workspace (called My Workspace).

Why Use the Power BI Service?

Your users may be used to storing the reports they create – for example in Excel - on their own PC and then distributing them by email, or storing them on a network file share or a SharePoint folder where other people can access them. Reports created in Power BI, saved as Power BI Desktop (.pbix) files, could be distributed in the same way but there are many reasons why these methods should be abandoned in favor of the centralized approach that the Power BI Service offers, including:

- The storage of reports in a variety of locations makes it difficult for users to discover all of the content that is available to them, and for organizations to enforce security or compliance policies.
- Different versions of the same report may exist in different places making it difficult for users to know which one is the one that is correct. Distributing reports as files makes it very likely that individual users will alter calculations or even the source data.
- Distribution and updating of reports is usually a very time-consuming, manual process and one that often breaks down when the person responsible is ill, on vacation or leaves the organization.
- When a .pbix file is opened within Power BI Desktop, much more than just a report is visible to the user: they can also see raw data, calculation definitions and other technical details of the implementation. For executives and consumers, this can be confusing and require extra training to deal with.
- Row-level security cannot be implemented when delivering reports via Power BI Desktop alone.

The Power BI Service addresses all of these problems, making it easy for users to discover the information they need to do their job and providing a place where all users can collaborate on, share, and distribute content in a secure and reliable manner.

Most users of Power BI use the Power BI Service exclusively, although as discussed below some prefer to use Power BI Report Server instead or use the two in combination. The following are reasons why you may want to consider using the Power BI Service in your Power BI deployment:

Fastest setup and least maintenance

Once you have decided to use Power BI, the Power BI Service can be up and running for your organization very quickly and you can start to share content. The Power BI Service includes data storage and connectivity to services such as Dynamics and Salesforce – this allows business users to quickly start using data to make decisions without waiting for IT to provision data sources or integrate data. As a cloud service, Microsoft handles all maintenance, updates, and patching activities.

Lower up-front costs

All licensing options relating to the Power BI Service are charged on a monthly basis (although there may be some options such as Power BI Premium that have an annual commitment too), so the up-front costs are smaller to a server-based approach that requires the purchase of software licenses and hardware.

Pay for what you need when you need it

Unlike a server-based approach, the Power BI Service does not require you make any difficult decisions about the number of users or reports or the amount of data you will be working with in advance. As your usage of Power BI grows you can purchase additional licenses and services to support your growth when you need them.

Secure distribution and collaboration

The Power BI Service is available in the cloud, allowing your users to access content from anywhere through device-specific mobile apps, the web, or customized portals. This allows your users to use data effectively wherever they are located. The insights can be shared within the organization, or securely with external partner organizations.

Governance, administration, and usage monitoring

When the Power BI Service is utilized as the destination for publishing user reports and dashboards, it gives administrators a single, centralized place to enforce governance and support users. This also permits [auditing of user activities](#), and understanding how solutions are (or are not) being utilized is crucial to a successful deployment. Additionally, Power BI administrators have a variety of [options available to govern use of the environment](#).

Access to the full functionality of Power BI

The Power BI Service provides access to the full functionality of Power BI and includes several features (such as Q&A, dashboards, QuickInsights, alerts, and streaming data) that may not be available in Power BI Report Server. As discussed [later in this section](#), the main objective for Power BI Report Server is to provide an on-premises organizational reporting portal. The

product vision for Power BI Report Server is not to support the same broad range of scenarios that the Power BI Service does.

Power BI Service Release Cycle

Updates to the Power BI Service are released frequently, usually on a weekly basis. New features and improvements are announced on the official [Power BI blog](#) and in described in more detail in the documentation. The goal for the frequent release cycles is to continually enable greater productivity.



Some preview features in Power BI Desktop may not be supported in the Power BI Service immediately.

Power BI Report Server

What is Power BI Report Server?

Power BI Report Server is a server-based product that is the on-premises alternative to the Power BI Service. Like the Power BI Service, it provides a central repository for publishing, sharing and consuming business intelligence content. Although it is built on top of SQL Server Reporting Services and has a superset of its functionality, it is a distinct product from SQL Server Reporting Services and is licensed and installed separately.

Why Use Power BI Report Server?

The following are reasons why you may want to consider using Power BI Report Server:

On-premises only solution or a complement to the Power BI 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, some organizations require that certain reports remain within the corporate firewall. These reports could be deployed to the Power BI Report Server. Some organizations mix and match, using both Power BI Report Server and the Power BI Service in order to utilize its broad feature set.

- ✓ Pro: Allows customers to have flexibility and choice.
- ✗ Con: If both the Power BI Service and Power BI Report Server are deployed, users will have to visit two different portals to access content.

Power BI report 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, Power BI Report Server offers an option for publishing and securing reports.

- ✓ Pro: Makes available an option for customers whose data must remain on-premises for regulatory or legal reasons
- ✓ Pro: More granular permissions are available in Power BI Report Server than the Power BI Service

Build on existing investment in Reporting Services

There are many thousands of customers who have 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 Power BI reports.
- ✗ Con: Additional Power BI Pro licenses may be required in order to publish Power BI reports. Power BI Pro is required for anyone who publishes Power BI reports, whereas 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 (Power BI Service) or on-premises (Power BI Report Server).

Power BI Report Server Release Cycle

Power BI Report Server releases occur approximately 3 times per year which coincide with a specific Power BI Desktop release that is optimized for the Power BI Report Server. Support for Power BI reports in Power BI Report Server is backward-compatible but not forward-compatible.



Power BI Report Server follows a modern lifecycle policy, which means that updates are released more frequently than is customary with on-premises server products. The Power BI Report Server updates will not be auto-installed. It is recommended that you plan to keep current as per the [support timeline for Power BI Report Server](#).

Comparing Power BI Report Server Functionality with the Power BI Service

Although Power BI Report Server and the Power BI Service are developed by the same team and share much of the same code base, there are substantial differences in the functionality of the two products. The differences include:

- Unlike the Power BI Service, Power BI Report Server does not support the creation of dashboards (which, to be clear, are not the same thing as reports in Power BI).
- Power BI Report Server does not have support for row-level security in imported datasets, though that is a future roadmap item.
- Some custom visuals that rely on cloud services, such as the ArcGIS map visual, are not available in Power BI Report Server.
- The methods for securing and distributing reports used by the Power BI Service and Power BI Report Server are very different.

As a result, you should verify that critical features you intend to use are present in whichever one, the Power BI Service or Power BI Report Server, that you intend to deploy.

A more detailed comparison of functionality can be found in [this table in Section 12](#).

Power BI Premium

What is Power BI Premium?

Power BI Premium is an offering for organizations with larger numbers of read-only users and/or larger data volumes, and/or more demanding data refresh requirements. As such, Power BI Premium is aimed squarely at the enterprise market. It provides the following high-level benefits:

- **Higher limits** in terms of the size of an individual dataset and the overall amount of data that can be stored in Power BI.
- Access to functionality that supports working with **larger data volumes**.
- **Isolated, dedicated hardware**, operating in single tenancy mode rather than multi-tenancy.
- Additional **control** over what goes on inside Power BI, for example in data refresh.
- More **consistent query performance**, especially when there are a large number of users viewing reports and dashboards.
- More **cost-effective** licensing options for scenarios where there are large numbers of users who only need to view (rather than create their own) dashboards and reports. In certain scenarios Premium can allow these users to access content with just a Power BI Free license, without having to pay for a Power BI Pro license.
- The option to use **Power BI Report Server** on-premises.



Purchasing Power BI Premium will **not** affect the amount of time it takes for a report to render for a single user. Scale, control, consistency, and cost-effectiveness are the main objectives for Premium. Although performance may be more predictable with Premium, improved performance is not a primary objective.

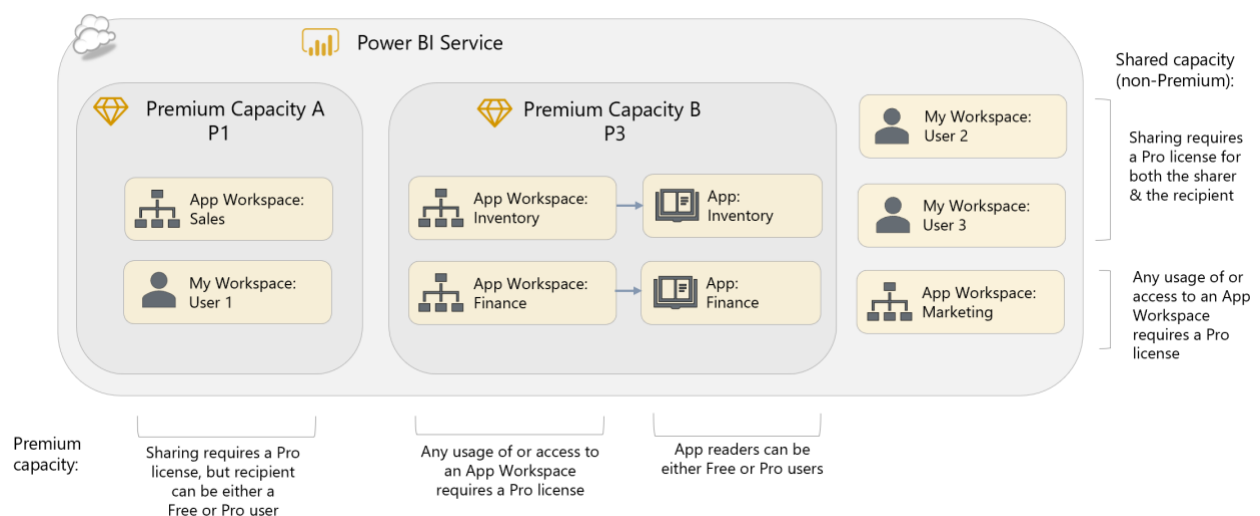
To help you understand these benefits, the remainder of this section is devoted to understanding the technical details of how Premium works while the licensing benefits are explored in detail in [Section 4](#). Throughout the rest of this whitepaper there will be frequent references to how Premium enhances various aspects of Power BI. A discussion of managing Power BI capacity can also be found in the [Managing Power BI Premium](#) section of this whitepaper.

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, and accordingly Microsoft has implemented a series of limits and throttles to ensure that everyone receives the same high quality of service. Examples of these limits and throttles include those on the maximum size of a dataset, the maximum amount of data that can be held in a workspace, and the maximum number of dataset refreshes that can take place in a day.

When you purchase Power BI Premium, you purchase one or more Power BI Premium capacity nodes. A Power BI Premium capacity node can be thought of as something like a virtual machine: it is a set of resources within the Power BI Service that can only be used by your organization. Since these resources are no longer shared, the limits and throttles that exist in Power BI shared capacity can be relaxed or even removed altogether. After a capacity node has been purchased, the resources it provides must be configured as one or more Premium capacities within the Power BI Service. These Premium capacities may then be used to host workspaces. When a workspace is moved from shared capacity to a Premium capacity it can take advantage of the fact that the limits imposed in shared capacity are relaxed or removed.


The following is a depiction of how workspaces can be allocated to shared capacity and Premium capacities within the Power BI Service:



In the above diagram, Capacity A contains a Power BI App Workspace called Sales which has a demanding workload; Capacity A also supports a “My Workspace” area for User 1, which is a private workspace where User 1 can create and manage personal reports or dashboards. An Inventory App and Finance App are supported by Capacity B, whereas User 2, User 3, and the Marketing App Workspace remain in shared capacity (i.e., the standard multi-tenant service).

Buying Power BI Premium Capacity Nodes

Power BI Premium capacity nodes can be purchased by visiting the Power BI Admin Portal. Click on the Premium Settings pane, then the “Buy” (or “Purchase More”) button. This redirects the administrator to the Office 365 Admin Portal where the purchase takes place.

 Only an Office 365 Global Administrator or Billing Administrator can purchase Power BI Premium capacity.

Premium Tiers and SKUs

Power BI Premium is available in two tiers for organizational use, each of which are divided up into several SKUs. Each of the two tiers is intended for different use cases. They are:

1. **EM-series**, which is intended for use in scenarios where Power BI reports and dashboards are embedded in organizational portals or line of business (LOB) applications.
2. **P-series**, which supports the full functionality of Power BI via the Power BI Service portal, the Power BI mobile apps, and/or through embedding in organizational portals or LOB applications.



Embedding involves making Power BI reports, dashboards, and individual visualizations visible to end users from within a portal or a line of business application instead of through the Power BI Service. These applications could be custom developed within your organization or be applications developed by Microsoft or third-party ISVs. For example, it is possible to embed a Power BI report in a SharePoint Online modern web page or within a channel in Microsoft Teams.

Power BI supports the embedding of content in two main ways. **Organizational embedding** is for users who have Power BI licenses (either Free or Pro) and who need to be signed in with their organizational account to view embedded content. The **Power BI Embedded Azure service** allows users without Power BI licenses to view embedded content without needing to sign in with an organizational account; the application that contains the embedded content handles the authentication to Power BI separately.

Power BI Premium is not required for organizational embedding, but the ability to allow users with Power BI Free licenses to access content stored in Premium capacities, and the ability to handle larger numbers of concurrent users with more predictable performance, are particularly useful in organizational embedding scenarios.

The Power BI Premium EM-series SKUs have a subset of the functionality available in the P-series SKUs and are less powerful in terms of the resources they provide. The key difference between the two sets of SKUs is that the EM-series SKUs only allow users with Power BI Free licenses to view content in organizational embedding scenarios, while the P-series SKUs allow users with Power BI Free licenses to view content through the Power BI Service web portal as well as through organizational embedding. Also, only the P-series SKUs provide the right to install Power BI Report Server on-premises.

The differences between the two sets of Power BI Premium SKUs can be summarized as follows:

	Power BI Premium EM-Series	Power BI Premium P-Series
Share content broadly to users with Power BI Free licenses in the Power BI Service portal		✓
Share embedded content to users with Free licenses via Microsoft Teams, SharePoint Online and custom applications	✓	✓
Includes Power BI Report Server		✓

Within each tier, the SKUs are differentiated by the number of virtual cores (known as “v-cores”), the amount of storage each one provides, other soft limits including the number of DirectQuery/live connection queries per minute, and the number of page renders per hour (above which throttling and slow-downs occur), and their cost. The v-cores themselves are divided into two types:

- Back-end cores which handle query processing, data refresh, natural language processing and other data-related activities.
- Front-end cores which are responsible for the web service and tasks such as access rights management, uploads and downloads.

A given SKU might have a different number of front-end and back-end v-cores. For a complete list of all of the SKUs within each tier see the [official documentation](#). The online [Power BI Premium Calculator](#) can aid in estimating the Premium SKU needed to support your usage patterns and the associated costs. We also recommend that you discuss your requirements with a Microsoft representative.

Key considerations for choosing a capacity level 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 datasets (vs. DirectQuery or live connections, both of which delegate much of the query workload to the source system)
- Use of streaming datasets and the volumes of data they are expected to handle
- Number of datasets which are actively queried (dynamic memory management offloads idle datasets from memory)
- Frequency of potentially resource intensive operations (such as the use of the Quick Insights feature)

Since a larger Premium capacity node can be divided into multiple smaller capacities after purchase, it is worth considering why you might consider buying one larger capacity node rather than several smaller capacity nodes:

- Larger capacities allow for larger individual datasets to be published. For example, the maximum size of a dataset in a P1 capacity is 3GB; for a P2 capacity it's 6GB. More details on limits are given in [Section 12](#).
- Multiple smaller capacities allow for isolation of resources. For example if there is one group of users who run a lot of reports, their App Workspace(s) can reside on one capacity where it won't affect the performance of other users whose App Workspaces are on different capacities.
- In the future, it will be possible to create Premium capacities in the same tenant but in separate Azure regions. This multi-geo capability will be useful for multinational organizations this will allow content to be stored close to where the employees who will consume it are based.

Deciding Whether To Use Power BI Premium

The benefits Power BI Premium offers are diverse, and it may be that you make the decision to buy it because you only need one or two of them. More importantly it can be difficult to understand the details of these benefits and whether they are relevant to your organization's needs. It may be that you will only know whether you need to buy Power BI Premium after you have used Power BI in production for several months.

Following are some questions you will need to bear in mind while planning your Power BI deployment that will help you make your decision on whether to use Power BI Premium:

What is your budget?

Go to the Office 365 Admin Portal and/or use the [Power BI Premium Pricing Calculator](#) to familiarize yourself with the pricing for Power BI Premium. If your project does not have the necessary budget for Premium, you may need to prepare a business case to secure the extra funding.

Do you intend to use Power BI Report Server?

Be aware that the only two ways to use Power BI Report Server are if you buy a Power BI Premium P-series capacity node or if you have SQL Server Enterprise Edition licenses with Software Assurance. You will need to determine which option is the most cost-effective for your organization. Details on licensing for Power BI Report Server are discussed in [Section 4](#).

How many users need to publish content to the Power BI Service and how many only need to view content there?

These numbers will be critical for calculating your licensing costs and, in turn, working out whether purchasing Power BI Premium is more cost-effective than buying Power BI Pro licenses

alone – please review [Section 4](#) to gain an understanding of the different licensing options. At the beginning of a Power BI deployment it will be difficult to estimate these numbers. The more content you publish the more users you will gain, and as Power BI spreads through your organization you may be surprised by the number of users who embrace it and the way it gets used. Continually monitor actual usage (see [Section 10](#) for more details) as your deployment proceeds, and periodically review your licensing so that you know whether you have reached the point where buying Premium is the right choice.

Do you expect your users to view content in the Power BI Service or the Power BI mobile apps, or will it be enough to view Power BI content embedded in another portal or line of business application?

If you have a relatively small, stable number of Power BI reports that can be delivered exclusively as embedded content, for example in SharePoint Online, this means you can use the less expensive Power BI Premium EM-series SKUs rather than the P-series SKUs. However, do not underestimate how much flexibility the Power BI Service adds to the Power BI experience in terms of being able to add new content easily. You should also be aware of how much importance many users place on making reports available on mobile phones and tablets.

How much data do you intend to store within the Power BI Service, and how often do you need to refresh that data?

The increased limits on the size of a dataset, on the overall amount of data that can be stored in Power BI, and on how often that data can be refreshed, are compelling reasons to purchase Power BI Premium. To reiterate: estimating data volumes at the beginning of a deployment can be very difficult, so monitoring data volume growth over time is necessary to determine when Premium becomes necessary. In addition, as discussed in [Section 6](#), using different data storage options such as Analysis Services can reduce the amount of data that needs to be stored in Power BI.

How many concurrent users will need to view content within the Power BI Service?

As with data volumes, ongoing monitoring is necessary to be aware of how many concurrent users view content in Power BI at any one time, and how well Power BI is able to support their requests. Poor performance can lead to user dissatisfaction and discourage users from going to Power BI. Purchasing Power BI Premium may alleviate some of those concerns.

Power BI Embedded

What is Power BI Embedded?

Power BI Embedded is an Azure service primarily aimed at independent software vendors (ISVs) who intend to include BI and reporting functionality in their applications. It is also available for the creation of internal corporate applications and, as such, can be considered as an architectural option for an enterprise deployment of Power BI (even though it is less frequently used in this way).

The Power BI Embedded service works in a very similar way to Power BI Premium in that you purchase capacities (albeit in the Azure Portal and not from the Office 365 Portal), and move App Workspaces into a capacity. Once this has happened, content can be consumed by embedding it in a custom-built application. The cost is based on purchase of scalable capacity, rather than licensing for individual users. More details are provided in [Section 4](#).

It is important to understand that when accessing content via Power BI Embedded, users do not use their organizational credentials. Instead the ISV enables the application user to use the same identity as the application uses to access content in Power BI Embedded programmatically, and the application users themselves do not need Power BI licenses to access the content.

Deciding Whether to Use Power BI Embedded

There are two scenarios where using Power BI Embedded could be an appropriate choice for an enterprise (as opposed to the traditional ISV scenario in which embedding occurs within their software product):

Internal Line of Business Applications

Many organizations develop custom applications for their employees to use: for example, a bank might have its own internal application for managing customers and accounts. Often these applications, which are used by a large number of employees, need some kind of reporting functionality to support their main purpose. Power BI Embedded can be used to embed reports and dashboards into such applications.

Custom-Built BI Portals

Some organizations might prefer to create their own custom-built portal for displaying reports and dashboards, perhaps to integrate with an existing intranet, rather than use the Power BI Service portal. However, in most cases, it would be hard to justify the significant cost involved as well as the time needed to build it, and it would likely only be considered by organizations with strong existing skills in web development and some very specific requirements.

Power BI Embedded vs Organizational Embedding

The alternative to using Power BI Embedded in both of these scenarios is to use organizational embedding. This is possible if all users have Power BI Pro licenses, or if content is published to a Power BI Premium capacity. As stated above, Power BI Premium EM-series SKUs are not required for organizational embedding but provide a combination of licensing terms and functionality that is specifically targeted at organizational embedding scenarios.

The decision whether to use organizational embedding or Power BI Embedded is based on the following factors:

- **Access using organizational identity.** If a user's identity is used to grant access to content using Azure Active Directory user groups, or to enforce row-level-security in centralized data sources, you must use organizational embedding.
- **Re-use or customization of content by end-users.** Within organizations, content is often shown embedded within an a line of business (LOB) application workflow, but is also reused in dashboards outside of the LOB application, or is customized by end users within departments. Using organizational embedding makes these options easier to achieve, with less coding and maintenance.
- **Strict use of content in a single application.** When analytics is required to augment a LOB application workflow, but that content would not be applicable to be shared outside the LOB application, then Power BI Embedded is an ideal solution.

Additional details on the distinction between external embedding (i.e., Power BI Embedded A-series SKUs) and organizational embedding (i.e., Power BI Premium P-series and EM-series SKUs) can be found in the [Section 12 comparison table](#).

Supported Features in Power BI Embedded

As noted in the scope section at the beginning of this whitepaper, an extremely thorough discussion of Power BI Embedded is out of scope for this whitepaper. Be aware that some functionality is not supported in the Power BI Embedded A-series SKUs, such as:


- Publish to Web
- SharePoint Online integration
- Teams integration
- Reporting Services reports
- Export to Excel doesn't exist like it does in the Service, nor does export to PowerPoint
- Viewing content via the Power BI mobile applications (content can only be viewed via a browser)

Section 4. Power BI Licensing and User Management

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 options to simplify choices and meet the varying needs of individuals, small groups, and large organizations.

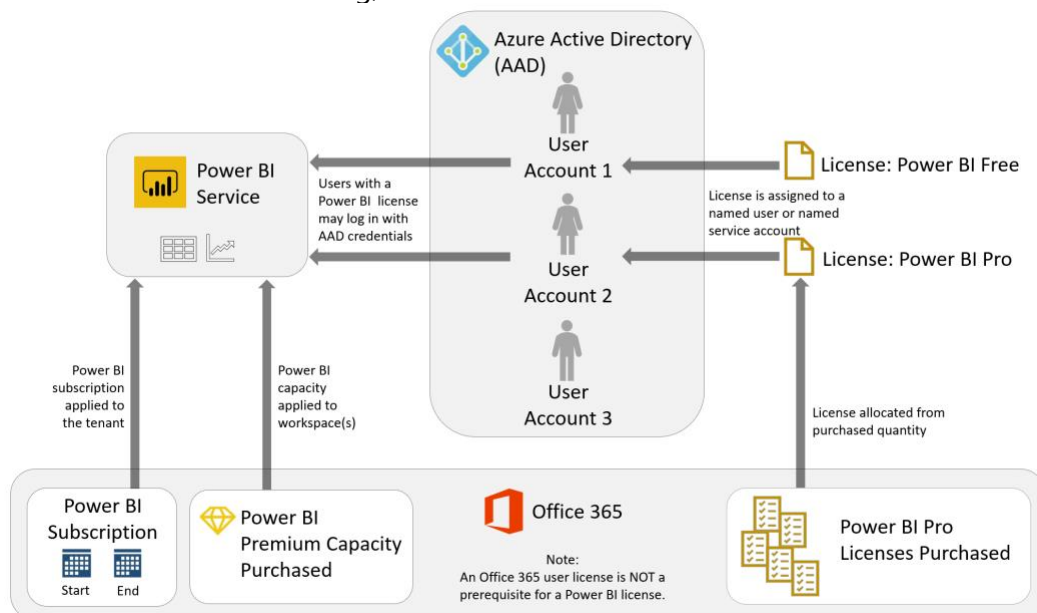
With respect to terminology: a single organization will typically have a single Power BI [tenant](#) (though some large organizations may be decentralized). To use the Power BI Service, one or more subscriptions must be obtained for that tenant. A Power BI subscription may be free or it may require a paid subscription. 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. No license is required to use Power BI Desktop, though users will be prompted to sign in.

 Although the Office 365 admin portal can be (optionally) used to create an AAD account, and it is where licenses are purchased, an Office 365 user license is NOT a prerequisite in order to use the Power BI Service. To utilize the Power BI Service, the user requires an:

- AAD account, and
- Appropriate Power BI license (Free or Pro)

All users of the Power BI Service require a Power BI license to be assigned to them (either Free or Pro).

As indicated in the following, User AAD Account 3 does not have access to the Power BI Service:



If a user does not have an Office 365 license, then a user will be automatically created which can then be assigned a Power BI license. However, if the organization utilizes ADFS (Active Directory Federation Services), then a user will not be auto-generated and the user may receive a message to contact their administrator if they are trying to utilize a Pro feature or sign up for a Pro trial.

Power BI Licensing Summary

The following is a summary of the Power BI licensing options:

Power BI Free	Power BI Free is a user-based license which includes the ability to use the Power BI Service for personal use. There are no collaboration or sharing options with the free tier. It is very important to be aware that Power BI Free licenses are not at all suited to an enterprise deployment, except in conjunction with a Power BI Premium license (discussed below).
Power BI Pro	Power BI Pro is a user-based license which includes all Power BI Free features plus the ability to share and collaborate with colleagues in the Power BI Service.
Power BI Premium	Power BI Premium is a capacity-based license which can be purchased via either the P-series SKUs or the EM-series SKUs. See Section 3 for a discussion of Power BI Premium capabilities and use cases. From a licensing perspective, Power BI Premium opens up the capability to distribute content to report consumers with Power BI Free licensing.
Power BI Embedded	Power BI Embedded is an Azure service which provides the ability to embed Power BI content in custom business applications. Power BI Embedded is purchased via A-series SKUs in the Azure portal (as opposed to the EM-series and P-series SKUs from O365 mentioned above).

The above licensing options can be mixed and matched within an organization, based on the workload for individual users and groups.

The following matrix is a summary of high level capabilities by license type:

...you are purchasing these Power BI capabilities:

<i>If you purchase this...</i>	Power BI Service	Custom application: Organizational Embedding	Custom application: External Embedding	Power BI Report Server
Power BI Pro	Ability for a named user to use the Power BI Service, including collaboration, sharing, and distribution capabilities	Ability to distribute content in an embedded solution <i>(all users require Pro licenses)</i>		Ability to publish PBIX content to Power BI Report Server
Power BI Premium	Capacity for content in a workspace, including isolated workloads, higher size limits, incremental refresh, and the ability to distribute content to read-only users with Power BI Free licenses	Capacity to support distribution of content in an embedded solution to users with Power BI Free licenses		Access to use of Power BI Report Server, within a corporate data center (same # of cores available on-premises as purchased with Premium)
Power BI Embedded (Azure Service)			Capacity for making API calls to distribute reports in an embedded solution	
SQL Server Enterprise Edition with Software Assurance				Access to use of Power BI Report Server within a corporate data center
Power BI Free	Ability for a named user to use the Power BI Service for personal use only <i>(limited to My Workspace with no enterprise capabilities, unless in conjunction with Power BI Premium)</i>			

The remainder of this section will clarify how to purchase licenses, and define each license type in a little more detail.

How to Purchase Power BI Licenses

A Power BI license can be assigned to:

- An Azure Active Directory (AAD) account, or
- A Microsoft account (MSA) that has been [invited to be a guest user through AAD](#), or
- An identity which has been federated with AAD

Power BI licenses are purchased in the Office 365 admin center (with the exception of the Power BI Embedded A SKUs, [discussed later in this section](#)). The purchase must be made by someone who is a member of the 'Global administrator' or the 'Billing administrator' role in Office 365.

Purchasing Power BI User Licenses

Within the [Office 365 admin center](#), you can purchase Power BI user licenses one of these ways:

1. Select **Power BI Pro**. This is a standalone license which can be purchased for one user, or in bulk (for example, purchasing a quantity of 10 Pro licenses).
2. Select **Office 365 Enterprise**. E5/G5/A5 represents the full suite of Office 2016 desktop and Office 365 services. Power BI is included as a component of the suite.
3. Select **Power BI Pro Developer (MSDN)**. This benefit is available to users with a [Visual Studio Enterprise subscription, or MSDN Platforms subscription](#). This option will have no additional cost since Power BI Pro is included as part of the Visual Studio or MSDN Platforms subscription.

Purchased		
Power BI Pro	Office 365 Enterprise E5	Power BI Pro Developer (MSDN)
\$9.99 user/month	\$35.00 user/month	Active
A cloud-based business analytics service that enables anyone to visualize and analyze data with greater speed, efficiency, and understanding. Power ...	The Office suite, plus email, instant messaging, HD video conferencing, 1 TB personal file storage and sharing, and advanced security, analytics and ...	\$0.00 user/month Next billing: May 18, 2018 1 user purchased
Office 2016 desktop & mobile apps Not included	Office 2016 desktop & mobile apps	Office 2016 desktop & mobile apps Not included
Office 365 services	Office 365 services	Office 365 services
...

Purchasing Power BI Premium

[Power BI Premium](#) (P SKUs or EM SKUs) capacity nodes are purchased via the Office 365 admin center. A purchase involves the selection of a capacity level such as P1, P2, EM1, or EM2. The purchase of multiple capacity nodes would be for scale-out scenarios, whereas the purchase of a higher level node would handle a scale-up scenario. This is discussed in more detail in the [Power BI Architectural Choices section](#) previously covered in this whitepaper, as well as the [Managing Power BI Premium](#) section later in this whitepaper.


For subsequent purchases of additional capacity nodes, you can start from within the Power BI Admin portal which will direct you to the Office 365 admin center to complete the purchase. If you do not see the SKU you are looking for, check with your Microsoft account representative because the SKU may only be available via volume licensing.

The Premium capacity purchased can be upgraded or downgraded within the P or EM family of products. However, a P SKU cannot be downgraded to an EM SKU. The purchase of Power BI Premium may involve a yearly commitment (paid for monthly over the course of the commitment year).

Purchasing Power BI Embedded

[Power BI Embedded](#) (A SKUs) are purchased within the [Azure portal](#) as opposed to Office 365. A purchase involves selection of a capacity SKU such as A1 or A2.

When selecting a location, try to select the same data center as other Azure services if possible in order to improve performance and reduce data egress charges.

 The Power BI Workspace Collection service in the Azure portal is deprecated. Be sure to utilize the Power BI Embedded service instead.

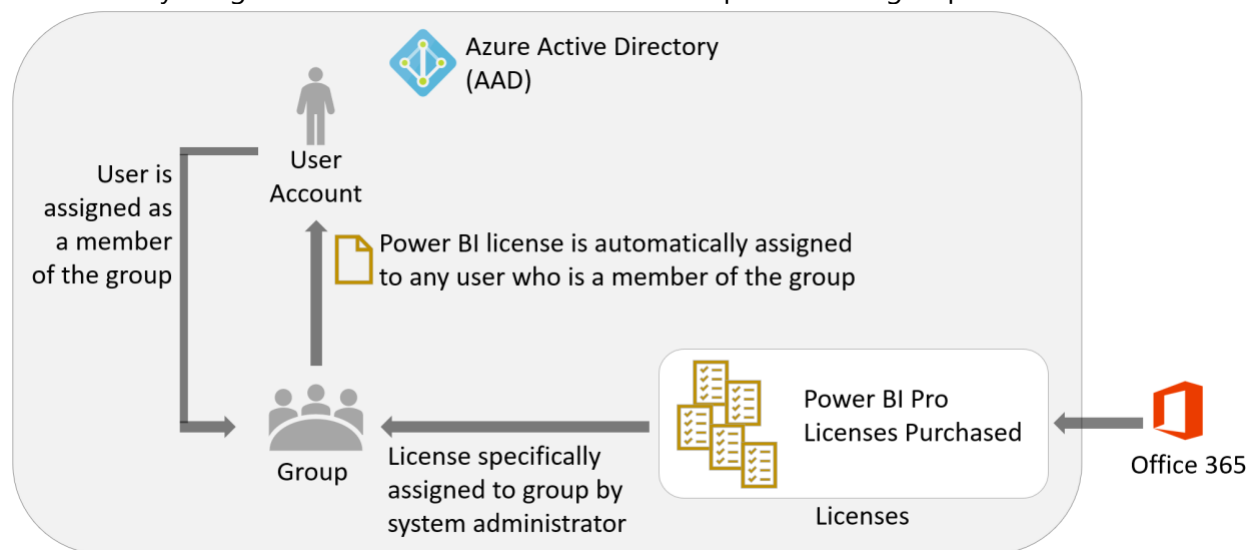
One very big difference between Power BI Premium and Power BI Embedded is that the Power BI Embedded service can be paused when it is not in use. While the service is paused, no user queries can be served and there are no billing charges. Pausing capacity is particularly helpful for dev/test scenarios, when the capacity does not need to be running 24x7. Billing for Power BI Embedded is hourly, based on minutes the capacity is running, and there is no commitment.

Ways to Assign a Power BI User License

Licenses can be assigned to users in the [Office 365 admin center](#), within the [Azure portal in Azure Active Directory](#), or via PowerShell.

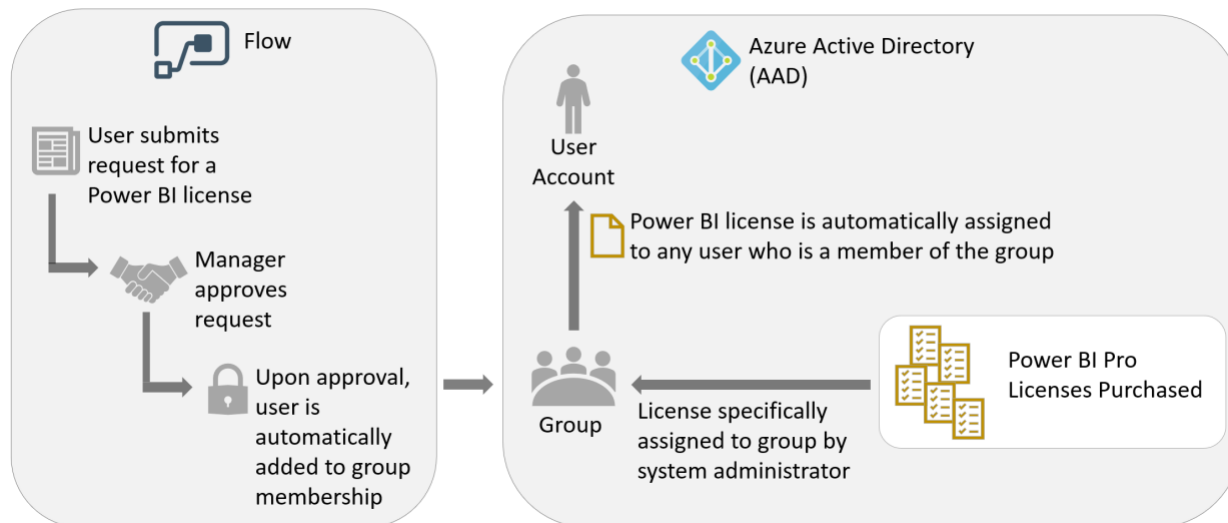
Group-Based Licensing Assignment

In large organizations, handling requests for user licenses can involve a lot of administrative effort. One option to improve efficiency is to utilize [group-based licensing](#) in which a license is automatically assigned to the user based on membership in an AAD group:



Workflow-Based Licensing Assignment

Group-based licensing can be particularly effective for assignment of Power BI Free licenses. For assigning Power BI Pro licenses which incur a subscription cost, additional workflow or approvals may be required. One way this could be accomplished is via the use of Flow:



Managing Power BI Trials and License Assignments

Above, we discussed scenarios for an administrator to assign a Pro license for a user. It is also very common for a user to get started with Power BI Pro with a 60-day free trial. The free trial offer is available via the web, or via the Power BI Service. If an existing Pro user shares content with a Free user, the Free user is prompted to start a Pro trial in order to view the content.

Automating the Conversion of Trial Accounts to Pro User Licenses

For a user to continue using Power BI Pro capabilities after their 60-day trial has concluded, they will need to have a valid user license assigned. It is useful to avoid interruptions of service for the user. To avoid this being a time-intensive manual process on the part of the administrator, there is a PowerShell technique discussed [here](#) to:

- Search the Office 365 audit logs for operations with a description of "OptInForProTrial"
- Assign Pro license to users who signed up for a trial per the audit log

Handling Auto-Created Office 365 Tenants from Self-Service Signups

If there is not an Office 365 tenant in existence associated with an e-mail domain (ex: Contoso.com), then when the first user from Contoso.com signs up for a Power BI account (Free or Pro), an unmanaged directory will be auto-created. This is sometimes referred to as a shadow tenant. If an official Office 365 tenant is later desired, the administrator can take over the existing tenant and manage it normally.

- [Info for Office 365 takeover](#)
- [Info for Azure Active Directory takeover](#)

Blocking Users from Signing Up for Power BI

Most organizations do permit Power BI Free licenses to be assigned automatically. However, certain organizations deem it necessary to prohibit the user of the Power BI Service due to data privacy or other concerns, or they need to control the assignment of Power BI Free user licenses very closely. In that situation, you can utilize PowerShell to [disable automatic license distribution \(aka ad hoc subscriptions\)](#). If ad hoc subscriptions are disabled, the user will see a message that the IT department has turned off access to Power BI. This affects the Free subscription type only. Note that this process is only applicable to the commercial cloud, not the national clouds such as US Government.



There is currently not a capability available to block users from enrolling in a Power BI Pro trial.

Finding Power BI Licenses Assigned to Users

For auditing which users have been assigned a Free or Pro license, that can be done in one of two ways:

- [View Assigned Users in Office 365 Admin Center by Users](#)
- [Utilize PowerShell to Obtain License and Service Details](#)

Types of Power BI User Licenses

Power BI Free

Power BI Free makes the Power BI Service available for personal use. Note that it is not suitable for an enterprise-level deployment, except when used in conjunction with Power BI Premium.

Power BI Free can be used one of two ways:

- 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: for example no report or dashboard sharing is allowed and a user cannot access App Workspaces if they have a Free license. A user in the free tier will only be able to utilize 'My Workspace' in the Power BI Service.
and/or
- In conjunction with Power BI Premium.** When content is deployed to an App Workspace which has been assigned to Premium capacity, an unlimited number of users with Power BI Free licenses may view the content (read-only) as consumers in several different ways, for example if the content has been published as an App or if it is embedded in a custom application.

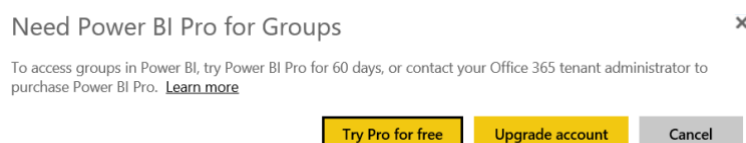
As the name suggests, Power BI Free costs nothing to use but nonetheless requires a subscription.

Power BI Pro

The Power BI Pro license introduces collaboration, distribution, and sharing features within the Power BI Service. A Power BI Pro license is required for any form of sharing or content distribution. 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 below).



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 license, nor is an error displayed if the recipient does not have a Pro license. When the Free user attempts to view the content, the Free user will be prompted to upgrade to a Pro trial at that time:



A detailed comparison of the functionality available to Power BI Free and Power BI Pro users can be found in [Section 12](#).

Other Power BI License Types

Power BI Premium

As discussed in [Section 3](#), Power BI Premium is an offering designed to address the workload demands of larger enterprise deployments.

The Pro licensing model, discussed in the previous section, 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): buying Power BI Pro licenses for all users can become very expensive. If this is the case, purchasing Power BI Premium may be a better option because it permits users with a Power BI Free license to consume content published to App workspaces that are hosted in Premium capacity.

The point where Premium becomes most cost-effective from a licensing perspective depends upon the number of users who only need to view content, whether that content needs to be accessed via embedding or via the Power BI Service portal, and the pricing arrangements your organization has with Microsoft. For more information on how to make this decision, see the discussion of [Power BI Premium tiers and SKUs](#) in Section 3, including a list of considerations for selecting a Premium capacity level.

Multi-geo support is a future feature coming to Power BI Premium, wherein different Premium capacity nodes may be provisioned in the data region of your choosing independently of the region your Power BI tenant is located. From a licensing perspective, a multi-geo implementation will impact cost since it will require more than one Premium capacity, each located in different geographic regions.

Power BI Report Server

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. [Section 3](#) includes additional detail about Power BI Report Server.

Customers can acquire a license to use Power BI Report Server in one of two ways:

1. **Purchase of a Power BI Premium subscription.** With the purchase of Power BI Premium P-series capacity node, you are permitted to also install Power BI Report Server on an on-premises server with the same number of cores as there are v-cores included in the capacity node. 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 who have purchased SQL Server Reporting Services (SSRS) may elect to install Power BI Report Server, or migrate an existing SSRS 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 Power BI Report Server must reside in an on-premises data center; it is not authorized to run in an Azure virtual machine except with a developer license (i.e., non-production development or learning environments).

Power BI Embedded

As discussed in [Section 3](#), Power BI Embedded is an Azure service which provides the ability to embed Power BI content in custom business applications.

Power BI Embedded is licensed via A-series SKUs purchased in the Azure portal (as opposed to the EM and P-series SKUs from Office 365 mentioned above which are associated with Power BI Premium). Billing is handled on an hourly basis, with no required pre-commitments. Unlike Power BI Premium, capacity purchased via Power BI Embedded can be paused when not in use,

thereby not incurring any cost whatsoever (and, accordingly, reporting capabilities are not available to users when the capacity is paused). Also, capacity can be scaled up and down as needed based on demand in order to optimize cost.



Cost savings tip: Because there is no billing commitment, and the service can be paused, the capacity purchased via A-series SKUs for Power BI Embedded can be used for testing different capacity levels to determine which level of Premium SKU to purchase. For instance, a Power BI Embedded A4 SKU is equivalent to a Power BI Premium P1 SKU.

Section 5. Power BI 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 is 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 their issues.

Once the user base has begun using Power BI (in particular, the Power BI Service), the Power BI administrator will have 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 sanctioned and/or 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. In turn, this increases the value of the reports created by subject matter experts. Consider using [Azure Data Catalog](#), a cloud-based service for cataloging 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 as sources when practical	<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, or 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 display 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 datasets 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 source files in a central location, accessible by all co-authors of the Power BI solution	<p>If Excel workbooks, csv files, or other text files are used as data sources, it is strongly recommended they be stored 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. Locations such as Azure Data Lake Store or Azure Blob Storage are equally acceptable. Specifically, avoid referring to a path such as c:\Users\User1\FileName.txt 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>
Be aware of API connectivity issues and limits	<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 which many business users and experienced IT professionals may struggle with. Developing a Power BI custom data connector, and then distributing to your end users, can remove the need to write M code when connecting to an API and make importing data much easier.</p> <p>There may also be limits on how often the APIs can be called and how much data can be retrieved from them. Downloading large amounts of data from APIs may potentially be slow due to throttling imposed by the service provider. The API owner typically documents limits on their website.</p>
Know how to support SaaS solutions from AppSource and expect further data integration requests	<p>Datasets, reports, and dashboards from several popular SaaS solutions, such as Salesforce and Google Analytics, can be accessed using AppSource. These pre-built solutions 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 discussed in the previous point. Also, the datasets which are created as part of the SaaS solutions may not be easily modified; 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 query load on source systems	<p>Querying data from source systems may, in some cases, cause problems for the data source itself. For example, if users frequently connect to line of business (LOB) databases and retrieve large amounts of data, or run complex queries (particularly with DirectQuery), this can result in the LOB applications becoming slow or unresponsive because they are experiencing contention or locking conflicts.</p> <p>Because it can be challenging for systems to satisfy both transactional and analytical reporting needs, Power BI users may be prevented from accessing certain transactional/OLTP 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 Power BI Service 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 an internet connection may take some time. This is one of several reasons to minimize the number of Power BI files which import data (that is, re-use an existing Power BI dataset when possible rather than creating a new one entirely).</p> <p>The On-Premises Data Gateway, discussed in detail in Section 7, will need to be installed within the corporate firewall to allow Power BI to connect to on-premises data sources (enterprise mode is strongly recommended over personal mode 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 using M functions to make certain types of call 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, are one of the most suitable 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>

Make accessing data as easy as possible	Power BI can access a wide range of data sources, but not all data sources are as easy to connect to and work with as others. For example, connecting to web services requires M programming skills that many users do not possess, and the data returned by many web services often requires further transformation before it can be used. The use of – and the creation of, if necessary – Power BI custom data connectors can make it much easier for your users to connect to a wider range of data sources; they usually return data in a Power BI-friendly format too, meaning that less transformation work needs to be done by the report designer.
Assume referential integrity when possible to improve performance	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.

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 out unnecessary columns or rows
- Handling dirty data, incorrect data, or errors
- Integrating data from multiple, inconsistent data sources
- Denormalizing a dataset (i.e., “flattening the data” to reduce the number of tables) to make the dataset more suitable for reporting



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. Therefore, 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 or reporting database 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.

Dataflows

In the future, a major new feature in Power BI called dataflows (initially known as the Common Data Model for Analytics, or datapools, during the private preview period) will make it much easier to create a simple, centralized repository like the one described above, albeit one that is not as sophisticated as a full-fledged data warehouse. Dataflows will be available for all users with a Power BI Pro license, with additional features available for Power BI Premium customers.

As depicted in the [Section 2 scenario](#), this feature will add a new object to a Power BI App Workspace: a dataflow. Dataflows are created using a web-based version of the same Power Query engine that is used in Power BI Desktop. A dataflow provides a way of loading data into Azure Data Lake Storage that can be used by multiple datasets both inside and outside the App Workspace. This area is comprised of Common Data Model compliant folders stored in [Azure Data Lake Storage Gen2](#) (a compliant folder indicates the presence of a file containing metadata and schema associated with the data file). Data stored here will count towards the data storage limits for Power BI Pro users; moving an App Workspace to a Premium capacity will allow for storing much larger data volumes.

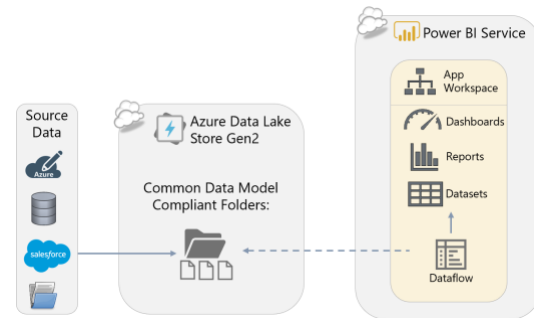


Power BI Premium will permit use of critical features such as larger datasets and incremental refresh. Since the dataflows concept becomes most useful at scale, Premium capacity will be required to use dataflows most effectively.

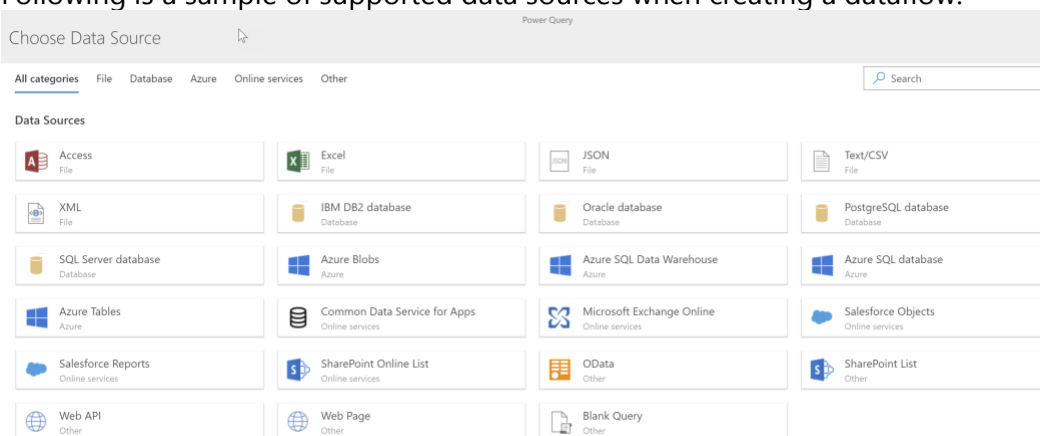
Optionally, it will be possible to attach existing Azure Data Lake Storage, so Power BI content developers can access the same data that may also be used by other Azure services such as

Azure Machine Learning, Azure Databricks, and Azure Data Factory. Because the data resides in Azure Data Lake Storage, it can be accessible not only to Power BI but to a wide array of tools.

Data is loaded from the source, via a dataflow, into an entity in the Common Data Model. A number of pre-defined entities for standard objects, such as customers and products, are available out of the box. Power BI developers may create custom entities.



Following is a sample of supported data sources when creating a dataflow:



There are three main scenarios that will be enabled by dataflows:

1. To create a **centralized data repository via self-service data preparation**. By utilizing a central data repository, content developers can share clean, integrated data, such as an organization's master list of products or customers, with multiple Power BI datasets. This also allows Power BI content developers to perform complex and slow operations that extract data from source systems only once, instead of having each dataset repeat these operations unnecessarily.
2. A more flexible, powerful replacement for the Apps currently found in AppSource mentioned earlier, **Power BI Apps** allow Microsoft and third party ISVs to create packaged applications that can connect to software-as-a-service data sources, map the data in common data model into entities, manage the data load, and/or enrich that data using techniques such as machine learning.
3. **Integration with the many cloud-based big data and machine learning services** available in Azure, so that data and insights from these services can be used more easily in Power BI. Utilizing the data lake as a central location, data preparation and enrichment work can be shared between Power BI content developers and others such as data scientists and the IT department.

Section 6. 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
- 2 Live connection to Analysis Services
- 3 DirectQuery mode
- 4 Composite models (future feature at the time of this writing)
- 5 Real-time streaming datasets (aka push datasets)

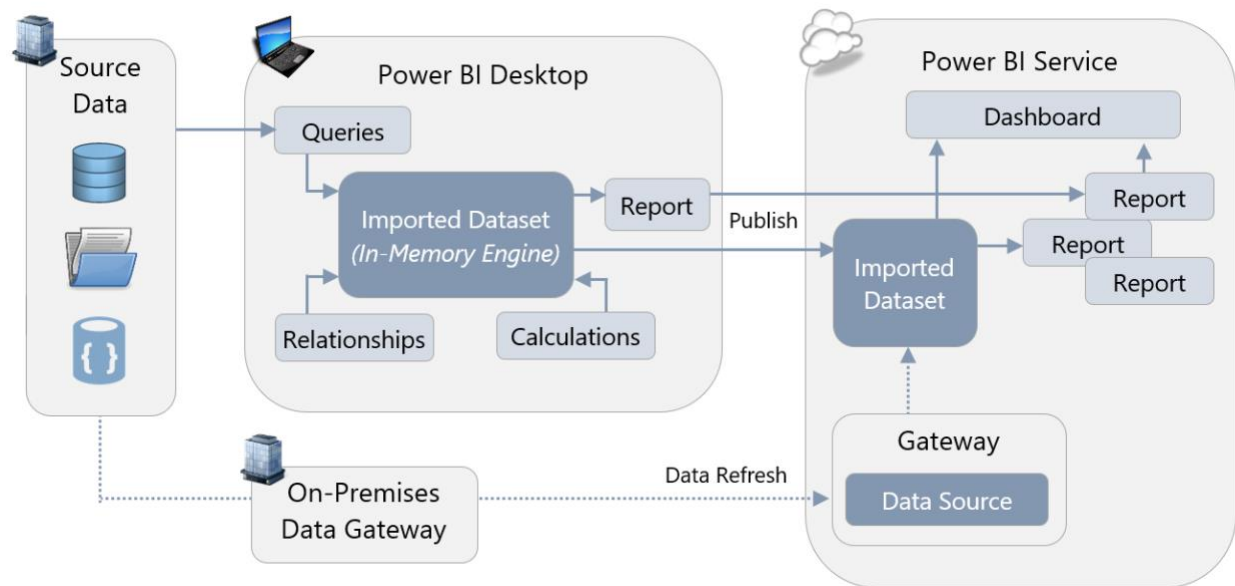
Each of the five storage modes has different advantages and disadvantages, and you may choose to use different storage modes for different reports. The following is a short summary of the technical details of each storage mode:

Importing Data

The most common way of working with data in Power BI is to 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 in the dataset.

In this scenario, Power BI uses its 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.

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



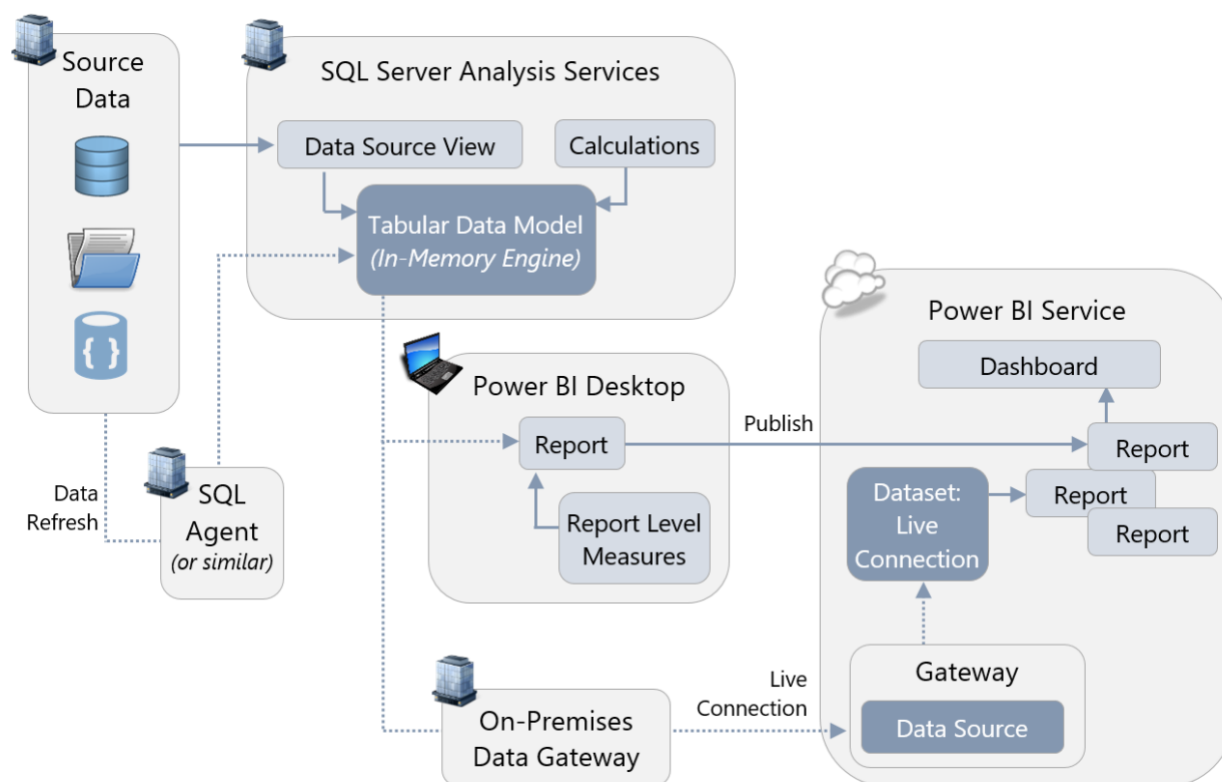
Refer to [Section 7](#) for more details on when the On-Premises Data Gateway is required for data refresh, plus when and where credentials are stored.

Live Connection to Analysis Services

The database engine that Power BI uses to store imported datasets is, essentially, the same engine that runs inside 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. In this situation, DAX queries are issued via a live connection back to Analysis Services whenever a report is rendered. Any version and edition of SQL Server Analysis Services Tabular, and any tier of Azure Analysis Services, can be used in live connection mode.

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 and 2017 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 unless Azure Analysis Services has been added to a virtual network which limits connectivity to within certain ranges of the corporate network.

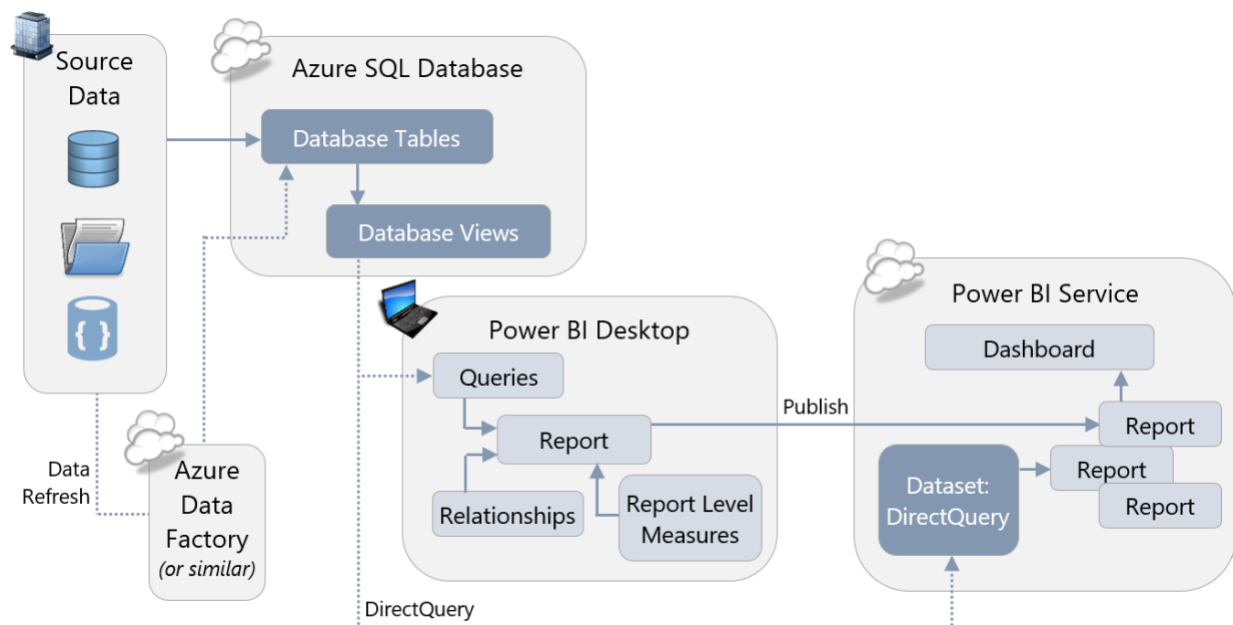
The above image indicates usage of [report level measures](#). This feature is available for SSAS Tabular live connections, or Power BI Service live connections.

Refer to [Section 7](#) for more details on when the On-Premises Data Gateway is required to support a live connection, plus when and where credentials are stored.

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 sources, such as relational databases, can be used in DirectQuery mode. See [here](#) for a full list of data sources that can be used in DirectQuery mode. When a data source is listed as Beta or Preview, that source is not yet supported for production use and may not work in the Power BI Service. Certain custom data connectors also support DirectQuery mode too.

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



Some, but not all, DirectQuery data sources also support [single sign-on \(SSO\)](#) via the on-premises Gateway using Kerberos. This allows Power BI to pass the identity of the user running the report back to the data source so that security defined in the data source can be applied correctly. At the time of this writing, SQL Server, SAP HANA, SAP BW, and Teradata support SSO; support for Oracle is under development as a future feature.



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.

Composite Models

In the future, the composite models feature will allow individual tables in a Power BI dataset to be stored either as imported tables, in DirectQuery mode, or in dual mode where either import or DirectQuery storage is used depending on the context. This will allow Power BI content developers to gain the benefits of DirectQuery mode only for the tables in a dataset that

actually need them, and use import mode for the other tables where it is the most appropriate choice.

Further out, an additional future feature called aggregations will allow for aggregated data from tables in a dataset to be created to improve query performance. This will allow scenarios where, for example, a large table is stored in DirectQuery mode but aggregated data from that table is stored in import mode. If a user's query can be answered from the aggregated table then Power BI will use it automatically and the query will run more quickly; if the query cannot use the aggregated table, the more granular table stored in DirectQuery mode will be used and the query will be slower.

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

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.

Is Analysis Services Already in Use?

If you already have an investment in SQL Server Analysis Services (or Azure 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 translations and calculated members on parent/child hierarchies. A full list can be found [here](#). In addition, users who are members of roles that use cell security will not be able to connect to a cube at all from Power BI.
- ✖ Similarly, when using live connections to Analysis Services Multidimensional certain Power BI features such as Q&A are not available.



As Power BI Premium develops, it will acquire more and more of the features of Azure Analysis Services and support both IT-owned and self-service workloads from a single, open platform. The ultimate goal is for Power BI Premium to contain a superset of functionality from Azure Analysis Services. This should be taken into account when planning your organization's usage of Azure Analysis Services and Power BI Premium in the long term.

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. Premium capacities also allow for larger individual datasets depending on the SKU, with P3 capacities capable of handling datasets of up to 12GB. More details on the size of dataset that each Premium SKU can handle can be found [here](#). Microsoft has announced that, in the future, the limit on the size of a dataset will be determined by the Premium capacity.
- 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.
- Use a DirectQuery connection back to a relational database such as SQL Server or a cloud-based service such as Azure SQL Database for some or all of the tables in your dataset. 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 result in additional costs, and for some small and medium-sized organizations these costs 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 Considerations](#) 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 as [this article](#) explains. 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.

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. Streaming datasets cannot be created in Power BI Desktop - they can only be created programmatically via the API or via the Power BI Service, making development more complex (although once the streaming dataset exists in the Power BI Service, you are able to set up a live connection to the dataset from Power BI Desktop in order to create reports & report measures).

 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 Service, 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 for datasets stored in shared capacity. Imported datasets stored in Premium capacity may be refreshed up to 48 times a day, which equates to every ½ hour. While datasets stored in Premium can be scheduled more reliably, there are still limitations on the number of scheduled refreshes that can take place in parallel. The limits are related to the number of v-cores available on the Premium SKU that you are using (detailed [here](#)), and this can also influence when scheduled refreshes actually take place and how long they take.

[Incremental refresh](#) is also very important for dataset refresh performance on large datasets, because it means that only new data, or data that may have changed needs, to be loaded from

the data source. When configuring incremental refresh the dataset designer can specify a date range to store data for (for example, only keeping data for the last two years in a dataset), a date range where data may have changed and so where data does need to be loaded from the data source, and optionally a column in the data source that indicates precisely which data has changed. At present incremental refresh is only supported for datasets stored in Premium capacity, but in the future it will be available for all datasets.

For now, at least, Analysis Services provides a much greater degree of control over incremental refresh. 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. When the composite models feature is available, only the tables where data changes frequently will need to be stored in DirectQuery mode making it a more compelling option.

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 connections 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.

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. 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 modeling 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 (this includes the use of complex row-level security), performance of a DirectQuery dataset will usually degrade 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 support 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.

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 stored in shared capacity (i.e. not 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 stored in shared 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 because it runs on dedicated hardware not shared by other customers, and therefore has fewer restrictions on the usage of that hardware.

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. You may choose to use Power BI Premium in combination with live connections to Analysis Services in some cases 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 below.

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 whitepaper available [here](#). Also refer to the compliance information and list of certifications for Power BI available in the Microsoft Trust Center [here](#).

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. Administering Power BI Premium, in contrast, requires less time and technical skill.

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 in conjunction with imported datasets.

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.

If you are considering using Power BI Premium or Azure Analysis Services, 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 within this section can be summarized as follows:

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

Consider 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

Consider using composite models (when they are available) if:

- ✓ Performance of your reports in DirectQuery mode is a problem, but you still need the benefits of DirectQuery mode, and
- ✓ You only need the benefits of DirectQuery for a few of the tables in your dataset

Utilize 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, or scale out, on demand

Use Power BI Premium with imported datasets if:

- ✓ You prefer easy management over cost and fine control over configuration, and
- ✓ You need to create a large number of small or temporary datasets that can be managed outside the IT department in a decentralized manner
- ✓ 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 7. Power BI Data Refresh and 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, it is important to 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 many enterprise data sources are on-premises and Power BI is a cloud-based service (with the exception of Power BI Report Server), 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 (or incremental refresh, if applicable, in Power BI Premium) 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 for the dataset

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 connections datasets or Manually with the 'Refresh Dashboard Tiles' menu item

Cached Visual Container Refresh

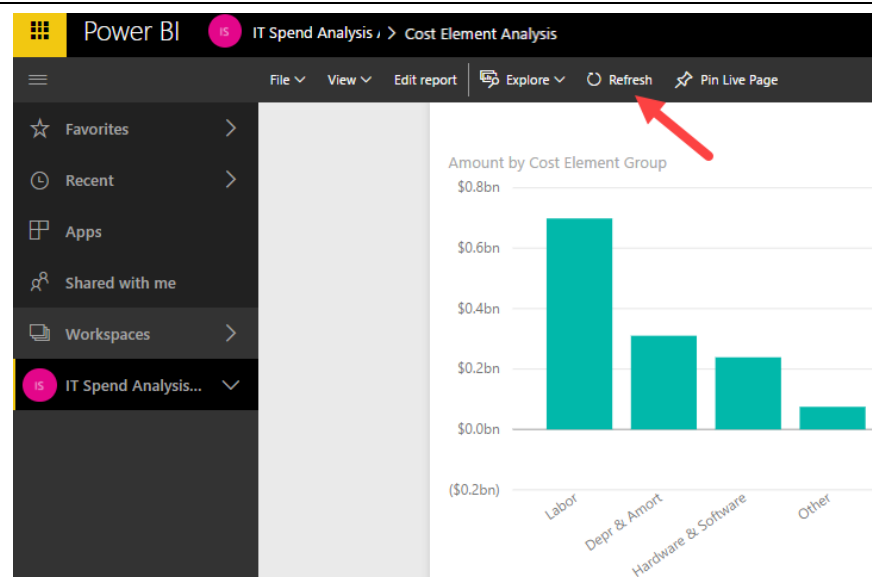
Description	The visual container refresh updates the charts, graphs, and tables which have been cached in reports for performance reasons. This is useful after a data refresh has occurred to bring the visuals up to date.
How Refresh Is Handled	Automatic by the Service or Manually with the 'Refresh' menu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

With respect to cached tiles and visuals, DirectQuery and live connections 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 typically most useful for DirectQuery or live connection datasets. It can also be useful if you manually refreshed an imported dataset a few moments earlier.



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 6](#) 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.

Power BI content published to a Power BI Report Server does not require implementation of a data gateway. Therefore, this section applies exclusively to the Power BI Service.

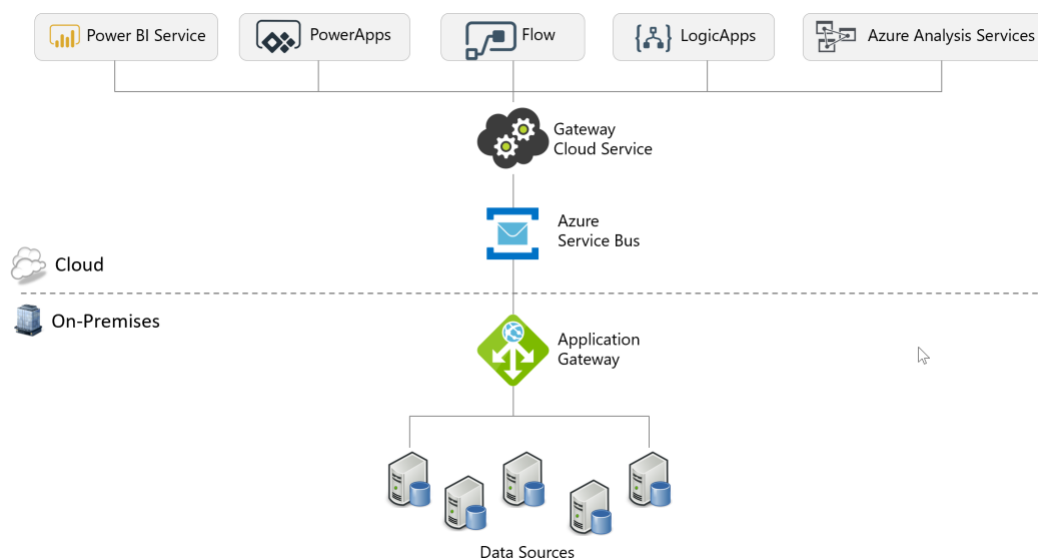


This section makes reference to LogicApps, Flow, and PowerApps. These are separate services which are considered part of the Business Applications Group at Microsoft. Although these services are beginning to introduce some interesting integration points with Power BI, they are not directly related nor 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 now 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**.

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 (Enterprise Mode)	On-Premises Data Gateway (Personal Mode)
Power BI Service	✓	✓
PowerApps	✓	
Flow	✓	
LogicApps	✓	
Azure Analysis Services	✓	

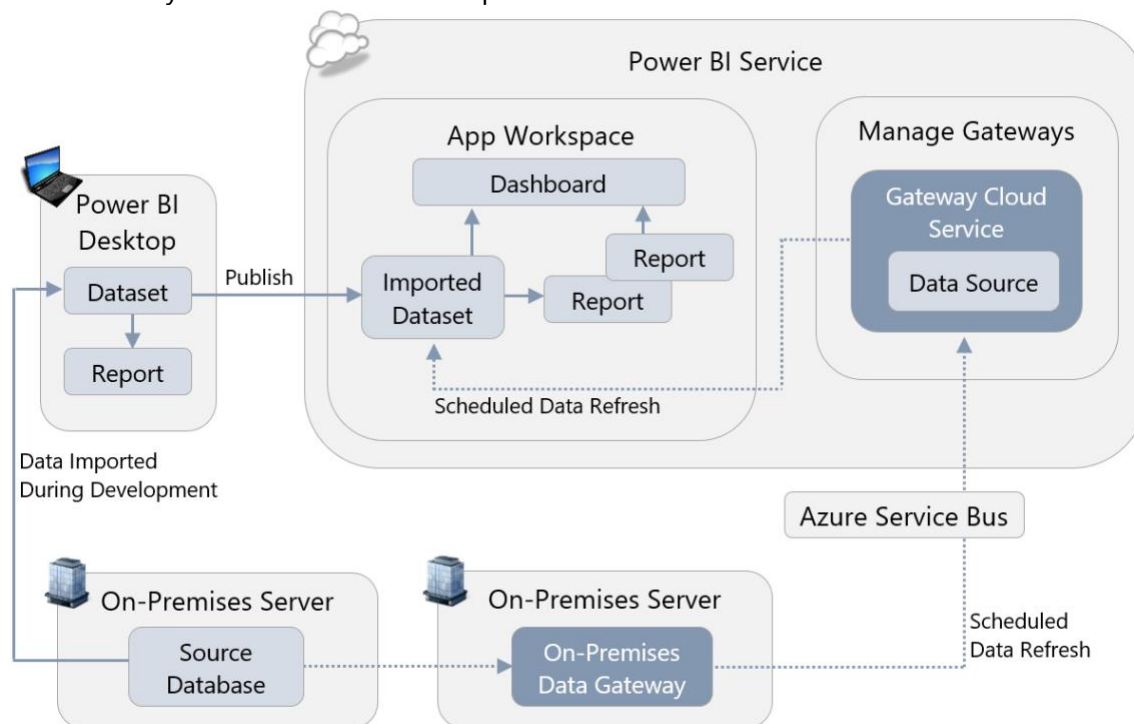
When sharing a gateway with the other above-mentioned services, each service needs to reside in the same geographic data region in order to be able to utilize a single gateway.

Note that both types of gateways in the above table can co-exist on the same machine. However, the other gateways for Azure Data Factory or Azure Machine Learning cannot 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:



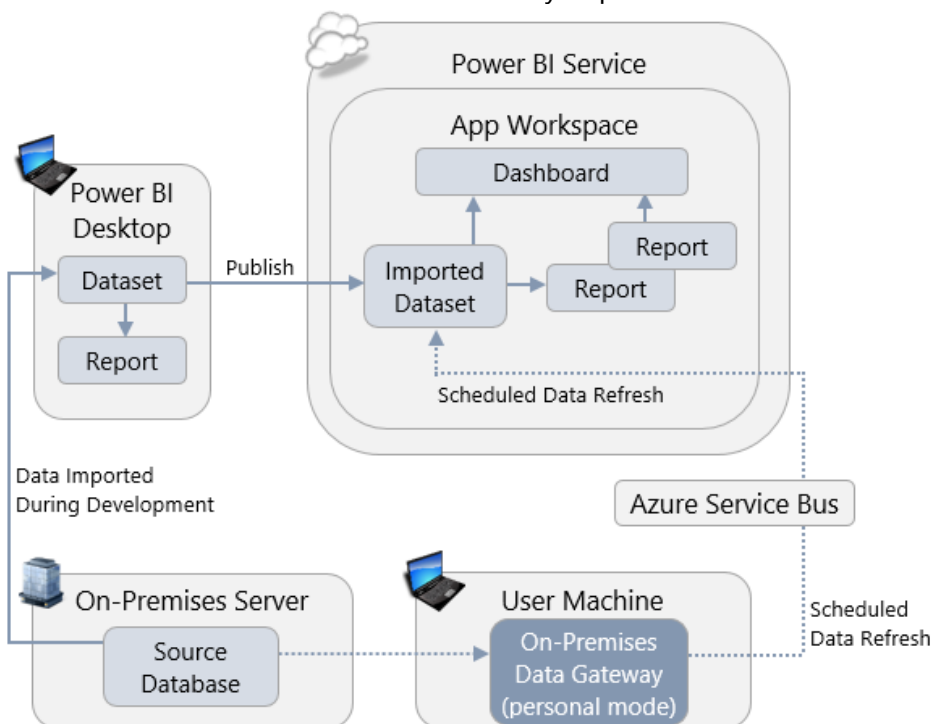
Advantages of the enterprise-oriented On-Premises Data Gateway over personal mode include:

- ✓ 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 Service (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 applications, such as Azure Analysis Services, LogicApps, PowerApps, and Flow (personal mode supports Power BI only).
- ✓ 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 mode 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 On-Premises Data Gateway in personal mode:



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 10](#) for more details on data privacy levels.
- ✓ New features tend to come to the personal mode gateway first.



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 the personal mode gateway 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 datasets	✓	✓
Executes scheduled data refresh for imported data in Excel files	✓	✓
Sends DirectQuery queries		✓
Sends Analysis Services live connection queries		✓
Integrates with other services: Azure Analysis Services, PowerApps, Flow, LogicApps		✓
Supports clustering for high availability		✓
Intended for use with enterprise-wide scenarios		✓
Intended for limited, personal use by one individual	✓	

The 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 enterprise-level On-Premises Data Gateway.

When a Data Gateway is Required

A data 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.

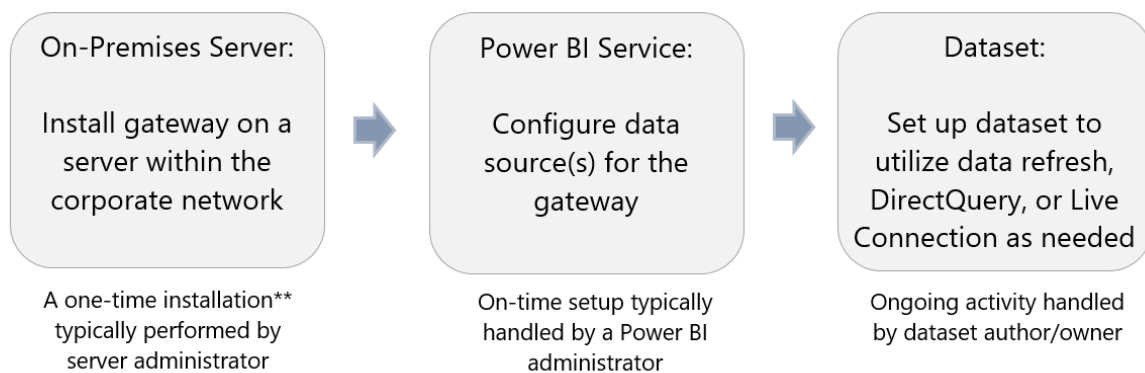
Typically, a data gateway is not necessary to access cloud sources, with the following exceptions:

1. Data sources which reside in an IaaS (Infrastructure-as-a-Service) virtual machine.
2. If the Web.Page() function is utilized in an M query.
3. When a single M query combines data from cloud and on-premises data sources.

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	Tenant-wide visibility to all gateways, data refresh activity, and data sources in use is not currently available (whereas a gateway owner does has visibility into their own gateway).
Load Balancing	Currently load balancing can be accomplished to a certain extent by manually distributing data sources across gateways. Additional features for basic load balancing will be released in the future.

Unsupported Data Sources	Certain data sources, such as R, are not yet supported by the On-Premises Data Gateway.
Prevention of Gateway Creation	There is not currently a mechanism to prevent a gateway from being created, nor to blacklist servers.



Tip for working with a combination of on-premises and cloud data sources: Combining on-premises and cloud data can be accomplished easily when handled via separate M queries. When combining data in a single M query, the On-Premises Data Gateway requires the “Allow user’s cloud data sources to refresh through this gateway cluster” option to be explicitly enabled on the gateway.

Considerations for Using the On-Premises Data Gateway

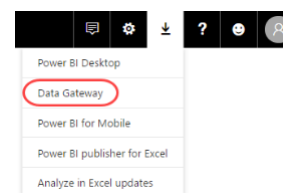
Data Gateway Installation File

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 mode).

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. For instance, if this gateway is dedicated to supporting data refresh operations, versus another gateway being dedicated to DirectQuery connectivity.

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

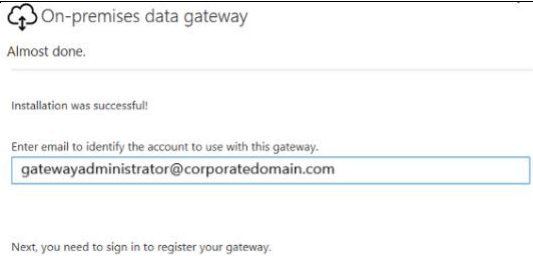

The data source name should equate as closely as possible to the source itself, to avoid any ambiguity. For instance, the name of the database.

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.

Identities and Credentials Used by the Data Gateway

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

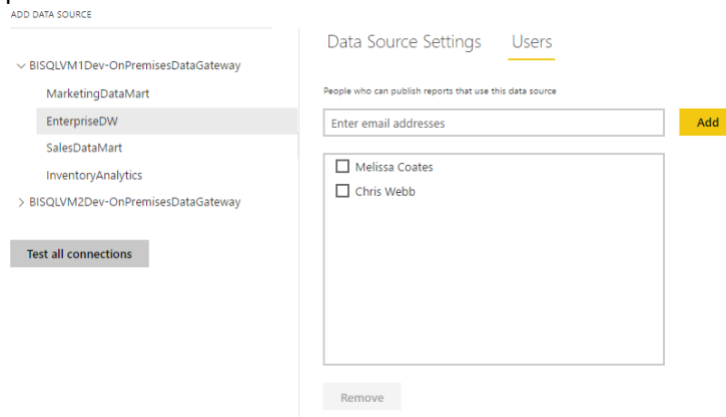
Gateway owner	<p>The gateway owner is specified when the gateway is initially installed on the corporate server. The e-mail account input during installation will be marked as the gateway owner, and will be specified as the contact in the Power BI Service.</p> <p>The extent to which you can utilize multiple Microsoft Business Application Group 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> <p>When changing the service account, be sure to also update permissions for the default location located at: C:\Program Files\On-premises data gateway.</p>	

Data source credentials in the Power BI Service	<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: Analysis Services requires the account to have administrator privileges on the instance. With Analysis Services, 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 data 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 for the data source in the Power BI Service	<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 Data 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 Price Paid

This dataset has been configured by [chris@onmicrosoft.com](#).

Last refresh succeeded: Thu Jun 14 2018 13:07:06 GMT+0100 (British Summer Time)
[Refresh history](#)

Gateway connection

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

Use a data gateway

☒ On

Gateway	Department	Contact information	Status	Actions
<input type="radio"/> Personal Gateway	Personal Gateway		Running on [redacted]	
<input checked="" type="radio"/> ChrisGateway	On-Premises Gateway (usually preferred when available)	chris@[redacted].on...	Running on [redacted]	

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 or a web service), the user can expect to see the following instead since a gateway is not required:

Settings for PBMeetupAPI

This dataset has been configured by [chris@onmicrosoft.com](#).

Last refresh succeeded: Thu Jul 13 2017 22:55:35 GMT+0100 (British Summer Time)
[Refresh history](#)

Gateway connection

You don't need a gateway for this dataset, because all of its data sources are in the cloud, but you can use a gateway for enhanced control over how you connect. [Learn more](#)

Use a data gateway

☐ Off

Apply Discard

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. This is extremely important because it may impact permissions on data retrieved from source systems.

Sharing of Data Gateway with Other Applications

As mentioned earlier, the On-Premises Data Gateway can be reused among Power BI and Azure Analysis Services, as well as 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 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.

It is also possible to create [gateway clusters](#), where multiple gateways installed on different servers appear as a single gateway to Power BI. This supports the need for high availability: all requests for data that result from the refresh of imported data sources, or come from DirectQuery or live connections, are routed to the primary gateway in the cluster if it is online; if it is not online these requests will be routed to another instance in the cluster.



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 the same 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 serves data refresh operations for import models vs. DirectQuery models, if the activity for certain types of activities becomes significant.

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 On-Premises Data Gateway running in personal mode only), but this option should be used with extreme caution.

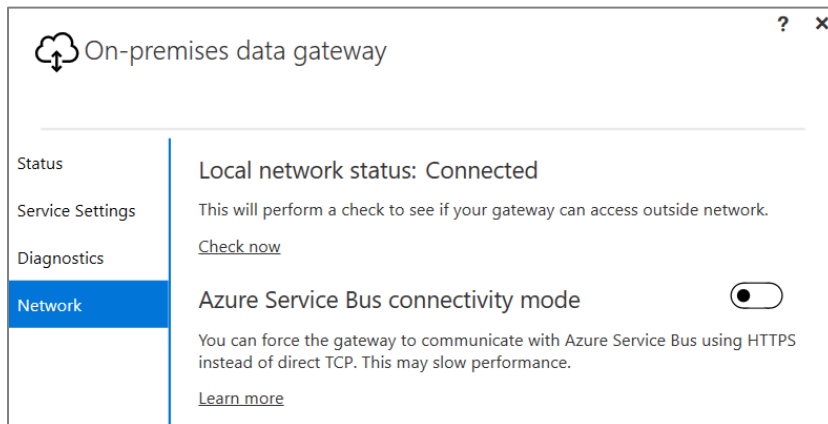
Where to Install the On-Premises Data Gateway

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.
- 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 IT server administrators prefer to use a machine dedicated to running data gateway activities, 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. The [Azure Speed Test](#) site can be used to check network latency between a client machine and an Azure region.
- A potential starting point for a gateway server is 8 CPU cores and 16GB RAM, as noted in the [Gateway Best Practices](#). Plan to scale up on CPU and/or RAM if bottlenecks present themselves, and/or scale out to multiple gateways if the volume of traffic becomes heavier than one data gateway can handle

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 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 5](#), 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.

Keeping the On-Premises Data Gateway Updated

Updates are not auto-installed for the On-Premises Data Gateway. It is highly encouraged to remain current with the latest data gateway version. Data gateway updates are released as often as once per month. Some updates may introduce new dependencies (for example on the .NET framework), so be sure to check whether this is the case before starting the update process.

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

The screenshot shows the Windows Event Viewer application. The left pane displays the 'Event Viewer (Local)' tree with 'On-premises data gateway service' selected under 'Applications and Services Logs'. The right pane shows a list of 6 events. The details pane for 'Event 0' is expanded, showing the message 'Started the 'On-premises data gateway service'.' and various event properties.

Level	Date and Time	Source	Event ID	Task Category
Information	5/28/2017 8:08:27 AM	On-premises data gateway service	0	None
Information	5/28/2017 8:42:23 AM	On-premises data gateway service	0	None
Information	5/27/2017 6:13:00 PM	On-premises data gateway service	0	None
Information	5/21/2017 3:56:44 PM	On-premises data gateway service	0	None
Information	5/21/2017 3:56:43 PM	On-premises data gateway service	0	None
Information	5/21/2017 3:56:34 PM	On-premises data gateway service	0	None

Event 0, On-premises data gateway service

General Details

Started the 'On-premises data gateway service'.

Log Name: On-premises data gateway service
Source: On-premises data gateway s Logged: 5/28/2017 8:08:27 AM
Event ID: 0 Task Category: None
Level: Information Keywords: Classic
User: N/A Computer: BISQVMI1Dev
OpCode:
More Information: [Event Log Online Help](#)

Verbose Logging and Log Exports

The screenshot shows the 'On-premises data gateway' configuration page. The 'Diagnostics' tab is selected in the left sidebar. The 'Additional logging' toggle is turned on. The 'Gateway logs' section indicates that logs will be exported to a single .zip file, with an 'Export logs' button.

Status

Service Settings

Diagnostics

Network

Additional logging ☒

You can enable additional logging to output queries and their timings to help understand what is performing slow. This will change the "TraceVerbosity" value from 4 to 5 in gateway core configuration file. It is not recommended to leave this setting enabled long term.

[Learn more](#)

Gateway logs

This will export all the gateway's configuration and service logs to a single .zip file.

[Export logs](#)

Logs	Gateway logs are typically available under: C:\Users\PBIgwService\AppData\Local\Microsoft\on-premises data gateway\Gateway*.log
Performance Counters	Relevant performance counters are documented here . We recommend setting up a process to extract performance counter data at a regular interval, and store it in a database so that time-oriented analysis can be performed. Having historical data, and benchmark data, for resources such as CPU and memory is extremely helpful for determining the state of performance issues.

Available for SQL Server Source Systems:

Extended Events	Extended events can be queried to retrieve a wealth of information about what is happening on a source system, such as SQL Server or Analysis Services. We recommend setting up a process to extract extended events data at a regular interval for the relevant performance counters, and store it in a database so that time-oriented analysis can be performed.
-----------------	---

Be sure to check these logs on a regular basis for problems: users may not always report issues when they encounter them so you should be proactive in this regard. It is common for a data gateway to be installed, yet not monitored or managed, in which case issues are not identified until functionality stops working completely.

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 is 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.

Data 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 8. Power BI 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 almost all scenarios. It allows you import data from the largest number of data sources, 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, though reports can be created in Power BI Desktop that are connected to a streaming dataset.
- Power BI Desktop is available as a Windows desktop application. It does not run natively on other platforms such as Apple Mac (though third party virtualization platforms can enable this scenario).

- 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. This is discussed in more detail in [Section 10](#).

Power BI Service

The web-based Power BI Service 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 an App 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 have to download a copy of it (as a file in .pbix format) if you want to keep a backup copy or store it in source control.

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.
- ✖ Some data sources that are available in Power BI Desktop are not available in Excel.
- ✖ Excel reports that use live connections to Analysis Services or Power BI Service live connections to published 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 some 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. In addition, in the future Excel will also support the [use of Power BI custom visuals](#) in a workbook.
- ✓ 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 / Source Control for Power BI Content

A lot of work can go into designing a dataset or a report, and you should take care that this work is not lost by making sure that you save a copy of your .pbix file somewhere safe after you make any changes. 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 repositories, 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 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 the files in and out as changes occur.

Because some version control systems impose file size limits (for example, by default GitHub places a limit of 100MB and OneDrive for Business imposes a 15GB max file size), 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. Typically template files are utilized for standardization of look and feel, so this usage of them will need to be documented and communicated to the team.

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.

Source control and versioning for Power BI Desktop files becomes particularly challenging when report changes are done directly in the Power BI Service. To minimize the risk of overwriting/losing changes, we recommend implementing a standard practice for editing reports and publishing them.

In the future, the ability to connect to Power BI datasets stored in Premium capacities using the XMLA protocol will allow tools that are currently used for Analysis Services development and deployment to be used for Power BI datasets as well. This will make it easier to store the definition of a dataset in source control.

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.

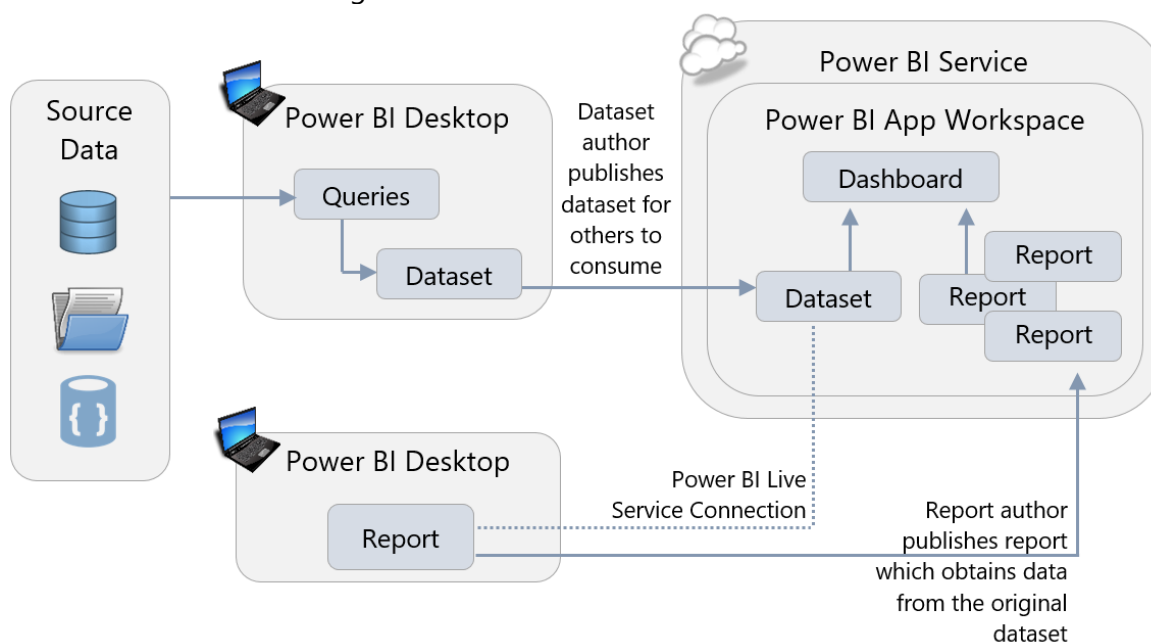
This can be achieved in two ways:

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 6](#) of this whitepaper.

2 **.pbix file with dataset only**

The second 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.

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, be aware that 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](#) based on either the file size or capacity size. 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 there is 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 in cases where there are tables with many date columns.

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.

Naming Objects and Q&A

For users who want to build their own reports it is important that tables, columns and measures have meaningful, human-readable names that are consistent with the terminology used throughout your organization. This is often not the case in the source data, even when you are using well-maintained data sources like a data warehouse: the naming convention used in a relational database may not make much sense to a business user. As a result you should take care to name all the objects in your dataset appropriately. Additionally, the Description property for fields can be utilized as a data dictionary to help report authors understand the data model.

This is also relevant if you have users who want to build their own visualizations using Q&A, Power BI's natural language query feature, rather than view pre-built reports. For Q&A to work well, however, you may also need to add Synonyms to your model and [edit the linguistic schema](#) to improve Q&A's ability to interpret a user's requests.

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 may begin utilizing “[high density sampling](#)” techniques 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 6](#).

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 they require. Alternatively, consider using slicers and filters in your reports to allow the user to select just the data that is needed to be displayed 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.

You may also consider using [drillthrough pages](#) to reduce the amount of data displayed on a page. This feature delays the rendering of detailed data until the user clicks on the link to drillthrough to an additional page (for instance, from a sales summary report to a sales detail report). In addition to improving performance, drillthrough can also simplify the higher level reports thus improving the end user experience.

Accessibility

When designing reports it is important to take into account the accessibility needs of your users. For example:

- Adding Alt Text to visualizations in a report helps users with low vision who use screen readers understand what a visualization is showing
- Similarly, choosing a color theme that has a high degree of contrast between backgrounds and text helps make reports easier to read, especially for users with low vision

- Remember that some users may be color-blind and therefore may not be able to see visual indicators such as using red and green to indicate good and bad results.

For more tips on creating accessible reports see [this article](#).

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 [AppSource](#) 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 AppSource have been through a validation process by Microsoft called Certification (for details on what this involves see [here](#)) that ensures their quality and safety, but there is no guarantee that all code paths have been tested. Custom visuals downloaded from sources other than AppSource 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.

Power BI administrators can create a repository of custom visuals that are approved for use within an organization by using the Organizational Custom Visuals feature. Approved custom visuals and specially developed custom visuals that are not available in AppSource can be uploaded via the Power BI Admin Portal in the service by an administrator, and when a user needs to import a custom visual when designing a report in Power BI Desktop they will see the list of Organizational custom visuals alongside the custom visuals available in AppSource.

Custom Data Connectors

It is 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 apply to using custom data connectors created by third parties: 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. At the time of this writing, custom data connectors are only supported for data refresh of published datasets if an On-Premises Data Gateway in personal mode is used.

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 Power BI Report Server, you can use dashboards to create a unified view of metrics from different reports created in different tools.

Developing for the Power BI Service and Power BI Report Server

Different versions of Power BI Desktop exist for use with the Power BI Service and Power BI Report Server. For users who publish content to **both** the Power BI Service and 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 are available to users approximately every 4 months in this scenario. This is the recommended approach because it's simplest for report authors to understand, and for administrators to manage.
2. Run two versions of Power BI Desktop side by side. This is not typically recommended. Running two versions is only suggested for early adopters who are keen to obtain new features as quickly as possible and are willing to accept the additional complexity of managing two versions. If a .pbix file from the newer version is inadvertently published to Power BI Report Server, it may not be able to render properly.

Section 9. Power BI Collaboration, Sharing and Distribution

Creating content (that is, datasets, reports, dashboards and Excel Workbooks) in Power BI is almost always a collaborative process, and content is usually intended to be used by more than just the people who created it. Power BI offers a number of different ways to collaborate, share and distribute content throughout an organization and it is important to understand which of these methods are appropriate for different scenarios.

The Role of Power BI Desktop

Reports built in Power BI Desktop are saved to .pbix files. It is very important to stress that .pbix files are not intended to be used as a method of sharing reports between multiple users in the way that Excel files often are, saved to a network file share or emailed around. There are several reasons for this:

- **Versioning** – There is always a risk that multiple copies of the same report will be created, which then means that changes and bug fixes will be made to one version and not another.
- **Stale data** – When data is imported, the data in a dataset held in Power BI Desktop can only be refreshed manually. This leads to a risk that the data in the report may be out of date.
- **Security** – Another effect of importing data into Power BI is that, since the .pbix file contains data that is very sensitive, it must be kept in a secure location. Distributing .pbix files containing imported datasets may present a security risk and may not conform to your organization's security policy. Also, although row-level security is defined in Power BI Desktop, it only works within the Power BI Service.
- **File size** – Power BI Desktop files may be very large, frequently larger than the maximum allowed size for an email attachment and large enough to make them slow to open when stored on a network file share.
- **Accidental changes** – While Power BI Desktop does have the option to lock visualizations in a report to prevent them from being moved accidentally during design, this option is not preserved when the file is saved which means that users are likely to move visualizations around accidentally. What's more there is no foolproof way to prevent users from making other changes to a report and saving them.
- **Printing** – Unlike the Power BI Service or Power BI Report Server, Power BI Desktop does not allow users to print reports at the time of this writing.

Therefore, the role of Power BI Desktop focuses on authoring of content. All collaboration, sharing, and distribution of content in Power BI should be done via the Power BI Service, Power BI Report Server, or via embedding content in a custom application.

Collaboration, Sharing, and Distribution in the Power BI Service

Workspaces in the Power BI Service

When a report author has finished developing in Power BI Desktop, he or she should publish the .pbix file to the Power BI Service. When a file 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 created for them automatically called "My Workspace", which is intended purely for personal use.
App Workspace	App Workspaces are shared workspaces where multiple users can collaborate on report development. App Workspaces must be specifically created.

A user's "My Workspace" should only be used for:

- **Personal usage.** Content that will only ever be consumed by the report designer, or content that is only to be shared with a very small number of other users on a temporary basis.
- **Testing purposes.** The report designer may use My Workspace temporarily to see what a report looks like in a browser or on a mobile device before it is published to an App Workspace.

In all other cases, reports should be published to an App Workspace. Using My Workspace for anything other than the scenarios described above can be dangerous: If a user leaves an organization their My Workspace area may be deleted and you risk losing any content that has been published there. Also, content within My Workspace can only be managed and maintained by that one individual.

Collaboration and App Workspaces

App Workspaces are where users can collaborate, but what does it actually mean to "collaborate" in the context of Power BI? There are a number of ways that collaboration can happen:

- Multiple users can work together to build one or more datasets, reports, dashboards, or Excel workbooks (collectively known as "content").
- One user can take over the design of content that was started by someone else.
- One user may work on the design of an imported dataset, while other users to build reports from that dataset using Power BI live connections in Power BI Desktop.
- Users may need to test (either in the formal, IT development sense of the word, or perform an informal "sanity check") a report even if they are not the person who is building it.

- Members of the same team, department, project, or even company will not just view content but discuss what it shows, take action on information shown, and suggest changes to existing content or create new content as circumstances change.

The key feature here is that collaboration involves a group of people taking an active part in the report design process rather than just being passive consumers of content. These people all know each other and are likely to work on the same team – with the term “team” defined in the loosest sense.

What Should an App Workspace Contain?

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

Scope:	Per Department or Team	Per Individual Report	Per Project, Type of Analysis, or Subject Area
Example:	Finance reports	Sales summary report	Quarter-end reports, or Product launch reports
Use of the Workspace:	Very broad (a large # of reports)	Extremely narrow (very few reports)	Medium (a manageable # of reports)

Per Department or Team

Create one App Workspace for each department or team in your organization, and put all the reports and dashboards needed by the people that work in that department inside the App Workspace. This technique is so broad that it’s not recommended for teams with a large amount of content.

- ✓ Makes it easy for users to find all the reports they need to access if they are all contained in a single App Workspace 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 App Workspaces which contain a large number of reports and dashboards, making it more difficult for users to locate specific items.
- ✗ Since there is currently a 1:1 relationship between an App Workspace and an App, a broadly-defined App Workspace will result in Apps for users which have a lot of content to navigate.
- ✗ If users need to see reports and dashboards from departments they do not work in, then managing permissions can become more complex.
- ✗ If some content authors need permission to edit some reports but not all of them in the App Workspace, that cannot be accomplished since permissions are at the App Workspace level.

Per Individual Report

Create one App Workspace for each report. This is not recommended except for very specific circumstances.

- ✓ Makes it easy to identify individual reports and dashboards.
- ✓ Permits fine-grained permissions related to editing of content (for instance, if a user is permitted to edit one report but not another).
- ✗ Will result in an exceedingly large number of App Workspaces being created, making finding the right one very difficult.
- ✗ Makes it difficult to share datasets between multiple reports, leading to duplication of datasets and increased maintenance across App Workspaces which serve similar purposes (although Power BI does intend to add the ability to use datasets across workspaces as a future capability.)

Per Project, Type of Analysis Conducted, or Subject Area

Create one App Workspace for each project, type of analysis, subject area or business question that your organization needs to answer, and put all the relevant reports and dashboards inside the App Workspace. Typically this is thought of as being a manageable compromise between the previous two options presented.

- ✓ Makes it easy to find reports and dashboards across departmental boundaries.
- ✓ A good compromise between App Workspaces that contain too much information and App Workspaces that contain too little.
- ✓ A good compromise with respect to permissions for content editors.
- ✗ Depending on how broadly it's defined, can still potentially lead to many App Workspaces being created, making finding the right one difficult.

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 of them is the name of the App Workspace: 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 an App Workspace in the Power BI Service or, if content from the App Workspace is published as an App, from Apps in AppSource.




Because content may be delivered via an App rather than the App Workspace, it is generally **not** recommended to include the word 'workspace' in its name.

Current Workspaces and New Workspace Experience

Up to this point in time, creating an App Workspace has also resulted in the creation of an associated Office 365 group. However, at the time of this writing, Power BI is in process of rolling out a new workspace experience designed to simplify workspace creation, make managing workspace membership easier and to lay the foundation for other new capabilities. As the new workspace experience rolls out, the current App Workspace infrastructure built atop Office 365 groups will remain in place and will gradually migrate into the new workspace experience. During the transition period, users will be able to choose the kind of workspace that suits their immediate needs. Once out of Preview, the new workspace experience will become the default way to create workspaces in Power BI. When creating a workspace using the new workspace experience, none of the other artifacts associated with an Office 365 group (such as Calendars, OneNote notebooks, or Team sites in SharePoint Online) will be created automatically when you create an App Workspace.

Below, details are provided for both types of workspaces that you can expect to encounter or utilize in your Power BI deployment.

 Since creating an App Workspace currently creates an Office 365 Group, naming policies set by the Office 365 administrator are enforced when the workspace is created. This behavior will be modified as the new workspace experience is introduced.

Settings and Permissions for an App Workspace

The following includes recommendations when specifying settings and permissions for an App Workspace. In each case the differences between the current (V1) workspace experience and the new workspace experience are described.

Visibility	<p>V1 workspace experience: Visibility can be set to either Public or Private. The recommendation here is that App Workspaces should be set to Private so that only users specifically added to the workspace can see the content. Otherwise, users may see App Workspaces that are not relevant to them and have trouble finding the App Workspaces that they do need. This setting is managed by Office 365 Groups with the V1 workspace experience.</p> <p>New workspace experience: All groups will be private. However, as described below, it will be possible to allow broad access to the workspace without needing to make it public by granting permissions to user groups rather than individual users.</p>
------------	--

Workspace Permissions	<p>V1 workspace experience: All members of a workspace can be granted permission to either edit the contents, or to only view the contents. This decision is made when the workspace is created. Since an App Workspace should be thought of as a place for collaboration, the typical recommendation is for group members to be able to edit its contents.</p> <p>At the time of this writing, users can only connect to a dataset published to the Power BI Service by using a Power BI Service live connection from Power BI Desktop, or Analyze in Excel, if users have permission to edit the contents of the workspace.</p> <p>Alternatively, members of an App Workspace (not administrators) can be given read-only access to the content in the App Workspace, turning it into a place for passive consumption of content for small teams of users. However, keep in mind that publishing an App from the App Workspace can be a much better means of delivering content when larger groups of consumers are involved.</p> <p>New workspace experience: you may specify which users can edit content and which users can only read content within the same workspace. This will improve the granularity of permissions you can set on workspaces, making them more flexible to meet enterprise needs.</p>
Users and Group Membership	<p>V1 workspace experience: Only individual users can be added to the App Workspace. When a user is added to an App Workspace, the user is also added to the Office 365 Group associated with the workspace.</p> <p>New workspace experience: Access may be granted to:</p> <ul style="list-style-type: none"> • Office 365 groups • Security groups • Mail-enabled security groups • Distribution lists • Individual users <p>This will allow you to manage workspace membership using the many different types of user groups that are in use throughout an organization.</p> <p>In the new workspace experience, the addition of a user to a workspace will not change Office 365 Group membership.</p>

App Workspace Administrators	<p>V1 workspace experience: Users can be ordinary members or administrators of a App Workspace. Each App 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.</p> <p>New workspace experience: you'll be able to give user groups rather than individual users administrator permissions to a workspace, ensuring you can manage administrators without needing to update individual workspaces.</p>
------------------------------	---

Options For Making Content Available To Users in the Power BI Service

As you have just read, adding users to an App Workspace is one way to make the content in that App Workspace available to users throughout your organization. There are other options available as well. While App Workspaces are a place for **collaboration** within a team, reports and dashboards may also be **shared** with other users outside App Workspace; finally, content may be **distributed** by publishing Apps from App Workspaces.

While the meaning of "collaboration" in the context of Power BI has already been discussed above, it is useful to understand the meaning of the terms "sharing" and "distribution" in the context of Power BI too. Sharing can be thought of as an act where one user gives another user, or a small group of users, access to a specific piece of content. Sharing usually takes place between users who know each other. Distribution, on the other hand, can be thought of as an act where a user pushes out a number of related pieces of content to a large number of users outside the team who may not have explicitly requested it but do need it as part of their job. The users who consume distributed content may or may not know the original creators of the content. As such, distribution is usually a more formal mechanism within the organization.

Sharing Dashboards and Reports

Both reports and dashboards may be [shared](#) – that is to say you can give one or more users outside an App Workspace read-only access to individual reports and dashboards by clicking on the Share button in the Power BI Service. When a user shares a dashboard with other users, the supporting reports, workbooks, and datasets are made available as well.

When you click on the Share button you must provide a list of the individual users, distribution groups, or security groups to share the report or dashboard with; the individual users may be inside or outside your organization. You also have the option of adding a message describing what has been shared, sending an email notification to individual users (though not groups),

and indicating whether or not to allow users with whom you have shared the content to share it with other people in turn. Users inside your organization who are recipients of shared content will see this content in the “Shared with me” area of the Power BI Service.

It is also possible to grant users access to content using the Manage Permissions pane, which is accessible from the Access tab of the pane that appears when you click the Share button. When you grant a user access to content in this way they will not receive any notification that they have access to the content and it will not appear in their “Shared with me” area.

Report and dashboard sharing is most appropriate for informal scenarios where content is to be shared with a small number of users in small organizations. It is a good idea to educate users to look into App Workspaces, rather than sharing, when attempting to distribute content to large numbers of users or people outside your team. This is because:

- It can be extremely difficult to determine which content has been shared with which users, since the permissions on each report and dashboard have to be checked individually.
- By default, users that have a dashboard or report shared with them can re-share the dashboard and report with other users. This presents the risk of content being shared with users that the original owner did not want to share it with. This can be prevented by unchecking the “Allow recipients to share your dashboard/report” option when sharing.
- Changes to reports and dashboards are viewable to others immediately, which may result in confusion for users if the content is a work-in-progress.
- When content has been shared from a user’s My workspace and that user leaves the organization, when the user’s account becomes disabled the recipients will lose access to their information.

Distribution via App Workspaces and Apps

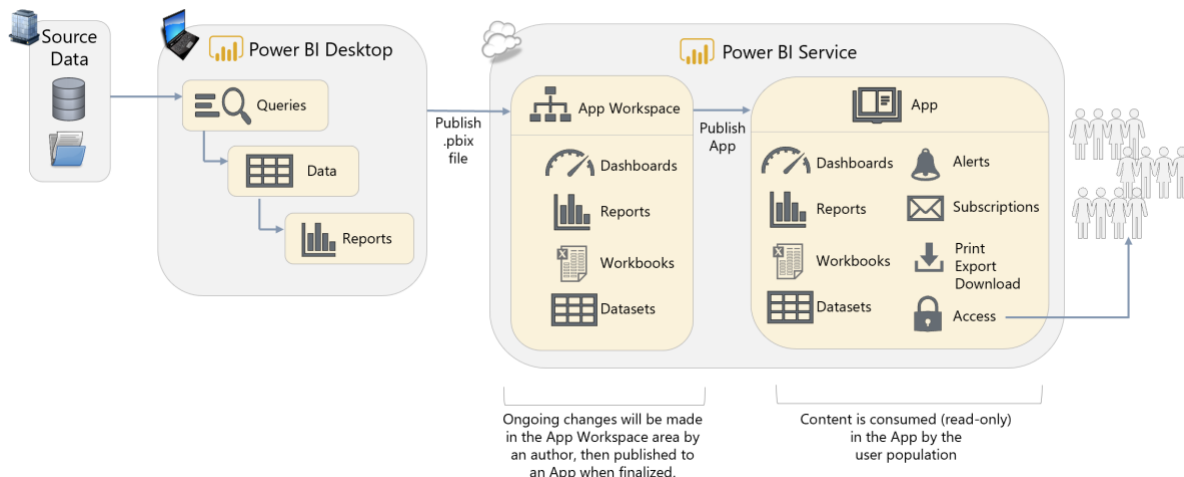
Content in an App Workspace may be distributed to a large number of users who are not members of that App Workspace by [publishing an App](#). An App is a read-only package consisting of datasets, reports, dashboards and Excel workbooks that originate from the same App Workspace. Only the administrators, or members with edit permissions of an App Workspace, can publish an App. For the purposes of this section, we will refer to this administrator or member with edit rights as the “content publisher.”


The process of creating an App is as follows:

1. An App Workspace content publisher chooses which dashboards, reports and Excel workbooks are to be included or excluded from the App.
2. The App Workspace content publisher adds a description for the App, a color scheme (which is completely independent of the look & feel within reports), and chooses a landing page (usually a dashboard) that will be displayed when the App is first opened.
3. The App Workspace content publisher then specifies whether the App will be available to the entire organization or just selected users or groups. In addition, the content

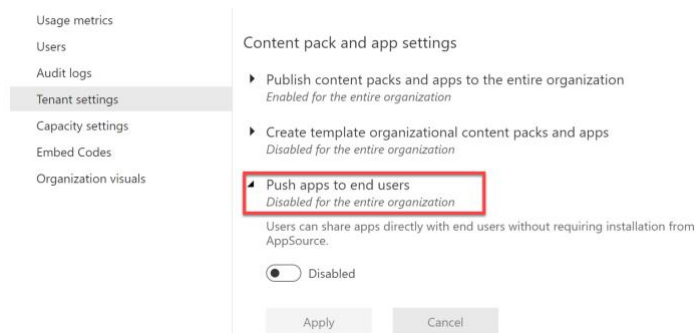
publisher can specify whether users must proactively install the App or whether users will have the App installed automatically for them (also referred to as “pushing the App”).

4. Finally, the content publisher publishes the App. When this happens the contents of the App Workspace that were selected in the first step of this process are duplicated and packaged up in the App. Any new changes to the dashboards, reports, and Excel workbooks in the source App Workspace do not affect their equivalents in the App until a content publisher explicitly updates and republishes the App.
5. When a user wants to view content, they can find it the Apps area of the Power BI Service rather than by going to an App Workspace. If the App has been installed automatically on their behalf, the user will find the App waiting there for them; otherwise the user will have to install the App manually by clicking the Get Apps button in the Power BI Service and finding the App they want in the AppSource website. After publishing an App, a link to the App is displayed and can be communicated in email or documentation, specified in an enterprise portal, and/or displayed on an internal web page to allow users to easily install the App.



 The ability to push Apps automatically to end users is turned off by default in the Power BI Admin Portal, and when it is turned on it can be restricted to certain users or groups.

Admin portal



The reason this is turned off by default is to stop users being overwhelmed with new Apps installed automatically for them by overzealous App Workspace administrators. Similarly, you can't push an App to the entire organization, again to prevent users from being overwhelmed. If you need to push an App to a broad set of users, create a user group such as a security group and push the App to the user group.

Distribution of content using Apps is in many cases a better choice than sharing, or adding large numbers of users as members of an App Workspace, in an enterprise environment:

- ✓ An App allows for the distribution of multiple related reports, dashboards, and Excel workbooks rather than individual pieces of content, simplifying permissions management.
- ✓ Changes to content are only visible once the App is published or updated. This means that reports and dashboards can be continue to be developed without affecting what is seen in the App.
- ✓ If an App Workspace is moved to a Premium capacity then users who consume an App built from that that App Workspace only need a Power BI Free license to do so. Power BI Free users cannot have direct access to an App Workspace in the Power BI Service, and although they can get access to content in an App Workspace stored in a Premium capacity that has been shared with them in other ways, an App is by far the easiest way to get the benefits of the Power BI Free license/Premium combination.

Distribution via Embedded Reports

An alternative way of distributing content to a large number of users is to embed a report within a portal or collaboration application. It is straightforward to embed a report in either Microsoft Teams or a SharePoint Online modern web page. It is possible to embed reports in a custom-built application too, but this involves a certain amount of development effort.

Embedding has a number of advantages and disadvantages compared to distribution via an App:

- ✓ Reports distributed in this way can be embedded in an existing, familiar portal (such as the corporate intranet site), so users do not have to go to the Power BI Service.
- ✓ Whereas a user with a Power BI Free license can only view an App if its associated App Workspace is in a P-Series Premium capacity, reports embedded in this way can be accessed by Power BI Free users if their associated workspace is in a more cost-effective EM-Series Premium capacity.
- ✗ Embedding a report takes a certain amount of effort, even if it is just the trivial amount needed to embed a report in a Microsoft Teams channel or a SharePoint Online modern web page. This makes distribution of reports in this way more time-consuming than using Apps.
- ✗ Using EM-Series Premium capacities does **not** allow Power BI Free users to access the content directly in the Power BI Service's portal, which can limit collaboration options for your teams. P-Series capacities are required to broadly distribute content to Power BI Free via the portal.



Recap of terms:

Collaboration is associated to multi-person content creation within an App Workspace.

Sharing is associated with the individual report and dashboard sharing feature in Power BI.

Distribution is associated with the ability to publish Apps from an App Workspace or the ability to embed content in portals or collaboration applications.

Sharing and Distribution with External Users Using Azure B2B

It is also possible to share dashboards, reports and to distribute using Apps with users outside your organization through [Power BI's integration with Azure Active Directory B2B](#). When you share or distribute with an external user a guest user account is created in Azure Active Directory and this allows users to sign in to Power BI to view your content using their own email address (either a corporate email account or a consumer email account such as Outlook.com). This is useful in scenarios where you need to make content available to customers and partner organizations.

At the time of writing, there are several restrictions on functionality with content that is shared to external users, notably that:

- External users cannot consume content through the Power BI mobile apps.
- External users will not see content shared from external organizations when they login to their own organization's Power BI Service portal.
- External users cannot edit content shared with them.

An external user who possesses a Power BI Pro license from their own organization can use the Power BI Pro license level to access content in external organizations. This 'bring your own license approach' decreases total cost of collaboration across organizations which all utilize Power BI Pro. If an external user does not have a Power BI Pro license, they can be granted a license by the organization which hosts the content, allowing the user to directly access content in an App workspace that is not in a Premium capacity. However, those external users still will not be able to edit content in the App workspace. To enable editing of content by external users, it is necessary to issue them login credentials to your organization and provision them with a Power BI Pro license.



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

Admin portal

Usage metrics
Users
Audit logs
Tenant settings
Capacity settings
Embed Codes
Organization visuals

Export and sharing settings

Share content with external users

Enabled for the entire organization

Users in the organization can share dashboards with users outside the organization.

☒ Enabled

Apply to:

☒ The entire organization

☐ Specific security groups

☐ Except specific security groups

Apply

Cancel

Distributing Reports Publicly on the Web

It is possible to make a report 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.

WARNING: The Publish to Web feature presents **serious security risks**. It should **not be used for data that is confidential** or proprietary in any way. A common misconception is that viewers can only see the summarized, visual-level data. This is not accurate – viewers they can still use 'See Records' to view detail rows. An experienced user could also craft their own queries and retrieve anything from the underlying data model. In the future, reports published in this way may be added to a public gallery of reports. As such it is not a secure way of making content available within an organization, and it not a substitute for sharing or distribution via Apps. If you need to make Power BI reports available over the web to external users, a more flexible and secure option is to use the external sharing capabilities through Azure B2B or Power BI Embedded.



Publish To Web is enabled by default, but can be turned off by a Power BI administrator in the Power BI Admin Portal. It is strongly recommended that it is disabled for all users by default in an enterprise environment, and only enabled for specific users and groups if there is a clear need for it.

Admin portal

Usage metrics
Users
Audit logs
Tenant settings
Capacity settings
Embed Codes
Organization visuals

Export and sharing settings

► Share content with external users
Enabled for the entire organization

✓ Publish to web
Enabled for the entire organization

Users in your organization can publish reports viewable by anyone on the web. Authentication is not available when viewing reports using Publish to web. Go to [Embed Codes](#) to view embed codes created by your organization. For more information, see [Publish to web from Power BI](#).

☒ Enabled

Apply to:

☒ The entire organization

☐ Specific security groups

☐ Except specific security groups

Apply

Cancel

Organizational Content Packs (Deprecated)

Organizational content packs are an older way for distributing content through an organization. They are now deprecated and will soon be removed from the Power BI Service. As such, they should no longer be used for new development, and existing organizational content packs should be replaced with Apps.

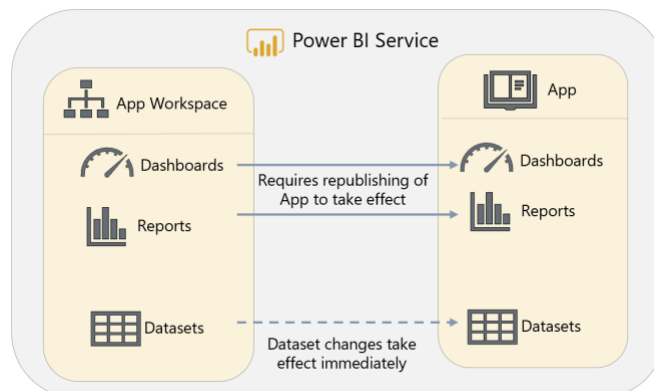
Controlling the Timing of Publishing Content

The three key benefits of publishing content from an App Workspace to an App are:

- ✓ Ensuring all content is read-only within the Power BI App.
- ✓ Managing access to all App content using a single consistent access list that supports user groups for easy distribution to broad audiences.
- ✓ Specifically controlling the timing of when report and dashboard updates are published and visible to the consumers of the App.

There are two additional caveats that are very important to understand.

When republishing a .pbix file to the App Workspace, any dataset changes always take effect in the App immediately. The App publish process does precisely control when report and dashboard changes are visible; however, dataset changes in the underlying App Workspace are always immediately visible in the App too.



This includes data refreshes and changes to DAX calculation logic. It also includes things that can break visuals, such as relationship changes or edits to column names.

Also, when republishing an App, all reports, dashboards, and workbooks that are set as “Included in App” in the App Workspace are always all republished. This can prove challenging in a collaborative work environment. For that reason, some teams find themselves ready to grow into using separate Development, Test, and Production App Workspaces.

Deployments To Development, Test, and Production

Enterprise IT standards usually require separate development, test, and production environments for BI, with a system of moving reports from development to testing, and from testing to production, only when it is appropriate to do so. Currently, there are several ways to handle this:

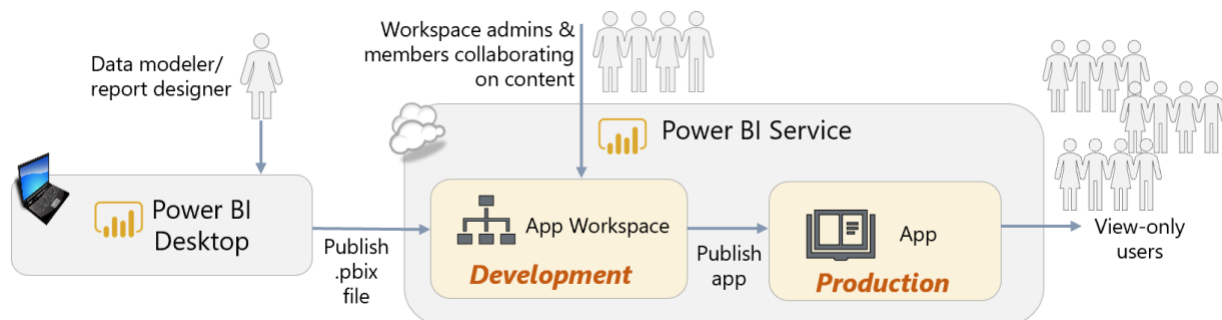
Option 1:

Dev = App Workspace; Prod = App

Publish Type = Manual

The first option involves utilizing an App Workspace as the Development area. The act of publishing or updating an App represents moving content into Production.

- ✓ This option is lightweight and easy to implement.
- ✓ It offers the ability to control exactly when the user base sees changes to reports and dashboards.
- ✗ A middle tier to handle testing is not available.
- ✗ If there are different data source connections (ex: to query from, say, a test system instead of a production system), that cannot be handled in this option.



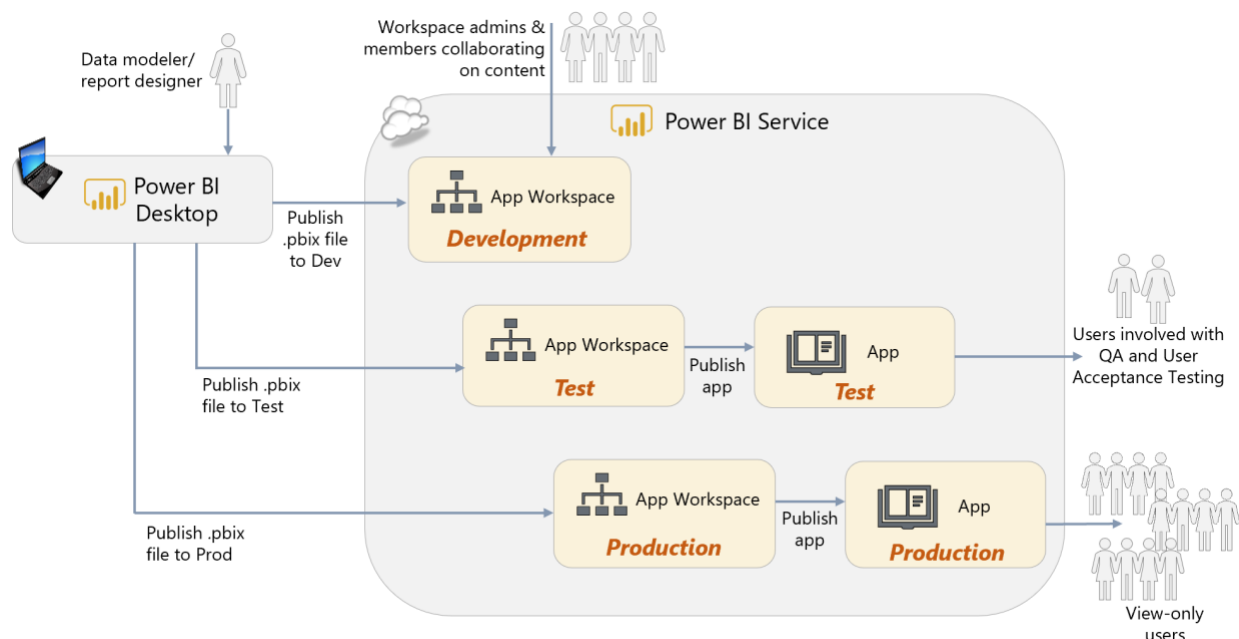
Option 2:

Dev, Test, Prod = Distinct App Workspaces + Distinct Apps

Publish Type = Manual

The second option involves more complex application lifecycle management. This option depicts the usage of separate App Workspaces for the separation of Development, Test, and Production.

- ✓ Since the different tiers are specifically enumerated, the number of environments needed can be easily customized to exactly what is needed.
- ✗ This option requires some processes/procedures, and efficient team communication, to execute seamlessly. IT teams are typically used to this type of rigor, whereas business-oriented teams are often less familiar with deployment processes.
- ✗ Since this technique relies on publishing from Power BI Desktop, usage of the Power BI Service for content creation (i.e., web only reports) is discouraged.
- ✗ A future feature coming to App Workspace management is the ability to copy content between App Workspaces. At that point in time, the publishing process beyond the Development environment would no longer involve Power BI Desktop. (See Option 3 next for a current alternative.)



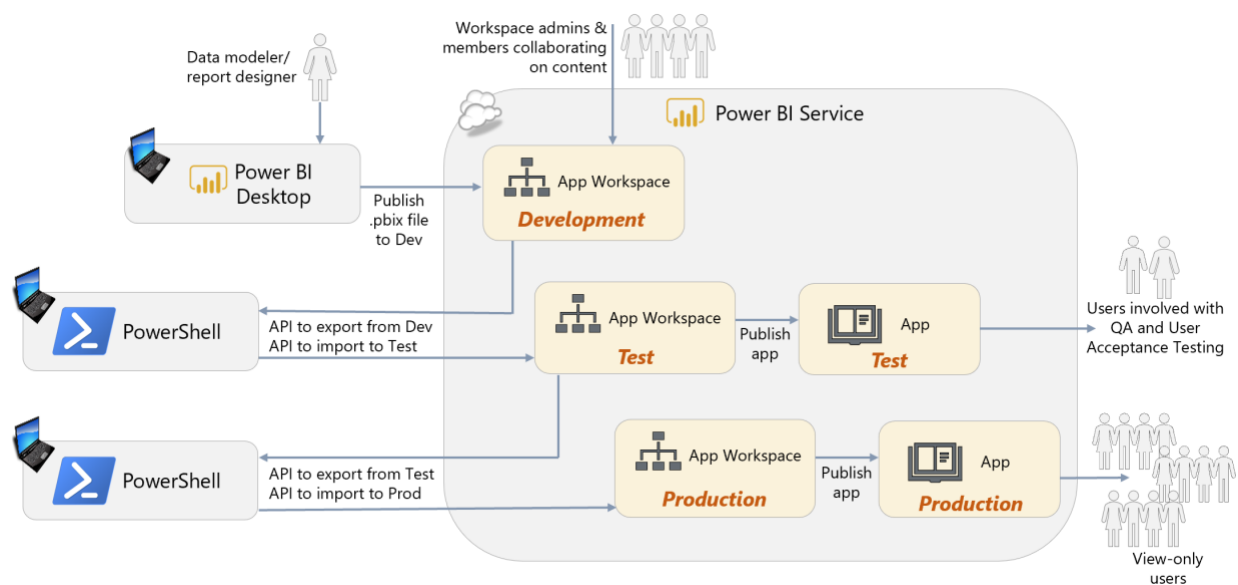
Option 3:

Dev, Test, Prod = Distinct App Workspaces + Distinct Apps

Publish Type = Automated

The third option also involves separate App Workspaces for the separation of Development, Test, and Production. However, instead of deploying from Power BI Desktop as shown in Option 2, the Power BI REST APIs are utilized to export content from a workspace and re-import to another. This technique is discussed in [this blog post](#), which also includes a sample PowerShell script.

- ✓ Since the different tiers are specifically enumerated, the number of environments needed can be easily customized to exactly what is needed.
- ✓ This technique changes the Test and Production deployment techniques to a programmatic process, which is considered a much more sound practice for deployments as compared to deploying from a client tool like Power BI Desktop.
- ✓ This technique clones all of the content from a workspace, so it can also deploy dashboards and web-only reports.
- ✗ Requires knowledge of using PowerShell to call the Power BI REST APIs.
- ✗ This option requires some processes/procedures, and efficient team communication, to execute seamlessly. IT teams are typically used to this type of rigor, whereas business-oriented teams are often less familiar with deployment processes.
- ✗ There is not currently an API to publish an App. Therefore, it is not possible to completely automate the full end-to-end process.



Suggestions to ease the implementation of options 2 and 3 above:

- Use parameters in the Query Editor to store data source connection information, if it differs between source Dev/Test/Prod locations. Parameter values can be updated in the Power BI Service, or through the REST API.
- Consider enabling certain people/groups to have the ability to “push apps” to users via the tenant settings in the Power BI admin center. These users would only push the Production App (since a large number of people will not see Dev or Test), and will greatly simplify the steps for end users to access the content. Since users need to manually add an App (when the push setting is not enabled) from the Power BI Service (i.e., it cannot be done via the Power BI mobile app), the push process can be particularly helpful for mobile members of the workforce.

Collaboration, Sharing and Distribution with Power BI Report Server

The options for collaboration, sharing, and distribution in Power BI Report Server are very different to those currently available in the Power BI Service. Instead, they work in the same way as SQL Server Reporting Services (SSRS).

Power BI Report Server is structured as a file system and all content that is published to it is either published to the root or to a folder; folders may contain subfolders. It is strongly recommended that you never publish to the root and that folders are created along the same lines as described for App Workspaces in the section above on [“What Should an App Workspace Contain?”](#) for different departments, teams, or business questions.

Access to content is controlled through Power BI Report Server’s role-based security model. Permissions can be applied at three levels:

- **Portal access.** This controls which users have access to Power BI Report Server.
- **Folder level.** This is the ideal level to control access because it provides control over all of the content in the folder (including any subfolders) with a single setting.
- **File level.** At the level of individual pieces of content such as a single Power BI report. In most cases this level of granularity is not necessary and it can make managing permissions tedious and error-prone.

When permissions are assigned, individual users, or preferably Active Directory groups, are assigned to roles, and the role determines what the user or members of the group are able to do. A full description of all of the built-in roles available in Power BI Report Server is available [here](#), but in summary:

- The **Content Manager** role provides full permissions to publish and delete content and also manage other users in Power BI Report Server. It is broadly similar to being an administrator or member for an App Workspace in the Power BI Service.
- The **Publisher** role allows users to publish content to Power BI Report Server. Think of this as similar to being a contributor of an App Workspace in the Power BI Service’s new

workspace experience, where contributors have the ability to edit the content of the App Workspace.

- The **Browser** role only allows users to view content in Power BI Report Server. This is similar to these types of capabilities from the Power BI Service:
 - Being a member of an App Workspace that is configured so that members are read-only, or
 - Having a report or dashboard shared with you, or
 - Being granted access to an App

Options for Consuming Content

After a user has been given access to content, the next question is to decide through which medium they should consume it: via a web browser, through the Power BI mobile apps, through SharePoint Online and so on. There is such a wide variety of ways that end users can consume content that they may not be aware of all the options that are available, and as a result they may need guidance as to which options are appropriate for them.

Viewing Reports and Dashboards in a Web Browser

The most common method for viewing published reports and dashboards is via a web browser. The majority of users will use this option most of the time and it provides the richest set of options for interacting with content.

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 it 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 functionality.

For organizations with large touchscreen devices running Windows 10 in their conference rooms, like Microsoft SurfaceHub, the Power BI Mobile App for Windows 10 offers the [Surface Hub Presentation Mode](#) feature. The presentation mode makes presenting on a large screen much more striking. It also allows the user to make annotations with support for Pen input 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 content (paginated and mobile reports) published to SQL Server Reporting Services or Power BI Report Server are integrated into the mobile apps as well.

Analyze in Excel

The [Analyze in Excel](#) feature of Power BI allows users to analyze data stored in datasets that have been published to an App workspace in the Power BI Service using Excel. It is not available for users of Power BI Report Server. For users that want to explore data, Excel PivotTables provide a much faster and more intuitive way of slicing and dicing data compared to any functionality that is currently 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 discussed earlier) 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 or change filters or slicer values, and data does not change after a refresh of the dataset- something that reports connecting to data stored in the Excel Data Model do not suffer from.

Embedded Reports in Microsoft Teams

[Microsoft Teams](#), a part of the Office 365 suite, is a collaboration application for teams of people working together on projects. Power BI reports can be added as tabs to a channel in Teams, and this allows members of a channel to discuss the contents of a report using threaded conversations. It is only possible to embed a report published to an App Workspace; it is not possible to display the contents of an App inside Teams. If Microsoft Teams is already widely used within your organization then using it as a means of distribution of Power BI reports can promote much greater engagement with those reports.

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 are also third-party SharePoint add-ins, for instance [Power BI Tiles](#), available in the Office Store which use the API in the background and

avoid the need for custom code. Since other types of reports, such as SQL Server Reporting Services reports and Excel-based reports, are often managed through SharePoint it might be easier for users to embed Power BI reports in SharePoint, too. This could be done to create a single enterprise reporting portal.

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 into 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 (in-app analytics).
- 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.

|

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 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.

Users can create subscriptions for other users (assuming the recipient has a Power BI Pro subscription). If the report is stored in a Premium capacity, the subscription can be sent to security groups or distribution lists.

Subscriptions are best utilized when:

- ✓ A point-in-time snapshot of the report is desired.
- ✓ The user does not need or want an interactive experience, so a PDF is adequate.
- ✓ The user prefers a report to be delivered on a schedule.
- ✓ Data security is not at risk of being compromised by distributing PDF reports to a set of recipients.

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 an effective way of making 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 10. Power BI Administration

The Power BI documentation contains a lot of detailed information on [administering Power BI](#) and configuring security. This whitepaper will not be duplicating content which is available in the online administration documentation. 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 is part of the larger Microsoft cloud ecosystem in Azure. If your organization is already using services such as Office 365 or Dynamics 365, then Power BI will be within the same tenant. This common tenant is built on Azure Active Directory.

Initial Power BI Tenant Setup

Creation of a Power BI Tenant

Ideally, planning a Power BI deployment should start before anyone within the organization has 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 whomever it is in the organization that 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 with a corporate email address. The first user to sign up creates a new auto-generated Power BI tenant, sometimes called a viral tenant or an unmanaged directory, 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](#).

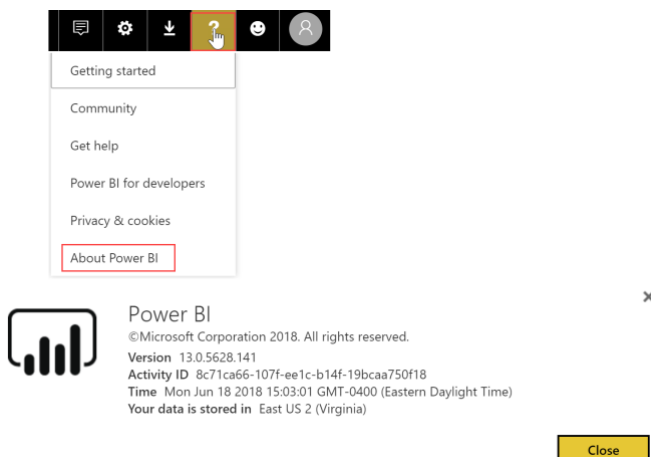
Geographic Data Region

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 currently not possible to move a Power BI tenant to another Azure data region. A future capability will be released to migrate content from one region to another. 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.

Multi-geo support is also coming in the future with respect to Power BI Premium, wherein different Premium capacity nodes may be provisioned in the data region of your choosing regardless of the region the Power BI tenant is located. The flexibility will enable organizations to meet data residency requirements, and improve ability to locate data near the user base.

Administrator Roles Related to Power BI

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

Type of Administrator	Scope	Permissions Available
Office 365 Global Administrator	Office 365	All administrative features for all Office 365 services.
Office 365 Billing Administrator	Office 365	Manages subscriptions and purchases services (such as Power BI Premium).
Power BI Administrator	Power BI Service	Access to the Admin portal in the Power BI Service, including tenant-level settings.
Reports Reader	Office 365	View activity reports and usage data in Office 365, including usage from Power BI.

Type of Administrator	Scope	Permissions Available
Power BI Premium Capacity Administrator	One or more Premium capacity nodes in the Power BI Service	Management of Power BI Premium capacity settings and workspaces assigned. View Premium usage measurements for CPU and memory. Access the Power BI Report Server product key.
Power BI App Workspace Administrator	A single App Workspace	Manage the list of members and admins for an App Workspace.
Power BI Embedded Capacity Administrator	Azure Power BI Embedded	Management of Power BI Embedded capacity settings.
Azure Active Directory Administrator	Azure Active Directory	Configure conditional access to Power BI and other security features

Global Administrators, via Office 365 or Azure Active Directory, will implicitly possess administrator rights in Power BI. An Office 365 Global Administrator can assign 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. Multiple administrators are necessary to cover vacations, sickness, and employees leaving the organization.



The Power BI Service Administrator role does not automatically grant the ability to assign licenses to users, or to view audit logs in Office 365 (which requires the Reports Reader role).

The Power BI Service Administrator role affects management capabilities across the entire tenant; the Power BI Service Administrator responsibilities cannot be delegated to individual groups or functional areas within the Power BI tenant. In some larger organizations, there is some friction between the centralized and decentralized Power BI development teams because the admin role is applicable tenant-wide. An example of this friction is when functionality like “push apps to end users” capability does not work and the user does not know why--a decentralized colleague may not be aware that there is a tenant-level setting to control it, nor is the colleague able to view or verify the current setting.

Power BI Admin Center

Tenant-Level Settings

Power BI Service Administrators have access to the [Power BI Admin Portal](#) which includes the following settings:

Property	Description
<i>Export and sharing settings</i>	
Share content with external users	Enabling this setting controls whether users may share dashboards and reports with others who are outside of the organization's domain. If enabled, users are able to do an "ad hoc invitation" to an external user which will auto-create that external user as a guest inside of Azure Active Directory.
Publish to web	This setting controls whether users are permitted to use the Publish to Web feature, which is an insecure way of distributing reports on the public internet. Recommendation: Disable this for the organization, and permit to groups based upon request.
Export data	This setting allows controls the following: <ul style="list-style-type: none">○ Export data from a tile or visualization○ Use of Analyze in Excel○ Use of Power BI Service live connection
Export reports as PowerPoint presentations	This setting permits exporting Power BI reports to PowerPoint files. Each report page is a slide in the exported file. Each slide contains an image of the report, with no interactivity.
Print dashboards and reports	This setting controls whether the print options are available when viewing a report or dashboard in the Power BI Service.
<i>Content pack and app settings</i>	
Publish content packs and apps to the entire organization	This setting permits publishing Apps (from App Workspaces) or Organizational Content Packs (which are deprecated) to everyone in your organization. Recommendation: Since this is applicable to the entire organization, we recommend this be disabled to avoid accidental publishing to the entire organization.
Create template organizational content packs and apps	This setting relates to Template Content Packs. Note that Organizational Content Packs are now deprecated.

Property	Description
Push apps to end users	The ability to push Apps is controlled by this setting. Pushing apps automatically can simplify and improve the end-user consumer experience. This setting is disabled by default. Recommendation: Enable this setting for the teams that you know are using Apps for broad distribution. Meanwhile, educate the App Workspace owners/editors to use this capability judiciously so as not to confuse users by having too much unexpected content continue to show up.
Integration settings	
Ask questions about data using Cortana	This setting allows users to utilize Cortana on their Windows device to search for data within published reports and dashboards.
Use Analyze in Excel with on-premises datasets	This item will control if Analyze in Excel is able to be used. Analyze in Excel allows users to connect from Excel on the desktop to published Power BI datasets.
Use ArcGIS Maps for Power BI	Enabling this setting permits users to utilize the ArcGIS mapping visualizations provided by Esri, subject to Esri's terms and conditions.
Use global search for Power BI	This setting allows the use of Azure Search features, in conjunction with Cortana, to locate information in published reports and dashboards.
Custom visuals settings	
Custom visuals	This setting specifies whether users across the organization are permitted to utilize custom visuals.
R visuals settings	
Interact with and share R visuals	This setting specifies whether users across the organization can interact with and share visuals created with R scripts.
Audit and usage settings	
Create audit logs for internal activity auditing and compliance	Enabling this setting will permit actions taken in Power BI to be captured in the audit logs in Office 365. Recommendation: Enable this setting so that activity is captured for auditing purposes.
Usage metrics for content creators	This setting permits individual users to view the usage metrics related to the dashboards and reports they generated. Recommendation: Enable this setting so that individual content creators are able to understand the effectiveness of the solutions they create.
Per-user data in usage metrics for content creators	This setting specifies whether the names and e-mail addresses of users are exposed via the usage metrics.
Dashboard settings	

Property	Description
Data classification for dashboards	Configuring this list will provide a pre-defined list of data classifications which users can assign to dashboards. When used, the data classifications are also captured in the Office 365 audit logs.
<i>Developer settings</i>	
Embed content in apps	Disabling this setting will block an application from using the multiplexing ability of Power BI Embedded with an embed token. This is a security feature to prevent any user within an organization from purchasing Power BI Embedded (an Azure service) and embedding content outside of Power BI without the appropriate approvals.
<i>Dataflows</i>	
Create and use dataflows	This setting controls whether or not dataflows are permitted to be created and utilized as a data source by Power BI.

Some tenant-level settings are enabled by default, while others are disabled by default.

Certain tenant settings can be specified for users and groups, whereas other settings are enabled or disabled across the entire organization.

Managing Distribution of Organizational Custom Visuals

Another responsibility of the Power BI Administrators is managing the approved custom visuals. Because of the nature of custom visuals, they do have inherent privacy or security risks. To mitigate the risk, there should be an approval process, including technical assessment for code quality, for all organizational custom visuals.

There are three types of custom visuals:

- **Created by Microsoft** – these are 100% approved Microsoft code and trustworthy
- **Microsoft certified visuals** – these are created by third-party software vendors or community members, but they have been tested and inspected by Microsoft to work in all scenarios (note: the certification process does not represent a full security certification, but it is verified that no external URLs are present in the code which could be malicious)
- **Other custom visuals** – these are created by third-party software vendors and community members with no checks made by Microsoft

Within the Power BI Admin Portal, the organization visuals feature provides a way to easily distribute a set of custom visuals throughout the organization:

Admin portal

Usage metrics

Users

Audit logs

Tenant settings




Capacity settings

Embed Codes

Organization visuals

Organization visuals

Add new custom visuals for your organization or manage them. [Learn more](#)

NAME	PUBLISHED BY	LAST UPDATED	ACTIONS
 Liquid Fill Gauge	Adam Wilson (adam@[REDACTED].com)	Apr 10, 2018	 
Add a custom visual			

Organization visuals may be either proprietary visuals that are not available in the public custom visual marketplace in AppSource, or publicly available visuals that have been officially approved for use by Power BI authors within the organization. Any Power BI administrator can upload a custom visual (a .pbiviz file), so using this capability effectively involves having a good process in place to evaluate and approve visuals. Once a visual has been added, users will be able to find it in the [Organizational Visuals Store](#). One important benefit of organizational visuals is that if you use it to distribute custom visuals that are not available in the public custom visual marketplace (for example, custom visuals that you have bought from a third-party vendor), any reports that use them will be automatically updated with a new version of the visual when the new version is published to the Power BI Admin Portal.

If an organization visual is deleted from the Power BI Admin Portal, it will become disabled in all downstream reports and dashboards which utilize it.



If you are thinking of creating a custom visual, even if you do not intend to make the custom visual publicly available, we recommend following all the [guidelines for getting a custom visual certified](#).

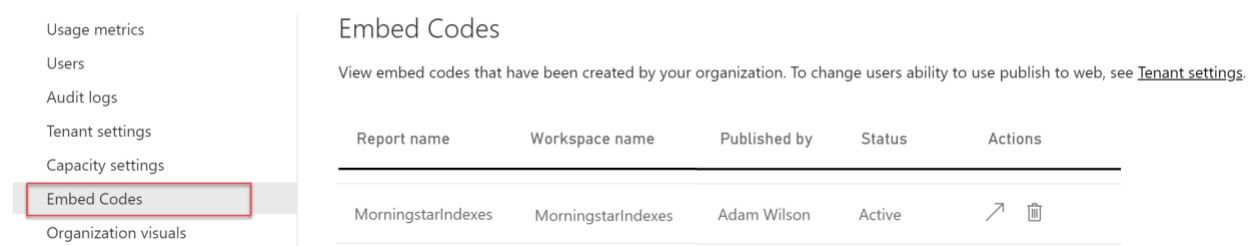
The organization visuals capability can be disabled within the tenant settings in the Power BI Admin Portal. The use of organization visuals is either enabled or disabled for the entire organization.



Managing Embed Codes for Publicly Available Reports

As discussed earlier in this whitepaper, one capability that is available to all users (even those with a Power BI Free license) is the Publish to Web capability. We do recommend this capability be disabled for all users and only enabled on-demand. This feature is intended for data journalism and other public visuals – for example, Microsoft uses this to display quarterly financial data to the public.

All reports which have an embed code associated with them, across all workspaces, can be viewed in the Power BI Admin Center:

Admin portal



Report name	Workspace name	Published by	Status	Actions
MorningstarIndexes	MorningstarIndexes	Adam Wilson	Active	 

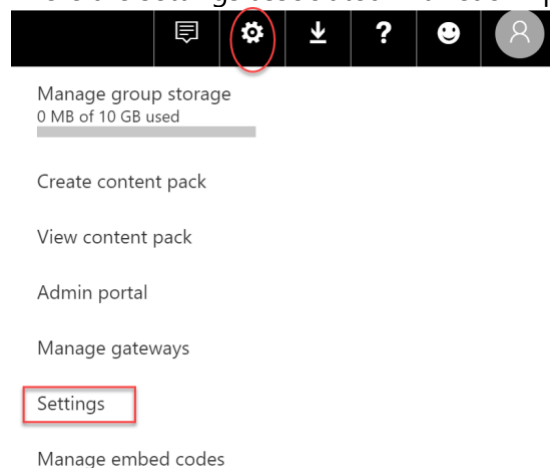
The arrow under the Actions column displays the public web page.



Although you may feel that an internal website is “safe enough” to use the Publish to Web capability, that technique is strongly discouraged. Anyone, anywhere, with the embed code can view the report. Viewers can not only see the report data, they can still use ‘See Records’ to view detail rows. An experienced user could also craft their own queries and retrieve all data available from the underlying data model.

App Workspace Settings

There are settings associated with each App Workspace as well as My Workspace:



The workspace settings are where you can obtain a list of all dashboards, datasets, workbooks, alerts, subscriptions, and dataflows:

General Dashboards Datasets Workbooks Alerts Subscriptions Datapools

Privacy

Language

Close account

Developer

ArcGIS Maps for Power BI

Privacy Settings

Search Term Collection

☒ Help improve Power BI by sharing the search terms you used.
[Learn more](#)

Apply Discard

Monitoring Usage of Power BI

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 information. 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 are used the most, potentially being candidates for moving into Power BI Premium capacity.
- To look for suspicious usage patterns that indicate a user is exporting data for unauthorized purposes. Alternatively, perhaps users are exporting data because the reports as designed are not providing what they need.
- 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.

Power BI Service Usage

There are three types of auditing related to Power BI, starting with the smallest scope to the broadest scope:

Type of Auditing	Scope	Targeted Towards	Where to Locate	Retention Period
Usage metrics	Individual reports + dashboards	Report authors and editors	Usage metrics icon in the page header for report or dashboard	90 days
Usage metrics	Power BI tenant	Power BI administrator	Power BI Admin Portal	90 days
Audit logs	Power BI tenant + other Office 365 services	Office 365 administrator	Office 365 Admin Portal	90 days

Usage Metrics for Individual Reports and Dashboards

The [usage metrics applicable to individual reports and dashboards](#) are available from a built report available in the Power BI Service. This usage reporting is targeted towards report authors so they can see usage patterns and assess the impact of the solutions they have published:



In order to customize this report, a user who possesses edit rights for the workspace can do a File > Save As which will create a copy of the report, and a copy of the dataset, within the same workspace. The following options are now available to create a customized usage report:

- **Edit the report in the browser.** The report can be edited and customized directly within the web browser. If the filter for one specific report/dashboard is removed, the scope becomes usage for the entire current workspace. You can then add individual report fields, such as the name of the dashboard to make the customized usage report work in the context of the entire workspace.

- **Connect to the dataset from Power BI Desktop.** You will not be able to download the .PBIX file from the Power BI Service which was generated from the Save As operation. However, you can connect to the dataset from Power BI Desktop using the Power BI Service live connection capability in order to create your own customized report from scratch within Power BI Desktop. The associated dataset is quite simple and cannot be altered or edited currently.

With the above techniques, you can create customized usage reporting for:

- All reports in a workspace
- All dashboards in a workspace
- One specific report or dashboard in a workspace which is particularly crucial
- Shared reports in a workspace
- Apps in a workspace

The important thing to remember is that, using this approach, the broadest range is still a workspace and the history element is still 90 days. For the most flexible and customizable approach, see the Audit Logs in Office 365 discussed later in this section.

Usage metrics are captured for activity which occurs within the Power BI Service. Activity which occurs via embedding scenarios or Publish to Web will *not* be reflected. Also, the usage metrics capability is unavailable in the US Govt cloud implementation of Power BI.



This type of usage reporting can be disabled in the tenant settings in the Power BI Admin Portal. We recommend enabling usage metrics for the entire organization.

There are two tenant-level settings, one which enables the metrics to be captured, and one which allows the activity for individual users to be captured and identified.

Usage metrics for content creators

Enabled for the entire organization

Users in the organization can see usage metrics for dashboards and reports they created.

☒ Enabled

Apply to:

☒ The entire organization

☐ Specific security groups

☐ Except specific security groups

Apply Cancel

Per-user data in usage metrics for content creators

Enabled for the entire organization

Usage metrics for content creators will expose display names and email addresses of users who are accessing content.

☒ Enabled

Apply to:

☒ The entire organization

☐ Specific security groups

☐ Except specific security groups

Apply Cancel

Usage Metrics for the Power BI Tenant

The Power BI Admin Portal provides a built-in usage metrics dashboard which focuses on displaying high level metrics for activity across the entire Power BI tenant.

This dashboard differs from the usage metrics reporting discussed in the previous section because it traverses beyond the scope of one report or one workspace.

This particular dashboard provides a quick high level view of the most popular dashboards, reports, users, and groups but it does have several limitations:

- This dashboard cannot be modified or customized
- The tiles do not drill-through to an underlying report
- There are no filters or interactivity

Admin portal



For tenant-wide usage reporting, an alternative to using the built-in usage metrics within the Power BI Admin Portal is the [solution template available in App Source](#). Be sure to review the terms and conditions, as there is a subscription fee and licensing agreement from the third party provider of this solution template.

Audit Logs in Office 365

The most powerful and thorough usage reporting is available via the [audit logs in Office 365](#). However, it also involves additional effort and [additional Office 365 and Exchange Online permission levels](#) beyond the Power BI Administrator role. If enabled, up to 365 days of history (based on UTC datetime) is available from the Office 365 audit log (the extent of history available depends upon how the technique used to extract the data).



There are some limits to the number of API calls which can be made. To avoid experiencing incomplete data or throttling, we recommend the process to extract usage logs be a centralized activity.

Integration with Office 365 is an advantage because every individual service within Office 365 does not have to implement its own unique way of logging.

The audit data is in JSON format, which involves parsing out the names and values for each element. Following is a sample of the output produced from the Office 365 audit log:

CreationDate	UserIds	Operations	AuditData
2018-06-18T15:44:27.0000000Z	mcoates@contoso.com	ViewDashboard	{ "Id":"3de2cfcf-2b0d-4001-9d41-ee79e7ff6048", "RecordType":20, "CreationTime":"2018-06-18T15:44:27", "Operation":"ViewDashboard", "OrganizationId":"a5d8b9d4-07b2-455f-914e-0b9a9f0c04bb", "UserType":0, "UserKey":"10037FFEA191AFD7", "Workload":"PowerBI", "UserId":"mcoates@contoso.com", "ClientIP":"69.132.22.167", "UserAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87 Safari/537.36", "Activity":"ViewDashboard", "ItemName":"Customer Profitability", "WorkspaceName":"Customer Analysis", "DashboardName":"Customer Profitability", "WorkspaceId":"516051e4-3276-476a-88b3-91ebe125ed71", "ObjectId":"Customer Profitability", "DashboardId":"fa3db701-8357-471a-a6ec-8a9887c10c7f", "Datasets":[{"DatasetId":"03000ed9-6629-4d50-9a50-ff801b7f8130", "DatasetName":"Customer Profitability"}], "IsSuccess":true, "DatapoolRefreshScheduleType":"None"}
2018-06-15T00:56:30.0000000Z	cwebb@contoso.com	ViewReport	{ "Id":"991fd973-c62a-4a74-9ddd-472718b1c632", "RecordType":20, "CreationTime":"2018-06-15T00:56:30", "Operation":"ViewReport", "OrganizationId":"a5d8b9d4-07b2-455f-914e-0b9a9f0c04bb", "UserType":0, "UserKey":"10030000AB61218F", "Workload":"PowerBI", "UserId":"cwebb@contoso.com", "ClientIP":"62.144.42.117", "UserAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.87 Safari/537.36", "Activity":"ViewReport", "ItemName":"Product Sales Analysis", "WorkspaceName":"Product Sales", "DatasetName":"Product Sales Analysis", "ReportName":"Product Sales Analysis", "WorkspaceId":"c945a0d5-af10-432d-bae1-c0506e2472a6", "ObjectId":"Product Sales Analysis", "DatasetId":"fe162969-0a16-4ae1-bd4a-efedc7ba3be3", "ReportId":"f10a31b3-b5b2-496f-80da-

At the time of this writing, over 70 Power BI events are captured by the Office 365 audit log. Key data elements captured by the audit log include (but are not limited to) the following:

- User email address
- User IP address
- User agent (ex: browser version, mobile app)
- Activity performed by the user (ex: view report, view dashboard, dataset refresh schedule updated)
- Object name and ID (ex: applicable report, dashboard, dataset, or workspace)
- Datetime of the activity performed
- Capacity ID (Premium or Power BI Embedded)
- Gateway ID and Name

Via the Office 365 audit log you can determine which users are checking [usage metrics for their individual reports and dashboards](#), which was previously discussed.

There are a few ways you can approach accessing this data:

1. **Office 365 Admin Portal.** From within the Office 365 Admin Portal, the audit log can be searched, and results can be exported to a CSV file. This approach is highly limited because:
 - a. It is an entirely manual process.
 - b. It is constrained to 90 days of history.
 - c. There are too many events within the Power BI category to choose them all using the user interface (currently over 70 to select from).
 - d. The export is limited to 1,000 rows.
2. **API Call from Within Power BI.** You can call an API (discussed in step 3b below) from a connection made directly from within Power BI. The JSON data can be parsed and structured into a dataset for reporting. However, the issue with this approach is data

longevity. When calling the API, only 7 days of history is available so this “direct to Power BI approach” has limited value unless you are only focused on current activity.

3. **Programmatic Exports to Data Storage.** Export the Office 365 data to a location where the history can be stored long-term. This storage could be a database, a data lake, or object storage depending on your preference. Typically, this approach is recommended so that longer time-oriented analysis can be performed. When exporting, there are two techniques:
 - a. **PowerShell cmdlets:** Using the [Search-UnifiedLog](#) cmdlet within PowerShell returns records from the audit log. This technique has a 5,000 row record limit. Although you could implement a looping strategy to overcome the record limit, it is generally recommended that the PowerShell cmdlet approach be reserved for occasional ad hoc export operations rather than automated exports on a schedule. Using this technique you may be able to retrieve history as far back as 365 days. However, due to API limits, retrieving large amount of history can be challenging in large organizations.
 - b. **Office 365 Management Activity API:** Using the set of [Office 365 REST APIs](#) is the recommended approach for programmatically downloading data from Office 365 on a schedule because it is more scalable. Per the [API schema](#), retrieving just Power BI audit events can be done with an AuditLogRecordType of 20. By default, the API returns 24 hours of data; a maximum of 7 days history can be retrieved. Note that the API can be called from within PowerShell or the tool of your choice for loading into data storage.

Once the audit log history has been written to the data storage of your choice, then Power BI can be effectively used to handle all of the reporting capabilities (known in the industry as “BI on BI.”)



Audit logs will be captured in Office 365 if enabled in either the Office 365 portal, or if enabled in the Power BI Admin Portal.

We recommend audit logs be enabled in all cases.

Audit and usage settings

- ▲ Create audit logs for internal activity auditing and compliance
Enabled for the entire organization

Users in the organization can use auditing to monitor actions taken in Power BI by other users in the organization. Power BI audit logs are now always available for tenants that have enabled recording user and admin activity in the Office 365 Admin Portal.

☒ Enabled

Apply

Cancel

This setting applies to the entire organization

Power BI Report Server Usage

The capabilities for tracking report executions for Power BI Report Server are consistent with how it's handled for SQL Server Reporting Services server running in native mode. Specifically, a database view within the Report Server catalog database called "[ExecutionLog3](#)" can be queried with T-SQL commands. The execution log captures information to help you determine:

- How often a report is executed.
- Duration of report execution, including time spent for data retrieval vs. processing the report vs. rendering the report.
- What output formats are being used to render a report.
- The extent that e-mail subscriptions are utilized.
- Parameter values selected when rendering a report.
- Data size and row counts returned from a report.

Because Power BI Report Server is an on-premises implementation (whereas the Power BI Service is a cloud-based Software-as-a-Service), there are also additional logs for monitoring and managing the environment which are discussed in [this documentation](#).

Automating and Managing Power BI Programmatically

In addition to the user interface discussed elsewhere in this whitepaper, there are two primary techniques for managing Power BI in a programmatic way:

- PowerShell cmdlets, and
- Power BI REST APIs (which can be called via PowerShell or the tool of your choice)

PowerShell Cmdlets for Power BI

The PowerShell Gallery, the central repository for PowerShell content, contains a [small set of Power BI cmdlets](#) which have been released. At the time of this writing, the current version is 1.2, and the following cmdlets are available:

```
1 Get-Command -Module Microsoft.PowerBI.PowerShell
```

CommandType	Name	Version	Source
Function	Add-PowerBIDataset	1.2	Microsoft.PowerBI.PowerShell
Function	Add-PowerBIRows	1.2	Microsoft.PowerBI.PowerShell
Function	Connect-PowerBI	1.2	Microsoft.PowerBI.PowerShell
Function	Get-PowerBIDataSets	1.2	Microsoft.PowerBI.PowerShell
Function	Get-PowerBIGroups	1.2	Microsoft.PowerBI.PowerShell
Function	Get-PowerBITables	1.2	Microsoft.PowerBI.PowerShell
Function	New-PowerBIColumn	1.2	Microsoft.PowerBI.PowerShell
Function	New-PowerBIDataset	1.2	Microsoft.PowerBI.PowerShell
Function	New-PowerBITable	1.2	Microsoft.PowerBI.PowerShell
Function	Remove-PowerBIRows	1.2	Microsoft.PowerBI.PowerShell
Function	Switch-PowerBIContext	1.2	Microsoft.PowerBI.PowerShell
Function	Update-PowerBITableSchema	1.2	Microsoft.PowerBI.PowerShell

Resources with documentation and sample scripts include:

- [PowerShell modules targeting Power BI on GitHub](#)
- [PowerShell Modules for Power BI from the PowerShell Gallery](#)
- [Trello Board, run by community members, for voting and commenting on desired PowerShell cmdlets for Power BI](#)

For the purposes of managing a Power BI environment, the PowerShell cmdlets are not sufficient. There are significantly more options available via the Power BI REST APIs.

PowerShell Cmdlets for Power BI Report Server

The PowerShell Gallery contains a set of [cmdlets which can be installed for managing Power BI Report Server](#). The PowerShell cmdlets can be used for various tasks such as:

- Export reports
- Deploy new or updated reports
- Deploy reports through the environments (dev/test/prod)
- Modify connection strings and passwords stored for data sources
- Create a folder
- Back up the encryption key

Resources with documentation and sample scripts include:

- [PowerShell Utilities for Managing Report Server on GitHub](#)
- [PowerShell Modules for Power BI Report Server from the PowerShell Gallery](#)

PowerShell Cmdlets for On-Premises Data Gateway

The PowerShell Gallery contains a set of [cmdlets for use with managing the On-Premises Data Gateway](#). These cmdlets can be used for tasks such as:

- Retrieve list of gateway clusters
- Retrieve list of gateway instances registered in a cluster, including online/offline status
- Modify the enabled/disabled status for a gateway within a cluster
- Delete a gateway

Resources with documentation and sample scripts:

- [January 2018 Data Gateway Updates](#)
- [PowerShell Modules for On-Premises Data Gateway from the PowerShell Gallery](#)

Power BI REST APIs

The [Power BI REST APIs](#) provide service endpoints for embedding, administration, and user resources. At this time, the following groups of Power BI REST APIs are available:

REST Operation groups

Operation group	Description
Admin	Operations for working with administrative tasks.
Available Features	Operations that return available features.
Capacities	Operations for working with capacities.
Dashboards	Operations for working with dashboards
Datasets	Operations for working with datasets
Embed Token	Operations for working with embed tokens.
Gateways	Operations for working with gateways.
Groups	Operations for working with groups.
Imports	Operations for working with imports.
Push Datasets	Operations for working with push datasets.
Reports	Operations for working with reports.

The Power BI REST APIs can be used to handle many common administrative activities. For example:

- Download and/or clone objects in a workspace, for the purpose of migrating from Dev to Test environments, or from Test to Prod
- Retrieve metadata about an On-Premises Data Gateway, for instance to locate connectivity status issues or find gateways which need to be updated
- Check refresh history for a dataset
- Trigger a data refresh for a dataset
- Grant user permissions to an App Workspace
- Take ownership of a dataset
- Re-bind a dataset to a new On-Premises Data Gateway
- Locate a list of workspaces on dedicated Premium capacity

Resources with documentation and sample scripts include:

- [Power BI REST API Reference](#)
- [Power BI PowerShell Samples on GitHub](#)
- [What Can Developers Do with the Power BI API?](#)
- [Power BI Developer Samples in GitHub](#) (focuses on embedding scenarios)

Following are suggestions and tips for getting started with using the Power BI REST APIs.

An [App Registration in Azure Active Directory](#) is required to use the Power BI REST APIs. This is required for embedding scenarios, as well as when you want to call APIs via PowerShell scripts or other means. You can register the AAD app one of two ways:

Choice A:

The Power BI App Registration page at <https://dev.powerbi.com/apps>

The App Type selection should always be Native app, even if you are working with a web app in relation to Power BI Embedded. This is because only the Native app choice can handle the granting of permissions properly.

If you use the Power BI App Registration page, only a subset of permissions are exposed in step 3. You'll still need to visit the Azure portal in order to assign permissions for the Power BI Service.

Power BI for Developers

Register an Application for Power BI

Register a new application that can be used to call Power BI APIs

Step 1: Login to your Power BI account
Welcome, **Melissa Coates**! (Wrong account? No problem, logout and try again.)

Step 2: Tell us about your app
Let's start with some basic details.

App Name:

App Type:

Redirect URL:

Step 3: Choose APIs to access
Select the APIs and the level of access your app needs.

Dataset APIs	Report and Dashboard APIs	Other APIs
<input checked="" type="checkbox"/> Read All Datasets	<input checked="" type="checkbox"/> Read All Dashboards	<input checked="" type="checkbox"/> Read All Groups
<input checked="" type="checkbox"/> Read and Write All Datasets	<input checked="" type="checkbox"/> Read All Reports	<input checked="" type="checkbox"/> Create Content
	<input checked="" type="checkbox"/> Read and Write All Reports	

Step 4: Register your app
Once you've set everything the way you want it, click the button below and we'll register your app. Your client ID and secret (for web apps only) will appear below. Be sure to copy the values into your app. By clicking the Register App button, you have accepted the [terms of use](#).

Register App

Client ID:

Choice B:

Azure Portal - App Registrations for Azure Active Directory at <https://portal.azure.com>

The required permissions for API access need to be specified for this AAD App.

If you are using one AAD App for all Power BI-related management activities, you want to select all of the delegated permissions. Alternatively, you may wish to treat this like a service account and instead create different AAD Apps for different purposes, with different delegated permissions. For example, an AAD App specifically for managing

App Registrations

Overview | Quick start | Users | Groups | Enterprise applications | Devices | App registrations | Application proxy

PowerBI_PowerShellMgmt_ReadWrite_AADApp

Application ID: a72396fa-3b76-4864-b6e9-832fa9aa34

Object ID: 8b6c434-9118-42ba-bd41-47125b4672d

Managed application in local directory

Create Service Principal

Settings

Display name: PowerBI_PowerShellMgmt_ReadWrite_AAD...

Application type: Native

Home page: ...

Required permissions

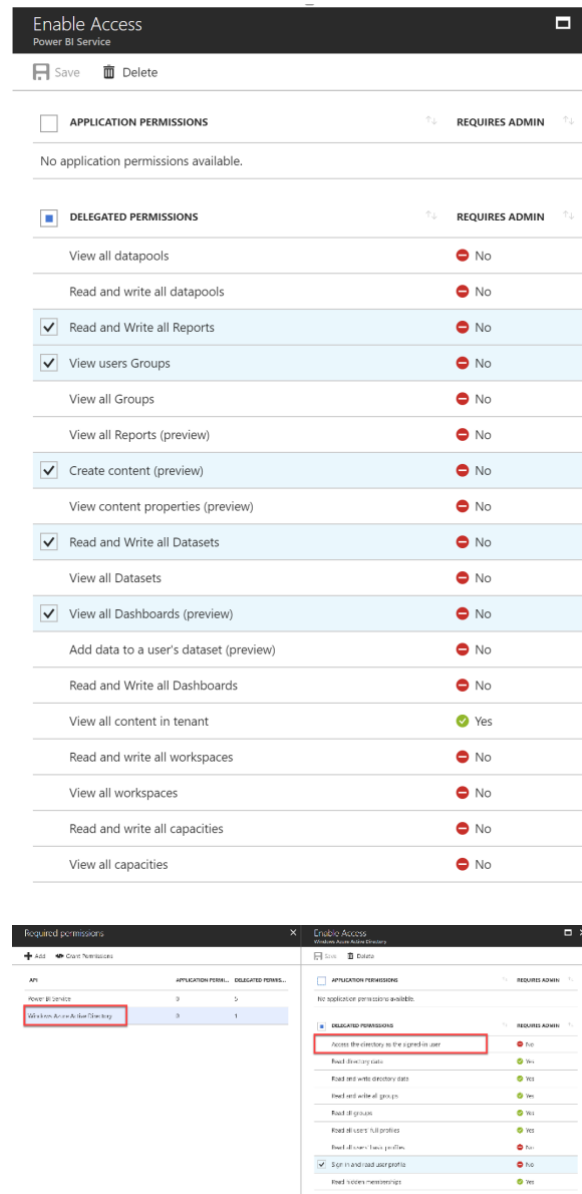
API	APPLICATION PERMISSIONS	DELEGATED PERMISSIONS
Power BI Service	0	5
Windows Azure Active Directory	0	1

Premium Capacity, and a specific AAD App for managing dataflows.

The Grant Permissions step that is done here to enable access for the delegated permissions is in lieu of receiving a one-time security prompt that accepts permissions. This is the only way currently to set permissions for an AAD Native App such as this.

If you are a global AAD administrator, when you grant these permissions it will apply to all users in your tenant. Therefore, it is recommended that you first sign-on as a specific account (ex: a service account, or a master account you're using for Power BI Embedded). When signed in as a non-global AAD administrator, these permissions will apply only to that specific user.

Additionally, in the Windows AAD permissions, also need to select the "Access the directory as the signed on user."



For a secure enterprise solution, we recommend creating a separate AAD App for dev, test, and production environments.

Office 365, Azure Active Directory, and Microsoft Graph REST APIs

In addition to the Power BI REST APIs discussed previously, there are also REST APIs from Office 365, Azure Active Directory and Microsoft Graph which are very helpful for managing user and licensing aspects of Power BI, such as:

- Retrieving audit logs from Office 365
- Assigning Power BI Pro licenses
- Find users who have signed in to Power BI
- Find users with Power BI licenses assigned who have not signed in
- Manage Power BI Embedded capacities

Resources with documentation and sample scripts include:

- [Using Power BI Audit Log and PowerShell to Assign Power BI Pro Licenses](#)
- [Office 365 API Reference](#)
- [Microsoft Graph APIs](#)
- [Azure AD Graph API](#)
- [Azure Resource Manager \(ARM\) REST API Reference for Power BI Embedded](#)



Many of the examples you will find online use PowerShell modules which have become deprecated. A common example of this are the MSOnline modules which have “Msol” in the cmdlet name. Be sure to verify examples and scripts online.

Managing Power BI Premium

Managing Power BI Premium Capacities

Within the Power BI Admin Portal, the following can be viewed and updated as it relates to one Premium capacity:

Admin portal

Power BI Premium > Contoso Capacity Settings

USAGE IN THE LAST 7 DAYS

ⓘ We've made some updates to the Capacity usage metrics. [Learn more](#)

Metric	Status
CPU	Exceeded 80% utilization 0 times
Memory Threading	Exceeded 80% utilization 0 times
Memory Usage	Average: 1GB
Direct Query	Exceeded 80% utilization 0 times

Learn more about [Usage measurements](#)

CAPACITY SIZE
This capacity is a P1, which is 8 v-cores.
[Change capacity size](#)

REGION
North Central US

USER PERMISSIONS

- ▶ Capacity admins
- ▶ Users with assignment permissions
Enabled for a subset of the organization

WORKSPACES (38)

Search content...

Showing 38 item(s) Sort Workspaces by...

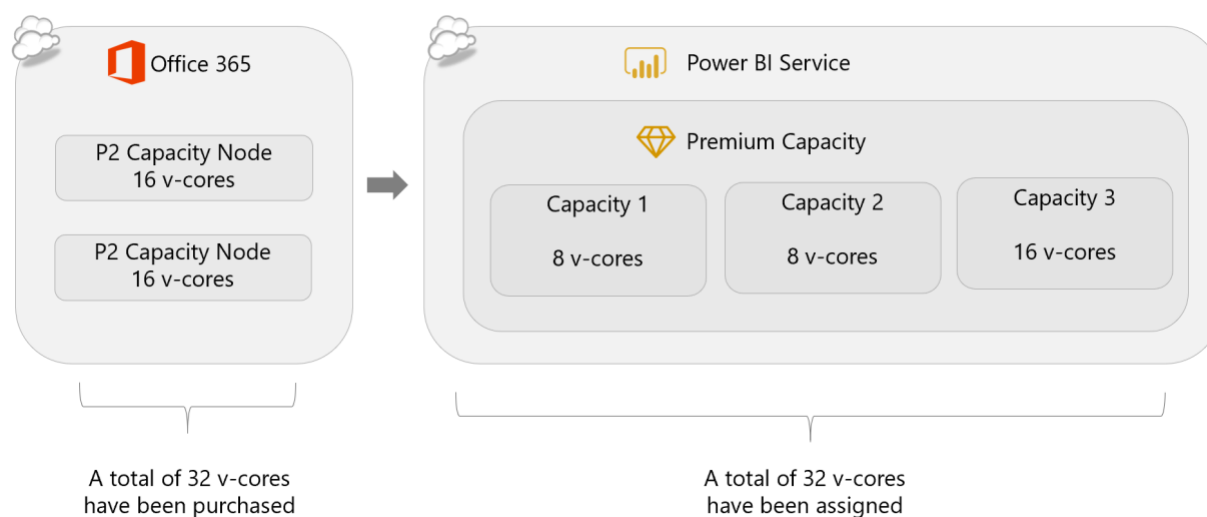
WORKSPACE NAME	WORKSPACE ADMINS	ACTIONS	STATUS
Finance Reports	View admins	×	Assigned
Fraud Detection	View admins	×	Assigned

In the above image, a capacity called Contoso Capacity has been created with 8 v-cores. Thirty-eight workspaces have been added to this node, the first two of which are shown in the screenshot.

Managing Capacities in Power BI

When you purchase a Power BI Premium node, your tenant will receive the corresponding number of virtual cores (v-cores) for use in running capacities. Before being able to take advantage of these resources, one or more capacities need to be set up in the Power BI Admin Center.

The capacities defined in the Power BI Service do not have to be allocated exactly 1:1 with the capacity nodes that were purchased. This is depicted in the following image, in which 32 purchased v-cores are assigned to 3 capacities created in the Power BI Admin Center:



Assigning Workspaces to Power BI Premium Capacity

Once a capacity exists in the Power BI Admin Center, one or more workspaces must be assigned to it to take advantage of its benefits. This can happen in several ways:

- Assign an individual workspace to a capacity.
- Supply a list of users or security groups and assign all workspaces of which these users are administrators to a capacity.
- Assign all workspaces to a capacity, making it the default storage location for all workspaces.

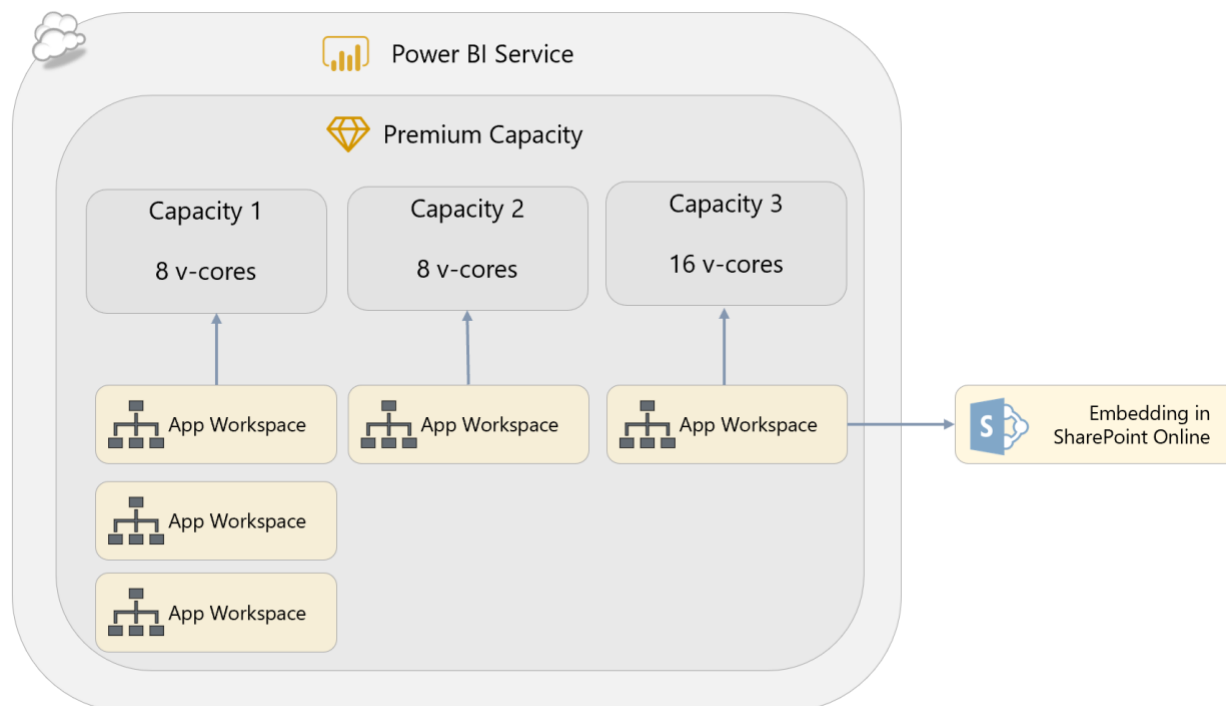
Workspaces may be moved from one Premium capacity to another, or they may be moved out of Premium capacity back to Power BI shared capacity.

If the entire organization's Power BI content can fit into a Premium capacity then it may make sense to do that. Some companies tend to leave My Workspace for each user in shared capacity and only consider App Workspaces for Premium capacity.

Workspaces may 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. Within Premium capacity, the allowed dataset size ranges currently from 1GB-10GB depending on the SKU purchased. See the Limits section for details.
More Frequent Data Refresh Required	A dataset can be refreshed up to 8x/day in App Workspaces in shared capacity. Within a workspace assigned to Premium Capacity there is a limit of 48x/day that a dataset can be refreshed.
Short Data Refresh Time Window	By default, Power BI does a full refresh for all tables within an entire dataset each time the refresh executes. Within a workspace assigned to Premium capacity, incremental refresh can be configured, which significantly reduces the load on the source system and speeds up the time it takes to refresh data stored in Power BI.
Number of Passive (Read-Only) Viewers	Users who only need to view App content distributed from a workspace in Premium capacity can view content with a Free license rather than a Pro license. Therefore, moving certain App Workspaces which have a higher number of read-only consumers into Premium capacity may save the organization on licensing costs.

For the 32 v-cores discussed a few moments ago, workspaces could be assigned to each capacity node to balance out specific workloads:



For certain App Workspaces that have extremely inconsistent and variable usage, you may choose to overprovision a capacity node. This technique only works when there is little chance of concurrent activity across the App Workspaces.

Assigning Capacity Admins and Capacity Assignment Permissions

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.

Capacity Admin

A Power BI Premium Capacity Administrator can perform the following tasks within a capacity:

- Grant capacity assignment permissions
- Grant workspace permissions
- Bulk assign workspaces to capacity
- Assign and remove workspaces from capacity
- Monitor capacity usage

A capacity administrator is assigned within the scope of one single capacity within the Power BI Admin Center (ex: to Capacity 1). All Office 365 Global Administrators and Power BI Administrators implicitly have Capacity Admin rights to all capacities.

A Capacity Admin does not have permissions back in Office 365 to change the SKU or billing; their permissions extend only to managing the capacity settings within the Power BI Service. Specifically, that includes adding App Workspaces to a capacity and adding users with assignment permissions. If an App Workspace from multiple functional teams or divisions is sharing capacity, it may be helpful to have a member from each of the teams as a capacity administrator. Alternatively, you may elect to centralize all capacity administrator responsibilities.

Users with Assignment Permissions

Capacity assignment permissions can be delegated to certain users (who must have a Power BI Pro license) so they can:

- Assign App Workspaces to a capacity, provided they are an administrator of that workspace
- Allow other users with Power BI Pro licenses to access these App Workspaces

This is helpful functionality in order to delegate responsibility for moving App Workspaces in and out of capacity. If you prefer instead to implement a workflow or require a ticket in your tracking system, then you will likely not make use of this permission setting.

Monitoring Capacity Resources

There are 4 types of resources specifically monitored in the Power BI Admin Center:

CPU	Displays the # of times that CPU utilization has exceeded 80% over the last 7 days. CPU can be high when data refresh occurs, and from report query activity.
Memory Thrashing	Measures how many times a dataset was evicted from memory in the last 7 days. When a Premium Capacity is over-provisioned (which can be a valid strategy), yet usage among multiple datasets is consistent, the least recently used dataset is evicted from memory. Memory thrashing occurs when a dataset is evicted from, then reloaded to, memory constantly.
Memory Usage	Displays the average amount of memory usage over the last 7 days. Premium capacity memory is consumed by: <ul style="list-style-type: none">• Memory used by the loaded datasets• Memory used by dataset refresh• Memory used by report queries
DirectQuery	Shows the number of times that DirectQuery or live connections exceeded 80% of the limit allowed per SKU over the last 7 days. Connections are throttled above the specified limit.

Additional details for each resource can be viewed in a drill-through chart, and can be exported.

More details on monitoring Premium capacity can be found [here](#).

Deploying and Managing Installers and Applications

There are several downloads available directly within the Power BI Service:



Power BI Desktop

Data Gateway

Power BI for Mobile

Power BI publisher for Excel

Analyze in Excel updates

Power BI Desktop

Nearly all users who develop Power BI content will require Power BI Desktop to be 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. The 64-bit version is

strongly recommended: 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](#).

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 from 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 after each release.

There are 3 ways to install Power BI Desktop:

1 **Web-based MSI**

Pros:

- ✓ Timing controlled by the owner of the PC.

Cons:

- ✗ Requires administrative rights on the machine to install from an MSI.
- ✗ Introduces the risk that different user machines will not be synchronized, leading to compatibility issues.
- ✗ Every update requires action by the owner of the PC.

2 **Microsoft Store**

Pros:

- ✓ Updates are automatically installed in the background when they come out.
- ✓ Administrative privileges are not required for the installation to succeed.
- ✓ Download sizes are smaller and more efficient.
- ✓ Can be integrated with an organization's roll-out policies and available via Microsoft Store for Business.

Cons:

- ✗ Does not provide for the same degree of control as System Center Configuration Manager.
- ✗ Requires Windows 10.
- ✗ Only applies to Power BI Desktop which is updated monthly--Power BI Desktop Optimized for Report Server is only available from the Web.

3 **Organizational IT deployment tools such as Microsoft System Center Configuration Manager**

Pros:

- ✓ The process is managed and handled by the organizational IT dept.
- ✓ Offers the most control.

Cons:

- ✗ More complex.



A side-by-side installation of Power BI Desktop from the web (item 1 above) and the Microsoft Store (item 2 above) is not supported. It is possible to do, just not supported.

No subscription or license is required to install or use Power BI Desktop. However, users of Power BI Desktop will be prompted to log in for the best experience.

Power BI Desktop runs two processes: one being a front-end process to render reports, and another back-end process to fetch data and perform calculations. The front-end process communicates with the back-end process using TCP. Error reporting from Power BI Desktop is now integrated with Windows Error Reporting, as announced [here](#).

Power BI Mobile Apps

The Power BI mobile apps can be managed and mass deployed to users' corporate mobile devices using [Microsoft Intune](#). Policies such as requiring an access PIN and other data access policies can be configured.

We recommend assessing mobile user needs at the beginning of a Power BI project, because some users will strongly prefer to consume Power BI content through the mobile apps rather than through a browser. This availability can make a big difference in adoption rates and user satisfaction.

Mobile apps are available for iPhone, iPad, Android phone, Android tablet, and Windows 10 devices. For users with Windows 10 PCs, especially those with touch screens such as Surface Pros, the Power BI Mobile App for Windows offers a slightly different user experience which some users may prefer.



If a mobile device is stolen or lost, we recommend logging into the Power BI Service to disable all data-driven alerts. In addition, if using Intune to manage the app or the device, you can activate selective-wipe to remove the Power BI mobile application and all relevant data from the mobile device.

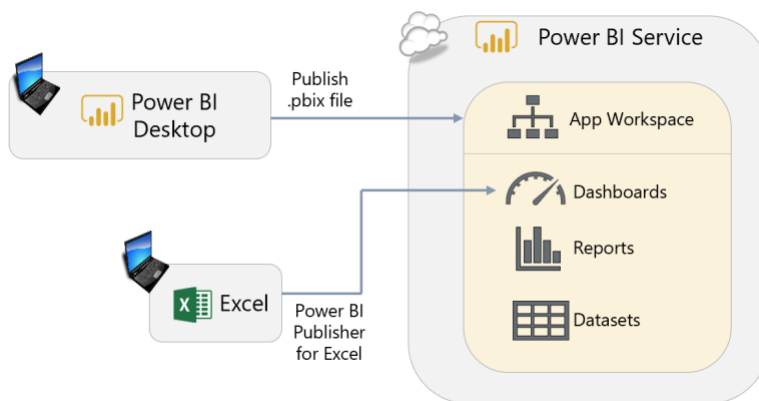
On-Premises Data Gateway

The [On-Premises Data Gateway](#) acts as a bridge, providing quick and secure data transfer between on-premises data within your corporate network and Power BI.

Please refer to the "[Power BI Data Refresh and Data Gateway](#)" section of this whitepaper for all details and recommendations for managing the On-Premises Data Gateway.


Power BI Publisher for Excel

Some users may wish to install an add-in for Excel called the [Power BI Publisher for Excel](#). This allows for pinning of objects such as PivotTables, PivotCharts, and cell ranges to a dashboard in the Power BI Service:



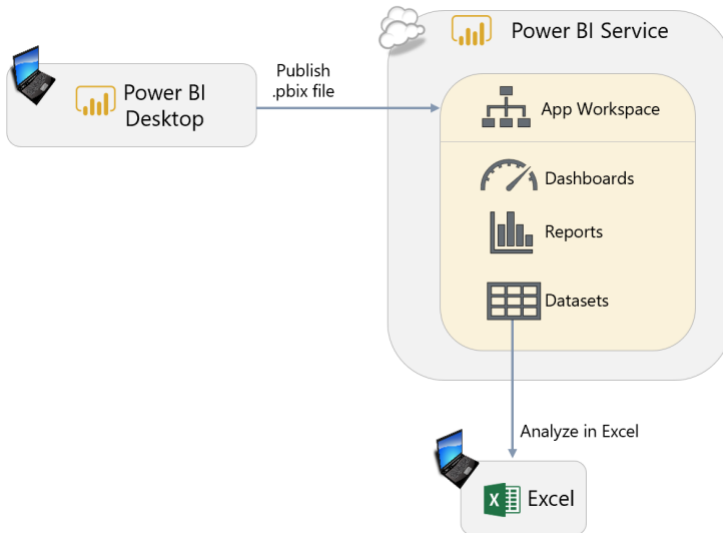
This capability allows for the creation of composite dashboards which contains objects from within local Excel files as well as the Power BI Service. Usually integration between products is a positive aspect; however, if this behavior is of concern, it can be prevented via group policy: "Disable UI extending from documents and templates."

We recommend storing Excel workbooks used in this manner in a centralized area, such as OneDrive for Business so they are available to the same team that manages the App Workspace.

 OneDrive for Business currently has a 15GB data size limit.

Analyze In Excel

The Excel experts in the company will appreciate being able to utilize the [Analyze in Excel](#) capability, discussed earlier in this document:



This capability requires installation of the Microsoft Analysis Services OLE DB Provider for Excel. These libraries do need to be updated from time to time, as discussed in [this troubleshooting page](#).

Usage of Analyze in Excel can be prevented via group policy.

Firewall Settings

The On-Premises Data Gateway section communicates with [Azure Service Bus](#) as follows:

- Inbound ports: N/A
- [Outbound ports](#): TCP 443, 5671, 5672, 9350-9354

When using [Bing Maps in Power BI](#), you may need to update your firewall to allow access to the following URLs which Bing uses for geocoding:

- <https://dev.virtualearth.net/REST/V1/Locations>
- <https://platform.bing.com/geo/spatial/v1/public/Geodata>
- <https://www.bing.com/api/maps/mapcontrol>

Managing Power BI Report Server

As discussed earlier in this whitepaper, Power BI Report Server allows you to publish Power BI reports to an on-premises server. An existing SQL Server Reporting Services implementation may be migrated to Power BI Report Server (if appropriate licensing is obtained). Note that for running other components of SQL Server, such as Analysis Services or Master Data Services, an appropriate separate SQL Server license is required. Typically you will want to run those

workloads on a separate server. If licensing for Power BI Report Server was acquired via a Power BI Premium SKU which allows for hybrid deployment scenarios, then you may be responsible for administering Power BI, both on-premises and in the Power BI Service in the cloud.

One advantage to Power BI Report Server is the ability to host Excel reports in the web-based portal, without the need to configure and maintain SharePoint Online or SharePoint (this capability does require Office Online Server and Analysis Services in PowerPivot mode).

Resource planning guidance for a Power BI Report Server implementation can be found [here](#).

Guidance for integration with the Power BI Service can be found [here](#).

Managing Users, Licenses, and Power BI Trials

See the [Power BI Licensing section](#) of this whitepaper for discussion of users and licenses.

Monitoring Service Health

Power BI Support Site

Status for the Power BI Service is posted on the Power BI Support site:

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

Issues relating to the sovereign/national clouds (US Govt, Germany, China) are typically not included in this status page. Rather, any issues there are noted in the Office 365 site.

<u>Service Status Per Region</u>	
✓	Europe
✓	America
✓	Asia Pacific
✓	Good
⚠	Degraded
❌	Outage
i	Information

Azure

Status for Power BI Embedded comes from the Azure status page:

<https://azure.microsoft.com/en-us/status/>

Azure status

Last updated 8 seconds ago

Refresh every

2 minutes

Good

Warning

Error

Information

Americas

Europe

Asia Pacific

Azure Government

PRODUCTS AND SERVICES	NON-REGIONAL*	EAST US	EAST US 2	CENTRAL US	NORTH CENTRAL US	SOUTH CENTRAL US	WEST CENTRAL US	WEST US	WEST US 2	CANADA EAST	CANADA CENTRAL	BRAZIL SOUTH
Event Hubs		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Power BI Embedded	✓											
Azure Analysis Services		✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Data Catalog		✓					✓	✓				
Data Lake Store			✓	✓								

Up to 30 days of history for service incidents and advisories is available:

Home > Service health

History: Last 30 days

2018-06-20 17:11 (UTC) [View current status](#)

Search

Id	Title	Service	Status	User count	Start time (UTC)
MO1419...	Office 365 application connectivity issues	Office 365 Portal	Service restored		2018-06-15 12:40
MO1417...	Access and rendering issues with the admin portal	Office 365 Portal	Service restored		2018-06-12 17:00
PB138058	Excel Online refreshes not working	Power BI	Service restored		2018-05-17 00:02

PB138058 - Excel Online refreshes not working

Status: Service restored

User impact: Users may have been unable to successfully refresh Excel Online workbooks with on-premises data models.

Latest message: Title: Excel Online refreshes not working

User Impact: Users may have been unable to successfully refresh Excel Online workbooks with on-premises data models.

Final status: We've determined that the fix is fully deployed to the affected environments and have confirmed that impact is resolved.

Scope of impact: Impact was specific to all users except those in Western Europe and the North Central United States that were attempting to refresh their online workbooks with on-premises data.

Start time: Thursday, May 17, 2018, at 12:02 AM UTC

End time: Saturday, May 26, 2018, at 9:55 AM UTC

Preliminary root cause: An update to Excel Online workbooks was causing connections to become non-responsive while refreshing workbooks.

Next steps:

- We're reviewing our update procedures to better identify similar issues during our development and testing cycles.

This is the final update for the event.

Section 11. Power BI Security

This section covers security aspects of Power BI. The focus of this section is not on duplicating the technical content which is already available online. Rather, the focus here will be on providing an overview of key functionality to be aware of and, more importantly, providing advice on how this functionality should be used. Following are the key pieces of documentation related to security that you should review carefully:

- Whitepaper on [Power BI Security](#)
- Documentation on [Power BI Security](#)
- Whitepaper titled "[Supporting Your EU GDPR Compliance Journey With Microsoft Power BI](#)"
- Security documentation available in the [Service Trust Portal](#)
- Whitepaper on [distributing Power BI content to external guest users via Azure Active Directory B2B](#)

The reports and dashboards in Power BI potentially contains 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 administrative tasks in Power BI.

Integration with Azure Active Directory

Power BI's security is built upon the foundation of having an [Azure Active Directory tenant](#). This is advantageous because it allows a common set of users, groups, and credentials to be used for securing content in Power BI, as well as Office 365, Dynamics 365, and other related Microsoft services.

Since Power BI uses Azure Active Directory (AAD) for authentication, it can use AAD's full range of functionality. [AAD Conditional Access policies](#) can be configured to enforce policies (including in the Power BI mobile app) such as:

- Only locations or specific IP address ranges inside the corporate network can connect.
- Only domain-joined machines are permitted to connect.
- Only machines which are compliant with network security policy are permitted to connect.
- Only users from specific security groups may connect.

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 enable multi-factor authentication (MFA) when signing in to Power BI, adding an extra layer of protection against stolen credentials. Currently there are some limitations with MFA integration, so be certain to validate and test with your use case.

Managing Access to the Power BI Service

Access to the Power BI Service requires each user to be assigned a Power BI Free or Power BI Pro license. As discussed earlier, all activity for users with a Power BI Free license is constrained to within My Workspace (the exception to this being when a free user is accessing content stored in a Premium capacity). A Pro license opens up additional features and functionality.

By default, any user is permitted to sign up for a Power BI Free license. The ability to disable automatic sign-ups, as well as techniques to automate and/or introduce workflow to the process, was discussed in the [Power BI Licensing](#) section of this whitepaper.

Azure B2B

Power BI supports Azure B2B (Business-to-Business). External users, for instance from a customer or partner company, can be invited as guest users for collaboration purposes in Power BI or other Azure and Office 365 services.

If the external user also has Azure Active Directory, their originating/owner AAD tenant stays in control of the identity and management of corporate credentials. This external guest account functionality permits the ability to work safely and securely without having to synchronize identities. Although there are some limitations, for the most part an external guest user is treated like an internal corporate user: you can provide them a Pro license if they do not have one, provide access to Premium resources, and so forth. Consumer accounts such as Outlook and Gmail can be utilized as well for sharing and distributing content.

There are two types of ways of inviting guest users to your tenant:

- **Planned invitations.** A guest account is set up ahead of time in AAD; this can be done with PowerShell for handling larger volumes of users.
- **Ad hoc invitations.** A guest account is auto-generated in AAD at the time that a Power BI user shares or distributes content to an external user who wasn't previously set up.

It is very important to ensure that the original e-mail address for the external user is utilized when setting up a guest account. If an alias is used, the B2B invitation will succeed – however, Power BI and other services will not be able to identify the alias properly.



For larger scale deployments, it is recommended to utilize “planned invitations” for your AAD guest account (B2B) users. Additionally, usage of security groups in relation to the guest accounts is also strongly encouraged.

Users added via external B2B do face some limitations. Guest users cannot create content in Power BI; they are limited to consuming content from an App, App Workspace, or Shared With Me. Also, the Power BI Mobile App is not available for use by guest users.

Also, keep in mind that B2B does involve the external sharing capability of Power BI. This capability can be disabled in the tenant-level settings in the Power BI Admin Center. Also, the Azure Active Directory administrator may implement various [policies for B2B invitations](#).

Managing Access to App Workspaces and Apps

At the most basic level, whether a user can see a given piece of information in an App Workspace is determined by whether he or she has access to an App Workspace or App that displays that data. Note that this is a distinctly different experience from the sharing feature in the Power BI Service.

As previously discussed in [Section 9](#), in most cases App Workspaces should be treated as places where groups of users can collaborate. In the original V1 implementation of workspaces (i.e., Office365-backed workspaces), App Workspace administrators and members could only be added by individual email address which really only worked well for small teams (or involved

automation-oriented workarounds). In the future the new workspace experience will be far more flexible, permitting assigning permissions to:

- Security groups
- Mail-enabled security groups
- Distribution lists
- Office 365 groups
- Individual users

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

1. **Entire organization.** It can be made available throughout the entire organization. This is only likely to be appropriate in very small organizations, or for data that is non-sensitive.
2. **Private - Individual users are granted access.** We do not recommend this approach unless an App is to be restricted to a very small number of people. Managing individual users can become tedious and error-prone, and can only be done by an administrator of the Power BI App Workspace.
3. **Private - Groups are granted access.** This is the best option for larger organizations for two reasons:
 - The same group or distribution list 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.
 - Access to the App can be controlled centrally by someone who is responsible for managing security, which makes administration within Power BI much easier.

Note that the three choices listed above are not mutually exclusive. It is certainly possible that a combination of individual users and groups can be specified together.

Securing Data in Transit

When data is actively being used or accessed, it is considered in transit. This includes activities such as interacting with a report, or refreshing a dataset. Power BI uses HTTPS, TCP/IP and TLS to ensure data is encrypted and maintains integrity during the transport.

While the data is held in-memory within the Power BI Service, it is unencrypted to allow for report display to occur. If data becomes cached, it is encrypted and stored in Azure SQL Database (some artifacts can also be stored in Azure REDIS Cache unencrypted for a limited period of time).

Within a client's web browser, data is cached only for the duration of the session. The cache is destroyed when the user navigates the browser tab out of the Power BI Service, or closes the browser window.

More details are covered in the [Power BI Security Whitepaper](#).

As discussed in [this post](#) (which mentions Azure SQL Database, but is applicable to other data services as well), the easiest way to allow connectivity is the “Allow access to Azure services” setting. However, doing this may open up connectivity more broadly than the corporate security policy permits. Instead, as the post describes, using a combination of virtual network rules and usage of the On-Premises Data Gateway is a better choice to secure access between the data in Azure and Power BI Desktop and/or the Power BI Service.

Securing the On-Premises Data Gateway

Data source credentials are encrypted using asymmetric encryption so that they cannot be decrypted in the cloud. The credentials are sent to the gateway machine, where they are decrypted when the data sources are accessed.

The On-Premises Data Gateway uses data compression, as well as transport encryption.

Data Transmitted Outside of the Power BI Service

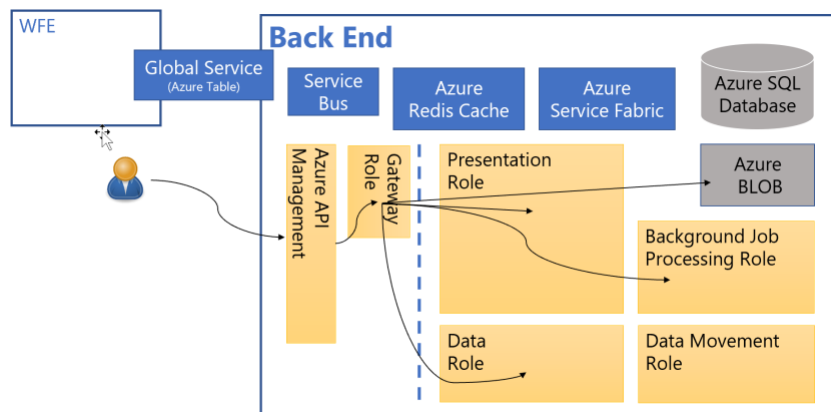
If map visuals are displayed on reports which utilize [Bing Maps](#) or [Esri](#), those services do transmit data out of the Power BI Service. Additionally, some custom visuals also may transmit data back to the originator of the custom visual.

Monitoring activities are transmitted for retention in the Office 365 audit log which is discussed in [Section 10](#).

Securing Data at Rest

There are several types of data at rest including, but not limited to:

1. The data stored in the dataset
2. Metadata for reports and dashboards
3. Visual caching of dashboard tiles
4. Credentials to connect to source data
5. Usage tracking data



Power BI uses two repositories for storing and managing data and metadata: Azure Blob Storage and Azure SQL Database.

Encryption keys for Azure Blob Storage are stored and encrypted in a separate location in the Power BI Service. The process is fully managed by an internal Microsoft service.

The encryption keys for Azure SQL Database are managed by Transparent Data Encryption (TDE).

Dataset 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 one of two ways:

1. In the most common scenario, a report developer creates multiple security roles, gives each role a filter to view a specific slice of data, and then assigns users or security groups to those roles.
2. 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.

As a best practice, it is recommended to use the DAX `UserPrincipalName()` function in row-level security expressions as it will yield the same results when testing results in Power BI Desktop and the Power BI Service.



It is important to keep in mind that row-level security that is set up within Power BI Desktop applies to only that one .pbix file. This increases the potential for inconsistencies, errors, and potentially maintenance issues across multiple .pbix files when a change occurs. If there are numerous Power BI datasets with row-level security, then specifying security at the data source level instead 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 Analysis Services live connections 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). Therefore, 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 currently not possible with Power BI Desktop. 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 permit connectivity to the database for users who connect from Power BI Desktop. This is handled by either opening firewall ports for the relevant IP address range for an office, or by including the Azure SQL Database in the corporate virtual network (i.e., Azure vNet).

Encrypted Data Storage for Imported and Push Datasets

For imported datasets, the data is securely stored and encrypted in Azure Blob Storage. Azure Blob Storage encryption is encrypted with a randomly generated 256-bit CEK (content encryption key) as well as a pre-defined 256-bit KEK (key encryption key).

For push datasets, the data is securely stored and encrypted in Azure SQL Database.

Encrypted Storage for Metadata and Visual Cache

The metadata related to a Power BI subscription (such as report/dashboard layouts, recent data sources, workspaces, tenant & organizational information) as well as visual caching of tiles and visuals (discussed in [Section 7](#)) are securely stored and encrypted in Azure Blob Storage and Azure SQL Database. For Analysis Services source data, there is also a metadata reference to Azure SQL Database.

Encrypted Storage of Credentials to Connect to Source Data

Data source credentials used by the Power BI Service are securely stored and encrypted in Azure SQL Database.

Limiting Functionality in the Power BI Service

Restricting Sharing and Publishing

As well as restricting access to App Workspaces and Apps, you will likely also want to specify who may publish Apps. While this isn't strictly a security feature, it does affect secure distribution of data.

Only users with Power BI Pro licenses can publish Apps, but unless Apps are published to Power BI Premium capacity users will also need Power BI Pro licenses to view shared content. Restricting the type of license assigned to a user is not a workable way of doing this. Instead, in the Tenant Settings area of the Power BI Admin Portal it is possible to apply the following restrictions:

- Prevent the ability to publish Power BI Apps to the entire organization. Notice that this does not prevent the creation of Apps completely, just the ability to publish them so that the whole organization can see them. In this case, specific groups will need to be specified for visibility to a published Power BI App.
- If publishing to the entire organization is enabled, then it is also possible to restrict the ability to specific security groups, or alternatively to exclude specific security groups from being able to publish to the entire organization.
- There is also a tenant-level setting to disable the ability to automatically push Power BI Apps to users. This setting does not affect the ability to publish Power BI Apps – rather it focuses on *automatic pushing* of those Apps. If a Power BI App has been *pushed* to a

user, the recipient will not have to take action to look for it within AppSource. The push capability can be a user convenience for the recipient; however, it can also be frustrating for users who begin to see things they did not set up so it should be used judiciously.



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. However, in certain circumstances it may be justifiable to restrict these capabilities in order to protect sensitive data.

Data Privacy and Classification

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

The Power BI Service has the ability to assign [data classification tags](#) to dashboards which indicates the sensitivity of the data being displayed by the dashboard. This could include data classifications such as Confidential, Highly Sensitive, Low Business Impact, Regulatory, or Public. It's also possible to utilize the classifications for subject matter areas. Utilization of data classification tags should align with data governance standards established for the organization.

To utilize data classifications, a Power BI Service Administrator needs to enable the feature in the Power BI Admin portal and create the tags. Additional documentation can be made available for each classification via a URL associated with the tag. A default must also be specified, which is utilized in the event a dashboard developer does not explicitly assign a tag.



Consider utilizing a data classification called "Default" which is specified as the default. This could reduce misunderstandings when a data classification is not expressly specified.

Once the tags are available for use, it is important to educate all Power BI dashboard developers why the classifications are important, who relies on them, and how to assign them correctly and consistently. Since data classification tags cannot be applied to reports, this feature is most useful when users navigate to reports via a dashboard. Only one data classification can be assigned to a Power BI dashboard.

Power BI Premium

Although Power BI Premium offers dedicated infrastructure in which your corporate data is more isolated, Power BI Premium is not to be thought of as a security feature.

Compliance Certifications

Major certifications such as ISO and HIPAA are available. The current reference for compliance certifications can be found in the [Microsoft Trust Center](#).

Securing Power BI Report Server

The functionality for collaboration, sharing and distribution of content in Power BI Report Server is similar, and is discussed in [Section 9](#).

The management of security in Power BI Report Server is slightly more granular, as it is configured for folders (preferably) or individual files/content. This was discussed earlier in this whitepaper.

Row-level security is not currently supported in Power BI Report Server. It is a future feature on the roadmap.

Managing Encryption Keys

On-Premises Data Gateway Recovery Key

As discussed in [Section 7](#), 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. This key cannot be changed or reset, so it is important to store it in a safe place where the Power BI Administration team can access it.

Encryption Keys for Credentials

For on-premises data sources, the encryption key does not leave the on-premises infrastructure. The encryption key for the Data Movement Service remains stored in the customer's On-Premises Data Gateway. Power BI cannot access or intercept the key value.

For cloud data sources, the encryption key is stored in Azure SQL Database and encrypted with Transparent Data Encryption (TDE).

Power BI Report Server Key

As discussed earlier in this whitepaper, one feature of Power BI Premium is access to (optionally) deploy to Power BI Report Server. This key can be located in the Power BI Admin Portal:

Admin portal

Usage metrics
Users
Audit logs
Tenant settings
Capacity settings
Embed Codes
Organization visuals

Power BI Premium Power BI Embedded

V-CORES
16 of 16 used 16 v-cores
0 available
[Learn more about capacity sizes](#)
Set up new capacity

PREMIUM CAPACITIES

CAPACITY NAME	CAPACITY ADMINS	ACTIONS	SKU	V-CORES	REGION	STATUS
Contoso Capacity	Adam Wilson, Ajay ... + 2 more		P1	8	North Central US	Active

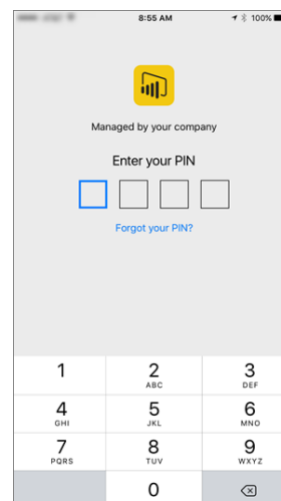
Power BI Report Server key

Mobile User Access Control

The Power BI Mobile Apps for iOS, Android, and Windows can be integrated with [Microsoft Intune](#) for mobile device management. Intune allows for functionality such as requiring the user to enter a PIN code before the app can be opened, and specifying a maximum number of attempts.

Encryption of data on the mobile device and sharing from the Power BI Mobile App to other apps on the phone can be specified through Intune as well.

Intune can also be used in conjunction with [Mobile Device Management \(MDM\) for Office 365](#), which offers additional options for access control.



Power BI in Sovereign Clouds

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](#).

Currently, Power BI is available in 3 national clouds: [US Government, China, and Germany](#). The data location, oversight, and management of those data centers are in compliance with local government regulations.

There are some feature differences between the commercial cloud and each national cloud. The US Government is [discussed in this post](#).

Private Connectivity with ExpressRoute

Power BI supports the use of [ExpressRoute](#), which allows you to create a private network connection from your organization to Power BI (or using an internet service provider's colocation facility). ExpressRoute bypasses the public internet, providing for more secure and performant connectivity.

Section 12. Power BI Limits + Feature Comparisons

Following is a reference list of limits and feature comparison tables applicable as of the time of publishing this whitepaper. Please be diligent in verifying this information at a later date.

Individual File Size Limit (Per .pbix File)

Refer to the following for Premium dataset sizes: <https://docs.microsoft.com/en-us/power-bi/service-premium-large-datasets>

Applicable To	File Size Limit	Comments
Power BI Desktop	None	Limited based on specifications of the user machine it's running on
Power BI Desktop file uploaded to a standard workspace in the Power BI Service	1 GB	
Power BI Desktop file uploaded to a premium workspace in the Power BI Service: P1 SKU	3 GB	
Power BI Desktop file uploaded to a premium workspace in the Power BI Service: P2 SKU	6 GB	
Power BI Desktop file uploaded to a premium workspace in the Power BI Service: P3 SKU	10 GB	
Power BI Desktop file uploaded to Power BI Report Server	2 GB	Requires a configuration change to increase max size from 1GB to 2GB
Excel file uploaded to the Power BI Service	250 MB	
Excel file uploaded to Power BI Report Server	2 GB	Requires a configuration change to increase max size from 1GB to 2GB

Power BI Workspace Object Limits

Applicable To	Limit	Comments
Workspace size (non-Premium) V1 workspaces	1 GB Free 10 GB Pro	Total of all files in My Workspace or an App Workspace if using the standard Power BI Service offering
Workspace size (non-Premium) New workspace experience	10 GB	Total of all files in My Workspace or an App Workspace if using the standard Power BI Service offering
Workspace size (Premium)	100TB across the entire capacity	
Reports per dataset	1,000	Multi-page reports still count as 1
Datasets per workspace	1,000	
Reports per workspace	1,000,000	1,000 datasets per workspace x 1,000 reports per dataset
Data alerts per user	250	

Scheduled Data Refreshes Per Day

Applicable To	Refreshes Per Day Per Dataset	Comments
Power BI Service - standard workspace	8 per day	
Power BI Service - premium workspace	48 per day	
Power BI Report Server	Unlimited	Utilizes SQL Server Agent

V-Cores Per Premium Capacity

For additional dedicated capacity details including backend vs. frontend cores, memory, DirectQuery and live connection query limits per second, and page renders per hour, refer to the following documentation:

- [Power BI Premium](#)
- [Power BI Embedded](#)

Applicable To	Total v-cores
Power BI Premium EM1 or Azure Power BI Embedded A1	1 v-core
Power BI Premium EM2 or Azure Power BI Embedded A2	2 v-cores
Power BI Premium EM3 or Azure Power BI Embedded A3	4 v-cores
Power BI Premium P1 or Azure Power BI Embedded A4	8 v-cores
Power BI Premium P2 or Azure Power BI Embedded A5	16 v-cores
Power BI Premium P3 or Azure Power BI Embedded A6	32 v-cores

Concurrent Data Refreshes Per Premium Capacity

Applicable To	Concurrent Refresh Operations per Capacity
Power BI Premium EM1 or Azure Power BI Embedded A1	1
Power BI Premium EM2 or Azure Power BI Embedded A2	2
Power BI Premium EM3 or Azure Power BI Embedded A3	3
Power BI Premium P1 or Azure Power BI Embedded A4	6
Power BI Premium P2 or Azure Power BI Embedded A5	12
Power BI Premium P3 or Azure Power BI Embedded A6	24
Power BI Premium P4	48
Power BI Premium P5	96

Streaming Dataset Limits

Applicable To	Limit	Comments
Data retention period	1 hour	Data is temporarily stored for 1 hour to render visuals
Max rate of data ingestion	5 requests/second; 15 KB/request	

Push Dataset Limits

Applicable To	Limit
Max rate of data ingestion	1 request/second; 16 MB/request
Data throughput	1 million rows per hour

REST API Limits

Applicable To	Limit
POST Rows	75 max columns 75 max tables 10,000 max rows per single POST rows request 1,000,000 rows added per hour per dataset 5 max pending POST rows requests per dataset 120 POST rows requests per minute per dataset Table with $\geq 250,000$ rows: 120 POST rows requests per hour per dataset 200,000 max rows stored per table in FIFO dataset 5,000,000 max rows stored per table in 'none retention policy' dataset 4,000 characters per value for string column in POST rows operation

Comparison of Power BI Free vs. Power BI Pro

The following is a summary of the key feature differences per tier when Power BI Premium is *not* in use. Note this is not intended to be an exhaustive list of all features.

When referring to this chart, be aware that Power BI Free is only suitable for extremely small organizations. Unless used in conjunction with Power BI Premium, the Free licenses are not suitable to an enterprise deployment (see the [Power BI Premium](#) discussion in Section 3 for more details). For this particular comparison, all features in the Power BI Service permitted for a Power BI Free license are limited to within the user's 'My Workspace' area.

Feature/Capability	Power BI Free	Power BI Pro
<i>Power BI Desktop:</i>		
Create datasets and reports in Power BI Desktop	✓	✓
Access to all data sources (Get Data)	✓	✓
<i>Publishing Content:</i>		
Publish datasets and reports to My Workspace in the Power BI Service	✓	✓
Publish datasets and reports to App Workspace in the Power BI Service		✓
Publish datasets and reports to Power BI Report Server		✓
Publish datasets and reports to Power BI Service via an API		✓
Publish to Web (public report distribution)	✓	✓
<i>Power BI Service:</i>		
Create dashboards	In My Workspace	✓
Save favorites	In My Workspace	✓
Download reports	In My Workspace	✓
Save as another report	In My Workspace	✓
Persistent report filters	In My Workspace	✓

Feature/Capability	Power BI Free	Power BI Pro
Data alerts	In My Workspace	✓
E-mail subscriptions ("subscribe")		✓
Export data	In My Workspace	✓
Export to PowerPoint	In My Workspace	✓
Print current page	In My Workspace	✓
Q&A natural language queries	In My Workspace	✓
Quick Insights	In My Workspace	✓
Dashboard performance inspector	In My Workspace	✓
Maximum size of an imported dataset (non-Premium workspace)	1GB per file	1GB per file
Row-level security (defined in Power BI Desktop)		✓
Maximum size of an imported dataset (Premium workspace)		10GB per file
Maximum data storage quota (non-Premium workspace)	10GB per user	10GB per user
Maximum data storage quota (Premium workspace)		100GB per capacity

Data Connectivity:

DirectQuery and Analysis Services live connection via On-Premises Data Gateway	In My Workspace	✓
Use of streaming datasets	In My Workspace	✓
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)		✓

Data Refresh in the Power BI Service:

Feature/Capability	Power BI Free	Power BI Pro
Scheduled data refresh via On-Premises Data Gateway (non-Premium)	Up to 8x/day in My Workspace	Up to 8x/day
Scheduled data refresh via On-Premises Data Gateway (Premium)		Up to 48x/day

Sharing and Distribution:

Create and publish Apps		✓
Access Apps for which user is included in the access list		✓
Share reports and dashboards with individual colleagues (peer-to-peer small-scale sharing)		✓
Share reports and dashboards with external users		✓
View reports and dashboards shared by another colleague		✓
Be a workspace administrator or member		✓

APIs and Embedding:

Publish content via APIs		✓
Embed in SharePoint Online or Teams		✓
Embed in custom application		✓
Embed in a public website via Publish to Web	✓	✓

Other:

Free support		✓
Custom visuals from Office Store	✓	✓
Power BI Mobile Apps	✓	✓
Cortana integration for report rendering	✓	✓
Third party SaaS Apps (from AppSource)	Desktop or My Workspace	✓

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

The 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.

See also [this comparison](#) between Power BI Report Server and the Power BI Service.

Feature/Capability	Power BI Service	Power BI Report Server	SQL Server 2017 Reporting Services
<i>Publish and View Reports:</i>			
Publish and view Reporting Services paginated reports (RDL)		✓	✓
Publish and view Power BI reports (.pbix)	✓	✓	
Publish and view Excel workbooks (.xlsx)	✓	✓	
Rendering of custom visuals utilized in a Power BI Report	✓	✓	
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 .pbix file (non-Premium)	1GB	2GB	
Maximum size of an individual imported .pbix file (Premium workspace)	10GB		

Feature/Capability	Power BI Service	Power BI Report Server	SQL Server 2017 Reporting Services
Maximum size of an individual Excel file	250MB	250MB	
Reuse of imported Power BI datasets (Analyze in Excel)	✓		
Reuse of shared datasets (Reporting Services paginated reports only)		✓	✓
Streaming datasets	✓		

Additional Capabilities:

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 web browser	✓		
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)	✓		
R integration	✓		
Quick Insights (machine learning)	✓		
Release cadence	Monthly	3x/year	Approx. annually
Ability to opt into preview features	✓		

Comparison of Power BI Premium vs. Power BI Embedded

When it comes to embedding content and planning for capacity, there are some similarities and some differences between Power BI Embedded and Power BI Premium. This section attempts to offer some clarity and a high level reference.

The Power BI REST APIs can be freely used for embedding reports into custom applications. The purchase of Power BI Premium or Power BI Embedded is not a prerequisite for embedding (in which case, all users of the application require a Pro license). Power BI Premium and Power BI Embedded are typically mentioned in conjunction with embedding scenarios because the ability to utilize Power BI Free licenses for every user of the application is typically a requirement. Following is a reference of the similarities and comparisons between Power BI Premium (P and EM SKUs) and Power BI Embedded (A SKUs):

Power BI Premium		Power BI Embedded
P SKUs	EM SKUs	A SKUs
Organizational analytics		Embedded analytics

Use Cases and Functionality:

Distribution of content	Organizational reporting via Power BI Service or organizational embedding	Organizational embedding: Analytics embedded in portals or line of business applications that are intended for business users	External embedding: Analytics embedded in a packaged application, service, or portal for users outside your organization
Offering is primarily targeted to (but not limited to)	Enterprises		Software vendors (ISVs)
Primary use case	Enterprises who want to take advantage of dedicated capacity in the Power BI Service and (optionally) embed visuals into custom applications	Embedding of visuals into custom applications	
Offering includes	Use of all Power BI Premium functionality in the Power BI Service + use of Power BI Report Server + use of Power BI REST APIs for embedding	Use of the Power BI REST APIs for embedding	
Key benefits in the Power BI Service	Enterprise distribution to free read-only users, larger data models, incremental data refresh, dedicated/isolated hardware	N/A	N/A

	Power BI Premium		Power BI Embedded
	P SKUs	EM SKUs	A SKUs
	Organizational analytics		Embedded analytics
Embedding referred to as	Embed for your organization		Embed for your customers
Type of service offering	Software as a Service (SaaS)		Platform as a Service (PaaS)
Ability to embed in custom application for internal/organizational users	Yes		Yes <i>(as a secondary use case)</i>
Ability to embed in custom application for external users such as customers, suppliers, partners	Yes <i>(external users could be invited as guest users)</i>		Yes
Ability to embed reports in Teams	Yes		No
Ability to embed reports in SharePoint Online	Yes		No
Ability to embed visuals in SharePoint (on-premises)	Not currently a pre-built web part, but could be done using a generic application hosted in SharePoint		No
Ability to distribute content from a Premium-backed workspace in the Power BI Service	Yes	No	No
Ability to utilize Power BI Report Server	Yes	No	No

User Authorization and Security:

User authentication requirement	Individual user identity is required	One master service account
Authentication referred to as	"User owns data"	"App owns data"
User authentication managed by	Power BI	Custom application
Row-level security available	Yes	Yes
Free user licenses may be used for read-only users	Yes	N/A <i>(Custom app manages all users)</i>
Pro user licenses required for	Authors & editors, developer using REST APIs, tenant admin, capacity admin, capacity assignment users	Master account, developer using REST APIs, tenant admin, capacity admin

	Power BI Premium		Power BI Embedded
	P SKUs	EM SKUs	A SKUs
	Organizational analytics		Embedded analytics
Consumer of content can log into Power BI Service	Yes	N/A (no access to the Power BI Service)	N/A (no access to the Power BI Service)

Billing, User Authorization, and Security:

Where to purchase	Office 365		Azure
Billing interval	Monthly (with pre-defined commitments)		Hourly (no commitment)
Ability to pause compute service + billing	No		Yes
Elasticity; ability to scale up/down to handle workload	Yes P1 - P5 SKUs <i>Not intended to be scaled up/down frequently</i>	Yes EM1 - EM3 SKUs <i>Not intended to be scaled up/down frequently</i>	Yes A1 - A6 SKUs <i>Handling variable-level workloads via scalability is a major feature</i>
Can be used in conjunction with Azure Active Directory B2B or B2C	Yes		No
Purchase of capacity is required to use Power BI REST APIs for embedding	Yes		Required for moving to Production (a small # of API calls can be used for free during Dev/Test)

Section 13. Power BI Deprecated Items

Following are the features and functionality which are deprecated, or on a deprecation path, at the time of this writing.

Deprecated: A feature which has been announced as discontinued and will be fully removed from the product. Customers should be actively planning a migration or discontinuing use of any deprecated features. Since Power BI is a cloud service, the transition time for permanently removing deprecated features is shorter than it has historically been for on-premises Microsoft products.

Deprecation path: A feature which will be discontinued at a future date. New development is strongly discouraged.

Organizational Content Packs

Distributing content to other users via organizational content packs is on a deprecation path. [Power BI Apps](#) are the replacement functionality, and should be utilized going forward. Note that third party apps (formerly referred to as third party content packs) from AppSource are still fully supported.

V1 Workspaces in the Power BI Service

With the introduction of the new workspace experience, discussed earlier in this whitepaper, V1 workspaces (i.e., Office 365 group-backed workspaces) are now on a deprecation path. There will be a migration process from V1 to the new workspace experience.

Get Data Experience in the Power BI Service

Connecting to databases or systems, via Get Data, in the Power BI Service is deprecated. The capabilities to connect to data, transform and clean data, as well as creation of a dataset, relationships, and calculations are all fully supported in Power BI Desktop.

Azure Workspace Collections

Azure Workspace Collections are deprecated, and should be [migrated](#) to the replacement service, [Power BI Embedded](#). Beginning [June 2018](#), customers will actively begin to be blocked from utilizing Power BI Workspace Collections. Customers with an Enterprise Agreement signed before June 2017 will have until the end of that agreement to migrate to Power BI Embedded.

Power BI CLI

The Power BI command line interface (CLI), for managing workspace collections, is deprecated and no longer supported. Automation capabilities for Power BI can be invoked via [PowerShell cmdlets](#) or [REST APIs](#).

Section 14. 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. The support 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.

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

Power BI Embedded: <https://azure.microsoft.com/en-us/status/>

Reporting Issues

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. Before you post, please search the forum to check whether someone else has already posted the same issue. If you have a new issue, 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.

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

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. 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.

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

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 that have been picked up by the product team (such as started, or under review). As always, please 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 - the more votes an issue receives the more likely that it will be implemented.

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

Finding a Consultant or Partner to Accelerate Your Power BI Roll-Out

Sometimes, building or extending an analytics solution requires bringing onboard a professional with deep knowledge in Power BI to architect the solution, setup the infrastructure, implement the first set of reports and dashboards and do a knowledge transfer to your team. Microsoft has a rich ecosystem of [partners](#) to build [packaged industry solutions](#) that can be customized and will result in reducing the time/cost required to build analytics solution by up to 50%-70%. High quality [consulting service offerings](#) that have clear deliverables and predictable prices make it easier to start quickly.

Power BI Partner page: <https://powerbi.microsoft.com/en-us/partners/>

Learning More About Power BI

Power BI Team Blog and Videos

Power BI is a very powerful product that continually evolves and improves, 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. The .pbix files used in the monthly release videos are now posted on GitHub.

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

GitHub: <https://github.com/Microsoft/powerbi-desktop-samples>

Guy In a Cube YouTube Channel

The Guy In a Cube YouTube channel is another excellent resource available for you to learn about Power BI:

Guy In a Cube: <https://www.youtube.com/channel/UCFp1vaKzpfvoGai0vE5VJ0w>

Power BI Documentation

The official Power BI documentation site is very detailed and updated on a regular basis, and should be your primary reference when you need to learn about a feature of Power BI.

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

What's New in the Power BI Service: <https://docs.microsoft.com/en-us/power-bi/service-whats-new>

Power BI Community Site

The Power BI Community Site includes many contributions including a community knowledgebase, forums for Q&A, announcements, and several galleries such as the Themes Gallery, Data Stories Gallery, R Scripts Gallery, Quick Measures Gallery, among others.

Power BI Community: <http://community.powerbi.com/>

Power BI User Groups

Power BI user groups have been established in many places around the world. User groups are a great place to hear presentations by Power BI experts, talk to other Power BI users, and potentially even recruit new staff. Conference dates and locations, such as the Power BI World Tour, are also announced through this site.

User Groups: <https://www.pbusergroup.com>

Guided Learning

For introductory training on Power BI, the videos in the Guided Learning area of the Power BI site are useful place to start. Videos and written instructions are available in an easy-to-learn format.

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

Power BI Community Bloggers

The Power BI blog roll occasionally aggregates posts from many bloggers in the Power BI community. There's a wealth of knowledge available an enthusiastic community of bloggers. As with all online resources, pay attention to the date published as information could be out of date due to the pace of change with Power BI.

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

Power BI Roadmap

Details of the Power BI roadmap and planned features can be found in the Power BI section of the Business Applications Group's release notes site.

Roadmap: <https://docs.microsoft.com/en-us/business-applications-release-notes/april18/power-bi/overview>

Community Tools

Following are some of the tools contributed by the vibrant Power BI user community. These are independent tools not supported by Microsoft, some of which are free and others are paid products.

DAX Studio

DAX Studio is a client tool for executing and troubleshooting DAX queries.

Contributed and supported by: [SQLBI](http://daxstudio.org/)
<http://daxstudio.org/>

DAX Formatter

The DAX Formatter tool improves the readability of DAX.

Contributed and supported by: [SQLBI](http://www.daxformatter.com/)
<http://www.daxformatter.com/>

Lingo

Lingo is a web-based code editor for editing the linguistic schema in Power BI, for the purpose of improving how the natural language Q&A functionality.

Contributed and supported by: [Power BI Tips](https://powerbi.tips/2018/04/introducing-lingo/)
<https://powerbi.tips/2018/04/introducing-lingo/>

Report Theme Generator

The Report Theme Generator is a web-based tool which allows a user to specify properties and download into a JSON file to be used in a Power BI theme.

Contributed and supported by: [Power BI Tips](https://powerbi.tips/tools/report-theme-generator-v3/)
<https://powerbi.tips/tools/report-theme-generator-v3/>

Power BI Helper

The Power BI Helper aids in removing unused data elements from a model, finding dependencies, and in documenting a model.

Contributed and supported by: [RADACAD](http://radacad.com/power-bi-helper)
<http://radacad.com/power-bi-helper>

Power BI Documenter

Power BI Documenter is a tool which auto-generates documentation from Power BI Desktop, for the purpose of identifying and documenting data usage, visuals, and development practices.

Contributed and supported by: [Data Vizioner](http://www.datavizioner.com/power-bi-documenter/how-to-use-power-bi-documenter/)
<http://www.datavizioner.com/power-bi-documenter/how-to-use-power-bi-documenter/>

Power Update

Power Update is a tool which moves scheduled refresh operations for datasets in Power BI Desktop or Excel to a local machine.

Contributed and supported by: [Power On BI](#)

<http://poweronbi.com/power-update/>

Power BI Visual Planning

The Visual Planning tool allows for users to edit data directly in a report or dashboard.

Contributed and supported by: [Power On BI](#)

<http://poweronbi.com/powerbi-visual-planning/>

Power Pivot Utilities

Power Pivot Utilities is an add-in for Excel. It aids with documenting models, relationships, calculated columns, unused columns, and memory usage via a set of VBA macros.

Contributed and supported by: Bertrand d'Arbonneau

<https://www.sqlbi.com/tools/power-pivot-utilities/>

Turbo.net Power BI Desktop Application

The Turbo.net tools allow applications to run on any desktop. Their version allows Power BI Desktop to run on an Apple Mac.

Contributed and supported by: [Turbo.net](#)

<https://turbo.net/run/powerbi/powerbi>

Additional Resources and Links

Power BI Premium

Microsoft Power BI Premium Whitepaper

This whitepaper, which was published October 2017, describes Power BI Premium.

<https://aka.ms/pbipremiumwhitepaper>

Power BI Premium Planning and Deployment Whitepaper

This whitepaper provides guidance and best practices for planning and deploying Premium capacity for well-defined workloads.

<https://aka.ms/Premium-Capacity-Planning-Deployment>

Power BI Premium Calculator

The online calculator is helpful for estimating pricing for dedicated Premium capacity in Power BI.

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

Power BI Embedded	<p>Plan Capacity for Embedded Analytics Whitepaper</p> <p>This whitepaper provides guidance for application developers and ISVs (independent software vendors) for how to determine capacity for Power BI Embedded.</p> <p>https://aka.ms/pbiewhitepaper</p>
Power BI Report Server	<p>Support Timeline and Version History</p> <p>This page describes the timeline for support for each release of Power BI Report Server, as well as the history of releases.</p> <p>https://powerbi.microsoft.com/en-us/documentation/reportserver-support-timeline/</p>
	<p>Capacity Planning Guidance</p> <p>This whitepaper offers guidance on capacity planning for Power BI Report Server by sharing results of numerous load test executions of various workloads.</p> <p>https://docs.microsoft.com/en-us/power-bi/report-server/capacity-planning</p>
Security and Privacy	<p>Power BI Security Whitepaper</p> <p>The Power BI Security Whitepaper was updated December 2017.</p> <p>https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-power-bi-security/</p>
	<p>Power BI and GDPR</p> <p>This whitepaper discusses GDPR (General Data Protection Regulation) compliance, data protection, privacy, and how it relates to Power BI.</p> <p>https://aka.ms/power-bi-gdpr-whitepaper</p>
	<p>Microsoft Trust Center</p> <p>The Trust Center publishes the status for compliance, certifications, and attestations. Security-related information is available as well.</p> <p>https://www.microsoft.com/en-us/trustcenter/cloudservices/powerbi</p>
	<p>Microsoft Enterprise Privacy Statement</p> <p>Includes the privacy terms for Microsoft Online Services, including Power BI. Data collection, and how personal data is used, is described here.</p> <p>https://www.microsoft.com/EN-US/privacystatement/OnlineServices/Default.aspx</p>
Business-to-Business Distribution	<p>Distribute Power BI Content to External Guest Users Whitepaper</p> <p>This whitepaper outlines how to distribute content to users outside the organization using Azure Active Directory Business-to-business (AAD B2B) capabilities.</p> <p>https://aka.ms/powerbi-b2b-whitepaper</p>

Governance and Best Practices	<p>Power BI Governance and Deployment Approaches Whitepaper</p> <p>This whitepaper discusses governance and ways to deploy Power BI. It was published March 2016, but the ideas and processes discussed are still relevant.</p> <p>https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-governance/</p>
	<p>Best Design Practices for Reports and Visuals</p> <p>This page provides a wealth of information about design principles for effective Power BI data visualizations.</p> <p>https://docs.microsoft.com/en-us/power-bi/power-bi-visualization-best-practices</p>
	<p>Bidirectional Filtering in Power BI</p> <p>This whitepaper discusses the impact of using bidirectional cross-filtering in Power BI Desktop (and Analysis Services), including behaviors a data modeler/author needs to be aware of in order to use the capability without unintended side effects.</p> <p>https://docs.microsoft.com/en-us/power-bi/desktop-bidirectional-filtering</p>
Other	<p>Advanced Analytics with Power BI</p> <p>This whitepaper discusses advanced analytics capabilities of Power BI, including predictive analytics, data visualizations, R integration, and DAX (Data Analysis eXpressions).</p> <p>https://info.microsoft.com/advanced-analytics-with-power-bi.html</p>
	<p>Power BI and SAP BW</p> <p>This document describes how SAP customers can benefit from connecting Power BI to existing SAP Business Warehouse (BW) systems.</p> <p>https://aka.ms/powerbiandsapbw</p>
	<p>Microsoft Online Services Terms</p> <p>This page documents the terms which govern Microsoft Online Services, including Power BI.</p> <p>http://www.microsoftvolumelicensing.com/DocumentSearch.aspx?Mode=3&DocumentTypeId=31</p>
	<p>Sovereign Clouds</p> <p>This page lists Power BI Service availability for sovereign clouds which comply with local regulations.</p> <p>https://powerbi.microsoft.com/en-us/clouds/</p>