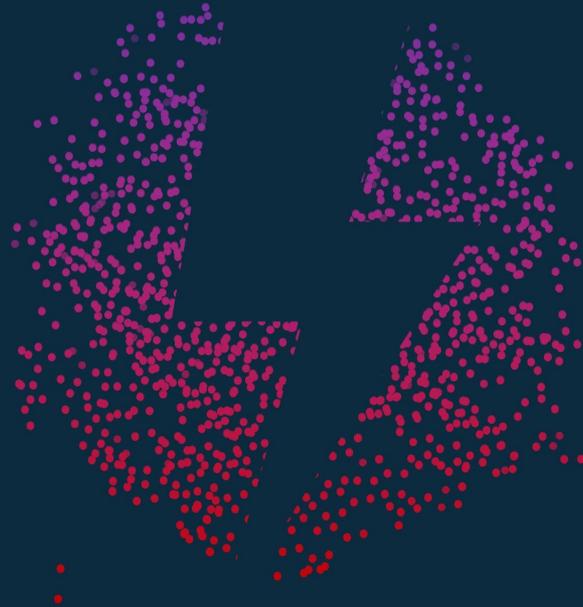




## **Data Governance in Power BI:**

An overview for application owners,  
infosec & data governance teams.



# Introduction

Self-service analytics empowers business users to create and share their own reports, without having to overly rely on data engineers. Because they help democratize data and insights, self-service BI tools play an important part in many organizations' data strategy, with Power BI as a market leader.

However, as organizations are moving from a centralized function to the federated model of self-service analytics, they're faced with a new set of risks. Where a central team used to build and share the data models and reports, this task can now be performed by different teams across the organization which makes it much harder to govern access to data and reports. Because self-service analytics is a relatively new concept, there is a lack of access management tools, security best practices and governance frameworks, which leaves significant leeway in how users manage access controls, resulting in inconsistent security across your data consumption plane. This exposes the organizations to privacy & security risks to their sensitive and business critical data, which is increasingly becoming a competitive disadvantage.

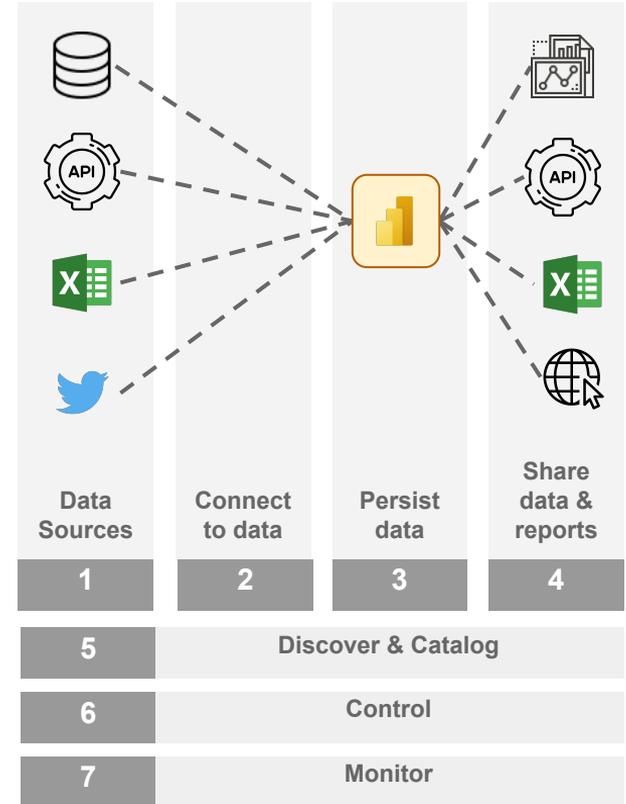
Conceptually, PowerBI offers various ways of accessing, storing and sharing data and reports, which complicates access management further, given the many different ways data and reports can flow as a result of this.

Fortunately, PowerBI also offers various ways to monitor and control access to the data and reports. This report serves as an overview for the Power BI application owner, InfoSec and Data Governance teams of the data governance capabilities in Power BI.



# Overview document

1	<b>Data Sources:</b> What are the possible data sources Power BI can connect to? (slide 5)
2	<b>Connect to data:</b> What are the different ways Power BI can connect to data? (slide 7)
3	<b>Persist data:</b> How is data modeled and persisted in Power BI? (slide 21)
4	<b>Share data &amp; reports:</b> What are the different ways Power BI can share data & reports? (slide 24)
5	<b>Discover &amp; Catalog:</b> How can you discover data and how is it cataloged in Power BI? (slide 38)
6	<b>Control:</b> How does Power BI control access to data and reports? (slide 41)
7	<b>Monitor:</b> How does Power BI monitor access to data and reports? (slide 53)



# Volta - Protect your data without slowing it down

Let's face it. You need to invest in the Cloud to stay competitive, but having all that sensitive data in the Cloud is keeping you and your coworkers awake at night. It is unclear whether your data is sufficiently protected, and the business is always kept waiting to get access to the data. The growing adoption of self-service complicates matters further as this leads to a shift from a centralised to a federated data management model where each team can use its own set of tools to create data sets and reports. As a result, it becomes close to impossible to keep track of the data, and control how it's being used.

Volta lets you scale cloud adoption and self-service analytics by simplifying access management, without sacrificing performance. Volta achieves this through:



**01**

A single pane of glass of your access controls across all your data sources and BI tools for informed access management



**02**

Giving the business ownership of the access controls, and enabling collaboration with data engineers to scale access management.



**03**

Promoting fine-grained access controls to comply with demanding privacy & security requirements.



**04**

Enabling data-centric access controls that are consistently applied across all your data sources and BI tools.



**05**

Making access management more resilient to change through tag-based access controls that can adapt to that change.



[www.volta-data.com](http://www.volta-data.com)



[info@volta-data.com](mailto:info@volta-data.com)



# Data Sources



# You can query data from different data source categories

- **File data sources:** Excel, flat files, XML, JSON, PDF, Sharepoint
- **Database data sources:** SQL Server, Oracle, MySQL, IBM Db2, IBM Informix, SAP HANA, Teradata, Snowflake, MariaDB, Google BigQuery and Amazon Redshift
- **Power Platform data sources:** Power BI datasets, workflows, Microsoft Dataverse
- **Azure data sources:** Azure SQL Database, Azure Analysis Services database, Azure Data Explorer, Azure HDInsight (HDFS), Azure Databricks and Azure Blob Storage
- **Online Services data sources:** Salesforce, GitHub, LinkedIn Sales Navigator, Dynamics 365, Microsoft Exchange Online, Emigo Data Source, Smartsheet, Google Analytics and Adobe Analytics
- **Other data sources:** Web, ODBC, OLE DB, Active Directory, SharePoint, Python, R scripts, Hadoop File (HDFS) TIBCO



# Connect to Data



# Connect to Data

Power BI has multiple ways of querying data. We start by explaining Power Query, DAX, and Data Flows which are the 3 building blocks for extracting, transforming and loading data. Then we discuss the different ways in which Power BI can query data, which are:

- Import
- Direct Query
- Composite
- Live Connection
- Streaming Data
- Push Data
- PubNub
- Excel
- CSV



# Power Query

## What?

Power Query is a data extraction and transformation tool that enables end users to import and reshape data from into a wide range of Microsoft products, including Excel, Power BI, Analysis Services, Dataverse, and more. Users use its UI to create the ETL-scripts in M code, which is the underlying language for the transformations. When used in Power BI, the Power Queries are part of the Power BI report. Power Query is used by Import and Direct Query, which are discussed below. In the context of PowerBI, Power Query is mainly used to filter and transform data. Depending on the data source these transformations are done at the data source, or by the Power Query engine.

## Where is data stored?

Power Query in itself does not store data. However, depending on the connection type (Import, DirectQuery), data can be stored. This is discussed below.

## Data Sources

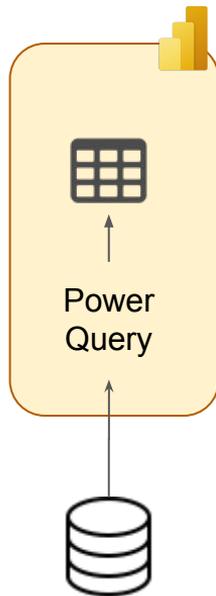
All databases, Excel, csv, Data Flows, Web, and Social Media.

## Advantages

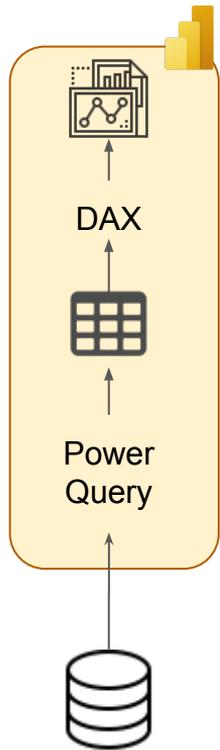
- You can combine data sets from multiple data sources
- You can perform powerful transformations
- Intuitive

## Disadvantages

- Difficult to govern as data and transformations are not centrally managed
- Dynamic parameters introduce security risks
- Difficult to share as the code is part of the Power BI file (pbix)



# Data Analysis Expressions (DAX)



## What?

Where Power Query is used to pre-process the data *before* it is loaded, DAX is used to analyse the data in-depth *after* the data is loaded into the Data View Model. DAX is a formula expression language used in Analysis Services, Power BI, and Power Pivot in Excel. In Power BI, it can be used to create:

- Relations and Hierarchies
- Measures: Formulas such as Sum and Count
- Calculated Columns: New columns based on other columns
- Calculated Tables: New tables based on other tables
- Row-Level Security: True/False condition to filter a table and its related tables (see below)

## Where is data stored?

DAX does not store data

## Data Sources

DAX does not connect to the data

## Advantages

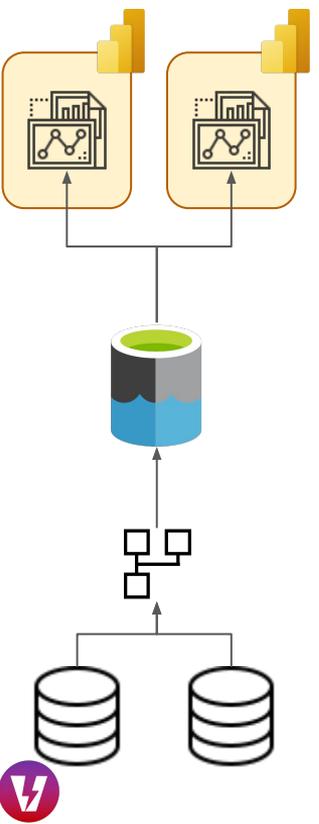
- Intuitive

## Disadvantages

- You cannot edit DAX with Power BI Service
- DAX can slow down the report significantly
- RLS in DAX is hard to govern because they are decentralised



# Data Flows



## What?

Dataflow is a Power Query process that runs in the cloud independently from any Power BI report and which stores the data in the Azure Data Lake storage. Report Builders can query Data Flows or Azure Data Lake from within Power BI using Import, or Direct Query. Other Data Flows can also query a Data Flow.

## Where is data stored?

Data is stored as csv-files in Azure Data Lake Storage Gen2 (tenant-assigned or your own ADLS Gen 2 store). Meta-data is stored as JSON-files.

## Data Sources

Databases, Files, Power Platform, Data Flows, Web, and Social Media

## Advantages

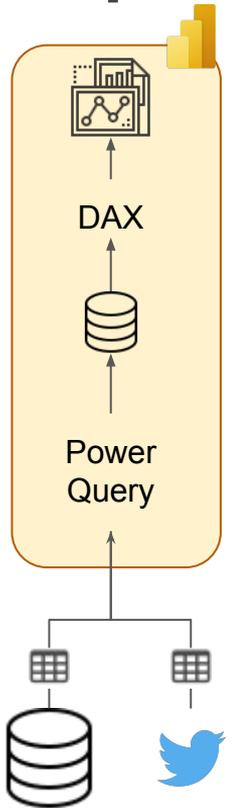
- ETL can be decoupled from report building so they can be shared
- Dataflows act as a single point of truth which simplifies governance
- You can combine data from disparate sources
- Data is stored in a data model defined by the user
- Data is refreshed on a recurring basis
- Users can pick the entities from the data model that they want to query

## Disadvantages

- Dataflows can only be created in shared workspaces in Power BI Service
- You cannot liveconnect to a dataflow
- Storage is not optimized for compression
- Output is not a relational database



# Import



**What?**

Import leverages Power Query to extract, transform and load the data into Power BI. With Import, you store the whole dataset and its preparations into memory. As a result, the reports are very responsive, but also require scheduled refreshes to be up-to-date. Import is typically the recommended connection type.

**Where is data stored?**

Power BI Desktop => Local Machine (Analysis Server)  
 Power BI Service => Azure Blob

**Data Sources**

All databases, Excel, csv, Data Flows, Web, and Social Media

**Advantages**

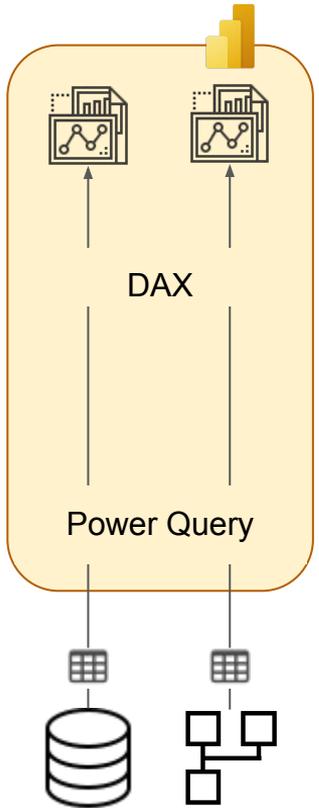
- You can combine data sets from multiple data sources
- You have all PowerQuery and DAX-functionality
- Responsive reports

**Disadvantages**

- Difficult to govern as data and transformations are not centrally managed
- Small data sets
- Max 1 refresh/hour and max 8 refreshes/day
- Data Modeler and Report Author has to be the same person
- Reidentification risk from combining data sets
- No user authentication at the data source



# Direct Query



## What?

Direct Query uses Power Query to extract, do limited transformations, and visualise the data in Power BI. Contrarily to Import, Direct Query only copies the schema of the data and, does not take a copy of the actual data. Therefore, modeling capabilities are limited, and it is recommended to do all modeling has at the source.

## Where is data stored?

Data is not stored. However, the data is cached.

## Data Sources

- Databases
- Data Flows
- Azure Analysis Services (in preview)

## Advantages

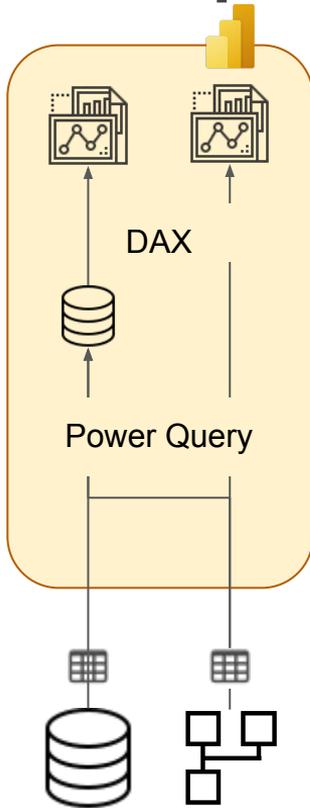
- No size limitations
- Less data proliferation => easier to govern
- Data Modeler and Report Author can be different persons
- Fresh reports
- Report user can be authenticated at the data source

## Disadvantages

- Only connects to Databases and Data Flows
- Limited Power Query, and DAX functionality
- Slower reports. Performance tuning at data source is a must.



# Composite



## What?

Report creators can use the Composite connection type to combine multiple data sources in Direct Query, or combine Direct Query with Import. You can also create relationships between, write measures across, and create visualisations using columns from different sources in preview.

## Where is data stored?

The report builder can select which tables they want to store in the report. This helps fine-tune the report's performance

## Data Sources

- All databases, Excel, csv, Data Flows, Web, Social Media, and DataFlow
- Not for Live connect to multi-dimensional sources
- DirectQuery on Azure Analysis Services and other Power BI Data Sets is in preview

## Advantages

- Many-to-many relationships possible
- Combination of data sources possible
- Best of Import and Direct Query

## Disadvantages

- Security implications when querying data across multiple sources
- Slower performance possible when querying multiple sources
- No user authentication at the data source
- Reidentification risk from combining data sets



# Live Connection

## What?

Similarly to Direct Query, Live Connection does not store the data. Live Connection has no Power Query and limited DAX, which makes that modeling has to be done at the Source. Power BI will get the entire model from the data source. As such, Live Connection can only be used as a visualisation tool. Furthermore, sources are limited to SSAS Tabular and other Power BI data sets. You can live connect to data sources using XMLA endpoints which are discussed later.

## Where is data stored?

- Data is not stored. However, the data is cached.

## Data Sources

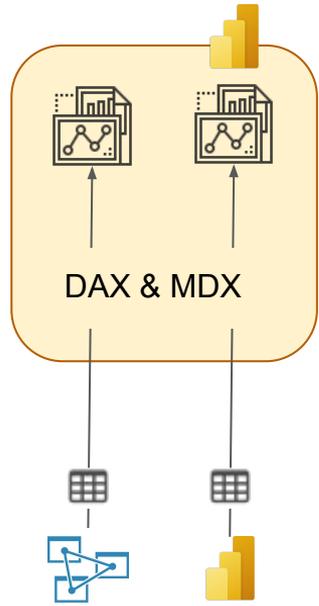
- SSAS Tabular
- Power BI Service
- Microsoft Dataverse

## Advantages

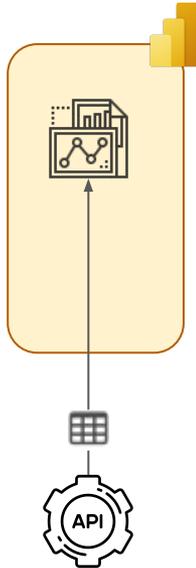
- No size limitations
- Less data proliferation + Authentication at data source => easier to govern
- Data Modeler and Report Author can be different persons
- Fresh reports
- Has RLS

## Disadvantages

- Limited number of data sources
- No Power Query, and limited DAX functionality



# Streaming Data



## What?

You can use the Power BI API to stream data. The data is not stored, and visuals are updated with very little latency.

## Where is data stored?

Data is stored in a temporary cache (Azure Redis Cache), which quickly expires.

## Data Sources

- API
- Azure Stream Analytics

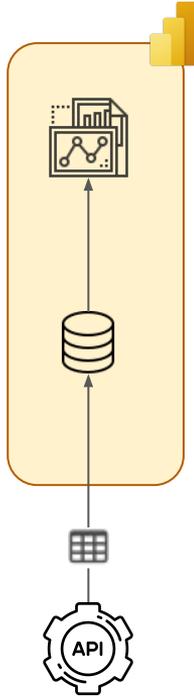
## Advantages

- Streaming visuals with very little latency

## Disadvantages

- Authentication is not mandatory
- Very limited reporting capabilities as data is not stored
- No data modeling

# Push Data



**What?**

Similarly to Streaming Data, you can use the Power BI API to push data into a Power BI dataset. The API can be used for creating new datasets, adding tables, and adding and deleting rows. As rows are added to the dataset, dashboard tiles update automatically. Therefore, this is typically used for streaming data. The difference with Streaming Data, is that historical data is stored.

**Where is data stored?**

Power BI online (Azure Blob Storage)

**Data Sources**

- API
- Azure Stream Analytics

**Advantages**

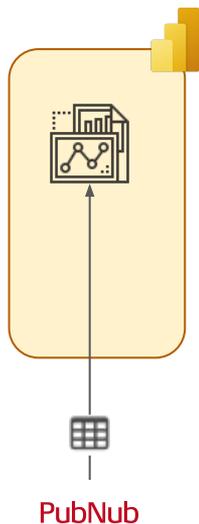
- All API-calls are subjected to authorization through integration with Azure AD OAuth
- Data Modeling is possible
- Reporting capabilities are available

**Disadvantages**

- Difficult to govern as data is not managed centrally, and API's can be used to add tables.
- Same data size limitations as for Import. By default FIFO is applied when limit is reached. For large data sets it is recommended to store data in warehouse.
- Data throughput is limited.



# PubNub



	Desktop	⊗	Service	⊙	
Extract	⊙	Transform	⊗	Load	⊗

## What?

With a PubNub streaming dataset, the Power BI web client uses the PubNub SDK to read an existing PubNub data stream. No data is stored by the Power BI service.

## Where is data stored?

Data is not stored. Not even in a cache.

## Data Sources

PubNub Data Streams

## Advantages

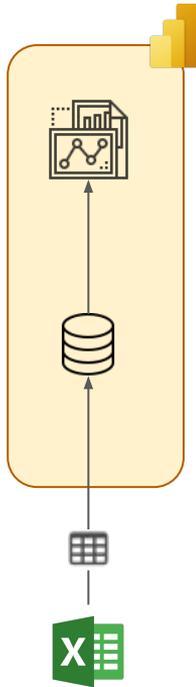
- Streaming visuals with very little latency

## Disadvantages

- No user authentication
- Limited reporting capabilities as data is not stored



# Excel



## What?

Excel is one of the most common ways to get your data in Power BI. Users can Import their Excel files including all tables, relationships, and calculations into their or a shared workspace in Power BI, and share the Power BI report. Now you can also use DirectQuery to access Excel files in preview. This will not store the data in PowerBI.

## Where is data stored?

Power BI workspace (Azure Blob Storage), unless when using DirectQuery.

## Data Sources

Excel

## Advantages

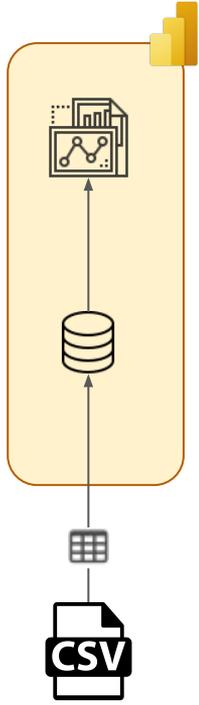
- Intuitive to create reports
- Users can import the data model from Excel into Power BI

## Disadvantages

- Decentralised so very hard to govern
- No access controls
- Excel can also query external data sources. This complicates access governance



# CSV



## What?

Users can upload data into Power BI from a csv file.

## Where is data stored?

Power BI workspace (Azure Blob Storage)

## Data Sources

Excel

## Advantages

- Intuitive to create reports

## Disadvantages

- Decentralised so very hard to govern
- No access controls



# Persist data



# Does Power BI persist data?

Depending on how Power BI queries the data, it can persist or cache the data. How the data stored depends on the data model which is discussed in the next slide.

Connection Mode	Power BI persists data?	Data Modeling possible
Import	Yes	Yes
Direct Query	No, but data is cached	Yes
Live Connection	No, but data is cached	No
Composite	When data is imported	Yes
Streaming	When configured	No



# PowerBI's data model

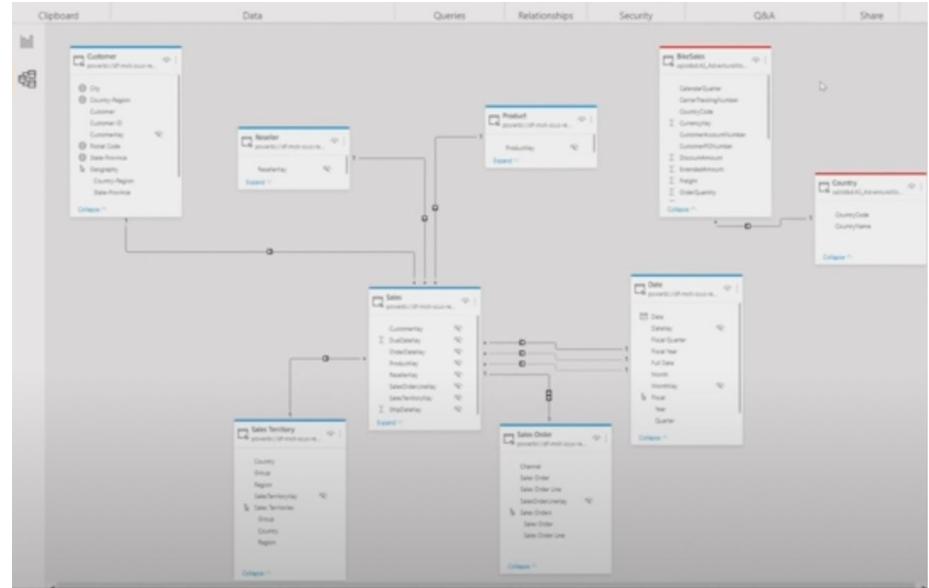
The power of PowerBI is that you can manage the data model for your report in Power BI itself. This means that you don't have to import your data as a flat file but you can use Power Query to model your data for your report.

A good data model makes it easier for business users to understand and share the data, and build meaningful and fast reports in a self-service analytics framework. The downside is that in a federated environment users can create bad data models resulting in slow reports, expensive queries, slower database performance, and can expose the organization to privacy and security risks which are hard to govern.

Users can model the data by creating

- relationships between tables, even across data sources
- new calculated tables and columns
- measures, such as totals and averages

You can model your data when using Import, Direct Query or the Composite Model where you query data from different sources.



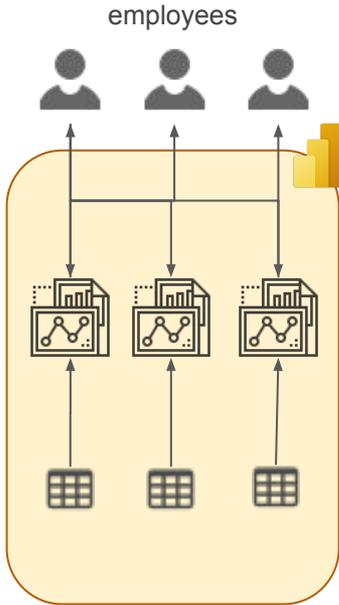
# Share data and reports



# Shared Workspace

## What?

Power BI's shared workspaces are places to collaborate with colleagues to create collections of dashboards, reports, datasets, and paginated reports. Where report builders can use their Personal Workspace to design reports, the Shared Workspace is best suited to collaborate with your team members on Power BI artifacts. Typically, a report builder designs their report in PowerBI desktop after which they publish it to Power BI Service.



**Can data be moved outside of PowerBI?**

Yes, even users with the Viewer role in the new workspace can export data if they have Build permission

**What can the person you shared with do?**

There are 4 role types (Admin, Member, Contributor, Viewer) with Admin having full control, and the Viewer having only view permissions. This is discussed in detail below.

**Advantages**

- Easy way to collaborate on Power BI artifacts
- RLS on Viewers available
- You can organize workspaces to separate DEV from PROD and TEST

**Limitations**

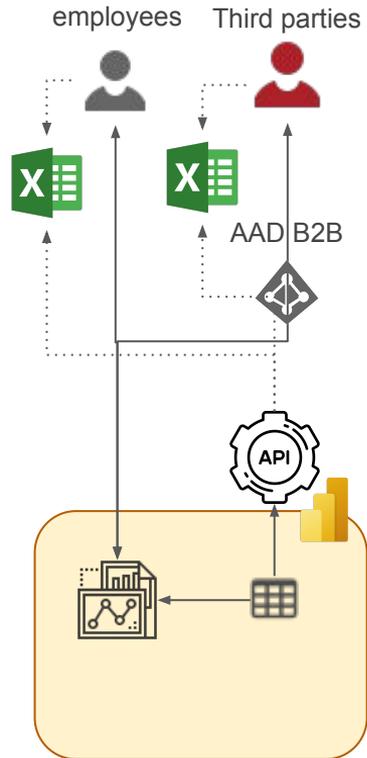
- Can be confusing for the end user. It's better to share the report or to publish an app
- Requires Power BI Pro or Premium Per User (PPU) license
- No easy way to copy reports between workspaces

**Note**

This applies to the new workspaces, not to the classic workspace.



# Shared Dashboards and Reports



**What?**

**Can data be moved outside of PowerBI?**

**What can the person you shared with do?**

**Advantages**

**Limitations**

Power BI allows report owners to share dashboards and reports in the Power BI service with people inside and outside of the organization. When you share a dashboard or report, the people you share it with can view and interact with it, but can't edit it. They see the same data that you see in the dashboard and reports and get access to the entire underlying dataset unless row-level security (RLS) is applied to the underlying dataset. In case you want to share with third parties, you'll have to use Azure AD B2B. To share dashboards and reports from within Power BI desktop, you have to upload to Service and share it from there.

Yes, users can access the underlying dataset by using Analyse in Excel, unless restricted by the Admin.

- View and interact with dashboards and reports
- Edit Dashboards & Reports
- Co-workers can share with other co-workers when allowed
- Build reports with the underlying dataset when permissioned
- Analyse in Excel when permissioned

- Allows for a single source of truth => better governance
- Granular permissions (RBAC, RLS)
- Easy to refresh the reports

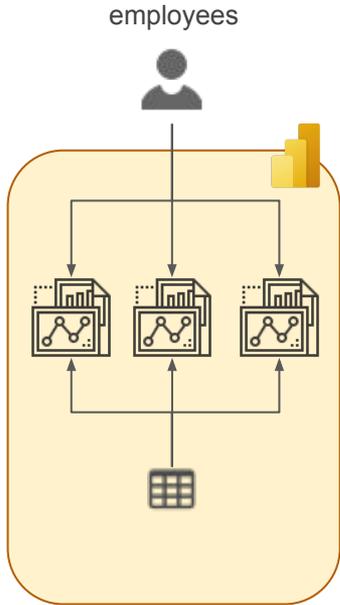
- You cannot share from Power BI Desktop
- You can only share one Power BI artifact at a time
- Cannot be used for collaboration



# Shared DataSet

## What?

In order to curb the proliferation of data models, organizations can introduce shared data sets. Skilled data modelers can publish their data sets to Power BI Service so that other users can create multiple reports using the same data set from within Power BI Desktop or other workspaces in Service. The underlying technology is XMLA which is discussed below.



**Can data be moved outside of PowerBI?**

No, you can only connect with Live Connection

**What can the person you shared with do?**

Users with build permissions can discover and built reports with the data set

## Advantages

- Allows for a single source of truth => better governance
- You can discover data sets in the catalog
- You can promote and certify reports and data sets to attest their quality

## Limitations

Users that query shared data sets cannot:

- Select a subset of the datamodel
- combine data sets
- Use Left-navigation or model the data

## Note

You can now query data sets in Preview with Direct Query which will resolve some of the limitations of using live connection.

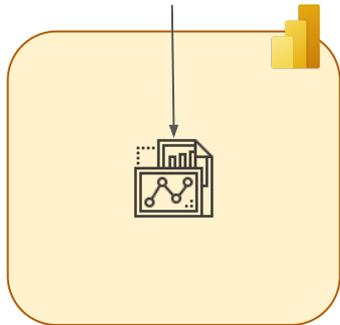


# Paginated Reports

## What?

Paginated reports are designed to create printable reports. They fit well on a page, and all data in a table will be printed, even when it extends multiple pages. You can build Paginated Reports with the dedicated tool, Power BI Report Builder.

employees



**Can data be moved outside of PowerBI?**

The users can export reports that can contain tables, but they cannot export the underlying data sets

**What can the person you shared with do?**

Readers can read and export the reports.

## Advantages

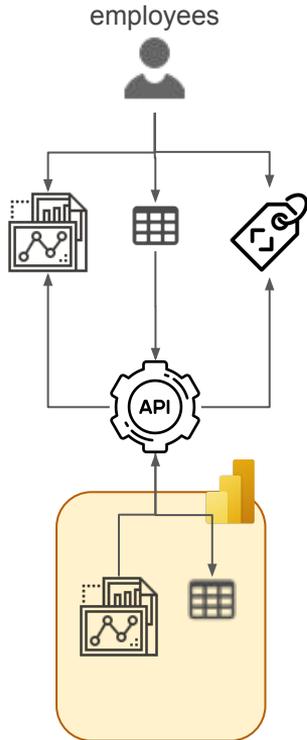
- Paginated Reports do not truncate tables when printing
- Paginated Reports follow a deployment pipeline
- You can combine multiple data sources
- You can use shared data sets

## Limitations

- Limited number of possible data sources
- Paginated Reports do not have a data model
- Paginated Reports need embedded data sources and data sets that cannot be shared
- No drill through reports



# Power BI API



## What?

With the Power BI API, you can use other applications to manage and embed Power BI reports, dashboards and tiles. These applications have to be registered in Power BI.

## Can data be moved outside of PowerBI?

Partially, users can export paginated reports that can contain data. Only the report owner can access the underlying dataset through API.

## What can the person you shared with do?

- Read & Export (Paginated) Reports
- Push Data into Data Sets (see above)
- Get meta-data

## Advantages

- Active Directory user authentication required
- Access Controls (RBAC, RLS) possible

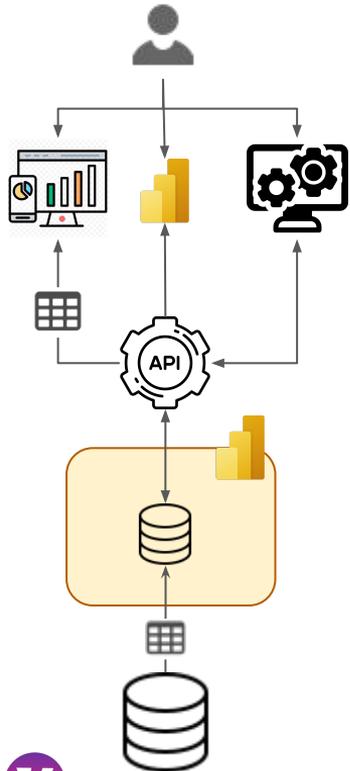
## Limitations

- Limited functionality
- No easy way for a report builder to share a report



# XMLA

employees



## What?

XMLA is a industry standard which allows any tool to use the Power BI API's to access, control, manage and monitor the datasets stored in the Analysis Server engine underlying Power BI Service. As a result, you can use other BI tools such as Tableau for data visualisations. This applies only to datasets stored in PowerBI. Hence, dataflows, and data queried through live connection are out of scope.

## Can data be moved outside of PowerBI?

Yes

## What can the person you shared with do?

Users have 2 types of permissions:

- Read-only: Query the data
- Read/Write: Query the data, manage and govern datasets, semantic modeling, debugging, and monitoring

## Advantages

- Allows for a single source of truth => better governance

## Limitations

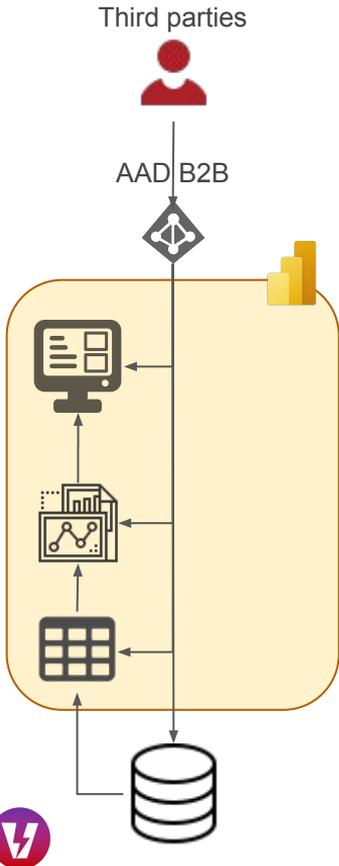
- Only for Power BI Premium
- Exporting data can lead to data proliferation
- No easy way for a report builder to share a report
- Data Sets based on LiveConnection or Push Data cannot be accessed through XMLA endpoints

## Note

XMLA has very little adoption beyond Microsoft, SAP and Oracle Hyperion.



# Collaborate with third parties



## What?

You can use Azure AD B2B to invite external guest users to collaborate on Power BI reports, while maintaining control. The partner can use their identity manager and their credentials. Both users as admins can add external guests.

## Can data be moved outside of PowerBI?

Yes, when guest users have access to the data sources

## What can the person you shared with do?

- Can access the data source
- Access to power BI apps
- Comment and subscribe
- Access, edit and manage reports
- Invite other external guest users

## Advantages

- You keep control with security roles, (dynamic) RLS, and conditional access
- You can collaborate with external guests without having to duplicate identities

## Limitations

- This can let the business add external guests which can create risk.

# Publish to Web

everyone



**What?**

Report builders can publish their reports to the Internet by embedding visualizations in blog posts, web sites, social media, and other online communications. Anyone on the Internet can view these reports, and you have no control over who sees what you've published. To publish a report to the web, you need to be a edit member in the corresponding workspace.

**Can data be moved outside of PowerBI?**

No. Users can only access reports.

**What can the person you shared with do?**

Everyone can see the report

**Advantages**

- Easy to embed interactive reports in webpages
- Report refreshes when the data refreshes unless refresh is switched off.
- Easy to publish

**Limitations**

- Once published there is not control over who can see the report. Anyone with the link can access the report.
- The types of reports that can be published is limited.
- No power BI native add-ons such as alerts, Q&A,...



# Embed for your customers

customers



## What?

With embedded analytics and power bi embedded you can embed Power BI content such as reports and dashboards into your customer application. You have to register the application in Azure AD which will be used to authenticate and authorize the access requests. Azure AD B2C is not compatible with Power BI so you cannot use AD authentication ('App owns data'). Instead, you authenticate the report users with embed tokens. Embedded analytics is recommended for software developers that want to embed reports in their applications.

## Can data be moved outside of PowerBI?

No

## What can the person you shared with do?

Consult reports and dashboards

## Advantages

- Easy way to share Power BI content with customers
- Customers do not need a Power BI account
- Safer than publishing to web, as you need an embed token to access the report.
- You can configure RLS when authenticating the user with embed token

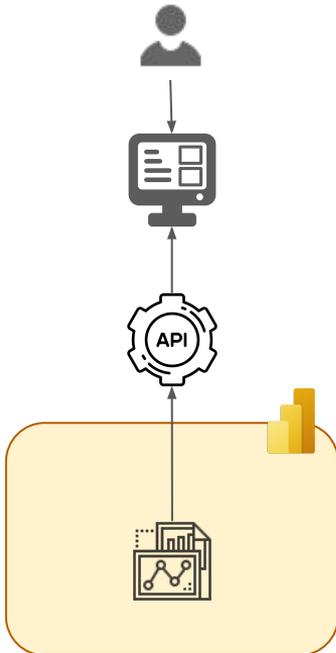
## Limitations

- Requires Power BI Pro
- Needs to be embedded in an application. You cannot share through a one-click publish
- No power BI native add-ons such as alerts, Q&A,...



# Embed for your organisation

employees



## What?

You can embed Power BI artifacts such as reports, dashboards and tiles into an application that you can share with your organisation. Contrarily to embed for your customers, embed for your organisation uses AD to authenticate the users ('User owns data'), so that you can customize the reports, and apply granular access controls. You can embed in Microsoft Teams, Sharepoint, or custom apps. Embedded analytics is recommended for software developers that want to embed reports in their applications.

## Can data be moved outside of PowerBI?

No

## What can the person you shared with do?

See Power BI API

## Advantages

- Easy way to share Power BI content within your organisation
- Users are authenticated which allows for customizations and granular access controls
- Same permissions as on workspace

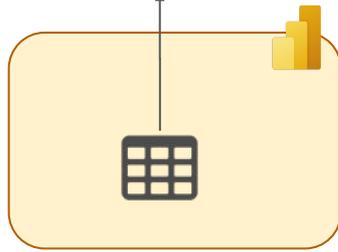
## Limitations

- Requires Power BI Pro or Premium Per User
- Needs to be embedded in an application. You cannot share through a one-click publish
- No power BI native add-ons such as alerts, Q&A,...
- You cannot embed multiple reports



# Export to Excel

employees



## What?

Users can export copies of the underlying datasets of Power BI reports for further analysis in Excel. You can export the full data set, not just the data which is visualised in the report.

## Can data be moved outside of PowerBI?

Yes, users get a copy of the data when they use Export to Excel

## What can the person you shared with do?

Export the dataset to Excel. From there on, they can do anything with that data

## Advantages

- Practical for additional analysis
- Report owners have control over what can be exported
- Data which is labeled as Confidential cannot be exported. This only applies to the labels in the Power BI report.

## Limitations

- Hard to govern. There are no access controls once the data has been exported to Excel
- Data is not refreshed
- The number of rows that can be exported are limited

# Analyse in Excel

## What?

With Analyze in Excel, you can bring Power BI datasets into Excel, and then view and interact with them using PivotTables, charts, slicers, and other Excel features. Contrarily to Export to Excel, the Excel file will have a live connection to the data set, and you can also access the full data model, rather than just the data set. Analyze in Excel uses a private XMLA endpoint.

employees



Can data be moved outside of PowerBI?

Yes, users can copy paste the data from the Excel file.

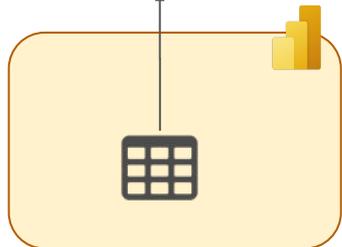
What can the person you shared with do?

- Read the data and run Pivot tables, and charts on that data
- Share the Excel file with other users



Advantages

- Data is refreshed
- Number of rows are unlimited
- Security measures such as RLS are applied, even when the Excel file is shared with other users.
- The Excel file inherits the sensitivity labels of the data, and will be protected as such. The sensitivity label is kept up-to-date with any changes of the label in Power BI.
- Can be easily turned off by the Admin

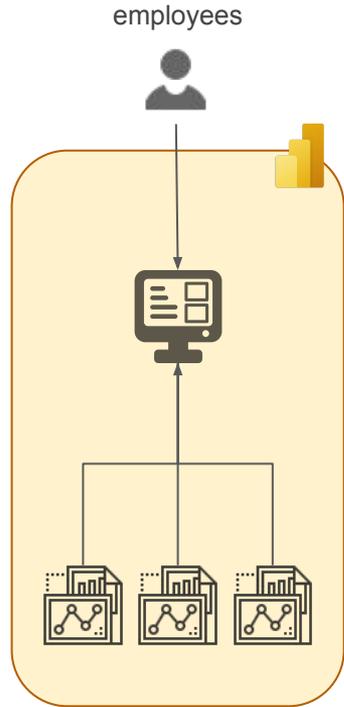


Limitations

- Harder to govern than Export to Excel as users can also query the full data model.



# Power BI Apps



**What?**

In Power BI, you can package dashboards and reports, and share them with your colleagues as an app. The app can bundle multiple dashboards and reports. The app can be shared in different ways.

**Can data be moved outside of PowerBI?**

No

**What can the person you shared with do?**

With the right build permissions users can:

- access the underlying dataset to create their own reports
- copy reports to other workspaces

**Advantages**

- User friendly dynamic reporting for the receiver
- Easier to manage permissions
- Developer can manage refresh rate
- Giving build permissions to an App, does not automatically give build permissions to data sets in other workspaces.
- Integrates with Azure B2B

**Limitations**

- There's a maximum of 200 dashboards in an App
- The recipient must either have a Power BI Pro or Premium Per User (PPU) license or the app must be shared with the recipient in a special type of cloud storage called Premium capacity.
- No separation of dev and production publishing environment as all changes in the underlying workspace are directly reflected in production



# Discover & Catalog



# Discover & Catalog (1/2)

In order to manage privacy and security risks, it is important to have a good view of the data and reports accessible through Power BI. When Power BI Desktop creates reports, it also creates dataset metadata in the corresponding PBIX and PBIT files. In 2020, PowerBI enhanced this meta-data to expose models, tables, columns and relationship objects using the Tabular Object Model (TOM). Power BI offers several approaches to discovering the data and reports. In the next slide, we list the features in Power BI that help you with discovering and cataloging data.



# Discover & Catalog (2/2)

Aside dedicated software such as Purview, Collibra, and Alation, Power BI also offers discovery & cataloging capabilities.

	Discovery Hub	Power BI API	Tabular Object Model (TOM)	TMSL
<b>What?</b>	The discovery hub lists all curated, non-curated and recommend datasets that have been shared with the user. Users use datahub to find datasets for their reports.	Under the enhanced meta-data, meta-data for all Power-BI artifacts is stored as JSON. Admins can use the API's for very granular scans of the meta-data.	TOM is a C# library which exposes PowerBI's meta-data on its models, tables, columns, and relationships, which is stored as JSON. TOM uses the XMLA protocol to communicate with PowerBI.	The TOM can also be exposed through TMSL scripts, which is a JSON scripting language to manage Power BI's data model through the XMLA endpoints. You can use powershell cmdlets or SSMS for this
<b>Pros</b>	<ol style="list-style-type: none"> <li>1) User friendly UI</li> <li>2) Contains sensitivity Label</li> </ol>	<ol style="list-style-type: none"> <li>1) Very granular meta-data on all artifacts (workspaces, data sources datasets, measures, tables, columns, rows,...), such as ownership, sensitivity labels, data category, lineage, DAX expressions, encryption status, and data category.</li> <li>2) Meta-data is cached for performance</li> </ol>	Very granular technical meta-data on measures, tables, columns and relationships such as lineage, data category, data type, format, and description	<ol style="list-style-type: none"> <li>1) TOM's granular meta-data Scripting language uses JSON which is intuitive</li> <li>2) Can be used to read and write meta-data</li> </ol>
<b>Cons</b>	<ol style="list-style-type: none"> <li>1) You can only see the data sets for which you have build permissions. Hence cannot be used for governance.</li> <li>2) Not yet fully released</li> </ol>	<ol style="list-style-type: none"> <li>1) Not compatible with older versions of Power BI desktop</li> <li>2) No UI</li> <li>3) Only shows meta-data for datasets that have been refreshed</li> <li>4) No meta-data for data sets that use Push, Streaming, PubNub, LiveConnection.</li> <li>5) No meta-data for datasets with OLS</li> </ol>	<ol style="list-style-type: none"> <li>1) The API is only available for managed code over the .NET framework, and is limited to C#</li> <li>2) Does not get meta-data on server and model</li> <li>3) No meta-data for data sets that use Push, Streaming, PubNub, LiveConnection.</li> </ol>	<ol style="list-style-type: none"> <li>1) Less functionality than TOM</li> <li>2) No meta-data for data sets that use Push, Streaming, PubNub, LiveConnection.</li> <li>3) No access to sensitivity label</li> </ol>

# Control



# Access controls in Power BI

Power BI is an innovative leader when it comes to access management. A few of the areas where PowerBI excels are:

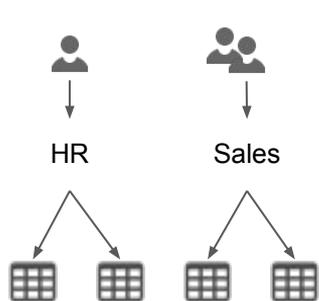
- Row-Level Security: Where you can filter out certain records depending on the user
- Single Sign On (SSO): Where you authenticate the user at the data source

In this section we will elaborate on the different ways Power BI authorizes access to the data. For other topics such as encryption, and authentication, we refer to the documentation. Given the tight integration with Azure, we will sporadically expand into authorization at the Azure data sources. The structure of this section is:

- Terminology: Where we introduce you to the different access control techniques (Slide 43)
- Overview of the different levels to govern access (Slide 44)
- Access Controls at the Data Source (Slide 45-46)
- Access Controls on the Data Set (Slide 47-49)
- Access Controls on the Workspace (Slide 50)
- Access Controls when sharing (Slide 51-52)

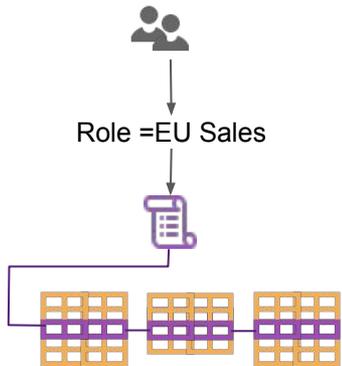


# Terminology



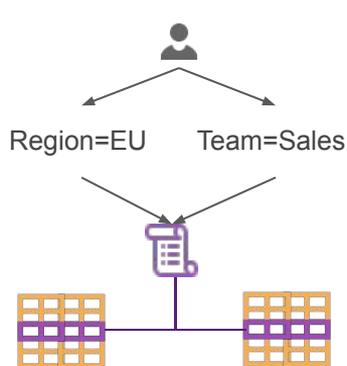
## Role-based permissions

PowerBI and most of the data sources it connects to let you govern access by role, and assign users or user groups to those roles. Roles are set at the database, and only apply to the database where they've been defined.



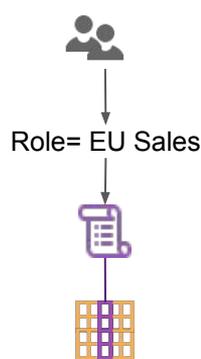
## Row-Level Security (RLS)

Certain use cases such as Chinese Walls, and localisation, require you to restrict access to rows. In that case, you can write RLS rules that filter records, to which you can assign Roles. For instance, EU Sales can only see the data of European Customers. You can cross-filter related tables, extending RLS to multiple tables.



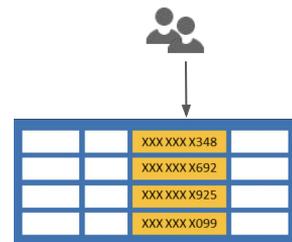
## Dynamic RLS

RLS rules can be hard to maintain. With every change you have to update the rules. With Dynamic RLS, you filter the records on attributes of the user, rather than the role they're assigned to. Soon as the attributes change, different rules apply. As with RLS, you can cross-filter related tables.



## Object-Level Security (OLS)

Tighter privacy regulations and increased security risks require more granular access controls. With OLS, you can restrict access to tables and columns, but also their meta-data so that the objects are hidden from malicious users. When permissions are managed at column level, this is called Column Level Security (CLS)



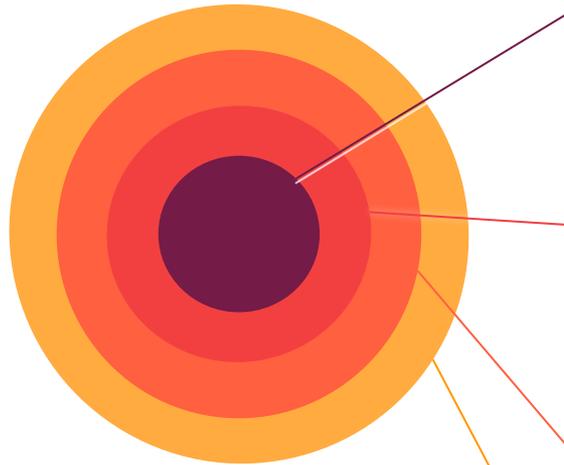
## Data Masking

With Data Masking, you still have access to the data, but the data is masked to preserve privacy and security. Depending on the data type and the level of information you want to disclose, there are multiple ways of masking data. The available levels of granularity are schema, table and column level.



# Levels of governing access in Power BI

You can govern access to Power BI at several levels. Access Control at each level has its use, and they're typically used in combination.



## Data Source (SSO) (slide 45-46)

When the data source integrates with Active Directory and supports Single Sign On (SSO), you can define Access Controls on the Data Source, rather than in Power BI's data model. The resulting centralisation of access controls means a smaller risk surface and as a result, less maintenance and less risk of inconsistent or missing access controls. Where possible, it is recommended to use SSO and configure access controls at the data source.

## DataSet (slide 47-49)

Power BI lets Admins protect data in

- The data set: who can read, build, and share.
- The data: granular access permissions to the underlying data in the report's data model.

## Workspace (slide 50)

The Admin can also govern permissions at the Workspace level. These permissions determine which actions a user can take

There are 4 roles Viewer, Contributor, Member and Admin, each with different permissions to view, build and publish reports in the workspace.

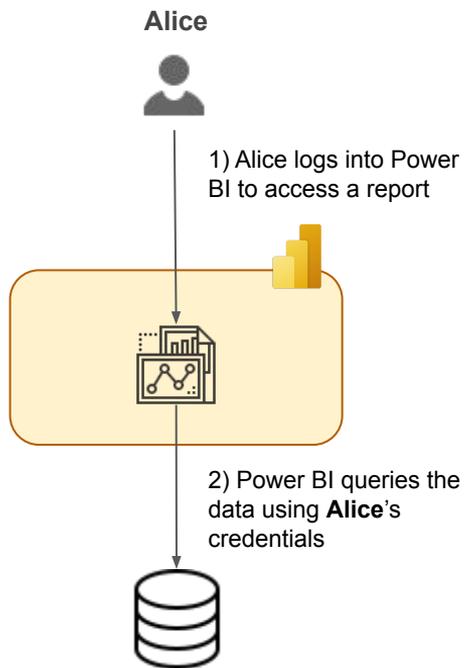
## When sharing (slide 51-52)

There are various ways to share Power BI reports and data sets. Depending on the method, different types of access controls are available.



# Data Source (SSO)

## Authentication at the Data Source



With Power BI, it is possible to govern access to the report data using the access controls available at the data source. This means that the PowerBI report will only contain the data that the report user is allowed to see as if they were to query the underlying data source directly.

This simplifies data governance considerably as you can manage your access controls in a central location instead of across a disparate set of Power BI reports. It also reduces the risk surface because the report user cannot bypass the controls by directly querying the data source. Therefore, access controls at the data source are recommended where possible.

In order to govern access at the data source, the report user has to be authenticated to that data source, through Single Sign On (SSO).

This functionality is available for:

- DirectQuery
- LiveConnection
- Paginated Reports

Note that Import also has the option for SSO. However, in that case, the Report Builder's credentials are always used, irrespective of the identity of the report user.

Next, we list the data sources that integrate with Azure Active Directory for SSO.



# Data Sources that support SSO through integration with Azure AD

	Windows Authentication	Kerberos	SAML	OAuth2
SQL Server Database	X	X		
SSAS	X			
Azure SQL Database				X
Azure Synapse Analytics				X
Oracle Database	X	X		
SAP Business Warehouse server		X		
SAP HANA Database	X	X	X	
Teradata	X	X		
Spark		X		
Impala		X		
Denodo		X		
Snowflake				X
Azure Databricks	X			
GCP			X	



# DataSet

There are 2 different ways of governing permissions on the dataset level, that are closely intertwined with the roles managed at the level of the workspace as discussed further.

- **On the DataSet:** Determines what you can do with the dataset.
  - *Read:* this allows users to see the reports based on the dataset. Users automatically get read permissions when they're part of, or have view permissions on the workspace.
  - *Build:* this allows users to create reports from the dataset.
  - *Share:* this allows users to reshare the dataset.
- **On the DataModel:** Determines which data you can see within the dataset.
  - *Role-based permissions:* Power BI lets you determine roles, assign users to those roles, and map the roles to RLS and OLS. Mind you, this only applies to users with View permissions on the dataset's workspace. It is not applied to users with build permissions.
  - *Row-Level Security (RLS):* With RLS you can filter out records from a table and the related tables per role . RLS works by applying filters to every DAX query. As a result, RLS can be limited in DirectQuery which has less DAX functionality than Import. For RLS to work across tables, and to be performant, you need a good data model. Poorly configured RLS can significantly impact performance. Finally, RLS rules are also applied to all linked data sets.
  - *Dynamic RLS:* Power BI allows you to make your RLS dynamic which saves you a lot of time in maintenance. Rather than mapping RLS to fixed roles, you can use DAX expressions to set RLS using the report user's attributes:
    - Their name using `username()` or `userprincipalname()`
    - Other attributes stored in a permission table in your data model using `contains()` or `lookupvalue()`. This means you have to define the logic of security inside the data model (tables, their relationships, etc). For a change in the logic, you just need to add/edit/delete records in the tables
  - *Object Level Security (OLS):* OLS is in Power BI is in public preview, and lets you set permissions at the tables and columns in the Power BI data model. Contrarily to RLS, OLS also hides the meta-data from the users.
  - These techniques are only available in case of Direct Query and Import. For LiveConnection, the controls have to be on datasource.



# Control languages & tools (1/2)

	Description	Which Controls	Tools to set controls	Scope of the controls
DAX	Data Analysis Expressions (DAX) is a formula expression language used in Analysis Services, Power BI, and Power Pivot in Excel. RLS rules are configured as DAX expressions that are automatically applied to all DAX queries performed by roles mapped to the RLS rules. As a result the user only sees the records they're allowed to see. You can also implement dynamic RLS where you filter on user attributes stored in the data model using the DAX expression USERNAME(), rather than manually assigning the RLS rules to Roles. RLS expressed as DAX-rules are limited in case of DirectQuery and LiveConnection, and only filter READ requests.	RLS	Volta Power BI Tabular Editor	Power BI SSAS AAS
XMLA	XML for Analysis (XMLA) is a SOAP-based XML protocol, which allows you to access the data models in Power BI, SSAS and AS over HTTP. As mentioned earlier, you can use XMLA to live connect to the data, but it can also be used to manage meta-data and controls. You can use XMLA to define RLS rules (DAX), define Roles and assign users to Roles.	Roles Role assignments RLS (DAX) OLS	Volta Visual Studio SSMS PowerShell cmdlets Power BI Report Builder DAX Studio	Power BI SSAS AAS
API	Azure lets you use REST API's to manage controls in Azure SQL Database, and Synapse Analytics. <i>Note 1:</i> Data Masking Policies can only be set the user level with API's. If you need role based Data Masking Policies, you'll have to use T-SQL. <i>Note 2:</i> If you need dynamic data masking where you maintain the original format, you'll have to use T-SQL. <i>Note 3:</i> You cannot use the API's to define roles, only to assign users to roles. <i>Note 4:</i> Although you cannot set access controls in PowerBI using the REST API's, you can use the API's to get the users and their access levels.	Role assignments Data Masking	Volta Various	Azure SQL Database Azure Synapse Analytics <del>Power BI</del>



# Control languages & tools (2/2)

	Description	Which Controls	Tools to set controls	Scope of the controls
TMSL	<p>Tabular Model Scripting Language (TMSL) lets you formulate object definitions against tabular data models in JSON and XML using the XMLA endpoints. TMSL lets you manage objects like databases and tables, and manage access controls. You cannot use TMSL to configure write permissions. It can only be used for read, refresh and administrator permissions.</p> <p>Similarly to TMSL, there is Analysis Services Scripting Language (ASSL), which is restricted to multidimensional model databases like SSAS, which is not compatible with PowerBI.</p>	Roles Role Assignments RLS OLS	Volta SSMS PowerShell cmdlets	Power BI SSAS AAS
T-SQL	<p>Transact-SQL (T-SQL) is Microsoft extension to SQL used to interact with relational databases. T-SQL is central to using Microsoft SQL products and services. All tools and applications that communicate with a SQL database do so by sending T-SQL commands.</p> <p>T-SQL can be used to manage read, and write permissions. Write permissions can be governed using Block predicates which are DML triggers that check the integrity of any change in the database. Therefore they can be used to manage write per Roles, RLS.</p> <p>T-SQL is not compatible with Power BI.</p>	Roles Role Assignments RLS CLS OLS Data Masking	Volta SSMS Azure Data Studio SSDT sqlcmd	SQL Server Azure SQL Database Azure SQL Managed Instance Azure Synapse Analytics <del>Power BI</del>



# Workspace

PowerBI lets you use Roles to manage access on the Workspace level. You can assign individual users and user groups to those roles. There are 4 role types. Typically, data engineers will use workspaces-based separation to manage access to reports and the underlying data sets.

Action	Admin	Member	Contributor	Viewer
Update and delete the workspace.	X			
Add/remove people, including other admins.	X			
Allow Contributors to update the app for the workspace	X			
Add members or others with lower permissions.	X	X		
Publish, unpublish, and change permissions for an app	X	X		
Update an app.	X	X		
Share an item or share an app.	X	X		
Allow others to reshare items	X	X		
Feature apps on colleagues' Home	X	X		
Manage dataset permissions.	X	X		
Feature dashboards and reports on colleagues' Home	X	X	X	
Create, edit, and delete content in the workspace.	X	X	X	
Publish reports to the workspace, delete content.	X	X	X	
Create a report in another workspace based on a dataset in this workspace.	X	X	X	
Copy a report.	X	X	X	
Create goals based on a dataset in the workspace.	X	X	X	
Schedule data refreshes via the on-premises gateway.	X	X	X	
Modify gateway connection settings.	X	X	X	
View and interact with an item.	X	X	X	X
Read data stored in workspace dataflows	X	X	X	X



# When Sharing - Fine Grained Access Controls

When sharing reports and datasets from within PowerBI, it is also possible to set fine-grained access controls (RLS, OLS). As with access controls on the data set level, users will need view permissions on the workspace and will have to be authenticated.

When users can be authenticated, fine-grained RBAC are possible.

Method of sharing	Fine-Grained RBAC	Notes
Shared Workspace	X	
Shared Dashboards and Reports	X	
Shared DataSet	X	Users need view permissions to see the report and build permissions to use the Shared Dataset in their reports.
Paginated Reports	X	
Power BI API	X	User is authenticated with embed token. This does not work for paginated reports
XMLA	X	Users are authenticated with Azure Active Directory (AAD) and AAD B2B
Collaborate with third parties	X	Users are authenticated with Azure Active Directory B2B
Publish to Web		Reports that are published to Web do not have access controls
Embed for your customers	X	User is authenticated with embed token. This does not work for paginated reports
Embed for your organization	X	User is authenticated with Azure Active Directory
Export to Excel (After data has been exported)		Once data is exported to Excel, access cannot be controlled anymore
Analyse in Excel	X	Access controls are applied when refreshing the data in Excel
Power BI Apps	X	



# When Sharing - Sensitivity Labels

Power BI is integrated with Microsoft Integrated Protection (MIP) sensitivity labels, which encrypts documents to prevent unauthorized access. This means that you can add sensitivity labels to datasets, reports, dashboards and dataflows, so that when your data leaves Power BI it will be encrypted and only authorized users will have access. This applies to:

- Export to Excel, PDF and Powerpoint
- Analyse in Excel
- Download to .pbix from Power BI Service.

Power BI reports inherit sensitivity labels from shared data sets, and Azure Synapse and Azure SQL database.

Mind that this only encrypts the data and does not help with fine-grained access controls.



# Monitor



# Audit & Monitor

It is very important to monitor usage in Power BI for regulatory compliance and security reasons. Power BI offers 2 ways of monitoring

	User Activity Metrics	Usage Metrics (preview)
<b>Definition</b>	Monitor how users are managing resources in Power BI. There are 2 types of logs: Unified Audit Logs which store logs for Power BI, SharePoint Online, Exchange Online etc, and Power BI Activity Logs which only stores Power BI audit events as JSON files.	Monitor usage and performance of reports. These metrics rely on the user activity metrics.
<b>How</b>	<ul style="list-style-type: none"><li>- <i>Activity Log</i>: Power BI API, and cmdlets. There is no interface.</li><li>- <i>Audit Log</i>: Microsoft 365 Management Admin Center and APIs, export to csv, and cmdlets.</li></ul>	<ul style="list-style-type: none"><li>- Pre-built Power BI Usage Reports</li><li>- Custom Power BI Usage Reports</li><li>- Connect to Usage metrics dataset</li><li>- Analyze in Excel</li></ul>
<b>Who</b>	<ul style="list-style-type: none"><li>- <i>Activity Log</i>: Power BI service admin and Global admin role in Microsoft 365</li><li>- <i>Audit Log</i>: Global admin role in Microsoft 365</li></ul>	Users with admin, member or contributor role. Viewer permissions are not sufficient
<b>Retention</b>	<ul style="list-style-type: none"><li>- Activity Log: 30 days</li><li>- Audit Log: 90 days</li></ul>	30 days

