Power BI & Model Context Protocol (MCP)

Security & Technical Integration Overview

Prepared for Client Review

February 2026

---

1. Introduction

This document provides a comprehensive overview of how Microsoft Power BI integrates with the Model Context Protocol (MCP) for AI-assisted semantic model development. It is intended to give clients, IT security teams, and stakeholders a clear understanding of the security architecture, data access controls, and operational safeguards in place when Power BI MCP tools are used in development and consulting engagements.

The Model Context Protocol (MCP) is an open standard developed by Anthropic that defines how AI assistants interact with external tools and data sources in a structured, secure manner. Microsoft has adopted this standard for Power BI through two official MCP server implementations: the Modeling MCP Server (for local development) and the Remote MCP Server (for Fabric workspace access).

2. Architecture Overview

2.1 How MCP Works with Power BI

The MCP architecture consists of three core components that work together to enable AI-assisted Power BI development:

Component	Description
Host	The AI application that initiates the connection (e.g., VS Code, Claude Desktop, or a custom application)
Client	A component within the host that connects to MCP servers and consumes their capabilities (e.g., GitHub Copilot, Claude)
Server	A local or remote program that exposes Power BI semantic modeling tools, resources, and prompts to the AI client

2.2 Two Server Modes

Microsoft provides two distinct MCP server implementations for Power BI, each serving different use cases:

Feature	Modeling MCP Server (Local)	Remote MCP Server
Runs On	Developer's local machine	Microsoft Fabric cloud (api.fabric.microsoft.com)
Connects To	Power BI Desktop, PBIP files, or Fabric workspaces	Published Fabric semantic models
Authentication	Azure Identity SDK / Entra ID for Fabric; local SSAS protocol for Desktop	Microsoft Entra ID (required)
Capabilities	Full read/write: create, modify, delete model objects; execute DAX queries	Read and query: retrieve schemas, generate and execute DAX queries
Network Scope	Localhost only (for Desktop); HTTPS for Fabric	HTTPS to Microsoft Fabric endpoints
Use Case	Development and model management workflows	Data analysis and insights scenarios

2.3 Communication Protocol

All communication between the MCP client and server is handled via JSON-RPC 2.0, a lightweight remote procedure call protocol. For local connections (Power BI Desktop), communication occurs over the local Analysis Services (SSAS) protocol on localhost. For remote connections (Fabric), all traffic is transmitted over HTTPS/TLS to Microsoft's api.fabric.microsoft.com endpoint.

3. Authentication & Access Control

3.1 Microsoft Entra ID Authentication

The Power BI MCP Server uses the Azure Identity SDK for authentication. Tokens are not stored by the MCP server itself. For Fabric workspace connections, authentication is handled entirely through Microsoft Entra ID (formerly Azure Active Directory). This is standard enterprise authentication infrastructure, not a custom or proprietary implementation.

- Service Principal authentication is supported for automated and headless environments, using certificates or client secrets registered in Microsoft Entra ID.
- Interactive (delegated) authentication is supported for developer workflows, where the user authenticates via browser-based OAuth 2.0 flows.
- Token management follows Microsoft's standard token lifecycle, including automatic refresh and expiration handling.

3.2 Role-Based Access Control (RBAC)

MCP operations are executed under the authenticated user's existing Fabric RBAC permissions. The MCP server does not elevate or bypass any permissions. If a user has

Contributor access to a workspace, the MCP client can only perform actions that a Contributor can perform. If the user has Viewer access, the MCP client is limited to read-only operations.

Fabric Role	MCP Capabilities
Admin	Full read/write access to all semantic model objects, including creating, modifying, and deleting measures, tables, columns, and relationships
Member	Read/write access to model objects within assigned scope
Contributor	Read/write access to model content; publish and update reports and semantic models
Viewer	Read-only access; can query data but cannot modify model objects

3.3 Row-Level Security (RLS)

Power BI's Row-Level Security is fully enforced during MCP operations. RLS uses DAX filter expressions assigned to security roles to restrict which rows of data are visible to specific users. When an MCP client executes a DAX query, the results are filtered according to the authenticated user's RLS role assignments. Both static RLS (fixed filters per role) and dynamic RLS (filters based on USERPRINCIPALNAME()) are supported and enforced.

3.4 Object-Level Security (OLS)

Object-Level Security, when configured on a semantic model, restricts visibility of specific tables and columns from users who are not assigned to the appropriate roles. OLS is enforced during MCP operations just as it is during normal Power BI report consumption. Users and MCP clients that are not assigned to the appropriate role will not see restricted columns or tables in query results or schema metadata.

3.5 Tenant Administrator Controls

For the Remote MCP Server, a Power BI administrator must explicitly enable the tenant setting "Users can use the Power BI Model Context Protocol server endpoint" before any MCP connections are permitted. This provides organizational control over whether MCP access is available at all. The administrator can enable or restrict this setting for the entire organization or for specific security groups.

4. What MCP Can and Cannot Access

4.1 What MCP CAN Access

Data Category	Access Level
✅ Semantic model schema (tables, columns, data types)	**Fully visible**
✅ DAX measure definitions and expressions	**Read and write (create, modify, delete)**
✅ Calculated columns and calculated tables	**Read and write**

✅ Table relationships and cardinality	**Fully visible; can create and modify**
✅ Display folders, descriptions, format strings	**Read and write**
✅ Hierarchies and perspectives	**Fully visible**
✅ RLS role definitions (DAX filter expressions)	**Role names and filter expressions visible**
⚠️ DAX query results (actual data values)	**Visible only when a DAX query is explicitly executed**

4.2 What MCP CANNOT Access

Data Category	Access Level
❌ Data source credentials (SQL passwords, API keys)	**Never accessible**
❌ Connection strings to underlying data sources	**Not exposed through MCP**
❌ Power Query (M) source code and transformations	**Not accessible; MCP operates at the semantic model layer only**
❌ Power BI Service account credentials or tokens	**Handled by Azure Identity SDK; not stored or visible**
❌ Operating system files, network shares, or local files	**No file system or network access**
❌ Other Power BI models not explicitly connected	**Only the connected model is accessible**
❌ Power BI Service workspaces not assigned to the user	**RBAC enforced; no cross-workspace access**

5. How MCP Handles Measures, DAX, Connections & Power Query

5.1 Measures and DAX Expressions

The MCP server provides comprehensive capabilities for working with DAX measures within a Power BI semantic model. These operations are the primary use case for AI-assisted Power BI development:

- Create new measures with DAX expressions, format strings, display folders, and descriptions.
- Modify existing measures, including updating DAX logic, renaming, or reorganizing into display folders.
- Delete measures that are no longer needed.

- Batch operations allow creation or modification of hundreds of measures simultaneously with transaction support and error handling. If any operation in a batch fails, the entire batch can be rolled back.
- Execute DAX queries (EVALUATE, TOPN, SUMMARIZECOLUMNS, etc.) against the semantic model to validate measures, test calculations, or retrieve sample data.

All measure modifications are performed through the Analysis Services Tabular Object Model (TOM). Changes are applied to the in-memory model and take effect immediately. For Power BI Desktop, changes are saved when the user saves the .pbix file. For Fabric workspaces, changes are persisted to the cloud model.

5.2 Calculated Columns and Calculated Tables

MCP can create, modify, and delete calculated columns and calculated tables that use DAX expressions. These are part of the semantic model layer and follow the same security controls as measures. Calculated columns are computed during data refresh and stored in the model; calculated tables are evaluated and materialized at refresh time.

5.3 Connections and Data Sources

MCP does not interact with the data connection layer of a Power BI model. It cannot view, create, modify, or delete data source connections. All connection management (including credentials, gateway configurations, and refresh schedules) remains exclusively within the Power BI Desktop interface or the Power BI Service administration portal. The MCP server has no visibility into how data enters the model.

> **Important: Connection Security**
>
> Data source credentials (database passwords, API keys, OAuth tokens) are managed by Power BI's own credential store and are never accessible to the MCP server or any AI client. The MCP layer operates entirely above the data ingestion layer.

5.4 Power Query (M Language)

Power Query transformations, written in the M language, define how data is extracted, transformed, and loaded into the semantic model. The MCP server does not have access to Power Query code or transformations. MCP operates exclusively at the semantic model layer (the Tabular Object Model), which sits above the Power Query ETL layer. This means:

- MCP cannot view or modify Power Query scripts or M code.
- MCP cannot alter data source definitions, refresh logic, or transformation steps.
- MCP cannot trigger data refreshes or modify refresh schedules.
- All data ingestion and transformation logic remains under the control of the model developer through Power BI Desktop or the Power BI Service.

6. Data Transmission & Encryption

6.1 Data Flow

Understanding how data moves during MCP operations is critical for security assessment. The following outlines the complete data flow:

Step	Direction	Description
1	**User → AI Client**	User provides natural language instructions (e.g., “create a YTD measure for revenue”)
2	**AI Client → MCP**	The AI client generates an MCP tool call with appropriate parameters (e.g., CreateMeasure with DAX expression)
3	**MCP → Power BI**	The MCP server sends the command to Power BI via the local SSAS protocol (Desktop) or HTTPS to Fabric
4	**Power BI → MCP**	Power BI returns the result (success/failure status, query results, error messages, or schema metadata)
5	**MCP → AI Client**	The MCP server returns results to the AI client, which presents them in the conversation

6.2 Encryption

- **In Transit:** Local connections (Power BI Desktop): Communication occurs on localhost via the SSAS protocol. Data does not traverse any external network.
- **In Transit:** Remote connections (Fabric): All communication is encrypted via TLS/HTTPS to Microsoft’s api.fabric.microsoft.com endpoint.
- **At Rest:** Power BI data at rest within Microsoft Fabric is encrypted using Microsoft-managed keys or customer-managed keys (CMK) depending on the organization’s configuration.
- **Token Security:** The MCP server uses the Azure Identity SDK for authentication. Tokens follow the standard Microsoft token lifecycle and are not persisted by the MCP server.

6.3 AI Provider Data Considerations

When an AI client (such as Claude, GitHub Copilot, or another LLM) is used alongside the MCP server, model metadata and query results returned by the MCP server become part of the AI conversation context. This data is transmitted to the AI provider’s infrastructure (e.g., Anthropic for Claude, or Microsoft/OpenAI for Copilot). Organizations should review the AI provider’s data retention and privacy policies to ensure alignment with their data governance requirements.

7. Operational Safeguards

7.1 User Approval Workflow

The Power BI Modeling MCP Server supports the Elicitation MCP protocol, which requires explicit user approval before performing sensitive operations:

- Before the first modification made to a semantic model in a session.

- Before the first DAX query executed against a semantic model in a session.

This ensures that no changes are made to a client's model without the developer's explicit consent. The approval requirement can only be bypassed by explicitly passing a --skipconfirmation flag, which should only be used in environments with version control and rollback capabilities.

7.2 Version Control and Recovery

Microsoft Fabric automatically captures up to five versions of semantic models edited via the web or Direct Lake in Desktop. Versions are saved automatically when publishing .pbix files or restoring previous versions, and rollback is available for versions less than 14 days old. For robust version control, Git integration can be configured for semantic models, providing full change history and rollback capabilities.

For Power BI Desktop development, the standard practice of saving .pbix files before making changes provides a manual recovery point. Organizations with compliance requirements should implement a formal backup strategy that includes pre-modification snapshots.

7.3 Audit Logging

Power BI provides comprehensive audit logging through the Microsoft Purview compliance portal. All MCP operations that result in model changes or data queries are logged through Power BI's standard audit infrastructure. Audit logs capture user identity, timestamp, operation type, workspace, and semantic model affected. These logs can be exported to Microsoft Sentinel, Azure Log Analytics, or third-party SIEM systems for security monitoring and compliance reporting.

7.4 Network Isolation

For the local Modeling MCP Server, the SSAS connection to Power BI Desktop operates exclusively on localhost (127.0.0.1). No external network traffic is generated for local model operations. For Fabric workspace connections, organizations can leverage Azure Private Link and virtual network integration to ensure that traffic between the MCP server and Fabric endpoints remains within the organization's private network.

8. Recommendations for Secure Deployment

Apply Least-Privilege RBAC Roles

Assign the minimum Fabric workspace role necessary for the task. For development work that requires creating and modifying measures, Contributor access is typically sufficient. Avoid granting Admin access unless absolutely necessary.

Use Development Models When Possible

When working with models that contain sensitive production data, use development or staging copies with anonymized or synthetic data. This ensures that even if DAX queries return row-level data, no sensitive values are exposed to the AI conversation context.

Review DAX Query Scope

DAX queries executed through MCP return actual data from the semantic model. Review queries before execution and avoid broad, unfiltered queries on tables containing sensitive row-level data. Use aggregations, filters, or sample-size limits (TOPN) to minimize data exposure.

Enable Audit Logging

Ensure that audit logging is enabled in the Microsoft Purview compliance portal. Monitor MCP-related operations alongside standard Power BI activity to maintain a complete audit trail of all model changes and data access.

Review AI Provider Data Policies

Understand how your AI provider handles conversation data, including retention periods and training data policies. If your organization has strict data governance requirements, confirm alignment before transmitting model data through the MCP integration.

Maintain Backup and Recovery Procedures

Before performing bulk operations or significant model changes via MCP, ensure that a backup exists. Use Power BI's built-in version history, Git integration, or manual .pbix file backups to provide rollback capability.

Summary

The Power BI MCP integration provides AI-assisted semantic model development within Microsoft's established security framework. Authentication is handled through Microsoft Entra ID, and all operations are governed by the user's existing Fabric RBAC permissions. Row-Level Security and Object-Level Security are fully enforced. The MCP server operates exclusively at the semantic model layer and has no access to data source credentials, Power Query transformations, connection strings, or the underlying data ingestion infrastructure. For local development, all communication stays on localhost. For cloud connections, all traffic is encrypted via TLS. User approval is required before the first modification or query in each session. Organizations can maintain full control through tenant administrator settings, least-privilege RBAC, audit logging, and network isolation.

9. References

- Microsoft Learn: What are the Power BI MCP servers? — https://learn.microsoft.com/en-us/power-bi/developer/mcp/mcp-servers-overview
- GitHub: microsoft/powerbi-modeling-mcp — https://github.com/microsoft/powerbi-modeling-mcp
- Microsoft Learn: Power BI Security — https://learn.microsoft.com/en-us/fabric/enterprise/powerbi/service-admin-power-bi-security
- Microsoft Learn: Row-Level Security (RLS) with Power BI — https://learn.microsoft.com/en-us/fabric/security/service-admin-row-level-security
- Microsoft Learn: XMLA Endpoint Connectivity — https://learn.microsoft.com/en-us/fabric/enterprise/powerbi/service-premium-connect-tools

- Microsoft Learn: Remote MCP Server — https://learn.microsoft.com/en-us/power-bi/developer/mcp/remote-mcp-server-get-started