A G E N T I C A I I N P O W E R P L A T F O R M

MODULE 01

The New Power Platform Vibe

Copilot-First, MCP-First

Duration	2 Hours (Conceptual – No Lab)
Module Type	Foundational / Mindset Setting
Prerequisites	Basic Power Platform Familiarity
Target Audience	EY Developers Working with BFSI Clients
Course Version	1.0 – February 2026

Module Overview

This opening module sets the strategic context for the entire 40-hour course. It is designed not to teach specific tools but to shift your mindset from traditional low-code application development to the new Copilot-first, MCP-first paradigm that is rapidly redefining how enterprise solutions are built on Power Platform.

By the end of this module, you will understand why Microsoft has repositioned Power Platform as an agentic AI platform, what the Model Context Protocol (MCP) is and why it matters, and how every component you will learn in subsequent modules fits together into a coherent, agent-driven architecture tailored for the Banking, Financial Services, and Insurance (BFSI) sector.

Learning Objectives

- **Articulate the paradigm shift** from traditional low-code app development to agentic AI systems on Power Platform.
- **Explain the Model Context Protocol (MCP)** and why it is replacing ad-hoc connectors as the standard integration layer for AI agents.
- **Describe Dataverse's triple role** as a system of record, agent memory, and MCP control plane.
- **Map the agentic architecture** showing how Power Apps, Copilot Studio, Power Automate, Power BI, and Claude Desktop work together.
- **Identify high-value BFSI use cases** where agentic AI delivers measurable business outcomes in banking, insurance, and financial services.
- **Navigate the course roadmap** understanding what each subsequent module delivers and how modules interconnect.

Module Agenda

Time	Topic	Duration
0:00	Welcome and Context Setting	15 min
0:15	The Evolution: From Low-Code to Agentic Systems	20 min
0:35	Understanding MCP: The Universal AI Integration Protocol	25 min
1:00	Dataverse as the Agent Platform	20 min
1:20	The Agentic Architecture Map	15 min
1:35	BFSI Industry Lens: Why This Matters for Financial Services	15 min

1:50	Course Roadmap and What You Will Build	10 min

Topic 1: Welcome and Context Setting

We are witnessing one of the most significant architectural shifts in enterprise technology since the move to the cloud. In 2025, Microsoft publicly declared that Power Platform is no longer just a low-code platform – it is an agentic AI platform. This is not marketing language; it reflects a fundamental change in how applications, data, and automation are designed, built, and operated.

1.1 The Age of AI Agents

The 2025 Release Wave 2 for Microsoft Power Platform introduced hundreds of new features centred around a single vision: AI agents are not add-ons; they are first-class citizens of the platform. AI agents in this context are autonomous software entities that can reason about tasks, access enterprise data, make decisions, and take actions – with or without human intervention.

This represents a departure from the traditional model where a human user opens an application, navigates screens, fills forms, and clicks buttons. In the agentic model, a user (or another system) describes an intent in natural language, and one or more AI agents orchestrate the entire workflow: querying data, applying business logic, routing approvals, and returning results.

> **Key Insight**
> Microsoft's 2025 Release Wave 2 announced that Power Platform governance and administration will become the unified governance hub for managing intelligent agents, agent-driven apps, and automated workflows across the entire Microsoft ecosystem. This signals that agents are not experimental – they are the new default.

1.2 What This Means for EY and BFSI

For EY developers working with Banking, Financial Services, and Insurance clients, this shift has immediate, practical implications. BFSI organizations are under intense pressure to reduce operational costs, comply with evolving regulations (KYC, AML, Basel, GDPR), deliver hyper-personalised customer experiences, and accelerate processes like loan origination, claims settlement, and fraud detection.

Traditional automation approaches – including Robotic Process Automation (RPA) and rule-based workflow engines – have delivered incremental improvements. But they are fundamentally limited: they follow rigid scripts, break when processes change, and cannot reason about exceptions. Agentic AI changes this equation entirely.

According to McKinsey, agentic AI is already driving productivity gains of 200 to 2,000 percent in compliance domains like KYC and AML by autonomously executing end-to-end workflows rather than merely assisting humans. Over half of mortgage lenders had deployed AI underwriting by end of 2025, with two-thirds planning expansion in 2026.

1.3 Course Philosophy

This course is designed around three core principles that will guide every module:

1. Copilot-First: Every solution you build should start with the question “Can an AI agent handle this?” before defaulting to a traditional app or workflow.
2. MCP-First: Every data integration should use the Model Context Protocol as the standard interface, rather than custom connectors or bespoke APIs.
3. Human-in-the-Loop: Agents are powerful, but BFSI requires guardrails. Every critical decision – credit approvals, AML escalations, customer communications – must have a defined point where humans review, approve, or override.

Instructor Note

Spend a moment asking participants about their current Power Platform experience. Gauge how many have built canvas apps, model-driven apps, Power Automate flows, and whether anyone has used Copilot Studio. This helps calibrate the depth of subsequent explanations.

Topic 2: From Low-Code to Agentic Systems

2.1 The Three Eras of Power Platform

To understand where we are going, it helps to understand where Power Platform has been. The platform has evolved through three distinct eras:

	Era 1: Low-Code (2018–2022)	Era 2: Copilot-Assisted (2023–2024)	Era 3: Agentic AI (2025+)
Primary User	Citizen developer	Maker with Copilot	Agent builder / orchestrator
Development Style	Drag-and-drop, form-based	Natural language + low-code	Describe intent; agents build + execute
Data Access	Connectors + custom APIs	Connectors + AI search	MCP protocol (standardised, universal)
Automation	Cloud Flows (trigger → action)	Copilot-assisted flows	Agent Flows (event-driven, autonomous)
Intelligence	Power BI dashboards	AI Builder + GPT models	Autonomous agents with reasoning
Dataverse Role	Database	Database + AI knowledge	Agent platform + memory + MCP server

2.2 What Changed in 2025

Several landmark announcements in 2025 crystallised this transition:

- **Microsoft Build 2025 (May):** Dataverse was officially positioned as “the secure, scalable agent platform that extends agents with enterprise data.” Microsoft announced Dataverse MCP Server, Common Knowledge Graph, Document Processor Agent, and the Dataverse SDK for Python – all designed to make Dataverse the backbone of multi-agent operations.
- **Dataverse MCP Server GA (November 2025):** The MCP server became generally available, enabling standardised, natural-language-driven access to Dataverse from any MCP client including Copilot Studio, Claude Desktop, GitHub Copilot in VS Code, and ChatGPT.
- **Power Apps Vibe Coding (November 2025):** A new paradigm where you describe what you need in natural language or provide an image, and the AI agent builds the entire data-connected application experience for you.
- **Copilot Studio Agent Flows GA (April 2025):** A new type of automation designed specifically for agent-driven workflows, replacing many scenarios previously handled by traditional Cloud Flows.

- **Microsoft Ignite 2025 (November):** Power Platform governance and administration was positioned as the unified hub for managing agents, agent-driven apps, and automated workflows at enterprise scale.

2.3 The Shift in Developer Mindset

For developers trained in the traditional Power Platform paradigm, this transition requires a fundamental shift in how you approach solution design:

Traditional Mindset	Agentic Mindset
"I will build a screen for users to enter loan applications"	"An agent will process loan applications via natural language, escalating exceptions to humans"
"I need a connector to pull data from Salesforce"	"Dataverse MCP will provide the agent with structured access to all enterprise data"
"I will create a Power Automate flow for each workflow step"	"An agent flow will autonomously orchestrate the entire workflow, calling tools as needed"
"Users navigate my app to find information"	"Users ask questions in natural language; the agent finds and presents the information"
"I will build a Power BI dashboard for reporting"	"An AI agent connected to Power BI via MCP generates real-time insights on demand"

Instructor Note

This is a good point for a brief group discussion. Ask participants: “Think about a recent project you delivered for a BFSI client. If you were starting that project today with agentic AI capabilities, what would you do differently?” Allow 3–5 minutes of discussion.

Topic 3: Understanding MCP – The Universal AI Integration Protocol

3.1 What is the Model Context Protocol?

The Model Context Protocol (MCP) is an open-source standard introduced by Anthropic in November 2024 that provides a universal, standardised way for AI systems such as large language models (LLMs) to connect with external data sources, tools, and services. It was donated to the Agentic AI Foundation under the Linux Foundation in December 2025, ensuring vendor-neutral governance.

Think of MCP as the USB-C of AI integration. Just as USB-C provides a single, universal connector for hardware devices regardless of manufacturer, MCP provides a single, universal protocol for AI agents to connect to any data source or tool regardless of platform.

MCP by the Numbers (as of late 2025)

Over 97 million monthly SDK downloads. More than 5,800 MCP servers and 300+ MCP clients available. Adopted by Anthropic, OpenAI, Google DeepMind, Microsoft, and Amazon. Governed by the Linux Foundation's Agentic AI Foundation. Market projected to grow from $1.2B to $4.5B by 2025.

3.2 The Problem MCP Solves

Before MCP, connecting an AI model to enterprise data required custom, bespoke integrations for every combination of AI model and data source. If you wanted your AI assistant to query Dataverse, read a SharePoint document, check a Salesforce record, and write to an SQL database, you needed four separate, custom-built connectors – each with its own authentication model, error handling, and data format.

This created what the industry calls the "M×N problem": M AI models multiplied by N data sources, each requiring unique integration code. MCP reduces this to M + N: each AI model implements the MCP client specification once, and each data source implements the MCP server specification once. Any client can then talk to any server.

Before MCP: The M×N Problem

- Every AI model needed a custom connector for every data source
- Each integration required its own authentication, error handling, and data formats
- Changes to one data source broke all its custom connectors
- No standardised way to describe what tools or data a source provides
- Security and governance had to be implemented per-connector

After MCP: The M+N Solution

- AI models implement the MCP client spec once and can talk to any MCP server
- Data sources implement the MCP server spec once and are accessible to any MCP client
- Standardised authentication, tool discovery, and error handling
- Tools and data sources are self-describing through the protocol
- Centralised security and governance patterns

3.3 MCP Architecture: Clients, Servers, and Hosts

The MCP architecture consists of three layers:

Component	What It Is	Examples
MCP Host	The application that provides the user interface and manages connections	Copilot Studio, Claude Desktop, GitHub Copilot in VS Code, ChatGPT
MCP Client	A protocol-level component within the host that maintains a 1:1 connection with a specific MCP server	Built into the host application; manages session lifecycle
MCP Server	A program that exposes specific data sources and tools to MCP clients via the standardised protocol	Dataverse MCP Server, Dynamics 365 Sales MCP Server, community MCP servers for Power BI, Slack, GitHub

3.4 MCP Capabilities: Resources, Tools, and Prompts

MCP servers expose three types of capabilities to clients:

- **Resources:** Read-only data that provides context to the AI model. Think of these as files, database records, or API responses that the model can read. Resources are identified by URIs and can be static or dynamic.
- **Tools:** Functions that the AI model can invoke to take actions. Tools accept structured input parameters and return results. The Dataverse MCP Server provides tools like list_tables, describe_table, read_query, create_record, update_record, and search_knowledge.
- **Prompts:** Reusable prompt templates that provide structured ways to interact with the server. These enable consistent, optimised interactions between the AI model and data sources.

Dataverse MCP Server – Built-In Tools

The Dataverse MCP Server provides the following built-in tools: list_tables (view all Dataverse tables and descriptions), describe_table (get detailed schema of a specific table), read_query (read records from a table with filtering), create_record (insert new records), update_record

(modify existing records), search_knowledge (search across knowledge sources), and execute_prompt (run pre-configured prompts). No SDKs, no custom APIs – just natural language and secure execution.

3.5 Why MCP Replaces Ad-Hoc Connectors

For Power Platform developers, this is a critical conceptual shift. In the traditional model, you would build a Custom Connector or use a pre-built Power Platform Connector (from the library of 1,400+) to integrate with external data. Each connector required configuration, credentials, and often custom code to handle data transformation.

With MCP, the integration model is fundamentally different:

Aspect	Traditional Connectors	MCP Protocol
Integration effort	Per-connector configuration	Single MCP server setup
Data access	Predefined operations	Natural language queries
Schema awareness	Manual mapping required	Agent discovers schema dynamically
Cross-platform	Platform-specific	Any MCP client to any MCP server
Intelligence	None – connector moves data	Agent reasons about data
Maintenance	Update each connector separately	Update MCP server once

Instructor Note

Emphasise that MCP does not eliminate the need for all connectors. There are still scenarios (high-volume data movement, real-time streaming, legacy system integration) where traditional connectors are appropriate. MCP is best suited for agent-driven, natural-language interactions with enterprise data.

Topic 4: Dataverse as the Agent Platform

4.1 Dataverse's Triple Role

Microsoft Dataverse has evolved from a simple database into what Microsoft calls "the secure, scalable agent platform." In the agentic AI era, Dataverse serves three distinct but interconnected roles:

Role 1: System of Record

Dataverse remains the enterprise data store for structured business data – accounts, contacts, transactions, cases, policies, claims, and any custom entities your BFSI solutions require. It provides relational data modelling, security at the row and column level, business rules, calculated and rollup fields, and the ability to connect to external data sources through virtual tables and the 1,400+ Power Platform connectors.

Role 2: Agent Memory

In multi-agent scenarios, agents need a shared, persistent store for context, conversation history, intermediate results, and coordination state. Dataverse now includes a managed vector index that enables large language models to access up-to-date, contextualised information from enterprise data. When one agent processes a customer's KYC documents and another agent handles their loan application, both agents share context through Dataverse – the customer's verified identity, risk profile, and document status are available to all agents in the workflow.

Role 3: MCP Control Plane

As an MCP server, Dataverse provides the standardised interface through which AI agents access, query, and modify enterprise data. The Dataverse MCP Server became generally available in November 2025, enabling any MCP client – whether it is a Copilot Studio agent, Claude Desktop, GitHub Copilot, or a custom application – to interact with Dataverse tables, records, and knowledge sources using natural language.

> **Practical Example: BFSI Loan Origination**
>
> Consider a loan origination scenario. Dataverse stores the loan application data (system of record). When a Copilot Studio agent processes the application, it writes intermediate assessment results back to Dataverse (agent memory). A separate fraud detection agent reads the same data via MCP to cross-check against known fraud patterns (MCP control plane). A human loan officer reviews the agent's recommendation through a Power App that surfaces the data stored in Dataverse. The entire workflow is orchestrated through agents, with Dataverse as the single source of truth.

4.2 The Dataverse MCP Server in Practice

The Dataverse MCP Server works across multiple MCP hosts. Here is how it functions in each:

In Copilot Studio

Adding Dataverse MCP Server as a tool to your Copilot Studio agent is as simple as selecting it from the Tools tab. No local setup is required. Once connected, your agent can process user queries like “What is the status of support ticket #12345?” by automatically invoking the appropriate Dataverse MCP tools: determining the right table via list_tables or describe_table, then reading data with read_query.

This is a compelling alternative to the traditional approach of building custom Dataverse connector flows to store, update, or retrieve data. Makers can now build dynamic, intelligent workflows with conversational agents that interact directly with Dataverse – no flows, no APIs, just prompts and tools.

In Claude Desktop

With a configuration update, Claude Desktop can connect to your Dataverse environment. Business users can request operations like “Create a dashboard showing opportunities by region” and Claude will identify the relevant Dataverse tables, query the data, and generate an interactive visualisation – all through MCP.

In GitHub Copilot (VS Code)

Developers can interact with Dataverse directly from their IDE. Commands like “Store the attached CSV file as Employee in Dataverse” trigger GitHub Copilot to inspect the data and upload it into the appropriate table – all through MCP tools, without opening the Power Platform portal.

4.3 Common Knowledge Graph

Copilot Studio’s Common Knowledge Graph, tightly integrated with Dataverse, dramatically simplifies one of AI’s key bottlenecks: knowledge ingestion and index management. Enterprises can ingest data from diverse sources – files, Microsoft 365, Salesforce, ServiceNow, SAP, and more – to create vector embeddings for deep retrieval-augmented generation (RAG) capabilities. They can also index metadata from sources such as Azure SQL, Zendesk, Snowflake, and Oracle for fast, granular queries over structured enterprise data.

> ***Instructor Note***
>
> *This is a good point to preview Module 04 (Dataverse MCP Deep Dive). Let participants know that in Module 04 they will get hands-on experience connecting Dataverse as an MCP server to Copilot Studio and exploring the built-in tools. For now, the goal is to understand the concept and architecture.*

Topic 5: The Agentic Architecture Map

5.1 How the Components Fit Together

The agentic Power Platform architecture consists of five major component layers, each of which maps to specific modules in this course. Understanding how they interconnect is essential before diving into the individual technologies.

Layer	Component	Role in Architecture	Course Module
Design	Plan Designer	Describe problem; agents design the solution	Module 02
Build	Power Apps Vibe / Agent Builder	Create apps through natural language; embed agents in apps	Modules 03, 05
Intelligence	Copilot Studio Agents	Build conversational AI agents with knowledge, tools, and reasoning	Module 06
Data	Dataverse + MCP Server	Enterprise data store, agent memory, and MCP interface	Module 04
Automation	Agent Flows + Power Automate	Event-driven autonomous workflows; human-in-the-loop approvals	Modules 07, 08
Orchestration	Multi-Agent Patterns	Multiple specialised agents coordinating through shared data	Module 09
Analytics	Power BI + Claude Desktop MCP	AI-driven analytics and reporting via MCP connections	Modules 10, 11
Governance	Admin Centre + Purview + Sentinel	Security, auditing, compliance, and agent lifecycle management	Module 12

5.2 The Agent Workflow Pattern

Across all BFSI scenarios you will encounter in this course, the fundamental agent workflow follows a consistent pattern:

4. User Intent: A user (or external system) expresses an intent in natural language – for example, “Process this loan application” or “Check the status of claim #4521.”

5. Agent Reasoning: The Copilot Studio agent interprets the intent, determines which tools and data sources are needed, and creates an execution plan.
6. MCP Data Access: The agent uses the Dataverse MCP Server to discover relevant tables, query data, and retrieve contextual information without any hardcoded connector logic.
7. Tool Execution: The agent may invoke additional tools – calling an Agent Flow for complex business processes, triggering approvals through Power Automate, or querying external systems.
8. Human-in-the-Loop: For decisions that require human judgment (credit approvals above a threshold, suspicious transaction reviews, regulatory sign-offs), the agent pauses and routes to a human approver.
9. Response and Learning: The agent returns results to the user, logs the interaction in Dataverse (agent memory), and updates any relevant records.

5.3 Where Claude Desktop and External MCP Clients Fit

While Copilot Studio is the primary agent-building platform within Power Platform, this course also covers Claude Desktop as an external MCP client in Module 10. This is significant for two reasons:

- **Platform Agnosticism:** MCP's open standard means your Dataverse data is accessible not just to Microsoft tools but to any MCP-compliant AI client. This future-proofs your enterprise data investment.
- **Power BI Analytics:** Claude Desktop can connect to Power BI data models through community MCP servers, enabling natural-language-driven analytics that complement Power BI's traditional dashboard experience.

Topic 6: BFSI Industry Lens

6.1 Why Agentic AI Matters for Financial Services

The BFSI sector is uniquely positioned to benefit from agentic AI because of the combination of high-volume, rule-heavy processes; stringent regulatory requirements; massive data volumes; and intense competitive pressure to improve customer experience. Agentic AI addresses all four of these simultaneously in ways that traditional automation cannot.

The Pressure Points

- **Compliance Costs:** Global financial institutions spend billions annually on KYC, AML, and regulatory reporting. Agentic AI can automate end-to-end compliance workflows while maintaining full audit trails – a critical requirement for regulators.
- **Fraud Losses:** Financial fraud continues to escalate globally. AI agents can analyse millions of transactions in real time, detecting anomalies and preventing fraudulent transactions far faster than rule-based systems.
- **Customer Expectations:** Banking and insurance customers now expect instant, personalised, 24/7 service. Agentic AI delivers conversational interactions that can handle complex queries – from account enquiries to claims status to investment advice.
- **Operational Efficiency:** Processes like loan origination, policy issuance, and claims settlement involve multiple handoffs, document reviews, and approval chains. Agents can orchestrate these end-to-end, reducing cycle times from days to minutes.

6.2 High-Value BFSI Use Cases for Agentic AI on Power Platform

Throughout this course, you will build solutions aligned to these real-world BFSI scenarios:

Use Case	What the Agent Does	Business Impact
Loan Origination	Collects documents, checks eligibility, analyses credit scores, recommends approval paths, routes exceptions to human underwriters	Reduces turnaround from days to hours; ensures policy compliance
KYC Verification	Continuously updates and verifies customer identity data, identifies expired documents and suspicious patterns, ensures onboarding compliance	Cuts manual KYC effort by up to 80%; enables real-time compliance
Claims Processing	Extracts information from claim documents, validates against policy terms, assesses damage, routes for settlement approval	Reduces claims settlement time; improves customer satisfaction

Fraud Detection	Monitors transactions in real-time, flags anomalies, cross-references against known fraud patterns, escalates suspicious activity	Prevents financial losses; reduces false positives
Regulatory Reporting	Monitors regulatory changes, auto-fills filing templates with validated data, ensures audit readiness	Ensures compliance with FATCA, CRS, Basel requirements
Investment Advisory	Processes market data, customer behaviour, and portfolio performance to provide proactive recommendations	Deepens client relationships; increases advisory revenue

6.3 The Human-in-the-Loop Imperative for BFSI

One critical lesson that runs throughout this course: in BFSI, agentic AI must always operate with clear human-in-the-loop boundaries. Financial regulators require that certain decisions – credit approvals, AML escalations, customer-facing communications, investment recommendations – have documented human oversight.

This is not a limitation of the technology; it is a design principle. Module 08 (Human-in-the-Loop Workflow Design) is dedicated entirely to designing these boundaries effectively, ensuring that agents handle the workflow while humans retain decision authority at critical junctures.

BFSI Design Principle

An agent should always be able to answer the question: “At what point in this workflow does a human need to review, approve, or override my decision?” If the agent cannot answer this question, the workflow design is incomplete.

Topic 7: Course Roadmap – What You Will Build

7.1 Module-by-Module Overview

This course comprises 13 modules delivered across 40 hours. Each module builds on the previous ones, culminating in a capstone project that integrates everything into a complete BFSI solution.

#	Module Title	Duration	Type
01	The New Power Platform Vibe (This Module)	2 hrs	Conceptual
02	Plan Designer: From Business Problem to Solution Blueprint	3 hrs	Theory + Lab
03	Power Apps Vibe: Natural Language App Creation	3 hrs	Theory + Lab
04	Dataverse MCP: Connecting Agents to Enterprise Data	3 hrs	Theory + Lab
05	Agent Builder: Embedding Intelligence in Canvas Apps	3 hrs	Theory + Lab
06	Copilot Studio: Building Intelligent Agents	4 hrs	Theory + Lab
07	Agent Flows: Event-Driven Autonomous Workflows	3 hrs	Theory + Lab
08	Human-in-the-Loop Workflow Design	3 hrs	Theory + Lab
09	Multi-Agent Orchestration Patterns	3 hrs	Theory + Lab
10	Claude Desktop + MCP for Power BI Analytics	3 hrs	Theory + Lab
11	Power BI Agent-Driven Analytics	2.5 hrs	Theory + Lab
12	Enterprise Governance for Agentic Solutions	2.5 hrs	Theory + Lab
13	Capstone Project: End-to-End BFSI Solution	5 hrs	All Lab

7.2 The Progressive Build Pattern

The modules are designed so that each builds on the previous. By the end of the course, you will have constructed a complete, production-ready BFSI solution:

10. Modules 01–03 establish the foundation: understanding the paradigm, designing solutions with Plan Designer, and creating apps with Vibe coding.
11. Modules 04–06 build the technical core: connecting Dataverse via MCP, embedding agents in apps, and constructing intelligent Copilot Studio agents.
12. Modules 07–09 add enterprise capability: autonomous workflows, human oversight patterns, and multi-agent coordination.

13. Modules 10–11 extend into analytics: leveraging Claude Desktop and Power BI for agent-driven insights.
14. Module 12 ensures governance: security, auditing, compliance, and lifecycle management for agentic solutions.
15. Module 13 integrates everything into a capstone project that demonstrates a complete BFSI scenario end-to-end.

7.3 What to Expect Next

In Module 02, you will begin working with Plan Designer – Power Platform's AI-driven solution design tool. You will describe a business problem in natural language, and a team of AI agents will help you design an enterprise solution that includes apps, agents, Power BI reports, and more. This is where the theory from Module 01 begins to become practice.

> ***Instructor Note***
>
> *Close the module by recapping the three principles: Copilot-First, MCP-First, and Human-in-the-Loop. Ask if there are any questions. Remind participants that the next module (Plan Designer) will be their first hands-on experience with the new paradigm.*

References and Further Reading

Official Microsoft Sources

- Microsoft Power Platform 2025 Release Wave 2 Overview – learn.microsoft.com/en-us/power-platform/release-plan/2025wave2/
- Overview of Microsoft Dataverse 2025 Release Wave 2 – learn.microsoft.com/en-us/power-platform/release-plan/2025wave2/data-platform/
- Dataverse at Build 2025: The Agent Platform – microsoft.com/en-us/power-platform/blog/2025/06/03/dataverse-at-build-2025/
- Dataverse MCP Server: A Game Changer for AI-Driven Workflows – microsoft.com/en-us/power-platform/blog/2025/07/07/dataverse-mcp/
- Connect to Dataverse with MCP – learn.microsoft.com/en-us/power-apps/maker/data-platform/data-platform-mcp
- Overview of Microsoft Copilot Studio 2025 Release Wave 2 – learn.microsoft.com/en-us/power-platform/release-plan/2025wave2/microsoft-copilot-studio/
- Overview of Power Apps 2025 Release Wave 2 – learn.microsoft.com/en-us/power-platform/release-plan/2025wave2/power-apps/
- Overview of Power Automate 2025 Release Wave 2 – learn.microsoft.com/en-us/power-platform/release-plan/2025wave2/power-automate/

MCP Protocol Sources

- Anthropic: Introducing the Model Context Protocol – anthropic.com/news/model-context-protocol
- MCP Official Specification (November 2025) – modelcontextprotocol.io/specification/2025-11-25
- Wikipedia: Model Context Protocol – en.wikipedia.org/wiki/Model_Context_Protocol
- IBM: What is Model Context Protocol (MCP)? – ibm.com/think/topics/model-context-protocol
- MCP Enterprise Adoption Guide – guptadeepak.com (comprehensive 2025 analysis)

BFSI and Agentic AI Industry Sources

- Deloitte Insights: Agentic AI in Banking – deloitte.com/us/en/insights/industry/financial-services/agentic-ai-banking.html
- Moody's: Agentic AI in Financial Services – moodys.com/web/en/us/creditview/blog/agentic-ai-in-financial-services.html

- BAI: The Transformative Power of Agentic AI in Financial Services – bai.org/banking-strategies/
- McKinsey: Agentic AI Productivity Gains in KYC/AML Compliance (referenced in multiple industry analyses)

—— End of Module 01 ——