MS GenAI Enterprise Architecture Brief

This is very preliminary research on how MS is looking at Enterprise GenAI architecture.

An attempt at conceptualizing it.

# Basics

1. The Primary UXs are: Copilot chat, office, Teams
2. There are default data sets that are fully exposed to this chat like OneDrive, SP etc. Admins control what one can see or not, otherwise it is all
3. When a user chats in the Work context, Copilot quietly grounds on Microsoft Graph (SharePoint Online, OneDrive, Exchange, Teams, etc.) plus any external content your admins have indexed via Copilot/Graph connectors—all filtered by that user’s permissions. There’s no per-chat “pick a connector” UI for end users.
4. A special Graph Connector Agent can semantically index on premise data to be part of the search index. Index is stored in the cloud.
5. Agents get involved when external content needs to be read or written to

# Agent designs, connectors, MCPs

1. Copilot studio is used to create agents
2. Agents use many MS internal APIs to write connectors (MS Graph API, Power Platform etc)
3. OpenAPI (Swagger) is a key standard
4. uses MCP recently when available
5. MCPs are not directly exposed to chats
6. They go through Agents
7. Once defined and deployed by Copilot studio they are exposed to UXs
8. There is NO SEPARATE UX for agents or MCPs
9. You can only talk to 1 agent. But that can be an orchestrator that talks to other agents
10. Agents can be long running
11. Agents mostly in English
12. Configured Custom UX through MS Teams Message Extensions (Via modal controls for input and custom cards for output)

# Fabric Data Agent: An Example of dealing with SQL

1. A special agent for Lakehouse, tables, kql etc on fabric
2. Created in a Fabric workspace, with sources selected from the OneLake catalog
3. Limited to a set of tables and schemas selected
4. Direct answers based on the schema based on structured data
5. NL to SQL kind of a deal

# Bottom line for Enterprise architecture

1. Horizontal UXs: Excel, Word, Teams, Chat
2. Prebuilt RAG (Semantic indexing)
3. Extensions through Agents
4. Agents, now are heavily influenced by MS native connectors
5. MCPs are exposed through Agents
6. Chat is a bit weak: for example, such concepts as Projects, Canvases, etc.
7. Licensing at the moment is tilted towards per user (70%) while metering for GenAI may be 30%