



Microsoft Teams Application
Development Accelerator

Week 4: Class

Conversational Chatbots - with & without LLMs

October 15, 2025



Agenda

- **Overview of chatbots in Microsoft Teams solutions – with & without LLMs**
- **Historical context – libraries & SDKs**
- **Development & deployment**

Overview of Chatbots in Microsoft Teams



Overview of Chatbots in Teams

CHATBOT EXPERIENCES (UX & DEVX) HAVE RAPIDLY EVOLVED

- **Initially chatbots were implemented purely with imperative code**
 - Users had to explicitly pick options or use specific strings to command the chatbot
- **Gradually, natural language (NL) tools improved the experiences**
 - For users & developers
- **The rise of LLMs has changed the game**
 - Organizations (users) & developers have moved to LLMs
- **Present day: default to LLMs to improve...**
 - ... user interaction & satisfaction
 - ... developer productivity

Chatbot Implementation Options

Imperitive/Explicit Code

- **Required developers to explicitly map out conversation paths**
 - i.e. automated call systems
- **Challenge to implement complex conversation scenarios**
 - Inflexible
 - Unpredictable user behavior
- **Challenging user experience**

LLMs

- **LLMs improve efficiency in determining user intent**
- **Excel at taking instructions...**
 - prompt engineering
 - skills/functions/tools
- **... & navigat conversation paths for unpredictable users**
- **... & collecting additional data based on conversation flow**

How Chatbots Work

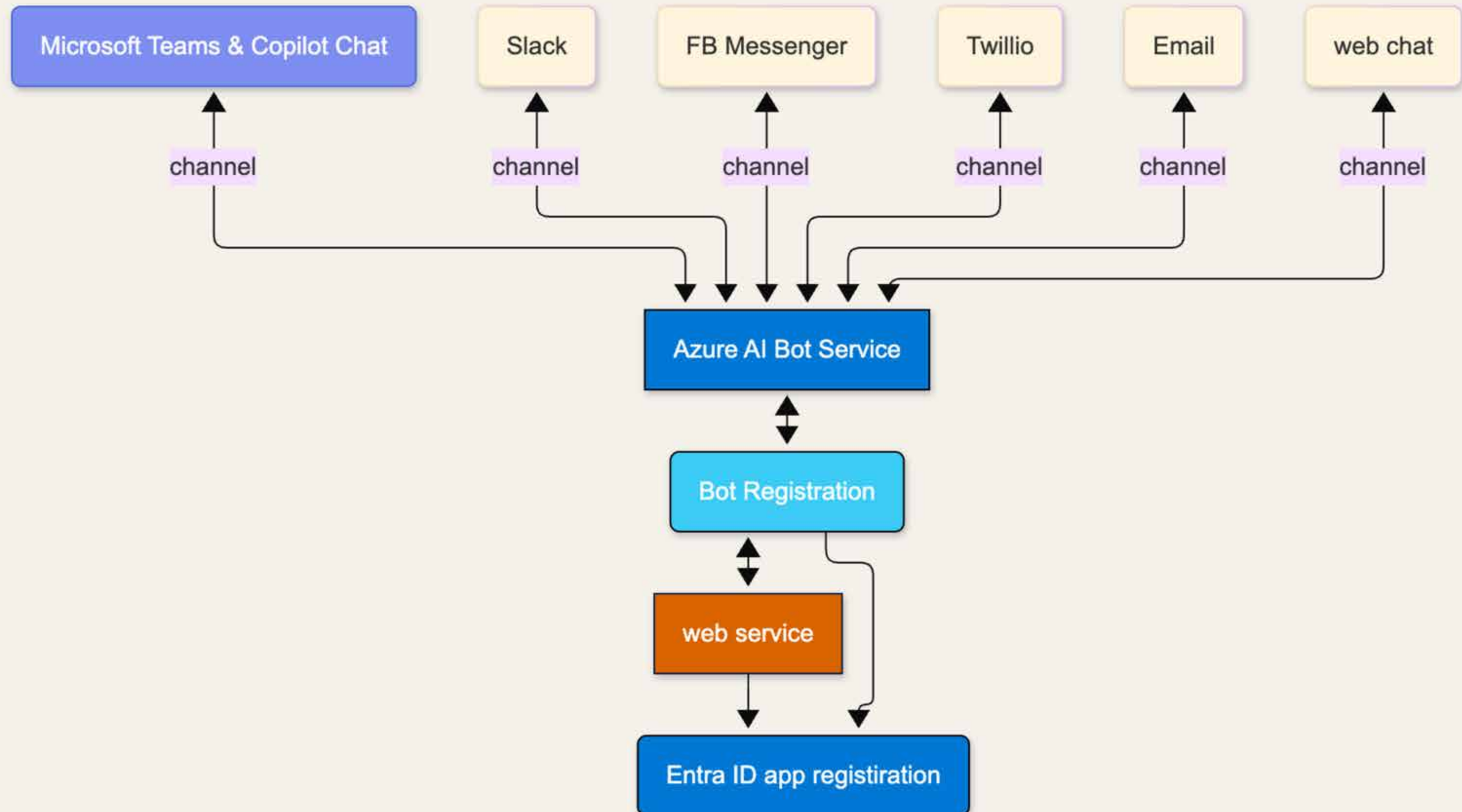
Chatbot Concepts

- **Activity:** every interaction between user & bot
- **Turn:** user's incoming activity & immediate response bot sends back
- **Adapter:** handles channel connectivity (Teams/Messenger/SMS)
 - Provides method for handling requests from + methods for generating requests to user
 - Includes middleware pipeline that includes turn processing outside of bot turn handler
 - Calls bot turn handler & catches errors not otherwise handled
- **Middleware:** linear set of components executed in order
 - Each component has chance to operate on activity BEFORE & AFTER turns
- **State:** same concept as with websites
 - Allows bot to “remember” more than just single turn with user

Chatbot State

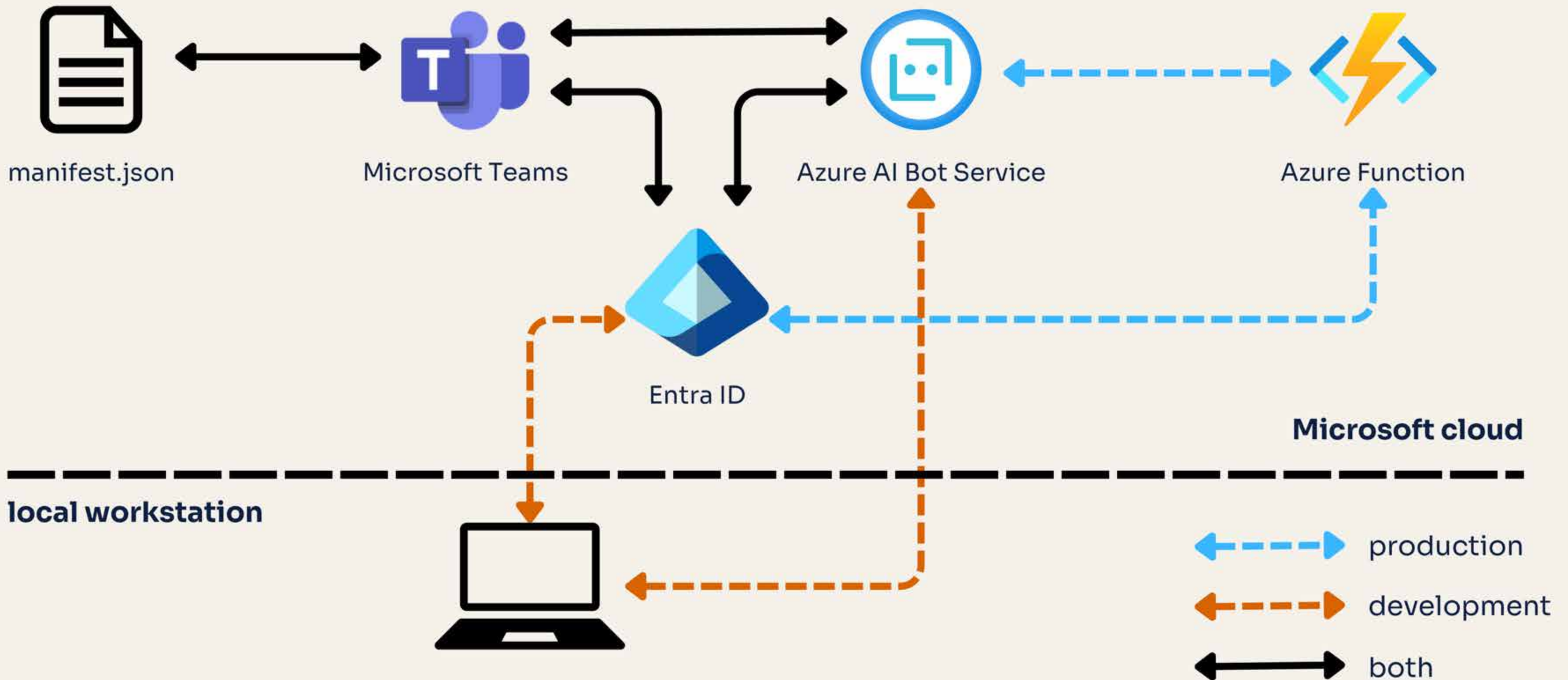
- **Various SDKs provide storage layers & state management abstractions**
- **User state:** avail in any turn the bot is conversing with that user on that channel, regardless of convo
 - IDEAL for... name & preferences / info about last conversation they had with bot
- **Conversational state:** avail in any turn in a specific conversation, regardless of user, such as group conversations
 - IDEAL for... did bot ask the user a question & what the question was / that the current topic of the conversation is, or last question was
- **Private conversation state:** scoped to both the **specific conversation** & to that specific user
 - IDEAL for... support group convos but when you want to track BOTH the user & CONVO specific info

Microsoft Chatbot Communication Flow



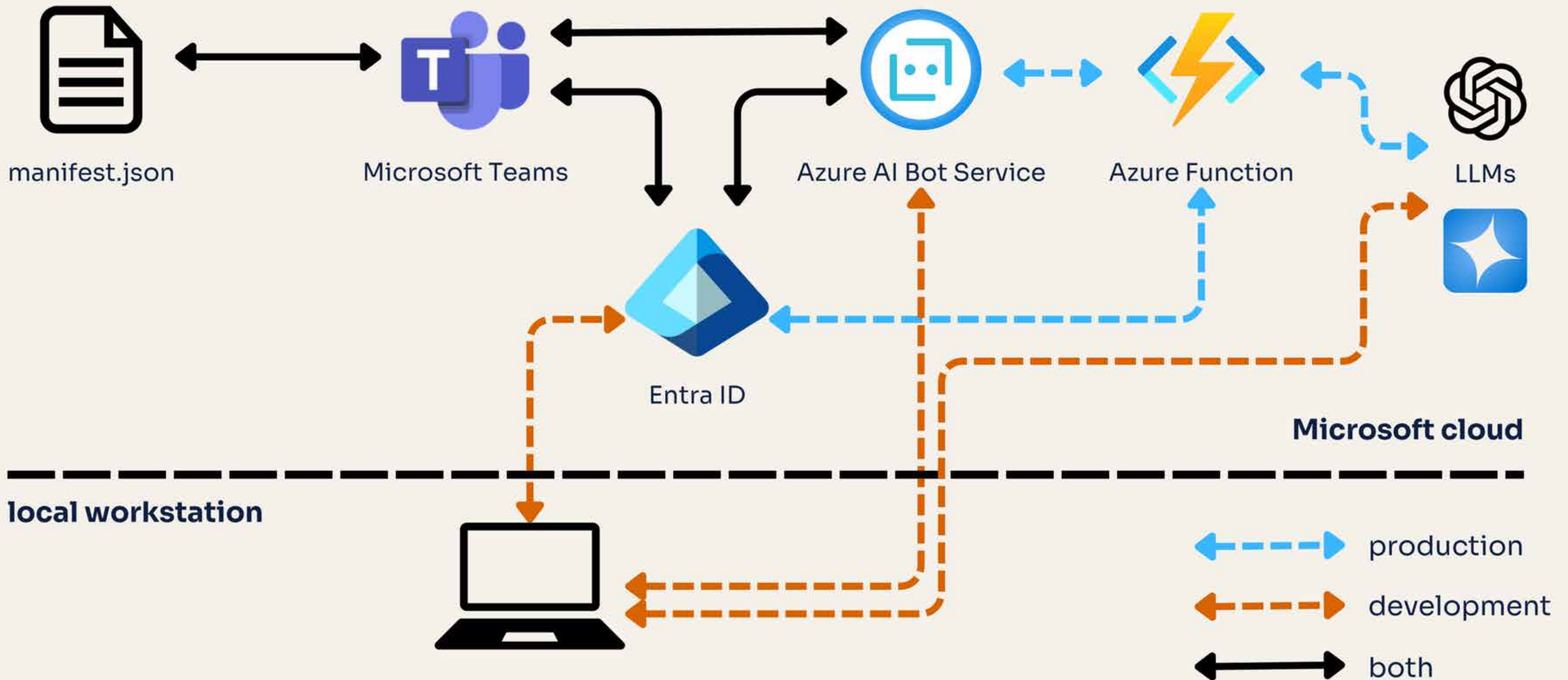
Microsoft Teams Chatbots

DEVELOPING WITH THE VARIOUS SDKS - WITHOUT AI/LLMS



Microsoft Teams Chatbots

DEVELOPING WITH THE VARIOUS SDKS - WITHOUT AI/LLMS



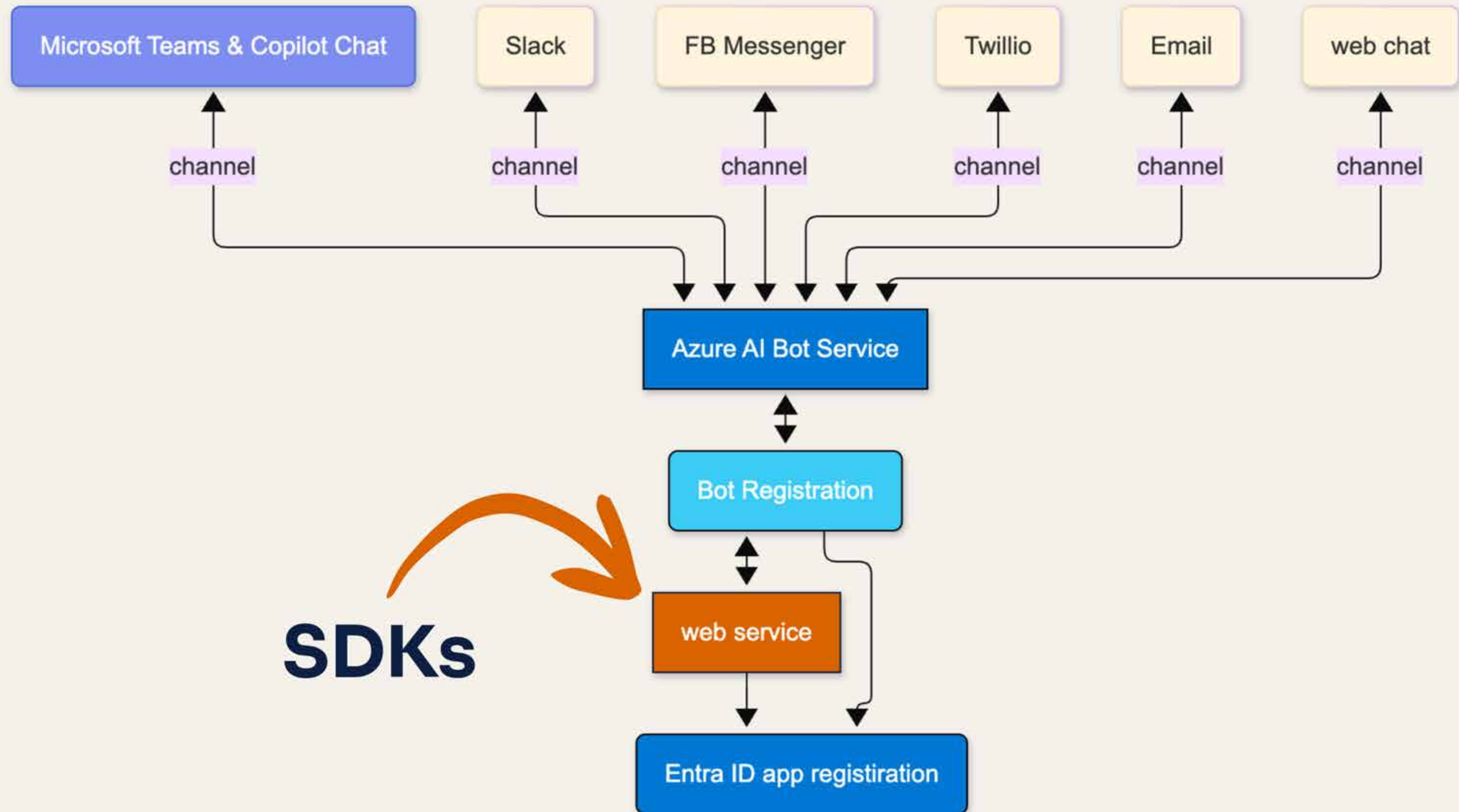
Historical Context Libraries & SDKs



Developer Tools, Libraries, & SDKs

- **TeamsFx**
 - Released H2 2020, now deprecated, & will be retired JULY 2026
- **Bot Framework v4**
 - Released September 2018, will be retired end of 2025
- **Teams AI Library v1**
 - Released September 2023, now deprecated
- **Microsoft 365 Agents SDK** (.NET, JS/TS, Python)
 - Released May 2025
- **Teams AI SDK v2** (.NET, JS/TS, Python)
 - Released September 2025

Microsoft Chatbot Communication Flow



Which SDK to Use?

- **Microsoft 365 Agents SDK**

- Chatbots / agents that will work with various channels supported by the Azure AI Bot Service
- Not just used in Teams & doesn't include Teams specific UX
- **Think:** Bot Framework v5

- **Teams AI SDK v2**

- Chatbots / agents that will be used only in Microsoft Teams or Microsoft 365 Copilot Chat

Development & Deployment



Developer Tools & Concepts

- **Services**

- Azure AI Bot Service vs. Bot Framework Dev Portal

- **Routing from Microsoft Teams backend => localhost in dev**

- ngrok & Dev Tunnels
- Issue for secured environments in development

- **Tools**

- Microsoft 365 Agents Playground
- DevTool (from Teams AI SDK v2)

Local, Deployed, Entra ID Apps, & Managed Identities

- **ATK bot-based templates use Azure User Assigned managed identities (UserAssignedMSI) for bot identity in prod**
 - No longer requires an Entra ID app to define the bot identity & code auth
 - **Benefits:**
 - Grant Azure resource access to the bot
 - Code running in the Azure resource assumes the identity of resource
 - Thus, no password needed
 - “The Azure Function has access to the bot registration in the Azure AI Bot Service”
- **Doesn't impact SSO**
 - Entra ID app still required to assign permissions, scopes, consent, etc
- **Complicates the running locally vs. deployed scenario**
- **M365 Agents SDK smart enough to detect the “bot type” at runtime**
 - **Bot Type:** single/multi-tenant vs. UserAssignedMSI

Week 4 Homework



Week 4

Assignments

- **Chatbots without LLMs**
 - Create a simple bot
 - Create a notification bot
- **Teams AI v2**
 - Create & obtain the credentials to use a LLM hosted by OpenAI or **Azure OpenAI**
 - Create a simple custom engine agent (bot) with commands – light bot