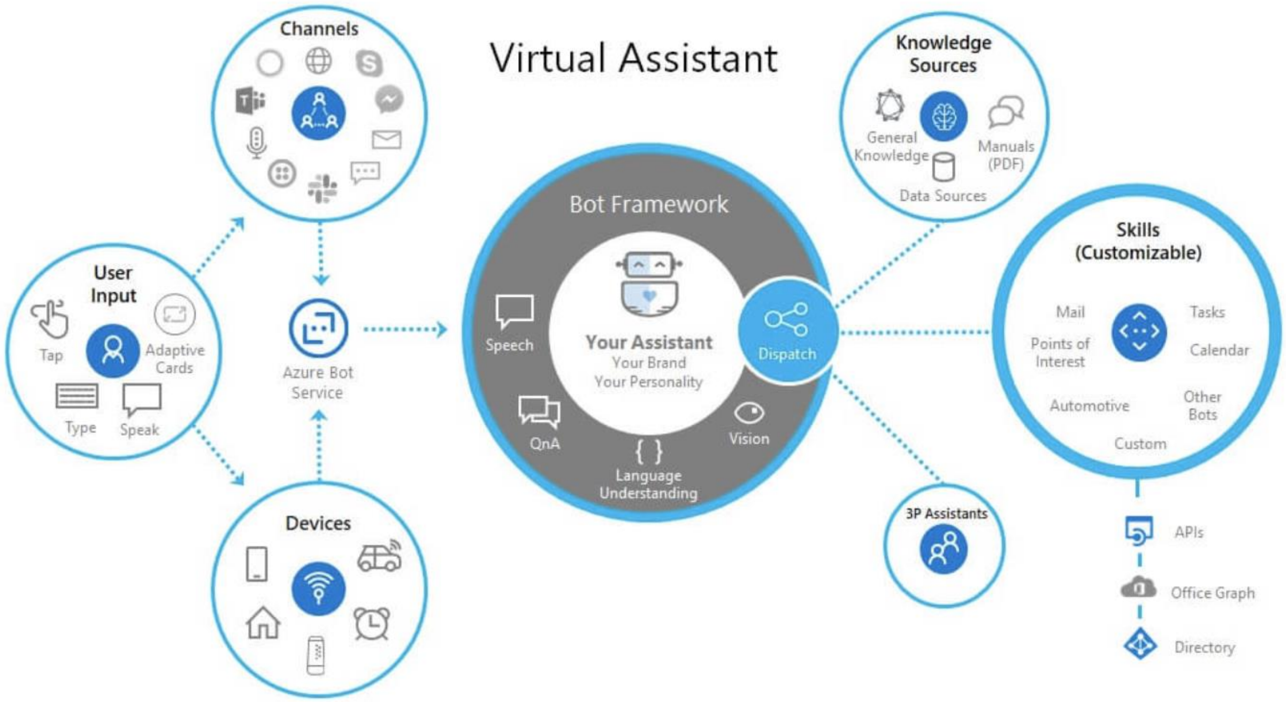
CS410

Tech Review: Azure Bot Service



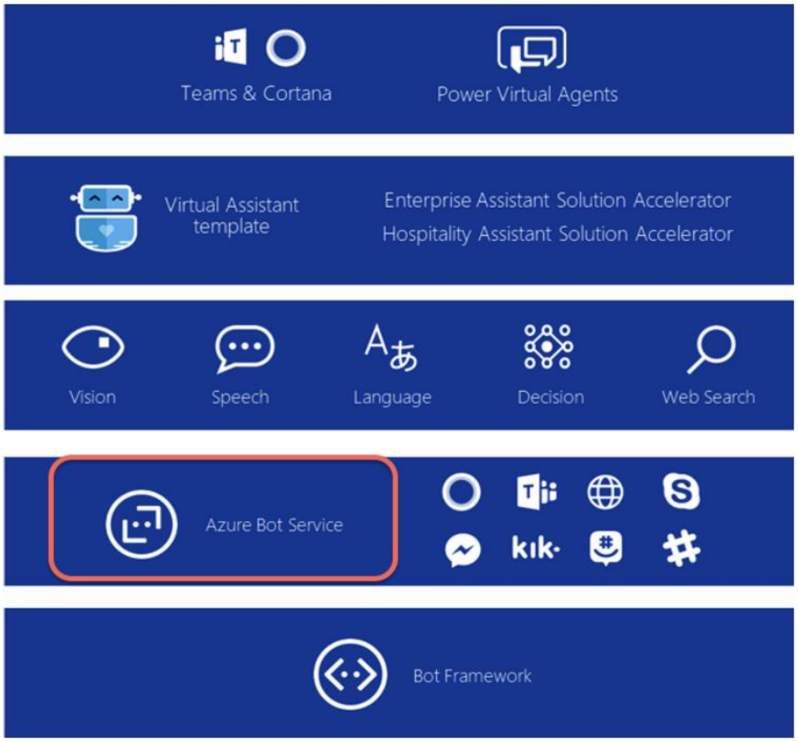
Introduction

Azure Bot Service is a part of Microsoft's Azure AI platform that aims to bring AI to every "developer" to build, manage, and deploy conversational AI. This service is

- Open-source
- Cloud-based
- Cross-platform
- "Developer-first" - may not be easy for users with non-computer-science knowledge.
- Built on top of Microsoft Bot Framework SDK (latest v4.9)
- Allows virtual assistance personalization
- Allows developers to have ownership and control their data.

While the algorithmic implementation details of the Azure Bot service are unknown except for that they rely on the LUIS engine the following features and capabilities are reviewed. In the end, the comparison table across Azure Bot Service, Wit.ai, and Dialogflow gives better insights on what to choose based on the channel.

A glimpse of Microsoft Bot Service Ecosystem


Teams & Cortana
Power Virtual Agents
Virtual Assistant template
Enterprise Assistant Solution Accelerator
Hospitality Assistant Solution Accelerator
Vision
Speech
Language
Decision
Web Search
Azure Bot Service
kik
Bot Framework

An Azure bot life cycle

1. Design:

Designing an Azure bot requires

- researching targeted users,
- defining bot personas (should it have a name, gender, tone, etc ?),
- storyboarding bot scenarios (rulebook),
- designing conversation flow (basic QnA to complex interaction), and

- defining an evaluation plan

A careful design makes sure that the bot service navigation doesn’t fall into the following pitfalls personalities:

- the stubborn bot - needs the exact expected answer to process the user’s question,
- the clueless bot - doesn’t know how to respond to a new query,
- the mysterious bot - doesn’t indicate any state of the user input processing,
- the captain obvious bo - provides unasked obvious information, and
- the bot that can't forget - provides answers from an irrelevant knowledge base.

2. Build:

Choose either building from scratch or on top of templates for local development that support the C#, JavaScript and Python languages with the following features:

- The Language Understanding (LUIS) enables NLP capabilities.
- QnA Maker adds a knowledge base to answer common questions.
- A Dispatch tool manages complex conversation flows and multiple knowledge domains.
- Adaptive Cards add graphics or menus.
- A command-line tool is provided to create, manage, and test these bots.

3. Test:

The Bot Framework Emulator enables developers to test conversations through the unit and early end-user tests using the Bot Framework SDK, which can focus on functionality testing of specific dialogs.

4. Publish:

The Azure bot can be hosted on Azure or to any web service or data center.

5. Connect:

Azure Bot Service can be configured to work on a range of channels and devices:

Channels	Platforms
Facebook Messenger	Desktop applications
Slack	Mobile apps
Microsoft Teams, Cortana, Email, Telegram, Twilio, LINE, etc.	Mobile devices such as cars, alarms, speakers, clocks, etc.

- Direct Line channel (in build 2020) connects bot to any client application.
- Direct Line Speech channel enables low-latency speech interfaces.

6. Evaluate:

Bot Framework and a Power BI dashboard are provided to query bot’s conversations with users and gain key insights into your bot’s health and behavior.

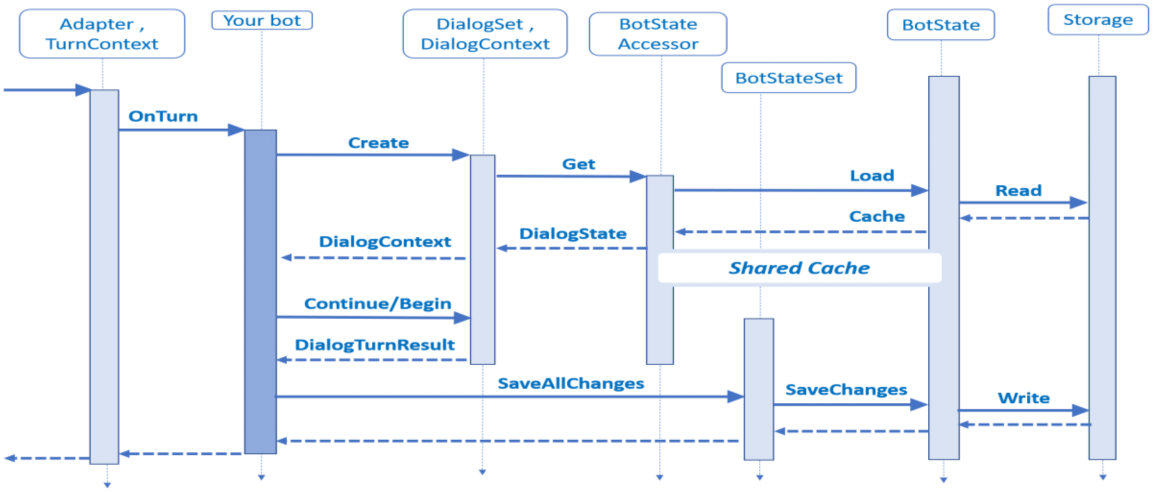
An Azure Bot State and Storage

A bot state represents certain things saved about a user or conversation. It is saved for more than the current turn time so that it keeps information over the course of a multi-turn conversation.

State	Availability				Used for tracking…
	Turns	Users	Channels	Sessions	
User	All	Specific	Specific	All	Conversation context
Conversation	All	All	Specific	Specific	User information

Private Conversation	All	Specific	Specific	Specific	User + Conversation context

A bot state is used in the storage layer, state management, and state property accessors. The storage layer can be any physical storage such as the in-memory, Azure (Blob storage, Cosmos DB or Azure Table), or any third party server.



A comparison of various bot services

	Facebook’s wit.ai	Google’s Dialogflow API.ai	Microsoft Bot Service
Model	User-defined Entity/Intent	System + User defined Entity/Intent	User-defined Entity Intent
Cross-platform	No, limited to FB Messenger	Yes	Yes
Handles simple queries?	Yes	Yes	Yes
Handles complex queries?	No	No	Yes

Pros	Uses ML to discover important bits of info	Supports 14+ Languages	Ease of use for developers
Cons	Unpredictable training period for complex queries	Still under Beta phase with many missing implementations	limited to Node.JS, C#, and Python development platforms

References

1. Azure Bot Service https://azure.microsoft.com/en-us/services/bot-service/
2. Build 2020 - Conversational AI updates https://techcommunity.microsoft.com/t5/azure-ai/build-2020-conversational-ai-updates/ba-p/1397685
3. A developer’s guide to building AI Applications https://clouddamcdnprodep.azureedge.net/gdc/1657747/original
4. Google Dialogflow https://cloud.google.com/dialogflow/docs/
5. Facebook Wit.ai https://wit.ai/