CopilotBuilder Security and Data Privacy

Introduction

One of the driving factors behind building CopilotBuilder is to provide the means for users and organizations to securely reference their own documents with generative AI. While tools like ChatGPT have access to uploaded files (and might even train models on them), CopilotBuilder does not give AI models access to files uploaded by its users. This document outlines how CopilotBuilder handles data privacy as a platform and with uploaded documents in detail.

Privacy in the Platform

CopilotBuilder was built from the ground-up with data privacy in mind:

- Models do not retain messages from users or responses from the models themselves
- Models will not train themselves on interactions with Copilots
- Models do not have access to user-uploaded files or their contents
- CopilotBuilder does not share its user data
- CopilotBuilder does not run analytics on user data
- CopilotBuilder will not access your data in any way outside of support requests, and even then, only with your permission
- CopilotBuilder has enterprise-level privacy agreements with all model vendors which have strict guidelines (that consumer-level products do not have). These agreements further enforce that model vendors will not retain your information or use prompts/responses for training purposes.

Privacy and Uploaded Documents

Large language models (LLMs) are trained on publicly available datasets out-of-the-box. This means that if you want to ask questions from an AI model based on your own data, which it does not have, you must provide it. You can use the natural language processing (NLP) capabilities of LLMs with your own files with a process called Retrieval Augmented Generation (RAG). CopilotBuilder achieves this in a way that makes sure your data stays private. Let’s explore this process.

File Upload

CopilotBuilder allows users to upload the following types of documents as a custom data source for Copilots:

- Word
- Excel
- PowerPoint
- Plain text
- CSV
- Images (JPG, GIF, PNG)

- Markdown
- JSON

In CopilotBuilder, your uploaded files are grouped into Data Collections. Copilots can refer to up to three (3) data collections.

The uploaded document is securely stored and encrypted in an Azure Storage container where it can only be accessed by CopilotBuilder. We ensure this via the use of a "Storage Account Firewall" where there is no public access to the storage account due to using a storage account private endpoint (ie, non-routable IP address).

Embedding

AI models require that the content of files be transformed into data vectors. These vectors are stored and encrypted in a QDrant vector store. Furthermore, each organization in CopilotBuilder has its own container within the QDrant vector store. This means that each organization's data is isolated. AI models do not have access to this vector store.

Per the security features we have via the Storage Account Firewall, access to our QDrant data cluster MUST have an API key to validate access to the cluster, and that API key is only available to our CopilotBuilder platform

Retrieval

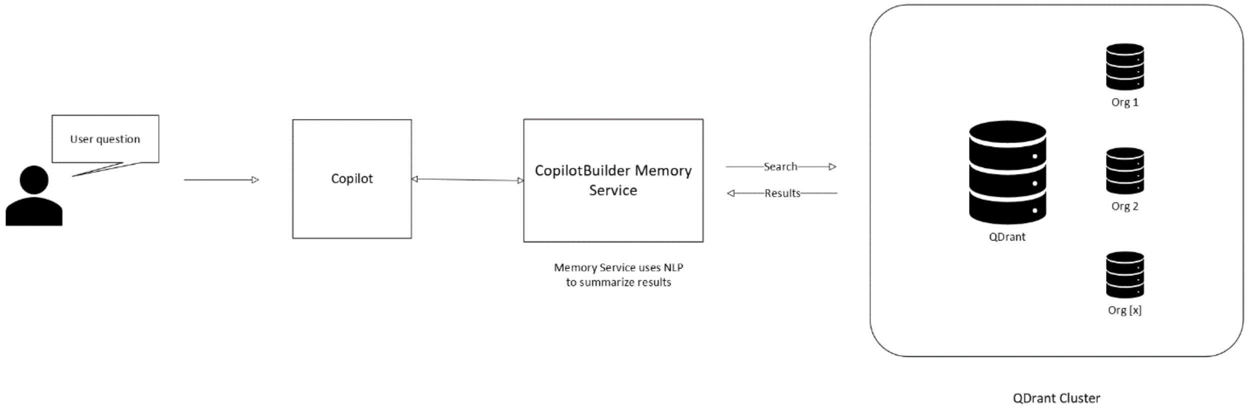
Finally, when a Copilot linked to a data collection receives a question, CopilotBuilder searches through these data vectors and generates an answer. If an appropriate answer is not found, the Copilot will either:

- **If the Copilot is set to only answer questions from data collections:** state that no answer could be found
- **If the Copilot is set to also allow accessing a LLM's public data set:** attempt to retrieve an answer

During the retrieval process, the AI model does not have access to the retrieved data or uploaded documents.

Data Retention

When a document or an entire data collection is deleted, its content is completely removed from CopilotBuilder. This includes its vectors and document in Azure storage.


User question
Copilot
CopilotBuilder Memory Service
Memory Service uses NLP to summarize results
Search
Results
QDrant
Org 1
Org 2
Org [x]
QDrant Cluster

Prompt Security

CopilotBuilder has built-in security for common exploits such as:

- Prompt injection
- Prompt leaking
- Prompt jailbreaking

Additionally, content filters are used to protect against offensive or potentially illegal requests.