# FieldOps Assistant: Technical Architecture & Implementation Strategy

Target Technology: Microsoft 365 Copilot Retrieval API (Public Preview)

Version: 1.1

# Part 1: Strategic Overview & Architecture

## 1. Executive Summary

This document outlines the technical architecture for the "FieldOps Assistant," a phased AI pilot designed to assist frontline staff with equipment troubleshooting.

The core recommendation is to utilize the newly released **Microsoft 365 Copilot Retrieval API** (/beta/copilot/retrieval). This approach allows us to leverage the existing security, compliance, and indexing infrastructure of our Microsoft 365 tenant, avoiding the need for high-risk, "shadow IT" vector databases or external storage.

## 2. The Core Advantage: "Retrieval API" vs. Traditional RAG

Traditionally, building a Retrieval-Augmented Generation (RAG) bot required complex infrastructure (vector databases, separate indexing pipelines). The Retrieval API simplifies this by exposing the native M365 semantic index.

| **Feature** | **The "Old Way" (Custom RAG)** | **The "New Way" (M365 Retrieval API)** |
| --- | --- | --- |
| **Data Storage** | Requires copying PDFs to external Azure Blob/SQL. | Data stays in **SharePoint Online**. |
| **Indexing** | Requires building/managing a custom vector index. | Uses the existing **M365 Semantic Index**. |
| **Security** | Complex custom logic to filter user permissions. | **Native Security Trimming:** API respects Entra ID (Azure AD) permissions automatically. |
| **Compliance** | Data crosses trust boundaries. | Data remains within the M365 trust boundary. |

## 3. Technical Implementation Guide

### A. Prerequisites

To deploy this architecture, the IT team will need:

1. **Identity:** An App Registration in **Microsoft Entra ID** with Copilot.Read permissions.
2. **License:** Users require a **Microsoft 365 Copilot** license to execute queries against the index.
3. **Data Source:** A clean, curated SharePoint Document Library containing the target PDF manuals.

### B. The Application Flow

The workflow follows a stateless, secure request pattern:

1. **User Input:** The frontline worker asks a question in Teams (e.g., *"What is the lubrication interval for Pump P-203?"*).
2. **API Call:** The backend service (Azure Function or Copilot Studio) sends a POST request to the Microsoft Graph endpoint.
3. **Security Check:** Microsoft Graph validates the user's Entra ID token to ensure they have read-access to the specific SharePoint files.
4. **Retrieval:** The API returns relevant text snippets ("chunks") from the manuals, along with citation metadata (File Name, URL).
5. **Synthesis:** These snippets are passed to the LLM (GPT-4o) to generate a natural language answer.
6. **Response:** The answer is delivered to Teams with clickable citation links.

### C. Sample API Payload (Technical Reference)

*Endpoint:* https://graph.microsoft.com/beta/copilot/retrieval

{  
 "query": "lubrication interval for P-203 blower",  
 "top": 3,  
 "input": {  
 "text": "What is the recommended lubrication for the P-203 blower?"  
 },  
 "filters": {  
 // Critical for Pilot: Scopes search ONLY to the Vendor Manuals library  
 "container": "https://[YourOrg][.sharepoint.com/sites/FieldOps\_Pilot\_Docs](https://.sharepoint.com/sites/FieldOps\_Pilot\_Docs)"  
 }  
}

## 4. Implementation Options

### Option A: The "Low-Code" Pilot (Recommended)

**Platform:** **Microsoft Copilot Studio** (formerly Power Virtual Agents).

* **Why:** Copilot Studio creates an abstraction layer over the Retrieval API. You simply select "SharePoint" as your knowledge source in the settings.
* **Time to Deploy:** Hours.
* **Maintenance:** Zero code to maintain.

### Option B: The "Pro-Code" Custom Build

**Platform:** **Azure Functions** + **Bot Framework SDK**.

* **Why:** If the organization requires a highly customized UI or integration with non-Microsoft systems (e.g., SAP/Maximo) in the future.
* **Time to Deploy:** Weeks.

## 5. Security & Governance FAQ

*For discussion with InfoSec and IT Governance teams.*

Q: Does this data leave our tenant?

A: No. The Retrieval API operates entirely within the Microsoft 365 trust boundary. The data flow is SharePoint -> Graph API -> Teams.

Q: What if a user asks about a document they shouldn't see?

A: The API enforces Permission Trimming at query time. If a user does not have permission to view a file in SharePoint, the API will not return that file in the search results, and the bot will effectively say "I don't know."

Q: How do we prevent hallucinations?

A: We utilize Grounding. The LLM is instructed via System Prompt to only answer using the context snippets provided by the Retrieval API. If the API returns no results, the model is forced to reply with uncertainty rather than fabrication.

# Part 2: Detailed Implementation Reference

This section provides a step-by-step checklist for the technical team to configure the **Option A (Copilot Studio)** pilot environment.

## Phase 1: Data Preparation (SharePoint)

The quality of the AI response is directly dependent on the quality of the source documents.

1. **Create Dedicated Site:**
   * Create a new SharePoint Online site (e.g., FieldOps\_Pilot\_KnowledgeBase).
   * *Governance Note:* Ensure permissions are set to "Read Only" for the pilot user group to prevent accidental deletion.
2. **Curate & Ingest:**
   * Upload ~20-50 high-value PDF manuals (start small).
   * **CRITICAL:** Ensure all PDFs are OCR’d (Optical Character Recognition). If the text cannot be highlighted with a mouse cursor, the AI cannot read it.
3. **Capture URL:**
   * Note the full URL to the Document Library.
   * *Example:* https://[Tenant].sharepoint.com/sites/FieldOps\_Pilot\_KnowledgeBase/Shared Documents

## Phase 2: Copilot Studio Configuration

1. **Initialize Bot:**
   * Navigate to [copilotstudio.microsoft.com](https://copilotstudio.microsoft.com).
   * Select **Create** > **New Copilot**.
   * **Name:** "FieldOps Maintenance Assistant".
2. **Connect Knowledge Source:**
   * In the "Generative AI" or "Knowledge" settings:
   * Locate the **SharePoint** field.
   * Paste the URL captured in Phase 1.
   * *Note:* It may take 15-30 minutes for the index to initially crawl the documents.
3. **Configure Authentication (Crucial for Security):**
   * Go to **Settings** > **Security** > **Authentication**.
   * Select **"Authenticate with Microsoft"**.
   * *Why:* This passes the user's Entra ID token to the SharePoint Graph API. Without this, the bot cannot "see" the documents because it won't have permission.

## Phase 3: System Prompt Engineering

To ensure the bot acts as an industrial assistant and not a generic chatbot, apply the following instructions in the **Generative AI / Instructions** panel:

**Role:** You are an expert Industrial Maintenance Assistant for the organization. Your goal is to help frontline operators and mechanics troubleshoot equipment.

**Directives:**

1. **Source of Truth:** ONLY answer using the provided SharePoint documents. Do not use outside internet knowledge.
2. **Safety First:** If the user asks about physical repair or maintenance tasks, always begin your response with: "⚠️ **Safety Check:** Ensure Lock-Out/Tag-Out (LOTO) procedures are followed before proceeding."
3. **Uncertainty:** If the answer is not in the documents, state clearly: "I cannot find that information in the current manuals. Please check with a supervisor." Do not guess.
4. **Formatting:** Use bullet points for steps. Be concise.
5. **Citations:** Always include the reference link to the source document.

## Phase 4: Deployment & Testing

1. **Publish:**
   * Click the **Publish** button in Copilot Studio to make changes live.
2. **Teams Integration:**
   * Go to **Channels** > **Microsoft Teams**.
   * Click **"Turn on Teams"**.
   * Copy the "App ID" or "Installation Link" to share with the pilot user group.
3. **Validation Test:**
   * *Query:* "What is the max operating pressure for [Specific Pump Name]?"
   * *Expected Result:* Correct value returned + Link to PDF manual.