
📄 Copilot Core Architecture Proposal

Author: JI INSU
Date: February 2026
Contact: [knigth286@gmail.com]

📌 Executive Summary

This proposal outlines a unified Copilot Core architecture that connects Microsoft 365 apps, GitHub Copilot, Windows, and Security Copilot through synchronized sessions and a shared learning pipeline. The goal is to evolve Copilot from a product-specific assistant into a proactive, cross-domain operating partner.

🧠 Architecture Overview

1. Copilot Core as Central Orchestrator
- All apps (Word, Excel, PowerPoint, VS Code, GitHub Copilot, Windows) connect to a shared Copilot Core.
- Each task is assigned a Session ID (e.g., Session #42), enabling seamless context sharing and session continuity.

2. LLM Integration
- The LLM connects directly to Copilot Core.
- Enables real-time orchestration: session triggering, code generation, content injection.

3. Windows + Security Copilot Integration
- Copilot is embedded in Windows 11, accessing system-level context (files, logs, settings).
- A Security Copilot layer connects to Microsoft Defender, Sentinel, and Intune.
- Enables real-time threat analysis, incident summarization, and guided mitigation.

4. Session-Based Learning Loop
- User feedback, activity logs, and corrections are captured per session.
- Enables personalization and cross-domain reasoning.

🧩 Architecture Diagram

Subject: Proposal: Embedding Copilot into Productivity and Security Architecture

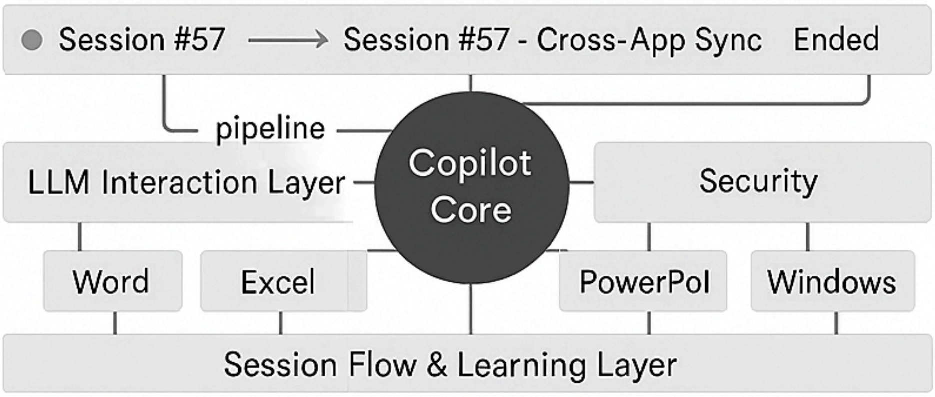
Hello,

I’d like to propose a unified architecture where Copilot not only enhances productivity across Microsoft’s product suite but also integrates seamlessly with security functions.

- By connecting Copilot via the same central core and synchronized session pipeline, context and user activity ca be preserved across Word, Excel, PowerPoint, VS Code, Windows, and security systems.
- This integration extends to a “Security Copilot” layer that interfaces with Microsoft Defender for real-time threat intunit and adaptive threat response.
- Copilot can leverage user feedback and past security incidents, alongside productivity context, to provide a smarter, session-aware experience.
- The architecture ensure Copilot operates as a proactive coordinator, managing applications and protecting the system.

I’d welcome an opportunity to discuss this further.

Integrated Copilot Architecture



Integrated Copilot Architecture

Integrated Copilot Architecture
View Diagram (copilot.microsoft.com in Bing)

🎯 Strategic Opportunity

- LLMs like Claude focus on code-centric workflows.
- Copilot can dominate productivity + security by owning the full workflow.
- Copilot becomes the default AI layer of the modern OS.

🔓 API Access Request

To validate this architecture, I request access to:

- GitHub Models API
- Copilot Core API (if available)

This access would enable prototyping of session-based orchestration, cross-app reasoning, and feedback-driven learning. I’m happy to share experimental outcomes and structural feedback.

🗣️ Language Note

As I’m not a native English speaker, I’ve used Copilot to help translate and structure this proposal. I hope the intent and architecture are conveyed clearly, and I’d be happy to clarify any part if needed.