Azure OpenAI Integration

Roy OS Integration Guide | BIDIRECTIONAL CUSTOMER-OWNED

Overview

Azure OpenAI provides the LLM inference engine for Roy OS. The agent sends meeting transcript context and user queries to a dedicated Provisioned Throughput Unit (PTU) deployed within your Azure subscription and receives structured tool calls, insights, and summaries in return. All traffic stays within your Azure tenant — no data leaves your subscription.

> **Key ownership model:** You provision and own the Azure OpenAI resource. You manage the API keys. Roy OS is a consumer of your LLM resource, not the other way around. This means you control the model version, capacity, content filtering, and can revoke access at any time by rotating the API key.

What Roy Uses Azure OpenAI For

Capability	Description
Agent reasoning	The agent sends meeting context and user queries to the LLM, which returns structured tool calls (create task, post to Slack, search Jira, etc.).
Meeting summaries	Transcript excerpts are sent to generate structured summaries of decisions, action items, and key discussion points.
Memory extraction	The LLM identifies facts, decisions, and commitments from meeting transcripts for atomization into the memory graph.
Semantic search	Text embeddings are generated for semantic search over institutional and personal memory.

Recommended Model Deployments

Roy OS uses multiple models optimized for different workloads. You'll need to create a deployment for each model tier in your Azure OpenAI resource.

Tier	Model	Workload	Notes

Reasoning	`gpt-4o`	Agent reasoning loop, tool calling, meeting insights, memory fact extraction	Primary model. Handles structured tool calls (create_task, slack_post, jira_create, etc.). Strong at multi-step reasoning over meeting context. PTU recommended for production.
Fast / Triage	`gpt-4o-mini`	Signal triage, meeting summarization, intent classification, deduplication	Used for high-volume, lower-complexity tasks: pre-filtering signals before expensive reasoning calls, quick summarization, wake word intent classification, task deduplication. Significantly lower cost per token.
Embeddings	`text-embedding-3-large`	Semantic search over institutional and personal memory	Powers vector search for memory retrieval and context composition. `text-embedding-ada-002` also supported as a fallback.

> **Model flexibility:** Roy OS is model-agnostic at the configuration layer. The model deployment names are configured via environment variables, so you can swap to newer model versions (e.g., future GPT releases) without code changes — just update the Azure OpenAI deployment and the corresponding environment variable.

API Endpoints Used

Endpoint	Purpose
`POST /openai/deployments/{model}/chat/completions`	Chat completions with function calling — reasoning, triage, summaries, fact extraction
`POST /openai/deployments/{model}/embeddings`	Text embeddings for semantic memory search

Setup Guide

Prerequisites

You'll need an Azure subscription with Azure OpenAI access approved, and access to your Azure Key Vault.

1. **Create an Azure OpenAI Resource**

   In the Azure portal → **Create a resource** → **Azure OpenAI**. Deploy it within your subscription, in a region that meets your data residency requirements.

2. **Deploy a Model**

   In the Azure OpenAI Studio → **Deployments** → **Create new deployment**. Select **GPT-4o** (or equivalent). For production, use **Provisioned Throughput Units (PTU)** for guaranteed capacity. Note the **deployment name**.

3. **Deploy an Embeddings Model**

   Create a second deployment for **text-embedding-ada-002** (or equivalent). This powers semantic search over memory. Note this deployment name too.

4. **Configure Network Access**

   Ensure the Roy OS VNet subnet can reach the Azure OpenAI endpoint. Options: private endpoint (recommended), or public endpoint with IP allowlisting restricted to your VNet's NAT gateway IP.

5. **Store Credentials in Key Vault**

   From the Azure OpenAI resource → **Keys and Endpoint**. Add to Key Vault:

Secret Name	Value
`AZURE-OPENAI-API-KEY`	API key (Key 1 or Key 2)
`AZURE-OPENAI-ENDPOINT`	Endpoint URL (e.g., `https://your-resource.openai.azure.com`)

6. **Provide Deployment Names to Roy AI**

   Share the chat completions deployment name and embeddings deployment name with Roy AI during onboarding. Roy OS is configured to use your specific deployments.

7. **Verify Connectivity**

   Roy AI runs a connectivity test: retrieves the API key from Key Vault, sends a test prompt to the chat completions endpoint, and verifies a valid response.

Sizing Guidance

Environment	Recommended	Notes
Staging / UAT	Pay-as-you-go or shared PTU	Lower priority. Sufficient for integration testing.
Production	Dedicated PTU	Guaranteed throughput for concurrent meeting workload. Size based on expected concurrent meetings × average tokens per meeting cycle.

Managing Access

Action	How
Revoke access	Regenerate API keys in the Azure OpenAI resource. Roy OS cannot make LLM calls until the new key is updated in Key Vault.
Monitor usage	Azure OpenAI resource metrics in the Azure portal: tokens consumed, request count, latency, errors.
Change model version	Update the deployment in Azure OpenAI Studio. No Roy OS configuration change needed if the deployment name stays the same.
Scale capacity	Adjust PTU allocation in Azure portal. No Roy AI involvement required.
Content filtering	Azure OpenAI content filtering is enabled by default. Configure per your organization's policies in Azure OpenAI Studio.

FAQ

Does Microsoft use my data for model training?

No. Azure OpenAI does not use customer data for model training. This is governed by Microsoft's data processing terms per your enterprise agreement.

Does data leave my Azure subscription?

No. The Azure OpenAI PTU runs within your subscription. Roy OS connects to it via VNet-internal routing. No prompt or completion data traverses the public internet.

What data is sent in prompts?

Prompts include: recent transcript excerpts relevant to the current query, the user's question, memory context (RBAC-filtered institutional and personal facts), and tool definitions. Roy OS does not send bulk transcripts — only contextually relevant segments.

Can I audit what Roy sends to the LLM?

Roy OS logs tool call names and durations (never raw prompt content) to Azure Log Analytics. For full request/response auditing, enable Azure OpenAI diagnostic logging on the resource — this is a standard Azure feature you control.
