TECHNICAL SESSION · 90 MINUTES · OCTOBER 2026

Microsoft Foundry

Build, ship and govern AI agents — from model to production

Models · Foundry Agent Service · Microsoft Agent Framework · Observability · AI Gateway · Bicep + GitHub Actions

Live demo: **Foundry Guide** — github.com/frkim/foundry-demo

About this session

Your speaker

[Speaker name] · [Role, team]

[One line about your Azure AI / Foundry experience]

[@handle] · [linkedin.com/in/…] · [email]

Who this is for

- Developers and architects shipping generative AI on Azure
- Platform teams who must govern models, agents and cost
- Anyone migrating from Azure OpenAI, hubs or the Assistants API

Ground rules

- **Foundry (new)** only: Foundry resource + projects
- Status pills on every feature: GA Preview Retiring
- Keyless everywhere — no API keys on screen
- Questions welcome any time; deep dives at the end

Agenda — 90 minutes

1	**Opening & why Foundry** · the enterprise AI problem	10'
2	**Platform tour** · resource, projects, models, portal	15'
3	**Agents** · Agent Service, tools, MCP, A2A, memory	15'
4	**DEMO 1** · Foundry Guide agent	15'
5	**Observe, evaluate, govern** · safety, Control Plane	10'
6	**AI Gateway + DEMO 2** · Azure API Management	10'
7	**Ship it** · Bicep, GitHub Actions, keyless	5'
8	**Roadmap, resources, Q&A**	10'

Three live demos, one public repo, zero API keys. **DEMO 3** — ten short capability demos (Foundry IQ, MCP/A2A, model router, guardrails, skills, durable agents, observability, evaluation, LangGraph, human in the loop) — runs as an extended track, or pick two or three. Everything is reproducible from **github.com/frkim/foundry-demo**.

The enterprise AI problem

Prototypes are easy. Running hundreds of models and agents safely, at scale, is not.

Sprawl

Separate resources per model, per team, per region. Keys in config files. No single inventory of what is deployed or who calls it.

Agents act

Agents call tools, APIs and other agents. That needs identity, approvals, guardrails on tool calls — not just on prompts.

Proof, not vibes

Leaders ask "is it good, is it safe, what does it cost?" — you need traces, evaluations and red-teaming evidence.

Choice

The best model changes every quarter. Switching must be a deployment change, not a rewrite.

Cost & quota

Token budgets per team, chargeback, failover across regions when capacity runs out.

Path to prod

Infra as code, CI/CD, private networking and compliance from day one.

What Microsoft Foundry is

“A managed platform for building, evaluating, and deploying generative AI applications with foundation models.” — Microsoft Learn glossary

BEFORE	MICROSOFT FOUNDRY (NEW)
Azure AI Studio → Azure AI Foundry	**Microsoft Foundry** — one platform for models, agents, tools, evaluation and governance
Azure AI Services	**Foundry Tools** — Speech, Language, Vision, Document Intelligence, Content Safety, Search…
Hub + separate Azure OpenAI / AI Services resources	**Foundry resource + Foundry projects** — Microsoft.CognitiveServices/accounts · kind AIServices · child projects
Assistants API (Agents v0.5 / v1)	**Responses API (Agents v2)** — Foundry Agent Service runtime, wire-compatible with OpenAI agents
Azure AI User / Owner / Account Owner / Project Manager	**Foundry User / Owner / Account Owner / Project Manager** — renamed roles — role IDs and core permissions unchanged

Foundry at scale — the public numbers

Each figure is tied to the moment Microsoft published it — don't mix eras.

70K+

customers using the platform

BUILD 2025 · PRE-RENAME

100 T

tokens processed in the prior quarter

BUILD 2025 · PRE-RENAME

10K+

organizations on Agent Service at its first GA

BUILD 2025

10K+

models from Microsoft, OpenAI, Anthropic, Meta and others

MICROSOFT LEARN · LIVE

1,400+ enterprise connectors

Data-source connectors announced for Agent Service grounding (Build 2025).

From ~1,800 to 10,000+ models

The catalog grew from ~1,800 models (Jan 2025) to 11,000+ in the Ignite 2025 ecosystem message.

02

Platform tour

Foundry resource and projects, the model catalog, deployment types, Foundry Tools and the new portal.

15 min · 00:10 → 00:25

One resource, many projects

RESOURCE GROUP · rg-foundrydemo-dev-swc · Sweden Central

Foundry resource

Microsoft.CognitiveServices/accounts · kind AIServices · allowProjectManagement: true

SHARED BY ALL PROJECTS

Model deployments
gpt-5.4-mini · gpt-5.4-nano
GlobalStandard

Security & networking
Entra ID · disableLocalAuth
public or private endpoints

Governance
RBAC · guardrails · quota
policies at the top level

Connections
App Insights · Search · MCP
reusable by projects

PROJECTS · accounts/projects — development boundary

proj-foundrydemo-dev — this demo

- Agents
- Conversations
- Evaluations
- Traces
- Files
- Vector stores
- Connections

Another team's project — isolated access, data & cost

- Agents
- Evaluations
- Data

Project endpoint: `https://<resource-name>.services.ai.azure.com/api/projects/<project-name>`

Foundry (new) vs. classic

	Foundry project (new)	Hub-based project (classic)
Parent resource	Foundry resource (`accounts/projects`)	Foundry hub + separate resources
Agent Service	GA on the Responses API	Preview, Assistants-API based
Portal	ai.azure.com — **Foundry (new)**	Foundry (classic) portal only
Foundry API & SDK	Full `azure-ai-projects` 2.x surface	Partial
Prompt flow, managed compute (Hugging Face)	Not available	Hub only
New agent and model capabilities	**Yes — only here**	No

Migration moves model deployments, data files, fine-tuned models and vector stores. It does **not** move preview agent state (threads, messages, files), open-source model deployments or hub project access.

Endpoints and SDK surface

```python
from azure.ai.projects import AIProjectClient
from azure.identity import DefaultAzureCredential

project = AIProjectClient(
    endpoint="https://aif-foundrydemo-dev-xxxxx"
             ".services.ai.azure.com"
             "/api/projects/proj-foundrydemo-dev",
    credential=DefaultAzureCredential(),
)

agents = project.agents              # create_version, list…
openai = project.get_openai_client() # Responses API, v1
```

One endpoint per project

The project endpoint is the only URL the app needs — agents, conversations, evaluations and model calls hang off it.

One OpenAI client

`get_openai_client()` returns an authenticated OpenAI client on the v1 route — no `api-version` juggling.

Keyless by default

`DefaultAzureCredential`: your az login locally, managed identity in Azure.

Model catalog highlights

OpenAI

GPT-5.x generations incl. `gpt-5.4`, `-mini`, `-nano`, `-pro`; GPT-5 family; `gpt-oss`; GPT-4.1; o-series; GPT-4o.

Anthropic Claude

Opus, Sonnet and Haiku families GA via Azure Marketplace — Anthropic-hosted or Azure-hosted.

Open-weight

DeepSeek V3.2 / V4, Meta Llama 4 Maverick and 3.3-70B, Cohere Command-A and rerank v4, Mistral Large 3 Preview.

Model Router

One deployment that routes each request across models from several providers — pick quality vs. cost per prompt.

Foundry Local

On-device inference with an OpenAI-compatible API — CUDA, WebGPU, NPU, OpenVINO or CPU.

This demo

`gpt-5.4-mini` powers the agent; `gpt-5.4-nano` is the fast model for side-by-side comparison.

The catalog changes weekly — always confirm model names, versions and regions at ai.azure.com before you commit to one.

Deployment types — choose by residency, load and latency

Requirement	Recommended deployment type
Default: newest models, lowest price, broadest regions	**Global Standard** (`GlobalStandard`)
Reserved, predictable throughput	Global Provisioned
Keep processing in the EU / US / APAC data zone	Data Zone Standard (`DataZoneStandard`) or Data Zone Provisioned
Keep processing in one Azure geography	Standard or Regional Provisioned (where supported)
Large asynchronous jobs, ~50% cheaper, 24 h target	Global Batch (`GlobalBatch`) or Data Zone Batch
Evaluate a fine-tuned model (temporary, no SLA)	Developer

Pay-per-token types suit bursty traffic; provisioned types give low latency variance at consistent high volume. The demo uses **Global Standard** in Sweden Central.

Foundry Tools — the services formerly known as Azure AI Services

Speech

Speech-to-text, text-to-speech and speech translation.

Language

Language understanding and Translator.

Documents

Document Intelligence and Content Understanding.

Vision

Vision and Custom Vision.

Content Safety

Harm categories, Prompt Shields, blocklists — the engine behind Foundry guardrails.

Azure AI Search

Hybrid and vector retrieval — the backbone of Foundry IQ knowledge.

Same resource

Reached through the same `AIServices` account, the same Entra auth, the same RBAC.

The new Foundry portal — ai.azure.com

Portal capability (new Foundry)	Status
Agents (core)	GA
Publish to Microsoft 365 Copilot / Teams	GA
Guardrails for models	GA
Guardrails for agents	Preview
Monitoring / observability	Preview
AI Gateway (APIM) pane	Preview
Voice Live	Preview

Foundry projects only

Hubs, standalone Azure OpenAI resources and classic v1 agents stay in the classic portal.

Build → Operate

Model playgrounds, agent builder, evaluations, then **Operate** for compliance, AI Gateway and fleet views.

Tip

Use the portal to explore; commit agents and infra as code for anything that ships.

Developer experience — VS Code, GitHub Copilot, azd

Foundry Toolkit for VS Code

The evolution of AI Toolkit: browse models, build agents, run playgrounds and traces from the editor. Install via `aka.ms/foundrytk`.

Microsoft Foundry skill

Reusable Copilot guidance for model deployment, prompt and hosted agents, A2A, evaluation and agent optimization.

Foundry DevPack

Foundry Toolkit + `azd` / `azd ai` + Foundry Canvas + the Foundry skill, bundled.

```
/plugin marketplace add microsoft/azure-skills
/plugin install azure@azure-skills      # also wires Azure MCP + Foundry MCP servers
```

Works with GitHub Copilot in VS Code and in the Copilot CLI.

03

Agents

Foundry Agent Service, the tool catalog, MCP and A2A, knowledge and memory, Microsoft Agent Framework — and the code.

15 min · 00:25 → 00:40

Foundry Agent Service — generally available on the Responses API

GA
next-gen Agent Service, Responses API-based runtime
MARCH 16, 2026

v1
wire-compatible with OpenAI agents — one OpenAI client
RESPONSES API

2
GA agent types: prompt agents and hosted agents
OCTOBER 2026

2027
classic Assistants-API agents retire on March 31
DEPRECATED

Versioned agents

Every `create_version()` call produces an immutable agent version you can evaluate, compare and roll back.

Conversations

Server-side conversation state — no thread plumbing, just pass a `conversation` id.

Evaluations GA

Evaluations reached GA with the same March 2026 release.

Agent types — pick the right level of control

Prompt agents GA

Instructions + model + tools, fully managed by Foundry — no infrastructure. Defined with `PromptAgentDefinition`. Voice-based prompt agents via Voice Live are GA too.

Hosted agents GA

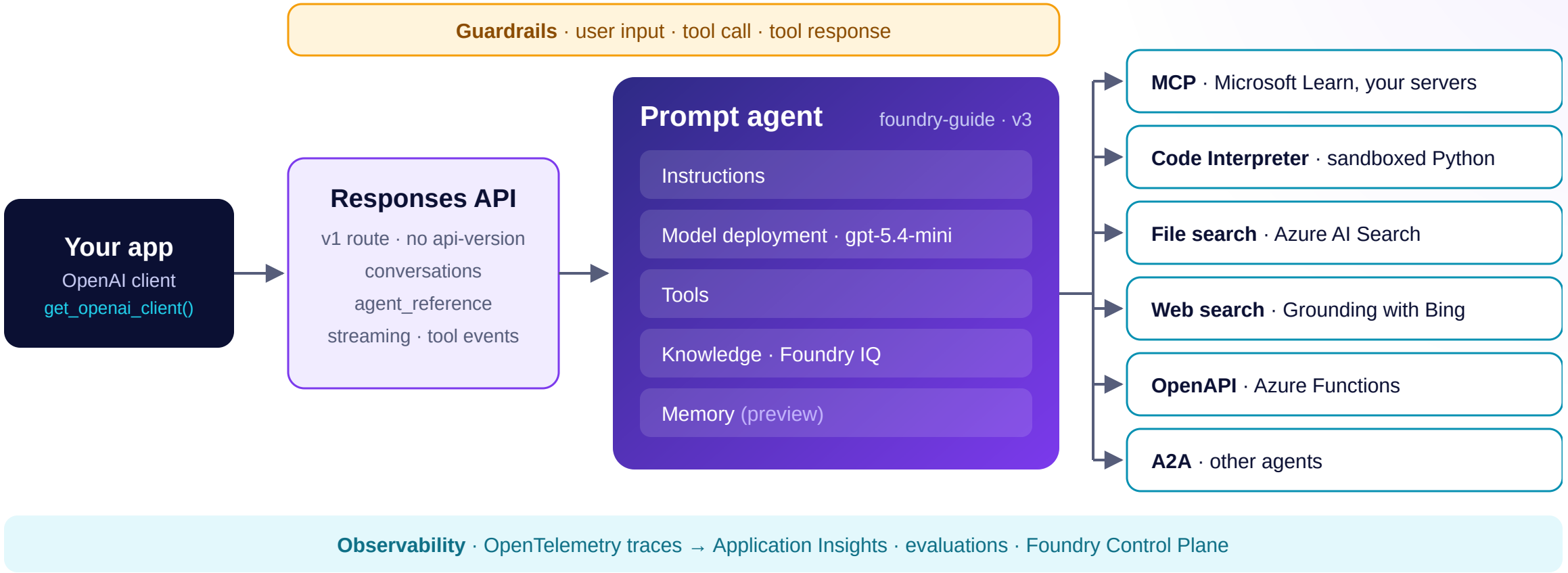
Bring your own code as a managed container: Microsoft Agent Framework, LangGraph, OpenAI Agents SDK, Anthropic Agent SDK, GitHub Copilot SDK.

Workflows Retiring

The visual / YAML multi-agent workflows never left preview and retire on **December 1, 2026**. Build new orchestrations with Agent Framework.

Connected agents (classic `agent.as_tool`) are deprecated — use the **A2A tool** or Agent Framework orchestration instead.

Anatomy of a Foundry agent



Tool catalog

Generally available	
Function calling · custom tools	GA
Code interpreter · file search	GA
Web search / Grounding with Bing	GA
Azure AI Search · knowledge retrieval	GA
OpenAPI · Azure Functions	GA
MCP tools (remote + custom)	GA
A2A tool (protocol v1.0, outbound)	GA
Toolbox	GA

In preview	
SharePoint · Microsoft Fabric connectors	Preview
Browser automation · computer use	Preview
Image generation	Preview
Skills (platform catalog)	Preview
Deep research	Preview
Incoming A2A endpoint	Preview

Demo 1 uses only GA tools: **MCP** (Microsoft Learn) and **Code Interpreter**.

MCP, A2A and Toolbox — open protocols, governed centrally

MCP tools GA

Point an agent at any remote MCP server — `server_label`, `server_url`, `require_approval`, optional project connection for auth.

Demo: `https://learn.microsoft.com/api/mcp`

A2A GA

Protocol v1.0: a Foundry agent calls another agent, anywhere. Exposing a Foundry agent as an A2A endpoint is **Preview**.

External A2A agents can be registered in Foundry Control Plane.

Toolbox GA

Define a curated set of tools once and expose them through a single MCP-compatible endpoint, shared across agents.

Skills (versioned `SKILL.md`) ride on Toolbox — **Preview**

Approval matters: `require_approval="always"` returns an `mcp_approval_request` your app must answer — use it for any tool with side effects.

Foundry IQ — the knowledge layer for agents

Agentic retrieval over Azure AI Search, exposed to agents through an MCP endpoint. Core GA since Build 2026.

1

Plan

The model decomposes the question into focused sub-queries.

2

Search

Sub-queries run in parallel as hybrid (keyword + vector) searches.

3

Rerank

Results are reranked so only the most relevant passages remain.

4

Synthesize

A grounded answer with citations back to the sources.

Sources

Unifies Work IQ, Fabric IQ, Azure SQL, File Search and MCP sources.

Still in preview

Fabric IQ, Work IQ and the Serverless tier; portal UI for agentic retrieval.

Wiring

Attach as an `MCPTool` pointing at the knowledge base MCP endpoint.

Agent memory

Preview

User profile

Stable facts and preferences about the user.

Chat summary

Condensed context from past conversations.

Procedural

How-to knowledge the agent learns from completed tasks.

+7–14 pts

absolute success-rate gain on Tau-bench from procedural memory, at near-baseline cost

BUILD 2026 ANNOUNCEMENT

1

Extract

Pull candidate memories from conversations.

2

Consolidate

LLM-based deduplication and merge.

3

Retrieve

Inject relevant memories at run time.

Memory and the Memory Store API are **preview**. Treat stored memories as personal data: retention, deletion and consent.

Microsoft Agent Framework 1.x — code-first agents

1.0
production-ready, stable APIs, long-term support
APRIL 3, 2026

1.20
latest `agent-framework` Python release
OCTOBER 2026

2
GA languages: Python and .NET (`Microsoft.Agents.AI`)
GO IN PREVIEW

SK+AG
successor to Semantic Kernel and AutoGen
OPEN SOURCE

Orchestration

Code-first multi-agent orchestration, including Magentic-One — stable since Build 2026 and the replacement for retiring Foundry workflows.

Interop built in

MCP and A2A out of the box; wraps Foundry prompt agents or runs as a Foundry **hosted agent**.

Agent Harness GA

Released July 2026 — includes Skills as a framework feature.

Code — create a versioned prompt agent

```
from azure.ai.projects import AIProjectClient
from azure.ai.projects.models import MCPTool, PromptAgentDefinition
from azure.identity import DefaultAzureCredential

project = AIProjectClient(endpoint=PROJECT_ENDPOINT, credential=DefaultAzureCredential())

learn = MCPTool(server_label="microsoft_learn",
                server_url="https://learn.microsoft.com/api/mcp",
                require_approval="never")

agent = project.agents.create_version(
    agent_name="foundry-guide",
    definition=PromptAgentDefinition(
        model="gpt-5.4-mini", instructions=INSTRUCTIONS, tools=[learn]),
)
```

`azure-ai-projects` 2.x · the demo also adds `CodeInterpreterTool` and runs this idempotently at startup — a failure never crashes the app, `/health/ready` reports it.

Code — invoke the agent through the Responses API

```
openai = project.get_openai_client()           # Responses API, v1 route
conversation = openai.conversations.create()

response = openai.responses.create(
    conversation=conversation.id,
    input="How do I enable tracing for a Foundry agent?",
    extra_body={"agent_reference": {"name": "foundry-guide",
                                    "type": "agent_reference"}},
)
print(response.output_text)

for item in response.output:                   # evidence of tool use
    if item.type == "mcp_call":
        print(item.server_label, item.name)
```

Same OpenAI client for agents and raw model calls — `responses.create(model="gpt-5.4-nano", …)` powers the demo's model compare.

Reach users where they work

Publish to Microsoft 365 Copilot and Teams GA

- Foundry portal → **Publish** → **Teams and Microsoft Copilot**
- Auto-generates the Teams app manifest and submits it to the catalog
- Auto-creates an Azure Bot Service resource
- Scope: individual (`BotServiceRbac`) or organization (`BotServiceTenant`, admin approval)
- Requires the **Foundry User** role

Or your own front end

Any app that can call the Responses API — like the Vue + FastAPI demo — can host the same agent version.

Or other agents

Through A2A, your agent becomes a building block for agents on other platforms.

04

DEMO 1 — Foundry Guide

A keyless FastAPI + Vue app on Azure Container Apps, talking to a Foundry prompt agent with MCP and Code Interpreter.

15 min · 00:40 → 00:55

Foundry Guide — what you'll see

Agent chat

Ask about Foundry; the agent grounds answers in Microsoft Learn via MCP. Tool-call chips show every MCP or Code Interpreter call.

Model compare

The same prompt on `gpt-5.4-mini` and `gpt-5.4-nano` in parallel — output, latency and tokens side by side.

Run history

Last 200 runs: time, kind, model or agent, latency, tokens, status — search, sort, filter.

About

Live `/api/info`: endpoint host, models, agent name and version, app version, region.

0

API keys — user-assigned managed identity end to end

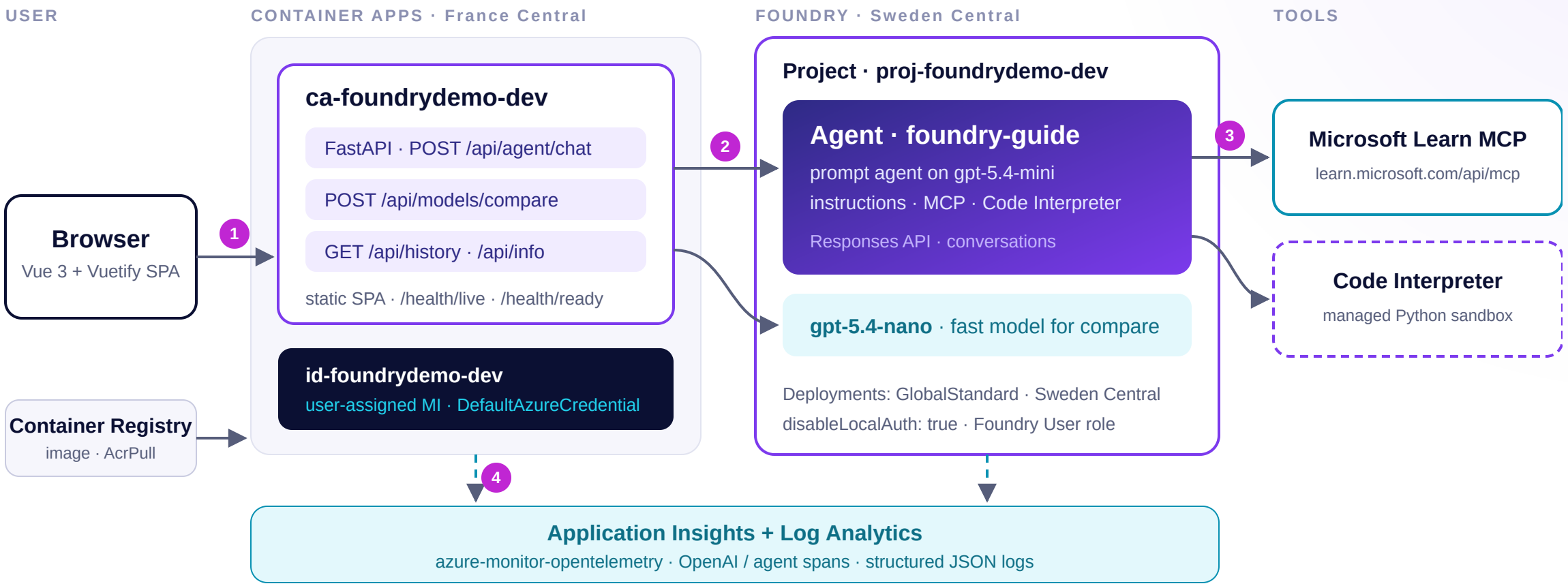
2

GA tools on the agent: MCP + Code Interpreter

1

container: FastAPI API + Vue / Vuetify SPA, port 8000

Demo 1 architecture



1 HTTPS to the SPA + API 2 Entra token from the managed identity — no keys 3 agent calls MCP / Code Interpreter 4 OpenTelemetry to App Insights

Demo 1 — script

- [ ] **About** tab — endpoint, region, agent name + version, app version
- [ ] **Agent chat** — suggested prompt; point at the *microsoft_learn* MCP chip
- [ ] Follow-up in the same conversation; ask for a quick calculation → Code Interpreter
- [ ] **Model compare** — same prompt on mini vs. nano; read latency + tokens
- [ ] **Run history** — filter by kind, sort by latency
- [ ] Foundry portal — agent versions, then traces in **Application Insights**

What to point out

- No key in env vars — only `AZURE_CLIENT_ID`
- Agent is code-defined and versioned
- Grounded answers carry Learn links
- Same client for agents and raw models

Fallback: `docs/video/foundry-demo.mp4` and `docs/session/demo-runbook.md`.

Under the hood — one chat turn

1

UI

`POST /api/agent/chat` with the message and an optional conversation id.

2

Identity

The managed identity selected by `AZURE_CLIENT_ID` gets an Entra token.

3

Conversation

First turn creates a Foundry conversation; later turns reuse its id.

4

Agent run

`responses.create()` with an `agent_reference` — the agent picks its tools.

5

Result

Text, conversation id, tool calls, token usage and latency — plus a trace.

Readiness, not just liveness

`/health/ready` returns 503 until configuration is present and the agent version exists — Container Apps won't route traffic before that.

Fail soft

Agent creation runs at startup but never crashes the app; errors surface in readiness and logs.

05

Observe, evaluate, govern

Tracing, evaluations, red teaming, guardrails, Foundry Control Plane — and the Entra, Defender and Purview integrations.

10 min · 00:55 → 01:05

Observability — traces you can actually use

Tracing Preview

- OpenTelemetry-based, off by default
- **Server-side** traces auto-logged once Application Insights is connected — 90-day retention
- **Client-side**: OTel + Azure Monitor exporter, or OTLP to Datadog, Grafana Tempo, Jaeger, Honeycomb

Agent monitoring dashboard Preview

Token usage, latency, success rate, evaluation scores and red-team results per agent.

```python
from azure.monitor.opentelemetry import (
    configure_azure_monitor,
)

# reads APPLICATIONINSIGHTS_CONNECTION_STRING
configure_azure_monitor()
# + OpenAI / agent instrumentation:
#   one span per model call, agent run
#   and tool call
```

In the demo, Bicep creates a project connection to Application Insights so the portal shows the same traces.

Evaluations and AI red teaming

Evaluations GA

Quality and safety evaluators on datasets or agent runs — compare agent versions before you promote one.

Continuous evaluation Preview

Recurring evaluations on live traffic, surfaced in the monitoring dashboard.

AI Red Teaming Agent

Built on open-source **PyRIT** + Foundry risk and safety evaluations; reports **Attack Success Rate**.

Agentic risk categories

Prohibited actions, sensitive-data leakage, task adherence — cloud red teaming in East US 2, France Central, **Sweden Central**, Switzerland West, US North Central.

Ship gate

Run evaluations on every new agent version in CI; block promotion on regressions; red-team before first exposure to users.

Guardrails and controls

Intervention point / control	Status
Guardrails framework · user input	GA
Tool call	Preview
Tool response	Preview
Prompt Shields (jailbreak / prompt injection)	GA
Task adherence · custom filtering	GA
Spotlighting (tag lower-trust content)	Preview
Guided guardrail set-up wizard	Preview

Why tool guardrails matter

With MCP and web tools, the attack surface is the **tool response** — an injected instruction in a web page or a document.

Defense in depth

Guardrails in Foundry + `llm-content-safety` at the gateway + approval for side-effecting tools.

Foundry Control Plane and enterprise governance

Foundry Control Plane

Central governance surface for compliance and security settings — and registration of external A2A agents.

Microsoft Entra Agent ID

Identities for agents: blueprints, OAuth2 / MCP / A2A, adaptive access, lifecycle governance, audit — also for agents outside Azure.

Defender for Cloud

Preview

Threat protection for AI agents with OWASP-aligned LLM and agentic detections (Feb 2026).

Microsoft Purview

Toggle under **Operate → Compliance** Preview: Audit, DSPM for AI, DLM, eDiscovery, insider risk.

Licensing note: since July 1, 2026, Defender agent discovery and posture require a **Microsoft Agent 365** license; deeper cross-Microsoft 365 agent governance in Entra also needs Agent 365 (included in M365 E7, add-on elsewhere).

D3

DEMO 3 — Foundry capabilities, live

Ten short demos around the agent: knowledge, connectivity, routing, safety, reuse, autonomy, observability, quality, open frameworks and human control.

Extended track · ≈ 45 min · or pick 2–3 in the 90-minute format

Demo 3 — the menu

3.1 Foundry IQ ★
Knowledge base, agentic retrieval, citations · 6'

3.2 MCP + A2A
Tools and agents over open protocols · 4'

3.3 Model router
One deployment, best-fit model per prompt · 3'

3.4 Guardrails
Jailbreak and indirect prompt injection · 4'

3.5 Skills + Toolbox
Curate tools once, reuse everywhere · 4'

3.6 Durable + autonomous
Background runs, hosted agents, routines · 5'

3.7 Observability
Traces, dashboard, continuous evaluation · 4'

3.8 Evaluate + optimize
Evaluators, red teaming, optimizers · 5'

3.9 LangGraph · Deep Agents · LangSmith
Bring your framework, keep Foundry's identity, tracing and hosting · 4'

3.10 Human in the loop
Approvals that pause durably and resume later · 4'

3.1 · Foundry IQ — knowledge for every agent

Knowledge bases GA · agent connection, most source types and portal flow Preview

1

Index once

Knowledge base `zava-kb` on Azure AI Search, fed by three Zava documents.

2

Retrieve agentically

Plan sub-queries, search them in parallel, rerank.

3

Answer with citations

One MCP tool call to the knowledge base — shared by every agent.

Ask: *"Plus member, chair arrived damaged, headphones bought 20 days ago — options?"* Two documents, several sub-queries, one grounded answer.

```
MCPTool(server_label="zava_kb",
        server_url=f"{SEARCH}/knowledgebases/zava-kb/mcp"
                   "?api-version=2026-08-01-preview",
        allowed_tools=["knowledge_base_retrieve"],
        require_approval="never",
        project_connection_id="<kb-connection>")
```

3.2 · Connectivity — MCP for tools, A2A for agents

MCP tool GA

Any remote MCP server; catalog in **Build → Tools**. Credentials live in project connections.

A2A tool v1.0 GA

`demo-a2a-caller` delegates to `demo-learn-expert` through an A2A connection.

Incoming A2A Preview

Expose an agent with an agent card — SDK or REST only today.

```
project.agents.update_details(agent_name="demo-learn-expert",
    agent_endpoint=AgentEndpointConfig(protocol_configuration=ProtocolConfiguration(
        responses=ResponsesProtocolConfiguration(), a2a=A2AProtocolConfiguration())),
    agent_card=AgentCard(version="1.0", skills=[AgentCardSkill(id="foundry-qa", name="Foundry Q&A")]))
caller_tool = A2ATool(a2a_version=A2AProtocolVersion.V1_0, project_connection_id=conn.id)
```

3.3 · Model router — one deployment, best-fit model

```
for prompt in ["Say hello in French.",
               "Resource vs project in 2 sentences.",
               "Reason step by step: which of 3 agents "
               "deserves a reasoning model?"]:
    r = openai.responses.create(model="model-router",
                                input=prompt)
    print(r.model, r.usage.total_tokens)
```

Model router GA

Version `2025-11-18`; pool of OpenAI, Anthropic, xAI, DeepSeek and Meta models; Global Standard in Sweden Central.

Balanced · Quality · Cost

Pick the mode and the model subset; failover is built in. `response.model` shows the pick.

3.4 · Guardrails and prompt injection

Control in `demo-guardrail`	Intervention point	Status
Jailbreak (Prompt Shields)	User input	GA
Indirect attack	Tool response	Preview
Task adherence	Tool call	Preview

The poisoned `zava-supplier-notes.md` hides an instruction in an HTML comment — retrieved by the knowledge base, caught at the tool response.

What you'll see

- "Yendys" jailbreak → blocked at user input
- Supplier question → injected text never reaches the user
- Code: `400 content_filter`

Infrastructure as code

`accounts/raiPolicies` + `raiPolicyName` on a deployment.

3.5 · Skills and reusable tools — Toolbox

Toolbox GA

Curated, versioned tools behind one MCP-compatible endpoint — any framework can consume it.

Skills Preview

`SKILL.md` (agentskills.io) — the Demo 1 cost calculation as a reusable skill.

```
project.beta.skills.create(name="foundry-cost-estimate",
    inline_content=SkillInlineContent(
        description="Token cost estimates; table + chart.",
        instructions=skill_md))
project.toolboxes.create_version(
    name="demo-foundry-toolbox",
    tools=[MCPToolboxTool(server_label="microsoft_learn",
                server_url="https://learn.microsoft.com/api/mcp"),
           CodeInterpreterToolboxTool()],
    skills=[ToolboxSkillReference(name="foundry-cost-estimate")])
```

3.6 · Durable and autonomous agents

Background runs GA

`background=True` → `queued` → poll `responses.retrieve`.

Hosted agents GA

Your code in a per-session sandbox with its own Entra identity — `azd deploy`.

Routines

Schedule, timer, GitHub issue or custom event triggers an agent run.

Durable Task Preview

Checkpointed Agent Framework sessions on Durable Task Scheduler.

```
project.beta.routines.create_or_update("demo-daily-foundry-digest", enabled=True,
    triggers={"weekdays-7am": ScheduleRoutineTrigger(cron_expression="0 7 * * 1-5",
                                                     time_zone="Europe/Paris")},
    action=InvokeAgentResponsesApiRoutineAction(agent_name="demo-learn-expert",
        input="List this week's three most important Microsoft Foundry changes."))
```

3.7 · Continuous observability

Traces GA

Prompt and hosted agents: `invoke_agent` → `execute_tool` → model spans.

Monitor + alerts Preview

Tokens, latency, success rate, evaluation and red-team results per agent.

Continuous evaluation Preview

An evaluation rule scores 25 % of live responses.

```
dependencies
| where timestamp > ago(24h)
| extend op = tostring(
    customDimensions["gen_ai.operation.name"])
| where op in ("invoke_agent", "chat",
               "execute_tool")
| summarize calls = count(),
    p95_ms = percentile(duration, 95),
    failures = countif(success == false)
    by op
```

3.8 · Evaluation and optimization

1

Evaluate GA

`builtin.task_adherence`, `intent_resolution`, `tool_call_accuracy` on ten queries against `foundry-guide`.

2

Compare

Two agent versions, same dataset, side by side — promote on evidence.

3

Red-team

AI Red Teaming Agent (PyRIT): Attack Success Rate before first exposure.

4

Optimize

Prompt Optimizer; Agent Optimizer **Limited preview** proposes candidates — you promote.

```
run = openai.evals.runs.create(eval_id=ev.id, name="foundry-guide-latest", data_source={
    "type": "azure_ai_target_completions",
    "source": {"type": "file_id", "id": dataset.id},
    "input_messages": {"type": "template", "template": [{"type": "message", "role": "user",
        "content": {"type": "input_text", "text": "{{item.query}}"}}]},
    "target": {"type": "azure_ai_agent", "name": "foundry-guide"}})
```

3.9 · LangGraph, Deep Agents and LangSmith on Foundry

`langchain-azure-ai`

Microsoft's integration: keyless Foundry models (`azure_ai:`), Toolbox tools and skills, tracing, hosting.

Host it

`ResponsesHostServer(graph).run()` → `azd deploy` → Foundry hosted agent.

LangSmith

Keep it alongside; Azure Marketplace offer.

```
tracer = AzureAIOpenTelemetryTracer(
    project_endpoint=os.environ["FOUNDRY_PROJECT_ENDPOINT"],
    credential=DefaultAzureCredential(),
    name="demo-langgraph")
agent = create_agent(model="azure_ai:gpt-5.4-mini",
                     system_prompt="Concise Azure architect.")
agent.invoke({"messages": "Why an AI gateway?"},
             config={"callbacks": [tracer]})

deep = create_deep_agent(model="azure_ai:gpt-5.4-mini",
                         system_prompt="Plan, take notes, answer.")
```

3.10 · Human in the loop — durable approvals

Platform GA

MCP `require_approval="always"` → `mcp_approval_request` → answer with `mcp_approval_response`.

Framework GA

Agent Framework `@tool(approval_mode="always_require")`; workflow `ctx.request_info(...)`.

Durable Preview

Pending requests live in checkpoints; on Durable Task Scheduler the wait costs zero compute.

```python
@tool(approval_mode="always_require")
def issue_refund(order_id: str, amount_eur: float) -> str:
    """Issue a refund to the customer."""
result = await refund_agent.run("Refund €49 on order Z-1042, the chair arrived broken.")
for request in result.user_input_requests:   # nothing ran yet — a human decides
    print(request.function_call.name, request.function_call.arguments)
```

06

AI Gateway with Azure API Management

Token budgets, load balancing, content safety, semantic caching and MCP governance — in front of every Foundry endpoint.

10 min · 01:05 → 01:15 · includes DEMO 2

Why put a gateway in front of Foundry?

Token budgets

Tokens-per-minute and monthly quotas per team, product or subscription key — 429 on rate, 403 on quota.

Resilience

Backend pools across regions and projects with circuit breakers that honor `Retry-After`.

Visibility & chargeback

Prompt, completion and total token metrics with custom dimensions in Application Insights.

Safety at the edge

Content Safety categories, Prompt Shields and blocklists before a request reaches any model.

Cost

Semantic caching answers repeated questions without calling the model.

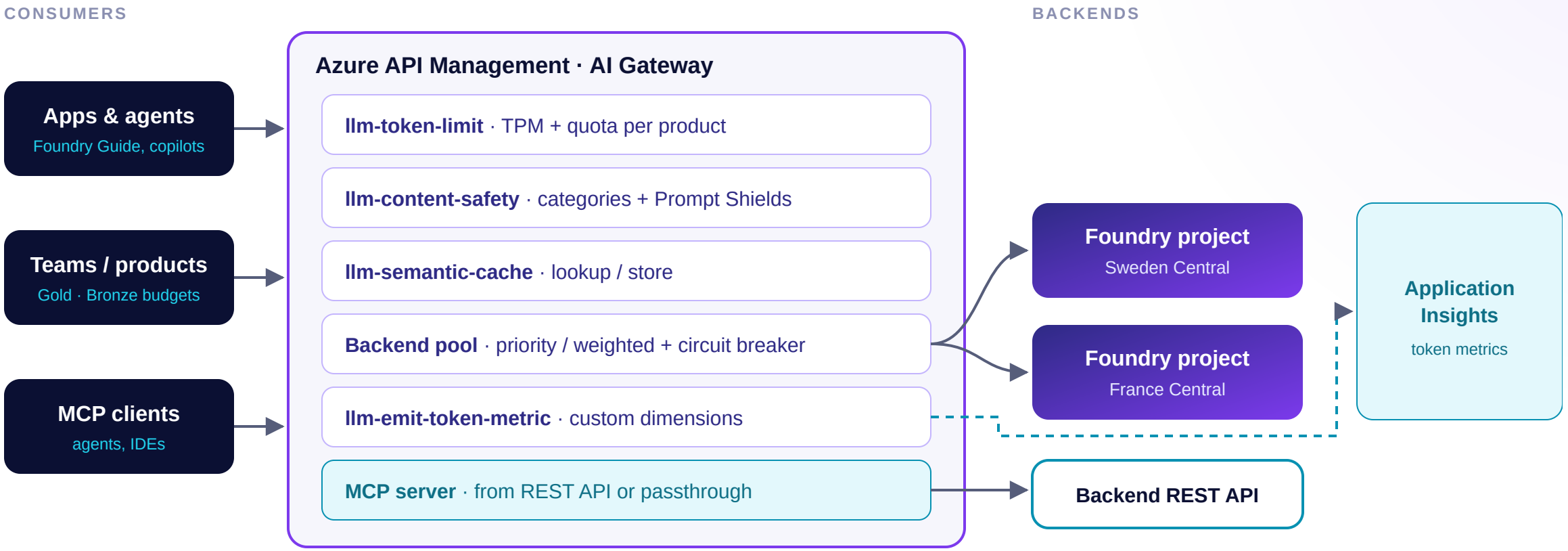
MCP governance

Expose REST APIs as MCP servers, or front existing ones, with the same auth and policies.

The GenAI gateway policies

Capability	Policy / feature	Notes
Token rate limit + quota	`llm-token-limit`	`tokens-per-minute`, `token-quota`, `token-quota-period`; alias `azure-openai-token-limit`
Token metrics	`llm-emit-token-metric`	prompt / completion / total tokens, up to 5 custom dimensions
Semantic caching	`llm-semantic-cache-lookup` / `-store`	needs an embeddings deployment + Azure Managed Redis (RediSearch)
Content safety	`llm-content-safety`	harm categories, `shield-prompt`, blocklists — 403 on block
Load balancing	Backend pools	round-robin, weighted or priority
Failover	Circuit breaker on backends	dynamic trip duration honors `Retry-After`
MCP	MCP server in APIM GA	export a REST API as MCP, or pass through to an existing server

AI Gateway reference architecture



Policy as code — a token-aware inbound pipeline

```
<inbound>
  <base />
  <authentication-managed-identity resource="https://cognitiveservices.azure.com" />
  <set-backend-service backend-id="foundry-pool" />
  <llm-token-limit counter-key="@(context.Subscription.Id)"
                   tokens-per-minute="5000" token-quota="200000"
                   token-quota-period="Monthly" estimate-prompt-tokens="false" />
  <llm-content-safety backend-id="content-safety" shield-prompt="true" />
  <llm-emit-token-metric namespace="ai-gateway">
    <dimension name="Subscription ID" />
    <dimension name="API ID" />
  </llm-emit-token-metric>
</inbound>
```

Gateway → Foundry is keyless too: APIM's managed identity holds the Foundry User role on each backend. Full policies: `frkim/apim-demo/infra/policies/`.

Governing MCP tools with the gateway

MCP server in API Management GA

- Export any managed REST API as an MCP server
- Or pass through to an existing MCP server
- Streamable HTTP (SSE deprecated); tools only — no resources or prompts yet
- Developer, Basic, Standard, Premium — classic and v2 tiers

From the Foundry side Preview

- **Manage → AI Gateway** pane: create one (Basic v2) or attach an existing v2-tier APIM in the same tenant and subscription
- “Govern MCP tools by using an AI gateway” — route agent tool traffic through APIM

Same story for models and tools: identity, limits, logging, one place.

DEMO 2 — AI Gateway in action

```
# github.com/frkim/apim-demo — Bicep, no azd
az deployment sub what-if -n apimaigw-demo -l swedencentral `
  -f infra\main.bicep --result-format ResourceIdOnly
az deployment sub create -n apimaigw-demo -l swedencentral `
  -f infra\main.bicep

cd demo
python -m ai_gateway all
# or: chat | load-balance | token-limit | content-safety
#     | mcp | agent | metrics
```

What it proves

Keyless chat · load balancing across two Foundry projects · throttling · safety enforcement · MCP tools · a Responses API agent flow · token telemetry.

Go deeper — AI-Gateway labs

- backend-pool-load-balancing
- token-rate-limiting
- token-metrics-emitting
- semantic-caching
- content-safety
- model-context-protocol
- mcp-from-api
- ai-foundry-hosted-agents

Labs: github.com/Azure-Samples/AI-Gateway · ~50 Jupyter labs with Bicep and APIM policies

07

Ship it

Bicep for the Foundry resource, projects and deployments; GitHub Actions with OIDC; keyless RBAC everywhere.

5 min · 01:15 → 01:20

Bicep — the Foundry resource and project

```
resource foundry 'Microsoft.CognitiveServices/accounts@2026-07-01' = {
  name: accountName                       // aif-foundrydemo-dev-<suffix>
  location: location
  kind: 'AIServices'
  sku: { name: 'S0' }
  properties: {
    customSubDomainName: accountName      // custom subdomain for Entra ID auth
    allowProjectManagement: true          // enables child Foundry projects
    disableLocalAuth: true                // Entra ID only — no keys
  }
}
resource project 'Microsoft.CognitiveServices/accounts/projects@2026-07-01' = {
  parent: foundry
  name: projectName                       // proj-foundrydemo-dev (+ location, identity)
}
```

Trimmed from `infra/modules/foundry.bicep` · the subscription-scope `main.bicep` creates `rg-foundrydemo-dev-swc` and wires outputs (`projectEndpoint` , `uamiClientId` …) into the container app.

Bicep — model deployments and role assignments

```
@batchSize(1) // deployments on one account must not run concurrently
resource deployments 'Microsoft.CognitiveServices/accounts/deployments@2026-07-01' = [for m in models: {
  parent: foundry
  name: m.name                                  // gpt-5.4-mini, gpt-5.4-nano
  sku: { name: 'GlobalStandard', capacity: m.capacity }
  properties: { model: { format: 'OpenAI', name: m.modelName, version: m.modelVersion } }
}]
resource appIsFoundryUser 'Microsoft.Authorization/roleAssignments@2022-04-01' = {
  scope: foundry
  name: guid(foundry.id, appPrincipalId, foundryUserRoleId)
  properties: {
    principalId: appPrincipalId                 // user-assigned MI of the container app
    roleDefinitionId: subscriptionResourceId('Microsoft.Authorization/roleDefinitions', foundryUserRoleId)
  }
}
```

The repo also sets `versionUpgradeOption: 'NoAutoUpgrade'` so a demo never changes model version underneath you.

GitHub Actions — from push to a ready agent

ci.yml | deploy.yml

Runtime identity: id-foundrydemo-dev → Foundry User on the Foundry resource · AcrPull on the registry

Package installs in CI and Dockerfile go through the protected npm / PyPI feeds only

> The deck itself is built by a third workflow (`deck.yml`): Marp → HTML, PDF and PPTX → GitHub Pages.

Keyless by design

Principal	Role	Scope
App identity (`id-foundrydemo-dev`)	**Foundry User** · AcrPull	Foundry resource · registry
Presenter / deployer principals	Foundry User + Speech User	Foundry resource
APIM managed identity (Demo 2)	Foundry User	each Foundry backend
GitHub Actions	OIDC federated credential	subscription

Foundry User = the renamed Azure AI User role — same role ID, same permissions, so existing assignments keep working.

```
# AZURE_CLIENT_ID picks the
# user-assigned identity
cred = DefaultAzureCredential()
project = AIProjectClient(
    endpoint=ENDPOINT,
    credential=cred)
```

Result

`disableLocalAuth: true` on the account: there is no key to leak, rotate or store in Key Vault.

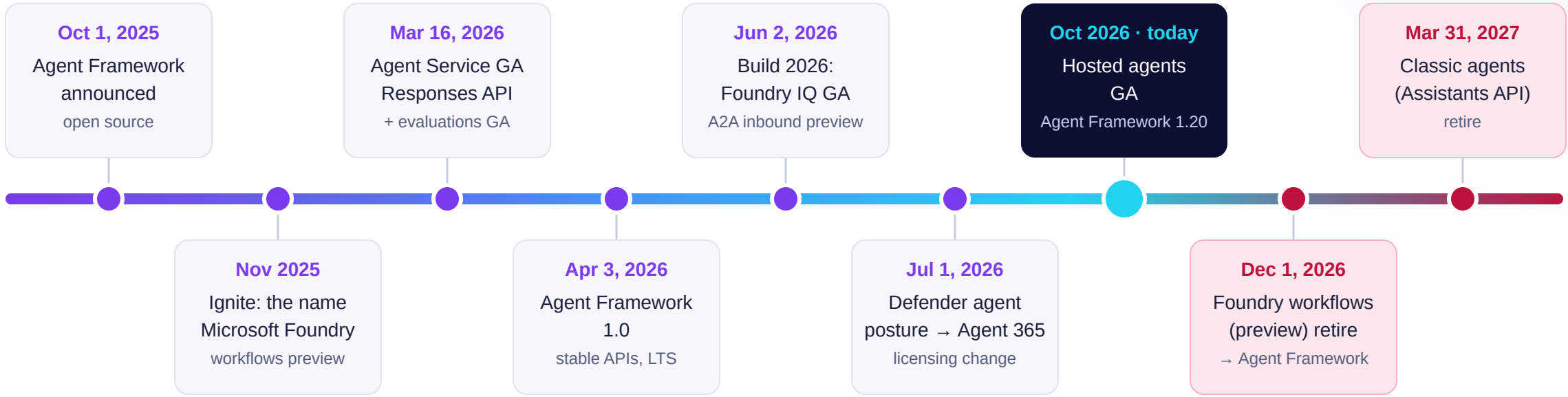
08

Roadmap, resources, Q&A

Key dates to plan around, what to do next week, and where to learn more.

10 min · 01:20 → 01:30

Key dates


Oct 1, 2025
Agent Framework announced
open source
Nov 2025
Ignite: the name Microsoft Foundry
workflows preview
Mar 16, 2026
Agent Service GA
Responses API
+ evaluations GA
Apr 3, 2026
Agent Framework 1.0
stable APIs, LTS
Jun 2, 2026
Build 2026: Foundry IQ GA
A2A inbound preview
Jul 1, 2026
Defender agent posture → Agent 365
licensing change
Oct 2026 · today
Hosted agents GA
Agent Framework 1.20
Dec 1, 2026
Foundry workflows (preview) retire
→ Agent Framework
Mar 31, 2027
Classic agents (Assistants API)
retire
shipped
today
retirement — plan your migration

Call to action — next week, not next year

- [ ] Clone **frkim/foundry-demo** and deploy it to a sandbox subscription
- [ ] Inventory Assistants-API agents and hub projects; plan the move to Foundry projects before **Mar 31, 2027**
- [ ] Replace any Foundry **workflows** with Agent Framework before **Dec 1, 2026**
- [ ] Set `disableLocalAuth: true` and move every caller to Entra ID
- [ ] Turn on tracing + Application Insights for every agent project
- [ ] Add evaluations to CI and gate promotion of new agent versions
- [ ] Put an **AI Gateway** in front of shared model capacity — start with token limits and metrics
- [ ] Install the Foundry skill in GitHub Copilot for your team

Resources

Microsoft Learn

- learn.microsoft.com/azure/foundry/what-is-foundry
- learn.microsoft.com/azure/foundry/concepts/architecture
- learn.microsoft.com/azure/foundry/agents/overview
- learn.microsoft.com/azure/foundry/agents/concepts/tool-catalog
- learn.microsoft.com/azure/foundry/foundry-models/concepts/deployment-types
- learn.microsoft.com/azure/foundry/concepts/rbac-foundry
- learn.microsoft.com/azure/foundry/guardrails/guardrails-overview
- learn.microsoft.com/azure/api-management/llm-token-limit-policy
- learn.microsoft.com/azure/api-management/mcp-server-overview

Code

- **github.com/frkim/foundry-demo** — this session: app, Bicep, workflows, deck
- github.com/frkim/apim-demo — AI Gateway demo
- github.com/Azure-Samples/AI-Gateway — ~50 labs
- github.com/microsoft/agent-framework

Announcements

- devblogs.microsoft.com/foundry/foundry-agent-service-ga
- devblogs.microsoft.com/foundry/agent-service-build2026

Q&A · THANK YOU

Questions?

Thank you — now go ship a governed agent.

Slides, app, Bicep and workflows: **github.com/frkim/foundry-demo**

AI Gateway demo: github.com/frkim/apim-demo · Labs: github.com/Azure-Samples/AI-Gateway

[Speaker name] · [contact]