

Agent Platform RFP Checklist: OpenAI Agents SDK vs Google ADK vs AgentCore vs Copilot Studio vs Azure Agent Service

A monetization play: sell platform selection as a paid audit. Use this RFP checklist to turn agent platform choice into a scoping memo, then an implementation sprint and an operations retainer.

Source <https://yetyield.com/blog/agent-platform-rfp-checklist-for-openai-agents-sdk-adk-agentcore-copilot-studio-and-azure-agent-service/>

Most buyers do not need you to “build an agent.”

They need you to choose a platform that can ship, govern, and operate agent workflows inside their constraints. That selection process is billable because it decides:

- which workflows are safe to automate
- how tools are integrated and governed
- where approvals must exist
- how you will monitor quality and cost

This article turns platform choice into a paid RFP + decision memo product, then expands into implementation and a monthly operations retainer.

The monetization angle

Sell a three-step ladder:

1. Paid platform selection audit (RFP + decision memo): shortlist + trade-offs + governance plan.
2. One-workflow implementation sprint: ship a reference agent workflow end-to-end.
3. Operations retainer: tool onboarding, approvals, observability, evaluation gates, incident response.

If you want to anchor the pricing logic, start with:

- [How to Price Agent Platform Operations Retainers \(Without Hand-Wavy AI ROI\)](/blog/how-to-price-agent-platform-operations-retainers/) (/blog/how-to-price-agent-platform-operations-retainers/)

Start with “who owns orchestration”

The single most important question:

Do you want your application to own orchestration, tools, approvals, and state, or do you want a hosted runtime to own more of it?

Official platform entry points you can cite when building the RFP appendix:

- OpenAI Agents SDK overview: <https://developers.openai.com/api/docs/guides/agents>
- Google ADK overview: <https://docs.cloud.google.com/agent-builder/agent-development-kit/overview>
- Amazon Bedrock AgentCore overview: <https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/what-is-bedrock-agentcore.html>
- Copilot Studio agents for M365 Copilot: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/microsoft-365-copilot-extend-with-agents>
- Azure AI Foundry Agent Service connector surface: <https://learn.microsoft.com/en-us/connectors/azureagentservice/>

The checklist (use as RFP sections)

1) Workflow definition and boundaries

Ask the buyer to provide:

- top 3 workflows (inputs, outputs, systems touched)
- definition of “done” (what artifact or system write counts as success)
- side effects allowed (write actions) vs prohibited
- required approvals and audit evidence

If they can't answer these, they are not ready for platform selection. Your audit is the product.

2) Tool governance and approvals

Your RFP must force a decision about side effects:

- Can the platform pause before sensitive tool calls?
- Can approvals be serialized and resumed?
- Are guardrails attached to tool calls, not only to chat outputs?

For an official approvals surface you can cite:

- OpenAI “Guardrails and human review”: <https://developers.openai.com/api/docs/guides/agents/guardrails-approvals>

For a governance boundary outside agent code:

- AgentCore Policy: <https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/policy.html>

3) Tool onboarding and integration model (MCP, OpenAPI, connectors)

Ask:

- How do we turn an internal API into a callable tool?
- How is auth handled (inbound + outbound)?
- Can we route tool calls through a centralized gateway?
- Can we maintain tool catalogs and discovery?

If your buyer is AWS-heavy, this is an official “platform wedge”:

- AgentCore Gateway: <https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/gateway.html>

If your buyer is OpenAI SDK-heavy, decide whether tools are hosted MCP vs runtime-managed MCP connections:

- OpenAI “Integrations and observability” (MCP wiring + tracing): <https://developers.openai.com/api/docs/guides/agents/integrations-observability>

4) Execution environments (sandbox vs normal runtime)

Ask:

- Do workflows require file manipulation, scripts, package installs, or previews?
- Do you need stateful work across runs?
- Do you need “container boundaries” for risky steps?

If yes, sandbox capability becomes a pricing wedge.

Official OpenAI reference:

- Sandbox agents: <https://developers.openai.com/api/docs/guides/agents/sandboxes>

5) Observability, debugging, and audit evidence

Ask:

- Do we get end-to-end traces of model + tool steps?
- Can we tag runs, filter, and review failure modes?
- Can we export or integrate telemetry into existing systems?

Official references you can use as evidence surfaces:

- OpenAI tracing + run inspection loop: <https://developers.openai.com/api/docs/guides/agents/integrations-observability>
- AgentCore Observability: <https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/observability.html>

6) Buyer-side runtime “harness” choice (especially in Microsoft ecosystems)

In Microsoft environments, “platform choice” often hides a second decision: what harness/

runtime powers the agent and what it enables.

Official surface:

- Copilot Studio “Choose a harness”: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/harnesses-overview>

This becomes a paid scoping decision because harness choice affects:

- file/tool capabilities
- how multi-step work is handled
- how the solution is billed

7) Evaluation strategy (pre-ship + post-ship)

Ask:

- What test set represents real inputs?
- What does failure look like?
- Who owns the evaluation rubric and updates it monthly?

Tie this to existing YetYield coverage:

- Agent Regression Tests Can Be a Retainer Business (/blog/agent-regression-tests-as-a-retainer-business/)
- LangSmith Online Evals as an Agent Quality Monitoring Retainer (/blog/langsmith-online-evals-as-agent-quality-monitoring-retainer/)

8) Commercial constraints (what procurement will reject)

Ask directly:

- Must the platform support a specific cloud?
- Is “preview” acceptable or must it be GA?
- Are external model calls allowed?
- Can any data leave the tenant?

This section is where “best platform” arguments die. That is why the audit is billable.

Deliverable: the platform decision memo (what you sell)

Your audit should end with a memo that includes:

- Workflow shortlist (ranked by ROI and risk)
- Platform fit matrix (why each platform fits/doesn't fit)
- Governance model (tools, approvals, policies)
- Observability plan (what gets traced, who reviews it)
- Implementation plan (first workflow reference build)

- Retainer scope and pricing tier

The memo is what the buyer can forward internally to get budget approval. Without it, you're just "the AI person."

What to avoid

- pretending "one platform wins" without constraints
- selling platform selection without naming governance and approval boundaries
- writing a generic comparison that doesn't produce a decision memo

Next research direction: a template SOW for "platform selection audit + first workflow build + 90-day operations retainer."