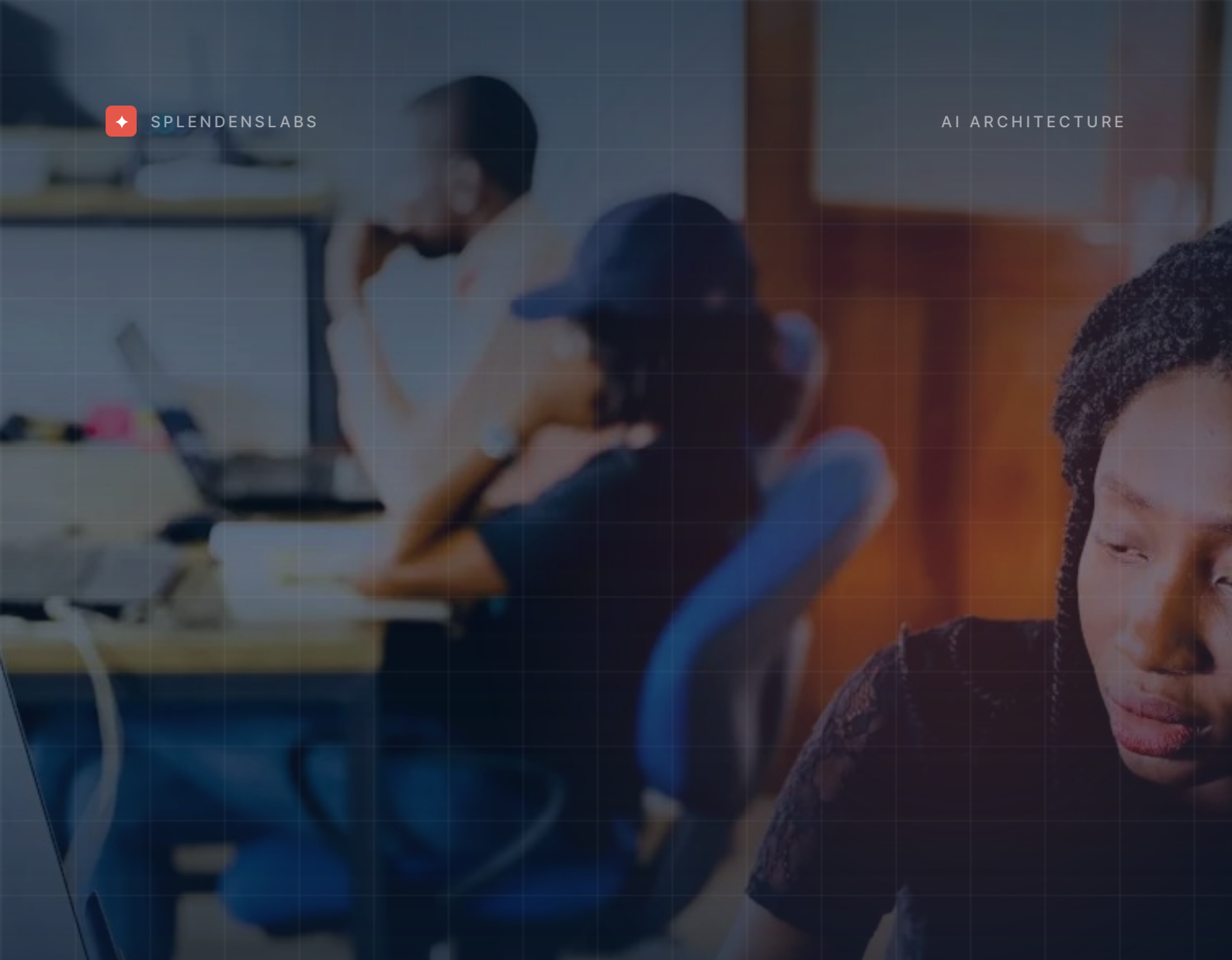


✦ COMPARISON · AI ARCHITECTURE

OpenAI vs Anthropic

A platform-level comparison of the two leading frontier model providers: model families, API features, cost levers, enterprise data controls and cloud availability — and why we design for both.

FORMAT	LEVEL	READ TIME	UPDATED
Comparison	Intermediate	10 min	Oct 2026

OVERVIEW

What this comparison covers

Choosing a model provider is less about which model tops this month's leaderboard and more about the platform around it: tool use, structured outputs, caching, batch processing, data controls and where you can buy it. This comparison looks at OpenAI and Anthropic as platforms, avoids version-specific claims that age quickly, and explains why we recommend a multi-provider design backed by your own evaluations.

KEY TAKEAWAYS

- Both providers offer capable model tiers, tool use, streaming, prompt caching and batch processing — the details differ.
- Model rankings change every few months; your own task-specific evals are the only ranking that matters.
- Cloud availability often decides the question: Azure for OpenAI models, AWS Bedrock and Google Vertex AI for Claude.
- Enterprise data terms, retention options and regional processing should be confirmed in writing for your account.
- Put a thin provider-abstraction layer in front of both so you can route, fall back and switch without a rewrite.

WHO IT'S FOR

CTOs and architects selecting an LLM provider

AI engineers building on model APIs

Procurement, security and compliance reviewers

Product leads planning AI features

HOW TO USE IT

Use the tables in 01–03 to understand the platforms, read 04 before committing to one, and take the checklist in 06 into your provider evaluation.

CONTENTS

01 Two platforms, different emphases
02 API features compared
03 Data controls and cloud availability
04 Single provider vs multi-provider
05 When to choose which
06 Decision checklist

01 / POSITIONING

Two platforms, different emphases

Both companies sell access to frontier models through an API, consumer and business chat products, and agent tooling. Their platforms overlap heavily but are shaped by different priorities.

OPENAI

Broad, multimodal platform

General and reasoning model families plus first-party embeddings, image, speech and realtime voice APIs under one account.

ANTHROPIC

Focused on text, code and agents

The Claude model family in tiers by capability and speed, with emphasis on long context, tool use, coding and agentic work.

Model families at a glance

ASPECT	OPENAI	ANTHROPIC
Model lineup	GPT general-purpose models, reasoning-focused models, smaller fast variants	Claude in three tiers: largest (Opus), balanced (Sonnet), fast and low-cost (Haiku)
Reasoning	Reasoning models with adjustable effort	Extended thinking with a configurable budget or effort
Modalities in	Text, images, audio	Text, images, PDFs
Modalities out	Text, images, speech, realtime voice	Text (tool calls can produce other artifacts)
Embeddings	First-party embedding models	No first-party embeddings; pair with a third-party model
Open-weight	Has released open-weight models	No open-weight models

READ THIS FIRST

Verify before you decide

Model names, context limits, prices, rate limits and feature availability change frequently and differ by region and cloud. Treat this document as a map of the terrain and confirm current specifics in each provider's documentation at the time of reading.

02 / DEVELOPER PLATFORM

API features compared

For production systems the API surface matters as much as raw model quality. Here is how the building blocks line up.

CAPABILITY	OPENAI	ANTHROPIC
Core API	Responses API (plus Chat Completions)	Messages API
Tool use	Function calling, built-in hosted tools	Tool use, server-side tools, computer use
Structured outputs	Strict JSON-schema structured outputs	Structured outputs and schema-validated tool inputs
Prompt caching	Automatic caching of repeated prefixes	Explicit cache breakpoints you control
Batch processing	Batch API at a discount, async results	Message Batches API at a discount, async results
Agent tooling	Agents SDK; hosted agent tools	Claude Agent SDK; originator of MCP
Long documents	Large context windows; file inputs	Large context windows; PDF support and citations
Fine-tuning	Available on selected models	Limited; offered through selected cloud partners

Practical differences engineers notice

- Caching: automatic caching needs no code changes, while explicit breakpoints give more predictable hit rates on long system prompts.
- Both support parallel tool calls and streaming of tool arguments; error and stop-reason semantics differ, so abstract them.
- Tool-calling formats differ in shape; a shared internal tool schema avoids writing every tool twice.
- The Model Context Protocol (MCP) is supported across both ecosystems, which makes tool servers portable.

03 / ENTERPRISE

Data controls and cloud availability

For regulated teams the deciding factor is often procurement: which cloud you already buy from, where data is processed and what is retained.

CONSIDERATION	OPENAI	ANTHROPIC
Direct API	OpenAI platform	Claude API
Major clouds	Microsoft Azure (Azure OpenAI / AI Foundry)	Amazon Bedrock, Google Cloud Vertex AI; also offered via Microsoft's Foundry
Training on API data	Not by default for business and API data	Not by default for commercial and API data
Retention options	Zero-retention arrangements for eligible use cases	Zero-retention arrangements for eligible use cases
Data residency	Regional options vary by product and cloud	Regional options largely via cloud provider regions
Compliance	Common attestations such as SOC 2; HIPAA via agreement	Common attestations such as SOC 2; HIPAA via agreement

FIELD NOTE

Buying through your cloud changes the contract

Using a model through Azure, Bedrock or Vertex AI typically puts it under your existing cloud agreement, billing, IAM and network controls. Feature availability can lag the direct API, so check that the specific capabilities you need are live in your region.

04 / HEAD TO HEAD

Single provider vs multi-provider

The bigger architectural question is not which provider, but whether to depend on just one.

COMMIT TO ONE PROVIDER	MULTI-PROVIDER BY DESIGN
Simplest code and one contract	Route each task to the model that wins your evals
Use provider-specific features fully	Fallback when one provider is degraded
Exposed to outages, price changes and deprecations	Negotiating leverage and less lock-in
Switching later means re-testing every prompt	Requires an abstraction layer and evals per model

A multi-provider request path

01 App calls internal model gateway → 02 Route by task and policy → 03 Primary provider call → 04 Fallback on error or timeout → 05 Validate structured output → 06 Log, trace and meter cost

05 / GUIDANCE

When to choose which

Starting points, not verdicts. Run your own evals before committing a workload.

LEAN OPENAI

One account for many modalities

You need embeddings, image generation, speech or realtime voice alongside text from a single vendor.

LEAN OPENAI

You are an Azure shop

Your security, networking and billing already run through Microsoft Azure.

LEAN ANTHROPIC

Long documents and coding agents

Workloads centered on large documents, code generation or multi-step agentic tool use often favor Claude in evals.

LEAN ANTHROPIC

You are on AWS or Google Cloud

Bedrock or Vertex AI lets you keep traffic and billing inside your existing cloud.

BOTH

Route by task

Use each provider where it wins your evals, behind one gateway with fallbacks.

NEITHER ALONE

Strict residency or offline needs

Consider open-weight models you host yourself for data that cannot leave your environment.

How to run a fair bake-off

01 **Collect real tasks**

Gather 50–200 representative inputs with expected outputs or grading rubrics from your own users.

02 **Compare comparable tiers**

Test flagship against flagship and fast tier against fast tier, with the same tools and context.

03 **Score quality, latency and cost together**

Measure end-to-end, including caching and retries, not list price per token.

04 **Re-run on every model release**

Keep the suite in CI so new versions are adopted on evidence.

06 / DECIDE

Decision checklist

Take this into your provider evaluation with engineering, security and procurement in the room.

CAPABILITY

- [ ] Task-specific eval suite run on both providers' comparable tiers
- [ ] Required modalities (voice, images, embeddings) confirmed per provider

- [ ] Tool use and structured outputs tested with our real schemas

ENTERPRISE

- [ ] Data retention, training and residency terms confirmed in writing
- [ ] Preferred cloud channel identified and features verified in our region
- [ ] Rate limits and quota increase process understood

ARCHITECTURE

- [ ] Internal gateway abstracts provider APIs and tool formats
- [ ] Fallback provider tested for critical paths
- [ ] Cost and latency tracked per model, per feature

GOVERNANCE

- [ ] Model deprecation schedules tracked
- [ ] Current pricing verified at time of decision
- [ ] Owner assigned to re-run evals on new releases

> **Pick a provider for today, design for two, and let your evals decide tomorrow.**
>
> SPLENDENSLABS ENGINEERING PRINCIPLE

GET IN TOUCH

Choose on evidence

We build production AI systems on OpenAI, Anthropic and open-weight models, often all three. Let us set up the gateway, the evals and the bake-off so your choice is based on your data.

- Provider bake-off on your real tasks
- Multi-provider gateway with fallbacks
- Cost, latency and quality dashboards

BOOK AN AI STRATEGY SESSION →

EMAIL
hello@splendenslabs.com

WEB
splendenslabs.com

OFFICE
Sona Incubation Foundation, Salem