Google Foundational Models & Vertex AI by Michaël Bettan

01. Core Definition & Architecture

Native Multimodality (Early Fusion) vs. Modular Perception Pipelines

Google's Gemini foundational models are engineered from the ground up around **native multimodality (early fusion)**. Instead of bolting separate vision or audio encoders onto a pretrained text-only backbone, all modalities — text, code, images, audio, video, and PDFs — are projected into a **unified, shared token space** during pretraining and processed by a single **Sparse Mixture-of-Experts (MoE)** transformer backbone.

Why bolted-on late-fusion cascades struggle on interactive multimodal tasks:

- **Inherent information loss:** Converting live audio or video into intermediate text strips paralinguistic cues (tone, hesitation, sarcasm, urgency) and fine-grained spatial-temporal alignment.
- **Compounded serial latency:** Chaining `ASR → Text LLM → TTS` creates sequential bottlenecks that break sub-second conversational voice and live video agents.
- **Cross-modal hallucination:** A text-dominant LLM frequently overrides visual or acoustic evidence across shallow adapter layers when signals conflict.

ARCHITECTURAL TRADE-OFF — NATIVE MULTIMODAL VS. MODULAR PIPELINE (PAA EXAM FOCUS)

Even with native multimodality, **modular pipelines** (`Chirp 3` / `Gemini 3.5 Transcribe` or `Document AI Layout Parser` → deterministic redaction/schema validation → text LLM) remain the right choice when compliance mandates **deterministic, auditable intermediate transcripts**, **PII redaction (`Sensitive Data Protection`) before reasoning**, custom domain phrase adaptation, or caching parsed text across many downstream passes. Choose **native multimodal** (`Gemini 3.8 Live` , `Gemini 4 Argon` GA agentic video understanding) when sub-second latency, prosody/visual nuance, or dynamic frame inspection is primary.

Evolution from PaLM 2 to Gemini Sparse MoE

- **PaLM 2 Baseline (Legacy, Retired):** Dense text-first transformer activating 100% of parameters on every token, requiring separate Vision or USM adapters.
- **Gemini Sparse Mixture-of-Experts (MoE):** Conditional learned routers dispatch each token to a specialized subset of expert feed-forward blocks — decoupling total model knowledge capacity from active per-token FLOPs, cost, and Time-to-First-Token (TTFT).

02. Active Ecosystem Tiers (October 2026 Lineup)

NAMING MILESTONE — GEMINI 4 PERIODIC-ELEMENT NAMING

Starting with **Gemini 4**, Google shifted frontier model naming to **periodic-table elements**, led by **Gemini 4 Argon (`gemini-4-argon` , GA October 13, 2026)** and **Gemini 4 Argon Cyber Permissive (`gemini-4-argon-cyber-permissive`)** (note: periodic-element naming starts at **Argon** — there is no "Gemini 4 Carbon" SKU).

Tier	Active flagship models (Oct 2026)	Architectural role & sweet spot
On-Device / Edge	**Gemini Nano** (Android AICore / Chrome Built-in AI), **Gemma 4 E2B / E4B**, **Gemma 3n**	Zero-cloud-cost, offline, privacy-preserving client inference under strict mobile/laptop RAM and battery envelopes
Flash-Lite (High-QPS)	**Gemini 3.5 Flash-Lite (`gemini-3.5-flash-lite`)**, `gemini-3.1-flash-lite`	Lowest cost & TTFT at massive scale: intent routing, inline guardrails, entity extraction, classification, bulk translation
Flash (Agentic Workhorse)	**Gemini 3.8 Flash (`gemini-3.8-flash`)**, `gemini-3.8-flash-cyber` , `gemini-3.7-flash` , `gemini-3.5-flash`	Default production agentic loop: long-horizon coding, high-velocity MCP/function calling, multimodal reasoning (default model in **CodeMender**)
Live, Speech & TTS	**Gemini 3.8 Live (`gemini-3.8-live`)**, `gemini-3.8-live-extended-thinking` , **3.8 Flash TTS**, **3.5 Transcribe**	Sub-second bidirectional voice/video agents with barge-in, background extended thinking during live speech, 130-language studio TTS, and low-latency diarized STT
Frontier Reasoning (Gemini 4)	**Gemini 4 Argon (`gemini-4-argon`)**, `gemini-4-argon-cyber-permissive` *(plus `gemini-3.1-pro` preview)*	Maximum System-2 reasoning for complex multi-step planning, repository-scale software engineering, legal/financial synthesis, math/science, computer use, and Fairwind-gated defensive cyber
Managed Autonomous Agents	**Antigravity Agent (`antigravity-preview-09-2026`)**, **Deep Research / Deep Research Max**, **CodeMender**	Task-level endpoints running multi-step browsing, coding, synthesis, or vulnerability patching inside managed harnesses/sandboxes

Legacy & retired lifecycle status: Gemini 2.5 (`2.5 Pro` , `2.5 Flash` , `2.5 Flash-Lite`) remains in legacy support for existing workloads (and managed SFT/preference tuning on Agent Platform), while new Gemini API projects default to **3.5 Flash-Lite**, **3.8 Flash**, and **4 Argon**. **Gemini 2.0 Flash / Flash-Lite**, `gemini-2.5-computer-use-preview-10-2025` , and standalone `Imagen 4` on Gemini API are shut down.

03. Model Garden & Serving Options

> **NAMING NOTE**
> Vertex AI Model Garden is now **Agent Platform Model Garden**, and Vertex AI MaaS is **Model as a Service (MaaS) on Agent Platform**.

Model Garden is the unified enterprise catalog of 200+ Google first-party, partner, and open-weight models with integrated evaluation, tuning, IAM, VPC-SC, and deployment controls.

The Three Model Families

Family	Representative models	Consumption model	Billing unit
First-party (Google)	Gemini 4 Argon, Gemini 3.8 Flash, 3.5 Flash-Lite, 3.8 Live, Nano Banana 2.1, Gemini Omni 1.1 Flash, Veo 3.1, Lyria 3.5, Chirp 3, Gemini Embedding 2	Fully managed serverless API only (cannot download weights)	Per token / media unit (PayGo) or reserved **GSUs** (Provisioned Throughput)
Partner (MaaS)	**Anthropic Claude**, **Mistral AI**, **DeepSeek** (e.g., `DeepSeek-V4.1-Flash` 1M context preview)	Managed serverless MaaS API inside Google Cloud governance	Per token (PayGo) or Provisioned Throughput where offered
Open-weight	**Gemma 4** (`E2B` , `E4B` , `26B A4B MoE` , `31B`), Gemma 3/3n, MedGemma, ShieldGemma 2, Llama, Qwen	**Either** curated serverless MaaS API **or** self-deployed on dedicated Agent Platform endpoints / **GKE**	MaaS: per token. Self-deployed: hourly VM + GPU/TPU compute **while deployed**

Partner Models on Agent Platform (Claude Example)

- **Google-managed Claude (`AnthropicVertex`):** Serverless inference inside Google Cloud's security perimeter with unified IAM, VPC-SC, Cloud Audit Logs, CMEK/data-residency controls, and Google Cloud billing.
- **Anthropic-managed Claude:** Marketplace-billed access to Anthropic-native platform features via a **separate** Anthropic organization (existing direct Anthropic org keys/ZDR configs do not transfer automatically).
- **Governance:** Enforce `constraints/vertexai.allowedPartnerModels` org policies to allowlist approved partner models. See the Anthropic & Claude guide.

Serving Open Models: MaaS vs. Dedicated Endpoint vs. GKE

Criterion	Serverless MaaS API	Self-deployed Agent Platform Endpoint	Self-deployed on GKE
Ops burden	Zero (Google autoscales accelerators)	Low–Medium (prebuilt vLLM / TGI / SGLang containers, autoscaling)	High (cluster, node pools, GPU drivers, serving stack)
Custom / LoRA weights	Base catalog models only	✅ Custom checkpoints, LoRA adapters, custom containers	✅ Full control + multi-LoRA dynamic serving
Traffic profile	Spiky, unpredictable, or prototyping	Steady, high utilization (lower TCO at sustained QPS)	Multi-model bin-packing across shared GPU/TPU pools
Billing trap	Pay strictly per token	**Billed continuously per node-hour even at 0 QPS** (configure idle undeploy)	Billed for provisioned node pools / reservations

Model and serving decision matrix

Model choice solves for: matching each agent node to the cheapest option that still clears its quality, latency and compliance gates. Three axes move independently: **scale** (frontier LLM vs. efficient SLM tier), **serving** (managed API vs. self-hosted capacity) and **weights** (proprietary vs. open).

Option	Data residency	Single-tenant compute in your network	Latency	Cost model	Customization	Best agent role
Frontier managed LLM (`gemini-4-argon`)	Regional, `us` / `eu` multi-region, or `global` (no residency guarantee)	No: multi-tenant Google API, reached privately through VPC-SC and Private Google Access / PSC for Google APIs	Highest (thinking tokens add time before output)	Per token (PayGo tiers) or Provisioned Throughput GSUs	Prompting, grounding, implicit caching; no managed tuning on Argon	Coordinator, planner, ambiguous tool selection, escalation path
Efficient managed tier (`gemini-3.8-flash` , `gemini-3.5-flash-lite`)	Same endpoint choices as above	No: same managed API path	Low; Flash-Lite lowest TTFT in the Gemini lineup	Lowest per-token rate; PT, Flex and Batch available	Prompting, caching, SFT/LoRA on eligible versions	Tool-loop worker, router, classifier, extractor, inline guardrail

Option	Data residency	Single-tenant compute in your network	Latency	Cost model	Customization	Best agent role
Partner MaaS (Claude on Agent Platform)	Regional, multi-region or `global` depending on the model card	No: serverless partner endpoint governed by Google Cloud IAM, VPC-SC and audit logs	Medium to high (extended thinking adds time)	Per token; PT ordered through a Google Cloud account team	Prompts and prompt caching; no weight access	Long-horizon coding or synthesis worker where the model is already standardized
Self-hosted open weights, Model Garden endpoint (Gemma 4 on a dedicated private endpoint)	Single region you pick for the endpoint	Yes: dedicated endpoint exposed only through a PSC endpoint in your VPC	Low to medium (set by accelerator shape and replica count)	Node-hours while deployed, even at zero traffic	Full weights, LoRA or full fine-tunes, custom vLLM containers	Specialist model with tuned open weights, served privately by a team that does not want to run Kubernetes
Self-hosted open weights, GKE + vLLM (Gemma 4 on GPUs or TPUs)	Region or zone of your node pools	Yes: pods in your private cluster; weights can load from your own storage instead of a public hub	Lowest achievable when tuned (continuous batching, prefix caching)	Node-pool or reservation cost; cheapest per token only at sustained high utilization	Full control of weights, quantization, serving flags, multi-LoRA	High-QPS fleet of fine-tuned specialists, strict isolation, existing GKE platform team

> **NOTE**
> Latency and cost columns are directional rankings, not benchmarks. Validate with your own load test and golden evaluation set before committing to capacity.

- **Isolation note:** VPC-SC and PSC keep managed-API traffic private and block exfiltration; inference still runs on Google's shared serving stack. Only self-hosted endpoints run on capacity dedicated to you.
- **Tuned Gemini caveat:** tuned Gemini models deploy only to shared public endpoints, not to PSC endpoints, so a "fine-tuned and PSC-only" requirement points to an open model on a private endpoint or GKE.

Regional, multi-region and global endpoints

Endpoint	Where ML processing runs	Gains	Gives up
Regional (`us-central1`, `europe-west4`, ...)	The named region	Strict residency; none of the `global` feature gaps; CMEK-protected regional resources	Lower burst headroom than `global`; more `429` risk under regional spikes
Multi-region (`us`, `eu`)	Anywhere inside that jurisdiction	Jurisdiction-level residency with better availability than one region	Not single-region; CMEK unavailable for Agent Runtime on multi-region; Private Google Access not supported (use PSC endpoints or backends)
Global (`global`)	Any supported Google Cloud location	Highest availability, fewer `429` errors	No residency guarantee; no tuning, no RAG corpus, no batch for Anthropic and open MaaS models; CMEK unavailable for Sessions and Memory Bank

```python
from google import genai
# "eu" keeps ML processing in the EU; "global" offers no residency guarantee
client = genai.Client(vertexai=True, project="my-proj", location="eu")
```

- **Rule of thumb:** if a requirement says "processing must stay in country X", rule out `global` first, then pick regional (single region) or multi-region (jurisdiction only).
- **CMEK:** Cloud KMS keys live in a geographic boundary, so the `global` location cannot hold CMEK-protected resources. Pair CMEK mandates with regional endpoints.

Network isolation: VPC-SC, PSC and air-gapped mandates

- **VPC Service Controls:** a perimeter around the Agent Platform API keeps prompts, online inference requests, batch results and tuned models from leaving the protected projects. Inside a perimeter, URL context live fetching is disabled to block exfiltration.
- **Private Service Connect:** dedicated Model Garden and custom endpoints can be published only through a PSC endpoint in your VPC (no public IP); PSC automation lets a network admin pre-authorize endpoint creation with a service connection policy.
- **Org policy allowlist:** `constraints/vertexai.allowedPartnerModels` limits which partner models a project can call (see the partner section above).
- **When self-hosting is forced:** choose self-hosted open weights when policy forbids any multi-tenant inference service, requires auditable weights you control, or needs custom fine-tunes served only on private networking.

- **Fully disconnected mandates:** if the site cannot reach any Google Cloud region at all, managed APIs and cloud GKE are both out. **Google Distributed Cloud air-gapped** is the disconnected sovereign option; it runs Gemini through the Gemini for GDC API as well as open models such as Gemma on vLLM.
- **Anti-pattern:** treating "open-weight" as a security control. Open weights inside a public endpoint are no more isolated than a managed API; the isolation comes from the private endpoint, perimeter and egress controls around them.

04. Agentic Model Selection & Routing Framework

Golden rule: Pick the **smallest, fastest, cheapest model that passes your golden evaluation suite**, and compose models across roles rather than routing every agent turn through a frontier model.

Multi-Model Agentic Routing Pattern

1. **Front-door router & inline guardrail (`Gemini 3.5 Flash-Lite` or `ShieldGemma 2`):** Classifies intent, extracts structured slots, screens policy, and answers trivial FAQs in `<200 ms` at minimal cost.
2. **Primary execution & tool loop (`Gemini 3.8 Flash`):** Executes multi-turn MCP tool calls, code generation, and RAG synthesis at high token velocity.
3. **Frontier escalation / planner (`Gemini 4 Argon`):** Invoked selectively for ambiguous multi-step decomposition, deep repository refactoring, legal/financial synthesis, or conflict resolution when confidence is low.
4. **Vertex AI Model Optimizer:** Managed dynamic router that automatically balances quality vs. cost/latency across Gemini tiers per prompt based on your target preference profile.

LLM vs. SLM & Open-Weight vs. Proprietary

- **LLM (Gemini 4 Argon / 3.8 Flash) vs. SLM (Gemma 4 / Gemini Nano / 3.5 Flash-Lite):** Use LLMs for open-ended planning, massive 1M multimodal context, and zero-shot generalization; use SLMs (especially fine-tuned LoRA SLMs or on-device LiteRT-LM / Android AICore) for high-QPS narrow schemas, strict edge latency, or offline client privacy.
- **Proprietary vs. Open-Weight:** Proprietary APIs deliver peak frontier capability (`Gemini 4 Argon`) with managed safety and zero infra ops, but enforce vendor retirement schedules (e.g., `gemini-4-argon` has a published earliest retirement horizon of Oct 2027+). Open weights (`Gemma 4`) give permanent weight ownership, air-gapped deployment, and custom fine-tuning, **but you own serving, patching, and external guardrails** (**Model Armor** / **ShieldGemma 2**).
- **Exam trap:** *"Open-weight"* does **not** force self-hosting — consume open models via **MaaS** first unless you need custom weights or dedicated VPC isolation.

Coordinator plus specialist workers (mixing LLMs and SLMs)

A multi-agent design solves for: paying frontier prices only on the turns that need frontier reasoning. The coordinator owns ambiguity; workers own narrow, typed subtasks.

```
          [ Coordinator: gemini-4-argon ]  plans, delegates, merges
                /              |              \
  [Router/guard]   [Tool-loop worker]   [Domain specialist]
  3.5 Flash-Lite      3.8 Flash          Gemma 4 + LoRA (private endpoint)
```

Role	Typical model tier	Why
Coordinator / planner	Frontier (`gemini-4-argon`)	Decomposes vague goals, picks among many tools and sub-agents, resolves conflicts
Router and inline guardrail	Flash-Lite or ShieldGemma 2	Runs on every request, so TTFT and per-call cost dominate
Tool-loop worker	Flash (`gemini-3.8-flash`)	Many short MCP or function-call turns at high token velocity
Fine-tuned specialist	Gemma 4 with LoRA, self-hosted	Fixed schema or domain syntax; weights and data stay on private capacity

- **ADK per-agent models:** each `LlmAgent` sets its own `model`, so a coordinator and its sub-agents can run different tiers in one agent tree.
- **ADK model routing (experimental):** `RoutedLlm` wraps several models behind one agent and calls a router function per request (fallback on error, A/B tests, routing by input complexity); `RoutedAgent` does the same when instructions and tools also change. If the chosen model fails before emitting output, the router is called again with error context.
- **Managed gateway routing:** API Gateway model routing accepts OpenAI-compatible requests and forwards them to Agent Platform models, replacing a self-run proxy layer.
- **Anti-pattern:** a frontier model at every node. It multiplies latency across sequential hops and spends planner-grade tokens on classification.
- **Anti-pattern:** an SLM as coordinator for open-ended goals. Weak planning causes wrong delegation, and every downstream worker inherits the error.

Customization Ladder: Prompting → Grounding/RAG → Tuning

Rung	Solves for	Key architectural rule
1. Prompt & Context Engineering	Task framing, structured JSON schemas, dynamic few-shot (**Example Store**)	Always establish an eval-measured prompt baseline first (and use **Agent Optimizer** for automated prompt tuning)
2. Grounding / RAG	**Missing, private, or volatile knowledge** + verifiable citations	Never fine-tune to inject changing facts or ACL-protected documents; use **Agent Search**, **RAG Engine**, or Search Grounding
3. Supervised Fine-Tuning (SFT)	**Consistent behavior**: strict domain syntax, terse classification labels, latency/token reduction by replacing huge few-shot prompts	Supported via LoRA adapter tuning on eligible Gemini 2.5/3.x and Gemma models (**not supported on** `gemini-4-argon`)

Rung	Solves for	Key architectural rule
4. Preference Tuning (RLHF/DPO)	Subtle tone, helpfulness, or stylistic trade-offs hard to specify as exact targets	Run after SFT using `(prompt, preferred, dispreferred)` pairs

Consumption Tiers: PayGo, Flex, Batch & Provisioned Throughput

Tier	Target workload	Behavior & exam nuances
Standard PayGo	Variable interactive traffic	Shared capacity; subject to `429` under regional spikes (use `global` endpoint where allowed)
Priority PayGo	Latency-critical bursts without term lock-in	Per-request priority header on shared pools (supported on `gemini-4-argon`, `3.8-flash`, `nano-banana-2.1`); can absorb PT overflow
Flex PayGo	Synchronous but latency-tolerant jobs (evals, bulk tagging)	Discounted token rate in exchange for variable latency and higher preemption
Batch Inference	Massive asynchronous offline pipelines	Deepest token discount (typically ~50% off Standard) with asynchronous turnaround
Provisioned Throughput (PT)	Mission-critical production needing guaranteed QPS/TPM & p95 SLA	Reserve **GSUs** for a fixed non-cancelable term (1 wk, 1 mo, 3 mo, 1 yr). **Spills over to PayGo by default**; send `X-Vertex-AI-LLM-Request-Type: dedicated` to cap spend strictly (`429` on overage) or `shared` to bypass PT

SaaS vs. Self-Hosted: who owns the capacity

Dimension	Managed API, PayGo	Managed API, Provisioned Throughput	Self-hosted (Model Garden endpoint or GKE)
Unit of spend	Tokens consumed	GSUs reserved for a term	Accelerator node-hours
Idle cost	None	Full commitment, used or not	Full while replicas run
Burst behavior	Shared pool, `429` risk under spikes	Reserved throughput; overage spills to PayGo unless `dedicated`	Limited by your replicas and autoscaling speed
Ops owner	Google	Google	You (containers, scaling, patching, guardrails)
Model scope	Gemini, partner and open MaaS catalog	Supported Gemini and partner models (partner PT ordered through Google Cloud sales)	Any open or custom weights you can serve

- **PT vs. self-hosting:** both buy predictability. PT keeps the managed model and its update path; self-hosting buys weight control and private-only execution at the price of running the serving stack.
- **Anti-pattern:** self-hosting a large open model for spiky, low-volume traffic. Idle node-hours usually cost more than PayGo tokens; start on MaaS and move once utilization is sustained.

Choose this when

- **Frontier coordinator with Flash or Flash-Lite workers:** an enterprise orchestrator fans out to many parallel extraction or tool calls and only the planning step needs deep reasoning.
- **Regional endpoint with Provisioned Throughput:** a regulated, steady, high-QPS chat workload needs in-region processing, CMEK on stored resources and predictable throughput.
- **Multi-region `us` or `eu` endpoint:** residency is defined at jurisdiction level and availability matters more than single-region pinning.
- **Global endpoint on PayGo:** prototypes or non-regulated workloads with bursty traffic and no residency clause.
- **Claude on Agent Platform:** a team standardized on Claude for coding agents but must keep billing, IAM, audit logs and the VPC-SC perimeter in Google Cloud.
- **Gemma on a Model Garden private endpoint:** fine-tuned open weights must be reachable only over PSC and nobody wants to operate Kubernetes.
- **Gemma on private GKE with vLLM:** many LoRA specialists share accelerator pools, utilization is sustained, and a platform team already runs GKE.
- **Google Distributed Cloud air-gapped:** the mandate forbids any connection to a public cloud region.

05. Gemini 4 Argon, Gemini 3.x & Context Mechanics

Deep Dive: Gemini 4 Argon (`gemini-4-argon`) Mechanics & Constraints

Released GA on **October 13, 2026**, **Gemini 4 Argon** introduces architectural behaviors that differ sharply from Gemini 2.5 and 3.x — **high-yield exam territory**:

- **Unified 1M Token Window (`1,048,576` tokens):** Unlike earlier models with separate input and output pools, `gemini-4-argon` shares a **single unified `1,048,576`-token budget** across **input prompt + internal thinking tokens + function calls + generated output**.

- **Long Decodes up to 1M Tokens:** Default single synchronous response cap is **`65,536` tokens** (`32,768` on non-streaming HTTP), **extendable all the way to `1,048,576` tokens (minus input)** for whole-codebase generation or exhaustive legal/scientific synthesis using: 1. `continuation_token` loops on `GenerateContent` (`v1beta1`), **or** 2. Server-side `background=true` execution via the **`Interactions API`** (`client.interactions.create` with `previous_interaction_id`).
- **Strict `thinking_level` Validation (`LOW` | `MEDIUM` | `HIGH`):** Defaults to **`MEDIUM`** . Passing legacy integer **`thinking_budget`** OR passing **`thinking_level="MINIMAL"`** on `gemini-4-argon` fails immediately with **HTTP `400 INVALID_ARGUMENT`** ! (Note: `MINIMAL` is also rejected on `gemini-3.8-flash` and `gemini-3.7-flash` , but is supported on `gemini-3.6-flash` and `gemini-3.5-flash*` ; integer `thinking_budget` belongs to Gemini 2.5, where `2.5 Pro` cannot be set to `0` .)
- **Sampling Parameters Ignored:** Custom `temperature` , `top_p` , and `top_k` values are **ignored** on `gemini-4-argon` (reasoning calibration is managed internally).
- **Implicit Context Caching Only:** `gemini-4-argon` uses **automatic durable implicit caching** (minimum **4,096 tokens**, zero storage fee, up to 95% input discount on Gemini API). **Explicit `cachedContents` creation, Managed Tuning, Live API WebSockets, and direct RAG Engine generator binding are NOT supported on `gemini-4-argon` .**
- **Native Agentic Capabilities:** Supports **GA Agentic Video Understanding** (actively zooming/re-sampling video regions during reasoning), native **Computer Use** (GUI screen action loops), Code Execution, MCP/Function Calling, URL Context, and Search/Maps/Agent Search/Enterprise Web Grounding.
- **Gemini 4 Argon Cyber Permissive (`gemini-4-argon-cyber-permissive`):** Gated behind Google DeepMind's **Fairwind Program** allowlist (`global` and `us` endpoints) for authorized defensive cybersecurity teams performing vulnerability discovery, reverse engineering, exploit verification, and automated patching with **CodeMender** (`cm find / verify / fix --model gemini-4-argon`). Defaults to `HIGH` thinking and **disables Computer Use** as a containment control.

`GenerateContent` vs. Stateful `Interactions API` (Preview)

Capability	`GenerateContent` API	`Interactions API` (`client.interactions.create`)
State model	Client-managed stateless request (must resend conversation history + **Thought Signatures** each turn)	Optional **server-side state** via `previous_interaction_id` — preserves multi-turn context and reasoning state automatically
Long-horizon async	Synchronous / SSE (`continuation_token` needed past 65K output tokens)	Native **`background=true`** mode for multi-minute/multi-hour autonomous decodes and **Managed Agents** (`Antigravity` , `Deep Research`)
Best fit	Low-latency custom orchestrators, fine-grained context compaction	Long-decode synthesis, stateful multi-turn agents, and hosted Managed Agents

Thought Summaries, Thought Signatures & Context Caching Across Generations

- **Thought Summaries & Thought Signatures:** Set `include_thoughts=True` to receive human-readable reasoning summaries for UI observability; pass opaque cryptographic **Thought Signatures** back in multi-turn `GenerateContent` tool loops so the model retains its latent reasoning chain without re-deriving steps.
- **Implicit vs. Explicit Caching Comparison:**
- **Implicit Caching:** Zero config, zero hourly storage charge; triggers automatically on shared prompt prefixes (`⩾4,096` tokens on Gemini 4 Argon). Keep static instructions and tool definitions at the top of the prompt!
- **Explicit Caching (`cachedContents`):** Supported on Gemini 2.5 and eligible Gemini 3.x base models (not fine-tuned models, **not Gemini 4 Argon**); bills hourly storage for a deterministic TTL cache handle.
- **ZDR & Multi-Tenant Safety:** Caches are isolated per project and honor VPC-SC, but disable caching when strict **Zero Data Retention (ZDR)** applies, and never share explicit cache handles across tenants with distinct data entitlements.

Multimodal Tokenization Mechanics

- **Text:** 256,000-token **SentencePiece** vocabulary.
- **Images:** Dynamic tiling into **768×768** pixel tiles (**258 tokens per tile**), plus configurable media resolution controls on Gemini 3.x/4 to trade detail vs. token budget.
- **Video:** Sampled visual frames + audio track (**~263 tokens/sec at 1 FPS** standard resolution; compressible to **~66–70 tokens/sec** in low-resolution mode for multi-hour recordings), with **GA Agentic Video Understanding** dynamically inspecting high-res regions on demand.
- **Audio:** Compact **32 tokens per second** (~1,920 tokens/minute) interleaved with timestamp markers (`MM:SS`) for exact temporal grounding.

06. Generative Media Models (Images, Video & Music)

Nano Banana 2.1 / Pro & Imagen 3 / 4 (Visual Synthesis & Editing)

- **Nano Banana Family — Native Multimodal Image Generation (Recommended Default):**
- **Nano Banana 2.1 (`gemini-nano-banana-2.1` , GA + Priority PayGo):** Flagship native Gemini image generator and conversational editor, superseding standalone `Imagen 4` on the Gemini API.
- **Nano Banana 2 (`gemini-3.1-flash-image`) & Nano Banana 2 Lite (`gemini-3.1-flash-lite-image`):** High-speed, cost-efficient native image generation and mask-free multi-turn editing.
- **Nano Banana Pro (`gemini-3-pro-image`):** Studio-grade **4K** generation with **131,072-token** context, search-grounded factual infographics/typography, and multi-reference subject consistency (up to **14 reference images**: e.g., 5 character + 6 object + 3 style references).
- **Imagen 3 (`imagen-3.0-generate-002`) & Imagen 4 (`imagen-4.0-generate-001` / `ultra` / `fast` on Agent Platform):** Specialized latent-diffusion text-to-image endpoints (480-token prompt window, up to 2K) with **SynthID** watermarking and **C2PA Content Credentials** (note: standalone Imagen 4 is retired on the developer Gemini API in favor of Nano Banana 2.1, while remaining available on enterprise Agent Platform).

Gemini Omni 1.1 Flash & Veo 3.1 (Video + Native Audio Generation)

- **Gemini Omni 1.1 Flash (`gemini-omni-1.1-flash`):** Fast native multimodal video generation, instruction-driven video editing, keyframe interpolation, and clip extension with synchronized native audio (replaces legacy Veo 3.1 preview endpoints on the Gemini API).
- **Veo 3.1 (`veo-3.1-generate-001` , `veo-3.1-fast-generate-001` , `veo-3.1-lite-generate-001` — GA on Agent Platform): Joint latent diffusion** compressing video pixels and audio waveforms into a shared spatio-temporal latent space — producing 4K 16:9 / 9:16 cinema-grade clips with frame-accurate lip-sync, Foley sound effects, and object permanence across camera pans (succeeding silent **Veo 2**).

Lyria 3.5, Lyria 3 & RealTime (Music Generation)

- **Temporal Latent Diffusion:** Compresses 44.1–48 kHz stereo PCM audio into compact latents and denoises structured musical waveforms without autoregressive long-horizon drift.
- **Active Lineup: Lyria 3.5 (`lyria-3.5`)** (flagship full-length songs with multi-section structural coherence and multilingual vocals), **Lyria 3 Pro** (184s compositions), **Lyria 3 Clip** (30s rapid hooks), **Lyria RealTime (`lyria-realtime-exp`)** (interactive live-steered music streaming), and **Lyria 2 (`lyria-002`)** — all protected by **SynthID-Audio**, C2PA, and artist vocal-likeness/recitation filters.

07. Live Voice Agents, Speech (3.8 TTS / 3.5 Transcribe / Chirp 3) & Embeddings

Gemini Multimodal Live API (`gemini-3.8-live`)

- **Stateful Bidirectional WebRTC / WebSockets:** Persistent low-latency streaming sessions ingesting continuous **16-bit PCM audio**, live camera/screen frames, and text while emitting native **24 kHz audio**.
- **Active Live Models:**
- **Gemini 3.8 Live (`gemini-3.8-live`):** Default sub-second voice & multimodal agent model with server-side VAD **barge-in**, affective prosody adaptation, and spoken function calling (**71.5%+ ComplexFuncBench Audio**).
- **Gemini 3.8 Live Extended Thinking (`gemini-3.8-live-extended-thinking`):** Performs asynchronous high-depth background reasoning while maintaining natural conversational responsiveness during live voice sessions.
- *(Note: `gemini-4-argon` does not support the Live API — pair `gemini-3.8-live` at the voice front-door with `gemini-4-argon` sub-agents for deep backend tasks!)*

Dedicated Speech Synthesis, Transcription & Live Translation

- **Gemini 3.8 Flash TTS (`gemini-3.8-flash-tts`) & 3.8 Flash-Lite TTS (`gemini-3.8-flash-lite-tts`):** Studio-fidelity controllable text-to-speech across **130 languages** (101 on Flash-Lite TTS) with natural-language style/voice design and low-latency streaming.
- **Gemini 3.5 Transcribe (`gemini-3.5-transcribe` / `gemini-3.5-transcribe-live`):** Dedicated low-latency speech-to-text with speaker diarization, per-utterance multilingual switching, and domain vocabulary biasing.
- **Gemini 3.5 Live Translate (`gemini-3.5-live-translate-preview`):** Real-time speech-to-speech translation preserving speaker prosody across **70+ languages**.
- **Chirp 3 (`chirp_3` in Cloud Speech-to-Text V2):** Built on Google's **2B+ parameter Universal Speech Model (USM)** (12M+ hours, 100+ languages) for enterprise compliance pipelines needing `Streaming Recognize` / `BatchRecognize` , **Model Adaptation** (up to 1,000 custom domain phrases), `denoise_audio=True` , word-level timestamps, and deterministic redaction before LLM ingestion.

Gemini Embedding 2 & `gemini-embedding-001`

- **Gemini Embedding 2 (`gemini-embedding-2` / `gemini-embedding-2-preview`):** Google's **first natively multimodal Gemini embedding model**, mapping **text, code, images, audio, video, and PDFs** directly into a single unified vector space — enabling cross-modal retrieval (e.g., matching a smartphone photo of a faulty valve to both a PDF schematic and a repair-video timestamp) without captioning pipelines.
- **Matryoshka Representation Learning (MRL):** Both `gemini-embedding-2` and text-focused `gemini-embedding-001` (up to 3,072 dimensions) order semantic signal from index `0` onward, letting architects truncate `output_dimensionality` (e.g., `1536` , `768` , `512` , `256`) with automatic L2 normalization to cut Vector Search / AlloyDB RAM by 2–12× at minimal recall loss.

08. Gemma 4 & Open-Weight Edge Architecture

The **Gemma** family brings Google DeepMind architectures to open weights for Model Garden MaaS, GKE/dedicated endpoints, air-gapped on-prem servers, and **Google AI Edge** (`LiteRT-LM` / MediaPipe).

Gemma 4, Gemma 3n & Gemma 3 Evolution

- **Gemma 4 (`E2B` , `E4B` , `26B A4B MoE` , `31B`):** Latest generation open-weight family combining edge-optimized **Per-Layer Embeddings (PLE)** on `E2B` / `E4B` (processing **text, image, video, and native audio** in a 2B–4B effective RAM footprint), a high-efficiency **26B Sparse MoE activating only 4B parameters per token (`26B A4B`)** for low-latency self-hosted agentic tool loops, and a dense **31B** reasoning checkpoint.
- **Gemma 3 (1B, 4B, 12B, 27B) & Gemma 3n (E2B, E4B):** Introduced **128K context** (140+ languages), **SigLIP** 400M vision encoding (256 soft tokens per **896×896 Pan & Scan** crop), **5:1 local-to-global interleaved attention** (5 sliding-window 1,024-token layers per 1 global layer to shrink KV-cache RAM), and **MatFormer** nested sub-networks.
- **T5Gemma & RecurrentGemma (Griffin):** Encoder-decoder open checkpoints (`T5Gemma`) for sequence-to-sequence tasks, alongside **Griffin (`RG-LRU`)** constant-memory linear recurrence for ultra-long edge streaming.

Specialized Gemma Ecosystem Variants

- **ShieldGemma 2 (4B) & ShieldGemma (2B, 9B, 27B):** Open text + image safety classifiers scoring prompts, completions, and visual inputs against harm taxonomies (ideal as fast local guardrails around self-hosted open models).
- **MedGemma (4B, 27B) & MedSigLIP (800M):** Clinical text, EHR summarization, and radiology/pathology/ophthalmology multimodal comprehension (the open-weight successor alongside Gemini to legacy **MedLM**).

- **TxGemma (2B, 9B, 27B):** Therapeutic small-molecule, protein, and clinical-trial prediction models.
- **PaliGemma 2 (3B, 10B, 28B), CodeGemma (2B, 7B) & FunctionGemma:** Specialized VLM fine-tuning (bounding boxes/segmentation), IDE Fill-in-the-Middle (FIM), and compact tool-calling adapters.

09. Safety, Provenance & Frontier Governance

SynthID Imperceptible Watermarking + C2PA

Metadata manifests (**C2PA Content Credentials**) provide rich cryptographic provenance chains when intact, but are stripped by screenshots or re-encoding. **SynthID** embeds imperceptible signatures directly into generated signals for defense-in-depth:

- **SynthID-Text:** Seeded cryptographic **g-function tournament sampling** over token logits without degrading perplexity or latency.
- **SynthID-Audio:** Psychoacoustic spectral watermarking surviving MP3 compression, pitch/tempo shifts, and the **"analog hole"** (speaker playback re-recorded via microphone).
- **SynthID-Image & Video:** Spatial-temporal latent perturbation resilient to cropping, color grading, and lossy transcoding.

DeepMind Frontier Safety Framework (FSF) & Enterprise Guardrails

- **Critical Capability Levels (CCLs):** Empirical pre-deployment evaluations across **CBRN uplift**, **Autonomous Cyber-Offense** (governing why `gemini-4-argon-cyber-permissive` is gated behind the **Fairwind Program** with Computer Use disabled), **Autonomous ML R&D / self-improvement**, and **Deceptive alignment / oversight circumvention**.
- **Coupled Mitigations: Security Mitigations** (weight exfiltration hardening) + **Deployment Mitigations** (safety tuning, runtime classifiers, allowlisted access).
- **Enterprise Runtime Defense:** Combine Gemini `SafetySetting` harm thresholds with **Vertex AI Model Armor** (prompt injection, jailbreak, malicious URL, and SDP PII screening) and **ShieldGemma 2**.

10. Scientific, Embodied & Agentic Discovery Systems

Multi-Agent Scientific & Algorithmic Discovery (Agentic Frontier)

- **AI Co-Scientist:** Multi-agent scientific research system built on Gemini where specialized agents (**Generate → Reflect → Rank → Evolve → Proximity → Meta-review**) compete in an Elo-rated self-play tournament over asynchronous tasks to synthesize novel, experimentally validated biomedical and scientific hypotheses (e.g., drug repurposing and epigenetic targets).
- **AlphaEvolve:** Evolutionary coding agent pairing Gemini's creative code synthesis with **automated deterministic evaluators** in an evolutionary loop — discovering provably correct, novel algorithms for matrix multiplication (surpassing Strassen's 1969 4×4 complex bound), open mathematical problems, and Google data-center Borg scheduling / TPU Verilog kernel optimizations.
- **CodeMender:** Autonomous defensive security agent powered by **Gemini 3.8 Flash** (default) and **Gemini 4 Argon** (`cm find`, `cm verify`, `cm fix`) that discovers vulnerabilities, proves root causes with sanitizers/fuzzers, and generates regression-tested patches.

Structural Biology, Mathematics, Robotics & World Models

- **AlphaFold 3:** Replaces Evoformer with a 48-block **Pairformer**, **atomic-level tokenization** for ligands/ions, and a **3D Conditional Diffusion Module** predicting joint complexes across proteins, DNA, RNA, small molecules, and covalent modifications.
- **AlphaProteo & AlphaMissense:** *De novo* high-affinity protein binder generation (3×–300× affinity uplift) and pathogenicity classification across 71M human missense variants.
- **AlphaGeometry 2:** Neuro-symbolic IMO gold-standard prover pairing a **Gemini** intuitive auxiliary-construction generator ("System 1") with the **DDAR** symbolic deduction verifier ("System 2").
- **Gemini Robotics ER 2 (`gemini-robotics-er-2-preview`):** Next-generation **Embodied Reasoning** Vision-Language-Action model succeeding `Gemini Robotics ER 1.6` (retired) and RT-2 — performing 3D spatial pointing, grasp/trajectory planning, task progress estimation, and physical safety reasoning for dexterous robots.
- **Genie 2 / 3 & SIMA 2:** Interactive physics-consistent 3D world generation from single prompts (`Genie`) and generalist language-following 3D virtual agents (`SIMA 2`).
- **GraphCast / GenCast, AlphaChip, AlphaQubit, AlphaTensor & AlphaDev:** GNN/diffusion global weather ensembles, RL TPU floorplanning, surface-code quantum error decoding, and RL-discovered matrix/LLVM sorting algorithms.