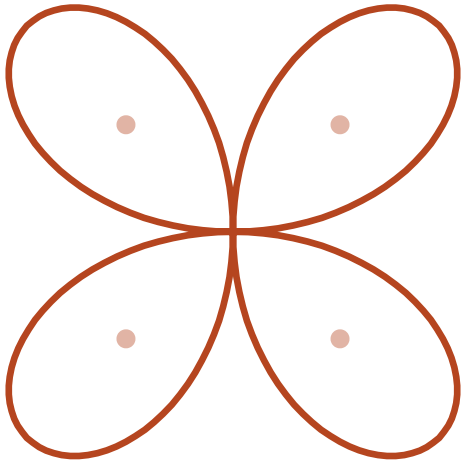
TECHNICAL TRAINING தொழில்நுட்பப் பயிற்சி

AI Systems Engineering

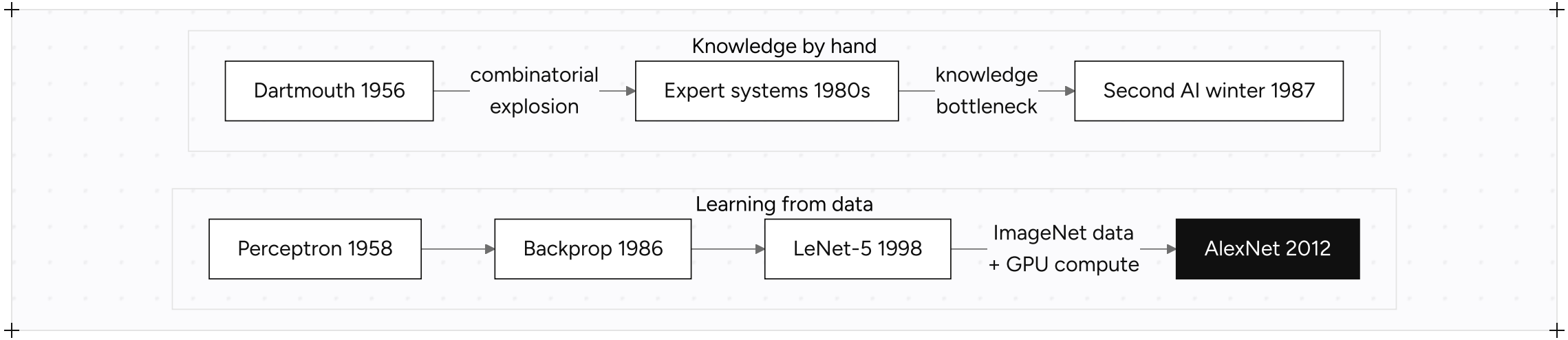


A large language model (LLM) is a probabilistic component in a deterministic system you own.

[01]	[02]	[03]	[04]
Foundations	LLM mechanics	Agentic systems	Production

From Dartmouth (1956) to deep learning (2012)

Learning from data dates to the 1950s. It won in 2012, once data and compute caught up.



[MILESTONES]

- **1969** *Perceptrons* shows single-layer limits
- **1973** Lighthill report; the first AI winter follows
- **2012** AlexNet: 15.3% top-5 error vs 26.2% runner-up

[THE LESSON FOR ENGINEERS]

- Hand-built rules capped expert systems
- Hand-built features capped statistical ML
- Hand-built workarounds cap upgrades

Keep per-model workarounds thin and eval-covered, so a model upgrade is a config change plus an eval run.

Rules vs discriminative vs generative

How behavior enters a system decides how you test it and how it fails.

	RULES	DISCRIMINATIVE ML	GENERATIVE MODEL
Built from	Written by experts	Labeled examples	Pretraining, post-training, then your prompt
What it models	Explicit logic	$P(y \mid x)$, a decision boundary	$P(x_t \mid x_{<t})$, the next token given context
How you test	Exact asserts	Holdout metrics, then drift monitoring	Graded evals, then sampled live review
Typical failure	Brittle on unseen cases	Silent decay under distribution shift	Plausible but wrong; varies by run and version
Example	Eligibility rules engine	Fraud or spam score	Drafted support reply

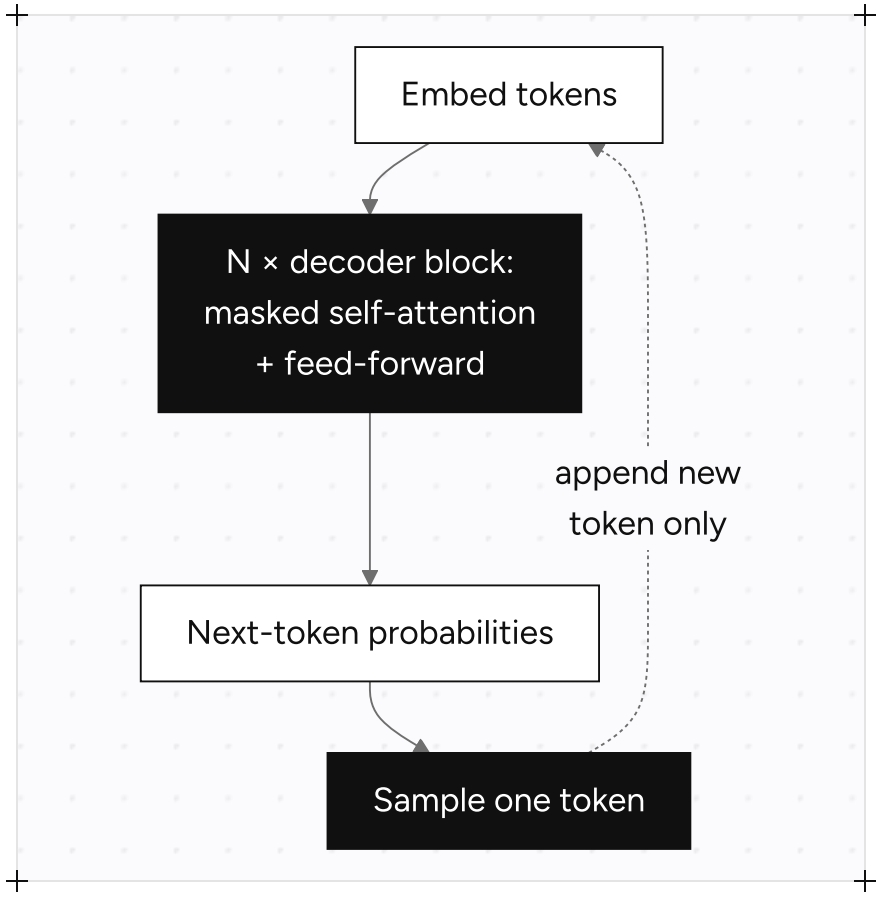
Treat generated output as untrusted: validate structure on every response, gate releases on eval scores. Baseline a prompted model before training a classifier.

The Transformer: attention and generation

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

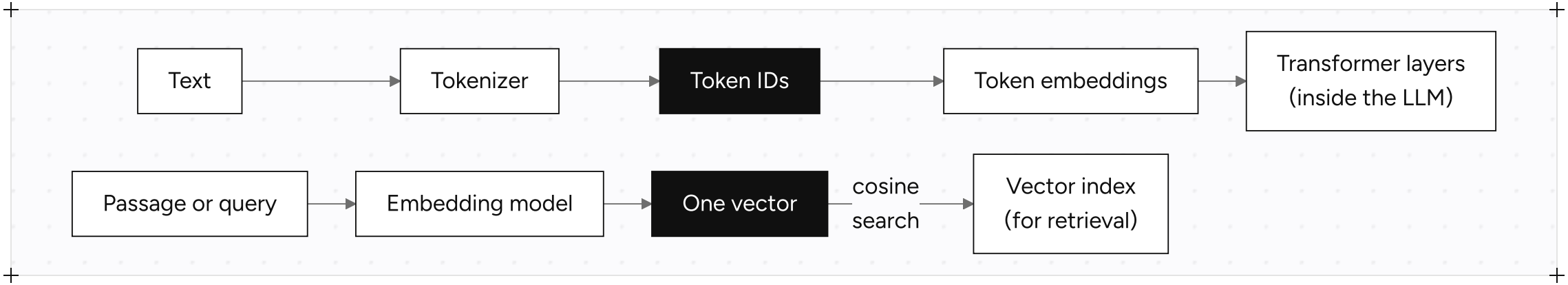
Each token's query scores every visible token's key; softmax turns the scores into weights that mix the values (Vaswani et al., 2017).

- **Seq2seq encoder-decoder:** the encoder reads the source; the decoder writes the target via cross-attention
- **Decoder-only:** prompt and answer share one sequence; a causal mask hides later tokens
- **Position:** sinusoids in 2017, RoPE in many models today
- **Parallel training:** attention compute is $O(n^2)$ in length
- **KV cache:** reuses prefix keys and values, at a GPU memory cost



Output is appended token by token and never revised: cap its length, and request reasoning before the verdict.

Tokens, embeddings, and vector space



[TOKENS]

- Subword units from each model's tokenizer (BPE or similar)
- English: about 3 to 4 characters each; varies by tokenizer
- Translated text can differ up to 15x in tokens (Petrov et al., 2023)

[EMBEDDINGS]

- A separate model gives one vector per passage
- Near means same topic, not same fact: negations can score close
- Approximate indexes trade recall for speed: measure recall@k

[ENGINEERING RULES]

- Use the provider's token counter, never string length
- Model change: re-embed, unless a shared space is documented
- Record the model version on every vector

Tokens are the unit of billing, latency, and limits. Vectors compare only within one embedding space.

Context windows and sampling

[CONTEXT WINDOW]

- One budget: system prompt, tools, history, documents, output
- Replies cut at max output tokens raise no error: check the stop reason
- The model is stateless: each call bills the full context, even if the API stores it

[SAMPLING WITH TEMPERATURE T]

$$p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

- Lower T sharpens, higher T flattens, T near 0 is near greedy
- Top-p samples from the smallest set whose cumulative probability reaches p (Holtzman et al., 2020)
- Tune temperature or top-p, not both; many current models reject non-default values

Low T for extraction only where model docs advise it; otherwise keep the default. T = 0 still varies: validate.

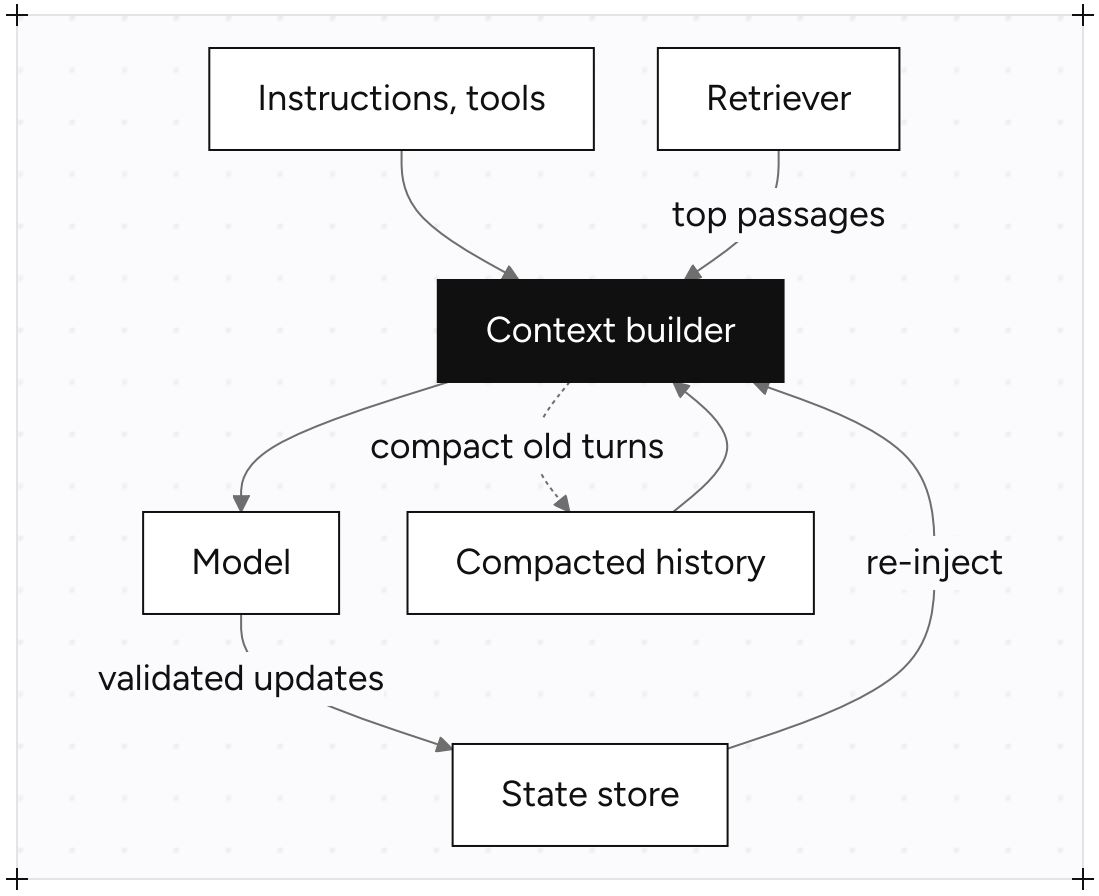
Context limits and context drift

[WHAT GOES WRONG]

- Effective below advertised: 11 of 13 models lost over half their score at 32K (Modarressi et al., 2025)
- Mid-context evidence used worst (Liu et al., 2023)
- Drift: old turns and stale facts dilute instructions
- Over the limit, hosted APIs error; frameworks and local servers may truncate silently

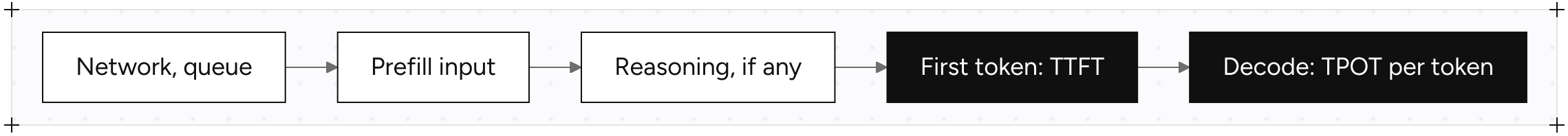
[DESIGN PATTERN]

- Token budget per section, plus output reserve
- Stable first: instructions, tools, history; then state, evidence, constraints
- Trim tool results to needed fields before history



The window is capacity, not quality: budget every section and test at your real length and turn count.

Latency and cost: TTFT vs TPOT



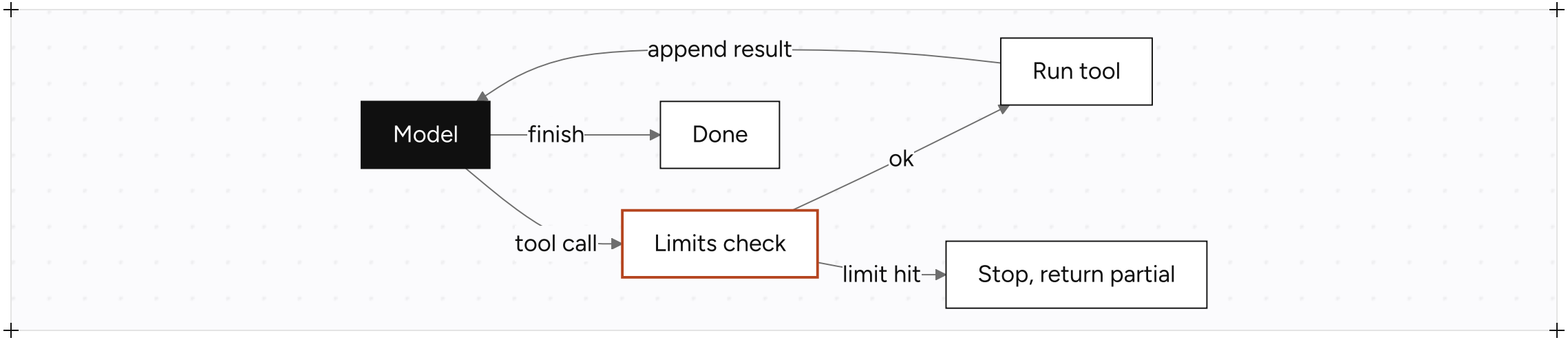
- **TTFT** (time to first token): queue, prefill, hidden reasoning; grows with input
- **TPOT** (time per output token): mean gap between tokens; grows with model, context, batch size
- Latency ≈ TTFT + TPOT × (visible output tokens − 1)
- Cost ≈ input × input price + output × output price, before cache rates
- Output tokens, reasoning included, often cost 4 to 8x input

LEVER	CUTS	COSTS YOU
Streaming	Perceived wait	Late validation
Compact output	Decode time, cost	Detail; may truncate
Prompt caching	TTFT, input cost	Fixed prompt order
Smaller model	TPOT, cost	Quality, check evals
Batch API	About half price	Up to 24 h wait

Short answers: cut TTFT. Long answers: cut TPOT. Agent loops: cut resent input. Set p95 targets per feature.

From completion to agent loop

A completion is one call your code controls. An agent loops, and the model picks each next action.



[AGENTIC SYSTEMS · ANTHROPIC 2024]

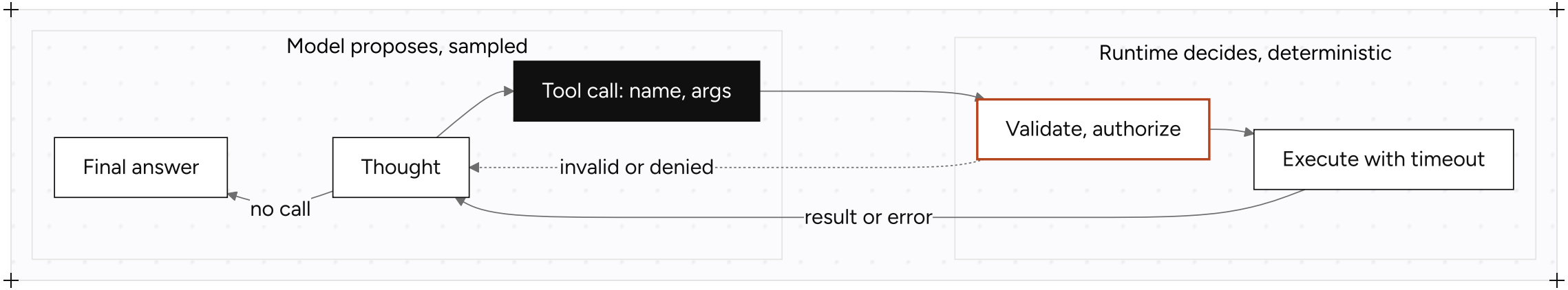
- **Augmented LLM**: model with retrieval, tools, memory
- **Workflow**: predefined code paths
- **Agent**: the model directs its own steps

[EVERY LOOP NEEDS]

- Max steps, token and wall-clock budgets
- Explicit stop conditions, scoped tool permissions
- Human approval before consequential actions

Can you code the path at build time? Ship a workflow. If not, let the model plan or loop, inside limits.

ReAct and deterministic tool calling



[TOOL DEFINITION]

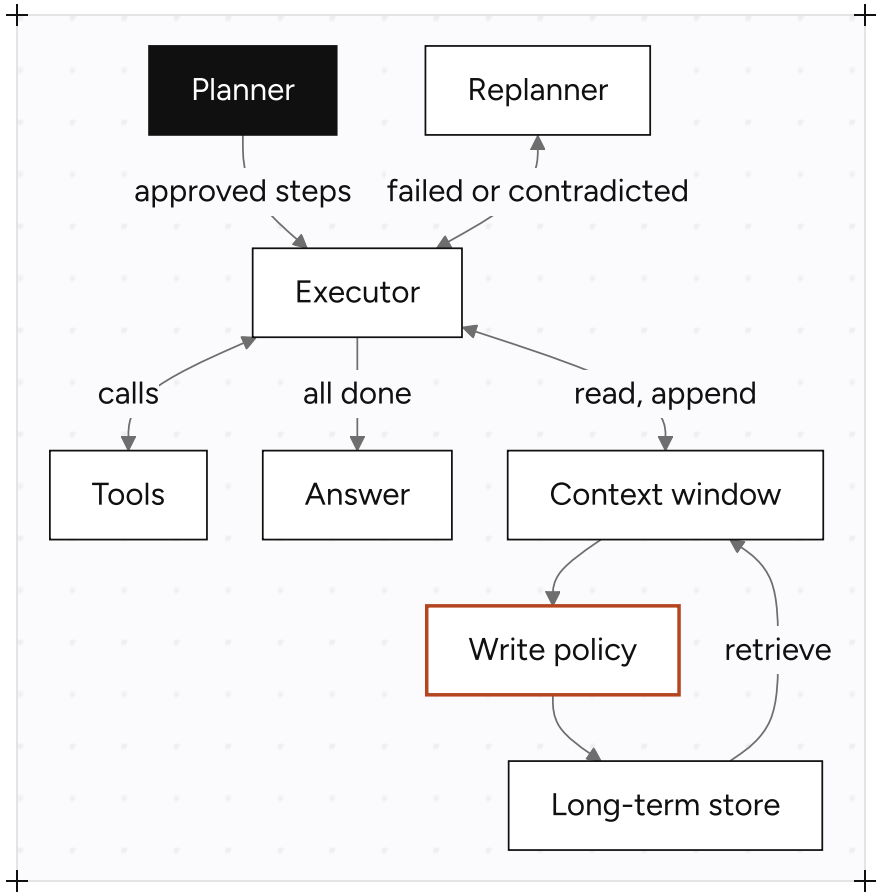
```
{"type": "function", "name": "refund_order",
 "description": "Refund a paid order in full.",
 "parameters": {"type": "object",
  "properties": {"order_id": {"type": "string"},
   "reason": {"type": "string",
    "enum": ["damaged", "late", "other"]}},
  "required": ["order_id", "reason"],
  "additionalProperties": false}, "strict": true}
```

[RUNTIME RULES]

- **ReAct** (Yao et al., 2022): thought, action, observation
- Known step? Call it from code or force the tool
- Strict mode guarantees shape, not the right call
- Idempotency key per business operation
- Errors return as observations; cap iterations

A tool definition is a contract: the model fills it in, your runtime enforces it.

Planning and memory



[PLAN, THEN EXECUTE]

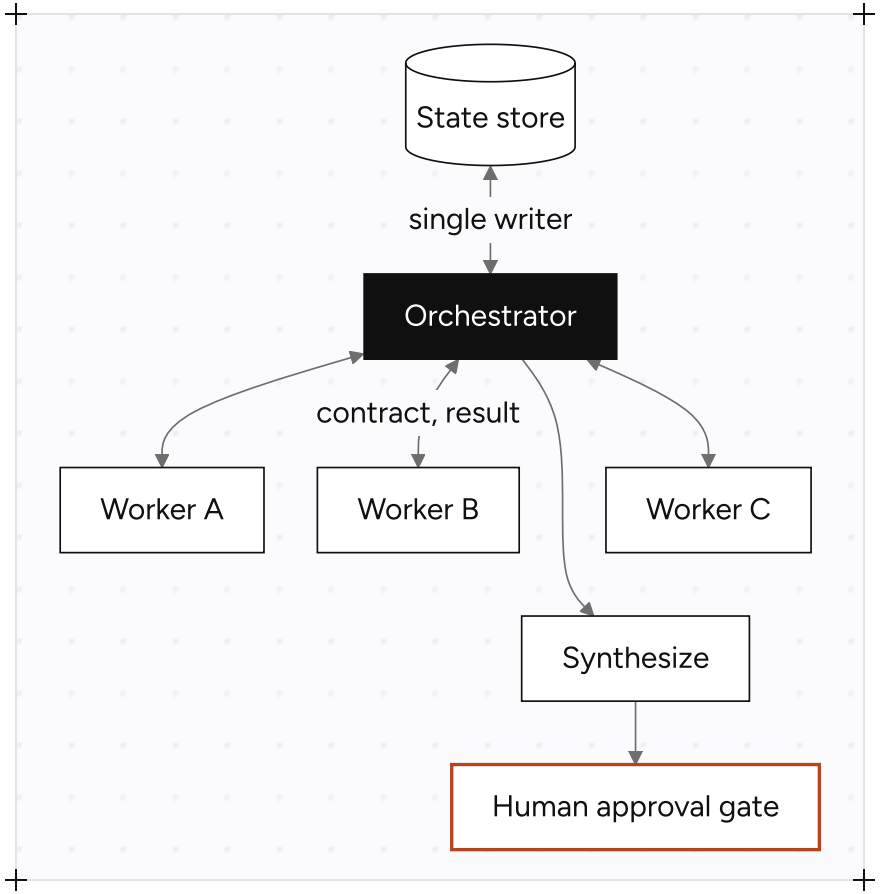
- **Plan-and-Solve** (Wang et al., 2023): one prompt, plan then solve
- **Plan-and-execute**: strong planner, cheaper executor per step
- Approve the plan; replan on a failed step or a contradiction

[MEMORY]

- **Short-term**: in the context window, billed again every call
- **Long-term**: episodic and semantic in external stores; procedural in weights and code
- Writes carry provenance, TTL, tenant key
- Risks: stale or poisoned entries, retention and deletion duties

Steps knowable when the request arrives? Plan first. Each step depends on the last result? Use ReAct.

Multi-agent orchestration



[ORCHESTRATION RULES]

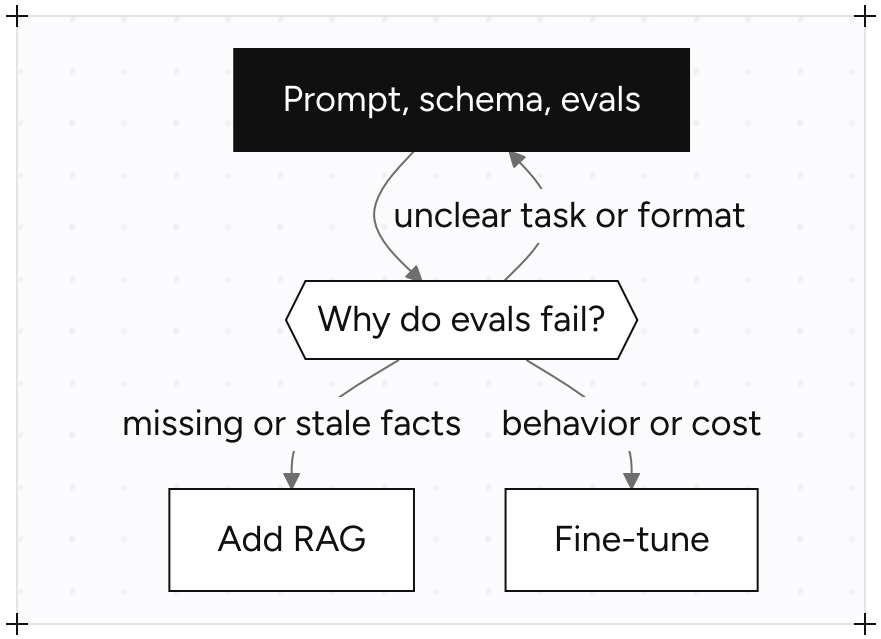
- **Why split**: focused contexts, parallel work, specialized roles
- **Patterns**: orchestrator-workers (shown), handoff to a specialist, prompt chaining, evaluator-optimizer
- **Delegation contract**: objective, inputs, output schema, allowed tools, budget, done criteria
- **State**: the orchestrator is the only writer; event log, checkpoints, idempotent steps
- **Tokens** (Anthropic, 2025): agents about 4x chat, multi-agent systems about 15x
- **Gain**: 90.2% over one agent on an internal research eval; token use alone explained 80% of BrowseComp variance

Start with one agent. Split only when work parallelizes or one context overflows.

Prompting vs RAG vs fine-tuning

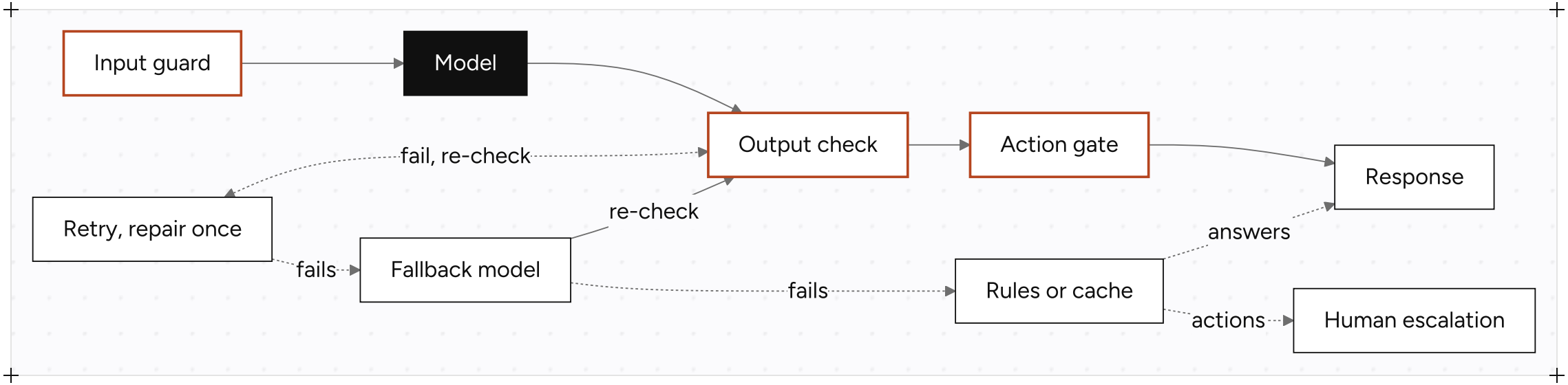
RAG, retrieval-augmented generation: embed document chunks, add the top matches to each prompt.

	PROMPTING	RAG	FINE-TUNING
Changes	Instructions, examples	Context per query	Model weights
Best for	Framing, format	Private, fresh facts; citations	Consistent behavior, narrow tasks
Data	A few examples	Corpus, ingestion pipeline	Dozens to thousands of curated pairs
Per call	Prompt tokens, cacheable	Retrieval hop, more input tokens	Shorter prompt, often pricier tokens
Risk	Prompt sprawl	Misses, stale index, permission leaks	Regressions, weak at facts, retraining



Prompt and eval first. Retrieve facts; fine-tune for behavior.

Guardrails and fallbacks



[INPUT]

- Auth, rate limits, scope checks
- Personal data (PII) redaction
- Injection screening

[OUTPUT]

- Schema, business-rule checks
- Grounding, policy classifiers
- Stop reason, truncation checks

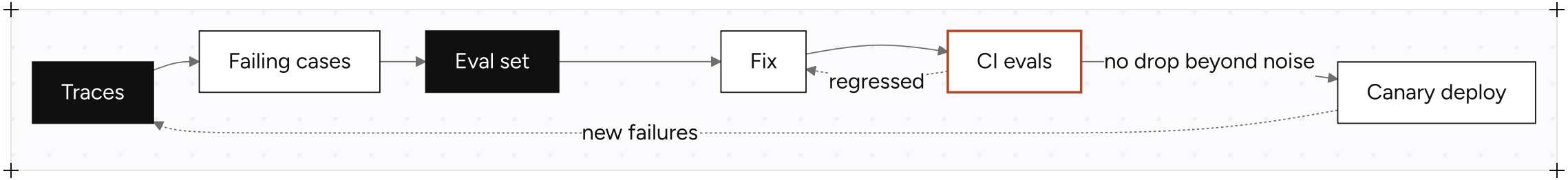
[ACTION]

- Allowlists, least privilege
- Spend caps, sandboxes
- Human approval if consequential

Treat model output and tool results as untrusted: actions fail closed, answers may degrade.

Evals and observability

Evals are the test suite for probabilistic parts. Traces are the debugger.



[OFFLINE EVALS]

- Cases from real traffic, edge cases, incidents
- Code graders first; LLM judges checked by humans
- CI: rerun each case, gate on drops beyond noise
- Agents: success, steps, tool errors, cost per task

[ONLINE EVALS AND TRACES]

- Canaries, A/B tests, user feedback, drift alerts
- Span tree: run › step › model, tool, guardrail
- Propagate trace context into tools and workers
- Log versions, tokens, TTFT, cost; redact PII

Every production failure becomes an eval case that blocks its return.