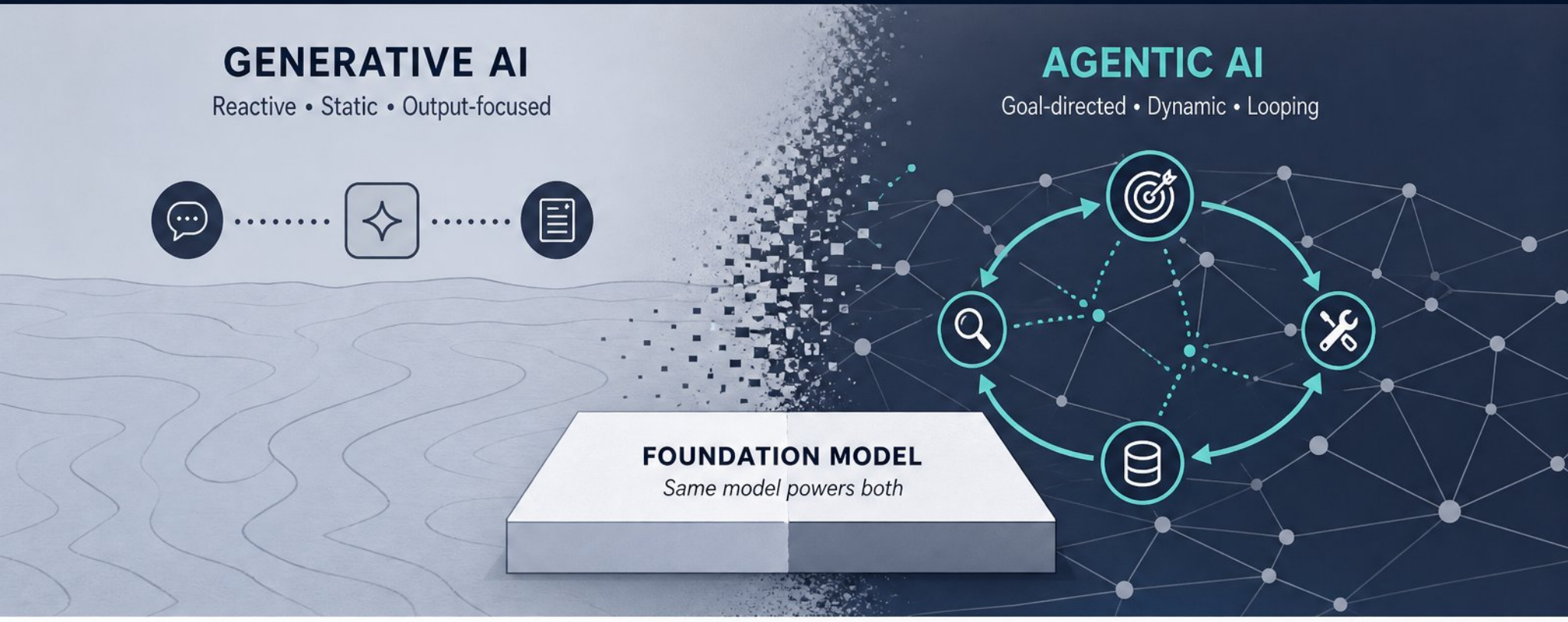
Agentic AI Vs Generative AI:

Differences, Tradeoffs, and Use Cases

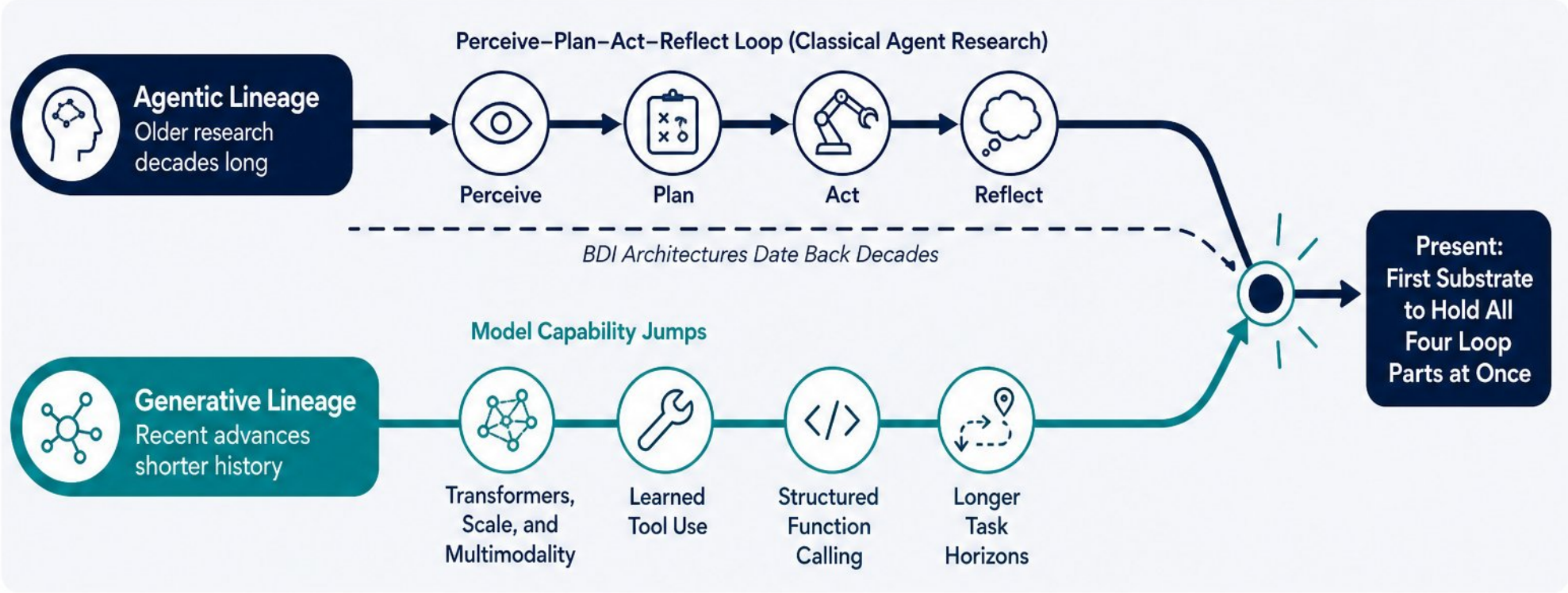
- **Generative AI:** creates content in response to prompts
- **Agentic AI:** pursues goals via planning, tools, memory, action
- **Core design question:** what may it do, and who answers for it?
- **Tradeoffs:** generation wins on speed and reviewability; agency adds latency, cost, new failure modes
- **Takeaway:** one decision framework plus a phased path to bounded autonomy

The 2026 Landscape: Why the Boundary Is Blurring



- Generative AI is now a baseline component, not a standalone category
- Plumbing advances made agents practical: reasoning models, tool calling, MCP, computer use
- Same model powers both — scaffolding, not the model, decides
- ‘Agent’ is now an industry term with reference architectures and standards
- Real question: is a model call enough, or does this need a bounded loop?

Background: From Content Generation to Goal-Directed Systems

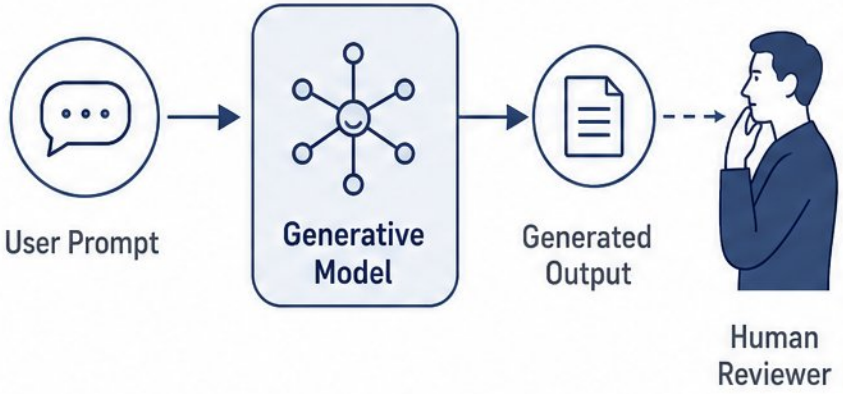


- Generative lineage: transformers, scale, and multimodality optimized for fluent single-call output
- Agentic lineage is older: perceive-plan-act-reflect loops and BDI architectures date back decades
- LLMs are the first substrate to hold all four loop parts at once
- **Enabling jumps:** learned tool use, structured function calling, longer task horizons
- **Progression:** generative models → generative agents → agentic systems with memory and multi-agent collaboration

Core Concepts: Capabilities, Autonomy, and Error Modes

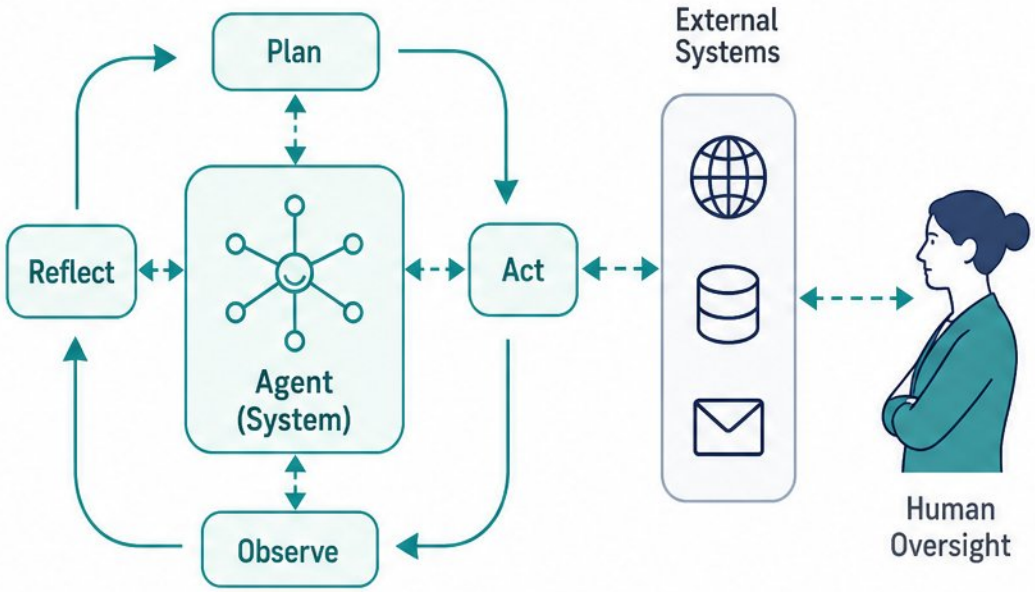
REACTIVE: GENERATIVE AI

One prompt, one output, no agency



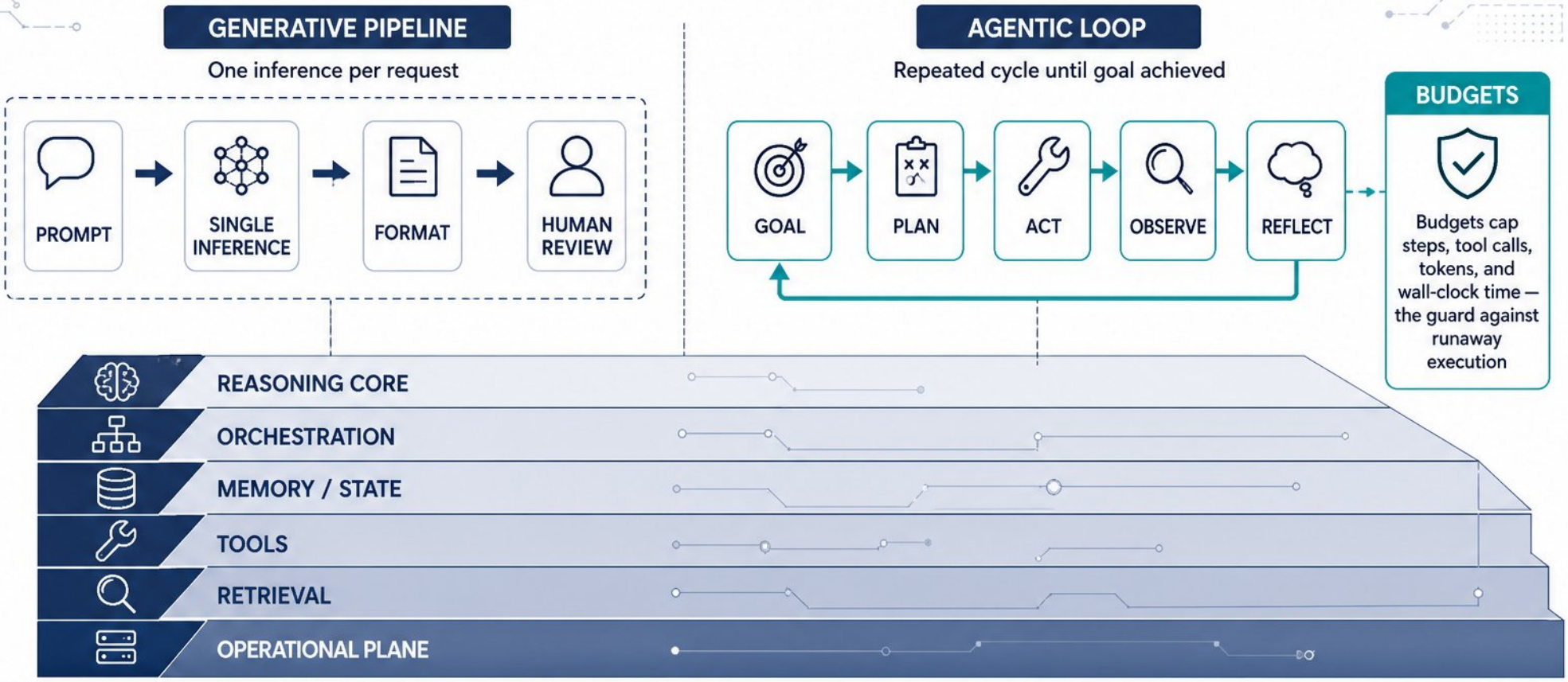
AGENTIC: AGENTIC AI

Goal-directed, stateful, planning and adjusting across steps



- Generative AI is reactive: one prompt, one output, no agency
- Agentic AI is goal-directed and stateful, planning and adjusting across steps
- Autonomy is a configurable dial, not a binary switch
- Generative failure is informational; agentic failure is operational
- Generation is a deliverable; agency is a process with outcomes

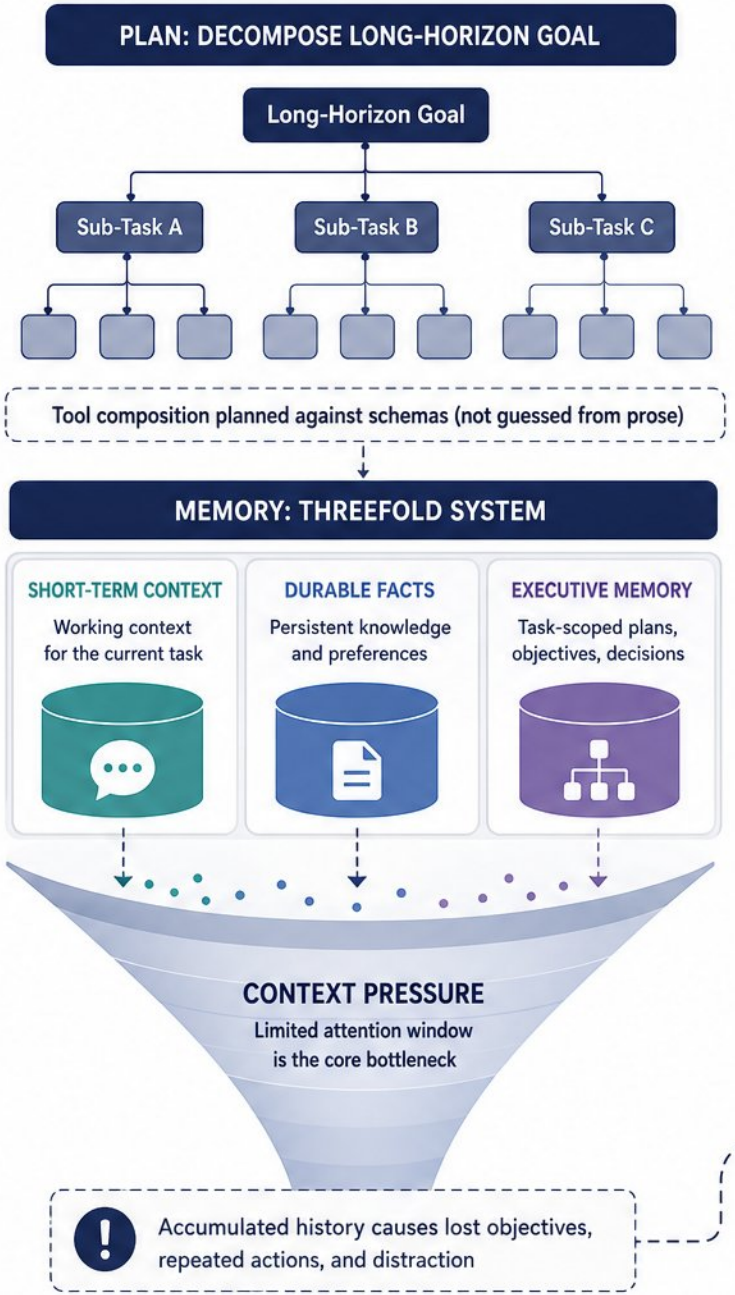
Architecture and Mechanics: How the Two Systems Are Built

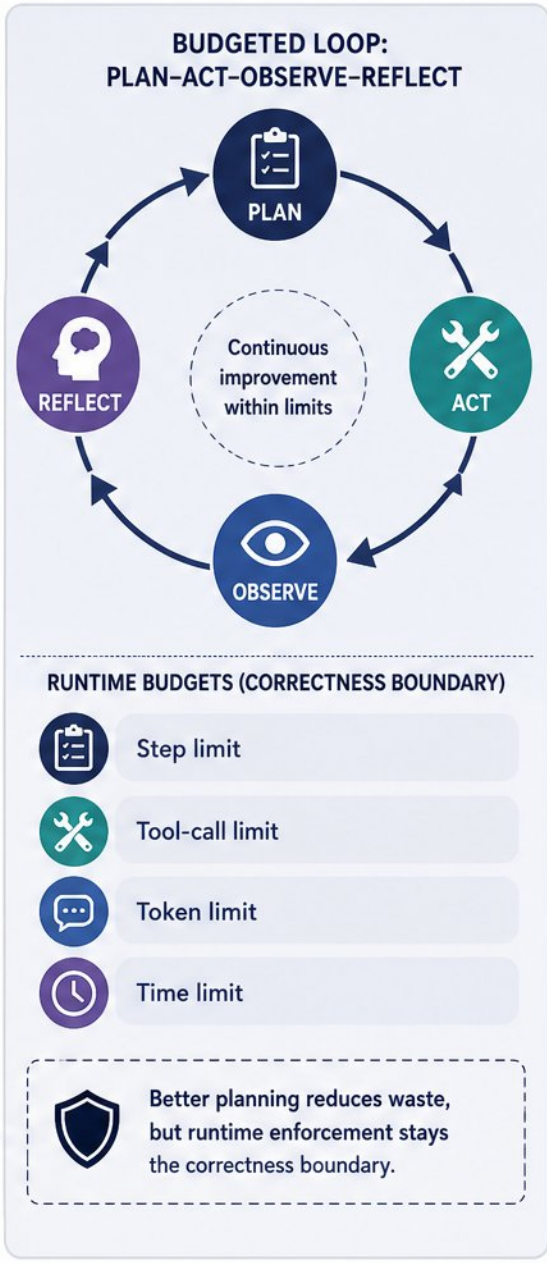


- Generative pipeline: prompt → single inference → format → human review
- Agentic loop: goal → plan → act → observe → reflect, repeat until done
- Budgets cap steps, tool calls, tokens, and wall-clock time — the guard against runaway execution
- Six recurring layers: reasoning core, orchestration, memory/state, tools, retrieval, operational plane
- Decomposition choice: static graphs win determinism and cost; dynamic plans win task breadth
- Nondeterminism makes retries, error handling, and explicit state first-class design concerns

Memory and Planning: What Actually Makes an Agent Capable

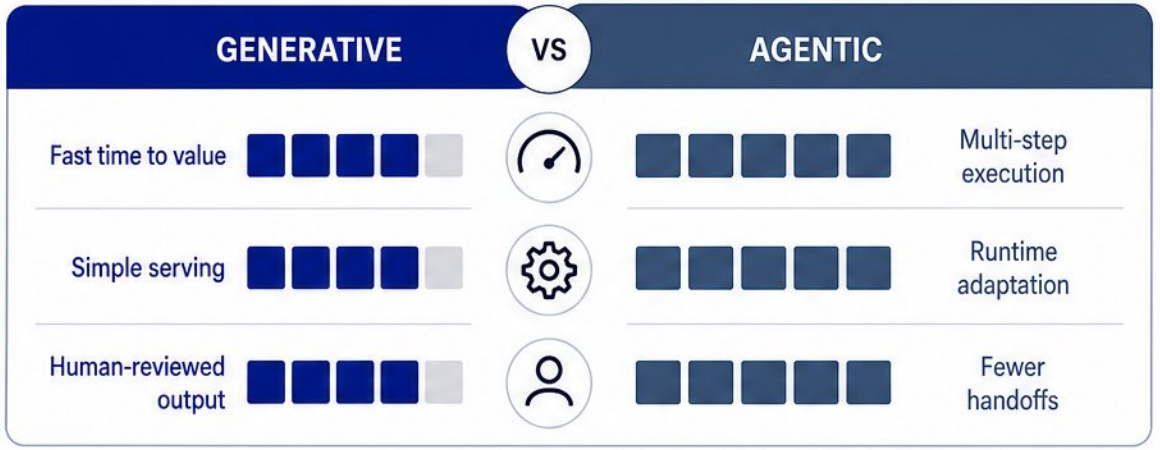
- **Planning turns a model into an agent:** decomposing long-horizon tasks
- **Tool composition** planned against schemas, not guessed from prose
- **Memory is threefold:** short-term context, durable facts, executive memory
- **Context management** is the core bottleneck for long-horizon work
- **Accumulated history** causes lost objectives, repeated actions, distraction
- **Separating tactical execution** from strategic oversight detects loops and drift
- **Better planning** reduces waste, but runtime enforcement stays the correctness boundary

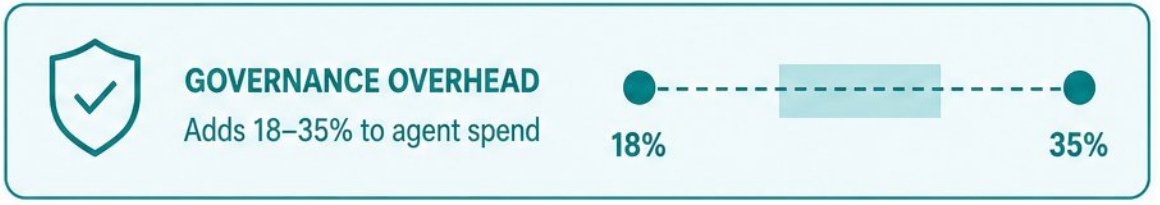




Tradeoffs: When Agency Is Worth the Complexity

- **Generative wins:** fast time to value, simple serving, human-reviewed output
- **Agentic wins:** multi-step execution, runtime adaptation, fewer handoffs
- **Cost:** agent tasks use ~10x more tokens; governance adds 18–35%
- **Agent failure modes:** unbounded loops, tool misuse, goal drift, cascading hallucinations
- **Heuristic:** generation when last step is content; agents when it's a verifiable state change



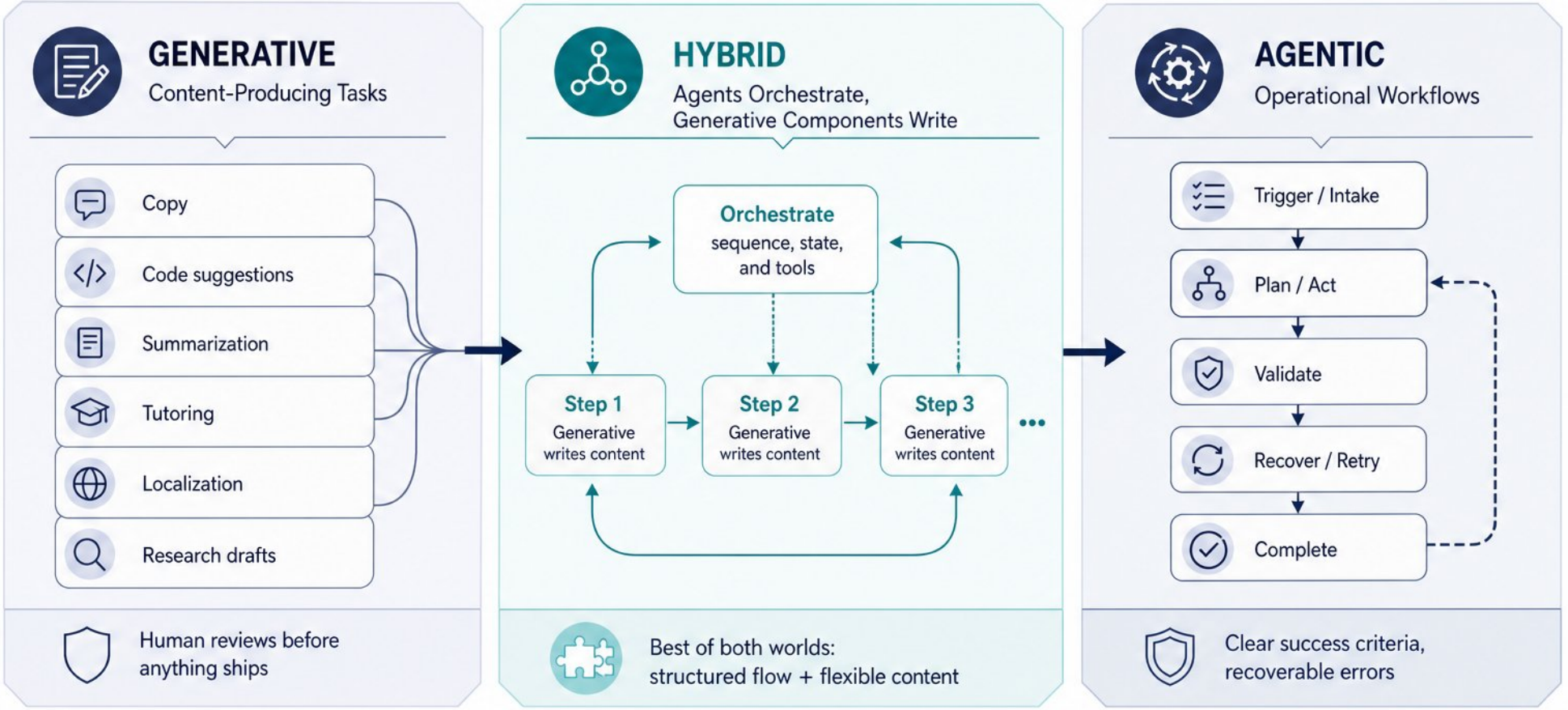


Evaluation and Observability: Measuring Agents Differently

- Generation is scored on output quality; agency requires scoring the whole trace
- Trace-level metrics: task success, planner quality, tool correctness, step efficiency, policy compliance
- Offline benchmarks miss live-tool and path-dependent behavior — blend curated golden sets with real traffic
- Instrument every loop iteration, model and tool call, retrieval, and token spend into replayable traces
- OpenTelemetry-compatible traces, layered evals, and continuous regression monitors on sampled production data
- Error-budget thinking extends to decision quality and cost per successful task

Use Cases: Matching the Approach to the Problem

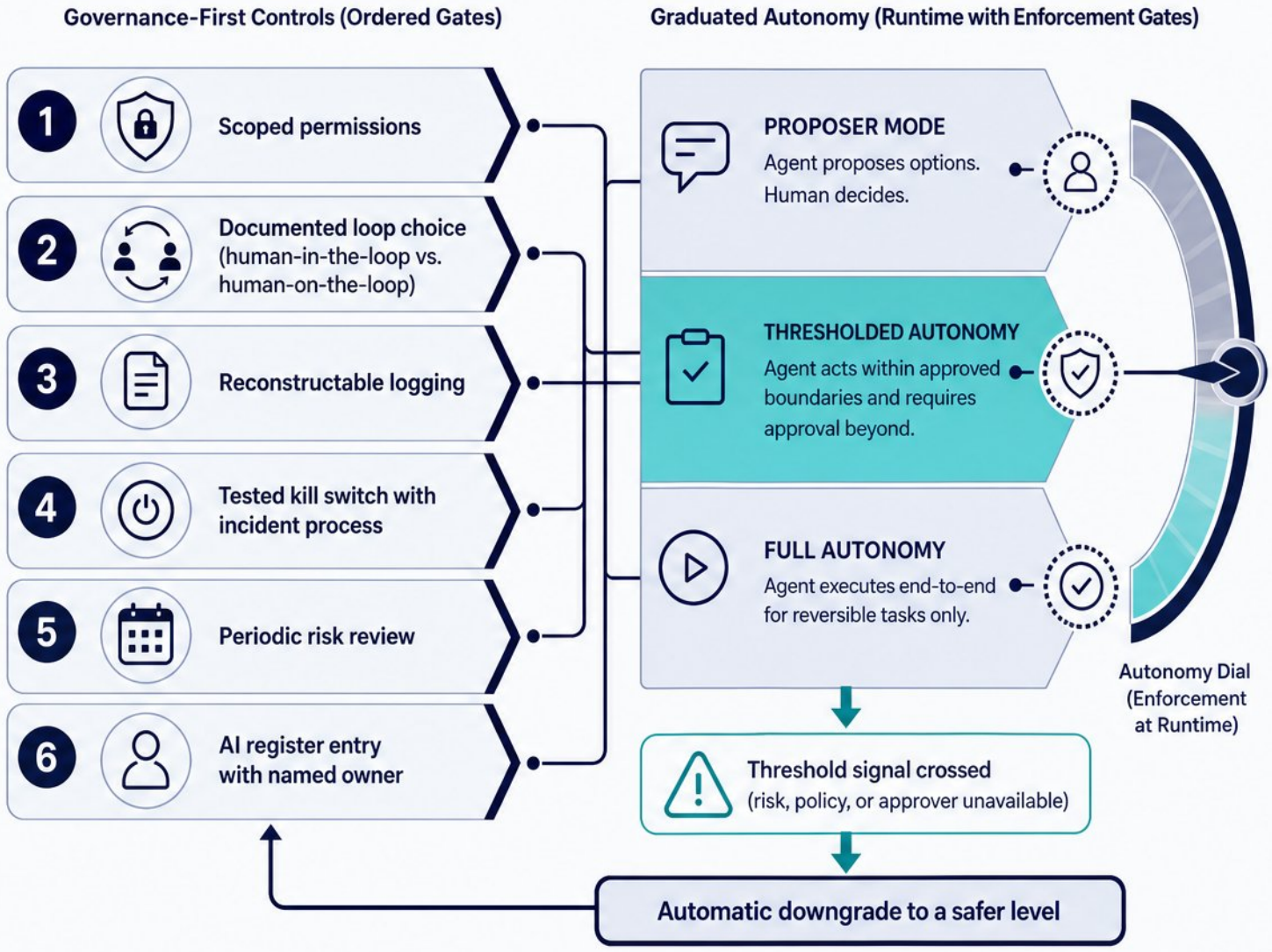
REACTIVE / STATIC ····· GOAL-DIRECTED / LOOPING



- **Generative fit:** copy, code suggestions, summarization, tutoring, localization, research drafts—human reviews before anything ships
- **Agentic fit:** high-volume, repetitive workflows with clear success criteria and recoverable errors
- **Proven agentic wins:** IT operations triage, support resolution, finance exception handling, procurement, engineering pipelines
- **Hybrid is the 2026 norm:** agents orchestrate sequence, state, and tools; generative components produce each step's content
- **Anti-patterns:** agency for low-volume or one-off tasks; persona-based agents that digitize silos rather than redesign workflows

Reliability and Governance: Making Agency Deployable

- Governance is a design input; retrofitting guardrails costs 3–4x more
- Six core controls: scoped permissions, documented loop choice, reconstructable logging
- Kill switch with incident process, periodic review, AI register entry with owner
- Enforce at runtime, not in prompts — default-deny when no approver can respond
- Graduated autonomy: proposer mode, thresholded autonomy, full only for reversible tasks
- Diminishing returns: past ~28% of AI budget, agent ROI goes flat



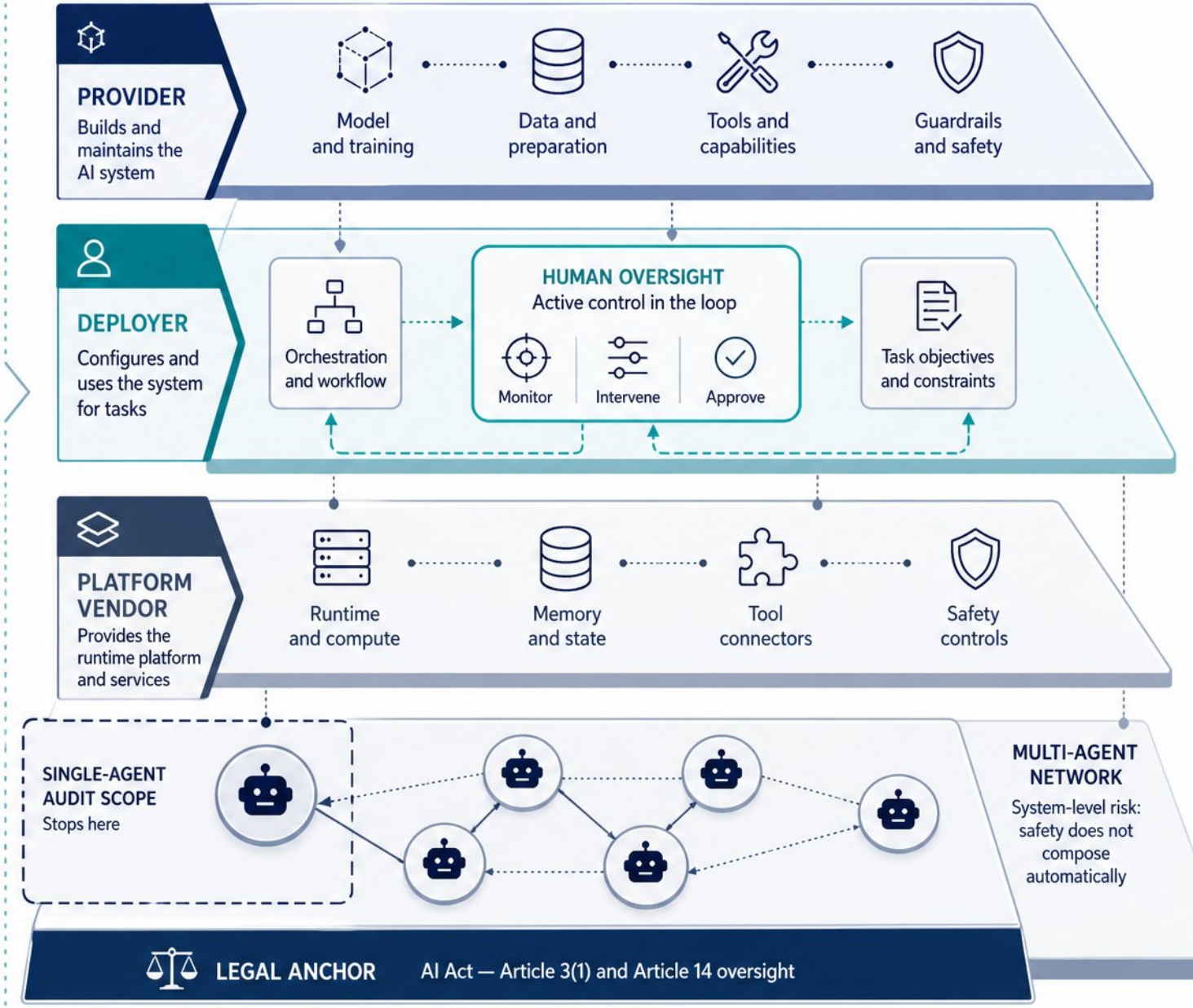
ENFORCE AT RUNTIME, NOT IN PROMPTS

Default-deny when no approver can respond.

Policy and people are always in the loop.

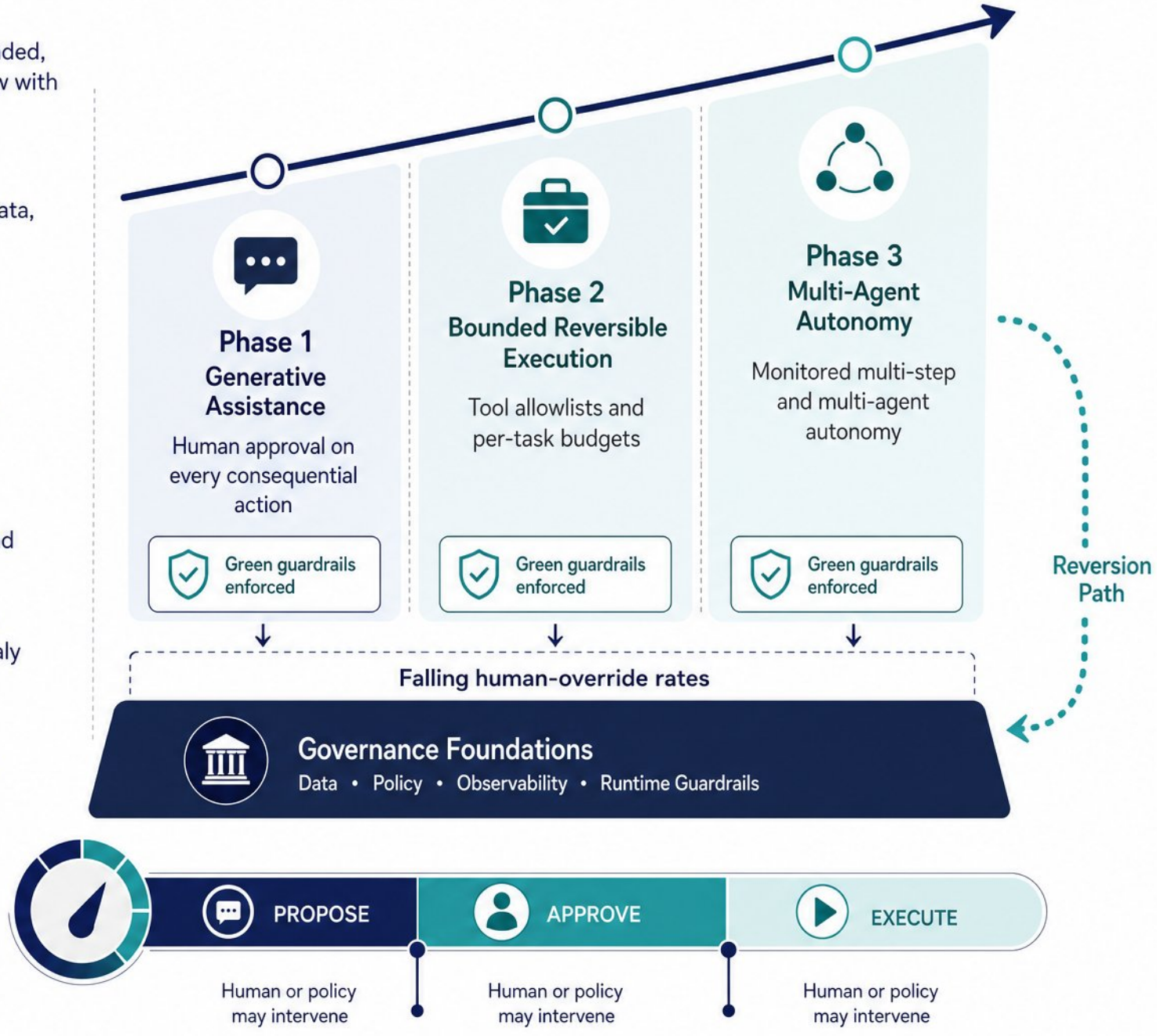
Regulation and Accountability: What 2026 Requires

- No separate agent regime: agents are AI systems under Article 3(1)
- Which duties apply depends on the task and your role, not the label
- High-risk obligations and Article 14 oversight are enforceable from August 2026
- Human oversight must be a measurable system property, not a claim
- Prompting shapes paths; only runtime enforcement constrains them
- Providers, deployers, and vendors hold different duties — settle before incidents
- Multi-agent risk is system-level: safety does not compose automatically



Implementation Playbook: A Phased Path to Bounded Autonomy

- Scope first: one bounded, high-volume workflow with recoverable errors
- Build governance foundations first — data, policy, observability, runtime guardrails
- **Phase 1:** generative assistance, human approval, full tracing, golden eval set
- **Phase 2:** bounded reversible execution with tool allowlists and per-task budgets
- **Phase 3:** multi-agent autonomy with anomaly detection and auto-reversion
- Roughly 70% people and process change; name agent owners and risk partners



Decision Framework: Choosing Between Generation and Agency

- **Step 1:** Classify the goal — content or state change?

↓

- **Step 2:** Test necessity — planning, tools, memory, or branching required?

↓

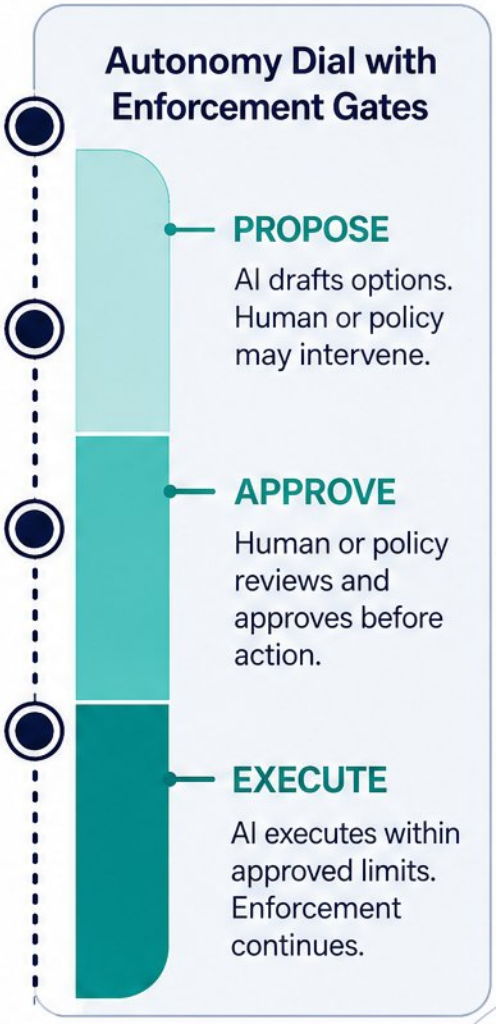
- **Step 3:** Score risk — action scope, reversibility, data sensitivity, blast radius

↓

- **Step 4:** Price oversight — evaluation, monitoring, escalation, human review

↓

- **Step 5:** Set autonomy limits and enforcement points before launch



- **Checklist:** goal type, autonomy need, reversibility, sensitivity, oversight cost, named owner

Key Takeaways and Next Steps by Role

- Generative AI produces content; agentic AI executes goals — scaffolding decides which you built
- Agency adds power and complexity together; choose it only when the workflow justifies both
- Evaluation, observability, and governance are prerequisites — retrofitting costs multiples more
- Start bounded, measure against a baseline, expand autonomy only as override rates fall
- This quarter: product defines metrics; developers instrument traces; practitioners build evals
- Decision-makers: register every agent with a named owner, oversight model, and kill switch

THIS QUARTER: WHO DOES WHAT

PRODUCT LEADERS

Define success metrics and reversibility tiers.

DEVELOPERS

Instrument traces and per-task budgets.

AI PRACTITIONERS

Stand up golden-set and trace-level evaluation.

EDUCATORS

Teach the framework.

DECISION-MAKERS

Run an agent register with named owners and kill switches.

PersonWise.

PersonWise • AI digital-human course platform

Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/74caa8cd/public