AIMONGER
WHITE PAPER

AI Discovery

From Loops to Crews: Choosing an Agent Architecture Without Fashion

Agent architecture discussions often jump to swarms because diagrams travel well. Technology leaders need the opposite discipline: choose the minimum topology that clears a quality, cost, and risk bar for one workflow. With agent features spreading through global enterprise software and cancellation risk rising for weak programmes, architecture choice is a capital decision, not a research preference.

David Saliba . 2026-07-15 . aimonger.com/whitepapers/agent-architecture-loops-stacks-crews/

The problem in one sentence

Architecture diagrams with six agent boxes and zero production metrics are a capital allocation smell. The CIO's job is to pick the dullest topology that fixes one named failure mode, then measure before climbing the ladder.

Agent architecture, plainly: how many model roles, tool loops, and handoffs you wire together to complete a workflow, from one-shot chat to multi-agent crews. **In practice:** invoice exception handling needs a tool loop and a critic, not a five-agent "swarm" nobody can debug at 2 a.m.

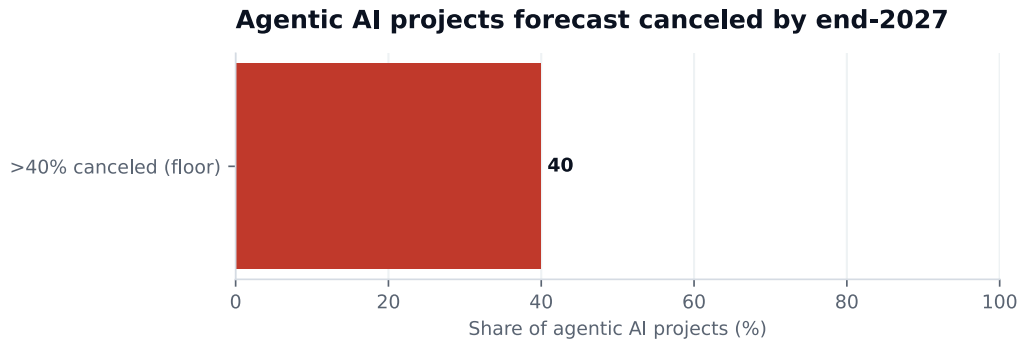
Tool loop, in short: the model repeatedly chooses tools, reads results, and continues until it answers or hits a stop rule (step cap, budget, human gate). **Worked case:** fetch open PO, compare line items, post discrepancy ticket, logged each iteration.

Architecture as fashion risk

Vendors and conference talks reward novelty. Finance rewards unit economics. Architecture must serve the second.

In firms, architecture fashion has a particular cost: smaller technical teams inherit a topology they cannot debug, finance sees variable usage without a unit model, and operators are asked to trust a graph nobody can explain.

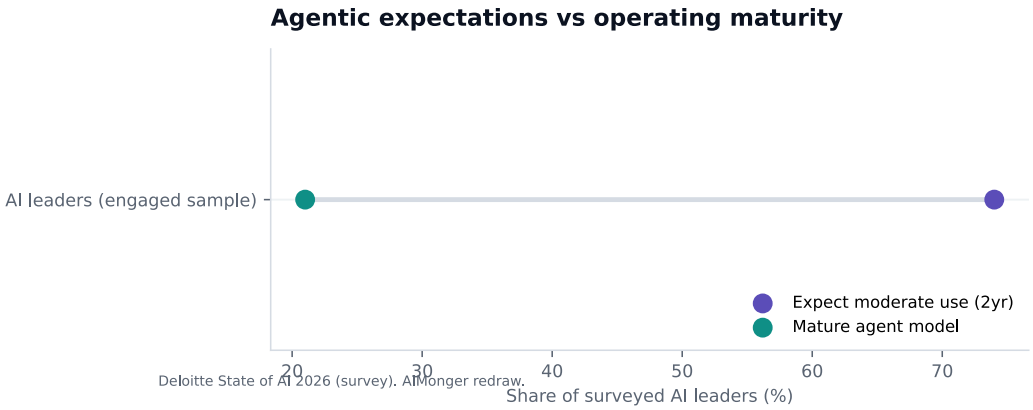
What you see globally right now is "agents everywhere" becoming a product cycle. Gartner predicts task-specific agents will be common in enterprise applications by end-2026, and that more than 40 per cent of agentic projects will be canceled by end-2027 for cost, unclear value, or weak risk controls. Architecture that maximises novelty accelerates cancellation.



Gartner press release, 25 Jun 2025 (forecast: more than 40%). AIMonger redraw.

Figure 1. Share of agentic AI projects Gartner forecasts will be canceled by end-2027. Plain read: expensive, opaque architectures get killed in steering. **On Monday:** the crew pilot that impressed in demo month is cancelled in month nine when cost per case is 4x baseline. Source: Gartner, "Over 40% of Agentic AI Projects Will Be Canceled by End of 2027," press release, 25 June 2025 (https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027). AIMonger redraw.

Deloitte's 2026 survey finds 74 per cent of AI leaders expect at least moderate agentic use within two years, but only 21 per cent report a mature operating model for autonomous agents. Teams climb the topology ladder faster than they build control planes.



Deloitte State of AI 2026 (survey). AIMonger redraw.

21 74

Figure 2. Leaders expecting moderate agentic use within two years vs those with a mature autonomous-agent operating model. In short: crews get funded before loop baselines exist. **Operating case:** token spend rises with agent count; graduation metrics do not. Source: Deloitte State of AI in the Enterprise 2026 (survey). AIMonger redraw.

Eurostat's 20.0 per cent EU enterprise AI adoption figure is a reminder that many local competitors are still forming their first durable patterns. Getting the topology ladder right now is cheaper than unwinding crew sprawl later.

Topology ladder: climb only with evidence

1. **Single completion.** One prompt produces one draft answer, which is fine for brainstorming but insufficient when the system must read live enterprise systems of record. The completion pattern has no tools, no retrieval, and no verification, so it cannot ground an answer in a contract clause or a current ERP figure. Use it for first-draft prose and nothing consequential.
2. **Tool loop.** The model calls tools, reads results, and continues under stop conditions until it can answer or must halt. This is the default production pattern for workflow work because it lets the agent fetch live data, compare it against policy, and produce an answer grounded in systems of record rather than training-time memory. Every rung above this one adds cost and failure modes, so the loop is where most workflows should stop.
3. **Loop plus critic.** A second pass verifies grounding, policy, and format before anything consequential leaves the firm. The critic is not a second opinion for its own sake; it is a calibrated check against frozen test cases that catches the failure modes a single pass misses. This rung fixes high-stakes errors without the coordination overhead of a full crew.
4. **Specialist pair.** Two named roles share one contract (for example research then draft) when a single loop cannot cover two skills cleanly. The pair is justified only when the two roles need genuinely different tools or corpora, and when the handoff between them can be written as a schema with required fields. A pair that passes prose back and forth is just a loop with extra latency.
5. **Crew or swarm.** An orchestrator coordinates several specialists, only after a pair still fails on latency or quality on the holdout set. Crews multiply token spend, debugging opacity, and ownership confusion, so the evidence bar for climbing to this rung is the highest in the ladder. If the holdout does not show a gap that only parallel work closes, the crew is fashion.

Crew means: multiple specialised agents coordinated by an orchestrator, often in parallel, each with its own tools and mandate. **Board reading:** research agent, drafting agent, and compliance checker, only justified when parallel work truly beats a sequential loop plus critic.

Climb only when evaluation proves the lower rung fails on your holdout set, not when a vendor diagram looks modern.

Evidence base: novelty is not a value proxy

Source	Finding	Architecture implication	Does not prove
Gartner (2025)	40% of enterprise apps with task-specific agents by end-2026	Suites ship agent surfaces; your topology still needs a case	That crews are default
Gartner (2025)	>40% agentic projects forecast cancelled by end-2027	Minimum topology reduces exposure	All failures are architecture-only
Deloitte 2026	21% mature agent model; 74% expect broader use	Match topology to operating maturity	Exact EU mid-market split
OWASP Agentic 2026	Cascading failures, insecure inter-agent comms	Crews need blast-radius design	Any framework removes risk
McKinsey 2025	~6% high performers (>5% EBIT from AI)	Redesign workflows; dull paths scale	Model brand predicts success

McKinsey reports high performers push transformative use, redesign workflows, and scale faster, not that they deploy the most agents. Stanford's AI Index 2026 compiles survey data showing organisational AI use rising while agent deployment in production remains single-digit in compiled figures. The gap is architecture discipline and absorption, not access.

Plain read: cancellation and immaturity cluster where cost, value, and controls are weak. Fashionable topologies increase all three.

Selection table: failure mode drives topology

Failure mode	Minimum topology	Why not higher
Weak first draft	Completion or light loop	Critic adds cost before grounding exists
Needs live data	Tool loop	Crew adds coordination without parallel benefit
High-stakes errors	Loop + critic + human gate	Crew multiplies failure modes
Parallel research + draft	Specialist pair with schema handoff	Crew if truly independent workstreams
Complex programme of work	Crew with isolation	Only after pair fails on latency or quality

Control plane, put simply: allowlists, budgets, provenance, evaluation sets, human gates, and operator training; the infrastructure that makes any topology safe. **Operating case:** without a control plane, a tool loop and a crew are equally dangerous; with it, a loop often suffices.

Control plane requirements (any rung)

These controls apply at every rung of the ladder. Topology without them is theatre.

- **Allowlists and budgets.** An allowlist is a fixed inventory of tools an agent may call, paired with a per-case ceiling on token spend or tool-call count. Without both, a loop stuck on a bad prompt can burn a month of FinOps allocation before anyone notices. A mid-market insurer we worked with in principle caps each claims-triage case at 2,000 tokens and four tool calls, with a hard stop that pages the on-call operator.
- **Provenance and versioning.** Record which prompt, corpus version, and model release produced each consequential answer, so operators can reconstruct a bad decision after the fact. Provenance is what turns a vague "the AI got it wrong" into a fixable "the 2024-03 handbook section was stale when the model read it." Without it, incident review becomes guesswork and the same failure recurs.
- **Evaluation sets with holdouts.** Promote only when a fixed test pack still passes on accuracy and cost, and never tune against the promotion set. Holdouts are the only honest way to know whether a topology change helped or hurt, because tuning against the set you measure on is self-deception. A frozen pack that the team cannot see the answers to is the minimum bar.
- **Human gates.** Require a named reviewer before consequential writes or external sends, especially payments, client filings, and contract changes. The gate is not a rubber stamp; it is the point where a human owns the outcome the system produced. A gate with no trained owner is a poster on the wall, and a workflow with no gate on payments is an incident waiting for its date.
- **Operator training.** Train the people who will override, refuse, and escalate, because a kill switch with no trained owner is decoration. Operators need to know what the system does, what it must not do, and how to shut it down under pressure at 2 a.m. A team that cannot demonstrate the kill switch in a drill cannot use it in an incident.

Without the control plane, topology is decoration. Gartner's cancellation drivers explicitly include inadequate risk controls.

Evolution story (shared vocabulary)

Teams talk past each other when words drift. Use this shared history so architecture debates stay concrete:

1. **Chat completion.** One prompt produces one answer, useful for drafting and weak when the system must act on live enterprise data. The pattern has no tools and no retrieval, so it cannot verify a claim against a system of record. It is the baseline that every more complex topology exists to fix.
2. **RAG stack.** Retrieve relevant passages from a corpus, then generate an answer grounded in those passages. RAG fixes open-web guessing for policy and contract questions by forcing the model to cite a governed source. Without it, the model invents clauses that sound plausible and fail in court.
3. **Tool loop.** The model iterates by calling a tool, reading the result, and deciding the next step. This rung is required when work must touch ERP, CRM, or ticketing systems, because the model cannot fetch a live purchase order from training-time memory. Every production workflow that reads or writes enterprise data starts here.
4. **Verified loop.** Add a critic, checks, and gates after generation so high-stakes errors get caught before they leave the firm. The verified loop is the loop plus a second pass that scores the output against frozen tests, which is where most quality gain actually comes from. It captures most of the value of a crew at a fraction of the coordination cost.
5. **Multi-agent.** Specialised roles coordinate on parallel workstreams, only after a single verified loop has already moved a metric. The multi-agent rung exists to fix latency or skill gaps that a single loop cannot, not to impress an architecture review. Climb here only when the holdout proves the verified loop fails.

Each step exists to fix a failure of the previous step, not to impress architecture reviews.

RAG (plain English): retrieve relevant documents from a corpus, then generate an answer grounded in those passages. **Architecture case:** policy Q&A retrieves the 2024 handbook section before drafting, not open-web guessing.

Latency and reliability budgets

Architecture choice must state these numbers before approval. A crew that cannot meet them is a lab toy regardless of cleverness.

- **P95 latency.** State the slowest acceptable response for the workflow SLA, for example under eight seconds for an interactive case or under two minutes for a batch overnight job. The P95 is the number the user actually experiences, and a topology that misses it cannot hide behind an average. State it before approval, not after the first steering complaint.
- **Idempotency for writes.** Define what happens if a write tool runs twice, so duplicate payments or duplicate tickets cannot appear after a retry. Idempotency keys are how a loop that retries a failed post does not charge a customer twice for the same invoice. Without this rule, a transient network error becomes a financial event.
- **Retry and circuit breakers.** Cap automatic retries and fail closed when a tool is unhealthy, instead of looping until the budget dies. A circuit breaker is the control that turns a degraded tool into a slow workflow rather than a budget fire. The alternative is a loop that keeps calling a 500-response endpoint until the token allocation is gone.
- **Partial failure behaviour.** Specify what the user sees when one step fails, which should be a clear handoff to a human, not a silent half-written record. Partial failure is where agents create the worst messes, because the system looks done but is missing a step. A defined handoff means the operator knows the work is incomplete; a silent failure means nobody does.
- **Kill switch ownership.** Name who can stop the system in production, where the switch lives, and how operators rehearse it. The kill switch is the last line of defence when a loop runs wild, and it is useless if nobody knows who holds it at 2 a.m. Rehearse it before the incident, not during it.

FinOps data shows token multiplication hits hardest on chatty multi-agent handoffs, so latency and cost budgets must be written together.

Cost topology reminders

Topology	Cost shape	Failure shape
Completion	Lowest	Weak grounding
Tool loop	Linear with steps	Tool storms without budgets
Loop + critic	~2x generate	Critic rubber-stamping if uncalibrated
Specialist pair	Superlinear if chatty handoffs	Contract gaps
Crew	High variance	Debugging opacity

Require engineering to place the proposal on this table with estimated P50/P95 cost before architecture approval.

Reference pattern: loop + critic

Often the best "second agent" is not a crew. It is a critic:

- **Agent A proposes action or draft.** The primary loop retrieves, calls tools, and produces a candidate output grounded in policy and systems of record. This is the loop that does the actual work of fetching live data and producing a first answer the firm can act on. Without it, the critic has nothing to check.
- **Critic checks policy rubrics and eval cases.** A second pass scores grounding, format, and refusal behaviour against frozen tests before anything consequential leaves the firm. The critic is calibrated against holdout cases the team cannot see, so it cannot rubber-stamp by coincidence. This is where most quality gain lives, at a fraction of crew cost.
- **Human gate on remaining risk.** A named reviewer approves, edits, or blocks outputs that still carry financial, legal, or customer impact after automated critic checks pass. The gate is the point where a human owns the outcome, not a formality. Outputs that clear the critic but touch payments or client commitments stop here.

This captures much of the quality gain without orchestration sprawl. Published multi-agent comparisons (including Google-internal benchmarks cited in industry literature) show centralised orchestration can reduce error amplification versus fully independent agents. Still treat these as lab multipliers, not your incident rate.

On the P&L: a claims summary loop plus critic that flags missing policy clauses beats a three-agent crew that duplicates retrieval.

Reference pattern: specialist pair

Use when research and drafting truly contend for different tools and corpora. Write the handoff as a schema (JSON fields, required citations), not a paragraph of hope.

Fail the fashion test if:

1. **More agent boxes than production metrics.** The architecture diagram outnumbers the scoreboard metrics steering actually reviews each month, signalling topology ahead of absorption. A crew with six agents and zero cycle-time metrics is a diagram in search of a workflow. Fix the metrics before the diagram.
2. **Vendor demo cannot show audit export.** Trace replay lives only inside the vendor console, which fails EU deployer and internal incident-review requirements for consequential workflows. A vendor that cannot export JSON traces cannot prove what its agents did during an incident. This is a procurement blocker, not a feature gap.

3. **No single-loop baseline attempted.** The team skipped a tool loop with logging and gates, jumping to crews before documenting what simpler topology failed on holdout. Without a baseline, the crew is a guess, not a graduation. The holdout result for the loop is the only honest justification for the crew.
4. **Write-tool autonomy on day one.** Agents may mutate systems of record before provenance, evaluation, and staffed human gates exist, multiplying OWASP-style blast radius immediately. Write autonomy is earned after the control plane, not before it. A crew with write tools and no gates is an incident with a countdown.
5. **Evaluation plan is watch in production.** "We will see how it goes live" replaces frozen holdouts, guaranteeing steering debates anecdotes instead of promotion criteria after the first incident. Watch-in-production is how a topology avoids ever being judged. Frozen holdouts are how a team learns whether the architecture helped.

Worked example: Munich industrial distributor

Composite: 600 staff, SAP plus SharePoint, customer quote workflow.

Failure mode: quotes slow because engineers manually search prior bids and spec sheets.

Wrong architecture: five-agent crew (researcher, pricer, drafter, reviewer, sender) before a baseline exists.

Right progression: 1. **RAG-assisted search over governed bid archive.** Read-only retrieval lets engineers find similar bids and spec sheets without agents writing to CRM or ERP on day one. This is the cheapest rung that fixes the named failure mode, and it builds the corpus that later rungs depend on. No write autonomy, no orchestration cost, no incident surface. 2. **Tool loop for line items and quote skeleton.** The loop extracts comparable fields and drafts a quote structure under allowlists, logging each tool call for audit and FinOps attribution. The loop is where the workflow starts touching live systems, so the control plane has to exist before this rung. Each call is logged, budgeted, and gated by the allowlist. 3. **Loop plus critic on pricing bands and discount policy.** A critic pass flags out-of-band discounts or missing mandatory clauses before humans review, capturing crew-like quality without orchestration sprawl. The critic is calibrated against holdout quotes with known correct pricing, so it catches the failure modes the loop misses. This rung is where most quality gain appears, and it costs a fraction of a crew. 4. **Human gate before CRM send.** No quote reaches the customer or CRM record without a named approver, keeping consequential sends off autopilot until metrics and evals justify narrower autonomy. The gate is the point where a human owns the outcome, and it is non-negotiable for customer-facing sends. Narrower autonomy is earned by the holdout, not granted by the calendar.

Metrics: quote cycle time, rework rate, cost per successful quote. Crew is considered only if research and pricing must run in parallel under a 90-second SLA, and evaluation must prove the pair fails first.

Research addendum: topology is task-structure

Use multi-agent topologies when work is parallel and decomposable with clear contracts. Prefer single loops or loop plus critic when work is sequential or dependency-heavy. Trace diagnostics and workflow pass-rate tests (τ-bench-style evaluations in academic literature) belong in architecture review, not only security review.

Prune topology until the failure mode is fixed. If the failure mode is unspecified, the topology is fashion.

Counter-position: modern platforms make crews cheap

Another board might argue low-code agent platforms reduce wiring cost, making crews the default. Platforms reduce assembly cost. They do not remove token multiplication, debugging opacity, or ownership confusion. Cheap to assemble is not cheap to operate.

Require P50/P95 cost envelopes and a kill metric before crew approval. Deloitte's finding that only 25 per cent of surveyed organisations moved at least 40 per cent of AI experiments to production, while 30 per cent redesigned key processes, suggests most firms still struggle to graduate any topology, let alone the heaviest one.

CIO architecture decision record

Copy into every production proposal. Leave the Entry column blank until the proposal is complete.

Field	Entry
Workflow	
Failure mode addressed	
Topology chosen	
Topologies rejected and why	
Control plane checklist score (0 to 5)	
Latency P95 budget	
Cost P50 and P95 per successful case	
Kill metric and owner	
Review date	

Plain read: empty cells are intentional. A proposal without a kill metric is not ready for steering.

Reference progression that survives review

1. **Assisted draft with human send.** Model drafts from retrieved context and every external send passes a human, establishing grounding and policy habits before any tool writes. This rung is where the team learns whether the retrieval and prompting are good enough to act on, before any automation risk appears. No workflow should skip it.
2. **Single tool loop, read-heavy.** The agent reads ERP, DMS, or ticketing under allowlists and logs each step, proving control plane discipline before write autonomy is considered. Read-only loops build operator trust in the logging and budgeting without risking a bad write. This is where the control plane is tested against real systems.
3. **Loop plus critic on high-stakes outputs.** A second automated pass checks policy rubrics and holdouts on outputs that could affect pricing, compliance, or customer commitments. The critic is calibrated against frozen cases, so it catches the failure modes a single loop misses. Most workflows that touch consequential decisions should stop here.
4. **Specialist pair with schema handoff.** Two roles exchange JSON with required citations when parallel skills truly beat a sequential loop, still short of a full crew orchestrator. The pair is justified only when the handoff is a schema, not prose, and when the two roles need genuinely different tools. A pair that passes prose is just a loop with latency.
5. **Crew only after prior steps are boring.** Multi-agent coordination funds only when steps one through four run reliably for weeks and evaluation still shows a holdout gap crews uniquely close. The crew is the most expensive rung, and the evidence bar is the highest in the ladder. If the gap can be closed by a pair, the crew is fashion.

Skipping levels correlates with the cancellation drivers Gartner forecast: cost, unclear value, weak controls.

Board decision criteria

Question	Pass	Fail
Named failure mode in writing	Yes	"We need agents"
Lower topology baseline tested	Documented fail	Skipped
Control plane complete	>=4/5 items	Missing budgets or gates
Cost envelope approved by finance	P95 within SLA	Unknown
Operators trained without project team	Certified	Demo-only
Audit export demonstrated	Yes	Vendor UI only

Fail two or more and defer crew designs; approve loop or loop plus critic only.

Closing position

Choose the dullest architecture that meets the metric.

Dull architectures survive Gartner's cancellation window. Fashionable ones often do not.

When every suite ships an agent surface, your differentiation is not the diagram. It is whether the workflow clears quality, cost, and risk bars with a topology you can operate on Monday morning.

References

1. Gartner, "Gartner Predicts 40% of Enterprise Apps Will Feature Task-Specific AI Agents by 2026, Up from Less Than 5% in 2025," press release, 26 August 2025. https://www.gartner.com/en/newsroom/press-releases/2025-08-26-gartner-predicts-40-percent-of-enterprise-apps-will-feature-task-specific-ai-agents-by-2026-up-from-less-than-5-percent-in-2025
2. Gartner, "Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027," press release, 25 June 2025. https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027
3. Deloitte AI Institute, "The State of AI in the Enterprise: The Untapped Edge" (2026 edition); survey of 3,235 leaders, Aug-Sep 2025. https://www.deloitte.com/us/en/what-we-do/capabilities/applied-artificial-intelligence/content/state-of-ai-in-the-enterprise.html
4. OWASP GenAI Security Project, "OWASP Top 10 for Agentic Applications for 2026," 9 December 2025. https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/
5. McKinsey & Company / QuantumBlack, "The State of AI: Global Survey 2025." https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai
6. Stanford Institute for Human-Centered Artificial Intelligence, "The 2026 AI Index Report," Economy chapter (organisational adoption compiled from surveys including McKinsey). https://hai.stanford.edu/ai-index/2026-ai-index-report/economy
7. Eurostat, "20% of EU enterprises use AI technologies," 11 December 2025. https://ec.europa.eu/eurostat/en/web/products-eurostat-news/w/ddn-20251211-2
8. NIST, "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile," NIST AI 600-1, July 2024. https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.600-1.pdf

9. Reuters, “Over 40% of agentic AI projects will be scrapped by 2027, Gartner says,” 25 June 2025. https://www.reuters.com/business/over-40-agentic-ai-projects-will-be-scrapped-by-2027-gartner-says-2025-06-25/

Published by AIMonger . https://aimonger.com . Malta . Europe