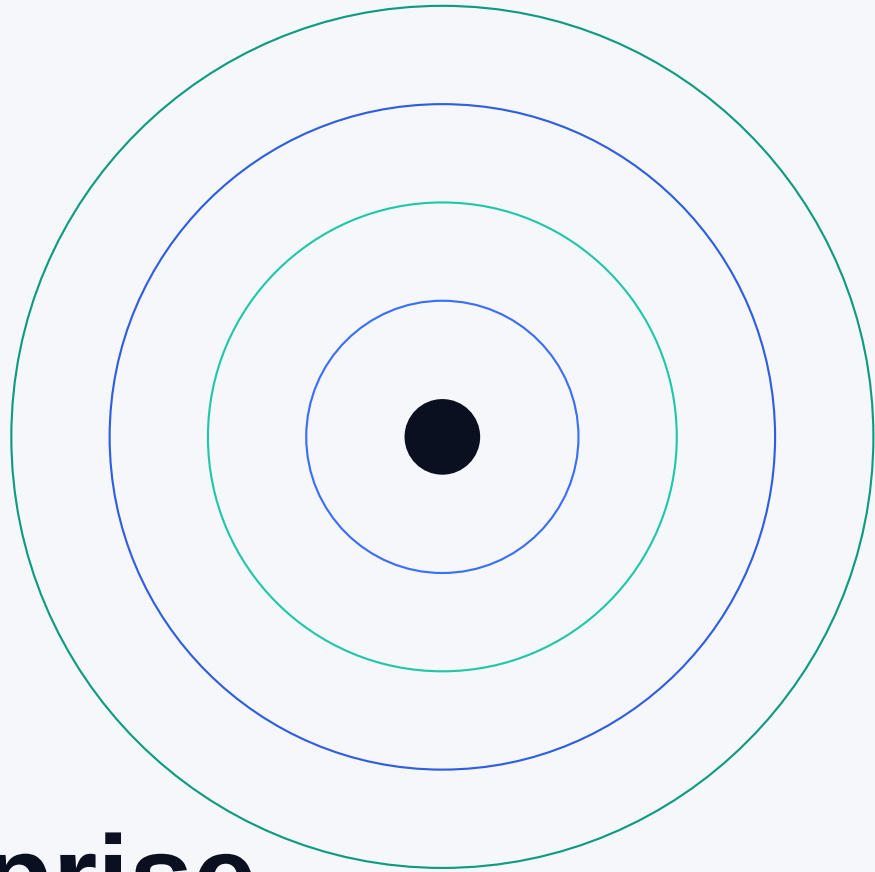




PLANEON FIELD NOTE / 01

The enterprise multi-agent systems blueprint.

Sixteen boundaries that turn a capable model into a system an enterprise can operate, govern, and trust.

Vendor-neutral architecture / 16 harnesses / 43 exchanges / 4 build phases

The model is necessary. It is not the system.

A production agent is a chain of identity, interpretation, retrieval, reasoning, authority, action, memory, evidence, and feedback. Most failures arrive in the joins: a missing workload identity, a retry without idempotency, retrieved text treated as instruction, or a release gate that consumes opinion instead of evidence.

The blueprint makes those joins explicit. It does not prescribe one platform or vendor. It names the jobs that fail differently, change at different speeds, or answer to different owners.

Only two of forty-three exchanges in the reference task touch the model core. Everything else establishes identity, context, authority, safety, continuity, cost, and accountability.

Group by concern, not hierarchy.

Runtime plane

Where it runs, how models are served, and how requests enter and leave.

Knowledge plane

What the system knows, may retrieve, and earns the right to remember.

Execution plane

How work is planned, continued, standardised, isolated, and completed.

Trust & lifecycle plane

Who or what is permitted, how evidence is gathered, and how change is governed.

The rings in the composition diagram are groupings of concern. They do not imply containment, priority, or dependency.

Every boundary needs an owner and proof.

Each profile below comes directly from the website data source. Tool and standards lists are intentionally omitted from this concise edition because they are the most perishable part of the landscape.

01

Infrastructure & Runtime

PHASE 0

The compute, scheduling, networking and resilience substrate every other harness sits on - Kubernetes across public and private cloud, GPU-aware and multi-tenant from the start.

ACCOUNTABLE OWNER

Platform Engineering / Cloud Infrastructure

WHAT DONE LOOKS LIKE

Done when a team can deploy an agent workload to a private-cloud cluster with GPU quota, secrets, a network policy and a tested restore - without filing a ticket.

SIGNALS

- + GPU utilisation and queue wait time
- + Scale-from-zero cold start P95
- + Cluster availability against SLO

02

Model & Inference

PHASE 1

Serve, version and scale the models - proprietary and open-weight - with predictable latency and cost, and keep them swappable so no single provider becomes load-bearing.

ACCOUNTABLE OWNER

ML Platform / AI Engineering

WHAT DONE LOOKS LIKE

Pilot needs one hosted model and one self-hosted fallback behind the same interface, both version-pinned, with a documented switch procedure.

SIGNALS

- + Time to first token and tokens per second
- + P99 end-to-end latency by model tier
- + Prefix cache hit rate

03

AI Gateway

PHASE 1

One control point for every model call: authentication, tenant quota, routing, caching, cost metering and the edge layer of guardrail enforcement.

ACCOUNTABLE OWNER

Platform Engineering, with Identity & Access Management

WHAT DONE LOOKS LIKE

Pilot needs authentication, per-tenant quota and token metering. Semantic routing and caching are optimisations for phase 2.

SIGNALS

- + Requests per second and gateway-added P99 overhead
- + Quota and budget rejections by tenant
- + Cache hit rate and cache-attributable savings

04

Experience & Interaction

PHASE 1

The surfaces where people meet the system - and where they steer it: channels, streaming, sessions, clarification, provenance display and the approval moments that make autonomy safe.

ACCOUNTABLE OWNER

Product Engineering / Digital Channels, with UX

WHAT DONE LOOKS LIKE

Pilot can ship one channel with streaming. Approval and clarification surfaces become mandatory the moment agents can write to production systems.

SIGNALS

- + Task completion and abandonment rate
- + Clarification rate and its effect on success
- + Approval latency and reviewer override rate

05

Domain & Semantic

PHASE 2

Shared, machine-usable meaning: ontologies, a semantic layer, agreed metric definitions and regulation expressed as knowledge the agent can reason over.

ACCOUNTABLE OWNER

Data Governance / Enterprise Architecture, with domain SMEs

WHAT DONE LOOKS LIKE

Start with the twenty terms that appear in disputed answers, not a full enterprise ontology. Expand only where ambiguity has actually cost something.

SIGNALS

- + Coverage of disputed terms with an approved definition
- + Definition change rate and steward review time
- + Metric disagreement incidents per quarter

06

Data Integration & Provenance

PHASE 2

Governed access to source systems with lineage and entitlements attached - the data plane behind retrieval, deliberately separate from the tool plane that takes actions.

ACCOUNTABLE OWNER

Data Engineering, with Data Governance and Privacy

WHAT DONE LOOKS LIKE

Connect three sources properly - contract, lineage, entitlements - before connecting thirty carelessly. MCP data servers shorten integration work substantially, but the governance is the slow part, not the plumbing.

SIGNALS

- + Freshness against contract per source
- + Contract test pass rate
- + Entitlement violations detected (target zero)

07

Retrieval & Context Engineering

PHASE 1

Assemble the smallest sufficient context for each step, inside a token budget: hybrid retrieval, reranking, graph traversal, iterative refinement, compaction and tool search.

ACCOUNTABLE OWNER

AI Engineering, with Data Engineering

WHAT DONE LOOKS LIKE

Pilot ships hybrid retrieval with reranking and a hard token budget. Add graph retrieval and iterative refinement once you can measure groundedness.

SIGNALS

- + Recall at k and reranker gain
- + Groundedness and citation accuracy
- + Context tokens per successful task

08

Memory & State

PHASE 2

Decide deliberately what persists beyond a task - working, episodic, semantic and procedural memory - with an explicit write policy, retention limits and poisoning controls.

ACCOUNTABLE OWNER

AI Engineering, with Privacy

WHAT DONE LOOKS LIKE

Pilot may use session-scoped memory only. Long-lived cross-session memory should not ship before poisoning controls and erasure both work.

SIGNALS

- + Recall usefulness - memory hits that improved the outcome
- + Write rate and store growth per tenant
- + TTL adherence and orphaned record count

09

Protocol & Interoperability

PHASE 1

Standard contracts at every boundary so frameworks stay replaceable: MCP downward to tools and data, A2A sideways to peer agents, OpenAPI and AsyncAPI to the rest of the estate.

ACCOUNTABLE OWNER

Enterprise Architecture, with Platform Engineering

WHAT DONE LOOKS LIKE

Adopt MCP in the pilot to avoid rewriting integrations later. A2A federation and a shared agent registry are the phase 3 workstream, once more than one team runs agents that must delegate to each other.

SIGNALS

- + Share of integrations on standard contracts rather than bespoke glue
- + Schema-break incidents per quarter
- + Registry coverage against deployed reality

10

Orchestration & Durable Execution

PHASE 1

Plan and decompose the task, route work to the right agent or tool, checkpoint durably so long tasks survive restarts, and stop cleanly when budget or policy says so.

ACCOUNTABLE OWNER

AI Engineering

WHAT DONE LOOKS LIKE

Pilot needs a single orchestrator with durable checkpoints and a hard step and budget ceiling. Multi-agent topologies only after a single agent has been measured and found insufficient.

SIGNALS

- + Task success rate and steps per successful task
- + Token cost per successful task, not per call
- + Circuit-breaker and termination trips

11

Tool, Skill & Sandbox

PHASE 2

The action surface: typed tool contracts, packaged skills, an explicit read-versus-write authority model, and isolation strong enough for code the model wrote.

ACCOUNTABLE OWNER

AI Engineering, with Security Engineering co-signing write authority

WHAT DONE LOOKS LIKE

Ship read-only tools in the pilot. Every write tool should arrive with an approval path, an idempotency key and a rollback story - no exceptions in phase 2.

SIGNALS

- + Tool error and timeout rate by tool
- + Share of write tools behind an approval gate (target 100%)
- + Blocked unsigned or unscanned tool loads

12

ML & Decision Intelligence

PHASE 3

Keep discriminative and optimisation models in the architecture as first-class tools and routing signals - often cheaper, faster and more accurate than asking a language model.

ACCOUNTABLE OWNER

Data Science / Machine Learning

WHAT DONE LOOKS LIKE

A scale-phase optimisation: introduce when volume makes a measured LLM path too slow, too costly or too imprecise for a repeated decision. Do not create a parallel ML organisation - reuse the same registry, evaluation and observability harnesses.

SIGNALS

- + Accuracy or AUC by model against its baseline
- + Feature and prediction drift
- + Share of decisions served by classical models versus the LLM path

13

Security, Safety & Guardrails

PHASE 1

Runtime enforcement on every call, plus the agentic attack surface: non-human identity, injection and PII defence, output validation, supply chain, circuit breakers and kill switches.

ACCOUNTABLE OWNER

Security Engineering / CISO organisation

WHAT DONE LOOKS LIKE

Pilot needs injection screening, PII redaction, output validation and workload identity. Full supply-chain control and inter-agent security are mandatory before production data or write access.

SIGNALS

- + Attack success rate, always published with the harness and judge configuration
- + Injection attempts detected and blocked
- + Coverage of just-in-time credentials versus static keys

14

Governance, Oversight & AgentOps

PHASE 2

Decide how much independence each action may have, prove it with policy-as-code and human checkpoints, and run the change control that lets the system improve safely.

ACCOUNTABLE OWNER

AI Governance Office / Risk & Compliance, with Legal

WHAT DONE LOOKS LIKE

Keep it lightweight in the pilot: an agent registry, an owner per agent and a written autonomy tier. Promote to full control the moment agents touch production systems or personal data - before any traffic, not after.

SIGNALS

- + Share of actions executed inside the policy envelope
- + Approval latency and override rate
- + Registry completeness against deployed reality

15

Observability & FinOps

PHASE 1

Make any task reconstructable end to end and attribute its cost: OpenTelemetry GenAI spans carrying trace, thread and task IDs, the acting identity, the policy decision and the token spend.

ACCOUNTABLE OWNER

SRE / Observability platform, with FinOps

WHAT DONE LOOKS LIKE

Instrument from the first pilot request. Retrofitting correlation IDs after the fact is the most common and most expensive omission in this whole composition.

SIGNALS

- + Trace completeness - share of tasks fully reconstructable
- + Share of incidents replayable from telemetry alone
- + Cost per successful task and budget burn rate

16

Evaluation & Assurance

PHASE 1

Grade the system at five scopes, prove reliability across repeated runs, red-team it adversarially, and gate every promotion on the evidence.

ACCOUNTABLE OWNER

Quality / AI Assurance, with the Security red team

WHAT DONE LOOKS LIKE

Pilot needs a golden set and a regression suite fed by real incidents. Adversarial campaigns arrive with production; simulated users and environments are the phase 3 workstream for rehearsing at scale.

SIGNALS

- + Pass rate by evaluation scope
- + Variance across N runs, not just the median
- + Cost-adjusted completion rate

04 / ONE TASK END TO END

Forty-three exchanges, four operating phases.

PH1 / Ingress / STEPS 1-5

Everything between the request arriving and the agent being allowed to plan. The job is to establish who is asking, on whose behalf and inside what budget - and to reject anything hostile before a single token is spent.

01	U	->	EXPR	task arrives on a channel
02	EXPR	->	GW	session and tenant context
03	GW	->	GUARD	screen before spending a token
04	GUARD	->	GW	cleared / scoped agent identity
05	GW	->	ORCH	plan, inside this budget

PH2 / Agent loop / STEPS 6-32

Plan, ground, gather, reason, gate, act - then observe the result and decide whether to go round again. The only phase that repeats, and where cost, latency and risk all accumulate.

06	ORCH	->	DOM	what do these terms mean?
07	DOM	->	ORCH	resolved entities and metrics
08	ORCH	->	RET	assemble context for this step
09	RET	->	DATA	governed rows and columns
10	DATA	->	RET	data with lineage and entitlements
11	RET	->	MEM	what do we already know?
12	MEM	->	RET	episodic, semantic, procedural
13	RET	->	ORCH	context inside the token budget
14	ORCH	->	GW	reason over this context
15	GW	->	INFER	routed, cached, metered
16	INFER	->	CORE	inference on a pinned model
17	CORE	->	INFER	completion
18	INFER	->	ORCH	next action, schema-valid
19	ORCH	->	GOV	score against autonomy policy
20	GOV	->	EXPR	approval at a durable checkpoint
21	EXPR	->	GOV	human decision
22	GOV	->	ORCH	proceed unattended, logged
23	ORCH	->	TOOL	execute the typed tool call
24	TOOL	->	INFRA	provision an isolated sandbox
25	INFRA	->	TOOL	microVM, gVisor or Wasm ready
26	TOOL	->	MLH	classify, rerank or forecast

27	MLH	->	TOOL	score or plan, far cheaper
28	TOOL	->	PROTO	cross on a standard contract
29	PROTO	->	EXT	MCP to systems / A2A to peers
30	EXT	->	PROTO	result or task handle
31	PROTO	->	TOOL	typed result or typed error
32	TOOL	->	ORCH	observe, then re-plan

PH3 / Commit and egress / STEPS 33-37

Persist what is worth keeping, validate what is about to be said, meter what it cost, and deliver it with enough provenance for the reader to judge it.

33	ORCH	->	MEM	write back what earns persistence
34	ORCH	->	GUARD	validate the final output
35	GUARD	->	GW	schema-valid and policy-clean
36	GW	->	EXPR	metered and charged to the tenant
37	EXPR	->	U	answer, with provenance

PH4 / Assurance and lifecycle / STEPS 38-43

Not a phase in time. These exchanges run underneath every step above: telemetry becomes evaluation cases, evaluation becomes release evidence, and approved changes are redeployed into the harnesses that produced the behaviour.

38	ORCH	->	OBS	a span for every hop
39	OBS	->	EVAL	traces become eval cases
40	EVAL	->	GOV	evidence for the release gate
41	GOV	->	ORCH	redeploy versioned artefacts
42	GOV	->	GUARD	policy compiled into enforcement
43	INFRA	->	OBS	capacity, cost and availability

Start with what must be true before autonomy.

00**Phase 0 / Foundation**

Before any agent runs.
Infrastructure & Runtime

01**Phase 1 / Pilot**

Single team, contained blast radius.
Model & Inference / AI Gateway / Experience & Interaction / Retrieval & Context
Engineering / Protocol & Interoperability / Orchestration & Durable Execution / Security,
Safety & Guardrails / Observability & FinOps / Evaluation & Assurance

02**Phase 2 / Production**

Real systems, real data, real users.
Domain & Semantic / Data Integration & Provenance / Memory & State / Tool, Skill &
Sandbox / Governance, Oversight & AgentOps

03**Phase 3 / Scale & federation**

Many agents, many teams, regulated load.
ML & Decision Intelligence

Each phase is complete when its controls can produce evidence under failure - not when the happy-path demo runs once.

Use the framework critically.

Taxonomy, standards facts and technology names derive from three architecture reviews synthesised in the harness composition deck, current to approximately mid-2026. Enterprise ownership and build phases are architectural recommendations, not findings. Re-verify all version-specific and statistical claims before publication.

Enterprise ownership models and build phases are architectural recommendations, not research findings. Tier badges reproduce the source deck's MVP-versus-full split; Phase 0-3 is a separate sequencing model. Tool lists and version-specific standards claims require a current review before adoption.

RESEARCH SNAPSHOT

- + Gravitee, State of AI Agent Security Report 2026 - deployed-agent monitoring and security survey.
- + Cloud Security Alliance, AI Agent Identity and Visibility Crisis, 2026 - agent versus human activity visibility.
- + Anthropic Engineering, How we built our multi-agent research system, 2025 - token-use and performance observations.

planeon.ai