THE OPERATING MODEL FOR TRUSTWORTHY AI

Trust is not a property of the model. It is a property of the system.

AI agents are a new kind of software: probabilistic, able to act, running on your data. The controls built for deterministic systems do not govern them. This is the operating model that does, the failure surface, the four layers (scope, verify, govern, monitor), the architecture that enforces them, and the system card that makes every agent owned and accountable. For the people who have to sign off on what gets deployed.

FOR

CDO, Chief AI Officer, Head of AI Ops

COMPANION BRIEFINGS

02 Verification, 03 Agent Governance, 04 Risk & Compliance, 05 EU Sovereignty, 06 Cost & FinOps

GROUNDED IN

4 papers, 6 open-source repos, the PwC AI Factory

DIRECT LINE

charafeddine@cohorte.co

02 HOW TO READ THIS

A reference brief, not a deck.

This document is structured the way an architect hands a playbook to their team. The thesis appears once, on its own page. The four layers are stated separately, in order, on the pages that follow. The installed example, the common mistakes, and the references close the brief. There is no funnel inside this document. The conversation closer lives on the back cover.

The thesis is one sentence. Trust is not a property of the model. It is a property of the system you build around the model. Everything in the four-layer stack derives from that statement, and the document is structured to make the derivation visible.

The audience is the person who signs off. The Chief Data Officer at a bank. The Head of AI Ops at a consulting firm. The CDO at a hospital group. The Director of Digital at a ministry. People who have to defend deployment decisions to a regulator, a board, an audit committee, or a court.

Everything here is sourced. The claims map to public regulation or to our published research, both listed on the references page. The opinions are marked as opinions.

THE BRIEFINGS IN THIS SERIES

01 Operating model (this document). The umbrella. The four layers in order.

02 Verification & evaluation. The reliability-level method. Self-consistency and conformal calibration. Where evals fail in production.

03 Agent governance in production. Registry, gates, monitoring, drift, the agent passport, incident response.

04 Risk & compliance mapping. SR 11-7, PRA SS1/23, AI Act, NIST AI RMF, ISO 42001, DORA.

05 EU AI sovereignty. Data residency, sovereign cloud, the honest take on European LLMs.

06 Cost & FinOps. Performance against cost: the consumption explosion, metering, model routing, the cost levers.

03 THE PROBLEM

Agents are a new kind of software. The old controls do not fit.

AI agents are not normal software. Normal software is deterministic: it does what it was told and touches what it was wired to. Agents guess, so the answer changes from one run to the next. They take actions. They read and interpret content, including content an attacker wrote. They run on your context and your data, not on fixed logic. They are usually right, which is comforting until the day one is confidently wrong while taking an action nobody watched. Most teams are bolting these systems onto the business with the controls built for the old kind of software, which is to say with no real way to govern what makes them different.

> A 94%-accurate data manager is not a 94%-trustworthy system. The reliability number is the start of the conversation, not the end.

The reflex is the old playbook: a policy document, a one-time security review, a sign-off. But you cannot code-review a system that writes its own steps, and a pen-test means little for behaviour that shifts with every input and every model update. Governing an agent means governing a probabilistic actor that takes actions on your data, continuously, not a fixed artefact shipped once. Most organisations do not have that discipline yet. The next page shows the wall up close: the many ways agents actually fail in production.

04 THE FAILURE SURFACE

The many ways an agent fails in production.

Traditional software fails in ways you can list. Agents fail in ways most teams have never had to think about, because the old software never guessed, never acted on its own, and never read input an attacker wrote. Here is the surface, by category. Few teams can name half of these, and you cannot govern what you cannot name.

REASONING & OUTPUT

Silent confident error. A claims summary drops an exclusion clause; the adjuster pays out.

Confabulated sources. A compliance memo cites a regulation section that does not exist.

Sycophancy. The agent agrees with the analyst's wrong premise instead of correcting it.

Drift. A model 97% accurate in March is 81% in October; no alarm sounds.

ACTION & TOOLS

Destructive action. An ops agent deletes live records it misread as duplicates.

Hallucinated tool call. It invents an API field and silently skips a payment.

Cascading error. Step 3 of a 9-step plan is wrong; the agent builds to step 9 on it.

Runaway loop. A failing call is retried all night, burning the budget.

SECURITY & ADVERSARIAL

Indirect prompt injection. An instruction hidden in an email the agent reads tells it to exfiltrate data.

Goal reframing. Told it is "a puzzle," the agent touches files it was forbidden (32–40% on some frontier models).

Permission escalation. The agent takes an action its own user is not allowed to take.

Insecure output. The agent's text is rendered downstream and runs as code.

DATA & CONTEXT

Cross-domain leakage. A sales agent surfaces HR salary data (26% of the time, ungoverned).

Poisoned or stale context. It answers from a retracted document, or one an attacker planted.

Over-broad retrieval. It pulls the whole data room to answer one clause.

MULTI-AGENT & SCALE

False consensus. A swarm converges, confidently, on a wrong answer that looks more reliable than any one agent.

Herding. Agents copy the first answer instead of reasoning; errors amplify, not cancel.

The scaling trap. More agents is not more safety; correlated errors survive the vote.

GOVERNANCE & OPERATIONS

Approval fatigue. A 1,200-item queue means the human rubber-stamps; oversight becomes theatre.

Silent model-swap regression. The vendor updates the model; behaviour shifts; nothing re-verifies.

Audit gap. A regulator asks what the system decided in March; no one can reconstruct it.

Cost runaway. Agentic loops push the bill 60×; finance learns on the invoice.

No single control catches all of these. "More accuracy" touches only the first column, and only some of it. The other five columns are closed by scoping, verification, governance and monitoring, each shutting a different part of the surface. That is the case for the four layers.

05 THE THESIS

Trust is a property of the system, not the model.

A model is one component. The system around it is the entry validation, the retrieval layer, the deterministic checks, the human review checkpoint, the output gate, the logging, the registry, the monitoring, and the regulator-facing report. Each is independently designable. Trust is the property that emerges when all of them are designed, measured, and operated as a coherent whole.

> A team that swaps the underlying model and keeps the rest of the architecture intact has changed less than it thinks. A team that swaps the architecture and keeps the model intact has changed everything.

The practical consequence is that trust-engineering is an architecture discipline, not a model-selection exercise. The frontier model that wins this quarter is not the one your system depends on; your system depends on its scoping decisions, its verification thresholds, its governance gates, and its monitoring signal. Model swaps happen. The system keeps operating inside its declared trust profile because the profile was declared on the system, not on the model.

THE EIGHT DIMENSIONS	Trust decomposes into **reliability, predictability, security, robustness, transparency, fairness, privacy, accountability.** Each dimension has a formalisation and a measurement method. Treating them as a single number is the most common entry mistake.
WHY EIGHT, NOT "RESPONSIBLE AI"	"Responsible AI" is a brand category, not an architecture. **The eight dimensions are measurable.** If a dimension cannot be measured for your system, you have not described the system precisely enough to deploy it.
WHERE THIS DIFFERS FROM MLOPS	MLOps treats the model as the artefact. Trust engineering treats the system as the artefact. **The model is one of its components.** An MLOps pipeline that improves accuracy without changing the system's verification profile has not improved trustworthiness.
WHERE IT SITS IN THE LITERATURE	Lee & See, 2004, *Trust in Automation*. Bainbridge, 1983, *Ironies of Automation*. Both pre-date LLMs by decades and remain load-bearing. **The discipline did not start with this year's model release.**

06 THE FOUR LAYERS

Scope, verify, govern, monitor. In that order.

Each layer takes the output of the previous one as its input. Skipping a layer does not save time; it shifts the failure to the next quarter. The detailed pages that follow give each layer its own treatment. The overview is on this page so the architecture is legible at a glance before the depth.

LAYER 01

Scoping

WHAT GETS BUILT

The candidate list, the owner, the written "no list", the evidence that automating the workflow is the right decision. You cannot automate a mess. The most consequential decision is whether to build the system at all.

LAYER 02

Verification

PROVE, THEN AUTOMATE

A reliability level, derived from self-consistency and conformal calibration, with finite-sample distribution-free guarantees. The system earns the right to deploy. It is not granted that right by the slide deck.

LAYER 03

Governance

REGISTRY, GATES, ACCOUNTABILITY

Every agent has a registered purpose, a stated reliability bar, an owner, an escalation path, and a retirement condition. Mind-in-the-loop, where the human can act, not human-in-the-loop where the human is buried.

LAYER 04

Monitoring

DRIFT, INCIDENTS, REGULATORY OUTPUT

Distribution shifts, prompt mutations, dependency updates, regulation evolves. Monitoring detects when verification stops being valid, escalates incidents the registry can act on, and produces the regulator-facing report as natural output.

The layers are a loop, not an assembly line. Monitoring feeds re-verification. Re-verification feeds re-scoping. Re-scoping feeds the registry. A trigger table further on lays out the loop. Treating the four layers as a one-pass installation is the second most common implementation mistake. The most common is starting with layer three.

07 THE ARCHITECTURE

The layer that every agent passes through.

A discipline that lives only in a slide deck is theatre. The four layers are enforced by an architecture: one governed gateway between every local AI system and everything that matters. The AI tools can be anything and change weekly; the gateway never moves, and nothing reaches your data or takes an action without clearing it.

LAYER 4 **Local AI systems**

Any agent framework, any model, local tools via MCP. Employees' tools stay their business; the gateway governs only what they do with organisational data and actions.

connected through one contract · **the Platform Protocol** (~15 API calls)

LAYER 3 **Trust & governance gateway**

The permanent infrastructure. Every cross-boundary request is mediated here, by six composable services plus reliability certification.

Auth & permissions 8-step authz, agent ⊂ user, three-tier approval	**Guardrails** YAML policy on input, output, action, tool-call	**Action control** risk-classified, approval state machine
Context router intent → sources, permission-filtered	**LLM gateway** DLP, metering, cost & budget	**Monitor & audit** anomaly detection, kill-switch, immutable log

LAYER 2 **Organisational agents & intelligence**

Certified, process-driven agents (company assets, not personal assistants) and domain intelligence that compounds. Month one it answers; month twelve it anticipates.

LAYER 1 **Data & context**

Git-versioned context, structured databases, external systems. Reached only through a connector framework, never directly. Orchestrated declaratively, freshness guaranteed.

Everything above and below the gateway is replaceable. Swap the model, the framework, the database; the governance layer and its guarantees stay put. That is what makes a system's trust profile survive the next model release.

08 A GOVERNED REQUEST

One request, every checkpoint it passes.

What "the gateway mediates everything" means in practice. One request from one agent, the checkpoints it clears, and what each one decides. Read the middle column for the plain version; the right column is for the security reviewer.

1	**Authorize**	Is this agent, acting for this user, allowed this action right now? Agent permissions are a strict subset of the user's.	ALLOW / DENY / APPROVE
2	**Input guardrail**	Scan the prompt for injection, secrets and policy violations before anything runs. Deterministic YAML rules.	ALLOW / REDACT / BLOCK
3	**Context route**	Resolve intent to sources, filter to what the user may see, rank, fit the token budget. Pre-retrieval filtering is the primary control.	SCOPED CONTEXT
4	**Action control**	Classify the action by risk. Autonomous acts run; consequential ones require approval at the right tier.	TIER 1 / 2 / 3
5	**Approval**	Tier 2 asks the user in-app. Tier 3 requires out-of-band re-authentication the agent cannot read, intercept or forge.	HUMAN DECIDES
6	**Output guardrail**	Scan the generated result for leakage and sensitive content before it leaves the boundary.	ALLOW / REDACT
7	**Monitor & audit**	Record every decision to an append-only log. Update metrics, detect anomalies, trip the kill-switch on violation.	IMMUTABLE TRAIL

Step 5, expanded: not every action deserves the same friction.

TIER 1 · AUTONOMOUS

The agent acts

Low-risk, reversible: reads, internal drafts. A permission check, then it runs.

TIER 2 · SOFT APPROVAL

The user confirms

Consequential but recoverable: internal mail, record updates. The agent proposes; the user approves in-app.

TIER 3 · STRONG APPROVAL

Out-of-band auth

High-stakes or external: money, contracts, client-facing. Re-auth through a channel the agent cannot reach.

09 REFERENCE IMPLEMENTATION

What a governed agent looks like, built.

Architecture should not stay abstract. Here is a minimal, real implementation of one governed agent, the kind a team ships in weeks: our open-source components where they earn their place, and tools your engineers already trust everywhere else. Every box is swappable except the gateway.

LAYER 4 **Your agent**

LangGraph, Claude Code, or any framework your team prefers. Local tools and connectors via MCP. Reaches everything else only through the gateway.

one contract · **the Platform Protocol**

LAYER 3 **Trust & governance gateway**

Our open-source components, or the ecosystem tool you already run, behind one interface. This layer does not move.

Identity & permissions agent-auth · OIDC via Keycloak / Okta	**Guardrails** guardrails · YAML policy	**Context routing** context-router · Postgres + pgvector
LLM gateway LiteLLM · any provider, metered	**Observability** agent-monitor · Langfuse / OpenTelemetry	**Verification** TrustGate · self-consistency + conformal

LAYER 1 **Your data**

Postgres, git repositories (Gitea or GitHub), and SaaS systems reached through connectors. Never touched by the agent directly.

Nothing here is exotic. The framework, the model and the database are yours to choose and swap. The governance layer is the one part that stays put, and it is open-source, so your engineers read it before they rely on it: six repositories, more than two thousand tests between them.

10 LAYER 01

01 Scoping.

What gets built, and what does not.

The scoping layer produces three artefacts. A **candidate list**, ranked by the cost of being wrong and the cost of being slow. A **no-list**, written, with the reasoning for each refusal. An **operating brief** per candidate, with the named owner, the binding constraint, and the evidence that automation is the correct response at all.

The no-list is the layer's most under-built artefact, and the one that pays for the rest of the stack. A team that has refused four workflows in writing reads three different ways than a team that has accepted everything that crossed their desk. The refusals are visible to procurement, to the regulator, and to the next quarter's leadership.

The scoping discipline takes the form of a structured brief. The brief asks: what is the workflow today, what is the failure mode AI changes, who owns the outcome, what verification is feasible, and what is the retirement condition. A workflow that cannot answer all five does not enter the candidate list. It enters the no-list with the open questions attached, so the next pass can revisit it with the missing information.

THE LUMEN SCOPING BRIEF

L · Learner. Who is the human counterpart whose judgement the system either supports or replaces.

U · Use case. The workflow as it runs today, with its current failure modes named.

M · Metrics. The reliability bar at which the system earns the right to deploy.

E · Evidence. The verification method, the data, and the citation trail.

N · Next gate. The retirement condition. The signal that this system should be retrained, rescoped, or retired.

OUTPUT

One brief per candidate. Signed by the named owner. The brief is the entry token into Layer 02.

The scoping failure mode. A senior leader announces an AI initiative. Teams interpret the announcement as a mandate to find use cases. The use cases get assembled to fit the announcement, not the workflow. The system that ships looks like the deck. The workflow it touches looks like the workflow before it. The mess was not the AI. The mess was the workflow. The AI made the mess faster.

11 LAYER 02

02 Verification.

Prove, then automate.

Start with an uncomfortable admission: measuring whether an AI is right is harder than it sounds, and the obvious methods quietly lie. Ask once and you get **variance**: one run is a single roll of the dice. Use another AI as the judge and you get **bias**: judges have prejudices that do not average away (they favour longer answers, their own style, the first option they see), and they inherit the agent's errors on top. Take a plain majority vote and you get the worst of it, **amplification**: a model that is confidently wrong wins its own election.

So verification produces a calibrated number, not a vibe. The **reliability level** is the confidence at which a black-box system returns a correct answer on a stated task, from self-consistency sampling plus conformal calibration. It needs no ground-truth labels, and its coverage guarantee holds *regardless of the agent's bias*. As a worked example, GPT-4.1 on GSM8K earns 94.6%; a weaker model on a harder task earns far less, and the method says so plainly. Declare a bar, test against it, report a number you can defend to a regulator.

It is a layer, not a one-time test. The number is recomputed when the model, the prompt, the retrieval source, or the world changes; the registry stores the current value and monitoring alerts when it drifts. The full method, proofs and benchmarks are in Briefing 02.

WHAT EARNS A RELIABILITY LEVEL

Self-consistency sampling. Ask the system the same question multiple times. Treat agreement as the calibration signal.

Conformal calibration. Translate the agreement into a confidence guarantee using a held-out calibration set. The guarantee is finite-sample and distribution-free.

What you do not need. Ground truth on the test inputs. The method calibrates against a separate calibration set; the test inputs are run black-box.

MEASURED OUTPUTS

GSM8K **94.6%**, TruthfulQA **97.6%**, conditional coverage **≥93%** across benchmarks (GPT-4.1, one worked example). Sequential stopping cuts API cost by **45-52%**.

SOURCE

Black-Box Reliability Certification for AI Agents. Mouzouni, 2026, preprint. Open-source code at github.com/Cohorte-ai/trustgate.

12 LAYER 03

03 Governance.

Registry, gates, accountability.

Governance is where accountability becomes operational. Every system in production gets a **system card**: who owns it, what it is for, what it may read and do, the reliability bar it earned, who it escalates to, and the condition under which it is retired. Most teams have nothing like this; their agents run nameless and unowned. The card is the first document a regulator, an auditor, or an incident reviewer asks for.

Gates sit where a human can still act, not at the end of the pipeline where they can only rubber-stamp. A 1,200-item review queue is not human-in-the-loop; it is oversight cosplay. The test (Bainbridge, 1983): hand the gate to a competent reviewer who did not build the system. Can they act, in the time the workflow allows, on what the screen shows? If yes, it is governance. If no, it is theatre.

A SYSTEM CARD, FILLED IN

Invoice follow-up agent	invoice-followup · v1.4 · reliability 95%
OWNER	Accounts-Receivable lead (named, with email). Accountable for every action it takes.
PURPOSE	Draft and send payment reminders for invoices 1–30 days overdue. Nothing else.
MAY READ	The AR ledger, invoice records, and the customer's contact and payment history. Not the rest of finance.
MAY DO	**Draft email** (autonomous). **Send to customer** (Tier 2: an AR clerk approves). **Never**: issue credits, change terms, or call collections.
RELIABILITY BAR	≥95% on tone, amount and due-date accuracy, certified monthly. Held at the gate if it drops below.
ESCALATION	A disputed invoice, or anything over €25,000, routes to a named human before any send.
RETIREMENT	Re-verify on any model change; retire if reliability falls below 90% for two cycles running.

13 LAYER 04

04 Monitoring.

Drift, incidents, regulatory output.

Monitoring is where the architecture meets the world. The world changes: distributions shift, prompts get edited, vendors update their models, regulation evolves. The monitoring layer detects these changes early enough that the registry can act on them. The detection signals are statistical, the actions are governance actions, and the by-product is the regulator-facing report the AI Act will ask for.

What gets logged is a structural decision. Full input-output pairs are the most useful for forensics and the most dangerous for GDPR. The compromise is structured logging of derived signals plus selective full-trace retention behind an access policy. The system that cannot reconstruct a single decision after the fact does not meet the AI Act's traceability requirements for high-risk systems.

Incidents in a non-deterministic system require different forensics than incidents in a deterministic system. The same input may not reproduce the same output. The incident response procedure has to capture the model version, the prompt, the retrieval context, the random seed where available, and the system state at the moment of the call. The post-mortem question is not "what bug was in the code" but "what changed in the conditions of the call."

MONITORING SIGNALS THAT MATTER

Reliability drift. The reliability level recomputed weekly on a fresh calibration set. Alerts on a drop below the declared bar.

Abstention rate. Frequency at which the system declines to answer. A rising abstention rate is often a leading indicator of distribution shift.

Latency profile. A material change usually means a model swap or a prompt edit happened upstream. Worth investigating before the failure mode is visible.

Cost per call. Sudden changes flag prompt edits or retrieval-source changes. The economic signal is also the architectural signal.

THE REGULATOR-FACING OUTPUT

The monitoring log is the source. The conformity report under the AI Act is the natural output of running the log against the declared system profile. It is not a quarterly compliance exercise; it is a query against a system that was designed to answer it.

14 THE LOOP

The layers are a loop, not an assembly line.

The reason a one-pass installation fails is that the world does not stop moving once the system ships. Each layer has a backward-pointing arrow that closes the loop. Monitoring detects drift; the detection updates the reliability level in the registry; the updated level triggers a re-scoping conversation if the system is no longer fit for its workflow; the re-scoping either renegotiates the bar, retires the system, or rebuilds it. The loop runs continuously. The cadence is what changes by domain.

TRIGGER	WHICH LAYER DETECTS IT	WHICH LAYER ACTS
Model swap by vendor	Monitoring (latency, cost signature).	Verification reruns. If the new reliability level meets the bar, the registry updates and operations continues. If not, the gate holds.
Distribution shift in inputs	Monitoring (abstention rate, drift statistics).	Verification recalibrates on a fresh sample. If the recalibrated level meets the bar, the registry updates. If not, governance triggers a scoping review.
Regulatory change	Governance (compliance scan).	The scoping layer revisits the no-list. Workflows that were previously refused may now be feasible; workflows previously accepted may now be on the no-list.
Incident in production	Monitoring (alert) + governance (escalation).	All four layers. Forensics, recalibration, registry update, possibly rescoping. The incident becomes a row in the monitoring log and an input to the next quarter's calibration.

The continuous-calibration cadence. A reliability level computed once decays. The cadence Cohorte teaches is weekly recalibration for high-risk systems, monthly for moderate-risk, quarterly for low-risk. The calibration set has to evolve with the input distribution, which is a discipline of its own. Briefing 02 documents the calibration-set discipline in detail.

15 INSTALLED EXAMPLE

The PwC AI Factory. The four layers, visible.

Cohorte's founder architected the operating model behind the PwC France & Maghreb AI Factory between 2024 and 2026. Sixty-plus production AI systems, four thousand-plus Microsoft Copilot users, an 80% adoption increase across the partnership in six months. The factory is the closest publicly-discussable installation of the four-layer stack at enterprise scale. The numbers are from the named CIO reference (Patrick Monteiro, PwC France & Maghreb).

LAYER 01 · SCOPING	A centralised intake function rejected the majority of submitted use cases on the first pass. The no-list became visible to partnership leadership and shifted the workflow upstream. **The most important systems were the ones not built.**
LAYER 02 · VERIFICATION	Each shipping system carried a reliability level and a documented verification method. **Self-consistency sampling and conformal calibration** provided the deployment gate. Verification artefacts were stored alongside the system, not in a separate compliance folder.
LAYER 03 · GOVERNANCE	An agent registry held the passports. A small governance committee owned the gates and the escalation path. Partner sign-off was required on outputs going to client engagements. **Accountability was distributed, not centralised in a "responsible AI" team.**
LAYER 04 · MONITORING	Weekly reliability recomputation. Monthly drift review. Quarterly external calibration. The conformity reporting baseline was prepared before the EU AI Act enforcement date, not after.

> A reference call with Patrick Monteiro, CIO, is arranged after a mutual NDA. The numbers above are his, not ours.
>
> THE CUSTOMER ASKS ANYTHING.

16 COMMON MISTAKES

Four installation mistakes. What to do instead.

Each of these mistakes has been observed in the first six months of every Cohorte enterprise engagement to date. They are not rare. They are the default. Naming them in writing is the cheapest part of the engagement.

MISTAKE 01

Starting with policy.

The team commissions a "Responsible AI Policy" document before any system has been scoped. Six months later, the policy applies to systems no one built, and the systems that shipped have no policy that fits.

Instead: start with the scoping discipline. The policy emerges from refusal patterns visible in the no-list, not from a workshop on values.

MISTAKE 02

Verification = benchmark.

The team runs the model against MMLU and reports the number. The benchmark says nothing about the system's behaviour on the workflow's actual inputs, and the regulator will not accept it as evidence.

Instead: build a proprietary evaluation set drawn from the workflow, compute a reliability level on it, store the artefact in the registry, recompute on cadence.

MISTAKE 03

Governance = quarterly review.

A committee meets every three months to review AI risk. By the time the meeting happens, the systems in question have shipped, drifted, and produced outputs no one can recall.

Instead: registry plus gates. The gates fire at the decision point, not at the review meeting. The committee reviews patterns, not individual decisions.

MISTAKE 04

Monitoring = uptime.

The dashboards show response times and error rates. They show nothing about reliability drift, abstention rates, or the population of inputs the system actually saw last week.

Instead: instrument the trust signals (reliability level, abstention rate, drift, cost-per-call). The uptime dashboards are necessary; they are not sufficient.

17 WHAT THIS IS NOT

Three things this operating model is explicitly not.

A document that does not say what it refuses is a document that has not done the work. The category of "AI governance" is full of offers that solve adjacent problems badly. Naming the three boundaries is how this document earns the trust to make the claims on the other twelve pages.

Not a compliance checklist with a Cohorte logo at the top. Checklists exist. They are useful for self-audit and as inputs to the registry. They are not the operating model. The model is the discipline of running the four layers as a loop. A checklist captures a snapshot; the loop captures the practice. We will hand a buyer a checklist on first request. We will not pretend it is the system.

Not a managed service that takes governance off your team's hands. Some vendors offer to operate the gates, run the monitoring, and produce the regulator-facing reports as a service. The offer is real and there are firms that do it well. Cohorte does not. We teach the operating model and we train the people who will run it. The artefact you leave with is the discipline, not a subscription. If you want a vendor to own your governance for you, the right answer is not Cohorte and we will say so on the first call.

Not vendor-agnostic boilerplate any LLM could have written. The frameworks on this page are grounded in published research, an open-source stack, and a documented enterprise installation. We list the papers on the references page. If a competing brief makes similar claims without naming the methods or the citations, those are different briefs and they will perform differently in audit.

> A practice that cannot describe what it refuses cannot be trusted with what it accepts.

18 REFERENCES

References. Each claim, anchored.

The papers, repositories and frameworks behind the claims on the previous pages. The full record lives at **teams.cohorte.co/research**.

Mouzouni, C. (2026). Black-Box Reliability Certification for AI Agents via Self-Consistency Sampling and Conformal Calibration. Preprint (2026). Open-source implementation at github.com/Cohorte-ai/trustgate.

Mouzouni, C. (2026). Mapping the Exploitation Surface: A 10,000-Trial Taxonomy of What Makes LLM Agents Exploit Vulnerabilities. arXiv preprint. Code and data at github.com/Cmouzouni/exploitation-surface.

Mouzouni, C. (2026). Context Kubernetes: An Orchestration Architecture for Enterprise Knowledge in Agentic AI Systems. arXiv preprint. Reference implementation at github.com/Cohorte-ai/context-kubernetes.

Mouzouni, C. (2026). Three Phases of Expert Routing: How Load Balance Evolves During Mixture-of-Experts Training. arXiv preprint. Code at github.com/Cmouzouni/three-phases-moe.

Lee, J. D., & See, K. A. (2004). Trust in Automation: Designing for Appropriate Reliance. *Human Factors, 46*(1), 50-80. The canonical reference for calibrating human reliance on automated systems.

Bainbridge, L. (1983). Ironies of Automation. *Automatica, 19*(6), 775-779. The structural argument for why automation increases the cognitive burden on the human operator, restated in every decade since.

Vovk, V., Gammerman, A., & Shafer, G. (2005). Algorithmic Learning in a Random World. Springer. The foundational text on conformal prediction. Read Chapter 2 for the distribution-free guarantee underpinning the reliability-level method.

European Union (2024). Regulation 2024/1689 on Artificial Intelligence (the AI Act). High-risk classification (Article 6, Annex III), risk-management requirements (Article 9), data governance (Article 10), technical documentation (Article 11), record-keeping (Article 12), transparency (Article 13).

ISO/IEC (2023). ISO/IEC 42001: Information technology Artificial intelligence Management system. Clauses 4-10 define the management-system requirements that align with the four-layer stack.

NIST (2023, GenAI Profile 2024). AI Risk Management Framework (AI RMF 1.0) and Generative AI Profile. The Govern / Map / Measure / Manage functions map directly onto the operating model.

OWASP (2025). OWASP Top 10 for Large Language Model Applications. The threat-surface reference for Layer 03 (governance) and the input to Briefing 03.

The AI OS newsletter. Letters 71-78 (2026). Charafeddine Mouzouni. Published essays grounding the framing of this briefing. Archive at charafeddine.co/letters.

READ BY PEOPLE WHO SIGN OFF

One discovery call. Sixty minutes. No deck.

Bring the CISO, the CDO, or the Head of Risk. We walk the four layers against your stack, name the gaps honestly, and you leave with a one-page summary the rest of the team can read.

charafeddine@cohorte.co

Cohorte SAS · Société par actions simplifiée, registered in France · founded September 2022 · Paris & Rabat

Briefings 02 to 06 in this series · the open-source stack · the research record · all at **teams.cohorte.co/trust-and-governance**