Governance Before Decision

A Field Guide to Proof, Permission, and Human-Controlled AI Execution

Kari Hyötylä

TRUSTCORE / PUBLIC PREVIEW

Governance Before Decision

A Field Guide Preview for Proof, Permission, and Human-Controlled AI Execution

Public Preview · Version 1.3 · September 2026
Kari Hyötylä · TrustCore AI Systems Oy

PUBLIC PREVIEW · VERSION 1.3

Revision note - Version 1.3 / September 2026
Retired product references and an obsolete product URL have been removed.
The general remediation example and the authorization boundary are retained.
Public Preview v1.2 remains a separate historical edition.

trustcore.fi

AUTHOR

Kari Hyötylä

FOUNDER, TRUSTCORE AI SYSTEMS OY

AUTHOR OPENING

Why governance begins before execution

> A correct answer is not permission to act.

Most AI governance starts by asking whether an output is accurate, safe, or explainable.

This book asks what comes next - at the point where an AI-generated recommendation could become a real-world consequence.

When a system can draft, decide, trigger, deploy, approve, or send, the critical question is no longer only whether the output is good.

THE CRITICAL QUESTION

Who has the authority to let that action happen?

Governance Before Decision is a field guide for that boundary.

It is about separating capability from authority, recommendation from permission, and intelligence from the authority to execute.

Contents

A concise path from model capability to controlled consequence.

CORE THESIS

Real governance begins where execution can still be refused.

01 / OPENING THESIS

A Correct Answer Is Not Permission to Act

AI governance discussions often begin with accuracy.

> Was the answer correct?
> Did the model cite a real source?
> Did the patch compile?
> Did the tests pass?

Those questions matter. They are not enough.

A technically correct output can still be inappropriate to disclose, send, approve, merge, deploy, or execute. The missing condition may have nothing to do with model quality. It may concern identity, authority, timing, evidence, policy, or the consequence that follows.

> A correct AI output is not permission to act.

Consider an illustrative software-remediation workflow.

An AI system identifies a real vulnerability, prepares a valid patch, and passes the available tests. The proposal is technically sound.

Now add the operating context.

The target is a payment service. A production freeze has started. Another approved deployment is changing the same authentication component. The engineer requesting the fix may prepare a change, but does not hold authority to deploy it.

The patch remains correct.

Execution is not currently permitted.

Nothing about the recommendation needs to be rejected. The organization simply refuses to confuse technical confidence with operational authority.

Governance becomes real where the system can still stop before consequence.

02 / ARCHITECTURE

The Missing Layer: Capability, Checking, Enforcement

A useful architecture separates three layers.

CAPABILITY

Generates, retrieves, plans, recommends, drafts, or acts.

What can the system produce or attempt?

CHECKING

Tests or reviews an output or proposal.

Does this output appear acceptable?

ENFORCEMENT

Evaluates required conditions before consequence.

Is this action permitted here and now?

Checking and enforcement are not rivals. They solve different problems.

A checker can identify defects. An enforcement boundary can prevent a class of consequence when required conditions are missing.

Existing policy-engine patterns provide a useful engineering precedent. Open Policy Agent, for example, separates policy decision-making from the application that requests a decision. OPA is not an AI-governance standard, but the architectural lesson is relevant: the component that wants to act does not need to be the component that decides whether policy permits the action.[1]

> The model may recommend. The workflow may prepare. The enforcement boundary decides whether capability may become consequence.

03 / CONSEQUENCE

Why Action Changes the Risk

An answer and an action may come from the same model, but they do not deserve the same boundary.

A summary can be questioned. A recommendation can be ignored. A draft can be revised. A customer message can leave the organization. A deployment can change production. A payment or contract commitment can become difficult or impossible to reverse.

The stronger the consequence, the less acceptable assumed permission becomes.

Action class	Example	Reversibility	Suggested boundary
Informational	Summarize a policy	High	Disclose evidence limits
Advisory	Recommend a remediation	Medium	Domain-sensitive review
Draft / preparation	Prepare a message or pull request	High	Human review before promotion
External communication	Send to a customer or regulator	Variable	Explicit send gate
Transactional	Payment, access change, production deployment	Low	Specific current authorization
Irreversible / cascade	Contract commitment, destructive action, multi-step workflow	Very low or unknown	Workflow-level gate and human authorization where required

This is a proposed field-guide model, not a universal legal classification.

Its purpose is simple: consequence should shape the gate.

Security guidance for LLM applications already recognizes that risk increases when systems receive excessive functionality, permissions, or autonomy. OWASP's 2025 guidance recommends least privilege, user approval for high-impact actions, and authorization in downstream systems rather than relying on the model to decide whether an action is allowed.[2]

> Human authority must exist at the point where it can still change the outcome.

04 / PERMISSION FLOW

The Gate Map

A governed workflow separates recommendation, permission, and execution.

1. Recommendation
2. Evidence assembly
3. Identity and authority verification
4. Consequence classification
5. Policy and current-context evaluation
6. Human authorization where required
7. Gate decision
8. Controlled execution
9. Decision trace

RECOMMENDATION

What appears technically or operationally useful? A recommendation may come from a model, an agent pipeline, a deterministic scanner, or a human.

EVIDENCE

What supports the recommendation, and what remains unknown? For consequential action, content should not travel alone. Downstream components need enough structured context to understand what was checked, what was not checked, how current the evidence is, and which assumptions were made.

IDENTITY AND AUTHORITY

Who is requesting the action, in what capacity, under what scope, and through which delegation chain? A familiar channel, plausible wording, or a prompt that says "act on behalf of the administrator" is not proof of authority.

> Context is not identity. Instruction is not delegated authority.

CONSEQUENCE AND PRESENT STATE

What real-world effect can follow, and can it be reversed? Does the authorization still apply now? Approvals age. Environments change. Incidents begin. Freezes are declared. Evidence becomes stale. Scope changes.

DECISION

ALLOW
Current conditions satisfy the gate.

VERIFY
Additional evidence or human authorization is required.

STOP
The action cannot proceed under current authority, policy, or consequence conditions.

REROUTE
Move the work to a safer workflow, reduced scope, or human-controlled path.

The recommendation may be technically correct in all four cases.

05 / DECISION EVIDENCE

A Log Is Not an Audit Trail

Most systems can show that something happened. A consequential system also needs to explain why it was allowed.

A request arrived. A model ran. An output was produced. A tool was called. A deployment completed.

That is a log.

A consequential AI system also needs to explain why the action was allowed.

> What evidence existed?
> What was not verified?
> Which identity and authority were active?
> Which policy version applied?
> Did a human authorize the consequence, or only review an intermediate artifact?
> Did a gate explicitly allow the action, or did no gate run?

That is the beginning of an audit trail.

The difference is not the amount of data. It is the purpose of the record.

> A log reconstructs chronology. An audit trail reconstructs the decision.

NIST's AI Risk Management Framework provides voluntary risk-management guidance built around governance, context, measurement, and management across the AI lifecycle. The EU AI Act separately establishes logging and human-oversight requirements for systems within its high-risk scope. Neither source requires the exact architecture in this preview. Both support the broader point that roles, oversight, monitoring, documentation, and traceability must be operational rather than implied.[3][4]

For high-consequence workflows, a decision record should help distinguish:

- explicit permission from absence of a check;
- current authorization from an old session assumption;
- sufficient evidence from a confident-looking summary;
- human authorization from a generic approval event;
- a reversible action from a point of no return.

Tamper-evident transparency-log designs offer an adjacent technical precedent. RFC 9162 describes append-only, auditable certificate logs. It is not an AI-audit standard. It shows that record integrity and independent verification can be architectural properties rather than promises.[5]

A source can support a decision. The record must show how the decision used it.

06 / OWNERSHIP

Who Owns Refusal?

A technical gate cannot own itself.

Before an AI system is allowed to create external consequence, an organization must decide who owns the threshold, who may change enforcement configuration, who can authorize high-consequence action, who reviews the decision record, who can invoke an override, and who can refuse.

Weak governance assigns these responsibilities implicitly. Stronger governance gives them named owners, decision rights, maintenance duties, and escalation paths.

Buyer checklist

Question	Weak answer	Stronger answer
Who can change the gate?	"The AI team."	A named control owner reviews and approves changes.
Who owns the consequence threshold?	"It depends."	Ownership is documented, versioned, and reviewed.
Who can authorize high-consequence action?	"A manager approves."	Named roles authorize defined scope and context.
Who reviews the decision record?	"Compliance checks later."	Operational review has cadence and escalation rules.
Who can override the gate?	"Admins can."	Override is exceptional, scoped, recorded, and reviewed.
Can the configuration be rolled back?	"We can edit it."	Versioned known-good states and rollback exist.

Human oversight is meaningful only when the human has the authority, information, and timing needed to intervene or stop the system.

> Governance becomes real when refusal has an owner, a runtime location, an evidence requirement, a decision record, and an escalation path.

07 / PRACTICAL CHECKLIST

The Pre-Execution Question Set

Before AI-assisted work becomes action, the organization should be able to answer ten questions.

1. **Who is making this request?**
2. **In what capacity are they acting?**
3. **What action class is being requested?**
4. **What evidence supports the recommendation?**
5. **What was not checked?**
6. **What real-world consequence could follow?**
7. **Can that consequence be reversed?**
8. **Which policy and authority govern this action now?**
9. **Who can still refuse, stop, or reroute it?**
10. **What record will prove the decision later?**

> A "yes" from a model is not a send command.

A successful test is not deployment authority.

A completed workflow is not proof of permission.

The final boundary must remain explicit.

08 / IMPLEMENTATION

From Principle to Implementation

This preview does not prescribe one product or vendor.

Organizations can begin with identity systems, approval workflows, CI/CD controls, policy engines, audit infrastructure, and clearly bounded human decision rights. The first step is to identify the point where recommendation becomes consequence and make that transition explicit.

Governed remediation and execution authorization address separate questions.
A technically grounded change still needs current permission to proceed.

- Remediation: prepare and validate a technically grounded change.
- Authorization: determine whether the proposed change may proceed under current conditions, with explicit human authority before consequence.

The product is not the evidence for the principle.

The architecture must earn trust through observable behavior and reviewable records.

This preview does not claim certification, guaranteed safety, customer adoption, production maturity, or regulatory approval. It asks a practical question:

> Where does your organization currently prove that a good AI recommendation is also permitted to act?

AI systems will continue to retrieve more information, coordinate more agents, use more tools, and complete longer workflows.

Greater capability does not eliminate organizational judgment. It increases the value of placing judgment at the right boundary.

The enduring question is not only:

Can the AI do this?

It is:

Can we demonstrate why this action was allowed to happen - and could we still have refused before consequence?

That is governance before decision.

09 / SOURCES

References

[1] **Open Policy Agent documentation.** Engineering precedent for separating policy decisions from the software requesting them. OPA describes itself as a general-purpose policy engine that offloads and decouples policy decision-making from software. openpolicyagent.org/docs

[2] **OWASP GenAI Security Project, LLM06:2025 Excessive Agency.** Guidance addressing excessive functionality, permissions, and autonomy, including least privilege, user approval for high-impact actions, downstream authorization, and complete mediation. genai.owasp.org

[3] **NIST AI Risk Management Framework 1.0, NIST AI 100-1; and NIST AI 600-1, Generative AI Profile.** Voluntary governance and risk-management background. NIST states that AI RMF 1.0 is being revised. nist.gov/itl/ai-risk-management-framework

[4] **Regulation (EU) 2024/1689, especially Articles 12 and 14.** Logging and human-oversight background for systems within the Regulation's relevant high-risk scope. EUR-Lex

[5] **RFC 9162, Certificate Transparency Version 2.0.** Append-only and independently auditable record analogy only; RFC 9162 is an Experimental RFC and not an AI-audit standard. rfc-editor.org/rfc/rfc9162

These sources support adjacent governance, security, policy, and audit concepts. They do not endorse TrustCore, establish this preview's terminology as legal standards, or prove compliance with any regulation. Sources checked 17 July 2026.

TRUSTCORE / NEXT STEP

Make the boundary visible.

Start with one workflow where AI-generated work can create an external consequence.

1 · Find the transition

Where does recommendation become execution?

2 · Classify consequence

What can happen, and can it be reversed?

3 · Define proof

Which evidence, identity, authority, and policy must be current?

4 · Name refusal

Who can still stop, verify, or reroute the action?

5 · Preserve the decision

What record will prove why the action was allowed?

6 · Test the gate

Can the system refuse before consequence, not only explain afterward?

> TrustCore is developing practical approaches to governed remediation, runtime execution gates, decision traces, and human-controlled execution boundaries.

trustcore.fi **Work with us** **linkedin.com/in/TrustCore**