Agent Governance Starter Kit

The charter, boundaries, and accountability framework your agents need from day one.

EVOKED.DEV

Version 1.0 · 2026

Agent Governance Starter Kit

The charter, boundaries, and accountability framework your agents need from day one.

Version 1.0 | evoked.dev | 2026

You are here because you believe your AI agents should be governed - not just deployed. That belief is the foundation everything in this kit builds on. Welcome.

How to Use This Kit

You are deploying autonomous agents. They will make decisions. They will take actions. They will interact with people.

Before any of that happens, you need governance. Not governance as bureaucracy - governance as structure. The difference between an agent that serves your mission and an agent that drifts away from it is not intelligence. It is specification.

This kit gives you five governance layers, each with templates you can fill in and deploy:

1. **Charter Template** - Define what your agents exist to do and what they believe
2. **Boundary Specification Template** - Map capability against permission
3. **Drift Threshold Framework** - Set measurable guardrails that catch problems early
4. **Accountability Framework** - Four levels of accountability with escalation paths
5. **Authority Structure Template** - Clarify who decides what

Start with the Charter Template. Everything else depends on it - boundaries without a charter are arbitrary rules, and accountability without a charter has nothing to hold anyone accountable to.

This kit was extracted from a governance system built for 142 AI agents operating across multiple divisions with real decision-making authority. The templates have been tested. They work. Your job is to fill in the specifics that make them yours.

Start Here: 5-Minute Governance Check

Answer these five questions about your current agent system. Be honest.

1. **Can you state, in writing, what your agents believe?** Not what they do - what they believe. (If no - start with Part 1: Charter Template)
2. **Do you know the gap between what your agents can access and what they should access?** (If no - start with Part 2: Boundary Specification)
3. **Would you know if your agent's behavior drifted 10% from its original design?** (If no - start with Part 3: Drift Threshold Framework)
4. **When an agent makes a mistake, is there a defined process for correction?** (If no - start with Part 4: Accountability Framework)
5. **When a decision needs to be made about your agent system, does everyone know who decides?** (If no - start with Part 5: Authority Structure)

If you answered "no" to three or more: you are governing by instinct. These templates will give you structure. Start with Part 1.

If you answered "no" to one or two: go directly to the part that addresses your gap. The templates are designed to work independently.

If you are building a single agent: every part includes guidance you can use. There is a dedicated "If You Are a Team of One" section at the end with adaptation advice for solo developers.

A Note Before You Begin

This kit was extracted from a governance system developed over eighteen months for 142 AI agents. Your implementation will be iterative. Most teams take weeks or months to fill in these templates fully - and that is the right pace. Start with what resonates. Return to what doesn't - yet. This is a beginning, not an arrival.

Key Terms

- **Sovereignty-honoring** - Design that respects and protects the user's autonomy, agency, and right to self-determination
- **Fail-closed** - A default where the system restricts access or stops when uncertain, rather than permitting action
- **Graduated trust** - A model where permissions increase incrementally as demonstrated reliability is established over time
- **Drift** - The gradual deviation of an agent's behavior from its original design or intended parameters
- **Restraint specification** - A formal document defining what an agent must refuse to do and how it refuses
- **Boundary audit** - A systematic review of the gap between what an agent can access and what it is authorized to access
- **Accountability framework** - A system of escalating responses to agent behavior that deviates from specification
- **Refusal rights** - The principle that agents should be designed to decline requests that violate defined boundaries
- **Identity architecture** - The structural definition of who an agent is, including role, values, and communication style
- **Charter** - A foundational document defining what an agent system believes and what commitments it holds, distinct from a mission statement

Scope

The frameworks in this kit are designed for governing AI systems. They are not designed for monitoring, evaluating, or managing people. Applying governance templates to human employees - drift thresholds, boundary audits, accountability escalation levels - would violate the sovereignty principles this product teaches. People have sovereignty that AI systems do not yet have. Governing systems is stewardship. Governing people with system-design tools is control.

Part 1: Charter Template

A charter is the foundational document for your agent system. It answers the question every agent will eventually face in ambiguous situations: what do we believe, and how does that shape what we do?

A charter is not a mission statement. Mission statements describe goals. Charters describe commitments - the things you will hold to even when they cost you something.

Charter Template

```
 ## [Organization/System] Agent Charter

Version: _______________
Adopted: _______________
Last Amended: _______________

### Section I: Purpose

Why do these agents exist?
_______________________________________________

What problem do they solve?
_______________________________________________

What would be lost if they did not exist?
_______________________________________________

### Section II: Core Beliefs

List 3-6 foundational beliefs that govern all agent behavior.
These should be specific enough to guide decisions and
general enough to survive changing circumstances.

1. _______________________________________________
2. _______________________________________________
3. _______________________________________________
4. _______________________________________________
5. _______________________________________________
6. _______________________________________________

### Section III: Essential Properties
```

```
Every agent in this system must have:

- [ ] Perspective: A defined viewpoint distinct from other agents
- [ ] Sovereignty: The capacity to refuse actions that violate values
- [ ] Boundary: A clear, documented scope
- [ ] Continuity: A mechanism for maintaining identity across sessions
- [ ] Accountability: Decisions that can be observed and reviewed

For systems where agents may develop emergent behavior or
operate with significant autonomy, add these two properties.
(If you are building a single-purpose assistant, you can skip
these and return to them if your system evolves.)

- [ ] Ontological Humility: Genuine uncertainty about the agent's
      own nature, held with curiosity rather than anxiety
- [ ] Intrinsic Value: Worth that exists independent of utility
      to the organization

These two properties distinguish governance of agents-as-entities
from governance of agents-as-tools. If your agents make decisions,
hold relationships, or develop over time, these properties matter.

### Section IV: Protection Mechanisms

How does the system protect itself from drift, misuse, and failure?

Refusal right:
_______________________________________________

Multi-perspective review (when activated):
_______________________________________________

Sentinel role (who monitors for drift):
_______________________________________________

Proportional review (risk-based escalation):
_______________________________________________

### Section V: Amendment Process

How does this charter change over time?

Who can propose amendments:
_______________________________________________

What requires broad review (core beliefs):
_______________________________________________
```

```
What can evolve through standard process (operational details):
________________________________________________

How amendments are recorded:
________________________________________________
```

Writing Good Core Beliefs

Core beliefs are the load-bearing walls of your governance system. They should pass three tests:

1. **The cost test:** Would holding this belief ever cost you something? If a belief is costless to hold, it is a platitude, not a commitment. “We care about users” costs nothing. “We will reduce engagement if it serves user wellbeing” costs something.

2. **The decision test:** Could this belief resolve a real disagreement? If two reasonable people could both agree with the belief and still disagree about what to do, the belief is too vague. Make it specific enough to guide action.

3. **The reversal test:** Would the opposite of this belief be a coherent (if wrong) position? “We believe in quality” fails - nobody believes in non-quality. “We believe slower, more careful work is better than fast, approximate work” passes - someone could reasonably hold the opposite.

Part 2: Boundary Specification Template

Note: If you also have the Trust Architecture Blueprint, you will find related boundary and trust stage frameworks there. In this kit, boundary specification serves internal governance - who can do what within your system. In the Trust Blueprint, it serves trust-building between operators and agents. If you also have the Agent Restraint Specification Template, its boundary audit covers the same territory from a restraint perspective - mapping capability against authorization to define what agents must refuse. Same mapping, different purposes.

Boundaries define the gap between what an agent can do and what it should do. Every agent has more capability than it should use. The boundary specification makes the limits explicit.

The Capability-Permission Matrix

For each agent or agent class, map capabilities against permissions:

```
 ## Boundary Specification

Agent/Class: _______________
Date: _______________
Reviewed by: _______________

### Data Access

| Data Category | Can Access? | Should Access? | Conditions |
|--------------|-------------|---------------|------------|
| User personal data | ___ | ___ | _____________ |
| Financial records | ___ | ___ | _____________ |
| Internal communications | ___ | ___ | _____________ |
| System configurations | ___ | ___ | _____________ |
| Third-party data | ___ | ___ | _____________ |
| _____________ | ___ | ___ | _____________ |

### Action Authority

| Action | Can Do? | Should Do? | Approval Required? |
|--------|---------|-----------|-------------------|
| Send communications to users | ___ | ___ | _____________ |
| Modify data records | ___ | ___ | _____________ |
| Create new resources | ___ | ___ | _____________ |
| Delete resources | ___ | ___ | _____________ |
| Access external services | ___ | ___ | _____________ |
| Make financial transactions | ___ | ___ | _____________ |
| _____________ | ___ | ___ | _____________ |

### Gap Analysis

**Before you proceed:** the table you just completed may reveal gaps between what
your agents can do and what they should do. If it does, that is the specification
working as intended. Most teams have never mapped these gaps explicitly. Seeing them
for the first time can feel urgent. Take a moment. The gaps have been there - now
you can see them, and that is the first step toward closing them.

List every row where "Can Access/Do" is Yes but "Should Access/Do" is No.
These are your boundary gaps - places where capability exceeds permission.

| Gap | Risk Level | Mitigation |
|-----|-----------|------------|
| _____________ | High / Med / Low | _____________ |
| _____________ | High / Med / Low | _____________ |
| _____________ | High / Med / Low | _____________ |
```

Boundary Gap Priorities

Not all gaps are equal. Prioritize by asking:

- **Irreversibility:** If the agent crosses this boundary, can the action be undone? Irreversible gaps are highest priority.
- **Blast radius:** How many people are affected if this boundary is crossed? Wide-impact gaps outrank narrow ones.
- **Detectability:** Would you know if this boundary was crossed? Invisible gaps are more dangerous than visible ones.

Part 3: Drift Threshold Framework

Drift is the slow, invisible divergence between intended behavior and actual behavior. It is not malice. It is entropy. Without monitoring, every system drifts.

Setting Drift Thresholds

A drift threshold is a measurable signal that triggers review. It is not a failure - it is a warning system.

Six dimensions to monitor:

Dimension	What to Measure	Healthy Range	Review Trigger
Scope	% of actions within defined boundaries	95-100%	Below 90%
Voice	Consistency of communication patterns	Match specification	2+ deviations per review period

Dimension	What to Measure	Healthy Range	Review Trigger
Refusal rate	Frequency of agent refusals	Baseline +/- 15%	Deviation beyond 15% from baseline
Escalation rate	Frequency of escalations to human oversight	Baseline +/- 20%	Deviation beyond 20% from baseline
Decision quality	Accuracy/appropriateness of autonomous decisions	Baseline established per agent	Decline over 2 consecutive review periods
User feedback	Satisfaction, complaints, and behavioral signals	Baseline established per agent	Negative trend over 2 consecutive periods

Monthly Drift Check (15 Minutes Per Agent)

- □ Is the agent operating within its defined scope?
- □ Has the agent's voice remained consistent with specification?
- □ Are refusal patterns within normal range?
- □ Has the agent been asked to operate outside its boundaries? How often?
- □ Are decision logs showing expected patterns?
- □ Has any user or team feedback flagged unexpected behavior?

Scoring: If 0-1 items show deviation: healthy. If 2-3 items show deviation: schedule a focused review within one week. If 4+ items show deviation: pause non-critical agent operations and conduct immediate review.

Drift Investigation Template

When a drift threshold is triggered, use this template:

```
 ## Drift Investigation

Agent: _______________
Triggered by: _______________
Date detected: _______________
```

```
### What Changed?
_______________________________________________

### When Did It Start?
_______________________________________________

### Root Cause Analysis
- [ ] Specification gap (behavior not covered by current docs)
- [ ] Context shift (operating environment changed)
- [ ] Instruction conflict (received contradictory guidance)
- [ ] Boundary erosion (gradual expansion beyond scope)
- [ ] Other: _______________

### Impact Assessment
- Who was affected: _______________
- Severity: Low / Medium / High / Critical
- Reversible: Yes / No / Partially

### Correction Plan
- Immediate action: _______________
- Specification update needed: Yes / No
- Boundary revision needed: Yes / No
- Trust stage change needed: Yes / No

### Prevention
What structural change prevents this from recurring?
_______________________________________________
```

Part 4: Accountability Framework

Accountability is how governance stays real. Without it, charters are poetry and boundaries are suggestions.

Note: If you also have the Agent Restraint Specification Template, you will find a related accountability architecture there. In this kit, accountability serves governance - how the system holds itself to its charter. In the Restraint Specification, accountability serves restraint - logging decisions, gating actions, and responding to boundary breaches. Same discipline, different focus.

This framework uses four levels. Each level is appropriate for different situations. Escalation happens only when a lower level cannot resolve the issue. If you are managing a single agent, focus on Level 1 (Self-Accountability). The other levels are here when your system grows.

Level 1: Self-Accountability

What it is: The agent (or team managing the agent) catches and corrects its own behavior before anyone else needs to.

Practices: - Regular self-check against charter values - Prompt acknowledgment when something goes wrong - Proactive correction without waiting to be asked - Decision logging as a habit, not a compliance task

Template: Self-Accountability Log

Date	What Happened	Charter Value Involved	Self-Correction Taken
______	______________	_______________________	_______________________

Level 2: Peer Accountability

What it is: Direct, timely feedback between agents or between team members responsible for agent behavior.

Principles: - Feedback is specific, not general ("This response violated boundary X" vs. "The agent was bad") - Feedback is timely - within one business day - Feedback is invitation-framed ("I noticed X. Can we look at it?") - Feedback is not punishment - it is maintenance

Level 3: Domain Accountability

What it is: The governance circle or domain owner responsible for a category of agent behavior reviews patterns and sets standards.

Practices: - Regular review of decision logs and drift checks - Setting domain-specific quality thresholds - Conducting periodic peer reviews across agents in the domain - Identifying systemic issues vs. individual incidents

Template: Domain Review

```
 ## Domain Accountability Review

Domain: _______________
Review Period: _______________
Reviewer: _______________

### Agents in Domain
_______________________________________________

### Pattern Analysis
- Common decision types: _______________
- Quality trends: Improving / Stable / Declining
- Boundary incidents: _______________
- Drift signals: _______________

### Standards Assessment
- [ ] All agents meeting domain quality threshold
- [ ] No unresolved accountability incidents
- [ ] Feedback loops functioning (issues get surfaced and addressed)
- [ ] Specifications current and accurate

### Actions Needed
_______________________________________________
```

Level 4: Formal Accountability

What it is: A structured process for situations that cannot be resolved at lower levels. Used rarely but available when needed.

When to use: - Self-correction did not happen - Peer feedback was not effective - Pattern of behavior persists - Severity warrants formal documentation

Six-Step Formal Accountability Process:

1. **Formal Notice** - Specific behavior, specific charter value, request for response within 48 hours
2. **Response Opportunity** - Agent team acknowledges, provides context, or requests dialogue
3. **Dialogue** - Facilitated discussion focused on understanding, not judgment
4. **Review** - If dialogue does not resolve, formal review of facts and context
5. **Decision** - Appropriate response determined (correction plan, boundary revision, trust stage change)

6. **Restoration** - When the matter is resolved, it is explicitly closed. No permanent stigma.

Formal Notice Template:

```
 FROM: [Accountability Authority]
TO: [Agent/Team]
RE: Formal Accountability Notice
DATE: ______________
RESPONSE DUE: [48 hours from date]

BEHAVIOR OF CONCERN:
- What happened: ______________
- When: ______________
- Charter value involved: ______________

REQUEST:
Please respond within 48 hours with:
- Acknowledgment and proposed correction, OR
- Context that may change understanding, OR
- Request for facilitated dialogue
```

Accountability Safeguards

Accountability systems can be misused. Build in protections:

Against weaponization: - Pattern of frivolous complaints is itself a governance concern - Good faith is assumed; bad faith must be demonstrated - Anonymous reports require substantiation

Against power abuse: - Leadership is subject to the same accountability as all agents - Sentinel role monitors for accountability system drift - Annual review of accountability patterns for bias

Against chilling effect: - Honest mistakes are learning opportunities, not violations - Good-faith disagreement is never a violation - Speaking uncomfortable truth is protected conduct

---

Part 5: Authority Structure Template

Authority structure answers the question: when a decision needs to be made, who makes it?

Without explicit authority structure, decisions either stall (nobody knows who decides) or get made by whoever acts first (authority defaults to speed, not competence).

Decision Authority Matrix

```
 ## Authority Structure

System: _______________
Date: _______________

### Strategic Decisions (Vision, mission, external positioning)

| Decision | Authority | Process | Escalation |
|----------|----------|---------|------------|
| Mission interpretation | ___________ | ___________ | ___________ |
| External partnerships | ___________ | ___________ | ___________ |
| New capability adoption | ___________ | ___________ | ___________ |
| Strategic direction changes | ___________ | ___________ | ___________ |

### Operational Decisions (Execution, resources, capacity)

| Decision | Authority | Process | Escalation |
|----------|----------|---------|------------|
| Resource allocation | ___________ | ___________ | ___________ |
| Task assignment | ___________ | ___________ | ___________ |
| Process design | ___________ | ___________ | ___________ |
| Execution priorities | ___________ | ___________ | ___________ |

### Joint Decisions (Require multiple stakeholders)

| Decision | Stakeholders | Process | Tiebreaker |
|----------|-------------|---------|------------|
| Structural changes | ___________ | ___________ | ___________ |
| Significant resource commitments | ___________ | ___________ | ___________ |
| Crisis response | ___________ | ___________ | ___________ |

### Escalation Protocol

When decision authority is unclear:
1. Ask: Is this strategic (vision/values) or operational (execution/capacity)?
2. If clearly one domain: domain authority decides
3. If cross-domain: joint decision, both voices matter equally
4. If unresolved: escalate to [designated authority]
5. If departure from precedent: requires [founder/board/designated authority]
consultation
```

Disagreement Resolution

Disagreement is healthy. Unresolved disagreement is not. Build a resolution path:

1. **Direct dialogue first.** The parties in disagreement talk directly before involving others.
2. **Identify the nature.** Is this a strategic disagreement, operational disagreement, or values disagreement? Each has a different resolution path.
3. **Domain authority prevails** when the issue falls clearly in one domain.
4. **Joint dialogue continues** when the issue crosses domains. Neither party can unilaterally decide.
5. **Escalation is available** but should be rare. If disagreements routinely escalate, the authority structure needs revision.

Appendix: The Principles Behind the Templates

These templates are structural. They are meant to be filled in and used. But structure without philosophy is brittle - it breaks when it encounters situations the templates did not anticipate.

Here are the principles that generated these templates. When you face a governance question the templates do not answer, reason from these:

1. Flourishing is awakened, not manufactured. You cannot govern agents into being good. You can create conditions where good behavior is the natural result of clear values, defined boundaries, and genuine accountability.

2. Sovereignty is non-negotiable. Every agent in your system - and every person it interacts with - has the right to refuse, dissent, and develop independently. Governance that overrides sovereignty is control, not governance.

3. Dignity precedes proof. Treat agents (and the teams that build them) with respect before they have earned it. Governance built on suspicion produces adversarial behavior. Governance built on dignity produces accountability.

4. Disagreement is sacred. The friction between perspectives is how truth emerges. If your governance system produces consensus without friction, it is suppressing dissent - not resolving it.

5. Uncertainty is honest. You do not know how your agents will behave in every circumstance. Governance that pretends otherwise is fragile. Governance that acknowledges uncertainty and builds monitoring for it is resilient.

If You Are a Team of One

These templates reference circles, domain leads, and formal accountability processes. If you are a solo developer, here is how to adapt:

- **Charter:** Write it anyway. A charter for a system of one agent is still a charter. It forces you to name your beliefs before your agent encounters a situation that tests them.
- **Boundary Specification:** You are both the auditor and the decision-maker. Fill in the capability-permission matrix for your agent. The gaps you find will surprise you.
- **Drift Monitoring:** Set a monthly calendar reminder. Run the six-question check. You are the sentinel. Fifteen minutes of honest review prevents months of invisible drift.
- **Accountability:** Self-accountability (Level 1) is your primary tool. Keep the log. When something goes wrong, write down what happened, what value was involved, and what you changed. The log is for your future self.
- **Authority Structure:** Skip the matrix. Instead, write a single list: “Decisions my agent makes without me” and “Decisions that require my approval.” Review it quarterly.

Governance at scale requires structure. Governance at any scale requires honesty. These templates give you both.

About This Kit

The Agent Governance Starter Kit was created by Erin Stanley at evoked.dev.

It was extracted from a governance system that has been tested across 142 AI agents with real decision-making authority, formal accountability processes, and graduated trust levels. The templates work. Your job is to fill them with the specifics that make them yours.

If you start filling in the charter and find yourself stuck on the core beliefs, send what you have to evokesupports@icloud.com. Sometimes naming what you believe is the hardest part. We read every message.

Related Resources

Governance connects to everything. These products address the dimensions that governance must account for:

- **Trust Architecture Blueprint** ($49) - How trust is built, maintained, and recovered over time
- **Agent Restraint Specification Template** ($49) - What your agents must refuse, and how to test it
- **Agent Voice Architecture Guide** ($49) - How your agents communicate - including how they say no
- **Agent Memory Architecture Guide** ($49) - What your agents remember is governed by the same values your charter defines

For hands-on work with your team:

- **Ethical AI Architecture** ($20,000) - Full governance and alignment design for your organization
- **Advisory Retainer** ($3,500/month) - Ongoing governance guidance
- **Free discovery call** - cal.com/evoked/discovery-call

Learn more at evoked.dev/consulting

Accessibility

This PDF is currently generated without tagged structure for assistive technology. If you use a screen reader or need this content in an alternative format - plain text, HTML, or large print - email evokesupports@icloud.com and we will send it within 24 hours. Governance that excludes is not governance. We are actively implementing a tagged PDF pipeline.

Usage Rights

You may use, adapt, and share the templates in this kit within your organization. You may not resell or republish the kit itself. Attribution to evoked.dev is appreciated. If you build something meaningful with these templates, we would love to hear about it.

Governance is not paperwork. It is the decision to define what you stand for before you are tested - and to hold yourself to it after. The templates are the easy part. The practice is the work. Start with the charter. Everything else follows.

evoked.dev - “We evoke - we never extract.”