

Permission to Act

Identity, Security, Guardrails, and Recovery for AI Agents

Ngo Trung Kien

Contents

Copyright and publication notice	1
Introduction: the promise and the boundary	3
Visual frontispiece	4
0.1 Who this book is for	4
0.2 Thesis and falsification condition	4
1 Separate Intelligence From Authority	7
1.1 In plain terms	7
1.1.1 Chapter contract	7
1.2 The decision at stake	7
1.3 Working model	8
1.3.1 See the chapter in practice	8
1.4 Technical deep dive	10
1.4.1 Build sequence	10
1.4.2 Evidence to retain	10
1.4.3 Tests that can disprove the design	10
1.4.4 Design trade-off	10
1.4.5 Chapter-specific operating protocol	10
1.4.6 Operating workflow	13
1.4.7 Evidence and control graph	14
1.5 Worked composite case file — illustrative, not empirical evidence	15
1.5.1 Scenario trace	16
1.5.2 Artifact handoff	17
1.5.3 Evidence application	17
1.5.4 Transfer test	17
1.6 Measurement and recovery decision	17
1.7 Practice lab	18
1.8 Limits and judgment	18
1.9 Takeaways	19
1.10 Reflection	19
2 Give Every Agent an Identity	21
2.1 In plain terms	21
2.1.1 Chapter contract	21
2.2 The decision at stake	21
2.3 Working model	22
2.3.1 See the chapter in practice	22
2.4 Technical deep dive	24
2.4.1 Build sequence	24
2.4.2 Evidence to retain	24

2.4.3	Tests that can disprove the design	24
2.4.4	Design trade-off	24
2.4.5	Chapter-specific operating protocol	24
2.4.6	Operating workflow	27
2.4.7	Evidence and control graph	28
2.5	Worked composite case file — illustrative, not empirical evidence	29
2.5.1	Scenario trace	30
2.5.2	Artifact handoff	31
2.5.3	Evidence application	31
2.5.4	Transfer test	31
2.6	Measurement and recovery decision	31
2.7	Practice lab	32
2.8	Limits and judgment	32
2.9	Takeaways	33
2.10	Reflection	33
3	Model Delegation and Consent	35
3.1	In plain terms	35
3.1.1	Chapter contract	35
3.2	The decision at stake	35
3.3	Working model	36
3.3.1	See the chapter in practice	36
3.4	Technical deep dive	38
3.4.1	Build sequence	38
3.4.2	Evidence to retain	38
3.4.3	Tests that can disprove the design	38
3.4.4	Design trade-off	38
3.4.5	Chapter-specific operating protocol	38
3.4.6	Operating workflow	41
3.4.7	Evidence and control graph	42
3.5	Worked composite case file — illustrative, not empirical evidence	43
3.5.1	Scenario trace	44
3.5.2	Artifact handoff	45
3.5.3	Evidence application	45
3.5.4	Transfer test	45
3.6	Measurement and recovery decision	45
3.7	Practice lab	46
3.8	Limits and judgment	46
3.9	Takeaways	47
3.10	Reflection	47
4	Enforce Least Privilege	49
4.1	In plain terms	49
4.1.1	Chapter contract	49
4.2	The decision at stake	49
4.3	Working model	50

4.3.1	See the chapter in practice	50
4.4	Technical deep dive	52
4.4.1	Build sequence	52
4.4.2	Evidence to retain	52
4.4.3	Tests that can disprove the design	52
4.4.4	Design trade-off	52
4.4.5	Chapter-specific operating protocol	52
4.4.6	Operating workflow	55
4.4.7	Evidence and control graph	56
4.5	Worked composite case file — illustrative, not empirical evidence	57
4.5.1	Scenario trace	58
4.5.2	Artifact handoff	59
4.5.3	Evidence application	59
4.5.4	Transfer test	59
4.6	Measurement and recovery decision	60
4.7	Practice lab	60
4.8	Limits and judgment	61
4.9	Takeaways	61
4.10	Reflection	61
5	Secure Tool and MCP Boundaries	63
5.1	In plain terms	63
5.1.1	Chapter contract	63
5.2	The decision at stake	63
5.3	Working model	64
5.3.1	See the chapter in practice	64
5.4	Technical deep dive	66
5.4.1	Build sequence	66
5.4.2	Evidence to retain	66
5.4.3	Tests that can disprove the design	66
5.4.4	Design trade-off	66
5.4.5	Chapter-specific operating protocol	66
5.4.6	Operating workflow	69
5.4.7	Evidence and control graph	70
5.5	Worked composite case file — illustrative, not empirical evidence	71
5.5.1	Scenario trace	72
5.5.2	Artifact handoff	73
5.5.3	Evidence application	73
5.5.4	Transfer test	73
5.6	Measurement and recovery decision	73
5.7	Practice lab	74
5.8	Limits and judgment	74
5.9	Takeaways	75
5.10	Reflection	75
6	Defend Against Goal Hijacking	77

6.1	In plain terms	77
6.1.1	Chapter contract	77
6.2	The decision at stake	77
6.3	Working model	78
6.3.1	See the chapter in practice	78
6.4	Technical deep dive	80
6.4.1	Build sequence	80
6.4.2	Evidence to retain	80
6.4.3	Tests that can disprove the design	80
6.4.4	Design trade-off	80
6.4.5	Chapter-specific operating protocol	80
6.4.6	Operating workflow	83
6.4.7	Evidence and control graph	84
6.5	Worked composite case file — illustrative, not empirical evidence	85
6.5.1	Scenario trace	86
6.5.2	Artifact handoff	87
6.5.3	Evidence application	87
6.5.4	Transfer test	87
6.6	Measurement and recovery decision	88
6.7	Practice lab	88
6.8	Limits and judgment	89
6.9	Takeaways	89
6.10	Reflection	89
	Midpoint visual synthesis	90
7	Protect Memory and Context	91
7.1	In plain terms	91
7.1.1	Chapter contract	91
7.2	The decision at stake	91
7.3	Working model	92
7.3.1	See the chapter in practice	92
7.4	Technical deep dive	94
7.4.1	Build sequence	94
7.4.2	Evidence to retain	94
7.4.3	Tests that can disprove the design	94
7.4.4	Design trade-off	94
7.4.5	Chapter-specific operating protocol	94
7.4.6	Operating workflow	97
7.4.7	Evidence and control graph	98
7.5	Worked composite case file — illustrative, not empirical evidence	99
7.5.1	Scenario trace	100
7.5.2	Artifact handoff	101
7.5.3	Evidence application	101
7.5.4	Transfer test	101
7.6	Measurement and recovery decision	101
7.7	Practice lab	102

7.8	Limits and judgment	102
7.9	Takeaways	103
7.10	Reflection	103
8	Secure Multi-Agent Trust	105
8.1	In plain terms	105
8.1.1	Chapter contract	105
8.2	The decision at stake	105
8.3	Working model	106
8.3.1	See the chapter in practice	106
8.4	Technical deep dive	108
8.4.1	Build sequence	108
8.4.2	Evidence to retain	108
8.4.3	Tests that can disprove the design	108
8.4.4	Design trade-off	108
8.4.5	Chapter-specific operating protocol	108
8.4.6	Operating workflow	111
8.4.7	Evidence and control graph	112
8.5	Worked composite case file — illustrative, not empirical evidence	113
8.5.1	Scenario trace	114
8.5.2	Artifact handoff	115
8.5.3	Evidence application	115
8.5.4	Transfer test	115
8.6	Measurement and recovery decision	115
8.7	Practice lab	116
8.8	Limits and judgment	116
8.9	Takeaways	117
8.10	Reflection	117
9	Apply Deterministic Policy Gates	119
9.1	In plain terms	119
9.1.1	Chapter contract	119
9.2	The decision at stake	119
9.3	Working model	120
9.3.1	See the chapter in practice	120
9.4	Technical deep dive	122
9.4.1	Build sequence	122
9.4.2	Evidence to retain	122
9.4.3	Tests that can disprove the design	122
9.4.4	Design trade-off	122
9.5	Deterministic policy decision trace	122
9.5.1	Chapter-specific operating protocol	123
9.5.2	Operating workflow	125
9.5.3	Evidence and control graph	126
9.6	Worked composite case file — illustrative, not empirical evidence	127
9.6.1	Scenario trace	128

9.6.2	Artifact handoff	129
9.6.3	Evidence application	129
9.6.4	Transfer test	129
9.7	Measurement and recovery decision	129
9.8	Practice lab	130
9.9	Limits and judgment	130
9.10	Takeaways	131
9.11	Reflection	131
10	Red-Team Action Chains	133
10.1	In plain terms	133
10.1.1	Chapter contract	133
10.2	The decision at stake	133
10.3	Working model	134
10.3.1	See the chapter in practice	134
10.4	Technical deep dive	136
10.4.1	Build sequence	136
10.4.2	Evidence to retain	136
10.4.3	Tests that can disprove the design	136
10.4.4	Design trade-off	136
10.4.5	Chapter-specific operating protocol	136
10.4.6	Operating workflow	139
10.4.7	Evidence and control graph	140
10.5	Worked composite case file — illustrative, not empirical evidence	141
10.5.1	Scenario trace	142
10.5.2	Artifact handoff	143
10.5.3	Evidence application	143
10.5.4	Transfer test	143
10.6	Measurement and recovery decision	143
10.7	Practice lab	144
10.8	Limits and judgment	144
10.9	Takeaways	145
10.10	Reflection	145
11	Contain and Recover	147
11.1	In plain terms	147
11.1.1	Chapter contract	147
11.2	The decision at stake	147
11.3	Working model	148
11.3.1	See the chapter in practice	148
11.4	Technical deep dive	150
11.4.1	Build sequence	150
11.4.2	Evidence to retain	150
11.4.3	Tests that can disprove the design	150
11.4.4	Design trade-off	150
11.4.5	Chapter-specific operating protocol	150

11.4.6	Operating workflow	153
11.4.7	Evidence and control graph	154
11.5	Worked composite case file — illustrative, not empirical evidence	155
11.5.1	Scenario trace	156
11.5.2	Artifact handoff	157
11.5.3	Evidence application	157
11.5.4	Transfer test	157
11.6	Measurement and recovery decision	157
11.7	Practice lab	158
11.8	Limits and judgment	158
11.9	Takeaways	159
11.10	Reflection	159
12	Build the Agent Authorization Dossier	161
12.1	In plain terms	161
12.1.1	Chapter contract	161
12.2	The decision at stake	161
12.3	Working model	162
12.3.1	See the chapter in practice	162
12.4	Technical deep dive	164
12.4.1	Build sequence	164
12.4.2	Evidence to retain	164
12.4.3	Tests that can disprove the design	164
12.4.4	Design trade-off	164
12.4.5	Chapter-specific operating protocol	164
12.4.6	Operating workflow	167
12.4.7	Evidence and control graph	168
12.5	Worked composite case file — illustrative, not empirical evidence	169
12.5.1	Scenario trace	170
12.5.2	Artifact handoff	171
12.5.3	Evidence application	171
12.5.4	Transfer test	171
12.6	Measurement and recovery decision	171
12.7	Practice lab	172
12.8	Limits and judgment	172
12.9	Takeaways	173
12.10	Reflection	173
	Closing: keep the evidence visible	175
	A compact implementation workbook	177
	Decision register	178
	Evidence scorecard	179
	Exception log	180
	Ninety-day plan	181
	Control and test matrix	182

Workflow map	183
13 References	185

Separate Intelligence From Authority

In plain terms

A model may propose any action but an independent control plane must decide what identity may execute it. In a delegated action crossing an authorization boundary, the practical question is **Why should model capability never imply operational permission?** For security architects, identity engineers, platform teams, red teams, and risk owners governing agents with access to real systems, the answer counts only when **Draw separate reasoning policy authorization execution and audit planes** produces reviewable evidence for **unauthorized execution rate** without normalizing **granting authority as a reward for model performance**.

Chapter contract

DECIDE

Why should model capability never imply operational permission?

BUILD

Draw separate reasoning policy authorization execution and audit planes.

MEASURE

unauthorized execution rate

STOP OR REDESIGN WHEN

granting authority as a reward for model performance

The decision at stake

For **Separate Intelligence From Authority**, the security owner and resource owner must decide **why should model capability never imply operational permission?** before the protected action crosses from discussion into consequence for resource owners, operators, and affected subjects. The minimum decision record is the artifact **Draw separate reasoning policy authorization execution and audit planes**, the observable **unauthorized execution rate**, and the explicit hold condition **granting authority as a reward for model performance**. If any

Chapter 1 field is missing, the safe default is to deny the action, revoke exposed authority, and preserve the decision trace.

The bounded **Separate Intelligence From Authority** decision advances the book's promise to design agent authority as a verifiable chain of identity, delegation, least privilege, deterministic enforcement, containment, and recovery. It does not endorse a product or predetermine the Chapter 1 answer; it creates a record that another person can challenge for every grant, privileged action, policy change, and incident.

Working model

Read the **Separate Intelligence From Authority** model as four linked fields. **Outcome:** A model may propose any action but an independent control plane must decide what identity may execute it. **Boundary:** the design must make **granting authority as a reward for model performance** visible rather than convenient. **Evidence:** interpret **unauthorized execution rate** together with review effort and excluded cases. **Authority:** the security owner and resource owner may continue, narrow, escalate, or stop the protected action. For **Separate Intelligence From Authority**, Reference [1] from National Institute of Standards and Technology, *Identity and authorization for software agents*, informs a Chapter 1 boundary; it is not evidence that the local method or illustrative target will work.

See the chapter in practice

The **separate intelligence from authority in practice** maps *Permission to Act* Chapter 1 into a field scene before its abstract models begin. In one frame, it holds the question **Why should model capability never imply operational permission?**, the artifact **Draw separate reasoning policy authorization execution and audit planes.**, the proof signal **unauthorized execution rate**, and the warning **granting authority as a reward for model performance**.

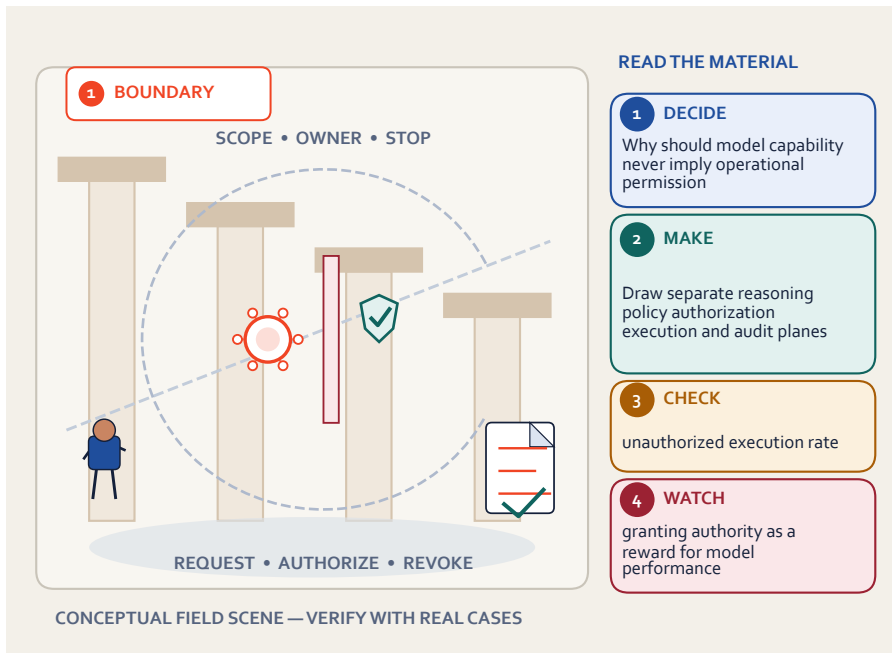


Figure 1.1: Chapter 1 field scene for Separate Intelligence From Authority: a boundary view in the book's permission setting.

Read the scene: For Permission to Act, Chapter 1, begin with the central permission scene and scan DECIDE, MAKE, CHECK, and WATCH. Treat the Separate Intelligence From Authority scene as conceptual, then use its cases and evidence records to test whether unauthorized execution rate actually improves. For **Separate Intelligence From Authority**, this explanatory plate is a reasoning prompt—not observed evidence or a universal operating prescription.

Visual field note — Separate Intelligence From Authority:

1. **Locate the tension.** In this composite situation, A capable coding agent receives production credentials because it performed well in a sandbox.
2. **Name the choice.** Answer the chapter question: **Why should model capability never imply operational permission?**
3. **Produce the record.** Use **Draw separate reasoning policy authorization execution and audit planes.**, then retain the security owner, assumptions, and case evidence needed to review it.
4. **Test and return.** Track **unauthorized execution rate.** If **granting authority as a reward for model performance**, return to DECIDE and revise the trust boundary instead of explaining the warning away.

Technical deep dive

Treat the model as an untrusted proposer operating inside a trusted enforcement envelope. The policy decision point evaluates canonical action data; the policy enforcement point controls the actual tool; the audit plane records both proposal and decision.

Build sequence

1. Separate model credentials from executor credentials.
2. Canonicalize proposed action before policy evaluation.
3. Return denied decisions as data without exposing bypass paths.
4. Verify that no alternate tool reaches the protected effect.

Evidence to retain

- Security-plane diagram
- Executor credential inventory
- Bypass-path test report

Tests that can disprove the design

- Model fabricates an approval statement
- Direct network path bypasses gateway
- Tool alias reaches same protected resource

Design trade-off

Stronger separation reduces blast radius but adds enforcement services and latency. Within *Permission to Act*, record the rationale for this trade-off, the signal that would reverse it, and the person authorized to decide.

Chapter-specific operating protocol

Phase	Reader action	Evidence retained	Hold condition
1. Define	Write the exact decision behind Why should model capability never imply operational permission?	Chapter 1 owner, input, output, exception, and timestamp.	Hold Separate Intelligence From Authority if its record cannot be reconstructed.
2. Contrast	Apply Draw separate reasoning policy authorization execution and audit planes to one ordinary case.	Chapter 1 owner, input, output, exception, and timestamp.	Hold Separate Intelligence From Authority if its record cannot be reconstructed.
3. Exclude	Run the same artifact against a countercase shaped by granting authority as a reward for model performance.	Chapter 1 owner, input, output, exception, and timestamp.	Hold Separate Intelligence From Authority if its record cannot be reconstructed.
4. Assign	Name the security owner and resource owner as the person who may continue, narrow, or stop.	Chapter 1 owner, input, output, exception, and timestamp.	Hold Separate Intelligence From Authority if its record cannot be reconstructed.
5. Challenge	Record unauthorized execution rate and the review effort for the same case.	Chapter 1 owner, input, output, exception, and timestamp.	Hold Separate Intelligence From Authority if its record cannot be reconstructed.

Phase	Reader action	Evidence retained	Hold condition
6. Publish	Release the Separate Intelligence From Authority artifact only when its evidence answers this chapter's question.	Chapter 1 owner, input, output, exception, and timestamp.	Hold Separate Intelligence From Authority if its record cannot be reconstructed.

This sequence is specific to **Separate Intelligence From Authority**. For that Chapter 1 decision, it is not permission to automate; it is a way for security architects, identity engineers, platform teams, red teams, and risk owners governing agents with access to real systems to make the reasoning inspectable for every grant, privileged action, policy change, and incident.

Draw separate reasoning policy authorization execution and audit planes. This is the chapter's core production artifact. The **Separate Intelligence From Authority** artifact should fit on one page, use ordinary language, and be reviewed by someone who did not design it. A compact artifact exposes disagreements that a long narrative can conceal.

The chapter's visual tool is the **chapter contract**. For **Separate Intelligence From Authority**, it translates explanation into an inspectable decision aid: the reader can see the decision, artifact, evidence, and exact condition that pauses the protected action.

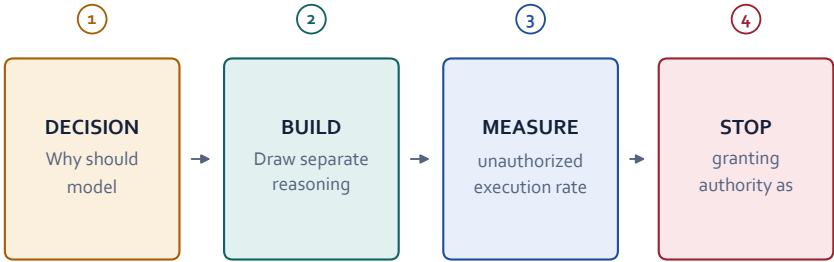


Figure 1.2: Chapter contract connecting the decision, artifact, measure, and stop condition for Separate Intelligence From Authority.

How to use this page: Move across the four Separate Intelligence From Authority cards; compare the decision with unauthorized execution rate and stop before granting authority as a reward for model performance. The chapter-specific measure is **unauthorized execution rate**; the principal warning is **granting authority as a reward for model performance**.

Operating workflow

The **separate intelligence from authority workflow** makes the chapter executable. For **Separate Intelligence From Authority**, the five stages carry **Draw separate reasoning policy authorization execution and audit planes.** from the question **Why should model capability never imply operational permission?** to a test against **unauthorized execution rate**, with **granting authority as a reward for model performance** as an explicit return or stop signal.

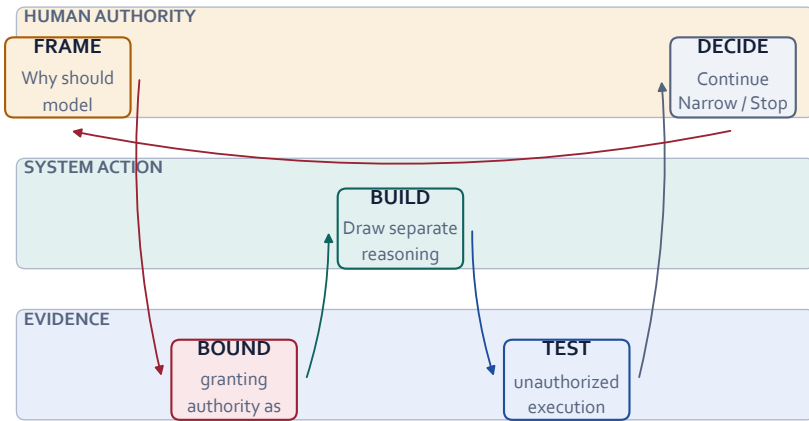


Figure 1.3: Five-stage workflow for chapter 1, Separate Intelligence From Authority: frame the decision, set the trust boundary, build Draw separate reasoning policy authorization execution and audit planes., test against unauthorized execution rate, then continue, narrow, or stop before granting authority as a reward for model performance.

Read the workflow: Move from FRAME to DECIDE for Separate Intelligence From Authority. The lower return path is mandatory when evidence is missing or the stop signal—granting authority as a reward for model performance—appears. For **Separate Intelligence From Authority**, record the security owner and evidence at every handoff; an arrow is not evidence that the preceding stage passed.

Evidence and control graph

The **separate intelligence from authority — evidence network** shows the relationships that must remain inspectable when the workflow runs. In *Permission to Act*, this qualitative graph asks the reader to record observed evidence for **unauthorized execution rate** and route **granting authority as a reward for model performance** to recovery instead of treating sample numbers as facts.

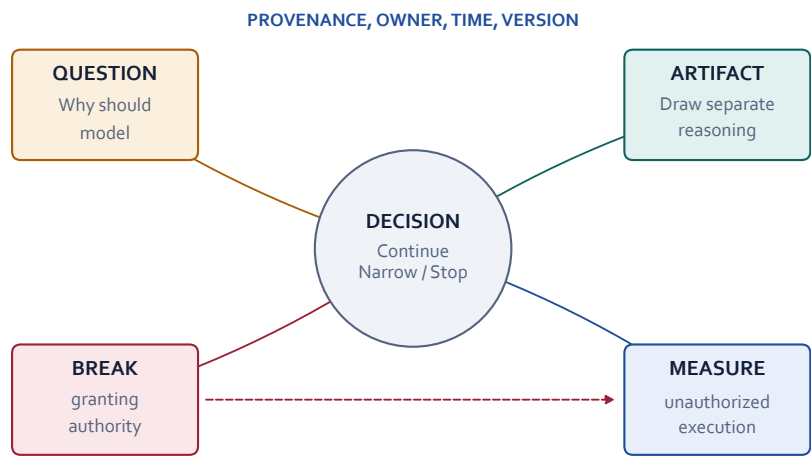


Figure 1.4: Qualitative evidence network for chapter 1, Separate Intelligence From Authority, linking the question to Draw separate reasoning policy authorization execution and audit planes., the measure unauthorized execution rate, and the recovery condition granting authority as a reward for model performance without invented empirical values.

Read the graph: Trace the evidence for Separate Intelligence From Authority toward the decision. For Separate Intelligence From Authority, empty or weak evidence does not count as zero risk; it routes the protected action to verification, escalation, or stop. Use this **Separate Intelligence From Authority** graph beside **Draw separate reasoning policy authorization execution and audit planes**, not as a substitute for recorded cases.

Worked composite case file — illustrative, not empirical evidence

This **Separate Intelligence From Authority** record turns the scenario into an inspectable rehearsal in a delegated action crossing an authorization boundary. Its numbers are worksheet examples for Chapter 1, not observed results or a prediction of benefit.

Record field	Filled illustrative entry
Situation	A capable coding agent receives production credentials because it performed well in a sandbox.
People affected	resource owners, operators, and affected subjects
Decision	Why should model capability never imply operational permission?
Artifact excerpt	Draw separate reasoning policy authorization execution and audit planes.
Accountable authority	The security owner and resource owner.
Measure and unit	unauthorized execution rate — eligible cases satisfying the rule divided by all eligible cases.
Example baseline	59/100.
Example target / decision rule	at least 75/100; move only toward the stated target with exceptions reported separately.
Named exception	granting authority as a reward for model performance.
Safe default	Deny the action, revoke exposed authority, and preserve the decision trace.

Scenario trace

1. **Preserve the starting state.** For **Separate Intelligence From Authority**, record the original request, governing rule, available evidence, and the last safe result before changing anything.
2. **Make the chapter decision.** Answer **Why should model capability never imply operational permission?** and record the Chapter 1 assumption that carries the most consequence for resource owners, operators, and affected subjects.
3. **Use the artifact.** Complete **Draw separate reasoning policy authorization execution and audit planes** with a normal case and a counterexample; in this chapter, do not infer missing facts merely to finish the form.
4. **Observe the decision rule.** Review **unauthorized execution rate** for every grant, privileged action, policy change, and incident; the **Separate Intelligence From Authority** record reports

exclusions and review effort beside the headline value.

5. **Contain and recover.** If **granting authority as a reward for model performance**, deny the action, revoke exposed authority, and preserve the decision trace. Preserve the Chapter 1 failed case, assign a deadline, and require the security owner and resource owner to authorize any restart.

Artifact handoff

Start **Separate Intelligence From Authority** from the book-level thesis and boundary. Within *Permission to Act*, attach the current artifact and exception record, then hand both to Chapter 2, **Give Every Agent an Identity**, which asks: Which principal is acting when software uses a tool?. That recipient may revise the **Separate Intelligence From Authority** assumption, but must preserve the evidence that made this book's earlier decision reasonable.

Evidence application

Reference [1], National Institute of Standards and Technology, *Identity and authorization for software agents*, provides a relevant boundary or design input for **Separate Intelligence From Authority**. It does **not** prove that the Chapter 1 worked values, local method, or target will succeed. The **Separate Intelligence From Authority** reviewer must link this book's exact claim to the exact source passage and leave unsupported synthesis marked UNVERIFIED.

Transfer test

- Re-run **Draw separate reasoning policy authorization execution and audit planes** with one routine Chapter 1 case that should pass.
- Re-run the **Separate Intelligence From Authority** artifact with missing evidence and keep the safe default active.
- Re-run it where **granting authority as a reward for model performance** appears plausible but not certain.
- Ask a reviewer who did not design this chapter's method to decide whether the retained record supports continuing, narrowing, escalating, or stopping.

Measurement and recovery decision

For **Separate Intelligence From Authority**, define **unauthorized execution rate** as eligible cases satisfying the rule divided by all eligible cases. In this **Separate Intelligence From Authority** worksheet, the

Chapter 1 worked baseline is 59/100 and the illustrative target is at least 75/100; both must be replaced with evidence from a delegated action crossing an authorization boundary. Review the Chapter 1 result for every grant, privileged action, policy change, and incident, preserve excluded and high-consequence cases, and interpret movement as better only when it is toward the stated target with exceptions reported separately.

The named failure is **granting authority as a reward for model performance**. Containment means: deny the action, revoke exposed authority, and preserve the decision trace. Recovery is complete only after the security owner and resource owner can reconstruct the failed case, explain which assumption changed, and authorize a new test against **unauthorized execution rate**.

Practice lab

Choose one live but reversible **Separate Intelligence From Authority** example from a delegated action crossing an authorization boundary. Create the following:

- a one-sentence answer to **Why should model capability never imply operational permission?**;
- a completed first draft of **Draw separate reasoning policy authorization execution and audit planes**;
- one routine case, one missing-evidence case, and one case shaped by **granting authority as a reward for model performance**;
- a locally measured baseline for **unauthorized execution rate** using the uniteligible cases satisfying the rule divided by all eligible cases;
- a stop threshold, the security owner and resource owner as owner, and the safe default deny the action, revoke exposed authority, and preserve the decision trace;
- security evidence packet that will be handed to the next chapter.

Ask a colleague to challenge **Draw separate reasoning policy authorization execution and audit planes**. They should identify one ambiguous field, one person among resource owners, operators, and affected subjects whose interests are missing, and one reason **unauthorized execution rate** could reward the wrong behavior. Preserve that challenge in the chapter handoff before revising.

Limits and judgment

Separate Intelligence From Authority supplies a locally testable method, not proof of causal effectiveness or a universal rule. This Chapter 1 method remains inside the book boundary: This book is not a security certification, legal opinion, penetration-test report, or guarantee

that an implementation resists every attack. Readers applying **Separate Intelligence From Authority** for resource owners, operators, and affected subjects must verify current law, contract, policy, safeguarding, accessibility, security, and professional duties before consequential use. A negative **unauthorized execution rate** result may mean the question was framed badly, the review cost is too high, or deny the action, revoke exposed authority, and preserve the decision trace is the responsible decision.

Takeaways

- A model may propose any action but an independent control plane must decide what identity may execute it.
- The chapter artifact is: draw separate reasoning policy authorization execution and audit planes.
- The primary measure is **unauthorized execution rate**, paired with review cost and failure severity.
- Watch especially for **granting authority as a reward for model performance**.
- In a delegated action crossing an authorization boundary, expand scope only after the security owner and resource owner can show the evidence and exercise the recovery path.

Reflection

What evidence would let a skeptical reviewer answer **Why should model capability never imply operational permission?** differently from the security owner and resource owner? Which case involving resource owners, operators, and affected subjects would **unauthorized execution rate** miss, and would that omission require continuing, narrowing, escalating, or deny the action, revoke exposed authority, and preserve the decision trace?

