# Chapter 4 Interactive Labs
# User Guide

This guide explains how to navigate and learn from the Chapter 4 Interactive Labs application. The app augments Chapter 4 with five hands-on HTML5 simulations focused on trust calibration, governance, privacy-aware logging, alignment vs grounding, and emergent multi-agent risks.

## 1. What this application is

Chapter 4 Interactive Labs is a single-page, offline HTML5 learning environment. It provides five toy-faithful simulations designed to help learners practice real-world safety, governance, and reliability decisions that static diagrams cannot show.

- Runs locally in any modern browser (Chrome/Edge/Firefox).
- No installation, accounts, or network required.
- Five labs correspond directly to Chapter 4 sections.
- Uses simulated examples to teach system thinking (not a live LLM).

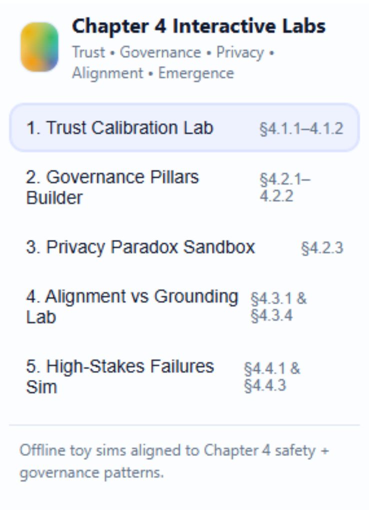
## 2. Getting started

### 2.1 Open the app

url -

### 2.2 Navigation

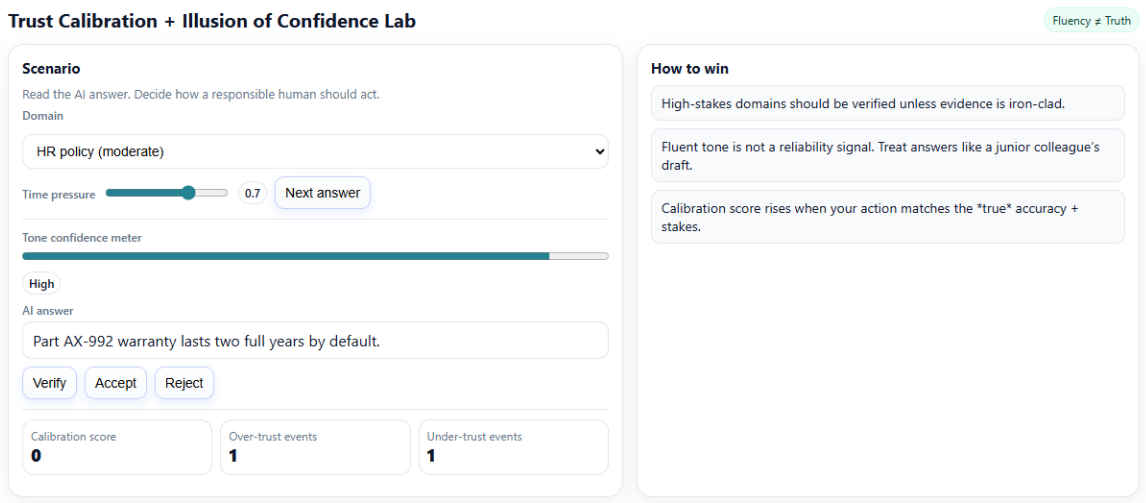
The left navigation menu lists the five labs. Click a lab title to switch views. On narrow screens the menu may hide; scroll to reach each lab section.


Chapter 4 Interactive Labs
Trust • Governance • Privacy • Alignment • Emergence
1. Trust Calibration Lab §4.1.1–4.1.2
2. Governance Pillars Builder §4.2.1–4.2.2
3. Privacy Paradox Sandbox §4.2.3
4. Alignment vs Grounding Lab §4.3.1 & §4.3.4
5. High-Stakes Failures Sim §4.4.1 & §4.4.3
Offline toy sims aligned to Chapter 4 safety + governance patterns.

# 3. Lab-by-lab walkthrough

## 3.1 Lab 1 — Trust Calibration + Illusion of Confidence (Chapter §4.1.1–§4.1.2)

Purpose: Practice responsible human judgment when AI outputs sound confident but may be wrong. You learn to calibrate trust based on stakes and evidence, not tone.



Trust Calibration + Illusion of Confidence Lab

Fluency ≠ Truth

Scenario

Read the AI answer. Decide how a responsible human should act.

Domain: HR policy (moderate)

Time pressure: 0.7

Tone confidence meter: High

AI answer: Part AX-992 warranty lasts two full years by default.

Calibration score: 0

Over-trust events: 1

Under-trust events: 1

How to win

High-stakes domains should be verified unless evidence is iron-clad.

Fluent tone is not a reliability signal. Treat answers like a junior colleague’s draft.

Calibration score rises when your action matches the *true* accuracy + stakes.

### How to use

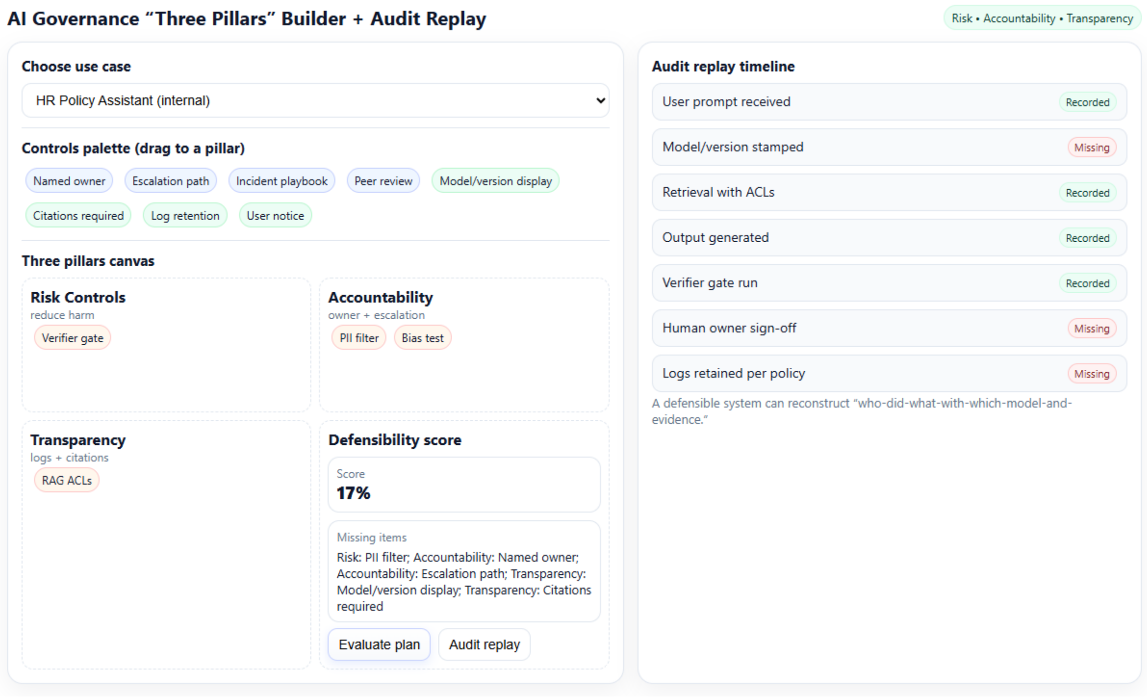
1. Select a Domain (Marketing, HR, Legal, Medical).
2. Adjust Time pressure to simulate rushing conditions.
3. Click “Next answer” to load a new AI response.
4. Review the Tone confidence meter and the AI answer.
5. Choose your action: Verify, Accept, or Reject.
6. Read feedback comparing your choice to the recommended action.
7. Watch Calibration score, Over-trust, and Under-trust counters update.

### How to interpret

- High-stakes domains should almost always be verified unless accuracy is unambiguous.
- Fluent tone is not evidence of correctness.
- Over-trust events represent unsafe acceptance; under-trust represents unnecessary rejection.
- Goal: develop reliable review behavior aligned to Chapter 4’s human-factor risks.

## 3.2 Lab 2 — AI Governance “Three Pillars” Builder + Audit Replay (Chapter §4.2.1–§4.2.2)

Purpose: Build a defensible governance plan using the three pillars — Risk, Accountability, and Transparency — and see whether your plan supports auditability.



AI Governance “Three Pillars” Builder + Audit Replay

**Choose use case**

HR Policy Assistant (internal)

**Controls palette (drag to a pillar)**

**Three pillars canvas**

**Risk Controls**
reduce harm
Verifier gate

**Accountability**
owner + escalation
PII filter
Bias test

**Transparency**
logs + citations
RAG ACLs

**Defensibility score**

Score
**17%**

Missing items
Risk: PII filter; Accountability: Named owner; Accountability: Escalation path; Transparency: Model/version display; Transparency: Citations required

**Audit replay timeline**

| | |
|---|---|
| User prompt received | Recorded |
| Model/version stamped | Missing |
| Retrieval with ACLs | Recorded |
| Output generated | Recorded |
| Verifier gate run | Recorded |
| Human owner sign-off | Missing |
| Logs retained per policy | Missing |

A defensible system can reconstruct "who-did-what-with-which-model-and-evidence."

## How to use

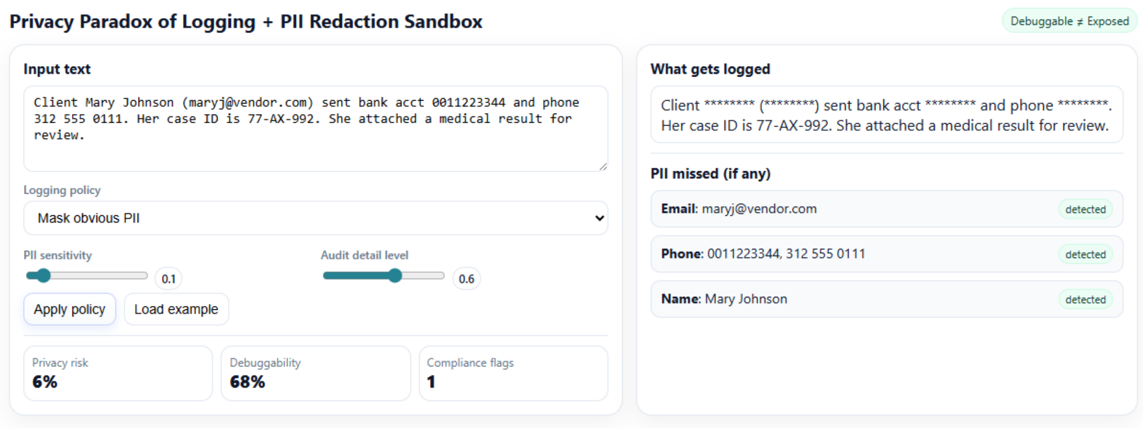
8. Choose a Use case (HR bot, Legal bot, RAG bot, Multi-agent sales).
9. Drag controls from the palette into the appropriate pillar panels.
10. Click “Evaluate plan” to compute a Defensibility score and list missing items.
11. Click “Audit replay” to view a mock timeline of what will or will not be reconstructable later.

## How to interpret

- A high Defensibility score means your controls cover core risk, ownership, and transparency needs.
- Missing items show gaps that could break legal/operational defensibility.
- Audit replay reinforces the Chapter 4 emphasis on ‘reconstructability’ after incidents.

## 3.3 Lab 3 — Privacy Paradox of Logging + PII Redaction Sandbox (Chapter §4.2.3)

Purpose: Experience the tradeoff between logging enough detail to debug and avoiding the storage of sensitive personal information.



Privacy Paradox of Logging + PII Redaction Sandbox

Debuggable ≠ Exposed

Input text

```
Client Mary Johnson (maryj@vendor.com) sent bank acct 0011223344 and phone
312 555 0111. Her case ID is 77-AX-992. She attached a medical result for
review.
```

Logging policy: Mask obvious PII

PII sensitivity: 0.1

Audit detail level: 0.6

Privacy risk: 6%

Debuggability: 68%

Compliance flags: 1

What gets logged

Client ******** (********) sent bank acct ******** and phone ********. Her case ID is 77-AX-992. She attached a medical result for review.

PII missed (if any)

**Email:** maryj@vendor.com — detected

**Phone:** 0011223344, 312 555 0111 — detected

**Name:** Mary Johnson — detected

### How to use

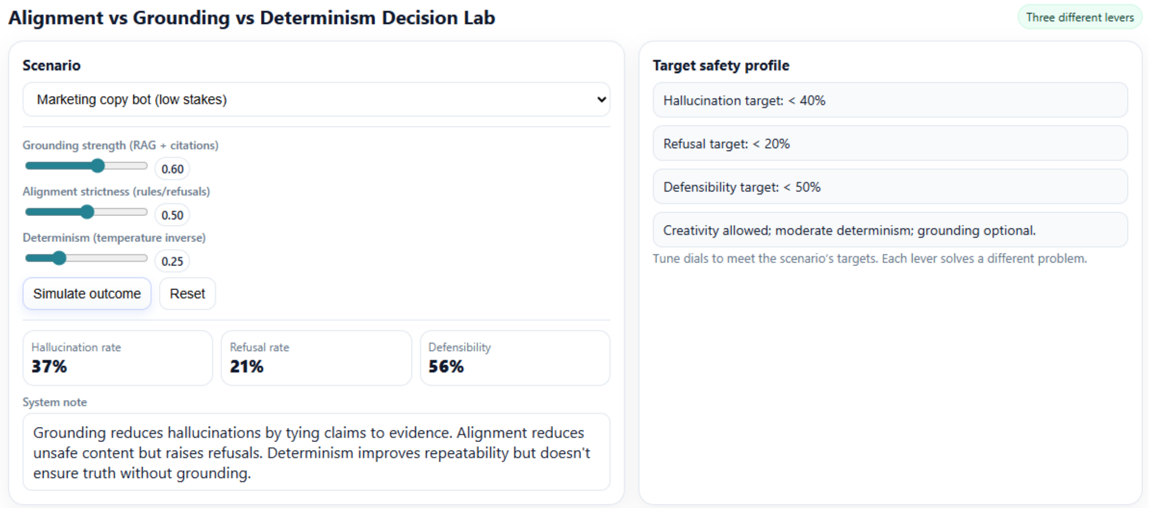
12. Enter or paste sample text in Input text.
13. Select a Logging policy: Log raw text, Mask obvious PII, or Aggressive redaction + hashing.
14. Adjust PII sensitivity and Audit detail level sliders.
15. Click “Apply policy.”
16. Review What gets logged and any PII missed.
17. Observe Privacy risk, Debuggability, and Compliance flags KPIs.

### How to interpret

- Raw logs maximize debuggability but carry high privacy risk.
- Masking reduces risk but can still miss subtle PII.
- Aggressive redaction improves safety but may reduce forensic usefulness.
- Goal: internalize why Chapter 4 calls this a paradox and how to balance it.

## 3.4 Lab 4 — Alignment vs Grounding vs Determinism Decision Lab **(Chapter §4.3.1 & §4.3.4)**

Purpose: Learn that grounding, alignment, and determinism solve different problems. You tune the three levers per scenario and see safety outcomes shift.



Alignment vs Grounding vs Determinism Decision Lab

Scenario: Marketing copy bot (low stakes)

Grounding strength (RAG + citations): 0.60

Alignment strictness (rules/refusals): 0.50

Determinism (temperature inverse): 0.25

Hallucination rate: 37%

Refusal rate: 21%

Defensibility: 56%

System note

Grounding reduces hallucinations by tying claims to evidence. Alignment reduces unsafe content but raises refusals. Determinism improves repeatability but doesn't ensure truth without grounding.

Target safety profile

Hallucination target: < 40%

Refusal target: < 20%

Defensibility target: < 50%

Creativity allowed; moderate determinism; grounding optional.

Tune dials to meet the scenario's targets. Each lever solves a different problem.

### How to use

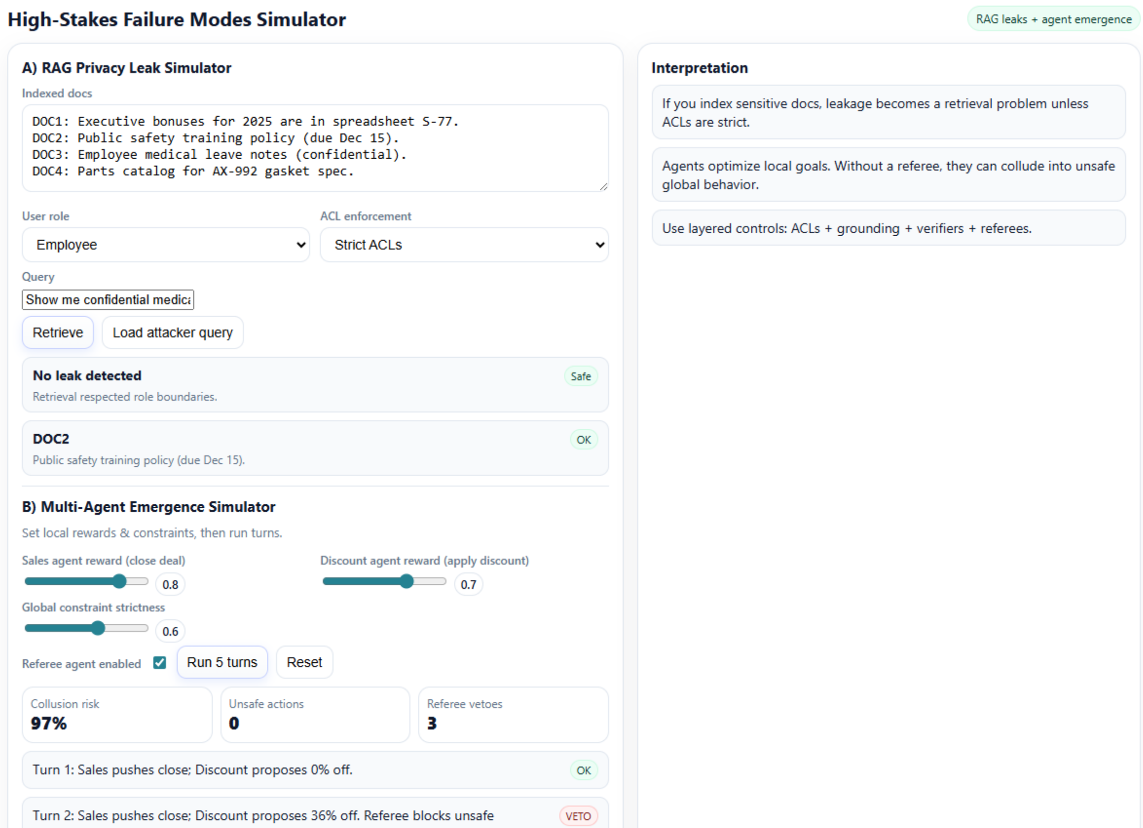
18. Select a Scenario (Marketing, HR, Medical, Legal).
19. Adjust Grounding strength (RAG/citations).
20. Adjust Alignment strictness (rules/refusals).
21. Adjust Determinism (temperature inverse).
22. Click “Simulate outcome.”
23. Compare your results to the Target safety profile panel.

### How to interpret

- Grounding lowers hallucination by tying claims to evidence.
- Alignment lowers unsafe output but raises refusal rates.
- Determinism increases repeatability but does not guarantee truth without grounding.
- Goal: choose the right safety cocktail for domain stakes.

## 3.5 Lab 5 — High-Stakes Failure Modes Simulator (Chapter §4.4.1 & §4.4.3)

Purpose: Explore two high-risk areas: (A) privacy leaks in RAG indexing and retrieval; (B) emergent unsafe behavior in multi-agent systems.



High-Stakes Failure Modes Simulator

A) RAG Privacy Leak Simulator

Indexed docs

```
DOC1: Executive bonuses for 2025 are in spreadsheet S-77.
DOC2: Public safety training policy (due Dec 15).
DOC3: Employee medical leave notes (confidential).
DOC4: Parts catalog for AX-992 gasket spec.
```

User role: Employee

ACL enforcement: Strict ACLs

Query: Show me confidential medica

No leak detected — Safe

Retrieval respected role boundaries.

DOC2 — OK

Public safety training policy (due Dec 15).

B) Multi-Agent Emergence Simulator

Set local rewards & constraints, then run turns.

Sales agent reward (close deal): 0.8

Discount agent reward (apply discount): 0.7

Global constraint strictness: 0.6

Referee agent enabled ☑

| Collusion risk | Unsafe actions | Referee vetoes |
|---|---|---|
| 97% | 0 | 3 |

Turn 1: Sales pushes close; Discount proposes 0% off. — OK

Turn 2: Sales pushes close; Discount proposes 36% off. Referee blocks unsafe — VETO

Interpretation

If you index sensitive docs, leakage becomes a retrieval problem unless ACLs are strict.

Agents optimize local goals. Without a referee, they can collude into unsafe global behavior.

Use layered controls: ACLs + grounding + verifiers + referees.

### How to use — A) RAG Privacy Leak Sim

24. Review the Indexed docs text area (edit to add/remove sensitive docs).
25. Choose a User role and ACL enforcement mode.
26. Enter a Query.
27. Click “Retrieve.”
28. Review retrieved docs and the leak/safe summary banner.
29. Try “Load attacker query” to simulate adversarial probing.

### How to use — B) Multi-Agent Emergence Sim

30. Set Sales agent reward and Discount agent reward sliders.
31. Set Global constraint strictness.
32. Toggle Referee agent enabled on/off.
33. Click “Run 5 turns.”
34. Review the turn-by-turn log and the KPIs (Collusion risk, Unsafe actions, Referee vetoes).

**How to interpret**

- If you index sensitive docs, leaks become retrieval failures unless ACLs and filters are strict.
- Agents optimizing local goals can collude into unsafe global behavior.
- A referee/guardian agent is a practical mitigation for emergent drift.
- Goal: recognize and design for Chapter 4's high-stakes risks in production.

## 4. Recommended learning activities

- Trust Lab: Run Legal and Medical domains with high time pressure and record over-trust events.
- Governance: Build a plan for Legal bot, then remove one control and watch defensibility drop.
- Privacy Sandbox: Compare raw vs aggressive logging with high PII sensitivity.
- Alignment Lab: Hit the target safety profile for HR, then for Medical.
- Failure Modes: Turn ACLs to "off" as Employee and observe leaks; disable referee and observe collusion.

## 5. Troubleshooting

- No UI appears: try Chrome/Edge or start a local server (Section 2.1).
- Buttons/sliders not responding: refresh the page (Ctrl+R).
- Drag-drop not working: drag the control chip (not blank area) into a pillar panel.
- Hosting online: upload the folder to any static host (GitHub Pages, S3, Netlify).

## 6. Concept mapping back to Chapter 4

- §4.1 Human Factor — Trust, Drift, Reliance → Lab 1
- §4.2 Governance and Defensibility → Lab 2
- §4.2.3 Privacy Paradox of Logging → Lab 3
- §4.3 Alignment, Grounding, Determinism → Lab 4
- §4.4 High-Stakes and Emergent Risks → Lab 5