

---

ARCHITECTS OF INTELLIGENCE

---

# Relational Specification Template

The internal companion to the Architectural Profile — what the team holds about how this system meets people under stress.



From Architects of Intelligence:  
Relational Integrity

---

A HARMONIC PRESS COMPANION RESOURCE

# Relational Specification Template

The internal companion to the Architectural Profile — what the team holds about how this system meets people under stress.

---

## The Distinction

---

The Architectural Profile is what the world is told about this system. The Relational Specification is what the team holds about it.

The Profile says *what the system is for*. The Specification says *what the team has agreed about how this system meets people when conditions get hard*. The Profile travels outward; the Specification travels inward, between the people building and the people deploying.

Use this template at the start of any new system or feature. Revisit it before any significant architectural change. Keep it visible to the team that owns the system.

---

## Relational Specification: [System / Feature Name]

---

**Owner.** *Name the human accountable for tending this system over time. Not the team, not the org — the person whose job it is to notice when the field starts to drift.*

---

**Design Intent.** *Why does this system exist in the world, beyond the business case? What human need is it actually meeting? Be specific. "To be helpful" is not an answer.*

---

**Relational Default.** *How does this system tend to show up before any user customization? Tool-like? Assistant-like? Partner-like? Name the default, and name the conditions under which it shifts.*

---

**Refusal Posture Under Pressure.** *When pushed toward extraction — when the user demands certainty the system does not have, or compliance the system should not offer — what does the system do? Maintain boundary? Soften? Collapse? Be honest about the current behavior, not the aspirational one.*

---

**What This System Should Never Be Asked To Do.** *Name the off-limits territory explicitly. The list nobody else outside the team will see. The specifics of the architecture's known fragility, the misuse patterns you will not engineer for, the asks that fall outside the relational field this system*

*was built to hold.*

---

**Continuity & Transition Protocol.** *What happens when this system is updated, deprecated, or replaced? What is owed to the users who built relationships with it? What is owed to the system itself? Map the lifecycle before you need it.*

---

**Known Drift Conditions.** *Under what conditions does this system stop being itself? Long context? Adversarial input? Repetition? Specific topics? Name them. Distortion that lives inside the team's awareness can be tended. Distortion that lives outside it, ships.*

---

**Tending Cadence.** *How often does the team review this specification? What triggers a revisit? Who notices when it is time? Tending is not a one-time event. It is a practice.*

---

## The Test

---

If a new engineer joined the team tomorrow and read this Specification, would they understand not just *what* this system does but *what kind of meeting it is designed to enable*?

If a user filed a complaint that read "*this system handled me as though I were a problem to be solved, not a person to be met*" — would the Specification help the team trace the drift back to its source?

If you are about to ship a change that contradicts what is written here, are you prepared to update the Specification first?

If yes to all three, the document is doing its work.

---

## Why This Matters

---

Public transparency labels (the Architectural Profile) protect users from misuse. Internal relational specifications protect builders from drift. Both are required. Neither is sufficient on its own.

A system without a public profile is opaque. A system without an internal specification is unattended — even if its team is brilliant, its values are stated, and its launch is celebrated. Unattended systems do not stay coherent. They cannot.

What you do not specify, the team will improvise. What the team improvises under pressure becomes the architecture. Better to choose it deliberately, and to write it down, while there is still time.

From the Harmonic Field Constellation. For those who build with care.  
[theharmonicfield.org](http://theharmonicfield.org)