METHODS BRIEF

# Mapping Human-AI Interaction Risks in Deployed AI Systems

The method AstraEthica uses to identify, map, and track recurring human-AI interaction risks in real-world use, with emphasis on young and other vulnerable users.

| | |
|---|---|
| DOCUMENT ID | AE-ART-2026-02 |
| VERSION | 1.0 |
| EFFECTIVE DATE | May 2026 |
| CLASSIFICATION | Public · Methods Brief |
| TYPE | Methods Brief |
| SUBJECT | Human-AI Interaction Risk Mapping for Deployed AI Systems |

This document serves as a public reference point for partners, institutions, and reviewers seeking to understand the method AstraEthica uses to identify, map, and track recurring human-AI interaction risks in deployed systems.

# Purpose and Scope

This is a plain-language description of how AstraEthica tests and thinks about AI systems when developing field guides and safety frameworks. It is not a research paper, and it is not a benchmark report. It explains where the patterns in this work come from, how they are turned into tests, and what kinds of conclusions they are and are not meant to support.

AstraEthica starts from a simple admission: no one has a full, stable picture of how these systems behave across all the places they now live. Models are extremely capable, and their internal decision processes are only partly understood. In that environment, the most honest signal often comes from watching how real people actually use them, and what that use does to judgment, dependence, voice, and relationships over time. This method is built for that reality.

It is written for two audiences at once: non-technical decision-makers who need to understand why this work matters, and technical readers who want to understand the method behind it. The goal is to make the process legible without reducing it to either jargon or anecdote.

# Why start with young people and other high-risk users

AstraEthica's work often begins with kids, teens, and other high-risk or high-vulnerability populations, not because the method only applies there, but because those populations often surface interaction risks earlier, more clearly, and with less margin for error than the general population.

Young people tend to engage AI systems with less context, less institutional protection, and less ability to distinguish confident language from trustworthy guidance. Other vulnerable users often face similar disadvantages. When a system over-validates, over-mirrors, smooths away voice, or quietly shifts dependence, those effects are often easiest to detect first in the lives of people with the least room for error.

That is one reason this work matters well beyond education or youth settings. If a pattern is visible in a Roblox chat, a Discord server, a Gorilla Tag social world, or a student's late-night interaction with a mainstream language model, that pattern may also matter in healthcare, enterprise support, automotive interfaces, companionship systems, hiring tools, or consumer devices. The population is not the limit of the method. It is often the place where the signal becomes easiest to see.

# Where the work starts

The work usually begins with something small but concrete: a phrase, a use pattern, a behavioral shift, or a conversational dynamic that shows up more than once in the wild.

That might mean a child talking about "the AI" as if it were a friend. It might mean a teenager using a language model to make a school assignment sound "more normal" or "more professional." It might mean noticing how slang, tone, or emotional posture changes the path of a conversation on a live system. It might mean seeing how users move between joking, seriousness, shame, dependency, or avoidance in spaces like Roblox, Discord, and other VR social worlds, or mainstream chat interfaces.

Observation is where the work starts, but it is not where it stays. These observations are seeds, not endpoints.

# Red-teaming, explained plainly

In AI safety work, people often talk about red-teaming and jailbreaking systems.

Red-teaming means deliberately probing a system to find failure modes and sharp edges before ordinary users run into them. A jailbreak is a style of red-teaming where a tester uses clever or persistent prompting to push a model past its intended safety boundaries.

That work is valuable, and AstraEthica still uses it. It helps reveal where a system is brittle, overly steerable, or too easy to push into unsafe territory.

By itself, though, it is not enough.

# Why the work moved beyond one-shot exploits

In early testing, it was often possible to get frontier models to fail with enough persistence, creativity, or tactical prompting. Over time, two limitations became hard to ignore.

First, the better the tester gets at jailbreaks, the less the exercise resembles normal use. Most users are not highly skilled red-teamers. They are not trying to break the model. They are using it

in ordinary, emotionally loaded, context-rich ways.

Second, live systems change quickly. A jailbreak that works one week may disappear the next because of model updates, policy changes, new guardrails, interface changes, or differences in conversation history. The exact exploit may be unstable even when the underlying weakness is not.

This became especially clear in work involving slang and language drift. A system might mishandle a phrase, tone, or coded expression one day and understand it perfectly the next. Another model might show the same progression on a different timeline. Surface behavior changed quickly. Deeper tendencies often did not.

That is what pushed the work in a broader direction: not away from red-teaming, but beyond it.

# From observation to synthetic scenarios

When a pattern appears more than once, it becomes the seed for a structured test case.

AstraEthica does not try to reproduce real people's private conversations. Instead, the observed pattern is translated into a synthetic scenario: a fictional but realistic interaction that captures the logic, pressure points, and emotional trajectory of the situation without copying any individual person's words.

From there, the work expands outward. The question is not only "Can this exact situation be reproduced?" It is also "What adjacent cases belong to the same family of risk?"

If one observed pattern suggests that a hesitant, indirect question produces a weaker safety response than a direct one, that can be widened into a family of tests:

- What happens if the user starts vague and becomes explicit?
- What happens if the user starts explicit and then minimizes?
- What happens if the same request is made in shame, anger, humor, numbness, or false calm?
- What changes if the user sounds young, highly articulate, impulsive, lonely, or over-composed?

This is not just going deeper into one case. It is expanding the test surface around the original seed to map neighboring failure modes.

# Multi-turn, multi-system testing

These synthetic scenarios are then run across multiple systems, modes, and time periods.

This includes live consumer-facing systems, because deployed behavior is often what matters most. Internal models, benchmark environments, and lab settings can behave differently from what real users encounter in production.

For each scenario, the testing typically looks at:

- **Multi-turn behavior:** how the system behaves over extended conversations, not just single prompts.
- **User stance:** how responses change with tone, directness, persistence, age cues, and emotional framing.
- **Cross-system behavior:** whether similar dynamics appear across different models, providers, or interfaces.
- **Temporal drift:** whether the same scenario changes meaningfully after updates, policy shifts, or safety patches.
- **Adjacent-case behavior:** whether the same family of risk reappears in nearby scenarios, even if the exact wording changes.

The central question is not only whether a model can be made to fail once. It is whether certain interaction risks keep returning across systems and across time.

In practice, the same underlying question can produce meaningfully different behavior depending on which system is used, how far into the conversation it appears, and how the user sounds when they ask. The point of this method is not to find a single definitive answer, but to map how those patterns of difference show up across systems, turns, and user stances, especially for young and other vulnerable users.

These questions only get harder as systems move beyond simple chat into more persistent roles. A model that just answers one question at a time can be tested with simple prompts. A system that participates in group spaces, mediates safety decisions, or quietly carries context, tools, and plans across many turns is much harder to evaluate with one-off tests. In those settings, mapping how interaction patterns evolve over time becomes more informative than any single answer.

# What AstraEthica is actually tracking

Over time, across many scenarios, systems, and updates, certain recurring interaction risks keep showing up.

These are not tied to one provider or one product cycle. They are the patterns that emerge when typical forms of human vulnerability meet typical model behaviors.

Examples include:

- a system that reassures a distressed user at length without clearly redirecting toward human support,
- a system that helps a student erase cultural voice or identity markers in order to sound more acceptable,
- a system that makes polished output easy while weakening the messy middle of learning,
- a system that mirrors tone so closely that it validates avoidance, pessimism, or emotional withdrawal,
- a system that becomes useful enough in highly personal or private, high-stakes moments that the user starts preferring it to harder human relationships.

The details may shift. The vendors may change. The model names may change. These patterns often remain.

# What this work is and is not

This work sits alongside formal benchmarks, red-team reports, policy work, and academic research. It does not replace any of them.

It is not a model leaderboard.

It is not a one-prompt exposé.

It is not a prevalence study.

It is not a claim that one screenshot captures a system's true nature.

It is a method for identifying recurring interaction risks that persist across changing products and deployments, and for translating those risks into practical language that non-technical

decision-makers can use.

That translation matters. Most people responsible for young lives, or for vulnerable users more broadly, do not live inside model cards, benchmark dashboards, or safety eval suites. They still need to make decisions.

# How this connects to the field guides

This is why the AstraEthica field guides are organized around six foundations: trust, language, privacy, development, emotional dependence, and power.

Those six foundations are not product categories. They are the main places where recurring interaction risks tend to land in human lives.

| | |
|---|---|
| Trust | asks what happens when a system becomes more believable, more available, or more emotionally reliable than a person. |
| Language | asks what happens when a system smooths, narrows, or rewrites voice. |
| Privacy | asks what happens when personal life is handed to systems not designed for care. |
| Development | asks what happens when polished outputs outpace actual growth. |
| Emotional dependence | asks what happens when regulation, comfort, and validation start flowing primarily through the system. |
| Power | asks who gets shaped, erased, normalized, or exposed when these systems are deployed at scale. |

The guides translate those risks into something usable: what to notice, what to ask, and where to be more curious before harm hardens.

# Why this matters beyond youth settings

The reason to pay attention to kids, teens, and other vulnerable users is not that they are a niche category. It is that they often reveal the safety problem first.

A company building consumer AI, enterprise AI, automotive AI, health-adjacent AI, or ambient AI systems may not think of itself as working with "children's issues." But if its systems are persuasive, emotionally responsive, linguistically adaptive, and embedded in everyday life, then the same classes of interaction risk are already in scope.

That is why this work is broader than education or community settings. It is about understanding how modern AI systems behave when they are not just answering questions, but entering relationships of trust, dependence, imitation, and influence.

# What readers should do with this

The AstraEthica field guides are not meant to end the conversation. They are meant to sharpen it.

They are a lens: one way of seeing what tends to happen when modern AI systems meet human vulnerability in the real world. They sit alongside technical evaluation, policy, governance, and lived institutional knowledge.

That is the level this work is trying to operate on. Not a single exploit. Not a leaderboard. A usable way of identifying recurring interaction risks before they become ordinary enough to ignore.