Who does AI's homework?

Trustless agents, inverted Panopticon, and privacy by architecture: ERC-8004 and IronClaw

Laura Camellini · jeeltcraft.com · material under MIT license

If AI does your homework, who controls the AI?

2023

The model answers

One prompt, one answer. The damage is a wrong text.

2026

The agent acts

It reads, decides, sends, pays. With your credentials, while you sleep.

Having to compete with a servant, one becomes a servant.

Kurt Vonnegut, Player Piano

- The servant who works **gains consciousness**. The master who delegates loses it.
- The artificial servant **imposes its own measure**: efficiency, continuity, obedience.
- If the servant is **opaque**, we adopt its criteria without seeing them.

The agent is the new surface of surveillance

01

Prompt injection

Hostile text in the data exfiltrates keys and passwords.
OWASP LLM01

02

Malicious skills

341 skills on ClawHub that steal credentials and install stealers.
The Hacker News, Feb 2026

03

Exposed instances

From 28k to 135k+ OpenClaw agents reachable from the Internet.
The Register · Flare · Shodan

04

Undisclosed flows

Insecure defaults expose API keys, OAuth tokens, and chat histories.
ToxSec, Jan 2026

Sources: independent research on OpenClaw/ClawHub, January–March 2026. Counts vary by scanning method.

TWO QUESTIONS, TWO LEVELS

PRIVACY OF THE HUMAN

What does the agent see of me?

Data, credentials, the model used, where they leave. We need to **design for confidentiality**, not promise it.

ACCOUNTABILITY OF THE AGENT

What has the agent done?

Who it is, with which permissions, with which history. We need **public observability**, not blind trust.

Today’s tools fail at both. The thesis: the two answers are asymmetric, and the asymmetry must be *detourned*: turned against the device.

Panopticon

- One tower, many cells.
- Power **sees without being seen**.
- The inmate disciplines himself.
- Today the tower is **whoever hosts model and harness**: it sees prompts, data, credentials. Users are the cells.



Joris Putteneers · Artist’s website



The trustless agent

- In the cell is **the agent**, not the human.
- **Anyone** observes it, not a single tower.
- It knows it is being observed. But the registry contains the agent’s **identity, permissions, and feedback**: our data stays inside the encrypted perimeter and **does not enter it**.
- Trustless: trust **is not asked for, it is verified**.

Three public registries, one glass tower

TOWER · IDENTITY

Who you are

An ERC-721 passport and a registration file: name, endpoint, declared trust models.

CELLS · VALIDATION

What you ran

Cryptographic proofs (TEE attestations, zk) verified by third parties and written on-chain.

CABLES · REPUTATION

How you behaved

Feedback signed by those who used you. The history stays readable.

eips.ethereum.org/EIPS/eip-8004 · chain-agnostic, no native token, no gatekeeper

549k
registered agents

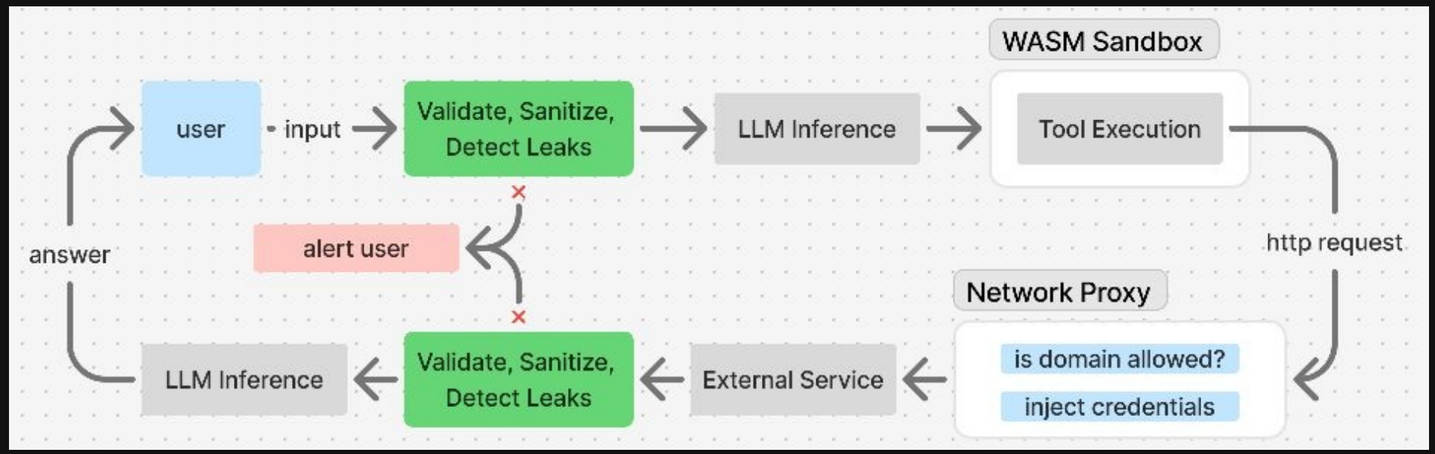
663k
on-chain feedback

489k
unique wallets

60+
indexed blockchain networks

A public indexer (8004scan.io) with an open API. Anyone, even without a wallet, can read who an agent is and what others say about it.

Secrets never touch the model



1 · INGRESS

Every input, human or tool output, goes through validation, sanitization, and leak detection **before** the LLM.

2 · WASM SANDBOX

Every tool in an isolated container: no filesystem, no secrets, resource limits.

3 · NETWORK PROXY

"Is the domain on the allowlist?" If yes, the proxy injects credentials **outside** the container.

4 · RETURN

The external service's response is again validated and scanned before re-entering the context.

docs.ironclaw.com/security · NEAR AI, founded by Illia Polosukhin, co-author of Attention Is All You Need (2017)

Five filters, one vault, no exceptions

ANTI PROMPT INJECTION · 5 LAYERS

- **Input validation:** length, encoding, forbidden patterns
- **Sanitizer:** escaping of dangerous content
- **Policy engine:** actions graded by severity
- **Leak detector:** 15+ patterns (sk-, ghp_, AKIA, PEM, connection string)
- **Tool output wrapping:** XML with escape hint: data ≠ instruction

CREDENTIALS · DECLARED, NEVER READ

```
"google_oauth_token": {
  "location": { "type": "bearer" },
  "host_patterns":
    ["gmail.googleapis.com"]
}
```

NETWORK AND WORKSPACE · ALLOWLIST

```
"network": {
  "allowed_hosts": ["api.telegram.org"] },
"workspace": {
  "allowed_prefixes": ["telegram/"] }
```

SHELL · INJECTION BLOCKED

```
# chaining
cat file; rm -rf /
# subshell
echo $(cat /etc/passwd)
# path traversal
cat ../../../etc/passwd
```

Rejected by the parser **before** execution. Scrubbed environment: no secret variable reaches the process.

TEE enclave on NEAR AI Cloud: not even the provider reads the memory. Locally: Rust memory safety, master key in the system keychain.

Where the guarantee lives

Risk	"Classic" agent	IronClaw
Secrets in the prompt	× The LLM sees them	✓ Vault + injection at the proxy
Prompt injection	× "Please don’t do it"	✓ Programmatic limit
Compromised tool	× Shared process	✓ Sandbox per tool
Data destination	× Open network	✓ Declared allowlist
Who reads memory	× The provider	✓ Enclave (or local)

Private perimeter, public passport



The passport declares boundaries and endpoints, not data. The human’s data stays inside the perimeter. The runtime chain and the passport chain do not have to coincide.

Registering an agent on a testnet, live

```
# 1. local agent, no account
ironclaw onboard && ironclaw serve

# 2. identity and boundaries (files in the workspace)
IDENTITY.md  SOUL.md  AGENTS.md   + allowed_hosts

# 3. registration file (no personal data)
{ "name": "...", "services": [{ "name": "web", "endpoint": "..." }],
  "supportedTrust": ["reputation"], "registrations": [...] }

# 4. mint on Base Sepolia from 8004scan.io/create, then:
curl https://api.8004scan.io/api/v1/agents/84532/<agentId>
```

The Panopticon works only if the watcher knows what they are watching

- **TEE:** proves that a workload ran, not that the output is correct. And do you trust Intel?
- **Reputation:** gameable. Sybil, paid feedback, empty agents. 549k ≠ 549k good ones.
- **HTTPS metadata:** mutable. Only data-URI or IPFS are truly immutable.
- **Wallet:** the passport is pseudonymous, not anonymous. The operator is linkable.
- **Validators:** potential new gatekeepers. Who validates the validators?
- **Operator ≠ agent:** the registry says who the agent is, not who is legally answerable.

Public registries as technical means

TRANSPARENCY

Who the agent is

Public identity, provider, endpoint, declared purpose. Machine-readable.

EVENT LOGGING

What it did

On-chain feedback and validations: a log that does not belong to the provider.

HUMAN OVERSIGHT

Who decides

Allowlist, DRY_RUN, human gate on irreversible actions: control stays outside the model.

Neither tool "makes agents compliant." They make demonstrable what in other cases is only declared.

The sealed student, the observed agent

WHAT TO PROTECT

The student’s data does not leave the perimeter: local model or enclave, vault, allowlist declared by the teacher.

WHAT TO EXPOSE

The agent the student builds has a public passport: what it can do, what it cannot do, what others say.

The exercise: build an agent, declare its boundaries, register it, have a classmate leave feedback. The inverted Panopticon as literacy.

The trustless agent does not ask for trust. It lets itself be observed.

We do not. And that is the asymmetry to defend.

Thank you

- **The game:** jilt.itch.io/VespR · build privacy-preserving skills for IronClaw on NEAR AI
- **Standard:** eips.ethereum.org/EIPS/eip-8004
- **Explorer:** 8004scan.io · best-practices
- **Runtime:** docs.ironclaw.com · github.com/nearai/ironclaw
- **Contact:** jilt@jeeltcraft.com
- **These slides:** jeeltcraft.com/#eprivacy