openclaw-trace

Recursive Self-Improving Agents

W&B Hack #2 | Jan 31-Feb 1, 2026

The Problem

Agents fail. They hit the same errors repeatedly.
Nobody is watching.

- Cron timeouts that cascade for hours
- State corruption that silently compounds
- User frustration that goes unnoticed

What if agents could fix themselves?

The Pipeline

```
Session Traces --> mine-signals --> rollup --> tickets
                                                  |
                    <-- experiments <-- briefs <---+
```

1. **Mine** signals from real agent work (errors, frustration, delight)
2. **Roll up** into clusters with fingerprints + severity scoring
3. **Route** to tickets with PII-scrubbed context
4. **Research** via actor-critic briefs (Claude + Codex)
5. **Fix** and measure the delta
6. **Repeat**

Evidence-First Grounding

Every claim links to concrete data (from rollups):

- **Fingerprint**: stable ID for dedup (`fp1:e9bb39f9...`)
- **Severity/tier**: rubric-based ranking (score 5.99 = tier 1)
- **Counts**: items + sessions per fingerprint (e.g., 1 item / 1 session)
- **Sample refs**: session_id + span (kept in rollup, not shown here)

No hallucinated metrics. No guessed root causes.

If we can't cite it, we don't claim it.

Actor-Critic Research Briefs

Actor (Claude Code): Drafts evidence snapshot first, then RCA + options

Critic (Codex): Challenges each section:

- "Where's the dashboard link?"
- "What's the baseline error rate?"
- "Is this falsifiable?"

Forces rigor. Catches hand-waving.

```
Evidence --> RCA --> Options --> Recommendation --> Plan
```

Tickets: Grounded, Not Guessed

Each ticket includes:

Field	Source
Fingerprint ID	Hash of normalized signal
Severity tier	Scoring rubric (incidents + kind weight)
Evidence	PII-scrubbed rollup (no raw traces)
Session count	Actual observed frequency
Repro steps	Extracted from signal data

Tickets update via fingerprint matching--no duplicates.

Demo: Top Rollup Issues

Tier 1 (High Severity)

Issue	Score	Sessions	Fingerprint
Cron Gateway API timeout	5.99	1/120	`fp1:e9bb39...`
Cron (error) repeated	5.10	1/120	`fp1:3d13a0...`

[screenshot: rollup.md output]

Snapshot shows two tier-1 issues from the partial run.

Briefs are being generated.

Demo: Research Brief Output

T154: Cron Gateway API Timeout (draft)

- **Root cause:** Unknown (needs RCA)
- **Evidence:** Timeout errors surfaced in rollup snapshot
- **Recommendation:** Generate brief + run experiment
- **Stop condition:** Define after baseline metrics

[screenshot: research-briefs/T154/.../oracle-brief-v1.md]

Actionable. Falsifiable. Owned.

What Makes This Different

Typical Approach	openclaw-trace
Manual log review	Automated signal mining
Anecdotal tickets	Fingerprinted, deduplicated
Guessed root causes	Evidence-first RCA
Single-pass fixes	Actor-critic validation
No measurement	Before/after metrics

The loop closes. Agents get better.

Live Status

Built this weekend:

- [x] Signal miner (LLM + heuristic modes)
- [x] Rollup with fingerprints + severity tiers
- [x] Ticket IR export (system-agnostic)
- [x] Actor-critic research briefs (Claude + Codex)
- [x] Evidence-first ordering

Running now (snapshot):

- 120-session run in progress
- Snapshot rollup: 2 tier-1 issues surfaced
- Briefs: in progress

Next Steps

Short-term:

- Wire to Phorge/Linear ticket creation
- Add circuit breaker for experiment validation
- Cross-session fingerprint matching

Long-term vision:

- Shared PII-scrubbed rollups across deployments
- Marketplace of verified fixes
- Agents that ship their own PRs

The recursive loop: agents improving agents.

Links

- Repo: `github.com/[org]/openclaw-trace`
- Architecture: `docs/architecture.md`
- Sample briefs: `docs/research-briefs/T154-*/`, `T155-*/`

Questions?