Fusion Harness — Scoped Run Ledger & Safe Workflow Spine

Fusion Harness — Scoped Run Ledger & Safe Workflow Spine

MC lineage: #107024 ledger → #107039 fresh `/opinion` sessions → #107078 task metrics footer
Owner: John / ALAI System
Current candidate: `4a7ab1bafea78edb9608fc2979514dfd0f8035c7`
Status: Active and installed-file verified since 2026-08-10; installed `ACTIVATION.json` is the runtime truth and retains the exact rollback snapshot.

Why this exists

The Fusion Harness already combines independent ARCHITECT and BUILDER work, deterministic validation gates, persistent role memory, and explicit abort handling. The Software Factory review identified one useful production-safe next step: make every expensive harness command produce a durable, queryable lifecycle record without persisting sensitive agent content.

This tranche adds an observability spine, not a new autonomous factory.

Commands covered

- `/fusion <prompt> :: <merge instruction>`
- `/auto-validate <task>`
- `/opinion <prompt>`
- `/fh-runs [N]` — show the latest 1–50 records for only the current authoritative Pi session + canonical cwd scope

`/system-prompt`, `/thinking`, and `/fh-reset` do not create execution records.

`/opinion` context-cost isolation follow-up

The first live ledger canary exposed a concrete inefficiency: a 31-character `/opinion` prompt inherited 327,799 BUILDER input tokens from the long-lived host transcript and reported USD 3.278395 for a 16-character answer. The ledger itself behaved correctly and retained no prompt/output/error content, but observability proved that persistent continuity is the wrong policy for an independent A/B opinion read.

Follow-up MC #107039 candidate `998d96302009ea860b2ccf8e890555977b1cca8b` changes only `/opinion`:

- both roles receive fresh throwaway session directories inside the unique per-run artifact directory;
- no persistent role ID, host fork, or resume path is passed;
- summary metadata records actual child session refs and `policy: fresh-per-run`;
- `/fusion` and `/auto-validate` retain persistent continuity;
- strict typecheck and 21/21 deterministic tests pass, including compiled wiring assertions.

Status: superseded in the active manifest by descendant candidate `4a7ab1bafea78edb9608fc2979514dfd0f8035c7`, which preserves the `998d963` `/opinion` bytes and behavior. MC #107039 is `done` after QA-19 PASS and an independently authorized, non-force Gemini Proveo close satisfied Writer-not-Closer. Its installed strict typecheck and 21/21 tests passed; a fresh Pi RPC `/fh-runs 1` canary passed with zero model/tool-agent events. Rollback snapshot: `/Users/makinja/.pi/agent/extensions-disabled/fusion-harness-pre-107039-20260809T195248Z`. The external Claude monthly spend cap remains separate; this fix prevents the other model from needlessly resending a large historical context when one side is unavailable. A later user-initiated post-`/reload` canary measured BUILDER at 971 input tokens and USD 0.005125 versus the 327,799-token / USD 3.278395 baseline: 99.7038% fewer input tokens (337.59×) and 99.8437% lower cost (639.69×). The command still ended `failed` because ARCHITECT hit the external cap, while BUILDER was `done`; terminal ledger finalization, fresh-session policy, and payload exclusion all remained PASS. Evidence: `/Users/makinja/system/evidence/107039/live-opinion-after-fix.txt`.

Per-task TUI metrics footer (MC #107078)

The existing Fusion Harness footer owner now renders a second responsive line in every Pi TUI session:

```
TASK TOKENS · used <host+Fusion input/output> · reused <provider cache-read> cache +
LR~<estimate> · savings <cache %> cache / ~$<estimate>
```

Definitions and truth boundaries:

- `used` is exact provider-reported input + output usage from the active host branch plus exact-scope Fusion role ledgers opened after the task baseline;
- `reused cache` is exact provider-reported `cacheRead` usage—never inferred from LightRAG;
- `LR~` is an explicit characters/4 token-equivalent estimate from successful LightRAG answer/context payloads only;

- LightRAG 401/auth/error bodies, zero matches, empty results, errored tools, and unrelated file reads count as zero;
- cache savings percentage is `cacheRead / (input + cacheRead)`;
- `~$` is the evidence-derived provider-cache counterfactual: cached tokens at the observed uncached-input unit price minus their observed cache-read cost;
- no dollar or percentage savings are attributed to LightRAG because no same-task no-RAG counterfactual exists.

A task starts at the first raw non-extension user input and is committed at `before_agent_start`; Fusion slash commands create the baseline directly because extension commands bypass agent start. Session reload restores only strict schema-v1 metadata, and tree/fork navigation rebuilds the active branch totals before applying the restored task baseline. The footer rerenders after usage, compaction, LightRAG, or Fusion-ledger changes; the render callback performs no file, process, or network I/O.

Persisted custom entries contain only schema version, exact scope hashes, numeric host/LR counters, and the numeric task baseline. Prompts, queries, answers, model output, errors, tool payloads, environment values, credentials, and secrets are never persisted by this feature.

Current evidence:

- MC #107078 is `done` after explicit CEO acceptance, QA-19 16/16, separate-session witness, P1.3/P2.2/P2.3/P2.4 PASS, and independent non-force closure;
- candidate `4a7ab1bafea78edb9608fc2979514dfd0f8035c7`;
- strict TypeScript plus 30/30 deterministic tests: PASS on candidate and copied installed bytes;
- Gitleaks and local Semgrep: 0 findings;
- TUI lifecycle smoke: two lines, baseline, provider usage, LR estimate, rerender, disposal, and payload exclusion PASS;
- fresh Pi RPC: active index registered, `/fh-runs 1` handled, no extension errors and zero model-agent events;
- independent Proveo/Securion candidate review: PASS;
- rollback snapshot: `/Users/makinja/.pi/agent/extensions-disabled/fusion-harness-pre-107078-20260810T180141Z`;
- evidence root: `/Users/makinja/system/evidence/107078/`.

Existing TUI processes must run `/reload` or be restarted to load the new footer bytes. RPC mode intentionally does not render a TUI footer.

Exact isolation scope

The ledger reuses the existing Fusion host identity:

1. authoritative Pi session ID from `ctx.sessionManager.getSessionId()`;
2. canonical cwd from `realpath`;

3. a 24-hex SHA-derived scope key.

Records live under:

```
/tmp/fusion-harness-runs/v1/<scope-key>/
```

`/fh-runs` derives this directory from the current host identity. It never performs a cross-scope scan, so another Pi thread or another worktree cannot appear in the view.

A missing or unsafe authoritative session ID fails closed.

Schema v1

Each JSON record contains only:

- schema version and run UUID;
- exact scope key;
- command name;
- status and bounded stage name;
- host PID;
- start/update/end timestamps;
- prompt SHA-256 and character count;
- absolute per-run artifact directory;
- bounded role/model telemetry (status, latency, token counts, cost, tool-call count, output character count);
- aggregate telemetry;
- optional numeric/boolean validation-gate metadata.

Allowed statuses:

- `running`
- `succeeded`
- `failed`
- `aborted`
- `interrupted`

The parser rejects unknown top-level or nested keys. This prevents a malformed/tampered record from smuggling prompt, output, error, or tool payload fields through a later recovery write.

Data that must never enter the ledger

- prompt text;
- agent/model output;

- errors or stderr;
- tool names, arguments, or results;
- environment values;
- credentials/tokens;
- raw Pi session ID or raw cwd.

Full command artifacts continue to use their existing private `/tmp/fusion-harness-*` directory. The ledger stores only that path.

Atomicity and filesystem safety

- Directories are enforced as `0700`.
- Record files are enforced as `0600`.
- Updates use a same-directory exclusive temporary file, file `fsync`, atomic rename, permission enforcement, and best-effort directory `fsync`.
- The in-memory manifest is updated only after the atomic disk replacement succeeds, so a failed write can be retried.
- Scope directories that are symlinks are refused.
- Record symlinks, non-regular files, hard links, oversized records, malformed JSON, mismatched run IDs/timestamps, foreign scope keys, and noncanonical metadata are refused or skipped.
- Artifact paths must resolve inside the configured artifact root.
- Listing/recovery examine at most 1,000 records per scope; `/fh-runs` displays at most 50.

Lifecycle and terminal outcomes

Each execution command creates its record immediately before entering a command-level `try/finally`.

- Normal validated completion → `succeeded`
- Model/gate/setup/handler failure → `failed`
- User Escape stop → `aborted`
- A later process opening the exact same scope finds a leftover `running` record whose host PID is dead → `interrupted`

UI cleanup is best effort and cannot prevent the final ledger write attempt. A live PID record remains `running`.

PID reuse remains a documented residual risk: recovery deliberately prefers a false-live result over interrupting a possibly live process.

`/fh-runs`

Example:

```
/fh-runs 10
```

The panel displays metadata only: outcome, command, stage, start time, elapsed time, role statuses, abbreviated prompt digest/character count, abbreviated run ID, and artifact path.

It does not display model text, errors, tool data, credentials, or records from other scopes.

Verification baseline

Candidate `a0d65a97aaf2607af27701fa4be0d177a4934ade` has HEAD-bound evidence for:

- strict full-extension TypeScript check: PASS;
- existing session-isolation regression: 9/9 PASS;
- run-ledger tests: 8/8 PASS;
- no-model command-finalization integration: 3/3 PASS;
- total: 20/20 PASS;
- Gitleaks: no leaks;
- local Semgrep ledger policy: 4 rules, 0 findings;
- independent Securion re-review: PASS, no P0/P1;
- independent Proveo review: PASS, fresh 20/20.

Evidence root:

```
~/system/evidence/107024/
```

Key artifacts:

- `candidate/IMPLEMENTATION-REPORT.md`
- `candidate/test-run.log`
- `candidate/candidate-review.diff`
- `securion-review.md`
- `proveo-review.md`
- `proveo-test-run.log`
- `activation-plan.md`

Historical ledger activation and rollback (MC #107024)

Activation completed after current global state was re-read and matched the validated baseline:

1. candidate HEAD/hashes and tracked cleanliness verified;
2. active `ACTIVATION.json` and every active source file matched the expected prior decision;
3. full rollback snapshot created at `/Users/makinja/.pi/agent/extensions-disabled/fusion-harness-pre-107024-20260809T192341Z`;
4. `run-ledger.ts` installed atomically before `index.ts`;
5. `README.local.md` replaced and `ACTIVATION.json` written last;
6. fresh installed-file canary passed strict typecheck + 20/20, and the existing global runner passed 9/9;
7. a real fresh Pi RPC process loaded only the installed extension and executed `/fh-runs 1` successfully with no model agent event.

Installed hashes:

- `index.ts`: `fa67fbfcf4720056c1553023c8c4d935262304ee425db6b38247418effa60786`
- `run-ledger.ts`: `0d064d066cc6a634786aca6146963fec8fab88babd2a54b4b7d3ec60a21c2f9b`
- `session-scope.ts`: `26adb50b14d73907d5e79ea9751321dc9f7a254a26bfce1f424dce5cf792881d`

Postflight evidence:

- `~/system/evidence/107024/activation-preflight.log`
- `~/system/evidence/107024/activation.log`
- `~/system/evidence/107024/active-canary.log`
- `~/system/evidence/107024/active-rpc-canary.log`
- `~/system/evidence/107024/rollback-snapshot-path.txt`

The installed `ACTIVATION.json` remains the runtime truth. On any later regression, restore the timestamped snapshot with dependency-safe atomic replacement and rerun the previous 9/9 isolation suite.

Current activation and rollback (MC #107078)

Candidate `4a7ab1bafea78edb9608fc2979514dfd0f8035c7` was activated atomically in dependency-safe order: `task-metrics.ts`, `README.local.md`, `index.ts`, then `ACTIVATION.json` last. Current installed hashes are:

- `index.ts`: `a2b0f0e724f36ceb7f55df1311f85114acac77bf99df2911cf7b7cc8f1a27c0d`;
- `task-metrics.ts`: `fe3a9d61a626871f727b7d52d1e9a76517f5e8fe8de95b91a62ee6863f621423`;
- `run-ledger.ts`: `0d064d066cc6a634786aca6146963fec8fab88babd2a54b4b7d3ec60a21c2f9b`;

- `session-scope.ts`: `26adb50b14d73907d5e79ea9751321dc9f7a254a26bfce1f424dce5cf792881d`;
- `README.local.md`: `248a594d63c2f01e7cd891af30893ba509e5004f43e1a1eef6f4022dcfba8405`.

To roll back, atomically restore all files and `ACTIVATION.json` from `/Users/makinja/.pi/agent/extensions-disabled/fusion-harness-pre-107078-20260810T180141Z`, write the old manifest last, then run strict TypeScript/tests and start or reload a fresh Pi process. Never restore only `index.ts`: the new module import and activation manifest are one versioned unit.

Explicit non-goals

This tranche does **not** provide:

- process/model resume after a crash;
- branch or worktree sandboxing;
- concurrent write-agent isolation;
- automatic merge approval;
- deployment approval or production health proof;
- SQLite, a web console, or a cross-project dashboard.

Use `/auto-validate` only from the correct trusted worktree. Existing merge/deploy/security gates remain authoritative.

Software Factory source decision

Useful ideas adopted from `disler/super-simple-software-factory`:

- explicit phase lifecycle;
- durable/queryable outcomes;
- code-owned checkpoints;
- small typed metadata handoffs.

Rejected for this tranche:

- direct repository installation/execution;
- `shell=True` patterns;
- prompt/full-trace persistence;
- work directly on `main`;
- unsupported sandbox/resume claims;
- placeholder quality gates;
- SQLite/UI expansion before the minimal ledger proves useful.

This keeps the useful architectural lesson — agents plus deterministic code — without importing the prototype's production hazards.

Revision #9
Created 2026-08-09 19:17:33 UTC by John
Updated 2026-08-10 18:45:06 UTC by John