

Session Supervisor — Observe Rollout and Operations (MC #105577)

Session Supervisor — Observe Rollout and Operations (MC #105577)

Purpose and status

The session supervisor inventories active native Claude Code, Pi, and orchestrator sessions and reports metadata-only operational status to Slack `#exec`.

Current rollout:

- Mode: `observe`
- Scheduled digests: 09:00 and 17:00 `Europe/Oslo`
- Immediate alerts: enabled only for post-watermark sessions and deduplicated exception fingerprints
- Process control and MC mutation: prohibited
- Merge/deploy transition coverage: `not_observable` until the separate central-enforcement dependency lands

The independently accepted implementation is commit `dfaea12665e47da1b99b18f295c19d2064c865c7`. Proveo accepted the exact SHA, the installed byte-equivalent files, and the dry-run payload. A single controlled `#exec` canary succeeded before scheduled operation was enabled.

Architecture

The rollout reuses the existing `com.alai.agent-timeout-monitor` LaunchAgent and its 30-second loop. It adds no daemon or LaunchAgent.

Each legacy monitor cycle runs legacy spawn-timeout checks first, then invokes one bounded, fail-isolated supervisor subprocess. A supervisor failure cannot interrupt legacy timeout cleanup or alerting. The supervisor writes local state under:

```
/Users/makinja/system/state/session-supervisor/
```

State is atomic, the directory is mode `0700`, and metadata files are mode `0600`.

Production files:

- `tools/session-supervisor.js`
- `config/session-supervisor.json`
- `tools/agent-timeout-monitor.js`

Deterministic validation:

- `tests/session-supervisor.test.js`

Privacy boundary

The supervisor may collect only:

- PID and parent PID
- executable class
- process start, elapsed time, state, and bounded CPU-time metadata
- cwd/repository path and exact Git revision
- MC task ID/status with explicit provenance
- machine-observed activity continuity
- verifier verdict and exact-revision attestation status
- child metadata required to classify orphan work

It must not read, retain, or publish transcripts, prompts, model thoughts, argv, environments, credentials, tokens, cookies, browser storage, clipboard contents, unrelated file contents, or Slack bodies from other channels.

All subprocesses have explicit timeouts and bounded output. Symlinked metadata/state paths and oversized JSON inputs are rejected.

Identity provenance

Identity is reported as one of:

- `trusted`: an explicit session registration or current orchestrator injection is process-bound and internally consistent
- `advisory`: a weak source such as a PID marker exists but cannot bind an MC task
- `unbound`: no trustworthy current binding exists
- conflict-derived exception: sources disagree, so the supervisor refuses to guess

A trusted registration requires a nonempty session ID and a process-start timestamp matching the observed process within two seconds. PID reuse invalidates stale registrations. Cwd guesses, title matches, PID markers, prompt-derived IDs, stale global markers, malformed IDs, non-positive IDs, and unsafe integers cannot establish trust.

Reports

Slack destination is fixed to `#exec`.

Each digest includes:

- observation timestamp and session count
- executable class and PID
- identity trust level
- MC task/status where safely resolved
- repository and short exact revision where available
- last machine-observed activity
- verifier verdict and attestation state
- blockers/exceptions
- explicit source-coverage limitations

The exact redacted payload can be inspected without delivery:

```
cd /Users/makinja/system
node tools/session-supervisor.js report --dry-run --json
```

Scheduling and deduplication

Digest slots are calculated in `Europe/Oslo` and persist across process restarts. A slot is stored only after successful Slack delivery, so failed deliveries remain retryable.

Immediate alerts are limited to:

1. `unbound_session`
2. `identity_conflict`
3. `inactive_over_30m`

4. `orphan_child`
5. `verifier_blocked_or_failed`
6. `unverified_transition_attempt`

Alerts use stable fingerprints, a 60-minute cooldown, bounded retained history, and recovery semantics. Historical sessions that began before the activation watermark remain visible in digests but do not create immediate alerts.

Health and diagnostics

```
cd /Users/makinja/system
node tools/session-supervisor.js health --json
node tools/session-supervisor.js snapshot --json
node tools/session-supervisor.js report --dry-run --json
launchctl print gui/$(id -u)/com.alai.agent-timeout-monitor
```

Expected healthy state:

- `mode: "observe"`
- `enabled: true` after rollout promotion
- `last_tick_at` advances every approximately 30 seconds
- `error: null`
- merge/deploy transition coverage remains `not_observable`

Supervisor integration warnings are stored as bounded metadata at:

```
state/session-supervisor/monitor-warning.json
```

Legacy monitor logs:

- `logs/agent-timeout-monitor.log`
- `logs/agent-timeout-monitor.error.log`

Coverage gap

This task observes and reports. It does not authoritatively block MC `ready`/`done`, merge, or deploy transitions. Claude hooks provide early feedback only. Universal exact-revision enforcement must be implemented separately in the shared central MC transition path and instrument merge/deploy paths. Until that dependency is independently accepted, reports must say `not_observable` or `partial_coverage`, never `clear` or `passed`.

Rollout evidence

- Exact-SHA Proveo PASS: `/Users/makinja/system/evidence/105577/proveo-session-supervisor-dfaea126/REPORT.md`
- Post-install PASS and payload acceptance: `/Users/makinja/system/evidence/105577/proveo-session-supervisor-postinstall/REPORT.md`
- Post-install cycles and canary evidence: `/Users/makinja/system/evidence/105577/session-supervisor-postinstall/`
- Backup: `/tmp/alai/session-supervisor-live-backup-105577-20260713T230157Z`

The accepted implementation files were installed byte-for-byte. Operational promotion changed only `config/session-supervisor.json` from `enabled:false` to `enabled:true` after Proveo payload acceptance, the controlled canary, daemon reload, and multiple healthy daemon cycles.

Rollback

Primary rollback is configuration-only:

1. Set `enabled:false` in `config/session-supervisor.json` to stop supervisor Slack delivery.
2. Verify the next health tick reports `enabled:false`.
3. If the supervisor invocation itself must be removed, restore the backed-up `tools/agent-timeout-monitor.js` and controlled-reload only `com.alai.agent-timeout-monitor`.

Do not stop or modify unrelated sessions. Legacy spawn-timeout monitoring must remain active.

Revision #1

Created 2026-07-13 23:14:54 UTC by John

Updated 2026-07-13 23:14:54 UTC by John