Drift Detection Design

ANVIL Filesystem Drift Detection Daemon — Design Specification

Purpose: Automated weekly detection of canonical path registry violations, CLAUDE.md scope drift, and filesystem chaos re-introduction. Prevents split-brain recurrence after ANVIL FS Sweep (ADR-022).

Status: Design complete, build phase deferred to separate MC

Owner: John (orchestrator)

Last Updated: 2026-05-07 (ANVIL FS Sweep Phase 3 Wave 2)

---

1. Problem Statement

ANVIL FS Sweep (MC #99637, ADR-022) resolved 15 split-brain dir names, archived deprecated content, and established a canonical path registry. However, **without automated monitoring**, agents can unknowingly recreate chaos:

Example Drift Scenarios:

1. Agent sees "no `~/system/clients/`" → creates it, not knowing `~/ALAI/clients/` is canonical
2. Scope-tied CLAUDE.md files edited without updating scope-comment headers → 4-way load context breaks
3. Surprise-canonical paths (`~/aisystem/`, `~/system/security/`) accidentally deleted → live code breaks
4. Organizational drift (personal project reappears under `~/ALAI/web-worktrees/`) goes unnoticed

Root Cause: No feedback loop. One-time cleanup is insufficient without ongoing compliance checks.

---

2. Design Goals

Primary Goals

1. **Detect split-brain re-introduction** — Weekly check that archived paths stay deleted
2. **Enforce CLAUDE.md scope hygiene** — Each file's header matches its load context
3. **Protect surprise-canonical paths** — Detect if live-referenced dirs disappear
4. **Monitor specialist mapping integrity** — `specialist-mapping.json` refs match actual dirs
5. **Flag org-fit violations** — Warn (not error) on semantic-fit issues like personal projects under commercial trees

Non-Goals

1. **Not a fixer** — Daemon detects, does not auto-fix. Alerts HiveMind, creates MC, escalates to John.
2. **Not a full FS audit** — Does not scan all 62,838 paths weekly. Targets known drift patterns only.
3. **Not real-time** — Runs weekly, not on every file change (too expensive).

---

3. Architecture

Trigger Mechanism

LaunchAgent: `com.john.anvil-fs-drift-detection.plist`

Schedule: Every 7 days (Sunday 03:00 local time)

Run Condition: ANVIL host only (not on remote VMs)

Timeout: 10 minutes max (if check hangs, daemon aborts and alerts)

Script Location

Path: `~/system/daemons/scripts/anvil-fs-drift-detection.sh`

Language: Bash (for filesystem ops, jq for JSON parsing)

Dependencies: jq, grep, curl, node

Output:

- **Success (no drift):** Log to `~/system/logs/anvil-fs-drift-detection.log`, no alert
- **Drift detected:** HiveMind alert + create H-priority MC + log

---

4. Drift Detection Checks

Each check runs sequentially. If ANY check fails, daemon immediately alerts and continues to remaining checks (fail-fast on alerting, but complete all checks for full report).

Check 1: CLAUDE.md Scope Headers

Purpose: Each of 4 CLAUDE.md files MUST have a scope-comment header matching its load context.

Expected Outcome: All 4 files have scope headers. If missing/wrong, flag as drift.

Rationale: Without scope headers, editors may accidentally write global rules into project-specific files (or vice versa).

Check 2: Specialist Mapping Integrity

Purpose: `~/system/agents/specialist-mapping.json` references to agent definition files MUST point to actual existing dirs/files.

Expected Outcome: All referenced agent files exist. If any ref is broken, flag as drift.

Rationale: Broken refs cause agent routing failures (John tries to dispatch to non-existent agent).

Check 3: MUST NOT Recreate List

Purpose: Paths archived during ANVIL FS Sweep MUST NOT reappear on disk. If they do, split-brain is re-introduced.

List of paths:

- `~/system/archive`
- `~/system/deprecated`
- `~/system/deployments`
- `~/system/plans`
- `~/system/clients`
- `~/system/infrastructure`

- `~/system/internal`
- `~/system/legal`
- `~/system/org`
- `~/system/pipeline`
- `~/system/processes`
- `~/system/products`
- `~/system/sales`
- `~/system/web`

Expected Outcome: None of these paths exist. If any exists, flag as split-brain re-introduction.

Rationale: Prevents silent chaos. If agent recreates `~/system/clients/`, future agents may write to it instead of canonical `~/ALAI/clients/`.

Check 4: Surprise-Canonical Paths Still Exist

Purpose: 4 paths upgraded to canonical during Phase 1.6 content-peek MUST still exist (live code reads from them).

Paths:

- `~/aisystem`
- `~/system/security`
- `~/system/schemas`
- `~/system/hooks`

Expected Outcome: All 4 dirs exist. If any missing, flag as regression (live scripts will fail).

Rationale: These paths were not initially canonical but are **read by live tools** (Mehanik, password-share.js, etc.). Deletion breaks runtime.

Check 5: Tree Ownership Violations (Warning-Level)

Purpose: Detect semantic-fit issues like personal projects under commercial brand tree. This is **organizational audit** territory (deferred in ADR-022 Consequences), so flag as WARNING not ERROR.

Expected Outcome: Logs warnings (not errors). Does NOT block or alert HiveMind. Just logs for human review.

Rationale: Org-fit is subjective (requires CEO judgment). Daemon flags suspicious patterns but doesn't escalate as hard failure.

5. Alerting & Escalation

Success Case (No Drift)

Log Entry:

```
[2026-05-14 03:00:01] ANVIL FS Drift Detection: All checks PASS. No drift detected.
```

No HiveMind alert, no MC creation.

Drift Detected (Any Check Fails)

Immediate Actions:

1. **Log detailed findings** to `~/system/logs/anvil-fs-drift-detection.log`
2. **POST HiveMind alert** (category: `filesystem-drift`, priority: `high`)
3. **Create MC** via `node ~/system/tools/mc.js add` with title: `[DRIFT] ANVIL FS canonical violation detected — see drift log YYYY-MM-DD`
4. **Set MC priority H**, owner: `john`, category: `system`

Warning Case (Org-Fit Issues)

Log Entry (not alert):

```
[2026-05-14 03:00:10] [WARNING] Personal project ~/ALAI/web-worktrees/ucenje-v2 under commercial
```

No HiveMind alert, no MC. Human reviews log weekly.

---

6. LaunchAgent Configuration

File Path: `~/Library/LaunchAgents/com.john.anvil-fs-drift-detection.plist`

Key Configuration:

- **NOT KeepAlive** (learned from mlx-router BLOCKER in ADR-022)
- Runs once weekly, not on every boot
- 10-minute timeout prevents infinite hangs

---

7. Success Criteria

Daemon is considered **successful** if:

1. **Runs weekly without hang** (10-minute timeout not hit)
2. **Logs output** to stdout/stderr paths
3. **Detects known drift patterns** (unit test: temporarily create `~/system/clients/`, verify alert)
4. **Creates MC on drift** (verify mc.js call succeeds)
5. **Does not false-positive** (clean system → no alert)
6. **Warnings logged, not alerted** (org-fit issues don't create MCs)

---

8. Testing Plan (Pre-Build)

Before building the daemon, validate design assumptions with 6 unit tests:

1. **Scope Header Detection:** Remove scope header from `~/.claude/CLAUDE.md`, verify drift flagged
2. **MUST NOT Recreate Detection:** Create `~/system/clients/`, verify drift flagged
3. **Surprise-Canonical Regression:** Rename `~/system/security/`, verify drift flagged
4. **Specialist Mapping Broken Ref:** Add fake ref to `specialist-mapping.json`, verify drift flagged
5. **Full Run (No Drift):** Clean system, verify log shows "All checks PASS", no MC created
6. **Full Run (With Drift):** Introduce 2 drift scenarios, verify log shows both, MC created with H priority

---

9. Dependencies

System Requirements

- **OS:** macOS (LaunchAgent-based)
- **Shell:** Bash 4.0+ (for arrays, `set -euo pipefail`)
- **Tools:** jq, grep, curl, node

ALAI Infrastructure

- **mc.js:** Mission Control CLI (`node ~/system/tools/mc.js`)
- **HiveMind API:** (endpoint TBD — currently TODO in script)

- **Canonical Registry:** Canonical Registry page (authoritative MUST NOT recreate list)

Related Systems

- **ZAKON #28 Max Depth Boundary:** Drift detection MC creation does NOT count toward emergent-spawn depth (it's a daemon, not agent-spawned)
- **Daemon Fleet Watchdog:** Monitors drift daemon's exit code (if non-zero, flags as silent failure)

---

10. Future Enhancements (Out of Scope for Initial Build)

1. **Real-Time inotify Monitoring:** Use `fswatch` or `inotify` for instant detection (higher CPU cost)
2. **Auto-Fix Mode:** Add `--fix` flag to auto-delete violated paths (risky, requires CEO approval)
3. **Trend Analysis:** Store drift events in SQLite DB, generate weekly trend report
4. **Integration with Archive-First Scan:** Merge into single weekly "filesystem health" daemon

---

11. Build Phase MC Stub

Title: `[DAEMON] Build ANVIL FS drift detection daemon (weekly canonical registry enforcement)`

Deliverables:

1. Bash script: `~/system/daemons/scripts/anvil-fs-drift-detection.sh` (5 checks + alerting)
2. LaunchAgent plist: `~/Library/LaunchAgents/com.john.anvil-fs-drift-detection.plist` (weekly Sunday 03:00)
3. Unit tests: All 6 test cases PASS
4. Integration: mc.js call verified, HiveMind POST stubbed (TODO endpoint)
5. Daemon fleet watchdog: Add drift daemon to monitored list

Acceptance Criteria:

- All 5 checks implemented
- LaunchAgent loaded: `launchctl load ~/Library/LaunchAgents/com.john.anvil-fs-drift-detection.plist`
- Manual run PASS on clean system

- Manual run ALERT on intentional drift (create `~/system/clients/`, verify MC created)
- Logs to `~/system/logs/anvil-fs-drift-detection.log`
- Proveo validation: Unit tests 1-6 PASS

Dependencies: ADR-022 (canonical registry established), mc.js (Mission Control CLI working)

Effort: ~2 hours (script + plist + tests)

Priority: M (not H — BLOCKER resolved, this is preventive maintenance)

Owner: FlowForge (or John if simple Bash task)

---

12. References

Authoritative Documents

- **Canonical Registry:** Canonical Registry page
- **ADR-022:** ADR-022 page

Related Systems

- **Daemon Fleet Watchdog:** `~/system/daemons/scripts/daemon-fleet-watchdog.sh` (monitors drift daemon health)
- **Archive-First Scan:** `com.alai.archive-first-scan` LaunchAgent (overlapping concern — candidate for merge)

Prior Art

- **MC #10043:** Reform Execution Backlog (drift detection was surfaced here)

Revision #2
Created 2026-05-07 15:31:36 UTC by John
Updated 2026-06-07 20:01:35 UTC by John