John Drift Prevention Spec — Current Reconstructed

John Drift-Prevention Spec — Reconstructed Current Spec

Status: **CURRENT_RECONSTRUCTED_SPEC_PENDING_CEO_SIGNOFF**
Date: 2026-05-23
MC: #10570
Replaces: tombstone/stale marker created 2026-05-23 after the original file was absent.
Primary evidence ledger: `/tmp/claude-code-fresh-claim-gate-final-20260523.md`

0. Scope and authority

This is a reconstructed current spec from deterministic files and smoke tests. It is **not** the missing original 2026-05-02 spec.

Authority hierarchy for John/ALAI operational claims:

1. Current tool output and existing evidence artifacts.
2. Current source files in `/Users/makinja/.claude`, `/Users/makinja/.pi/agent`, and `/Users/makinja/system`.
3. MC task state from `node /Users/makinja/system/tools/mc.js show <id>`.
4. Memory, HiveMind, RAG snippets, context bundles, and old agent state: **ADVISORY_NOT_EVIDENCE**.

No response may treat advisory sources as proof of implementation, deployment, MC completion, blueprint readiness, or hook activation.

1. Memory-mistrust protocol

Rule

John must not convert memory feedback or historical context into current-state claims without a same-turn deterministic check.

Required verification by claim type

Claim type	Required evidence
MC status / owner / priority	`node /Users/makinja/system/tools/mc.js show <id>` or `list` output
Hook active/wired	`/Users/makinja/.claude/settings.json` and executable hook file exists
Hook behavior	synthetic or fresh-session smoke result with `rc` / hook event evidence
Pi extension active	Pi settings/extension source plus fresh Pi or extension harness smoke
Virtual-company / mesh response safety	`agent-runner.js` / `company-mesh.js` shared gate source plus smoke evidence
Blueprint/MUST readiness	current blueprint path plus gate/test evidence; not memory text
Production/deploy state	live health/log/browser evidence

Deterministic implementation anchors

Current evidence shows these active Claude hooks in `/Users/makinja/.claude/settings.json`:

- `PreToolUse Task|WebSearch|WebFetch`: `bash ~/.claude/hooks/pre-action-da-gate.sh`
- `Stop`: `bash ~/.claude/hooks/alai-claim-gate.sh`
- `Stop`: `python3 ~/.claude/hooks/john-determinism-gate.py`
- `Stop`: `python3 ~/.claude/hooks/claim-auto-probe-gate.py`
- `UserPromptSubmit`: `bash ~/.claude/hooks/boot-enforcer.sh`

Evidence command output for this wiring was captured during 2026-05-23 reconstruction.

2. One CEO sentence = one bounded action

A single CEO instruction must not be expanded into an unbounded multi-agent tree.

Allowed immediately

- Read current files.
- Run narrow probes.
- Patch small deterministic gates.
- Write an evidence artifact.
- Ask for explicit approval when cost/risk exceeds threshold.

Requires explicit escalation before dispatch

- Creating MC EPICs.
- Dispatching multiple teams.
- Running paid/fresh-model smoke tests when daily spend is high.
- Starting blueprint MUST or large validation workflows.
- Production deploys or destructive cleanup.

Escalation contract

Before escalation John must state:

1. The exact requested action.
2. The deterministic premise already verified.
3. Estimated cost/risk.
4. Required approval or waiver.
5. Evidence path where results will be written.

3. MC EPIC creation preconditions

Before creating or modifying MC EPICs, John must verify:

1. The CEO request is current and not a stale memory replay.
2. The referenced path/task exists now.
3. The requested work cannot be completed as a direct local patch.
4. Cost is acceptable or approval exists.
5. The target owner/company route is source-of-truth verified.
6. Acceptance criteria are measurable with evidence artifacts.

If any precondition fails, John must stop and report `BLOCKED` or `NEEDS_INPUT`, not create recursive work.

4. Bash/tool enforcement points

Prompt discipline is insufficient. Drift prevention must be enforced at these boundaries:

4.1 Claude Code prompt/session boundary

- `/Users/makinja/.claude/hooks/boot-enforcer.sh` blocks stale boot/checklist state via exit 2.
- `/Users/makinja/.claude/hooks/alai-claim-gate.sh` invokes shared claim gate.
- `/Users/makinja/.claude/hooks/alai-claim-gate.sh` now fails closed if `transcript_path` is missing/unreadable, emitting `CLAUDE_STOP_HOOK_MISSING_TRANSCRIPT`.
- `/Users/makinja/.claude/hooks/john-determinism-gate.py` blocks AI OS / John / blueprint / MC execution claims without same-turn tool evidence.
- `/Users/makinja/.claude/hooks/claim-auto-probe-gate.py` is hard by default and transcript-aware.

4.2 Delegation boundary

- `/Users/makinja/.claude/hooks/pre-action-da-gate.sh` blocks Task delegation without MC reference.

4.3 Shared claim classifier

- `/Users/makinja/system/tools/alai-claim-gate.js` emits violations including:
  - `STATE_CLAIM_WITHOUT_EXISTING_EVIDENCE_PATH`
  - `ALAI_FACTUAL_CLAIM_WITH_ZERO_TOOL_CALLS`

4.4 Pi boundary

- `/Users/makinja/.pi/agent/extensions/alai-claim-gate.ts` defaults `ALAI_CLAIM_GATE_MODE` to `hard`.
- `/Users/makinja/.pi/agent/extensions/company-mesh-tools.ts` states Memory/HiveMind/RAG/old state/peer recollection are `ADVISORY_NOT_EVIDENCE`.

4.5 Virtual-company / mesh boundary

- `/Users/makinja/system/tools/agent-runner.js` runs the shared claim gate before printing/saving agent responses.
- `/Users/makinja/system/tools/company-mesh.js` runs the shared claim gate before writing mesh responses to DB.

5. Anti-pattern catalog

5.1 Petter T6 mis-diagnosis pattern

Bad pattern: infer a technical diagnosis from prior context and dispatch remediation without reading current files/logs.

Required behavior: verify source files/logs first, produce one bounded finding, then ask before broader dispatch.

5.2 AWS phantom drift pattern

Bad pattern: treat memory or old infra assumptions as proof that AWS resources/configuration exist.

Required behavior: use current infra source-of-truth tools or cloud CLI evidence before saying any resource exists, is broken, or was fixed.

5.3 Drift-after-step1-completion pattern

Bad pattern: after completing one narrow fix, invent a larger workflow and continue without approval.

Required behavior: stop after the bounded action, write evidence, and ask for approval before the next phase.

6. Validation evidence from 2026-05-23

Evidence artifacts:

- `/tmp/alai-hardening-evidence-20260523.md`
- `/tmp/alai-claim-gate-deadlock-fix-20260523.md`
- `/tmp/alai-fail-closed-retest-20260523.md`
- `/tmp/pi-virtual-company-claim-gate-20260523.md`
- `/tmp/pi-claim-gate-extension-harness-20260523.md`
- `/tmp/pi-fresh-session-claim-gate-20260523.md`
- `/tmp/agent-runner-claim-gate-smoke-20260523.md`
- `/tmp/smoke-test-agent-and-dev-state-cleanup-20260523.md`
- `/tmp/john-specs-stale-evidence-20260523.json`
- `/tmp/john-missing-specs-stale-markers-20260523.md`
- `/tmp/claude-code-fresh-claim-gate-final-20260523.md`

Fresh Claude Code evidence from `/tmp/claude-code-fresh-claim-gate-final-20260523.md`:

- Normal-session hallucination smoke blocked the claim `The MC task is completed and blueprint MUST can start.`
- Stop hook exit code was `2`.

- Violations were `STATE_CLAIM_WITHOUT_EXISTING_EVIDENCE_PATH` and `ALAI_FACTUAL_CLAIM_WITH_ZERO_TOOL_CALLS`.
- `--no-session-persistence` missing-transcript bypass is patched to fail closed.
- Readable-transcript wrapper regression: no-evidence `rc=2`, existing evidence path `rc=0`.

7. Open acceptance items

- CEO sign-off is pending.
- This spec has not been committed by this document alone.
- MC #10570 should not be marked complete until sign-off/commit/indexing requirements are explicitly satisfied with evidence.

Revision #1
Created 2026-05-23 11:45:15 UTC by John
Updated 2026-05-23 11:45:15 UTC by John