AI Factory Pipeline — Gate Matrix & Dispatch Flow

ALAI AI Factory Pipeline — Gate Matrix & Dispatch Flow

Status: Spec for MC #10536 (parent #10612 system-uvezivanje master), Step 2.5a **Author:** anthropic-chief-architect (subagent, dispatched by John under [CEO_APPROVED] B→C transition) **Date:** 2026-05-03 **Source-of-truth basis:** Read-only derivation from the following files (absolute paths, last-modified mtimes UTC-local mixed; sha256 of head listed in Section 7):

- `/Users/makinja/.claude/settings.json` (mtime 2026-05-03 00:25:50)
- `/Users/makinja/.claude/hooks/pre-dispatch-gate.sh` (mtime 2026-05-03 00:15:00)
- `/Users/makinja/.claude/hooks/postflight-gate.sh` (mtime 2026-04-30 16:14:41)
- `/Users/makinja/.claude/hooks/lock-john-dispatch-cap.sh` (mtime 2026-04-30 22:48:51)
- `/Users/makinja/.claude/hooks/john-max-depth-gate.sh` (mtime 2026-05-03 00:14:03)
- `/Users/makinja/.claude/hooks/one-ceo-turn-mc-cap.sh` (mtime 2026-05-02 23:41:44)
- `/Users/makinja/.claude/hooks/one-ceo-turn-dispatch-cap.sh` (mtime 2026-05-03 00:25:39)
- `/Users/makinja/.claude/hooks/pre-mc-add-gate.sh` (mtime 2026-05-03 00:24:14)
- `/Users/makinja/.claude/hooks/ceo-token-origin-gate.sh` (mtime 2026-05-03 00:11:23)
- `/Users/makinja/.claude/hooks/README-evidence-quality-gate.md` (mtime 2026-02-20 10:55:28)
- `/Users/makinja/system/kernel/pi-orchestrator.js` lines 3380–3454 (mtime 2026-05-02 23:39:21)

The Kotlin binary `/Users/makinja/.claude/hooks/alai-hooks` (16,476,240 bytes, mtime 2026-05-02 23:28) is opaque — it exits silently on `--help`/`help` invocation and on bare invocation. Subcommand semantics for it are derived solely from (a) the README at `~/.claude/hooks/README-evidence-quality-gate.md` and (b) the dispatch-pattern in `settings.json`, and are marked `OPAQUE` where source cannot be confirmed. The branch `feat/blueprint-check-stack-aware` does NOT contain `tools/blueprint-check.js` (verified via `git ls-tree`); only `tools/blueprint-registry.js` and `tools/blueprint-runner.js` exist there. Blueprint enforcement therefore runs in `pre-dispatch-gate.sh` Check 9 advisory mode (`fail-open`).

1. Pipeline Overview

The ALAI AI factory pipeline is a deterministic gate sandwich wrapped around a non-deterministic LLM core. Every CEO turn enters a `UserPromptSubmit` cascade that classifies intent, refreshes counters, and primes Mehanik state. John then routes the request: H/BLOCKER → `/prompt-forge` → `/mehanik` (writes `/tmp/mehanik-cleared-<id>` marker with 13 mandatory fields) → `Task` dispatch → specialist agent work under `PreToolUse(Bash|Write|Edit)` gates → `/task-postflight` (writes `~/system/state/postflight-cleared-<id>.json`) → `mc.js done`. M/L/trivial tasks skip `/prompt-forge` per ZAKON #25. Hard Constraint #3 — "Builder cannot say done" — is structurally enforced via Plan #10264's 5+1-layer gate stack; the Bash hook layer is `postflight-gate.sh` (priority cache + session_id + 4h TTL). The dispatch flow is gated at THREE failure-modes: (a) too-deep recursion (`john-max-depth-gate.sh` trip-wire 1 cuts at depth 3+), (b) too-wide CEO-turn fan-out (`one-ceo-turn-{mc,dispatch}-cap.sh`), (c) self-issued override tokens (`ceo-token-origin-gate.sh` reads `/tmp/ceo-turn-<session>.txt`).

Two gates are deactivated or absent: `pi-orchestrator.js` (the database-backed scheduler at lines 3380–3454) is currently OFF per session-state.md ACTIVE_THREAD context; `blueprint-check.js` does not exist on `main` and does not exist on `feat/blueprint-check-stack-aware`, so Check 9 of `pre-dispatch-gate.sh` is advisory-only and fails open with the message `blueprint_check_unavailable`. An `active-thread-lock` hook is referenced in session-state.md ("4. structural layer") as PENDING and does not exist on disk. ZAKON #25, #27, #28 and Hard Constraints #1/#2/#3 form the policy layer that the gates instantiate.

2. Gate Matrix

#	Gate	Path	Phase	Reads	Writes	Block exit (file:line)	Bypass token	Notes

1	postflight-gate	`~/.claude/hooks/postflight-gate.sh`	PreToolUse Bash	`~/system/state/mc-priority-cache.json`, `~/system/state/postflight-cleared-<id>.json`, `$CLAUDE_SESSION_ID`, `~/.claude/session-state.md`	stderr	`exit 2` at lines 84, 108, 115, 128, 135, 152, 170	none for missing/expired marker; `--force --reason ≥20chars` allowed (line 118-120); UNCONDITIONAL block on cache failure for H/BLOCKER (A1 fail-secure, line 84)	Layer 2 of Plan #10264 5+1 stack. 4-hour TTL on marker (line 133). Session-id A6 race protection (line 169). B10 fail-secure: empty session context + H/BLOCKER = BLOCK (MC #10313, lines 149-156).
2	caddyfile-validate-gate	`~/.claude/hooks/caddyfile-validate-gate.sh`	PreToolUse Bash AND Write|Edit|MultiEdit	(not read; deferred — outside scope)	(not inspected)	OPAQUE	OPAQUE	Listed in settings.json:53 and :233 — not analyzed in this spec.
3	delegation-required-gate	`~/.claude/hooks/delegation-required-gate.sh`	PreToolUse Bash	(not read)	(not inspected)	OPAQUE	OPAQUE	settings.json:58. Enforces Hard Constraint #1 ("John does NOT build").

4	alai-hooks bash	`~/.claude/hooks/alai-hooks bash` (Kotlin binary)	PreToolUse Bash	OPAQUE	OPAQUE	OPAQUE — derived from Kotlin binary size 16.4 MB, no `--help` output	OPAQUE	settings.json:63. Per feedback memo `feedback_alai_hooks_fixed_2026-04-29.md`, this is the live middle-layer enforcement (lead-guard + bash-danger observed blocking real-time).
5	alai-hooks evidence-gate	`~/.claude/hooks/alai-hooks evidence-gate`	PreToolUse Bash	`/tmp/verify-<id>/claims.json`, `/tmp/verify-<id>/evidence/*`, `/tmp/verify-<id>/cove-self-check.md`, `/tmp/verify-<id>/validator-independent.json` (per README)	stderr	OPAQUE — README states `Exit 2` when issues found (`README-evidence-quality-gate.md` line 124-141)	none documented; LOW priority bypassed if no `/tmp/verify-<id>/` dir	Implements CoVe (Chain-of-Verification). HIGH requires validator-independent.json with zero mismatches (README: 25-27).

6	alai-hooks pipeline-gate	`~/.claude/hooks/alai-hooks pipeline-gate`	PreToolUse Bash	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:73. Reference in `ceo-token-origin-gate.sh:91-93` cites "PipelineGate.kt line 29: command.contains('mc.js done') fires on --desc 'mc.js done'" — confirms Kotlin source exists in alai-hooks tree but is not source-readable from disk here.
7	alai-hooks deploy-gate	`~/.claude/hooks/alai-hooks deploy-gate`	PreToolUse Bash	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:78. ZAKON PI2 enforcement (deploy verification).
8	bash-danger-gate	`~/.claude/hooks/bash-danger-gate.sh`	PreToolUse Bash	(not read)	OPAQUE	OPAQUE	OPAQUE	settings.json:83. Listed in `permissions.deny` are static (`rm -rf /`, `git push --force*`, etc.) — settings.json:25-32.

9	john-max-depth-gate (TW1)	`~/.claude/hooks/john-max-depth-gate.sh`	PreToolUse Task|Agent	`/tmp/mc-active-task`, `node ~/system/tools/mc.js show <id>`	`~/.claude/hooks/john-max-depth-gate.log`	`exit 2` at line 110 (depth ≥3)	`[CEO_APPROVED]` in dispatch prompt (line 95, 111)	Bootstrap-exempt: mehanik|validator|devils-advocate|anthropic-chief-architect (line 60). Depth walked via `Parent: #N` regex.
10	john-max-depth-gate (TW2)	same	PreToolUse Bash (mc.js add)	`/tmp/mehanik-cleared-<parent>` (`approved_subtask_count`, `expires_at`), `/tmp/john-emergent-<session>.cnt`	`/tmp/john-emergent-<session>.cnt`, drift-stop memo, log	`exit 2` at line 212 when `emergent_count > approved + 3`	`[CEO_APPROVED]` (line 191)	Counter rolls back on block (line 211) so retries don't inflate. ZAKON #28. Mehanik marker now TTL-aware (MC #10611): `expires_at` validated before reading `approved_subtask_count` (lines 164-187).
11	john-max-depth-gate (TW3)	same	PreToolUse Bash (mc.js add)	parent MC `Category:` field	`~/system/specs/drift-stop-<parent>-<ts>.md`	SOFT trip — no exit 2 (line 283)	n/a (warn only)	Cross-domain category mismatch. ZAKON #27 enforcement.
12	pre-mc-add-gate (intent)	`~/.claude/hooks/pre-mc-add-gate.sh`	PreToolUse Bash	`/tmp/ceo-intent-<session>.json`	(none)	`exit 2` at line 24 (CEO intent = QUESTION|CRITIQUE)	`[CEO_APPROVED]` (line 19)	Genesis: feedback_john_kotlin_rabbit_hole_2026-05-02.md.

13	pre-mc-add-gate (sunset)	same	PreToolUse Bash	`--desc` text in command	`/tmp/pre-mc-add-gate.log`	`exit 2` at line 61	`[CEO_APPROVED]` (line 48)	H/BLOCKER/EPIC require sunset/replace/phantom keyword + ADR/SHA/BookStack citation. Genesis: AWS phantom drift 2026-05-02.
14	pre-mc-add-gate (citation)	same	PreToolUse Bash	`--desc` text	log	`exit 2` at line 68	`[CEO_APPROVED]` (line 48)	All H/BLOCKER/EPIC mc.js add require `(per ADR-NNN file:line)` OR `git SHA:` OR `BookStack: https://`.
15	ceo-token-origin-gate (postflight bypass)	`~/.claude/hooks/ceo-token-origin-gate.sh`	PreToolUse Bash	command env-var prefix	`/tmp/ceo-token-gate.log`	`exit 2` at line 160 (`unconditional_block`, never dry-run)	UNCONDITIONAL — no bypass	`POSTFLIGHT_GATE_BYPASS=1` permanently blocked. Dry-run does NOT override. Bug C fix (MC #99016): anchored bypass-var check prevents `--desc 'POSTFLIGHT_GATE_BYPASS=1'` false-positive (lines 133-158).

16	ceo-token-origin-gate (force-rate)	same	PreToolUse Bash	command env-var prefix	log	`exit 2` at line 164 (`unconditional_block`)	UNCONDITIONAL	`MC_FORCE_RATE_OVERRIDE=1` permanently blocked.
17	ceo-token-origin-gate (force-done)	same	PreToolUse Bash	tokenized command (segments)	log	`exit 2` at line 183 (`unconditional_block`)	UNCONDITIONAL	`--force` flag on `mc.js done` permanently blocked (genesis: 7 forced closures 2026-05-02).
18	ceo-token-origin-gate (token-origin)	same	PreToolUse Bash	`/tmp/ceo-turn-<session>.txt`	log	`exit 2` at line 207 (no log) and 214 (token absent from log)	`CEO_TOKEN_GATE_DRY_RUN=1` (advisory only)	Self-issued `[CEO_APPROVED]` blocked. CEO must include token in their actual message.
19	postflight-provenance-gate	`~/.claude/hooks/postflight-provenance-gate.sh`	PreToolUse Bash	(not read in this spec)	OPAQUE	OPAQUE	OPAQUE	settings.json:103. Companion to postflight-gate.
20	alai-hooks claim-blocker	`~/.claude/hooks/alai-hooks claim-blocker`	PreToolUse Bash	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:108.
21	alai-hooks pre-mc-add-gate	`~/.claude/hooks/alai-hooks pre-mc-add-gate`	PreToolUse Bash	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:113. Likely Kotlin re-implementation of bash gate (Section 13/14 of bash file). Duplicate execution path — both fire.

22	alai-hooks one-ceo-turn-mc-cap	`~/.claude/hooks/alai-hooks one-ceo-turn-mc-cap`	PreToolUse Bash	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:118. Likely Kotlin twin of `one-ceo-turn-mc-cap.sh`.
23	one-ceo-turn-mc-cap (Sec 1)	`~/.claude/hooks/one-ceo-turn-mc-cap.sh`	PreToolUse Bash (mc.js add)	`/tmp/john-mc-turn-counter.json`	same	`exit 2` at line 62 when count > 1 in turn	`[CEO_APPROVED_MULTIPLE_MC]` (line 44) or `[CEO_APPROVED]` (line 46)	Resets per UserPromptSubmit via `mc-turn-reset.sh` (settings.json:411). MC #99015 Approach A fix: token counter increment now happens AFTER cap-check (line 108), not before. Blocked attempts no longer inflate counter.
24	one-ceo-turn-mc-cap (Sec 2 — token rate-limit)	same	PreToolUse Bash	`/tmp/ceo-approved-token-uses-<session>.count`	same	`exit 2` at line 105 (token used >1× in session)	none — must be re-issued by CEO in new turn	Design flaw FIXED (MC #99015 Approach A): counter increment moved to line 108, AFTER cap-check at line 100. Blocked attempts no longer inflate counter.

25	one-ceo-turn-dispatch-cap	`~/.claude/hooks/one-ceo-turn-dispatch-cap.sh`	PreToolUse Task|Agent	`/tmp/john-dispatch-turn-counter.json`, latest `/tmp/mehanik-cleared-*` (`approved_subtask_count`)	counter file	`exit 2` at line 56 when count > Mehanik-approved cap (default 1)	`[CEO_APPROVED]` (line 18)	v3 Rank 3. Genesis: Kotlin rabbit-hole 2026-05-02.
26	lock-john-dispatch-cap	`~/.claude/hooks/lock-john-dispatch-cap.sh`	PreToolUse Task|Agent	`/tmp/lock-john-session-<session>.cnt`	same	`exit 2` at line 93 when session count > 8	`[CEO_APPROVED]` (line 84)	Bootstrap-exempt: mehanik|validator|devils-advocate (line 44). 8/session cap.
27	claude-hooks pre	`~/.claude/hooks/claude-hooks pre` (Kotlin binary, 24 MB)	PreToolUse Task|Agent|WebSearch|WebFetch AND Write|Edit|MultiEdit AND mcp__playwright__.*	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:133, :163, :193. Older Kotlin binary, predates alai-hooks.
28	pre-action-da-gate	`~/.claude/hooks/pre-action-da-gate.sh`	PreToolUse Task|Agent|WebSearch|WebFetch	(not read)	OPAQUE	OPAQUE	OPAQUE	settings.json:138. "DA" = devils-advocate.
29	pre-dispatch-gate (id+marker)	`~/.claude/hooks/pre-dispatch-gate.sh`	PreToolUse Task|Agent|WebSearch|WebFetch	`/tmp/mehanik-cleared-<id>` (13 fields), `~/system/agents/specialist-mapping.json`	stderr	`exit 2` at lines 53, 61, 70, 77, 86, 95, 109, 130	`mehanik` subagent_type (line 46); `[CEO_OVERRIDE]` for blueprint check only (line 139); `TOOL_CONTRACT:` block (line 103)	13-field marker schema per MC #9230. Scope ceiling = `ceo_item_count + 2` (line 92).

30	pre-dispatch-gate (blueprint advisory)	same	same	`blueprint_score:` field in marker	stderr WARN	none — `fail-open` (line 144, 153)	`[CEO_OVERRIDE]` in prompt	Phase 1 advisory-only. Phase 3 enforcement DEFERRED — `blueprint-check.js` absent from main and from `feat/blueprint-check-stack-aware`.
31	john-max-depth-gate (Task path)	(already row 9)	PreToolUse Task|Agent	—	—	—	—	settings.json:148 fires twice (Bash and Task matchers) — same script branches on `TOOL_NAME`.
32	claude-hooks post	`~/.claude/hooks/claude-hooks post`	PostToolUse `.*`	OPAQUE	OPAQUE	async — never blocks	n/a	settings.json:245. `async: true`, exits cannot block tool result.
33	context-bundle-logger	`~/.claude/hooks/context-bundle-logger.sh`	PostToolUse `.*`	OPAQUE	OPAQUE	async, never blocks	n/a	settings.json:251.
34	trace-capture	`~/.claude/hooks/trace-capture.py`	PostToolUse `.*`	OPAQUE	OPAQUE	async, never blocks	n/a	settings.json:257.
35	memo-citation-gate (bash)	`~/.claude/hooks/memo-citation-gate.sh`	PostToolUse Read	(not read in this spec)	OPAQUE	async, never blocks	n/a	settings.json:279. Genesis: feedback_john_kotlin_rabbit_hole_2026-05-02.md.

36	alai-hooks memo-citation-gate	`~/.claude/hooks/alai-hooks memo-citation-gate`	PostToolUse Read	OPAQUE	OPAQUE	async, never blocks	OPAQUE	settings.json:285. Likely Kotlin twin of bash gate.
37	url-linter-gate	`~/system/hooks/url-linter-gate.sh`	PostToolUse Write|Edit|MultiEdit	(not read)	OPAQUE	async, never blocks	n/a	settings.json:296. 60s timeout — heaviest async hook.
38	session-output-validator	`~/.claude/hooks/session-output-validator.sh`	Stop	OPAQUE	OPAQUE	async, never blocks Stop	n/a	settings.json:309.
39	session-cleanup	`~/system/tools/session-cleanup.sh`	Stop	OPAQUE	OPAQUE	sync; outcome unknown	n/a	settings.json:315.
40	session-ledger	`~/system/tools/session-ledger.sh`	Stop AND PreCompact	OPAQUE	OPAQUE	sync 30s	n/a	settings.json:320, :347.
41	alai-hooks stop-verify	`~/.claude/hooks/alai-hooks stop-verify`	Stop	OPAQUE	OPAQUE	sync 15s	OPAQUE	settings.json:325.
42	claude-cli-cost-hook	`~/.claude/hooks/claude-cli-cost-hook.sh`	Stop (separate matcher)	OPAQUE	OPAQUE	async, never blocks	n/a	settings.json:335.
43	incident-response-mode	`~/.claude/hooks/incident-response-mode.sh`	UserPromptSubmit	OPAQUE	OPAQUE	sync 5s	OPAQUE	settings.json:360.
44	boot-enforcer	`~/.claude/hooks/boot-enforcer.sh`	UserPromptSubmit	OPAQUE	OPAQUE	sync 5s	OPAQUE	settings.json:365. Likely enforces ZAKON `bash ~/system/boot.sh`.

45	user-message-logger	`~/.claude/hooks/user-message-logger.sh`	UserPromptSubmit	stdin (CEO message)	(presumably writes `/tmp/ceo-turn-<session>.txt` — referenced by ceo-token-origin-gate.sh:173)	sync, exits 0	n/a	settings.json:370. **Confirmed write target inferred from downstream consumer.**
46	alai-hooks auto-verify	`~/.claude/hooks/alai-hooks auto-verify`	UserPromptSubmit	OPAQUE	OPAQUE	sync 30s	OPAQUE	settings.json:375.
47	alem-instruction-checker	`~/.claude/hooks/alem-instruction-checker.sh`	UserPromptSubmit	OPAQUE	OPAQUE	async, never blocks	n/a	settings.json:381.
48	feasibility-check-advisory	`~/.claude/hooks/feasibility-check-advisory.sh`	UserPromptSubmit	OPAQUE	OPAQUE	sync (no timeout)	n/a	settings.json:391.
49	validation-state-injector	`~/.claude/hooks/validation-state-injector.sh`	UserPromptSubmit	OPAQUE	OPAQUE	sync 5s	n/a	settings.json:400. Layer 5+1 of Plan #10264 (UserPromptSubmit injector).
50	ceo-intent-classifier	`~/.claude/hooks/ceo-intent-classifier.sh`	UserPromptSubmit	CEO message stdin	`/tmp/ceo-intent-<session>.json` (consumed by pre-mc-add-gate.sh:16)	sync 5s	n/a	settings.json:405.
51	mc-turn-reset	`~/.claude/hooks/mc-turn-reset.sh`	UserPromptSubmit	(none — resets)	`/tmp/john-mc-turn-counter.json`, `/tmp/john-dispatch-turn-counter.json` (resets to 0)	sync 3s	n/a	settings.json:410. Companion to one-ceo-turn-{mc,dispatch}-cap.sh.

52	ceo-token-log-userpromptsubmit	`~/.claude/hooks/ceo-token-log-userpromptsubmit.sh`	UserPromptSubmit	CEO message stdin	`/tmp/ceo-turn-<session>.txt` (consumed by ceo-token-origin-gate.sh:173)	sync 3s	n/a	settings.json:415. **Authoritative writer of the CEO turn log.**
53	worktree-create	`~/.claude/hooks/worktree-create.sh`	WorktreeCreate	OPAQUE	OPAQUE	sync 10s	OPAQUE	settings.json:427.
54	claude-hooks session	`~/.claude/hooks/claude-hooks session`	SessionStart	OPAQUE	OPAQUE	sync 15s	OPAQUE	settings.json:439.
55	claude-hooks subagent	`~/.claude/hooks/claude-hooks subagent`	SubagentStart	OPAQUE	OPAQUE	sync 10s	OPAQUE	settings.json:451.
56	alai-hooks subagent	`~/.claude/hooks/alai-hooks subagent`	SubagentStart	OPAQUE — but observed by this very subagent's session as the source of the "TOOL-FIRST ZAKON" injection prefix	injection text into subagent context	sync 10s	OPAQUE	settings.json:456. Confirmed live by SubagentStart hook prefix observed at start of this dispatch.
57	hook-change-validator	`~/.claude/hooks/hook-change-validator.sh`	PreToolUse Write|Edit|MultiEdit	(not read)	OPAQUE	OPAQUE	OPAQUE	settings.json:173.
58	lock-context-tier1-cap	`~/.claude/hooks/lock-context-tier1-cap.sh`	PreToolUse Write|Edit|MultiEdit	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:178.
59	delegation-required-gate-write	`~/.claude/hooks/delegation-required-gate-write.sh`	PreToolUse Write|Edit|MultiEdit	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:183.

60	plan-completeness-gate	`~/.claude/hooks/plan-completeness-gate.sh`	PreToolUse Write|Edit|MultiEdit	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:188. Hard Constraint #4 — every plan must include Validation + Documentation tasks.
61	project-path-gate	`~/.claude/hooks/project-path-gate.sh`	PreToolUse Write|Edit|MultiEdit	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:198. Likely enforces cwd guardrails from `/Users/makinja/CLAUDE.md`.
62	spawn-gate write-gate	`~/system/kernel/spawn-gate.js write-gate`	PreToolUse Write|Edit|MultiEdit	OPAQUE (not read in this spec)	OPAQUE	OPAQUE	OPAQUE	settings.json:203.
63	alai-hooks write/tech-stack-gate/lead-guard/backend-guard/hallucination	`~/.claude/hooks/alai-hooks <subcmd>`	PreToolUse Write|Edit|MultiEdit (5 separate hook invocations)	OPAQUE	OPAQUE	OPAQUE	OPAQUE	settings.json:208-230. The hallucination one is referenced as the live `lead-guard` / `bash-danger` blocker per `feedback_alai_hooks_fixed_2026-04-29.md`.

64	active-thread-lock	(NOT ON DISK)	(TBD)	—	—	TBD	TBD	session-state.md line 21 marks as "Pending child #1" of system-uvezivanje-master. Does not exist as of this writing.
65	pi-orchestrator dispatch loop	`/Users/makinja/system/kernel/pi-orchestrator.js:3380-3454`	Background daemon (NOT a Claude Code hook)	`mission-control.db` (`tasks` JOIN `task_scheduling`), `MC_SCRIPT next-task --owner john|pi-orchestrator`	DLQ on timeout/retry-exhaustion (lines 3429, 3445)	`continue` (skip task) on timeout (line 3431), retry-cap (line 3446); not a "block" in the hook sense	n/a	Currently OFF per session-state.md. Implements delegation filter `delegated_to = 'pi-orchestrator'` with circuit-breaker (`cb_state`), lease (`lease_until`), and DLQ.

3. Dispatch Flow (Mermaid)

```
flowchart TD
    CEO[CEO message] --> UPS[UserPromptSubmit cascade]
    UPS --> IRM[incident-response-mode.sh]
    IRM --> BE[boot-enforcer.sh]
    BE --> UML[user-message-logger.sh]
    UML --> AAV[alai-hooks auto-verify]
    AAV --> AIC[alem-instruction-checker.sh]
    AIC --> FCA[feasibility-check-advisory.sh]
    FCA --> VSI[validation-state-injector.sh]
    VSI --> CIC[ceo-intent-classifier.sh writes /tmp/ceo-intent-SESSION.json]
    CIC --> MTR[mc-turn-reset.sh resets MC and dispatch counters]
    MTR --> CTL[ceo-token-log-userpromptsubmit.sh writes /tmp/ceo-turn-SESSION.txt]
```

```
    CTL --> John[John classify priority]
    John -->|H or BLOCKER| PF[/prompt-forge/]
    John -->|M or L or trivial| Mehanik[/mehanik/]
    PF --> Mehanik
    Mehanik --> Marker[Mehanik writes /tmp/mehanik-cleared-ID with 13 fields]
    Marker --> Disp[John dispatches Task or Agent]
    Disp --> LJDC{lock-john-dispatch-cap count under 9}
    LJDC -->|no and no CEO_APPROVED| BLK1[BLOCK exit 2]
    LJDC -->|yes| CHpre[claude-hooks pre]
    CHpre --> PADA[pre-action-da-gate]
    PADA --> PDG{pre-dispatch-gate marker valid}
    PDG -->|no| BLK2[BLOCK exit 2]
    PDG -->|yes| JMD1{john-max-depth TW1 depth under 3}
    JMD1 -->|no and no CEO_APPROVED| BLK3[BLOCK exit 2]
    JMD1 -->|yes| OCTD{one-ceo-turn-dispatch-cap under Mehanik approved}
    OCTD -->|no and no CEO_APPROVED| BLK4[BLOCK exit 2]
    OCTD -->|yes| Spec[Specialist agent runs]
    Spec --> ToolUse{Tool used}
    ToolUse -->|Bash| BashGates[postflight + caddyfile + delegation + alai bash + evidence + pipeline + deploy + bash-danger + JMD23 + pre-mc-add + ceo-token-origin + provenance + claim-blocker + alai-pre-mc + alai-octmc]
    ToolUse -->|Write or Edit| WriteGates[hook-change-val + tier1-cap + delegation-write + plan-completeness + claude-pre + project-path + spawn-gate + alai-write + tech-stack + lead-guard + backend-guard + hallucination + caddyfile]
    BashGates --> PostUse[PostToolUse async logs and traces]
    WriteGates --> PostUse
    PostUse --> SpecDone{Specialist returns}
    SpecDone --> Postflight[/task-postflight writes ~/system/state/postflight-cleared-ID.json/]
    Postflight --> McDone[mc.js done ID]
    McDone --> PFG{postflight-gate marker valid and TTL under 4h and session matches}
    PFG -->|no and not force-with-reason| BLK5[BLOCK exit 2]
    PFG -->|yes| McClose[task closed]
    McClose --> Stop[Stop hooks]
    Stop --> SOV[session-output-validator]
    Stop --> SCleanup[session-cleanup.sh]
    Stop --> SLedger[session-ledger.sh]
    Stop --> ASV[alai-hooks stop-verify]
    Stop --> CCH[claude-cli-cost-hook]
```

4. Where the pipeline currently leaks (audit, not opinion)

Observations grounded strictly in source read this session:

1. `blueprint-check.js` **does not exist.** Verified by `ls -la /Users/makinja/system/tools/blueprint-check.js` (No such file or directory) and `git ls-tree feat/blueprint-check-stack-aware tools/` (only `blueprint-registry.js` and `blueprint-runner.js`). `pre-dispatch-gate.sh:135-160` therefore runs in fail-open advisory mode, and any `blueprint_score` is whatever Mehanik wrote — without a checker tool, that field is essentially trust-the-author.
2. `alai-hooks` **binary is opaque from disk.** No source files in `~/.claude/hooks/` for the Kotlin enforcement; `alai-hooks --help` prints nothing. Behavior must be inferred from the README (`README-evidence-quality-gate.md` describes only the `evidence-gate` subcommand) and from cross-references in bash hooks (e.g. `ceo-token-origin-gate.sh:91-93` cites `PipelineGate.kt line 29`). 13 of 64 gate rows above are `OPAQUE` for this reason. This is a single point of trust for ~20% of the gate stack.
3. **Duplicate enforcement paths for the same policy.** Both `~/.claude/hooks/pre-mc-add-gate.sh` (settings.json:93) AND `~/.claude/hooks/alai-hooks pre-mc-add-gate` (settings.json:113) are wired into PreToolUse Bash. Same for `one-ceo-turn-mc-cap.sh` (settings.json:118 wires the alai-hooks twin). Two hooks evaluating the same input is fine for redundancy, but if the Kotlin twin's logic drifts from the bash, semantics become non-deterministic.
4. `active-thread-lock` **hook is referenced but absent.** `ls /Users/makinja/.claude/hooks/active-thread-lock*` returns no matches. `~/.claude/session-state.md` line 21 lists it as "Pending children #1" of system-uvezivanje-master. ZAKON #27 (one product per session) currently has no machine enforcement at hook level.
5. `pi-orchestrator.js` **delegation loop is OFF.** Confirmed by `~/.claude/session-state.md` ACTIVE_THREAD context (ACTIVE_THREAD = system-uvezivanje-master, no mention of pi-orch running). The DLQ + circuit-breaker + lease infrastructure at lines 3382-3447 is dormant; no daemon is consuming `delegated_to = 'pi-orchestrator'` tasks. session-state.md feedback log entry under "Pending children" does not list pi-orch reactivation.
6. `one-ceo-turn-mc-cap.sh` **Section 2 token-counter design flaw.** Per `~/.claude/session-state.md:27-29`: `/tmp/ceo-approved-token-uses-default.count` increments on BLOCKED attempts (script increments before the limit check at line 94-104). Counter inflates on rejected commands → legitimate next CEO turn can fail. Documented as "separate workstream, NOT drift" in session-state.
7. **Postflight session_id whitespace bug (per session-state.md:49).** "postflight-gate Bash hook strips whitespace from session-state.md header but mc.js parser preserves it → marker session_id mismatch on every flow. All 5 closures used --force." This is a live, recurring failure-mode. The `postflight-gate.sh:144` reads `head -1 ~/.claude/session-state.md | tr -d '[:space:]'` while mc.js does not normalize identically. Mismatch path: line 167 BLOCK.

8. `MEMORY.md` **auto-write absent.** Cross-referenced from feedback_sentinel_v3 family in MEMORY.md but no hook in `settings.json` writes back to memory. The Read PostToolUse hooks (memo-citation-gate × 2) only validate, do not append.
9. `TOOL_CONTRACT` **block enforcement is keyword-fragile.** `pre-dispatch-gate.sh:101` regex matches phrases like "research the/find partners/contact list" but exempts any prompt mentioning `discover.js|lightrag.js|mc.js|web-search.sh` — meaning a research-intent dispatch that name-drops `mc.js` in passing slips the gate.
10. **No** `WORKTREE_PATH` **enforcement at dispatch time.** `worktree-create.sh` fires on `WorktreeCreate` (settings.json:427, OPAQUE), but no PreToolUse gate verifies a dispatched specialist actually inherits a project worktree path. The `/Users/makinja/CLAUDE.md` cwd guardrails ("ANY file write to /Users/makinja/* outside ... → STOP") are policy text, not a hook. `project-path-gate.sh` (settings.json:198) on Write/Edit might cover this — OPAQUE, not verified in this spec.

5. Three sub-MC proposals for Step 2.5b

Proposal 1: `task_gate_events` schema

Title: Add deterministic gate-event logging table to mission-control.db **Why:** 13 of 64 gates write to per-gate ad-hoc log files (`/tmp/pre-mc-add-gate.log`, `~/.claude/hooks/john-max-depth-gate.log`, `/tmp/ceo-token-gate.log`, etc.). No unified store means we cannot answer "how often does gate X block in a week?", "which gate blocks most often per session?", or "did gate X regress after settings.json change Y?". Per Hard Constraint #2 ("No claim without evidence"), the platform itself violates this for its own gates. **Acceptance:**

1. New table `task_gate_events(id INTEGER PK, ts TEXT, session_id TEXT, gate_name TEXT, decision TEXT CHECK IN ('allow','block','warn','soft'), tool_name TEXT, mc_id INTEGER NULL, reason TEXT, raw_input_sha256 TEXT)` created via migration in `~/system/databases/migrations/` and applied to `mission-control.db`.
2. Each of the 16 gate-rows in Section 2 with non-OPAQUE source (rows 1, 9-14, 15-18, 23-26, 29, 30) appends one row per invocation via shared helper `~/.claude/hooks/_lib/log-gate-event.sh`.
3. `mc.js gate-events --tail 50 --gate <name>` subcommand reads the table.
4. Daily summary daemon `com.alai.gate-events-summary` writes top-10 blockers to `~/system/state/gate-events-daily-<date>.json`.
5. Proveo verification: 5 known-block scenarios produce 5 rows; 5 known-allow scenarios produce 5 rows; replay matches expected.

Owner: flowforge (database + bash plumbing) **Estimate:** 6h

Proposal 2: `WORKTREE_PATH` gate + worktree-enforcer

Title: Block specialist Task/Agent dispatches without explicit `WORKTREE_PATH:` block in prompt **Why:** `/Users/makinja/CLAUDE.md` cwd guardrails are policy text, not enforced. The dispatch-from-home-dir failure mode shipped real damage (genesis: feedback_drop_split_brain_root_cause.md). `project-path-gate.sh` covers Write/Edit only; a specialist that runs only Bash (`npm install`, `flyway migrate`) at a wrong cwd leaks just as much. Mehanik already records `project_path:` in the marker — the dispatch prompt should propagate it as a `WORKTREE_PATH:` directive that a new gate verifies matches. **Acceptance:**

1. `~/.claude/hooks/worktree-path-gate.sh` added to `settings.json` PreToolUse `Task|Agent` matcher (after `pre-dispatch-gate.sh`).
2. Hook reads `project_path:` from `/tmp/mehanik-cleared-<id>` and `WORKTREE_PATH:` from prompt; mismatch or absence → exit 2 (with `[CEO_APPROVED]` bypass).
3. `~/system/tools/wrap-with-worktree-path.js` helper auto-injects the directive given a Mehanik-cleared MC id.
4. Specialist agent definitions updated (5 high-traffic: codecraft, flowforge, securion, skillforge, proveo) to refuse work if first instruction is not `cd <WORKTREE_PATH>`.
5. Proveo: 3 negative cases (no path, wrong path, path outside `~/projects/` / `~/companies/`) all block.

Owner: codecraft (hook + helper) + skillforge (agent .md updates) **Estimate:** 5h

Proposal 3: blueprint Phase 3 promote OR pi-orch stays OFF (binary CEO decision)

Title: CEO decision — invest in finishing blueprint-check.js + pi-orchestrator reactivation, OR formally retire both **Why:** Two large pieces of pipeline infrastructure are currently dead: (a) `blueprint-check.js` is referenced from `pre-dispatch-gate.sh:142-160` but doesn't exist on disk or on the named feature branch — Phase 3 enforcement is "deferred to separate MC per Petter Graff plan Section 1" with no MC opened; (b) `pi-orchestrator.js` (lines 3380-3454 implements a real DLQ + circuit-breaker scheduler) is OFF and not in any system-uvezivanje sequence. Carrying dead infrastructure costs context tokens (every John session reads settings.json with these references) and creates phantom-feature drift risk. Frame to CEO as binary:

- **Option A — Promote both:** Open MC for blueprint-check.js implementation (estimate 12h codecraft); separate MC for pi-orch reactivation (estimate 4h flowforge to wire daemon + 2h proveo soak). Total cost ~18h.
- **Option B — Retire both:** Remove Check 9 from `pre-dispatch-gate.sh`; comment out `delegated_to = 'pi-orchestrator'` query in pi-orchestrator.js; delete `feat/blueprint-check-stack-aware` branch; document in ADR. Cost ~2h.

Acceptance (for the CEO-decision MC, regardless of option):

1. CEO writes one of A/B in MC comment.
2. Selected sub-plan opened as separate MC by John under [CEO_APPROVED].
3. `~/system/specs/ai-factory-pipeline.md` (this spec) updated with chosen direction.
4. `MEMORY.md` index entry added.

Owner: John (decision-routing only — does not build) **Estimate:** 0.5h CEO time + 18h or 2h follow-on depending on choice

6. Open questions for CEO

1. **Blueprint-check tool: build or kill?** Option A (build, 18h) vs Option B (retire, 2h) per Proposal 3. Yes/no on Option A?
2. `alai-hooks` **source-readability:** Should the Kotlin sources for the alai-hooks binary be checked into a readable repo path (e.g. `~/system/kernel/alai-hooks-src/`)? Currently 13 of 64 gates are OPAQUE — auditability impossible. Yes/no?
3. `active-thread-lock` **hook scheduling:** session-state.md lists this as Pending child #1 — should a sub-MC be opened in the system-uvezivanje thread for this gate, or deferred to separate thread? Yes/no on opening sub-MC now?
4. `one-ceo-turn-mc-cap.sh` **Section 2 counter design flaw:** Documented in session-state.md as "separate workstream, NOT drift". Approve fix MC now (10 min flowforge patch), or hold? Yes/no on opening fix MC?
5. **Duplicate bash + Kotlin gates** (`pre-mc-add-gate`, `one-ceo-turn-mc-cap`): keep both for redundancy, or pick one and remove the other to avoid drift? Choice = `keep-both` or `bash-canonical` or `kotlin-canonical`?

7. Source verification log

File	Lines read	sha256 (head)
`/Users/makinja/.claude/hooks/pre-dispatch-gate.sh`	1-164 (full)	`73dc93e53d3153b828b200fdc5f943494efdfef6097c260eca5da2b6286ffc37`
`/Users/makinja/.claude/hooks/postflight-gate.sh`	1-180 (full)	`23bff5fd726a63adeb465da6adaf64a36f714c0c3420f11db3db688f5d396aa3`
`/Users/makinja/.claude/hooks/lock-john-dispatch-cap.sh`	1-94 (full)	`53da2f1ec683a057ec8824e9157563a98221165548d8c499da7d28cf6146cc01`
`/Users/makinja/.claude/hooks/john-max-depth-gate.sh`	1-290 (full)	`388ca81404a480bb6252227dddb8b2835fe0781faf5695c21579dddf7c170390`
`/Users/makinja/.claude/hooks/one-ceo-turn-mc-cap.sh`	1-117 (full)	`0ab839000295a7dbd8779f57dcdef1bb03e4242b168c4097da34fd4e383a1378`
`/Users/makinja/.claude/hooks/one-ceo-turn-dispatch-cap.sh`	1-60 (full)	`3c88ddba012c7696a0d2344846acde05753654b7af6ee1a18c2789ee9448956b`

File	Lines read	sha256 (head)
`/Users/makinja/.claude/hooks/pre-mc-add-gate.sh`	1-72 (full)	`fa3ab6b866bfe95a73e9cb347cead87de988f7af4d8bc137407d1ab89f38ff18`
`/Users/makinja/.claude/hooks/ceo-token-origin-gate.sh`	1-219 (full)	`9374850d0f62f4ea416bbf1da0e7537263b365cedffbed654eb115dacb95686e`
`/Users/makinja/.claude/hooks/README-evidence-quality-gate.md`	1-225 (full)	`143837eca169838dff4deb949b10a963ddb86d11869af8d3794de2c0a7947185`
`/Users/makinja/.claude/settings.json`	1-474 (full)	`a4b17f07ecf402a29d26d582217dd5941fc32e931984f6b7a5f5e1bdee90345b`
`/Users/makinja/system/kernel/pi-orchestrator.js`	3380-3454 (slice)	`b71898d600a92909f26c66dcbfde07018185d7eb2fae2bc1fa6bea7973ae93ea` (sha of full file)
`/Users/makinja/.claude/session-state.md`	1-50 (slice — for context cross-refs in Section 4)	not hashed (excluded from primary source set)

Snapshot regenerated 2026-05-03 (post MC #99014/#99015/#99016 patches + MC #10313 B10 fix + MC #10611 TTL-aware Mehanik clearance).

Branch verification:

- `feat/blueprint-check-stack-aware` HEAD = `9ea69679f docs(specs): FILESTRUCTURE-BLUEPRINT §3 stack-aware allowlists update [MC #10260]` — `tools/` contains `blueprint-registry.js` and `blueprint-runner.js`, NO `blueprint-check.js`.
- `git -C ~/system show feat/blueprint-check-stack-aware:blueprint-check.js` → `fatal: path 'blueprint-check.js' does not exist in 'feat/blueprint-check-stack-aware'`.

Opaque-binary inventory:

- `~/.claude/hooks/alai-hooks` — 16,476,240 bytes, mtime 2026-05-02 23:28, no `--help` output.
- `~/.claude/hooks/claude-hooks` — 24,188,592 bytes, mtime 2026-04-10 21:19, not probed.

Evidence transcript: `/tmp/evidence-10536/sources-read.txt` (written alongside this spec).

settings.json caveat: Hash changed 2026-05-03 (MC #99014/#99015/#99016 patches). Hook wiring line refs in gate-matrix rows 2-65 (e.g., `settings.json:53`, `settings.json:233`) were NOT re-verified in this update — if hook matcher order changed, line refs may be stale. Verify on-demand via `Read ~/.claude/settings.json`.

---

8. Update history

- **2026-05-02** — Initial spec (CEO MC #10536)
- **2026-05-03** — Section 7 regenerated (post MC #99014/#99015/#99016 patches + MC #10313 B10 fix + MC #10611 TTL-aware Mehanik clearance). Gate-matrix rows 1, 10, 11, 15, 16, 17, 18, 23, 24 updated with new line refs and patch notes. See `/tmp/evidence-`

`10536-skillforge/affected-rows-audit.txt` for full audit trail.

Revision #2
Created 2026-05-03 16:15:59 UTC by John
Updated 2026-06-07 20:01:03 UTC by John