

John Drift-Prevention v3

John Drift-Prevention v3 (2026-05-02/03)

Genesis

On 2026-05-02, CEO asked "ovi hooks u kojem jeziku" (a QUESTION). John amplified this into a Kotlin port chain, opening MC #10586 and emergent MC #10589 before CEO intervened with "otisao si u rabbit hole again." This occurred despite v2 hooks (MC #10570) being live and registered. The root cause is structural: v2 gates fire at tool-invocation time (PreToolUse/PostToolUse), but the LLM's planning trajectory commits before the first gate fires. Both drift incidents (Kotlin rabbit-hole + AWS phantom earlier same day) unfolded across multiple turns, each individually compliant with v2 per-turn rules.

Context: [feedback_john_kotlin_rabbit_hole_2026-05-02.md](#)

v2 ? v3 Progression

v2 (MC #10570) provided:

- Per-turn MC counter (one-ceo-turn-mc-cap.sh) — caps mc.js add per turn
- CEO_APPROVED token rate limit — blocks John self-issuing overrides
- Turn boundary reset (mc-turn-reset.sh) — clears MC counter on UserPromptSubmit
- Depth/emergent spawn gates (john-max-depth-gate.sh) — 3 trip-wires on Task/Agent dispatch

v2 gaps:

- No intent classification before planning — QUESTION/CRITIQUE turns treated same as DIRECTIVE
- MC counter reset did NOT clear CEO_APPROVED token counter — re-issuance mechanically impossible
- No cap on Task/Agent dispatch — Kotlin incident dispatched codecraft + sentinel-tester in same turn, both uncapped

- Per-turn enforcement only — multi-turn accumulation (MC #10586 → #10589 across two turns) not detected

v3 adds:

- **Rank 1:** Fix CEO_APPROVED counter reset — now resets on every UserPromptSubmit (unconditional)
- **Rank 2:** Intent classifier at UserPromptSubmit — emits STOP reminder to LLM context before planning on QUESTION/CRITIQUE turns; blocks mc.js add unless [CEO_APPROVED]
- **Rank 3:** Task/Agent dispatch cap — per-turn limit derived from Mehanik approved_subtask_count (fallback: 1)

Three Hooks

Rank 1: mc-turn-reset.sh (UPDATED)

Path: `~/.claude/hooks/mc-turn-reset.sh`

Event: UserPromptSubmit (fires on every new CEO message)

What it does: Resets three counters to zero on every new CEO turn:

- `/tmp/john-mc-turn-counter.json` (per-turn MC count)
- `/tmp/ceo-approved-token-uses- $\{SESSION\_ID\}$ .count` (CEO_APPROVED token uses — v3 FIX)
- `/tmp/john-dispatch-turn-counter.json` (Task/Agent dispatch count — v3 NEW)

Sample output: Silent (exit 0). Counters visible via `cat /tmp/john-mc-turn-counter.json` → `{"count":0, "turn_id":"...", "ts":"..."}`

Override: None. This is a mandatory reset per documented contract ("token valid for one turn").

Rank 2: ceo-intent-classifier.sh (NEW) + pre-mc-add-gate.sh (UPDATED)

Paths:

- `~/.claude/hooks/ceo-intent-classifier.sh` (NEW, UserPromptSubmit)
- `~/.claude/hooks/pre-mc-add-gate.sh` (UPDATED, PreToolUse Bash)

Event: UserPromptSubmit (classifier) + PreToolUse Bash when tool=mc.js (gate)

What it does: Classifies CEO message intent into five classes:

- DIRECTIVE — action request (default, fail-open)
- QUESTION — inquiry (ends with ?, starts with Bosnian/English question word)
- CRITIQUE — rhetorical negative ("to ne radi", "broken", "nije ok")

- CONFIRM — short affirmative ("da", "ok", "yes")
- REISSUE — contains [CEO_APPROVED]

Writes result to `/tmp/ceo-intent-${SESSION_ID}.json`. On QUESTION or CRITIQUE, emits STOP system reminder to LLM context before planning begins. `pre-mc-add-gate.sh` blocks mc.js add on QUESTION/CRITIQUE unless [CEO_APPROVED] is in the command.

Sample output (stderr):

```
[ceo-intent-classifier] intent=QUESTION
SYSTEM REMINDER [intent-gate/QUESTION]: CEO message classified as QUESTION.
MANDATORY RESPONSE PATTERN:
1. Answer the question or acknowledge the critique in plain text.
2. Do NOT open MCs, dispatch agents, or run Mehanik in this turn.
3. If action seems warranted, surface a ONE-LINE option menu and STOP.
4. Wait for explicit CEO DIRECTIVE before any mc.js add or Task/Agent dispatch.
```

Override: Include `[CEO_APPROVED]` in the mc.js add command or Task/Agent dispatch prompt.

Rank 3: one-ceo-turn-dispatch-cap.sh (NEW)

Path: `~/.claude/hooks/one-ceo-turn-dispatch-cap.sh`

Event: PreToolUse Task|Agent

What it does: Caps Task/Agent dispatches per CEO turn. Cap derived from most recent Mehanik marker file (`/tmp/mehanik-cleared-*`) field `approved_subtask_count`. If no marker: cap=1. Reads and increments `/tmp/john-dispatch-turn-counter.json`. Blocks on count > cap unless [CEO_APPROVED] in prompt.

Sample output (stderr):

```
BLOCKED [one-ceo-turn-dispatch-cap]: dispatch #2 in this CEO turn (cap=1 from Mehanik marker).
Multi-dispatch from a single ambiguous CEO turn = drift signal (Kotlin incident 2026-05-02).
If CEO explicitly authorized N agents: include [CEO_APPROVED] in dispatch prompt.
```

Override: Include `[CEO_APPROVED]` in Task/Agent dispatch prompt, OR obtain Mehanik clearance with multi-agent plan (marker will contain `approved_subtask_count > 1`).

Live Evidence

Proveo validation: </tmp/v3-proveo-validation.md> (2026-05-02)

Suite scores:

- Suite A (Rank 1 — mc-turn-reset.sh): 4/4 PASS

- Suite B (Rank 2 — intent classifier + pre-mc-add-gate): 9/10 PASS (1 NOTE: "a" correctly defaults to DIRECTIVE per spec — not a defect)
- Suite C (Rank 3 — dispatch cap): 5/5 PASS
- Suite D (Deviations): 3/3 ACCEPT (all low-risk broadening of patterns)
- Suite E (E2E regression): 4/4 PASS

Verdict: PASS. No blocking defects. All 3 ranked recommendations correctly implemented and verified with real hook execution.

Override Tokens Table

Token	Usage	Scope	Reset
[CEO_APPROVED]	Override any single gate (MC cap, dispatch cap, intent gate, emergent spawn)	One turn	Every UserPromptSubmit (mc-turn-reset.sh)
[CEO_APPROVED_MULTIPLE_MC]	Explicitly authorize multiple MC adds in one turn (bypasses one-ceo-turn-mc-cap.sh)	One turn	Every UserPromptSubmit

Note: v3 Rank 1 fix ensures both tokens reset unconditionally on new CEO turn, making re-issuance mechanically possible (was structurally broken in v2).

Known Limitations

- **Intent classifier keyword list:** Initial set covers common Bosnian + English patterns. False-negatives possible on idiomatic phrasing. Fails open to DIRECTIVE (no regression). Tune iteratively from live incidents.
- **Rank 2 john-max-depth-gate.sh intent integration:** Spec section T1-A4 describes piggybacking intent check onto depth-gate Branch A. NOT implemented in v3. Explicitly ranked below top-3 (optional enhancement). Core Rank 1/2/3 fully functional.
- **Drift-pressure score (T2-A1):** Deferred to v3.1. Formula weights unvalidated; shipping guesses would cause false WARN floods.
- **Trip-wire 3 elevation (T2-B3):** Deferred. Root cause fix (intent classifier) is upstream; elevating downstream symptom adds CEO friction without addressing cause.

References

- **v3 spec:** ~/system/specs/john-drift-prevention-v3-spec.md
- **Parent ZAKON:** ~/system/rules/zakon-28-max-depth-boundary.md

- **v2 implementation:** MC #10570 (2026-05-02)
 - **Genesis incident:** MC #10586 + #10589 (closed parked),
[feedback_john_kotlin_rabbit_hole_2026-05-02.md](#)
-

Last updated: 2026-05-03. Owner: Skillforge. Status: Live in production.

Revision #3

Created 2026-05-02 22:40:03 UTC by John

Updated 2026-07-12 20:02:33 UTC by John