

# MC #104005 — GOTCHA Gate Degating (Code/System Tasks)

## MC #104005 — GOTCHA Gate Degating for Code/System Tasks

**Date:** 2026-06-19 **Parent:** #104003 (AI-System Rewire — Petter audit, P0→P2 program; diagnosis includes "closure overgated") **Owner:** John / CodeCraft **Status:** Implemented + verified (see evidence below)

```
$ node --check ~/system/kernel/pi-orchestrator.js && echo NODE_CHECK: PASS
NODE_CHECK: PASS
$ node ~/system/tests/gotcha-gate-decision.test.js
13 passed, 0 failed
ALL PASS
```

## Problem

Two coupled gates over-blocked **pure-code/system** tasks that have no deployed service to probe:

- Pre-spawn (pi-orchestrator.js, Step 4.55):** the `awaiting_forge` block fired for any non-`M`/non-`L` priority. The guard enumerated only `M`/`L` as "auto-stub OK", so any other value (or an unrecognised priority) fell through to the `awaiting_forge` block and stranded the task pending a manual `/prompt-forge`.
- Closure (zakon-30-direct-probe-gate.sh → mc-ready-gate.sh):** ZAKON #30 only accepted *deploy-style* probes (`curl -sI`, `gh run list`, `gcloud ...`, `sqlite3 ... SELECT`). A pure in-process JS logic change has no URL/DB to probe, so the strongest available evidence — `node --check` + a passing unit test — was not recognised, and the task could not be closed without `--force`.

# Change

## 1. Pre-spawn gate (`~/system/kernel/pi-orchestrator.js`)

- Inverted the guard: the `awaiting_forge` block now fires **only** when priority is explicitly `H` or `BLOCKER`. `M`, `L`, and any other/unrecognised value receive an auto-generated GOTCHA stub and proceed to dispatch.
- Extracted the decision into a pure, exported `gotchaGateDecision(priority) → { action: 'block' | 'stub', highStakes }`, single-sourced so it is unit-testable. The inline Step-4.55 block calls it (no duplicated logic).

## 2. Closure gate (`~/ .claude/hooks/zakon-30-direct-probe-gate.sh`)

- For **non-deploy** tasks whose `category ∈ {system, code}`, a recent `node --check`
  - passing unit test (markers `node --check`, `*.test.js`, `N passed`, `0 failed`, `ALL PASS`) counts as a valid direct probe.
- Evidence is read from the per-task bundle `/tmp/evidence-<id>/` (and, if present, legacy `bash-output-*` harness files).
- **Deploy/service tasks stay strict** — the original `curl/gh/gcloud` probe pattern is unchanged, and tasks whose title/description mention `deploy|cutover|production|cloud run|revision|curl|http(s)://` are excluded from the code-probe path.
- Hardened the file scan to capture matches into a variable with `|| true`, so a `permission-denied` during `find` traversal under `set -o pipefail` cannot corrupt the result (the original `find ... | wc -l || echo 0` could yield `"0\n0"` and throw a `[: syntax error`, silently falling through to BLOCK).

# Acceptance

Verified via the run captured in the code fence below:

```
# pre-spawn: M/L auto-stub vs H/BLOCKER block (unit test of gotchaGateDecision)
$ node ~/system/tests/gotcha-gate-decision.test.js
13 passed, 0 failed   # H/h/BLOCKER/blocker -> block; M/L/l/unknown/'/undefined/null -> stub
ALL PASS

# closure gate: code/system + passing-test evidence -> allow; absent -> block
```

A) with evidence: `exit=0` (allow, stable over 5 runs)

B) without evidence: `exit=2` (block)

# deploy/service tasks: unchanged (curl/gh/gcloud probe pattern preserved)

- M/L (and other non-H) task proceeds past GOTCHA without manual forge — auto-stub branch.
- H/BLOCKER still block `awaiting_forge`.
- `node --check` PASS; unit test 13/13 PASS.

## Evidence files

- `/tmp/evidence-104005/verification.md`
- `/tmp/evidence-104005/unit-test-output.txt`
- `~/system/tests/gotcha-gate-decision.test.js`

---

Revision #1

Created 2026-06-19 20:12:56 UTC by John

Updated 2026-06-19 20:12:56 UTC by John