

Spawn Gate Node-Side Parity (MC #10548)

Spawn Gate Node-Side Parity (MC #10548)

Context — Why This Exists

Problem: Pi-orchestrator spawns agents internally at Step 4.6 (~line 4291 in `pi-orchestrator.js`) without going through Claude Code's Task dispatch path. This meant PreToolUse Bash hooks (`~/.claude/hooks/pre-dispatch-gate.sh`) never fired, creating a bypass where internal spawns skipped Mehanik clearance verification.

Solution: MC #10548 implemented Node-side spawn gate parity — a JavaScript enforcement layer (`~/system/kernel/spawn-gate.js`) that mirrors all 9 checks from the Bash gate, called directly by pi-orchestrator before every agent spawn.

Genesis: Pi-orchestrator hardening Talas 2 (parent thread #10043 reform). Dependency on `δ` #10551 `worktree_company_enforcer.js` (completed).

Architecture — Dual Gate System

Gate	Location	Lines	Trigger
Bash Gate	<code>~/.claude/hooks/pre-dispatch-gate.sh</code>	163	Claude Code Task/Agent dispatches (PreToolUse hook)
Node Gate	<code>~/system/kernel/spawn-gate.js</code>	853	Pi-orchestrator internal spawns (Step 4.6)

Shared enforcement: Both gates implement the same 9-check validation sequence. Checks 1-3 existed in `spawn-gate.js` before this MC; checks 4-9 were added to achieve full parity.

The 6 New Checks (MC #10548 Scope)

Check 4: Marker TTL (`checkMarkerTTL`, line 143)

Verifies Mehanik clearance has not expired. Reads `expires_at` field from `/tmp/mehakin-cleared-{taskId}` marker and compares to current time. Rejects if expired or missing.

Check 5: Marker Schema (`checkMarkerSchema`, line 175)

Validates all 22 required marker fields are present:

- `task_id`, `timestamp`, `expires_at`
- `project_path`, `blueprint_read`, `deploy_map_read`, `deploy_path_summary`
- `ceo_item_count`, `approved_subtask_count`, `scope_ceiling_compliance`
- `estimated_cost_usd`, `cost_approval_required`, `skill_audit_summary`
- `blueprint_score`, `blueprint_threshold_applied`, `risk_level`
- `hallucination_surface`, `resource_lock_state`, `cross_product_check`
- `tool_contract_check`, `agent_registry_check`, `final_decision`

Check 6: Scope Ceiling (`checkScopeCeiling`, line 196)

Deterministic arithmetic enforcement: `approved_subtask_count ≤ ceo_item_count + 2`. Prevents scope creep beyond Mehanik-approved ceiling.

Check 7: Tool Contract (`checkToolContract`, line 226)

Research-class agent dispatches (`datavera`, `sentinel-*`) must include a `TOOL_CONTRACT:` block in the prompt. Exempts prompts with tool literal references (`discover.js`, `lightrag.js`, `mc.js`, `web-search.sh`) as implementation context, not research requests.

Fix on failure:

```
node ~/system/tools/wrap-with-tool-contract.js --agent research --tools web-search.sh
```

Check 8: Agent Registry (`checkAgentRegistry`, line 258)

Agent slug must exist in `~/system/agents/specialist-mapping.json`. Bootstrap-exempt agents skip this check:

- mehanik
- devils-advocate
- validator

Check 9: Blueprint Advisory (`checkBlueprintAdvisory`, line 311)

WARN-ONLY, FAIL-OPEN. Checks `blueprint_score < blueprint_threshold_applied`. On failure, writes warning to `/tmp/spawn-gate-warnings.log` and stderr, but does NOT block dispatch. Bypassed by `[CEO_OVERRIDE]` in prompt.

Wiring — Pi-Orchestrator Integration

Location: `~/system/kernel/pi-orchestrator.js` lines 4291-4316 (Step 4.6)

```
// Step 4.6: AAOS Spawn Gate – full 9-check parity with pre-dispatch-gate.sh (MC #10548)
if (_aaosSpawnGate && typeof _aaosSpawnGate.runGate === 'function') {
  const _taskPromptContext = `${task.title || ''} ${task.description || ''}`;
  const gateResult = await _aaosSpawnGate.runGate(task.id, 'pi-orchestrator', task.priority || 'M');

  if (!gateResult.allowed) {
    const priority = (task.priority || 'M').toUpperCase();
    if (priority === 'H' || priority === 'M') {
      log('error', `Task #${task.id} SPAWN GATE BLOCKED: ${gateResult.reason}`, { taskId: task.id });
      execFileSync('node', [MC_SCRIPT, 'block', String(task.id), `spawn-gate: ${gateResult.reason}`]);
      return; // HARD BLOCK – do not proceed to spawn
    }
    log('warn', `Task #${task.id} SPAWN GATE warned (L-priority): ${gateResult.reason}`, { taskId: task.id });
  }
}
```

On FAIL (H/M priority):

1. Log error
2. Call `mc.js block <taskId> "spawn-gate: {reason}"`
3. Return early (no agent spawn)

On FAIL (L priority): Log warning, proceed with spawn.

Fallback: Lines 4316+ preserve legacy `check()` path for compatibility if `runGate()` is unavailable (pre-MC#10548 deployments).

Current State

Status: Code deployed COLD. Pi-orchestrator daemon is STOPPED.

Activation: Requires MC #10542 reactivation go/no-go decision (separate CEO approval, ZAKON PI2 verification pending).

Test Plan

Test suite: `~/system/tests/spawn-gate.test.js` (23 tests)

Coverage: Each new check has PASS + FAIL path tests.

Run tests:

```
node --test ~/system/tests/spawn-gate.test.js
```

Expected output: 23 PASS, 0 FAIL

Validation

Proveo verdict: `/tmp/proveo-10548-spawn-gate-validation.json` — PASS 6/6 AC

Related Documentation

- [Pi-Orchestrator Operations](#)
- MC #10548 task record
- Parent reform thread: MC #10043 (Pi-orch hardening Talas 2)
- ZAKON PI2: Deploy verification protocol

Revision #2

Created 2026-05-04 10:37:44 UTC by John

Updated 2026-06-07 20:01:07 UTC by John