

# ZAKON Registry — Creation Requires Approval Gate

# ZAKON Registry — Creation Requires Approval Gate

**Meta:** MC #100464 (Track 5d) | CEO Board 2026-05-12 | Devil's Advocate Alternative | v1.0

## Genesis

The ZAKON Registry was created as the **Devil's Advocate Alternative** during MC #99911 CEO Board deliberation on 2026-05-12. It addresses the root concern: "Who watches the watchers?" — ensuring no agent (including Skillforge) can create new ZAKON rule files without explicit CEO approval.

**Board Endorsement:** All 5 Board members (CTO, CFO, COO, CMO, Devil's Advocate) endorsed the Registry concept as a necessary complement to enforcement hooks.

**Design Principle:** Fail-closed. If registry is missing or unparseable, all ZAKON writes are blocked with explicit fix instructions.

## What It Does

The ZAKON Registry is a JSON-based ledger (`~/system/rules/zakon-registry.json`) that acts as a creation gate for all ZAKON rule files (`~/system/rules/zakon-*.md`).

**Enforcement:** Pre-write hook (`blueprint-schema-validator-pre.sh`) calls `zakon-registry-check.js validate` before any write to `zakon-*.md` files.

### Exit Codes:

- `0` — PASS: File has approved registry entry

- 2 — BLOCK: File not registered OR status not approved OR missing [CEO\_APPROVED] token
  - 3 — BLOCK: Registry file missing/unparseable (fail-closed behavior)
- 

# Registry Schema

```
{
  "version": "1.0",
  "description": "Registry of all ZAKON rule files...",
  "policy": {
    "creation_gate": "Any write to ~/system/rules/zakon-*.md requires entry with
status='approved-pending-author' or 'approved-live'.",
    "ceo_approval_token": "Literal string [CEO_APPROVED] must appear in matching MC task.",
    "fail_closed": "If registry missing/unparseable, BLOCK with explicit fix command.",
    "hook_integration": "blueprint-schema-validator-pre.sh must call: node
~/system/tools/zakon-registry-check.js validate $FILE_PATH"
  },
  "backfill_metadata": {
    "scan_date": "2026-05-12",
    "scan_path": "~/system/rules/zakon-*.md",
    "files_found": 3,
    "notes": "All pre-2026-05-12 ZAKONs grandfathered as legacy-pre-registry."
  },
  "registry": [
    {
      "zakon_id": "feasibility-check",
      "file_path": "~/system/rules/zakon-feasibility-check.md",
      "mc_task": null,
      "ceo_approved_token": "GRANDFATHERED-PRE-2026-05-12",
      "status": "legacy-pre-registry",
      "backfill_metadata": { ... }
    },
    ...
  ]
}
```

---

# Tool Usage

## Validate (Hook Integration)

```
node ~/system/tools/zakon-registry-check.js validate ~/system/rules/zakon-example.md
```

**Exit Codes:** 0 = pass, 2 = blocked, 3 = registry error

**Hook Integration:** `blueprint-schema-validator-pre.sh` line ~75:

```
if [[ "$FILE" =~ ~/system/rules/zakon-.*\.md$ ]]; then
  node "$HOME/system/tools/zakon-registry-check.js" validate "$FILE" || exit 2
fi
```

## List All Entries

```
node ~/system/tools/zakon-registry-check.js list
```

**Output:** Human-readable list of all registry entries with status, MC task, and approval token.

## Statistics

```
node ~/system/tools/zakon-registry-check.js stats
```

**Output:** Count of entries by status (legacy-pre-registry, active, approved-pending-author, etc.).

## Current Registry State

As of 2026-05-12:

ZAKON ID	Status	MC Task	Approval Token
feasibility-check	legacy-pre-registry	N/A	GRANDFATHERED-PRE-2026-05-12
pi2-deploy-verification	legacy-pre-registry	N/A	GRANDFATHERED-PRE-2026-05-12
qa19-mapping	legacy-pre-registry	N/A	GRANDFATHERED-PRE-2026-05-12

ZAKON ID	Status	MC Task	Approval Token
blueprint-enforcement	active	99911	[CEO_APPROVED]

**Total Entries:** 4 (3 grandfathered legacy + 1 newly created via registry gate)

---

# Backfill Manifest

On 2026-05-12, a backfill scan identified **3 pre-existing ZAKON files** in `~/system/rules/`:

- `zakon-feasibility-check.md` — 84 lines, 3997 bytes
- `zakon-pi2-deploy-verification.md` — 165 lines, 6412 bytes (referenced in CLAUDE.md)
- `zakon-qa19-mapping.md` — 268 lines, 13811 bytes

**Grandfathering Policy:** All 3 files registered as `legacy-pre-registry` status with `GRANDFATHERED-PRE-2026-05-12` token. This is an **audit snapshot**, NOT a CEO approval retroactively applied. Future edits to these files are allowed without re-approval (legacy status).

---

# Adding New ZAKON Files

## Process:

- Create MC Task:** Title must include "ZAKON" or "rule". Description must contain `[CEO_APPROVED]` token.
- Update Registry:** Add entry to `~/system/rules/zakon-registry.json` with:
  - `zakon_id` — Short identifier (e.g., "cost-ceiling")
  - `file_path` — Full path with tilde notation
  - `mc_task` — MC task ID
  - `ceo_approved_token` — Must be `[CEO_APPROVED]`
  - `status` — `approved-pending-author`
- Author ZAKON File:** Write hook will validate against registry. If entry exists with approved status, write proceeds.
- Update Status:** After file is authored and verified, update registry entry to `status: "active"` and add `published_sha256`.

## Example Registry Entry:

```
{
  "zakon_id": "cost-ceiling",
  "file_path": "~/system/rules/zakon-cost-ceiling.md",
  "mc_task": 100500,
```

```
"ceo_approved_token": "[CEO_APPROVED]",
"ceo_approval_date": "2026-05-13",
"ceo_approval_method": "CEO Board deliberation (MC #100500)",
"status": "approved-pending-author",
"notes": "Cost ceiling enforcement rule for multi-week projects"
}
```

## Fail-Closed Behavior

If `zakon-registry.json` is missing or unparseable, the validation tool exits with code **3** and provides explicit fix instructions:

```
ZAKON_REGISTRY_ERROR: Registry file not found.
Expected: /Users/makinja/system/rules/zakon-registry.json
FIX: Create registry via MC #100464 or restore from backup.
```

**Design Rationale:** Fail-closed prevents silent bypass. If registry infrastructure is broken, ALL ZAKON writes are blocked until registry is restored.

## Hook Integration Details

**Hook File:** `~/.claude/hooks/blueprint-schema-validator-pre.sh`

**Integration Point:** After detecting `zakon-*.md` file pattern, hook calls:

```
node "$HOME/system/tools/zakon-registry-check.js" validate "$FILE"
EXIT_CODE=$?
if [ $EXIT_CODE -ne 0 ]; then
    exit 2 # Block write
fi
```

**Registration:** `~/.claude/settings.json` PreToolUse hook for `Write|Edit|MultiEdit` actions.

**Timing:** PreToolUse timing ensures disk write is blocked **before** tool executes. PostToolUse cannot block writes (correction signal only).

# Related Documentation

- [ZAKON #18B — Blueprint Liveness Enforcement](#)
- MC #99911 — FAZA 4 enforcement genesis (CEO Board deliberation)
- MC #100464 — Track 5d (Registry gate implementation)
- ADR-026 — Hook architecture (PreToolUse vs PostToolUse timing)

---

**Registry Location:** `~/system/rules/zakon-registry.json`

**Tool Location:** `~/system/tools/zakon-registry-check.js`

**Hook Integration:** `~/ .claude/hooks/blueprint-schema-validator-pre.sh`

**Version:** 1.0

**Current Entries:** 4 (3 grandfathered + 1 active)

**Published:** 2026-05-12

---

Revision #2

Created 2026-05-12 19:34:48 UTC by John

Updated 2026-06-14 20:03:12 UTC by John