

ZAKON PLAN Linter

Runbook: ZAKON PLAN Linter

Owner: Proveo

File: `~/system/tools/zakon-plan-lint.sh`

Version: 1.0.0

Last Updated: 2026-04-16

Purpose

Enforce ZAKON PLAN compliance: every plan MUST include validation task (Proveo/Angie) and documentation task (Skillforge/BookStack). This is a non-negotiable Hard Constraint from

`~/ .claude/CLAUDE.md`.

Problem solved: Plans were frequently shipped without these mandatory tasks. This linter makes compliance technical, not voluntary.

How It Works

The linter scans a plan file (Markdown) for:

1. Validation task indicators:

- Owner field containing: `Proveo`, `Angie`, `angie-jones`, `V1` (validator role)
- Task description containing: `validation`, `end-to-end`, `evidence`, `Proveo sign-off`

2. Documentation task indicators:

- Owner field containing: `Skillforge`, `D1` (docs role)
- Task description containing: `documentation`, `BookStack`, `runbook`

Detection method: Pattern matching with case-insensitive regex.

Exit codes:

- `0` — Plan compliant (both tasks found)
- `1` — Plan missing validation task OR docs task

- 2 — File not found or not readable
-

Usage

Command-Line

```
bash ~/system/tools/zakon-plan-lint.sh <path-to-plan.md>
```

Examples:

```
# Check a single plan
bash ~/system/tools/zakon-plan-lint.sh ~/system/specs/system-evolution-plan.md
# Output: [ ] ZAKON PLAN COMPLIANT

# Check a non-compliant plan
bash ~/system/tools/zakon-plan-lint.sh ~/system/specs/old-plan.md
# Output: [ ] MISSING validation task
# Exit code: 1

# Use in CI/validation script
if bash ~/system/tools/zakon-plan-lint.sh ~/system/specs/my-plan.md; then
    echo "Plan approved"
else
    echo "Plan rejected – add validation + docs tasks"
    exit 1
fi
```

Pre-Commit Hook (Recommended)

Automatically check plans before committing to git:

Setup:

```
# Create pre-commit hook
cat > ~/ALAI/.git/hooks/pre-commit << 'EOF'
#!/bin/bash
# ZAKON PLAN linter – runs on all staged *-plan.md files
```

```

STAGED_PLANS=$(git diff --cached --name-only --diff-filter=ACM | grep 'specs/.*-plan\.md$')

if [ -z "$STAGED_PLANS" ]; then
    exit 0 # No plans staged, skip check
fi

FAILED=0
for PLAN in $STAGED_PLANS; do
    if [ -f "$PLAN" ]; then
        if ! bash ~/system/tools/zakon-plan-lint.sh "$PLAN"; then
            echo "❑ $PLAN is not ZAKON PLAN compliant"
            FAILED=1
        fi
    fi
done

if [ $FAILED -eq 1 ]; then
    echo ""
    echo "Fix: Add validation task (Proveo/Angie) and documentation task (Skillforge) to your
plan."
    exit 1
fi
EOF

chmod +x ~/ALAI/.git/hooks/pre-commit

```

Now every commit with a plan file will be checked automatically.

Output Format

Compliant Plan

```

❑ ZAKON PLAN COMPLIANT: /Users/makinja/system/specs/system-evolution-plan.md
[✓] Validation task found (owner: V1, task: end-to-end evidence run)
[✓] Documentation task found (owner: D1, task: BookStack runbooks)

```

Non-Compliant Plan

- ❑ ZAKON PLAN VIOLATION: /Users/makinja/system/specs/my-plan.md
- [x] MISSING: Validation task (must include Proveo/Angie owner)
- [x] MISSING: Documentation task (must include Skillforge owner)

Required:

- Add task with owner: Proveo / Angie / V1 (validation role)
- Add task with owner: Skillforge / D1 (docs role)

What Gets Detected

Validation Task Keywords

- **Owner patterns:** Proveo, Angie, angie-jones, V1, validator
- **Description patterns:** validation, end-to-end, evidence, E2E, Proveo sign-off, quality gate

Example (PASS):

```
**Task 14 (VALIDATION – MANDATORY):** End-to-end evidence run
- Owner: V1 (Angie Jones / Proveo)
- Acceptance:
  - [ ] Synthetic task validated with real evidence
  - [ ] Proveo sign-off in MC task
```

Documentation Task Keywords

- **Owner patterns:** Skillforge, D1, docs, documentation
- **Description patterns:** documentation, BookStack, runbook, docs, indexed to LightRAG

Example (PASS):

```
**Task 15 (DOCS – MANDATORY):** BookStack runbooks
- Owner: D1 (Skillforge)
- Acceptance:
  - [ ] BookStack page created
  - [ ] Runbooks published
```

When to Use

Always Run For:

- New plans in `~/system/specs/`
- Plans attached to MC tasks with `--plan-ref` field
- Pre-commit hooks on any `*-plan.md` file
- CI/CD validation gates

Integration Points:

1. **mc.js add with plan:** `mc.js add --plan-ref <path>` should auto-lint
2. **Pre-commit hook:** Git commit of plan files
3. **Weekly regression suite:** `system-regression.sh` scans all plans (max 10)
4. **Manual review:** John reviewing plan before CEO approval

Bypass Procedure (EMERGENCY ONLY)

You should NEVER bypass this linter. If a plan fails, fix the plan.

However, in true emergency (production incident, CEO direct override):

1. **Skip pre-commit hook:**

```
git commit --no-verify -m "Emergency deploy"
```

2. **Forced MC task creation:**

```
# There is NO bypass flag – linter is read-only check  
# Fix the plan or discuss with Petter Graff
```

3. **Rationale:** Bypassing means shipping a plan that will fail at Phase 4. Builder will finish → no validator → no docs → system regresses. CEO Hard Constraint #2: "No claim without evidence."
-

Troubleshooting

Linters Says Missing But Task Exists

Symptom: You added validation task, but linter still rejects.

Fix: Ensure keywords are in plaintext (not code blocks) and match patterns:

```
# WRONG (in code block)
`Owner: Angie`

# CORRECT (plain Markdown)
Owner: Angie Jones (Proveo)
```

Linters Detects Wrong Section as Task

Symptom: Linter passes but you didn't add tasks — it detected something in background context.

Fix: This is edge case but acceptable. If pattern appears in "Related Work" section, manual review will catch it. Linter is first gate, not only gate.

What If Plan is Too Small for Validation?

Answer: Every plan needs validation, even tiny ones. Scale validation to match:

- **Tiny plan (1 task):** Validation = smoke test + evidence screenshot
- **Medium plan (5 tasks):** Validation = integration test + log capture
- **Large plan (15 tasks):** Validation = end-to-end suite + evidence bundle

Documentation also scales:

- **Tiny plan:** Update existing BookStack page
- **Large plan:** Create new runbooks

Hard rule: Size doesn't exempt. Every plan enriches system knowledge.

Integration with Other Gates

flowchart TD

A[Plan Draft] -->|ZAKON linter| B{Compliant?}

B -->|No| C[Reject: Fix Plan]

B -->|Yes| D[Git Commit]

D -->|Pre-commit hook| E[Linter re-runs]

E -->|Pass| F[Plan in Repo]

F --> G[Builder Starts]

G --> H[Task Done]

H -->|Proveo Gate| I{Evidence?}

I -->|No| J[Reject: Need Validation]

I -->|Yes| K[Task Complete]

K --> L[Skillforge Docs]

L --> M[Plan Fully Compliant]

style B fill:#fff3cd

style I fill:#fff3cd

style M fill:#e1f5e1

Three gates work together:

1. **ZAKON linter:** Checks plan structure (this runbook)
2. **Proveo gate:** Checks task evidence before `mc.js done` (see `~/system/docs/runbooks/mc-done-proveo-gate.md`)
3. **Blueprint liveness:** Checks blueprint mtime (see `~/system/docs/runbooks/blueprint-liveness.md`)

All three enforce Hard Constraints from `~/claude/CLAUDE.md`.

Exit Codes Reference

Code	Meaning	Action
0	Plan compliant	Proceed
1	Missing validation or docs task	Fix plan
2	File not found / not readable	Check path

Maintenance

Adding New Keywords

If agents start using new terms (e.g., "QA task" instead of "validation task"), update patterns:

File: `~/system/tools/zakon-plan-lint.sh`

```
# Validation keywords (case-insensitive grep)
VALIDATION_PATTERNS="Proveo|Angie|angie-jones|V1|validator|validation|QA"

# Documentation keywords
DOCS_PATTERNS="Skillforge|D1|documentation|BookStack|runbook|docs"
```

Test after change:

```
bash ~/system/tools/zakon-plan-lint.sh ~/system/specs/system-evolution-plan.md && echo PASS
```

False Positives Log

Track patterns that incorrectly pass/fail:

Location: `~/system/logs/zakon-linter-audit.jsonl`

```
{"timestamp": "2026-04-20T10:30:00Z", "plan": "my-plan.md", "issue": "Detected 'validation' in
unrelated section", "severity": "low"}
```

Review monthly. If >5 false positives/month, refine regex.

Examples

Example 1: Minimal Compliant Plan

```
# Plan: Add New Feature

## Tasks

**Task 1:** Implement feature X
- Owner: CodeCraft
```

- Acceptance: Feature works

****Task 2 (VALIDATION):**** E2E test

- Owner: Angie Jones

- Acceptance: Evidence bundle in ~/system/evidence/

****Task 3 (DOCS):**** BookStack page

- Owner: Skillforge

- Acceptance: Page published at docs.alai.no

Linter result: PASS

Example 2: Non-Compliant Plan (Missing Docs)

```
# Plan: Quick Fix
```

```
## Tasks
```

```
**Task 1:** Fix bug
```

```
- Owner: CodeCraft
```

```
**Task 2:** Test fix
```

```
- Owner: Proveo
```

Linter result: FAIL — Missing documentation task

Example 3: Tricky But Valid

```
# Plan: System Upgrade
```

```
## Phase 1
```

```
Tasks 1-10 by various builders...
```

```
## Phase 2: MANDATORY Validation + Docs
```

```
**Task 11 (VALIDATION):** End-to-end validation
```

```
- Owner: V1 (Proveo)
```

****Task 12 (DOCS):**** Update runbooks

- Owner: D1 (Skillforge)

Linters result: PASS (tasks can be anywhere in file, not just at end)

Related Documentation

- **CLAUDE.md ZAKON PLAN section:** [~/ .claude/CLAUDE.md](#) (search "ZAKON PLAN")
- **System evolution upgrade:** [~/system/docs/system-evolution-2026-04-16.md](#)
- **Proveo gate:** [~/system/docs/runbooks/mc-done-proveo-gate.md](#) (upcoming)
- **Blueprint liveness:** [~/system/docs/runbooks/blueprint-liveness.md](#)

Changelog

Date	Version	Change
2026-04-16	1.0.0	Initial implementation (MC #8020 Task 8)

Questions? Contact Petter Graff (team lead) or read [~/system/rules/john-operating-system.md](#) section on ZAKON PLAN.

Revision #4

Created 2026-04-16 21:43:59 UTC by John

Updated 2026-06-21 20:02:47 UTC by John