

# BUILD-BLUEPRINT Discipline

## 2.3 — BUILD-BLUEPRINT Discipline Audit

**Date:** 2026-05-09 **Auditor:** sentinel-ba **Scope:** 17 BUILD-BLUEPRINT.md files + Mehanik gate enforcement

### 1. Per-Blueprint State Matrix

#	Path	Bytes	Lines	Last Modified	Status	Project Liveness
1	~/projects/interナル/basicfakta/BUILD-BLUEPRINT.md	11,193	323	2026-04-29	SUBSTANTIAL	Last commit 10d ago (auto-backup only)
2	~/projects/boostack-api/BUILD-BLUEPRINT.md	12,366	352	2026-04-29	SUBSTANTIAL	Last commit 5 weeks ago (auto-backup)
3	~/projects/pa/BUILD-BLUEPRINT.md	13,238	354	2026-04-29	SUBSTANTIAL	Last commit 10d ago (auto-backup)
4	~/projects/ala-system/BUILD-BLUEPRINT.md	3,520	75	2026-04-30	THIN (75 lines, not stub)	Last commit 6d ago (auto-backup)
5	~/business/.../products/Tok/BUILD-BLUEPRINT.md	27,080	637	2026-04-27	SUBSTANTIAL	Last commit 10d ago — gradle-wrapper CI fix; active
6	~/business/.../products/BasicFakta/BUILD-BLUEPRINT.md	12,865	332	2026-03-07	STALE (63d, no recent activity)	Last commit 9 weeks ago — test/CI only

#	Path	Bytes	Lines	Last Modified	Status	Project Liveness
7	~/business/... /products/Lobby/BUILD-BLUEPRINT.md	18,707	396	2026-03-09	STALE (61d, repo semi-active)	Last commit 6 weeks ago — feat/RLS
8	~/business/... /products/Drop/BUILD-BLUEPRINT.md	8,846	208	2026-05-07	PRESENT (208 lines, recently updated)	Last commit 63 min ago — MOST ACTIVE
9	~/business/... /products/DropSrbija/BUILD-BLUEPRINT.md	10,657	386	2026-05-08	SUBSTANTIAL	Last commit 2d ago; git-repo shared with Gotiva (anvil-fs migration)
10	~/business/... /products/Plouk/BUILD-BLUEPRINT.md	24,175	512	2026-04-16	STALE (23d, repo dormant)	Last commit 5 weeks ago — smoke tests only
11	~/business/... /products/Gotiva/BUILD-BLUEPRINT.md	27,112	556	2026-03-11	STALE (59d)	Last commit 2d ago was chore/anvil-fs (migration commit, not product work)
12	~/business/... /products/Bilko/BUILD-BLUEPRINT.md	38,303	530	2026-05-08	SUBSTANTIAL	Last commit 10 min ago — extremely active
13	~/business/... /sales/outreach/sintef/BUILD-BLUEPRINT.md	1,943	49	2026-04-27	TEMPLATE/STUB (49 lines, 1,943 bytes — under threshold)	Last commit 2d ago was chore/anvil-fs only
14	~/business/... /web/BUILD-BLUEPRINT.md	4,636	110	2026-04-27	THIN	Last commit 2d ago — feat/redirect
15	~/business/... /finance/akershus-fylke/BUILD-BLUEPRINT.md	1,486	33	2026-05-08	TEMPLATE/STUB (33 lines; per MC #99886 Decision 7: "move akershus OUT of products/")	Last commit 2d ago chore only

#	Path	Bytes	Lines	Last Modified	Status	Project Liveness
16	~/clients-external/snowit-site/BUILD-BLUEPRINT.md	3,427	67	2026-04-28	THIN	Last commit 2 hours ago — active gitignore hygiene
17	~/clients-external/lumiscare-variants/lumiscare/BUILD-BLUEPRINT.md	37,426	637	2026-05-09	SUBSTANTIAL	Last commit 2 hours ago — security fix; MOST RECENTLY UPDATED

## Summary counts

- **SUBSTANTIAL** (>10,000 bytes, real content): 8 — basicfakta, bookstack-api, pa, Tok, DropSrbija, Gotiva, Bilko, lumiscare
- **PRESENT / ADEQUATE** (200-10,000 bytes, real content): 2 — Drop, alai-system
- **THIN** (< 5,000 bytes, functional but sparse): 3 — web, snowit-site, alai-system
- **TEMPLATE/STUB** (< 2,000 bytes or <50 lines with no real content): 2 — sintef, akershus-fylke
- **STALE** (>30d without update, repo active): 4 — BasicFakta (63d), Lobby (61d), Gotiva (59d), Plock (23d)

**Note:** STALE classification applies where the product repo has had meaningful commits but the blueprint has not been updated. Plock is borderline (23d, repo dormant).

## 2. Mehanik Gate Truth Check

What Mehanik requires (tool-verified from

~/ .claude/agents/mehanik.md)

Phase T of the GOTCHA workflow states:

1. `ls {project_path}/BUILD-BLUEPRINT.md` — MUST exist
2. Read the file (confirm contents match task scope)
3. **Circuit Breaker #2:** "BUILD-BLUEPRINT.md not read — evidence of Read call required in session"

**Assessment: The requirement is FORMALLY A HARD BLOCK.** CB#2 fires if the blueprint is not read (not just present). The hook `~/claude/hooks/pre-dispatch-gate.sh` also enforces a secondary check: it runs `blueprint-check.js` against the project path stored in the Mehanik cleared token and blocks dispatch if score < 60.

## Enforcement quality issues identified

**Issue A — Hook is warn-only for missing MC ID.** When the Task prompt has no `MC #NNNN` pattern, the hook exits 0 with a stderr warning only. Tasks dispatched without an MC ID bypass both the Mehanik cleared-token check and the blueprint-score gate entirely.

**Issue B — `mehanic_session_id: unknown` in all inspected tokens.** Both tokens inspected (99886 and 100150) show `mehanic_session_id: unknown`. The cleared token was written, proving Mehanik ran, but the session binding is absent — meaning the hook cannot verify that the *same session* cleared the task vs. a stale token from a prior session. Token expiry (4h) partially mitigates but does not eliminate this gap.

**Issue C — Blueprint score threshold set at 90 but tokens show WARN at 80 and 65.** Both inspected dispatches show `blueprint_check_result: WARN` with scores below the 90 threshold, yet dispatch proceeded. The hook's `blueprint-check.js` integration exists (`~/system/tools/blueprint-check.js` is present), but the pre-dispatch hook only exits 2 (block) if verdict is `NOT_READY`. The WARN path allows dispatch. The 90-point threshold in the token file is never enforced as a gate.

**Issue D — Token expiry not enforced in hook.** The hook does not parse `expires_at` from the cleared file. A token written 23 hours ago (within a session restart) would still pass. The 4h expiry in the token is advisory metadata only.

## Sample of 5 recent dispatches

MC ID	Cleared token exists?	Blueprint cited in token?	Blueprint score	Dispatch allowed?
99886	YES	Bilko/BUILD-BLUEPRINT.md	80 (WARN)	YES — WARN not blocked
100150	YES	Drop/BUILD-BLUEPRINT.md	65 (WARN)	YES — WARN not blocked
100150	YES	Drop/DEPLOY-MAP.md cited	—	YES
99910 (MC Claim Protocol)	YES ( <code>/tmp/mehanic-cleared-99910</code> )	—	Not inspectable (token may have expired and been overwritten)	YES

MC ID	Cleared token exists?	Blueprint cited in token?	Blueprint score	Dispatch allowed?
99886	YES	Bilko — per DOD evidence: "Mehanik CLEAR /tmp/mehanik-cleared-99886"	80	YES

Token count in `/tmp`: **113 mehanik-cleared tokens** present (range: #10063 to #100173). Volume indicates Mehanik is running regularly — it is not being bypassed entirely.

**Gate verdict: PARTIALLY REAL.** Blueprint presence is hard-blocked. Blueprint read is required and recorded in the token. However, the score-based quality gate (threshold 90) is advisory — WARN scores pass. The session-binding gap means cleared tokens could theoretically be reused across sessions. The missing-MC-ID path is a complete bypass vector.

## 3. Blueprint-vs-Reality Drift Score

### Bilko (MOST ACTIVE)

#### Blueprint claims:

- "API Framework: Ktor 3.4.0 / Kotlin 2.3.0 on JVM 25" — Cloud Run deployed
- "Database: PostgreSQL 15" — Cloud SQL
- "Status: MVP dev — frontend implemented with mock data, backend built"

#### Actual state (tool-verified):

- `gcloud run services list` shows: `bilko-api-stage`, `bilko-api-demo`, `bilko-web-stage`, `bilko-web-demo`, `bilko-intesa-demo` all TRUE; `bilko-staging-api` FALSE (unhealthy)
- Drop is on Azure VM; Bilko is on GCP Cloud Run — consistent with blueprint claim
- Blueprint says "Status: MVP dev" but there are 5 live Cloud Run services including `bilko-intesa-demo` (suggesting Intesa bank integration demo exists)

**Drift score: LOW-MEDIUM.** Infrastructure matches. The "MVP dev with mock data" status language is understated given live deployed services. Blueprint was last updated 2026-05-08 (yesterday) — reasonably current.

### Drop (MOST RECENTLY COMMITTED)

#### Blueprint claims:

- "Azure VM `vm-drop-prod` (Sweden Central)" + docker-compose

- "Database: PostgreSQL 16 via Drizzle ORM in docker-compose on Azure VM"

#### Actual state (tool-verified):

- `curl -sI https://app.getdrop.no` returns HTTP/2 200 — production is live
- Response headers show `nonce`-based CSP (Next.js pattern) — consistent with Next.js 15 claim
- Blueprint was rewritten 2026-04-30 to fix the AWS phantom; it now correctly reflects Azure VM
- Most recent commit (63 min ago): staging CI/CD OIDC fix — blueprint does NOT mention staging VM yet (deploy token shows `vm-drop-stage` staging path)

**Drift score: LOW.** Production deployment matches blueprint. Staging environment exists in deployment reality but blueprint only covers production — minor documentation lag.

## Tok (ACTIVE BUT NO RECENT BLUEPRINT UPDATE)

#### Blueprint claims:

- "Database: PostgreSQL 15 (Cloud SQL)"
- "PSD2 Cert: QWAC/QSEAL — DigiCert/GlobalSign — mTLS for Croatia"
- "Status: Core implementation complete — all 8 development gates DONE"

#### Actual state:

- No `gcloud run services list` results for Tok (not visible in current GCP project scope)
- Blueprint last updated 2026-04-27 (12d ago); last meaningful commit was 10d ago (gradle-wrapper fix unblocking CI since March)
- The gradle-wrapper CI was broken since March 2026 — meaning "all 8 gates DONE" may be technically true for code but CI was broken for 6+ weeks

**Drift score: MEDIUM.** The product-gate claim is technically accurate but CI was silently broken for 2+ months — a fact not reflected in the blueprint status line. PSD2 cert claim is unverifiable without SSH to the Tok deployment.

---

## 4. Cross-Cutting Findings

### No holding-company blueprint

`~/business/ALAI-Holding-AS/BUILD-BLUEPRINT.md` — ABSENT. There is no top-level document explaining how the portfolio of products relates, shared infrastructure, or cross-product dependencies (e.g., Tok feeding Bilko). Each product is an island. This is a gap for new agents onboarding to the system who need portfolio-level context.

## Blueprint versioning

Blueprints ARE git-tracked in their respective product repos. `git log --follow -- BUILD-BLUEPRINT.md` on Bilko shows at least 3 tracked commits; Drop shows the AWS-to-Azure canonical rewrite is a committed event with a clear commit message and MC reference. This is genuine version history — drift can be diagnosed by diffing commits.

However, there is no automated drift alert. Blueprint age vs. commit recency is never surfaced to John or CEO unless a sentinel audit runs manually.

## Tenants without blueprints

- `~/system/` — has `~/system/BUILD-BLUEPRINT.md` (EXISTS — confirmed)
- `~/personal/` — NO BLUEPRINT (expected: personal scope, not a product)
- `~/clients-external/` — only `snowit-site` and `lumiscare` are covered; MEDON client (`~/business/ALAI-Holding-AS/pipeline/CodeCraft/clients/MEDON/`) has a `CHANGELOG.md` in its shopify-app but NO `BUILD-BLUEPRINT.md`. This is a Mehanik bypass vector for any MEDON dispatch.
- `DropSrbija` blueprint exists but the Gotiva blueprint is 59d stale — yet the git repo for both was recently touched (anvil-fs migration). This creates a false "recently updated" signal.

## CHANGELOG without BUILD-BLUEPRINT

Within active project trees (excluding `node_modules`): MEDON shopify-app has a `CHANGELOG.md` without a blueprint. All `node_modules CHANGELOG.md` hits are false positives (dependency changelogs, not ALAI products).

---

# 5. Blueprint ? Mehanik ? Agent Dispatch Trace: MC #99886

**Task:** CI/CD Standardization — FAZA 2 — canonical refresh (Petter Graff)

**Mehanik ran?** YES. Token `/tmp/mehanik-cleared-99886` present. Timestamp: `2026-05-08T21:06:23.121Z`.

## Blueprint cited?

- `blueprint_read: /Users/makinja/business/ALAI-Holding-AS/products/Bilko/BUILD-BLUEPRINT.md`
- This is the Bilko blueprint. The task is a *system-wide canonical spec edit*, not a Bilko-specific build task.
- The project path assigned was Bilko's path, which means Mehanik's blueprint check was anchored to Bilko even though the deliverables (`~/system/specs/cicd-canonical-v3-drafts/`) are system-level. This is a scope-mismatch in the Mehanik gate — the blueprint read is nominally satisfied but the product checked (Bilko) is not the target of the changes.

**Blueprint score:** 80/100 (WARN). Dispatch allowed.

**Agent output referenced blueprint sections?** The DOD evidence in MC #99886 references the task as a "system-wide canonical spec edit" and notes 5 issue-areas in the v3 drafts — none reference Bilko blueprint sections. The blueprint read appears to have been a gate-pass ritual, not a content-informing step.

**Dispatch outcome:** Deferred (not dispatched to FlowForge) — "executive-side decision to defer flowforge run until parallel work coordinates." The Mehanik clear token was written but the agent run was held. This is the correct behavior per CEO decision, but it reveals that Mehanik clearance does not guarantee agent execution — it is one gate in a multi-gate flow.

**Trace verdict:** Mehanik ran and wrote a token. The blueprint cited was topically mismatched (Bilko blueprint for a system-spec task). The blueprint score gate passed despite being below threshold. Agent was not dispatched (deferred). Blueprint content did not visibly inform the dispatch.

---

## 6. Open Questions

1. **Mehanik project\_path heuristic:** How does Mehanik determine which `project_path` to use when the task is cross-product or system-level? For #99886, Bilko was used for a system-spec task. Is this John's input, or Mehanik's inference? If inference, the blueprint check is unreliable for cross-cutting tasks.
2. **Score threshold enforcement:** The `blueprint_threshold_applied: 90` field in cleared tokens is never enforced as a hard gate. Drop scored 65 and dispatch was allowed. Should the threshold be lowered to match operational reality, or should the WARN-to-BLOCK escalation be implemented?
3. **Token reuse across sessions:** `mehanik_session_id: unknown` in all inspected tokens. Is there a plan to enforce session binding? Without it, a cleared token from a prior CEO session could authorize a dispatch in a new context.
4. **Gotiva and Lobby stale blueprints:** Both products are 59d+ stale. Are they in maintenance mode or abandoned? If active, their blueprints are Mehanik bypass risks for any dispatch — the gate will pass but Mehanik will be reading outdated architecture.

5. **MEDON client coverage:** No BUILD-BLUEPRINT.md exists for the MEDON shopify-app. If John receives a MEDON task, Mehanik's Phase T will fire `ls {project_path}/BUILD-BLUEPRINT.md` → BLOCKED. Is the MEDON client expected to receive blueprint coverage, or is it out of scope?
- 

## 7. ROI Lens (sentinel-ba)

### Is the blueprint pattern earning its overhead?

#### Direct value delivered:

- Blueprint presence as a Mehanik gate prerequisite has prevented scope hallucination at the dispatch level. The 113 mehanik-cleared tokens in `/tmp` represent 113 gate events where someone was forced to confirm a blueprint existed and was read. This is a real forcing function.
- The Drop AWS phantom rewrite (MC #10353) is a concrete example where the blueprint served as the canonical source of truth that agents were required to consult — and where a discrepancy (aspirational AWS docs treated as ground truth) was detected and corrected with a committed blueprint update.
- The Bilko blueprint (38KB, 530 lines, git-tracked) is the most thorough — it provides stack, ADRs, domain context, and deployment architecture. It has demonstrably prevented repeated infra hallucination on Bilko tasks.

#### Overhead cost:

- 17 blueprints exist, 8 are genuinely substantial. The 2 stubs (sintef/akershus) add near-zero value and should be either expanded or removed (their Mehanik gate pass is hollow).
- Blueprint maintenance is manual and unalerted. Stale blueprints (BasicFakta 63d, Lobby 61d, Gotiva 59d) represent a risk: Mehanik passes the gate but the agent reads outdated architecture. The overhead of writing blueprints is paid; the staleness risk is not managed.
- The 90-point score threshold being advisory-only means the quality gate was designed but not deployed. This is overhead (blueprint-check.js runs on every dispatch) with only partial benefit (WARN path is free).

**Net verdict: POSITIVE ROI, but with a quality gap.** The blueprint pattern is not theatrical — it is a genuine gate that has caught real hallucinations. However, the enforcement has two systemic weaknesses: (1) stale blueprints pass the gate silently, and (2) the score threshold is never enforced as a block. Fixing these two issues would cost approximately 1–2 hours of system work and would sharply increase the ROI-per-blueprint.

#### Priority recommendations:

1. **HIGH — Enforce score threshold or lower it.** Either block at score < 60 (matching current floor observed in practice), or officially downgrade the threshold. WARN-at-65-and-

dispatch is worse than an honest 60-point threshold that blocks.

2. **HIGH — Add staleness alert.** A daily check: if blueprint last-modified > 30d AND project has had commits in last 14d → surface warning to John. Zero build cost (can be added to existing daemon fleet).
3. **MED — Expand or remove stub blueprints.** sintef (49 lines) and akershus-fylke (33 lines) are hollow gates. MC #99886 Decision 7 already proposes moving akershus out of products/ — execute this and either write a real blueprint or remove the gate.
4. **LOW — Session binding for Mehanik tokens.** Low urgency given 4h expiry, but `mehanik_session_id: unknown` should be resolved to prevent cross-session token reuse on long-running tasks.

---

Revision #2

Created 2026-05-09 19:44:21 UTC by John

Updated 2026-06-14 20:02:57 UTC by John