

ADR-020: Backend Canonical — Deprecate api-kotlin

```
# ADR-020: Canonical Backend is `backend/` – Deprecate `apps/api-kotlin/`

**Status:** Accepted
**Date:** 2026-04-28
**Author:** ALAI, 2026
**Related:** ADR-009 (superseded), ADR-011, ADR-015, ADR-016, ADR-017, ADR-018, ADR-019

> **MAJOR PATH UPDATE (2026-04-29):** `backend/` ? `apps/api/` (canonical Kotlin/Ktor
location now). `apps/api-legacy/` ? `.archive/api-legacy/`. The deprecation of `api-
kotlin/` in this ADR was executed in MC #10034 (deleted as `apps/api-kotlin-abandoned`).
See ADR-021.

---

## Context

### The Dual-Backend Incident

As of 2026-04-27, the Bilko repository contained two parallel, independent Kotlin/Ktor
backends
serving identical purposes – an architectural anomaly discovered during forensic audit MC
#9892.

**Timeline – git-verified (SHA + date + author):**

| SHA          | Date          | Author      |
| Event        |
| ----- | ----- | ----- |
|-----|-----|-----|
| `5f97eff` | 2026-03-04 | John AI | Earliest backup commit – `backend/` fully present with
`build.gradle.kts`, package `no.alai.bilko`, Kotlin 2.3.0 / Ktor 3.4.0 / JVM
25
| `e23ade3` | 2026-03-19 | Makinja | Security headers plugin and security audit added to
`backend/`
| `6b76981` | 2026-04-10 | Makinja | CI fixes applied to
`backend/`
| `6c71a79` | 2026-04-14 | Makinja | `apps/api-kotlin/` created – "Complete Kotlin/Ktor
backend – Auth, Invoices, Clients, Expenses, Health" – package `io.bilko`, Kotlin 2.1.20 /
Ktor 3.1.2 / JVM 21
| `f66ddec` | 2026-04-14 | Makinja | GCP Terraform + CI added (lands in both
directories)
| `ee27c6b` | 2026-04-15 | Makinja | `apps/api-kotlin/` scaffold finalised – 10 feature
modules, 17 source files, titled "migration
scaffold"

**Result:** `backend/` was first on 2026-03-04 (6 weeks before `apps/api-kotlin/`). A
generic
builder agent (dispatched without Mehanik gate clearance) created `apps/api-kotlin/` on
2026-04-14 while `backend/` was already the active implementation. From 2026-04-15 onward,
`apps/api-kotlin/` received no further commits. `backend/` continued active development
with
commits through 2026-04-23.

### Background – Why This Happened

CEO decision (2026-03-17, ALAI/CLAUDE.md) mandated migration of all products from
Express/TS to
Kotlin/Ktor. Migration task MC #5125 ("Bilko backend: Express/TS ? Kotlin/Ktor") was
created but
```

not formally unblocked. The Phase 1 Track A execution document (``docs/bookstack-sync/phase1-track-a-execution.md``, lines 241 and 299) designated ``apps/api-kotlin/`` as the FUTURE migration target – explicitly stating:

```
> "Do NOT start Track B until MC #5125 unblocks. All Track B work goes into `apps/api-kotlin/`."
```

However, a generic builder agent was dispatched into ``apps/api-kotlin/`` on 2026-04-14 before MC #5125 was formally unblocked and before Mehanik clearance was obtained. This created an unauthorized parallel scaffold while the actual canonical domain implementation continued to grow in ``backend/``.

State at Time of Discovery (MC #9892, 2026-04-27)

``backend/``:

- Package: ``no.alai.bilko``
- Kotlin 2.3.0 / Ktor 3.4.0 / JVM 25
- 51 ``.kt`` source files + 3 test files
- Full domain: HR-FISK, SEF, EInvoice, AdapterException (14 codes), Koin DI, Redis, Apache PDFBox, Sentry, EmailService, SecurityHeaders, CORS, RateLimit
- All ADR-015 through ADR-019 reference ``no.alai.bilko`` paths inside ``backend/``
- Last commit: 2026-04-23 (John, active)

``apps/api-kotlin/``:

- Package: ``io.bilko``
- Kotlin 2.1.20 / Ktor 3.1.2 / JVM 21
- 17 ``.kt`` source files + 3 test files
- Skeleton only: Auth, DB tables, feature route scaffolds
- Missing: HR-FISK, SEF, EInvoice adapter interface, AdapterException, Koin DI, Redis, PDFBox, Sentry, EmailService, RateLimit, CORS
- Last commit: 2026-04-15 ``ee27c6b`` (Makinja, ****13 days stale**** at discovery)
- NOT deployed (confirmed by ``docs/evidence/9386/verification.json`` line 36 and ``docs/evidence/9398/verify-cookie-fix.js`` line 76)
- NOT referenced in any architecture document

Prior Audit Gap

The preliminary architecture audit (2026-04-27) saw ``apps/api-kotlin/`` in the directory listing and noted: `_"need to confirm these are indeed empty/removed"_` – but did not follow through with a ``find`` verification. The tool-first discipline (ZAKON NULA) required explicit verification before any assumption about directory contents. The audit concluded without detecting the 17 active Kotlin files, full auth module, and complete Gradle build in ``apps/api-kotlin/``.

Decision

``backend/`` is the canonical Kotlin/Ktor backend for Bilko.

``apps/api-kotlin/`` is deprecated and will be archived as of MC #9894.

All present and future development of the Bilko API occurs in ``backend/``. The ``io.bilko`` package namespace is abandoned. The ``no.alai.bilko`` namespace (established 2026-03-04) is permanent for the Kotlin backend.

Rationale

Three hard facts – all git-verified, zero assumptions:

Fact 1 – Scale disparity (51 vs 17 files).
``backend/`` contains 51 Kotlin source files. ``apps/api-kotlin/`` contains 17. More critically, the qualitative gap is larger than the count suggests: ``backend/`` contains every domain-specific component (fiscal adapters, error registry, DI wiring, PDF generation, rate limiting, Sentry telemetry). ``apps/api-kotlin/`` contains only the routing skeleton.

```
**Fact 2 – All ADR paths reference `backend/`.**
ADR-014 through ADR-019 – every architecture decision record written for this product –
reference
`no.alai.bilko` paths inside `backend/`. ADR-019 was explicitly verified against
`backend/src/main/kotlin/no/alai/bilko/adapters/AdapterException.kt`. Zero architecture
documents
reference `io.bilko` or `apps/api-kotlin/`. An ADR is a commitment. Reversing ADR-015
through
ADR-019 evidence chains to point at `apps/api-kotlin/` would be rework with no technical
benefit.

**Fact 3 – Version inversion confirms direction of travel.**
`apps/api-kotlin/` is pinned to Kotlin 2.1.20 / Ktor 3.1.2 – the versions specified in the
original Phase 1 Track A scaffold spec. `backend/` runs Kotlin 2.3.0 / Ktor 3.4.0 / JVM 25
–
current as of 2026-04-28. This is not a coincidence: `apps/api-kotlin/` was scaffolded once
to a
fixed spec and never updated. `backend/` was actively maintained and upgraded. The version
delta
tells the entire story: one codebase is alive, the other is frozen at its creation point.

---

## Consequences

### For Lane B BLOCKER Tasks (#9852 / #9853 / #9854 / #9855)

All Lane B backend tasks are unblocked against `backend/` as the target. No work should be
directed to `apps/api-kotlin/`. Any task whose scope referenced `apps/api-kotlin/` or
`io.bilko`
must be updated to reference `backend/` and `no.alai.bilko` before execution begins.

### For MC #5125 (Bilko Express ? Kotlin Migration)

MC #5125 was the formal trigger for the Kotlin migration and the stated prerequisite for
any
work in `apps/api-kotlin/`. With this ADR:

- `backend/` fulfills the Kotlin/Ktor migration requirement – the migration is structurally
  complete at the backend layer.
- MC #5125 may be closed (DONE) once BUILD-BLUEPRINT.md is updated (MC #9897) to document
  `backend/` as the canonical backend and the Express legacy (`apps/api-legacy/`) as
  deprecated.
- The specific Track A intent ("apps/api-kotlin/ is the migration landing zone") is
  superseded
  by this ADR. Track B work proceeds in `backend/` directly.

### For BUILD-BLUEPRINT.md

BUILD-BLUEPRINT.md §3 currently documents `apps/api/` (now `apps/api-legacy/`) as the
backend
and makes no mention of either `backend/` or `apps/api-kotlin/`. This is pre-migration
documentation. MC #9897 (BUILD-BLUEPRINT update) must:

1. Replace the backend section to reference `backend/` (package `no.alai.bilko`, Kotlin
  2.3.0,
  Ktor 3.4.0)
2. Document the directory structure: `backend/` lives outside Turborepo workspace
  (independent
  Gradle + GCP Cloud Run deploy)
3. Update build commands to `cd backend && ./gradlew run`
4. Mark `apps/api-legacy/` as deprecated with a pointer to its decommission timeline

### For DEPLOY-MAP.md

DEPLOY-MAP.md currently records bilko-api as "Manual only (Kotlin TBD)". After Dockerfile
work
(MC #9898), this entry must be updated to reflect: source = `backend/`, build = Docker
multi-stage, deploy target = GCP Cloud Run via `gcp-deploy.yml`.

### Negative Consequences

1. **One-time porting effort.** `apps/api-kotlin/` contains a more rigorous implementation
  of
  refresh token rotation (`features/auth/AuthRepository.rotateRefreshToken()`). This
  pattern
```

should be reviewed against `backend/` before archiving – MC #9895 covers this comparison.

2. **Track A plan invalidated.** The Phase 1 Track A execution document explicitly designated `apps/api-kotlin/` as the future target. That plan is now superseded. The document must be annotated with a pointer to this ADR to prevent future agents from acting on stale guidance.

3. **Feature-sliced architecture not adopted.** `apps/api-kotlin/` used a feature-sliced layout (`features/auth/`, `features/invoices/`). `backend/` uses a layered layout (`routes/`, `services/`, `auth/`). The architectural pattern debate is resolved in favor of the layered approach by inertia – 51 files are not being reorganized. This is a deliberate trade-off: stability over structural preference.

Lessons Learned

Lesson 1 – Premature Scaffold Incident

The Track A document stated that `apps/api-kotlin/` should NOT be built until MC #5125 formally unblocked. A generic builder agent built it anyway (2026-04-14). This is the root cause of the entire incident. The constraint in writing was not sufficient – it required a hard gate (Mehanik) enforcing it programmatically.

Fix: Mehanik pre-dispatch gate (activated 2026-04-25, MC #9274) is the structural remedy. No backend task may be dispatched without Mehanik clearance that checks: task MC ID exists, BUILD-BLUEPRINT.md read, scope ceiling verified, CI green if deploy.

Lesson 2 – "Probably Empty" Hallucination in Audit

The preliminary audit saw `apps/api-kotlin/` in the `ls` output and wrote "need to confirm these are indeed empty/removed" – then concluded without verifying. The correct tool-first discipline (ZAKON NULA) required a `find apps/api-kotlin/src -name "*.kt"` call before any assumption about directory state. A single verification command would have revealed 17 Kotlin files and flagged the duplicate immediately.

Fix: Any directory flagged as "need to confirm" in an audit is an open obligation, not a closed finding. Audits are not complete until all flagged items are machine-verified. Post-audit review by a second agent (MC #9892 forensic) should be standard for architecture-level audits on active codebases.

Lesson 3 – ZAKON NULA Violation (Tool-First)

John dispatched the backend-dev agent to `apps/api-kotlin/` without reading BUILD-BLUEPRINT.md, without running `node ~/system/tools/mc.js show 5125`, and without querying the existing project structure. Had BUILD-BLUEPRINT.md been read first, it would have been apparent that `backend/` was the active implementation and that `apps/api-kotlin/` was the designated (but not yet active) future target – a distinction requiring a human (Alem) decision, not an agent initiative.

Fix: ZAKON NULA (CLAUDE.md) is enforced by the Mehanik pre-dispatch hook. The hook requires tool-verified project state before clearing any build dispatch.

Lesson 4 – ZAKON #1 Violation (Specialist Routing)

MC #5125 is a complex backend migration (Express/TS ? Kotlin/Ktor, domain logic, multi-market

fiscal adapters, DI framework selection). This requires a **specialist** – CodeCraft (Petter Graff / Hadi Hariri), not a generic builder agent. CLAUDE.md §5 is unambiguous: "Never generic builder/minion. Route to the right company." Generic agents lack the architectural judgment to navigate this class of decision (where does the backend live? which package namespace? which version pins?).

Fix: Complex backend migrations are categorically CodeCraft work. If the specialist routing table in CLAUDE.md is unclear for a given task, the correct action is to ask John, not to default to a generic pool. The Mehanik gate now enforces specialist routing as part of its clearance criteria.

Migration Path

The following tasks (C2-C6, MC #9894-#9898) execute the deprecation and consolidation. All tasks have Mehanik-cleared MC IDs. Sequencing matters: C2 (archive) must complete before C3 (port auth) to avoid confusion about which directory to edit.

C2 – Archive `apps/api-kotlin/` (MC #9894)

Owner: CodeCraft | **Effort:** S (1h)

1. Rename `apps/api-kotlin/` to `apps/api-kotlin-abandoned/`.
2. Add `README.md` at the root of the renamed directory:

```
DEPRECATED 2026-04-28 – see ADR-020
```

```
This directory is the abandoned migration scaffold created 2026-04-14 to 2026-04-15.
The canonical Kotlin backend is /backend/ (no.alai.bilko, Kotlin 2.3.0, Ktor 3.4.0).
Do not edit this directory. It will be deleted after 2026-05-28.
```

3. Verify `turbo.json` and root `package.json` workspaces do NOT include `apps/api-kotlin` or `apps/api-kotlin-abandoned` (Turborepo workspace scope must not resolve against it).
4. Annotate `docs/bookstack-sync/phasel-track-a-execution.md` lines 241 and 299 with:
`[SUPERSEDED by ADR-020 – apps/api-kotlin abandoned, backend/ is canonical]`.

C3 – Port Auth Improvements to `backend/` (MC #9895)

Owner: CodeCraft | **Effort:** M (4h)

Compare `apps/api-kotlin-abandoned/features/auth/AuthRepository.kt` (specifically `rotateRefreshToken()`) and the ThreadLocal side-channel pattern) against `backend/src/main/kotlin/no/alai/bilko/auth/AuthService.kt`. If the abandoned version is more rigorous, port the improvement. Do not port file structure or package names.

Scope: auth only. No feature modules, no table objects, no routing changes.

C4 – Update BUILD-BLUEPRINT.md (MC #9897)

Owner: CodeCraft | **Effort:** S (2h)

See Consequences section above for mandatory content. In addition, add an explicit architectural note: "`backend/` lives outside the Turborepo workspace by design. It is a standalone Gradle project with its own GCP Cloud Run deploy pipeline. Do not move it inside `apps/`."

C5 – Add Dockerfile to `backend/` + Update DEPLOY-MAP.md (MC #9898)

Owner: FlowForge | **Effort:** M (4h)

1. Port `apps/api-kotlin-abandoned/Dockerfile` (JVM 21, multi-stage, non-root user, health check) to `backend/Dockerfile`. Upgrade base image from JVM 21 to JVM 25 (matching `backend/` JVM target).
2. Verify fat JAR output name: `bilko-api.jar` (check `build.gradle.kts` shadowJar config).
3. Local build validation: `docker build -t bilko-api-test ./backend` must succeed.

```
4. Update DEPLOY-MAP.md bilko-api entry: source = `backend/`, Dockerfile =
`backend/Dockerfile`,
  deploy = GCP Cloud Run via `gcp-deploy.yml`.

### C6 – Proveo Verification (MC #9898 gate, Proveo)

**Owner:** Proveo (Angie Jones) | **Effort:** S (2h)

Acceptance criteria:

1. `docker build -t bilko-api ./backend` exits 0.
2. `docker run --rm -p 8080:8080 bilko-api` starts and responds to `GET /health` with HTTP
200.
3. `apps/api-kotlin/` directory no longer exists in repo root (renamed per C2).
4. `turbo.json` workspaces grep returns no match for `api-kotlin`.
5. `grep -r "io.bilko" backend/` returns no matches (no namespace contamination).

---

## References

- **MC #9892** – Forensic audit: dual Kotlin backend root-cause analysis
- **MC #9894** – C2: Archive `apps/api-kotlin/`
- **MC #9895** – C3: Port auth improvements to `backend/`
- **MC #9897** – C4: Update BUILD-BLUEPRINT.md
- **MC #9898** – C5/C6: Dockerfile + DEPLOY-MAP.md + Proveo verify
- **MC #5125** – Bilko backend migration: Express/TS ? Kotlin/Ktor (to be closed after MC
#9897)
- **ADR-015** – Four-Jurisdiction Plugin Architecture (references `no.alai.bilko`)
- **ADR-016** – E-Invoice Adapter and UBL 2.1 Canonical Model (references `no.alai.bilko`)
- **ADR-017** – RLS Multi-Tenancy (references `no.alai.bilko`)
- **ADR-018** – Market Locale Separation (references `no.alai.bilko`)
- **ADR-019** – Integration Adapter Registry (explicitly verified against `backend/` path)
- **docs/bookstack-sync/phase1-track-a-execution.md** – Phase 1 Track A intent document
(lines 241, 299 superseded by this ADR)
- **Forensic reports** – `/tmp/bilko-dual-backend-da.md`, `/tmp/bilko-dual-backend-
petter.md`

---

## Approval

**Accepted:** 2026-04-28
**Executed by:** ALAI, 2026
**Execution tasks:** MC #9894, #9895, #9897, #9898
```

Revision #2

Created 2026-05-14 19:44:49 UTC by John

Updated 2026-06-14 20:03:22 UTC by John