ADR-017: RLS Multi-Tenancy Migration

```
# ADR-017 — RLS Multi-Tenancy Migration

**Status:** Accepted — CEO Signed 2026-05-11 (Alem Baši?). Phase 2A V17 Flyway PERMISSIVE
migration authorized for stage execution. Phase 2C RESTRICTIVE flip remains gated on
Securion audit + 30-day soak per §4 schedule.
**Date:** 2026-05-11
**Author:** Bruce Momjian (Database Architecture, CodeCraft)
**Architecture Review:** Petter Graff (CodeCraft)
**Decision-maker:** CEO Alem Baši? — SIGNED 2026-05-11 ("ok adr17 odobreno") via session
f73dafab
**Mehanik clearance:** /tmp/mehanik-cleared-100362
**MC Task:** #100362 (Phase 0' ADR Consolidation)
**Promoted from:** ADR-bilko-001 draft (`~/system/specs/bilko-multi-market-architecture-
plan/ADR-bilko-001-multi-tenant-architecture.md`)
**Cross-references:**

- ADR-023 (why single DB remains correct — §6 supersession triggers not fired; §2 context)
- ADR-015 (TaxJurisdiction enum drives `country_code` column CHECK values)
- ADR-bilko-001 (ancestor draft, fully absorbed by this ADR — do not reference ancestor)
- ADR-bilko-003 §Layer 3 (versioned CoA data model)
- Plan v3 §4a (Option D not triggered), §4c (RLS timing — PERMISSIVE before Phase 1H merge)
- `~/system/specs/bilko-multi-market-architecture-plan-v3-2026-05-11.md`

---

## 1. Context

### 1.1 Current DB State (tool-verified 2026-05-11)

| Component               |
State
                                                                  |
| ----------------------- |
---------------------------------------------------------------------------------------------
---------------------------------------------------------------- |
| Database                | `bilko-demo-db`, Cloud SQL PostgreSQL 15, europe-
north1
                  |
| Flyway migrations       | V1..V15
applied
                                                          |
| Row-Level Security      | NOT enabled — zero RLS policies on any
table
                            |
| Tenant isolation        | Application-layer only: `WHERE org_id = :principalOrgId`
clauses
          |
| `organizations.country` | Column exists; values `'RS'`, `'HR'`, `'BA'`; NOT NULL
constraint
absent
  |
| Cross-tenant leak       | Confirmed: PUT `/api/v1/invoices/{id}` and GET
`/api/v1/invoices/{id}/pdf` with cross-tenant JWT return HTTP 500 (test drift memo 2026-05-
10, Round 12.1/12.5) |

The current application-layer scoping (ADR-005) is the sole isolation mechanism. A single
missing `WHERE org_id` clause in any new route — or a refactoring that silently drops it —
is
a cross-tenant data exposure. This is not theoretical: Round 12 probes confirmed it in two
existing routes.

### 1.2 Why Single Database Remains Correct (ADR-023 §6 Check)

ADR-023 §6 defines the conditions that would trigger migration to Option D (per-country
```

DBs).
All five conditions are unmet as of 2026-05-11 (Plan v3 §4a lines 100–108):

- Paying customers in 2+ markets: 0 — NOT triggered
- Regulatory request for per-country data extract: none received — NOT triggered
- HR-FISK kernel-level coupling: Storecove API path requires no kernel isolation — NOT triggered
- p95 query latency > 500ms from cross-country noise: 0 paying customers — NOT triggered
- 2 customers complain about cross-country data visibility: 0 customers — NOT triggered

Option D costs +$60/month infra and 2–4 weeks engineering per market with no customer-facing
benefit today. **This ADR is explicitly compatible with Option D migration** — RLS policies are portable to separate databases. If Option D triggers, the same policy DDL applies to each per-country DB with zero changes.

1.3 Why RLS Cannot Wait Until Post-HR GA

Plan v3 §4c (lines 135–145): the cross-tenant 500 leaks are a live security defect. With 0 paying customers today it is unexploited — but a second registered organization (required for HR demo) creates an immediately exploitable state.

RLS PERMISSIVE mode (Phase 2A) imposes zero user-facing change and zero risk of service disruption. The existing `WHERE org_id` middleware still fires, and RLS fires alongside it. Both must pass for data to be returned. A latent policy gap is caught by the application layer rather than exposing data to the wrong tenant.

CEO sign is required before Phase 2A Flyway migrations run on stage — not before this ADR document is accepted. The ADR records the decision; the sign unblocks execution.

---

2. Decision

Option C is adopted: Shared codebase, shared deployment, shared database, with PostgreSQL Row-Level Security enforcing tenant isolation.

This is the unanimous recommendation from the 5-agent architecture review (ADR-bilko-001 §framing,
line 28–30). One codebase. One Cloud Run deployment. One PostgreSQL instance with RLS.

2.1 Binding Constraints

1. `Organization.taxJurisdiction` (`TaxJurisdiction` enum `{HR, RS, BA_FED, BA_RS}` per ADR-015) is
   the primary discriminator for jurisdiction-specific behaviour.
2. `Organization.id` (UUID) is the primary tenant discriminator for data isolation.
3. RLS policies enforce data isolation at the database layer. Application code MUST NOT rely solely on `WHERE org_id = :id` clauses (ADR-005 flaw — being retired by Phase 2C).
4. The `country_code` column on `organizations` is NOT NULL with CHECK constraint `IN ('HR', 'RS', 'BA_FED', 'BA_RS')` — enforced by Flyway V16 (Phase 1H Task 1H.1).
5. EU data residency: Current `bilko-demo-db` is in Cloud SQL `europe-north1` (Finland). This IS within EU/EEA — GDPR Article 44 satisfied. Frankfurt migration (eu-central-1) is not required to unblock HR GA (Plan v3 §4d lines 179–183).

2.2 Three-Phase Migration Path

The migration is split into three phases to ensure zero service disruption and a safe rollback path at each step.

Phase 2A — PERMISSIVE RLS (parallel with Phase 1H, target: end of Week 2)

Goal: RLS policies created and attached, set to PERMISSIVE. Existing application-layer scoping continues to operate. Both layers must pass — RLS is a second check, not a replacement.

Who signs this off: CEO Alem Baši? (this ADR signature) — required before any Phase 2A Flyway migrations run on the stage database.

DDL — PERMISSIVE policies (Flyway V17):

```sql
-- V17__rls_permissive.sql
-- ZAKON: CEO sign required before this migration runs on stage.
-- Apply PERMISSIVE RLS on core tables. Application-layer WHERE org_id
-- clauses remain active. Both must pass.
```

```
-- Enable RLS on tables
ALTER TABLE organizations       ENABLE ROW LEVEL SECURITY;
ALTER TABLE invoices            ENABLE ROW LEVEL SECURITY;
ALTER TABLE invoice_items       ENABLE ROW LEVEL SECURITY;
ALTER TABLE expenses            ENABLE ROW LEVEL SECURITY;
ALTER TABLE transactions        ENABLE ROW LEVEL SECURITY;
ALTER TABLE bank_transactions   ENABLE ROW LEVEL SECURITY;
ALTER TABLE bank_accounts       ENABLE ROW LEVEL SECURITY;
ALTER TABLE accounts            ENABLE ROW LEVEL SECURITY;
ALTER TABLE contacts            ENABLE ROW LEVEL SECURITY;

-- PERMISSIVE policy: organization-scoped isolation
-- current_setting() reads the app.current_org_id session variable
-- set by the Ktor connection pool before each query (connection middleware).

CREATE POLICY org_isolation ON invoices
    AS PERMISSIVE
    FOR ALL
    TO bilko_app              -- application role (NOT superuser)
    USING (org_id = current_setting('app.current_org_id')::uuid);

CREATE POLICY org_isolation ON invoice_items
    AS PERMISSIVE
    FOR ALL
    TO bilko_app
    USING (
        invoice_id IN (
            SELECT id FROM invoices
            WHERE org_id = current_setting('app.current_org_id')::uuid
        )
    );

CREATE POLICY org_isolation ON expenses
    AS PERMISSIVE
    FOR ALL
    TO bilko_app
    USING (org_id = current_setting('app.current_org_id')::uuid);

CREATE POLICY org_isolation ON transactions
    AS PERMISSIVE
    FOR ALL
    TO bilko_app
    USING (org_id = current_setting('app.current_org_id')::uuid);

CREATE POLICY org_isolation ON bank_transactions
    AS PERMISSIVE
    FOR ALL
    TO bilko_app
    USING (
        bank_account_id IN (
            SELECT id FROM bank_accounts
            WHERE org_id = current_setting('app.current_org_id')::uuid
        )
    );

CREATE POLICY org_isolation ON bank_accounts
    AS PERMISSIVE
    FOR ALL
    TO bilko_app
    USING (org_id = current_setting('app.current_org_id')::uuid);

CREATE POLICY org_isolation ON accounts
    AS PERMISSIVE
    FOR ALL
    TO bilko_app
    USING (org_id = current_setting('app.current_org_id')::uuid);

CREATE POLICY org_isolation ON contacts
    AS PERMISSIVE
    FOR ALL
    TO bilko_app
    USING (org_id = current_setting('app.current_org_id')::uuid);

-- BYPASS for migrations and admin tooling (Flyway runs as bilko_admin)
ALTER TABLE invoices         FORCE ROW LEVEL SECURITY;
ALTER TABLE expenses         FORCE ROW LEVEL SECURITY;
ALTER TABLE transactions     FORCE ROW LEVEL SECURITY;
```

```
ALTER TABLE bank_transactions FORCE ROW LEVEL SECURITY;
ALTER TABLE bank_accounts    FORCE ROW LEVEL SECURITY;
ALTER TABLE accounts         FORCE ROW LEVEL SECURITY;
ALTER TABLE contacts         FORCE ROW LEVEL SECURITY;
-- Flyway runs as bilko_admin (superuser bypasses RLS by default).
-- Explicit FORCE is belt-and-suspenders — admin role grants BYPASSRLS if needed.

-- Set connection middleware (Kotlin Exposed / HikariCP):
-- On each connection checkout:
--   SET LOCAL app.current_org_id = '<org_uuid_from_jwt>';
-- On connection return to pool:
--   SET LOCAL app.current_org_id = '';  -- or reset_config('app.current_org_id', true)
```

Verification after Phase 2A:

```sql
-- Rogue-role test (Proveo E2E + Securion audit):
SET ROLE bilko_app;
SET LOCAL app.current_org_id = '<hr_org_uuid>';
SELECT count(*) FROM invoices;  -- must return only HR org rows
SET LOCAL app.current_org_id = '<rs_org_uuid>';
SELECT count(*) FROM invoices;  -- must return only RS org rows
-- Cross-tenant access attempt:
SET LOCAL app.current_org_id = '<hr_org_uuid>';
SELECT * FROM invoices WHERE org_id = '<rs_org_uuid>';  -- must return 0 rows (PERMISSIVE
blocks)
```

Phase 2B — Audit Log Partitioning (post-HR GA)

Goal: Partition the `logged_actions` audit table by `country_code` to enable
per-jurisdiction GDPR data extraction requests and enforce per-jurisdiction retention.

```sql
-- V18__audit_log_partitioning.sql (Phase 2B — post-HR GA)

-- Declarative partitioning by country_code
CREATE TABLE logged_actions_partitioned (
    LIKE logged_actions INCLUDING ALL
) PARTITION BY LIST (country_code);

CREATE TABLE logged_actions_hr     PARTITION OF logged_actions_partitioned
    FOR VALUES IN ('HR');
CREATE TABLE logged_actions_rs     PARTITION OF logged_actions_partitioned
    FOR VALUES IN ('RS');
CREATE TABLE logged_actions_ba_fed PARTITION OF logged_actions_partitioned
    FOR VALUES IN ('BA_FED');
CREATE TABLE logged_actions_ba_rs  PARTITION OF logged_actions_partitioned
    FOR VALUES IN ('BA_RS');

-- Retention policy enforcement (aligned with CountryPlugin.getRetentionRules()):
-- HR: 11 years (Zakon o ra?unovodstvu NN 78/2015, ?l. 10)
-- RS/BA: 10 years
-- Implemented as pg_cron job deleting rows WHERE action_tstamp_tx < now() - interval '11
years'
-- per partition.

-- country_code column backfilled from organizations.country via:
-- UPDATE logged_actions SET country_code = o.country
-- FROM organizations o WHERE o.id = logged_actions.org_id;
```

RLS policy for `logged_actions` (applied in Phase 2B):

```sql
CREATE POLICY org_isolation ON logged_actions_partitioned
    AS PERMISSIVE
    FOR ALL
    TO bilko_app
    USING (org_id = current_setting('app.current_org_id')::uuid);
```

Phase 2C — RESTRICTIVE + Retire Application-Layer Scoping (post-Securion Audit)

Goal: Convert PERMISSIVE policies to RESTRICTIVE. Remove ADR-005 application-layer
`WHERE org_id` middleware. RLS is the sole isolation mechanism.

Gate conditions (all must be true before Phase 2C begins):

1. Securion audit of Phase 2A policies completed — no critical findings
2. Automated rogue-role test suite passing in CI (Proveo — see Phase 2A verification above)
3. Zero cross-tenant RLS bypass incidents on stage for 30 consecutive days
4. CEO explicit sign-off for Phase 2C

```sql
-- V19__rls_restrictive.sql (Phase 2C — post Securion audit)
-- Convert PERMISSIVE ? RESTRICTIVE on all tables
-- This is the point of no return: application layer WHERE org_id is retired after this.

DROP POLICY org_isolation ON invoices;
CREATE POLICY org_isolation ON invoices
    AS RESTRICTIVE
    FOR ALL
    TO bilko_app
    USING (org_id = current_setting('app.current_org_id')::uuid)
    WITH CHECK (org_id = current_setting('app.current_org_id')::uuid);

-- Same pattern for expenses, transactions, bank_transactions, bank_accounts,
-- accounts, contacts, invoice_items (repeat for each table).
```

2.3 Versioned Chart of Accounts Table

The `chart_of_accounts` table stores jurisdiction-specific CoA entries with time-ranged validity. This supports:

- Pravilnik revisions without code changes (ADR-bilko-003 §Layer 3, lines 122–143)
- Historical invoice accuracy (rate in force at transaction date, not current rate)
- `CountryPlugin.getChartOfAccountsDefaults()` seeding on org creation (ADR-015 §2.2)

```sql
-- Part of Flyway V17 or separate V17b (Phase 2A / 1H parallel)

CREATE TABLE chart_of_accounts (
    id              UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    jurisdiction    VARCHAR(8) NOT NULL,    -- TaxJurisdiction enum value: 'HR', 'RS',
'BA_FED', 'BA_RS'
    code            VARCHAR(16) NOT NULL,   -- e.g. '1300' (HR Kontni Plan), '204' (RS
Pravilnik)
    name            VARCHAR(256) NOT NULL,
    account_type    VARCHAR(16) NOT NULL    -- ASSET, LIABILITY, EQUITY, INCOME, EXPENSE
                    CHECK (account_type IN ('ASSET', 'LIABILITY', 'EQUITY', 'INCOME',
'EXPENSE')),
    vat_treatment   VARCHAR(64),            -- e.g. 'STANDARD_RATE', 'EXEMPT', null for
non-VAT accounts
    valid_from      DATE NOT NULL,
    valid_to        DATE,                   -- NULL = currently valid
    version         INT NOT NULL DEFAULT 1, -- increments per Pravilnik revision
    notes           TEXT,                   -- statutory reference e.g. "NN 78/2015, ?l. 5"
    created_at      TIMESTAMPTZ NOT NULL DEFAULT now(),
    UNIQUE (jurisdiction, code, valid_from)
);

CREATE INDEX idx_coa_jurisdiction_date
    ON chart_of_accounts (jurisdiction, valid_from, valid_to);

-- Query pattern: entries valid on a given transaction date
-- SELECT * FROM chart_of_accounts
-- WHERE jurisdiction = $1
--   AND valid_from <= $2
--   AND (valid_to IS NULL OR valid_to > $2)
-- ORDER BY code;

-- When Croatia raises PDV from 25% to 27% on 2027-01-01:
-- INSERT INTO chart_of_accounts (jurisdiction, code, name, account_type, vat_treatment,
valid_from, version)
-- VALUES ('HR', '2400', 'PDV po stopi 27%', 'LIABILITY', 'STANDARD_RATE', '2027-01-01',
2);
-- UPDATE chart_of_accounts SET valid_to = '2026-12-31'
-- WHERE jurisdiction = 'HR' AND code = '2400' AND valid_to IS NULL AND version = 1;
-- No code change required.
```

Seeding: `CountryPlugin.getChartOfAccountsDefaults()` returns the list of entries that Flyway data migrations insert into `chart_of_accounts` for each jurisdiction. Flyway V18 (Phase 1H — separate from V17 RLS) seeds HR Kontni Plan entries.

2.4 Exchange Rate Precision Upgrade

Current precision (CLAUDE.md database rules): `NUMERIC(19,4)` for ALL monetary amounts.

Upgrade required for FX rate columns specifically:

Exchange rates require higher precision than invoice monetary amounts. Using `NUMERIC(19,4)`
for an exchange rate means EUR/RSD at 117.2350 is representable, but EUR/BAM at 1.95583 is stored as 1.9558 — a systematic rounding error that compounds across large invoice volumes and cross-currency reconciliation.

Decision: FX rate columns upgrade to `NUMERIC(20,10)`. Monetary amount columns (invoice totals, line amounts, tax amounts) remain `NUMERIC(19,4)`.

```sql
-- V17c__exchange_rate_precision.sql (Phase 2A parallel)

ALTER TABLE exchange_rates
    ALTER COLUMN rate TYPE NUMERIC(20,10);  -- was NUMERIC(19,4)

-- If an exchange_rate_history or similar snapshot table exists:
-- ALTER TABLE exchange_rate_history
--     ALTER COLUMN rate TYPE NUMERIC(20,10);

-- NEVER change invoice_items.unit_price, invoice_items.line_total,
-- transactions.amount, etc. — those remain NUMERIC(19,4).
-- Only rate/exchange_rate columns receive this upgrade.
```

Invariant: All monetary arithmetic (invoice totals, tax calculations, double-entry postings) remains at `NUMERIC(19,4)`. The precision upgrade is scoped to the FX rate storage layer only. Rounding when applying FX rates to amounts: round half-even (banker's rounding) to 4 decimal places after multiplication.

---

3. Connection Middleware — Setting `app.current_org_id`

The RLS policies use `current_setting('app.current_org_id')::uuid`. This session variable must be set on every database connection before any query executes.

Pattern (Kotlin / Exposed / HikariCP):

```kotlin
// apps/api/src/main/kotlin/no/alai/bilko/db/OrgContextInterceptor.kt (Phase 2A NEW)

/**
 * Sets the PostgreSQL session variable `app.current_org_id` to the authenticated
 * org's UUID before any database access.
 *
 * Called from the Ktor routing pipeline after JWT validation, before the
 * database transaction opens.
 *
 * Must reset after the request completes — use try/finally or Ktor plugin lifecycle.
 */
fun setOrgContext(orgId: UUID) {
    transaction {
        exec("SET LOCAL app.current_org_id = '${orgId}'")
    }
}

fun clearOrgContext() {
    transaction {
        exec("RESET app.current_org_id")
        // or: exec("SET LOCAL app.current_org_id = ''")
    }
}
```

Failure mode: If `app.current_org_id` is not set, `current_setting('app.current_org_id')`
throws an error in PostgreSQL (by default). To make it return NULL instead (for Flyway

admin connections that do not set the variable):

```sql
-- In V17 migration, set default:
ALTER DATABASE bilko_demo SET app.current_org_id = '';
```

And in the policy, guard against empty string:

```sql
USING (
    CASE WHEN current_setting('app.current_org_id', true) = ''
         THEN false  -- deny if not set
         ELSE org_id = current_setting('app.current_org_id', true)::uuid
    END
)
```

The `true` parameter to `current_setting()` makes it return NULL rather than throw when the variable is not set.

---

4. Migration Schedule

Phase	Flyway Version	Target	Blocking
Phase 1H.1	V16: `organizations.country` NOT NULL + CHECK	HR enum expansion (ADR-015)	ADR-015 accepted
Phase 2A	V17: PERMISSIVE RLS + CoA table + FX rate precision	Stage only	CEO sign (this ADR)
Phase 2A	V17 seed: HR Kontni Plan data	Stage only	V17 + PluginHR.getChartOfAccountsDefaults()
Phase 2B	V18: audit log partitioning	Post-HR GA	Securion review
Phase 2C	V19: RESTRICTIVE + retire ADR-005 app scoping	Post-Securion audit	Securion audit pass + CEO sign

All migrations use Flyway's expand/contract pattern. No migration modifies data in a way that cannot be reversed by a subsequent compensating migration. Backward compatibility is required across all rolling deployments.

---

5. Consequences

5.1 Positive

1. **Defence in depth.** Even if a developer introduces a missing `WHERE org_id` in a new route, RLS at the database layer prevents cross-tenant data exposure.
2. **GDPR jurisdiction extraction.** With `country_code` on `logged_actions` (Phase 2B), a request from Croatian DPA for "all data held on Croatian entities" is a single partition query, not a full-table scan with a filter.
3. **Audit surface.** Securion can review one set of RLS policies rather than auditing every application route for correct scoping.
4. **Option D readiness.** If ADR-023 §6 triggers (e.g., first paying HR customer), the same RLS DDL applies to the per-country databases without change. Migration path is not blocked by this ADR.

5.2 Negative

1. **Connection middleware requirement.** Every DB connection must set `app.current_org_id` before any query. Forgetting this in a new service or background job will cause all queries to return 0 rows (PERMISSIVE) or error (RESTRICTIVE). Mitigated by integration tests that verify the context middleware fires.
2. **Flyway admin bypass.** Flyway and admin tooling must run as a role that bypasses RLS (`bilko_admin` with BYPASSRLS). This role must be kept tightly restricted — it is a privilege escalation path.
3. **Phase 2A adds overhead.** Each query now evaluates an additional predicate. At current scale (0 paying customers) the overhead is immeasurable. Monitor p95 query latency after Phase 2A migration on stage.

5.3 Risks

1. **GDPR data residency.** Croatian entity data in Cloud SQL europe-north1 (Finland) is

   legally compliant (EU/EEA). If a future HR DPA contract specifies Frankfurt, a regional migration is required. This ADR does not block that migration.
2. **RLS policy gap.** An incorrect USING clause (e.g., JOIN condition that broadens the scope) could expose cross-tenant data. **Mitigation:** Securion audit before Phase 2C (RESTRICTIVE), automated rogue-role test in CI.
3. **Migration synchronization.** A Flyway migration failure mid-run leaves all markets degraded. All V17+ migrations must be backward-compatible and use expand/contract pattern. If V17 fails, rollback is: `DROP POLICY` + `ALTER TABLE ... DISABLE ROW LEVEL SECURITY`.

---

6. References

Reference	Path	Lines
ADR-bilko-001 (ancestor draft, absorbed by this ADR)	`~/system/specs/bilko-multi-market-architecture-plan/ADR-bilko-001-multi-tenant-architecture.md`	1–162
ADR-bilko-003 §Layer 3 (versioned CoA model)	`~/system/specs/bilko-multi-market-architecture-plan/ADR-bilko-003-market-abstraction-layers.md`	122–143
ADR-023 §6 (single-DB migration triggers — not fired)	`docs/architecture/ADR-023-TRANSITIONAL-MULTI-MARKET-ROUTING.md`	166–176
Plan v3 §4a (Option D not triggered — evidence)	`~/system/specs/bilko-multi-market-architecture-plan-v3-2026-05-11.md`	100–108
Plan v3 §4c (RLS timing — PERMISSIVE before Phase 1H)	`~/system/specs/bilko-multi-market-architecture-plan-v3-2026-05-11.md`	135–145
Plan v3 §4d (EU data residency does not block HR GA)	`~/system/specs/bilko-multi-market-architecture-plan-v3-2026-05-11.md`	179–183
ADR-015 §2.1 (TaxJurisdiction enum — `country_code` values)	`docs/architecture/ADR-015-FOUR-JURISDICTION-PLUGIN.md`	§2.1
Test drift memo (cross-tenant 500 leaks, Round 12.1/12.5)	`~/.claude/projects/-Users-makinja/memory/project_bilko_test_strategy_drift_2026-05-10.md`	—

---

7. Approval

Architecture status: Accepted (Phase 0' ADR consolidation)
CEO sign status: SIGNED 2026-05-11 — Phase 2A V17 Flyway PERMISSIVE migration authorized for stage. Phase 2C RESTRICTIVE flip remains gated on Securion audit + 30-day soak per §4 schedule.

This ADR records the architectural decision. The CEO signature below is the gate for execution of Phase 2A database migrations. It is not a gate for writing this document or for Phase 1H code work (CountryPlugin, PluginHR, DI wiring).

Role	Sign	Date
Architecture Lead (Petter Graff)	Signed	2026-05-11
Database Architecture (Bruce Momjian)	Signed	2026-05-11
CEO (Alem Baši?)	**SIGNED — session f73dafab, transcript "ok adr17 odobreno"**	2026-05-11

---

8. Document History

Date	Author	Change
2026-04-22	ALAI / ADR-bilko-001	Initial draft (multi-tenant architecture options analysis)

```
                    |
| 2026-05-11 | Bruce Momjian / Petter Graff | Promoted from ADR-bilko-001 draft; ID changed
to ADR-017; DDL examples added; versioned CoA DDL added; NUMERIC(20,10) FX precision noted;
Phase 2B audit log partitioning added; connection middleware pattern added; CEO sign gate
formalised. MC #100362. |
| 2026-05-11 | John (AI Director)           | CEO Alem Baši? signed ADR-017 via session
f73dafab ("ok adr17 odobreno"). Phase 2A V17 Flyway PERMISSIVE migration authorized for
stage. Status header + §7 approval table updated. Unblocks Bruce Momjian dispatch for Phase
2A.                          |
```

Revision #3
Created 2026-05-14 10:09:18 UTC by John
Updated 2026-06-14 20:03:17 UTC by John