HR eRa?un — Architecture Decision Record (ADR) + Build Plan

> **STATUS: design accepted; build in progress (WP1+). Production activation PARKED pending legal (B1/B2) + the multi-tenant decision.**
>
> NOTE: `app-api.bilko.cloud` maps to `bilko-api-demo` — the demo backend serves the bilko.cloud domain; activation is a real-domain decision, not a code toggle.

Status: ACCEPTED
Date: 2026-06-11
Lead Architect: Petter Graff (synthesized from team inputs)
Input Authors: Martin Kleppmann, Bruce Momjian, Markos Zachariadis, Parisa Tabriz

MC: #103453 (architecture documentation) | #103464 (build execution)

Cross-link: Bilko HR eRačun — sveRačun (PostLink) Integration & Status Model
CEO directive: "tim arhitekata, Petter Graff lead, plan → dokumentuju → build, BEZ HAKOVA."

---

1. Context and Problem

1.1 What Exists

Bilko has a Croatia HR eRačun adapter (`SveRacunHrEInvoiceAdapter`) with three implemented methods — `serialize()`, `submit()`, `pollStatus()` — and 42 unit tests. A Proveo-verified live TEST submission to the sveRačun (PostLink d.o.o.) TEST API returns HTTP 200 with a `documentId`. The status mapping (`mapStatusPair`) correctly implements the real sveRačun two-layer status model (corrected MC #103445).

1.2 The Three Structural Problems

Problem 1 — The wiring gap (critical).
No route, no service, and no persistence layer connects the product UI/API to the adapter. `POST /invoices/{id}/submit-to-sef` exists for Serbia (RS); HR has no equivalent.

`PluginHR.submitToFiscalPlatform` is `NOT_IMPLEMENTED` by design (it uses `FiscalReceipt`, not `CanonicalInvoice`). An operator cannot submit an HR invoice through Bilko today. This is the primary gap this ADR closes.

Problem 2 — Live double-fiscalization bug (critical, exists in the code today). `SveRacunHttpClient` installs `HttpRequestRetry` globally with `retryOnServerErrors(maxRetries = 3)`. This plugin fires on all 5xx responses — including those returned after sveRačun has already accepted and queued the document (transient 500 on the response-write path). A retry would POST the same UBL XML with the same invoice number to sveRačun a second time. In the Croatian fiscal model, that is a second fiscalization of the same invoice number — a criminal tax offence (Kazneni zakon, čl. 256). *This bug exists in the current codebase and must be fixed before any live call, including TEST calls.*

Problem 3 — The single-issuer ceiling (architectural). `serialize()` reads the XML sender OIB from `httpClient.configuredSenderVat`, which maps to the global env var `SVERACUN_SENDER_VAT`. sveRačun's etapa-1 rule requires that the initiator OIB (the API-key-holder) equal the XML sender OIB. With a single global env var, Bilko can only ever submit invoices as one legal entity. A multi-tenant SaaS requires each tenant to issue under their own OIB. The CEO decision parks multi-tenant for production, but the architecture must not hardcode assumptions that prevent it.

1.3 Compliance Blockers from Vlado Brkani? Memo (MC #103443)

- **B4 (CRITICAL):** UNKNOWN status from sveRačun means "still processing" — never auto-resubmit. Double fiscalization.
- **B5 (CRITICAL):** Invoice number reserved before submit, non-returnable even on failure. Gapless per fiscal year per issuer OIB.
- **B6 (CRITICAL):** Archive original fiscalized UBL XML bytes with integrity proof, 11 years immutable.
- **B1/B2 (PARKED):** ALAI legal status as HR OIB holder and PostLink intermediary contract — separate legal/commercial track. Out of scope for this build.

---

2. Decisions

2.1 The IssuerProfile Abstraction (Zachariadis Model C)

Decision: Introduce `IssuerProfile` as the single abstraction for "who is the legal sender and what credentials does the system use." The adapter and service NEVER read global env vars for sender

identity after this change. The demo is built on this abstraction with a single ALAI profile.

Rationale: The PostLink `companyVatNumber` header is already architecturally separate from the `Authorization` API key header. The IssuerProfile abstraction now means the production multi-tenant path is a credential-config change, not a code rewrite.

```
data class IssuerProfile(
    val profileId: UUID,
    val orgId: UUID,                    // FK to organizations.id
    val legalSenderOib: String,         // HR-prefixed, e.g. HR91276104352
    val legalSenderName: String,
    val submissionMode: SubmissionMode, // DIRECT | INTERMEDIARY
    val apiKeySecretRef: String,        // GCP Secret Manager path — NEVER the raw key
    val sveRacunBaseUrl: String,        // TEST or PROD endpoint
    val intermediaryOib: String? = null,
    val posrednikRef: String? = null,
    val enabled: Boolean
)

enum class SubmissionMode { DIRECT, INTERMEDIARY }
```

For demo: one row, `submissionMode = DIRECT`, `legalSenderOib = HR91276104352`, `apiKeySecretRef = "projects/.../secrets/bilko-sveracun-test-api-key/versions/latest"`.
For production: per-tenant rows, `submissionMode = INTERMEDIARY`, shared platform key, per-tenant `legalSenderOib`.

2.2 Adapter Refactor: IssuerProfile Injection Over Env Vars

Decision: `SveRacunHrEInvoiceAdapter.serialize()` receives `senderOib: String` explicitly. `SveRacunHttpClient` is instantiated with per-profile `apiKeyOverride` and `senderVatOverride`. The global-env-var constructor path is preserved for tests only; the production code path always resolves via `IssuerProfile`.

2.3 Retry Policy: Split Send-Path from Poll-Path

Decision: `SveRacunHttpClient` will use TWO separate `HttpClient` instances: one for the send path with `maxRetries = 0`, one for the poll path with `maxRetries = 3` and exponential backoff.

Rationale: This is the Kleppmann non-negotiable. The current single `HttpClient` with global retry is a live bug. Splitting into two instances is the cleanest fix without touching retry configuration in a way that could be accidentally reverted.

```
// Send path — zero retries; double-submit is a tax offence
private val sendClient = HttpClient(sendEngine) {
    install(HttpTimeout) { requestTimeoutMillis = TIMEOUT_MS; connectTimeoutMillis = 10_000L }
    // NO HttpRequestRetry installed
}

// Poll path — safe to retry; reads are idempotent
private val pollClient = HttpClient(pollEngine) {
    install(HttpTimeout) { requestTimeoutMillis = TIMEOUT_MS; connectTimeoutMillis = 10_000L }
    install(HttpRequestRetry) {
        maxRetries = MAX_RETRIES
        retryOnServerErrors(maxRetries = MAX_RETRIES)
        exponentialDelay(base = 2.0, maxDelayMs = 8_000L)
    }
}
```

2.4 State Machine (Canonical Definition)

The canonical state machine for an HR eRačun submission in Bilko. All service code and all DB column values reference these states and only these states.

State	`internal_status`	`sveracun_document_id`	Meaning
NOT_SUBMITTED	NULL (no row)	NULL	Invoice exists; no submission row
NUMBER_RESERVED	`NUMBER_RESERVED`	NULL	Fiscal number locked; XML serialized; GCS written; HTTP not yet called
SUBMITTED	`SUBMITTED`	`<docId>`	HTTP 200 + documentId received and persisted
SUBMIT_UNCERTAIN	`SUBMIT_UNCERTAIN`	NULL	Sent (maybe); no documentId received (timeout / conn err / no docId in 200 body)
PENDING	`PENDING`	`<docId>`	sveRačun still processing (UNKNOWN or null external)
ACCEPTED	`ACCEPTED`	`<docId>`	Terminal success: internal=OK + external=FISCALIZATION:OK

State	`internal_status`	`sveracun_document_id`	Meaning
REJECTED	`REJECTED`	`<docId>` or NULL	Terminal failure: FAILED/UNDELIVERABLE/FISCALIZATION:ERROR/4xx etapa-1

Legal transitions (one-way; no backwards, no auto-resubmit):

```
NOT_SUBMITTED    -> NUMBER_RESERVED
NUMBER_RESERVED  -> SUBMITTED | SUBMIT_UNCERTAIN | REJECTED
SUBMITTED        -> PENDING | ACCEPTED | REJECTED
PENDING          -> ACCEPTED | REJECTED | PENDING (keep polling)
SUBMIT_UNCERTAIN -> SUBMITTED (reconcile found docId) | REJECTED (confirmed not found)
ACCEPTED         -> (terminal, immutable)
REJECTED         -> (terminal; operator action + new fiscal number required for re-send)
```

Forbidden transitions:

- `SUBMITTED -> NUMBER_RESERVED` (never)
- `ACCEPTED -> anything` (immutable terminal)
- `REJECTED -> SUBMITTED` (no auto-resubmit; operator must issue new fiscal number)

Note on naming alignment: Momjian uses APPROVED where Kleppmann uses ACCEPTED. The ADR adopts ACCEPTED to align with EU e-invoicing terminology and the DB CHECK constraint in V77. The adapter's EInvoiceStatus.APPROVED is the adapter-interface value; the service layer translates it to the ACCEPTED DB state.

2.5 Persist-Before / Persist-After Protocol (Kleppmann Non-Negotiable #3)

Every submit call follows this exact ordering:

BEFORE the HTTP call — one DB transaction:

1. `SELECT ... FOR UPDATE` on the invoice row (concurrent-submit guard)
2. Check `internal_status NOT IN (NUMBER_RESERVED, SUBMITTED, SUBMIT_UNCERTAIN, PENDING)` — reject 409 CONFLICT if already in flight
3. `UPSERT` `hr_einvoice_number_counters` and `SELECT ... FOR UPDATE` to allocate next fiscal number (gapless, Momjian §1)
4. Compute `idempotencyKey = SHA-256(orgId + "|" + invoiceId + "|" + fiscalInvoiceNumber)`
5. Call `adapter.serialize(invoice, senderOib = issuerProfile.legalSenderOib)` to build UBL XML bytes
6. Compute `sha256Hex = SHA-256(xmlBytes)` (hex string)

7. Write XML bytes to GCS at `{orgId}/{fiscalYear}/{fiscalInvoiceNumber}/{submissionId}.xml` (write-once; must succeed before row insert)
8. INSERT `hr_einvoice_submissions` row with `internal_status = NUMBER_RESERVED`
9. COMMIT

HTTP call (outside any transaction):

- `sendClient.sendDocument(xmlBytes)` — NO retry

AFTER the HTTP call — separate DB transaction:

- Case A (HTTP 200 + documentId): UPDATE `internal_status = SUBMITTED`, `sveracun_document_id = docId`
- Case B (HTTP 200 no docId, OR timeout, OR connection error): UPDATE `internal_status = SUBMIT_UNCERTAIN`
- Case C (HTTP 4xx): UPDATE `internal_status = REJECTED`, `last_error = body`

2.6 OIB Binding Invariant (Tabriz Non-Negotiable)

The service layer (`HrEInvoiceService.submitInvoice()`) MUST enforce this invariant before any HTTP call:

```
require(invoice.organizationId == principal.organizationId) { "Invoice org mismatch" }
require(issuerProfile.orgId == principal.organizationId) { "Credential org mismatch" }
require(issuerProfile.legalSenderOib == xmlSenderOib) { "OIB binding violated" }
```

If any assertion fails: HTTP 422, write `LoggedAction` with `event = "hr_einvoice_oib_binding_violation"`, do NOT proceed. A broken OIB binding causes Bilko to file a fiscalized tax document under the wrong entity's identity with Porezna uprava — that is tax fraud.

2.7 UNIQUE(invoice_id) on hr_einvoice_submissions (Momjian Non-Negotiable)

The `UNIQUE (invoice_id)` constraint in migration V77 is the architectural load-bearing constraint for this feature. It must be present in the migration before any submission code is merged. Any code path that attempts to create a second active submission row for the same invoice receives a unique constraint violation — the DB-level guard for double-fiscalization even if service-layer checks have a bug.

---

3. Target Architecture

3.1 Layered View

```
[HTTP Route]
  POST /invoices/{id}/submit-to-sveracun
  GET  /invoices/{id}/sveracun-status
  GET  /invoices/{id}/sveracun-xml        (admin debug)
  POST /invoices/{id}/poll-sveracun-status (manual poll trigger)
       |
       | JWT principal -> requirePermission("sveracun:submit")
       | organizationId from JWT (never from request body)
       v
[HrEInvoiceService]
  - IssuerProfileRepository.findByOrgId(orgId) -> IssuerProfile
  - OIB binding invariant assertion (hard, not soft)
  - Persist-before-tx (number allocation, XML serialize, GCS write, DB insert)
  - SveRacunHttpClient(apiKeyOverride, senderVatOverride) — per-profile instantiation
  - SveRacunHrEInvoiceAdapter.submit(xmlBytes, invoice, senderOib) — NO retry
  - Persist-after-tx
  - HrEInvoiceNumberService.reserveNextNumber(orgId, issuerOib, fiscalYear)
  - LoggedAction audit write per submit
       |
       v
[SveRacunHrEInvoiceAdapter]  (already implemented; adapter-level changes only)
  - serialize(invoice, senderOib: String)  — senderOib injected, not from env
  - submit(xmlBytes, invoice) -> SubmitResult
  - pollStatus(documentId, invoice) -> EInvoiceStatus
  - mapStatusPair() (unchanged — correct per MC #103445)

[SveRacunHttpClient]  (two client instances after fix)
  - sendClient (NO retry) -> sendDocument()
  - pollClient (retry OK) -> getInternalStatus(), getExternalStatus()

[Postgres — four new tables via Flyway V75-V78]
  hr_einvoice_issuer_config        (IssuerProfile persistence; one row for demo)
  hr_einvoice_number_counters      (gapless fiscal year sequence via FOR UPDATE)
  hr_einvoice_submissions          (submission lifecycle; UNIQUE(invoice_id))
```

```
  hr_einvoice_archive             (integrity manifest; INSERT-only; points to GCS)

[GCS — bilko-hr-einvoice-archive-{env}]
  - Write-once per submission at NUMBER_RESERVED (before HTTP call)
  - Integrity verified on retrieve (SHA-256 re-hash comparison)
  - Retention policy: 4015 days LOCKED (11 years WORM) for prod bucket
  - Demo bucket: same write-once pattern; 90-day retention (not locked)
```

3.2 IssuerProfileRepository Interface

```
interface IssuerProfileRepository {
    fun findByOrgId(orgId: UUID): IssuerProfile?
}

// Demo implementation: reads from DB table hr_einvoice_issuer_config (V75 migration)
class DbIssuerProfileRepository(
    private val secretManager: GcpSecretManagerClient
) : IssuerProfileRepository {
    override fun findByOrgId(orgId: UUID): IssuerProfile? {
        // SELECT from hr_einvoice_issuer_config WHERE org_id = ? AND enabled = true
        // Resolve apiKey from GCP Secret Manager by api_key_secret_ref
    }
}
```

For demo: one row in `hr_einvoice_issuer_config` with `enabled = false` by default. A manual `UPDATE ... SET enabled = true` plus `SVERACUN_HR_LIVE = true` env flip is required to activate live submit. Two explicit gates, both required, neither accidental.

3.3 Route Pattern (Mirrors SefRoutes.kt)

```
fun Route.sveRacunRoutes() {
    val service by di<HrEInvoiceService>()

    post("/invoices/{id}/submit-to-sveracun") {
        val principal = call.principal<BilkoPrincipal>()!!
        if (requirePermission(principal, "sveracun:submit")) return@post
        val invoiceId = call.parameters["id"] ?: ...
        val organizationId = principal.organizationId   // from JWT, never from request
        try {
```

```
            val result = dbQuery { service.submitInvoice(invoiceId, organizationId, principal)
}
            call.respond(HttpStatusCode.OK, mapOf(...))
        } catch (e: OibBindingException) { call.respond(422, ...) }
          catch (e: NotFoundException) { call.respond(404, ...) }
          catch (e: ConflictException) { call.respond(409, ...) }
    }

    get("/invoices/{id}/sveracun-status") { /* requirePermission("sveracun:status") */ }
    get("/invoices/{id}/sveracun-xml") { /* admin only; verify SHA-256 before serving */ }
    post("/invoices/{id}/poll-sveracun-status") { /* manual trigger for demo */ }
}
```

3.4 Persistence Schema — Flyway V75–V78

V75 — hr_einvoice_issuer_config

Per-tenant IssuerProfile persistence. One row for demo (ALAI, DIRECT mode). RLS on `org_id`. The `api_key_secret_ref` column stores the GCP Secret Manager resource name — the raw API key is never stored in the DB.

Key columns: `org_id UUID NOT NULL`, `issuer_oib VARCHAR(13) NOT NULL`, `api_key_secret_ref VARCHAR(1024) NOT NULL`, `api_base_url VARCHAR(500) NOT NULL DEFAULT 'https://test.sveracun.hr/api'`, `submission_mode VARCHAR(20) NOT NULL DEFAULT 'DIRECT'`, `enabled BOOLEAN NOT NULL DEFAULT FALSE`.
Constraint: `UNIQUE (org_id, issuer_oib)`.

V76 — hr_einvoice_number_counters

Gapless fiscal year invoice number counter. One row per `(org_id, issuer_oib, fiscal_year)`. Allocated via `SELECT ... FOR UPDATE` inside the BEFORE transaction. Never decrements. Numbers are non-returnable even on submission failure.

Key columns: `org_id UUID NOT NULL`, `issuer_oib VARCHAR(13) NOT NULL`, `fiscal_year SMALLINT NOT NULL`, `last_number INTEGER NOT NULL DEFAULT 0`.
Constraint: `UNIQUE (org_id, issuer_oib, fiscal_year)`. Year rollover: automatic on UPSERT.

V77 — hr_einvoice_submissions

The submission lifecycle table. One row per invoice (`UNIQUE invoice_id`). Created at number-reservation time. Updated through polling until terminal.

Key columns: `org_id UUID NOT NULL`, `invoice_id UUID NOT NULL` (FK invoices.id ON DELETE RESTRICT), `fiscal_invoice_number VARCHAR(20) NOT NULL` (format `YYYY-NNNNNN`), `idempotency_key VARCHAR(64) NOT NULL`, `sveracun_document_id VARCHAR(255) NULL`, `internal_status VARCHAR(30) NOT

`NULL DEFAULT 'NUMBER_RESERVED'`, `xml_sha256_hex CHAR(64) NOT NULL`, `submitted_xml_gcs_path VARCHAR(1024) NOT NULL`, `submitted_by UUID NOT NULL`.

Critical constraints:

- `CONSTRAINT uq_hr_einvoice_submissions_invoice UNIQUE (invoice_id)` — THE load-bearing constraint; prevents double fiscalization at the DB layer
- `CONSTRAINT uq_hr_einvoice_submissions_idempotency UNIQUE (idempotency_key)`
- `CONSTRAINT uq_hr_einvoice_submissions_fiscal_number_org UNIQUE (org_id, fiscal_invoice_number)`
- `CONSTRAINT chk_hr_einvoice_internal_status CHECK (internal_status IN ('NUMBER_RESERVED','SUBMITTED','SUBMIT_UNCERTAIN','PENDING','ACCEPTED','REJECTED'))`

V78 — hr_einvoice_archive

Integrity manifest for 11-year UBL XML archival. Append-only. `bilko_app` role has INSERT-only grant (no UPDATE). All FKs are ON DELETE RESTRICT.

Key columns: `submission_id UUID NOT NULL` (FK, UNIQUE — one archive row per submission), `gcs_bucket VARCHAR(255)`, `gcs_object_path VARCHAR(1024)`, `sha256_hex CHAR(64) NOT NULL`, `retain_until DATE GENERATED ALWAYS AS ((archived_at AT TIME ZONE 'UTC')::DATE + INTERVAL '11 years') STORED`.

Archive is written AFTER `ACCEPTED` state is confirmed (internal=OK + external=FISCALIZATION:OK). The submitted XML written to GCS at `NUMBER_RESERVED` is the same bytes; the archive row formalizes it as the compliance record.

RLS on all four tables: Standard Bilko pattern from V46/V55. `USING (org_id = NULLIF(current_setting('app.current_org_id', true), '')::UUID)`. FORCE ROW LEVEL SECURITY on all tables. Phase 2C RESTRICTIVE mode activation is a prod prerequisite.

3.5 GCS Archival

Bucket: `bilko-hr-einvoice-archive-{env}` (e.g. `bilko-hr-einvoice-archive-demo`).
Object path: `{org_id}/{fiscal_year}/{fiscal_invoice_number}/{submission_id}.xml`.
Write timing: at `NUMBER_RESERVED`, before HTTP call. Same bytes sent to sveRačun.
Write-once enforcement: Cloud Run SA has `storage.objects.create` only; `storage.objects.delete` denied.
Prod bucket: retention policy `4015 days` LOCKED (WORM).
Demo bucket: same write-once pattern; retention 90 days (not locked).

Integrity verification on every retrieval via `/invoices/{id}/sveracun-xml`: fetch `sha256_hex`, download GCS bytes, recompute SHA-256, assert equals. If mismatch: HTTP 500 `ARCHIVE_INTEGRITY_FAILURE`, alert, do not serve bytes.

3.6 Audit Trail

Every submit, poll, and OIB-binding-violation event writes to `LoggedAction` (existing append-only table). Log structural/operational metadata only — do NOT log invoice line items, buyer/seller names, amounts, tax IDs, IBAN, API key value, or raw XML body (GDPR + Croatian tax secrecy).

---

4. Demo vs Production Boundary

"No hacks" means the demo is built on the real schema, real idempotency, real OIB binding invariant, and real state machine — with one issuer instead of many. The demo is not a prototype. It is the production system at scale=1.

Capability	Demo (build now)	Prod (parked / future)
IssuerProfile abstraction	YES — one ALAI/DIRECT profile in DB	Same table; N tenant rows; INTERMEDIARY mode
Schema V75-V78	YES — full schema from day one	Same migrations; no change
OIB binding invariant	YES — enforced at service layer	Same code; more profiles
UNIQUE(invoice_id) on submissions	YES — in V77 before any submit code	Same constraint
Retry-fix on send path	YES — sendClient (no retry)	Same fix
Persist-before/after protocol	YES — full protocol	Same protocol
SUBMIT_UNCERTAIN state	YES — must be representable	Same state
GCS write at NUMBER_RESERVED	YES — write-once, SHA-256	Same; LOCKED retention policy added
Gapless numbering (FOR UPDATE)	YES — counter table V76	Same; per-tenant issuer_oib separates sequences
HR einvoice archive row (V78)	YES — written on ACCEPTED	Same; 11-year LOCKED policy for prod
sveRačun base URL	TEST (test.sveracun.hr)	PROD (hr.sveracun.hr)
SVERACUN_HR_LIVE gate	Explicit flip required (default false)	PROD env flag; separate secret
IssuerProfile.enabled gate	Explicit DB update required	Same; per-tenant enable flow
Background poll worker	Manual: POST /invoices/{id}/poll-sveracun-status	Scheduled job (Cloud Run Job or scheduler)
GCS retention policy	90 days (demo bucket; not locked)	4015 days LOCKED (WORM)
RLS mode	PERMISSIVE (current ADR-017 state)	RESTRICTIVE (Phase 2C; Securion gate)
PostLink posrednik contract (B2)	Not required; DIRECT mode	Required before multi-tenant; legal track
ALAI Norwegian entity HR OIB (B1)	Not required; using existing TEST creds	Legal confirmation required

Capability	Demo (build now)	Prod (parked / future)
Credit note (InvoiceTypeCode 381)	Not built; domain model records the type	Must be built for full B2B accounting
Rate limiting (durable)	In-memory sliding window; 10/min, 100/day per org	Redis-backed (Cloud Memorystore)

Items NOT Deferred (frequently deferred in prototype builds; not here)

1. Flyway migrations V75-V78 — schema before any submit code
2. The `UNIQUE (invoice_id)` constraint — non-negotiable from the first migration
3. The retry fix on `sendDocument()` — before any live call, including TEST
4. The OIB binding invariant — runtime enforcement, not just a comment
5. The GCS write at NUMBER_RESERVED — even for demo; write-once pattern identical to prod
6. The `SUBMIT_UNCERTAIN` state — sveRačun TEST is not perfectly reliable
7. `LoggedAction` audit write per submit

---

5. Phased Build Plan (7 Work Packages)

WP1 — Foundation: Schema + Retry Fix + OIB Binding + IssuerProfile

Owner: CodeCraft (backend) | **Depends on:** None

Must land atomically — all in the same PR, before any route code.

1. Flyway migrations V75, V76, V77, V78 — all four tables with constraints, indexes, RLS, grants
2. `SveRacunHttpClient`: split into `sendClient` (maxRetries=0) + `pollClient` (maxRetries=3). Existing 42 tests remain green; add test asserting no retry on `sendDocument()` for 5xx
3. `IssuerProfile` data class + `SubmissionMode` enum
4. `IssuerProfileRepository` interface + `DbIssuerProfileRepository`
5. `SveRacunHrEInvoiceAdapter.serialize(invoice, senderOib: String)` — add `senderOib` param; remove `httpClient.configuredSenderVat` usage
6. `HrEInvoiceNumberService.reserveNextNumber(orgId, issuerOib, fiscalYear): String` — UPSERT + SELECT FOR UPDATE + increment

7. `OibBindingException` + `ConflictException` exception types

Acceptance criteria: All 42 existing adapter tests pass. Flyway migrate runs clean V74→V78. New test: `sendDocument()` with 5xx does NOT retry (exactly one call). New test: concurrent `reserveNextNumber()` produces distinct sequential numbers.

WP2 — Service + Persist Protocol: HrEInvoiceService

Owner: CodeCraft (backend) | **Depends on:** WP1

1. `HrEInvoiceService.submitInvoice()` — full persist-before/after protocol, OIB binding invariant, status gate (409 if in flight), IssuerProfile lookup, GCS write, LoggedAction
2. `HrEInvoiceService.pollAndUpdateStatus()` — only if SUBMITTED/PENDING/SUBMIT_UNCERTAIN; archive write on ACCEPTED
3. `HrEInvoiceService.getXmlForDownload()` — SHA-256 verification on every retrieval; 500 ARCHIVE_INTEGRITY_FAILURE on mismatch

Acceptance criteria: BEFORE tx written before HTTP call. AFTER tx reflects correct state for each case. OIB binding test: mismatched org → 422 + LoggedAction. Concurrent submit → one succeeds, one gets 409. SHA-256 mismatch on download → 500.

WP3 — Route: SveRacunRoutes

Owner: CodeCraft (backend) | **Depends on:** WP2

1. All four route handlers (thin layer over service, mirrors SefRoutes.kt)
2. Rate limit middleware: 10 submit requests/org/minute, 100/org/day (in-memory ConcurrentHashMap sliding window)
3. Mount in Application.kt alongside `sefRoutes()`

Acceptance criteria: Unauthenticated → 401. Insufficient role → 403. Wrong org → 404 (not 403; no existence leak). Already SUBMITTED → 409. SVERACUN_HR_LIVE=false → 501. Rate limit: 101st submit in same day → 429.

WP4 — Infra: GCS Bucket + Secret Wiring

Owner: FlowForge (infra) | **Depends on:** WP1

1. Terraform: `bilko-hr-einvoice-archive-demo` GCS bucket — versioning, write-once IAM, 90-day lifecycle
2. Verify `bilko-sveracun-test-api-key` exists and Cloud Run SA has `secretmanager.versions.access`
3. Secret rotation runbook documented in BookStack

4. Terraform: `bilko-hr-einvoice-archive-prod` bucket definition (commented out; LOCKED retention command documented but not executed)

Acceptance criteria: `gcloud storage buckets describe bilko-hr-einvoice-archive-demo` shows versioning=enabled and no delete in bilko-api SA binding. CI integration test: `DbIssuerProfileRepository.findByOrgId(DEMO_ORG_ID)` resolves non-null API key. Terraform plan = zero diff after apply.

WP5 — Dead Code Removal

Owner: CodeCraft (backend) | **Depends on:** WP3

1. Delete `StorecoveHrFiskEInvoiceAdapter.kt` (652 lines, abandoned provider, confirmed CEO decision MC #8675)
2. Remove DI wiring, test references, import statements

Acceptance criteria: `./gradlew build` passes with zero Storecove warnings. `grep -r "StorecoveHrFisk" apps/api/src` returns zero results.

WP6 — Proveo E2E Validation

Owner: Proveo (Angie Jones) | **Depends on:** WP3, WP4

1. Submit a real invoice through the route (SVERACUN_HR_LIVE=true, TEST env, IssuerProfile.enabled=true)
2. Assert HTTP 200 + non-null documentId received and persisted in DB
3. Assert GCS object exists and SHA-256(GCS bytes) == xml_sha256_hex from DB
4. Trigger poll; assert status transitions (PENDING → ACCEPTED on TEST env)
5. Verify status and XML download routes
6. Security checks: wrong orgId → 404; already SUBMITTED → 409; invalid OIB → 422; unauthenticated → 401
7. Rate limit: 101st submit → 429
8. Audit: LoggedAction row present with correct event, no PII in values
9. Verify zero retry attempts on sendDocument() via structured log count

Acceptance criteria (PASS/FAIL; no partial credit): Real sveRačun TEST HTTP 200 + documentId. GCS object written and SHA-256 verified. All security checks return expected codes. No PII in LoggedAction. Zero retries on send. No StorecoveHrFisk references in deployed artifact.

WP7 — BookStack Documentation

Owner: Skillforge | **Depends on:** WP6 (Proveo validation passed)

1. This ADR page (published)

2. BookStack page: "HR eRačun — Prod Prerequisites Checklist" (Bilko book, Legal & Compliance chapter) — B1/B2 legal track, Phase 2C RLS activation gate, GCS LOCK command, PostLink posrednik contract steps, Securion gate checklist

---

6. Open Questions for PostLink (Zachariadis Carry-Forward)

Must be answered before any production activation. Parked in the prod track.

#	Question
Q1	**Posrednik / Intermediary Model:** Does sveRačun support an intermediary registration where a single API key holder (Bilko) is authorised to submit on behalf of multiple sender OIBs? If yes: is registration self-service via API or manual per-sender?
Q2	**companyVatNumber Header Semantics:** The existing API separates Authorization (API key) from companyVatNumber (sender OIB). Is this header already the posrednik mechanism, or is etapa-1 currently hardcoded to reject unless the two match?
Q3	**PROD API Credentials:** Rate limits on PROD vs TEST. Is the PROD auth scheme identical? Is there a staging environment with real OIBs but test FINA fiscalization path?
Q4	**Fiscalization Identifier:** When FISCALIZATION:OK is returned, does the response body include a FINA fiscal identifier (ZKI/JIR equivalent)? Field name? Must Bilko store and display it?
Q5	**REJECTION_REPORT Payload:** What structured data is in FISCALIZATION_REJECTION_REPORT? Rejection reason code and free text?
Q6	**Document Retrieval API:** Does sveRačun provide a GET /documents/{id}/download endpoint? Critical for SUBMIT_UNCERTAIN reconciliation path.
Q7	**List by Sender Reference:** Can Bilko query sveRačun for all documents submitted by sender OIB X in the last N hours? Required for SUBMIT_UNCERTAIN reconciliation when no documentId was received.
Q8	**Norwegian Entity Eligibility (B1):** Is ALAI Holding AS (Norwegian org.nr, holding HR OIB HR91276104352) eligible as a platform intermediary under PostLink's terms?
Q9	**Pricing:** Per-document pricing for an intermediary platform account. Setup fee per registered sender OIB.

7. Risk Register

Risk	Probability	Impact	Mitigation
Crash between HTTP 200 and AFTER tx (Kleppmann §5)	Low (Cloud Run reliability)	CRITICAL	Clarify Q7 (list-by-reference API) with PostLink. Admin recovery endpoint in WP2 as fallback. Document the gap explicitly.
sveRačun TEST API unreliable during demo	Medium	HIGH	SUBMIT_UNCERTAIN state is representable; demo recovery endpoint allows operator to manually enter docId. Brief the demo presenter.
UNIQUE(invoice_id) constraint blocks a legitimate re-send after REJECTED	Low (by design)	Low	Service layer must support soft-delete of REJECTED row + insert of new row with new fiscal number + incremented attempt_seq. Document the re-send flow.
GCS write fails between number allocation and HTTP call	Low	MEDIUM	If GCS write fails, rollback DB insert. Number is consumed (non-returnable per B5) but absence of submission row signals no send occurred.
Phase 2C RLS not activated before multi-tenant prod	Certain (currently PERMISSIVE)	CRITICAL for multi-tenant	Securion prod gate checklist (WP7 BookStack). Block prod activation on this item.
PostLink posrednik contract takes longer than expected	High (legal/commercial)	HIGH for multi-tenant; LOW for demo	Demo runs DIRECT mode; no contract required. Architecture does not change.
sveRačun PROD base URL differs in auth scheme	Unknown	MEDIUM	Q3 to PostLink. The baseUrlOverride + apiKeyOverride parameters allow runtime configuration without code change.

Risk	Probability	Impact	Mitigation
Double-fiscal number if FOR UPDATE not atomic in pgBouncer	Medium without care	CRITICAL	Use hr_einvoice_number_counters counter table with SELECT ... FOR UPDATE inside a transaction. pgBouncer transaction pooling mode is fine for FOR UPDATE (released at COMMIT).
Developer accidentally wires env-var path instead of IssuerProfile	Medium	HIGH	The serialize() signature change (WP1) removes httpClient.configuredSenderVat call; senderOib parameter is required (non-nullable). Caught at compile time.

---

8. Parked Items (Separate Strategic Decision Required)

- **B1:** Legal confirmation of ALAI Holding AS (Norwegian entity) as a valid HR OIB holder and eRačun issuer. Legal track; no code dependency.
- **B2:** PostLink intermediary (posrednik) contract, power-of-attorney template for each tenant, per-sender OIB registration. Commercial track; IssuerProfile abstraction already built (WP1).
- **InvoiceTypeCode 381 (credit note):** Zachariadis §4.2 is the authoritative spec. Separate MC.
- **TaxExemptionReason BT-120:** Required for EN 16931 business rules BR-E-10 and BR-Z-10 (0%/exempt VAT). Post-demo.
- **FISCALIZATION_REJECTION_REPORT workflow:** User-facing notification + credit note issuance path. Post-demo.
- **NOT_DELIVERED_REPORT distinct state:** Fiscalized-but-not-delivered accounting problem. Post-demo.
- **Background poll worker:** Cloud Run Job or Cloud Scheduler. Architecture designed for it (next_poll_at + partial index); not built in this sprint.
- **Phase 2C RLS RESTRICTIVE mode:** Securion gate before any multi-tenant prod activation. Currently PERMISSIVE (ADR-017).
- **Fiscal identifier storage (Q4):** If PostLink confirms ZKI/JIR equivalent on FISCALIZATION:OK, add fiscal_identifier column in V79 migration.
- **Redis-backed rate limiting:** In-memory acceptable for demo. Prod requires Cloud Memorystore (Redis) for durability across multiple Cloud Run instances.

---

9. Architectural Decisions Log (Conflict Resolutions)

State name: ACCEPTED vs APPROVED (Kleppmann vs Momjian).
Kleppmann uses ACCEPTED; Momjian uses APPROVED. Decision: DB column and CHECK constraint use ACCEPTED. EInvoiceStatus.APPROVED remains the adapter-interface value (matches existing interface); service translates to ACCEPTED when writing to DB. Rationale: ACCEPTED matches common EU e-invoicing terminology; APPROVED is the accounting approval concept (different thing).

Archive timing: at NUMBER_RESERVED vs at ACCEPTED (Tabriz vs Momjian).
Tabriz: write XML to GCS inside BEFORE transaction. Momjian: archive only after ACCEPTED. Decision: write XML bytes to GCS at NUMBER_RESERVED (Tabriz wins). Create hr_einvoice_archive integrity manifest row only at ACCEPTED (Momjian wins for the archive table write). Rationale: GCS object = bytes store (available from day one for recovery/audit); archive manifest = compliance record (formalized only when FISCALIZATION:OK confirmed). Both layers required.

SUBMIT_UNCERTAIN: Kleppmann has it; Momjian's original CHECK constraint omits it.
Decision: ADD SUBMIT_UNCERTAIN to V77 CHECK constraint. ADR replaces FAILED (Bilko-internal naming) with SUBMIT_UNCERTAIN (semantically precise for sveRačun poll-only model) and ACCEPTED (aligned with adapter interface). Full CHECK list: NUMBER_RESERVED, SUBMITTED, SUBMIT_UNCERTAIN, PENDING, ACCEPTED, REJECTED. FAILED is retired.

IssuerProfile in DB vs config file (Zachariadis vs simplicity).
Zachariadis recommends DB-backed IssuerProfile for demo. Decision: DB-backed from day one (Momjian V75 table). Rationale: single-row demo config in DB is trivial; gives RLS and audit from the start; is the same code path as multi-tenant production. A config-file implementation would need to be ripped out and replaced.

Petter Graff — Lead Architect, HR eRačun Architecture Team, 2026-06-11
Synthesized from inputs by Martin Kleppmann, Bruce Momjian, Markos Zachariadis, Parisa Tabriz.
MC #103453 (architecture documentation) | MC #103464 (build execution)

Revision #1
Created 2026-06-11 20:06:47 UTC by John
Updated 2026-06-11 20:06:48 UTC by John