Bilko RBAC -- Users / Roles / Permissions

Overview

Bilko uses a **flat RBAC model**: users have one role per organisation; roles map to a permission catalog via a DB seed table. Permission resolution is live from the database on every request (no JWT role claim for authorisation). The system was built in WP1 (MC #103141, branch `feat/rbac-wp1-permissions-catalog`).

Roles

Role	Level	Scope
`owner`	3	All permissions including billing, account deletion, user management
`admin`	2	All permissions except billing and account deletion; can manage users and roles
`accountant`	1	Create and manage financial records; cannot delete; cannot manage users
`viewer`	0	Read-only access; default for newly JIT-provisioned Entra users

Roles are stored in `users.role` (VARCHAR 50) with a CHECK constraint added in V67 limiting values to these four. Single role per user per organisation (multi-role/multi-org deferred, MC #103089).

Permissions Catalog (V67 — 52 keys)

Source: `apps/api/src/main/resources/db/migration/V67__rbac_permissions_catalog.sql` (commit 66629bd). The catalog is stored in the `permissions` table; all application code references permission keys as string constants.

Format: `<resource>:<verb>` enforced by a DB CHECK constraint (`permission_key_format`). Example keys by resource group:

Resource group	Example keys
Invoices	`invoice:read`, `invoice:create`, `invoice:update`, `invoice:delete`, `invoice:submit`
Expenses	`expense:read`, `expense:create`, `expense:update`, `expense:delete`
Contacts	`contact:read`, `contact:create`, `contact:update`, `contact:delete`
Transactions	`transaction:read`, `transaction:create`, `transaction:reconcile`
Reports	`report:read`, `report:export`
Settings / billing	`settings:read`, `settings:update`, `billing:read`, `billing:update`
Users	`users:read`, `users:manage`, `users:invite`
Account admin	`account:delete`
Documents	`document:read`, `document:upload`, `document:delete`
Articles / products	`article:read`, `article:create`, `article:update`, `article:delete`

Full 52-key baseline stored in: `apps/api/src/main/resources/rbac/requireRole-baseline-v67.tsv` (commit 0bf18fd, 51 data rows).

Role-to-Permission Seed (Strategy A — Flat Inheritance)

Source: `role_permissions` table seeded in V67. Each row: `(role, permission_key)`. No runtime inheritance logic — the seed embeds the full flattened set for each role.

Role	Permissions count	Principle
viewer	13	Read-only: all :read + :export keys
accountant	40	viewer permissions + create/update on financial resources; no delete, no user management
admin	49	accountant permissions + delete + user management; no billing:update, no account:delete
owner	52	All 52 permissions (complete set)

The seed exactly reproduces the behaviour of the legacy `requireRole()` numeric hierarchy — verified by 204 RbacMatrixTest cases (0 failures). No behaviour regression.

PermissionService — Live DB Resolution

Source: `apps/api/src/main/kotlin/no/alai/bilko/services/PermissionService.kt` (commit dee4fb1)

- Interface method: `fun resolve(role: String): Set<String>` (2 implementations: interface + `DbPermissionService`)
- Live DB query against `role_permissions` on every resolve call
- **Fail-closed:** if role is unknown or DB returns empty set, resolves to `emptySet()` — no permissions granted
- CEO OCD O1 decision: global per-role cache (4 known values); result cached per role string key. Per CEO spec, cache keyed `userId+role-version` was the ideal; current implementation uses global per-role cache (simpler, advisory gap noted in Proveo verdict)

BilkoPrincipal + requirePermission

Source: `apps/api/src/main/kotlin/no/alai/bilko/auth/BilkoPrincipal.kt` and `RbacHelper.kt` (commit dee4fb1)

- `BilkoPrincipal` carries `permissions: Set<String>` — resolved at authentication time via `PermissionService`
- `RoutingContext.requirePermission(permissionKey: String)` — Kotlin extension function; throws `ForbiddenException` (HTTP 403 `BILKO-AUTH-003`) if key not in principal's permission set; calls `AuthzAuditLogger`
- All 51 formerly-`requireRole()` call sites migrated to `requirePermission()` (17 route files, 0 residual `requireRole` in routes — verified by grep)
- `requireRole()` is kept as a thin compatibility shim (RbacHelper.kt)

Role-to-Permission Matrix

Permission key	viewer	accountant	admin	owner
`invoice:read`	Y	Y	Y	Y
`invoice:create`	-	Y	Y	Y
`invoice:update`	-	Y	Y	Y
`invoice:delete`	-	-	Y	Y
`invoice:submit`	-	Y	Y	Y
`expense:read`	Y	Y	Y	Y

Permission key	viewer	accountant	admin	owner
`expense:create`	-	Y	Y	Y
`expense:delete`	-	-	Y	Y
`users:read`	Y	Y	Y	Y
`users:manage`	-	-	Y	Y
`users:invite`	-	-	Y	Y
`billing:read`	-	-	-	Y
`billing:update`	-	-	-	Y
`account:delete`	-	-	-	Y
`settings:read`	Y	Y	Y	Y
`settings:update`	-	-	Y	Y
`report:read`	Y	Y	Y	Y
`report:export`	Y	Y	Y	Y
... (52 total)	Full catalog in V67 seed			

Full read-only matrix visible to admins/owners in the web admin UI at `/admin/users` (component: `lib/permissions.ts ROLE_PERMISSION_MATRIX`).

Authorization Audit Log

Source: `apps/api/src/main/kotlin/no/alai/bilko/auth/AuthzAuditLogger.kt` (commit dee4fb1)

- Every `requirePermission()` call logs an `authz_decision` event (SLF4J structured log)
- Log fields: `userId`, `orgId`, `permissionKey`, `granted` (boolean), `route`
- `RbacHelper.kt` references `AuthzAuditLogger` at 4 call sites (verified)

V67/V68 Migration Summary

Migration	Contents
V67 (`V67__rbac_permissions_catalog.sql`)	Creates `permissions` table (52 keys, format CHECK); `role_permissions` table with full 4-role seed; adds `users.role` CHECK constraint; GRANT SELECT to bilko_app; no RLS (global catalog)

Migration	Contents
V68 (`V68__rbac_user_provisioning.sql`)	Adds `users:manage` and `users:invite` permission keys; SECURITY DEFINER function `bilko_auth.provision_user_with_org(issuer, oid, email, fullName)` returning new user UUID; seeds new permissions to admin + owner roles

Test Coverage

- 204 RbacMatrixTest cases (all 51 call sites x 4 roles): 0 failures — `feat/rbac-wp1-permissions-catalog`
- 8 UserProvisioningWp2Test cases (T1–T8: JIT, admin CRUD, role guards, self-escalation block): PASS
- Total test suite: 2534 tests (1070 unit + 1283 integration + 181 web), 0 failures — WP5 E2E evidence `/tmp/evidence-103145`

Out of Scope (v1)

- Multi-role per user (single role per org; MC #103089)
- Multi-org membership (single org per user; MC #103089)
- ABAC / conditional permissions (e.g. "delete only own drafts")
- Accountant Portal multi-tier permissions (Collaborator/Approver roles from ACCOUNTANT-PORTAL-SPEC.md §2.2)

Revision #1
Created 2026-06-08 07:41:03 UTC by John
Updated 2026-06-08 07:41:03 UTC by John