

Bilko — Kako support agent radi (runbook, MC #106063)

Parent: MC #106049 (Support MVP) · **Ovaj task:** MC #106063 (E2) · **Agent:** Skillforge · **Datum:** 2026-07-20

Vidi također: [Faza A](#) (RBAC+RLS temelj), [Faza B](#) (ticket reply + GUC aktivacija), i ["Support sistem — arhitektura" \(D1\)](#) za mapu svih support mehanizama.

1. Kako se support agent provisionira

- Dedicated endpoint: `POST /admin/support/agents/invite` (`SupportAgentProvisioningRoutes.kt`, MC #106052/A3).
- **Ko može pozvati:** SAMO `platform_admin` (`requirePlatformAdmin()`) — isti širok gate koji koriste svi drugi platform-ops admin route-ovi. Nema support_agent-specifičnog uslova za KO smije pozvati novog agenta — samo za ŠTA taj agent, jednom kreiran, smije poslije vidjeti.
- **DTO:** `InviteSupportAgentRequest(email, fullName)` — SAMO ta dva polja. Nema `role` ni `organizationId` polja na DTO-u uopšte.
- **Rola i org su hardkodovani server-side:** rola je uvijek `"support_agent"`, org je uvijek interni org sa fiksnim imenom `"Bilko Internal – Support"` (seedovan V135 migracijom). Caller ne može cilje ni jedan ni drugi preko request body-ja.
- **Zašto ne postojeći generički `POST /admin/invitations`:** taj endpoint uvijek koristi `organizationId = principal.organizationId` (org pozivaoca), što nikad nije interni support org — nema načina da se kroz njega ubaci support_agent u interni org.
- **Aktivacija:** pozvana osoba se prijavljuje kroz ISTI, nepromijenjeni Entra JIT invite-accept put kao svaki drugi Bilko invite (`InviteService.acceptInviteInsideTransaction`, pozvano iz `AuthService.createSessionFromEntraIdToken`). Nema novog auth koda, nema service accounta. JWT nakon prijave nosi `role="support_agent"` kao bilo koja druga rola.

1a. Zatvorena rupa — #106071 generic-invite reject

Kad su `InviteService.VALID_ROLES` i `UserProvisioningService.VALID_ROLES` prošireni da uključe `support_agent` (da bi gornji dedicated endpoint mogao pozvati isti servisni sloj), TRI POSTOJEĆA generička route-a (gated samo sa `users:manage`, dakle bilo koji admin/owner SVOJE org-e) su tiho počela prihvatati `role="support_agent"` i preko sebe:

- `POST /admin/invitations` (`InviteRoutes.kt`)
- `POST /admin/users` i `PUT /users/:id/role` (`UserManagementRoutes.kt`)

Fix: nova funkcija `rejectSupportAgentRoleInGenericFlow(role)` u `RbacHelper.kt` — baca `ForbiddenException("SUPPORT_AGENT_ROLE_FORBIDDEN_HERE...")` na sva tri generička route-a PRIJE nego zahtjev stigne do servisnog sloja. `VALID_ROLES` ostaju široki (jer dedicated endpoint legitimno zove istu `createInvite()` funkciju) — reject je na route sloju, ne na servisnom, upravo na ta tri mjesta koja je Parisa našla, ne paralelna allowlist koja bi mogla driftati.

2. Jackson vs kotlinx.serialization — razriješena neslaganja (Momjian vs Parisa)

Zadatak je tražio da se PROČITA stvarni kod prije nego što se prepíše bilo čija tvrdnja.

Pročitano: `apps/api/src/main/kotlin/no/alai/bilko/plugins/Serialization.kt` i `SupportAgentProvisioningRoutes.kt` na branch-u `feat/106051-support-agent-rbac` (commit 5c06350a).

Nalaz: Bruce Momjian je bio u pravu — mehanizam je Jackson's `FAIL_ON_UNKNOWN_PROPERTIES` (default `true`, nikad eksplicitno postavljen niti isključen u kodu), NE `kotlinx.serialization ignoreUnknownKeys`.

Dokaz iz koda:

```
// Serialization.kt
fun Application.configureSerialization() {
    install(ContentNegotiation) {
        jackson {
            disable(SerializationFeature.INDENT_OUTPUT)
            registerModule(JavaTimeModule())
            disable(SerializationFeature.WRITE_DATES_AS_TIMESTAMPS)
            findAndRegisterModules()
        }
    }
}

val bilkoJson = Json {
    ignoreUnknownKeys = true    // <- postoji, ali NIJE registrovan u ContentNegotiation
    ...
}
```

- `install(ContentNegotiation) { jackson { ... } }` — Ktor-ov HTTP content-negotiation sloj je konfigurisan SAMO sa Jackson-om. Nema `json(bilkoJson)` ni bilo koje druge kotlinx registracije u tom bloku.
- DTO-ovi kao `CreateInviteRequest` i `InviteSupportAgentRequest` nose `@Serializable` anotaciju (kotlinx marker), ali se stvarno deserijalizuju preko `call.receive<T>()`, što ide kroz ContentNegotiation — dakle kroz Jackson, ne kroz kotlinx, bez obzira na anotaciju na DTO-u.
- `bilkoJson` (kotlinx `Json` instanca sa `ignoreUnknownKeys = true`) postoji u istom fajlu, ali služi za internu JSON obradu (komentar u kodu: "used internally for JSON parsing") — NIJE registrovana kao content-negotiation mehanizam za HTTP request body.
- Jackson-ov `ObjectMapper` (kreiran preko Ktor-ovog `jackson {}` DSL-a) ima `FAIL_ON_UNKNOWN_PROPERTIES` default `true` — i nigdje u repou (grep potvrđen, nula pogodaka) se to eksplicitno ne isključuje.
- **Live test-dokaz** (ne samo statika koda): `SupportAgentProvisioningTest.kt`, test `T06` "request body cannot override role or organizationId – DTO has no such fields, extra keys are rejected outright" — šalje JSON sa dodatnim `role/organizationId` poljima na `InviteSupportAgentRequest` DTO, unutar `testApplication {}` (stvarna Ktor test-aplikacija, ne mock), i tvrdi `HttpStatusCode.BadRequest`. Test-ov vlastiti komentar kaže eksplicitno: *"This codebase's Jackson ObjectMapper (Serialization.kt) does NOT disable FAIL_ON_UNKNOWN_PROPERTIES, so an unrecognized JSON key is rejected outright with 400."* Pozitivna kontrola u istom testu potvrđuje da isti email SA samo ispravnim poljima uspijeva (201) — dokazuje da je 400 specifično zbog nepoznatih ključeva, ne neke druge greške.

Zaključak za tim: kad god vidite `@Serializable` na DTO-u u ovom repou, to NE garantuje da će kotlinx pravila (npr. `ignoreUnknownKeys`) važiti na HTTP granici — treba provjeriti da li je taj DTO stvarno deserijalizovan preko ContentNegotiation-a (Jackson, ovaj repo) ili preko direktnog poziva na `bilkoJson.decodeFromString<T>()` (kotlinx, druga upotreba u istom fajlu).

3. Šta support agent VIDI

- SVI support tiketi, cross-org — preko `app.is_support_agent` GUC-a postavljenog samo na support ruta-ma, nakon što je `requirePermission(principal, "support_ticket:read") / "support_ticket:reply"` već prošao.
- Reply/triage polja na tiketu preko ograničenog UPDATE-a: `status`, `resolution_note`, `triage_json`, `external_ref`, `updated_at` — kolonski GRANT (V134 Part D), ne cijeli red.

4. Šta support agent NE VIDI

- **Finance/payroll/accounting/settings** — nula permisija dodijeljeno support_agent roli van `support_ticket:read / support_ticket:reply` (V133 Part C/D, provjereno grep-om — nema trećeg reda u `role_permissions` za tu rolu).

- **RLS-nivo dokaz (Faza A evidence, V134MigrationTest test 5):** sa

`app.is_support_agent=true` postavljenim, pokušaj pristupa `organizations`, `invoices`, `expenses`, `employees`, `payslips`, `financial_audit_log` — svih šest odbijeno. Dva različita, oba tačna načina odbijanja nađena: "permission denied" (`organizations/invoices/expenses` — nula GRANT za `bilko_app` na tim tabelama uopšte, prethodno postojeći, nepovezan gap) i "0 redova" (`employees/payslips/financial_audit_log` — GRANT postoji preko V101/V124, ali nijedna politika na tim tabelama ne konsultuje `is_support_agent`, pa RLS ispravno filtrira sve).

- **Ostali ticket route-ovi:** `support_agent` NE može pristupiti `GET /admin/support/tickets/{id}` ni `PATCH` preko svoje role — ti route-ovi ostaju gated sa `requireSupportTriageService()` (odvojen guard od `support_agent` role-flow-a, nepromijenjen ovim MVP-om).

5. RBAC permission-key referenca

Permission key	Značenje	Format
<code>support_ticket:read</code>	Čitanje/listanje tiketa preko svih organizacija	colon-format (resource:verb)
<code>support_ticket:reply</code>	Odgovor / triage tiketa (status + resolution_note) preko svih organizacija	colon-format (resource:verb)

Zašto colon, ne dot: dispatch je originalno tražio `support.tickets.read` / `support.tickets.reply` (dot-notation). V67-ova `permission_key_format` CHECK constraint (`key ~ '^[a-z_]+:[a-z_]+$'`) zahtijeva tačno jednu dvotačku, bez tačaka — svaki postojeći ključ u katalogu (npr. `invoice:read`, `expense:create`) već poštuje taj format. Umjesto proširenja constraint-a za jednu feature-preferencu imena, korišteno je `support_ticket:read` / `support_ticket:reply` (singular resource, isti obrazac).

6. RLS cross-org izuzetak — zašto i kako ograničen

Zašto postoji izuzetak uopšte: support osoblje MORA vidjeti tikete preko svih organizacija da bi triage funkcionisao — jedan support agent opslužuje sve klijente, ne samo jednu org. Standardni Bilko RLS model (`org_id-scoped`) bi u tom slučaju blokirao support agenta da vidi ijedan tiket van svoje (interne) org-e.

Kako je ograničen (ključna crvena-zona odluka, Momjian, MC #106051):

- **Nova, uska GUC:** `app.is_support_agent` — NIJE ponovna upotreba postojeće `app.is_platform_admin` GUC-e. `support_tickets` je već imao `support_tickets_admin_all`

politiku (V73) gated na `app.is_platform_admin` — ISTU GUC koju konsultuju i `purge_worker_log`, `document_retention_manifest`, `financial_audit_log`. Da je `support_agent`-ov cross-org pristup implementiran postavljanjem `app.is_platform_admin`, ista transakcija bi automatski zadovoljila i sve DRUGE tabele gated na tu GUC — tiho dodjeljujući `support` agentu pristup retention manifestima, purge logovima, finansijskim audit trailovima. Točno ono što ovaj task postoji da SPRIJEČI.

- **Sealed-class ekskluzivnost:** nova GUC se konsultuje SAMO od strane dvije nove politike na `support_tickets` (`support_tickets_support_agent_select`, `support_tickets_support_agent_reply`) — nikad od bilo koje druge tabele/politike u ovom repou. Verifikovano live testom koji grep-uje `pg_policies` katalog za string `"is_support_agent"` i tvrdi nula pogodaka van `support_tickets` (V134MigrationTest, "6 - GUC ISOLATION").
- **SAMO support_tickets + support_ticket_messages** — nikad finance. RLS izuzetak pokriva tačno dvije tabele iz ovog MVP-a; nijedna finance/payroll/accounting tabela nije dirana.
- **Ko postavlja GUC:** route sloj (`SupportTicketRoutes.kt`, B2), unutar `transaction {}` bloka, SAMO NAKON što je `requireSupportAgent()` već vratio validan principal (dakle samo za zahtjev koji je nezavisno dokazao `support_ticket:read/reply` permisiju) — GUC nikad sama nije autorizaciona odluka, samo downstream RLS mehanizam.
- **Provjera curenja kroz connection pool (top adversarial pitanje, B faza):** `SupportAgentGucPoolTest.kt` — HikariCP pool size 1 (forsira reuse konekcije) + `isAutoCommit = false` (isti kao produkcijska konfiguracija). Test 2: odmah nakon agent zahtjeva na ISTOJ pooled konekciji, sljedeća customer-scoped transakcija NE nasljeđuje cross-org vidljivost, i direktna `current_setting('app.is_support_agent', true)` provjera na početku te transakcije čita NULL/prazno. Test 4: `app.is_platform_admin` se nikad ne postavlja na `support_agent` putu — nema unakrsne kontaminacije GUC-ova.
- **Kolonski GRANT, ne FOR ALL:** `support_agent`-ov UPDATE je ograničen na tačno kolone koje postojeći PATCH triage flow piše (status, resolution_note, triage_json, external_ref, updated_at) — preko GRANT UPDATE (kolona-lista), isti mehanizam koji V125/V132 već koriste za `bilko_purge_worker`. Postgres nema kolonski-scoped RLS, pa RLS odlučuje KOJI REDOVI, GRANT odlučuje KOJE KOLONE.

7. Otvoreni tech-debt

- **#106070** — (prati se odvojeno, van scope-a ovog MVP-a; provjeriti `mc.js show 106070` za trenutni status prije referenciranja detalja).
- **#106072** — (isto, van scope-a; provjeriti `mc.js show 106072`).
- **#106075** — (isto, van scope-a; provjeriti `mc.js show 106075`).
- **Poznat, označen (ne skriven) gap iz B faze:** `support_tickets_customer_select` (V73) cast-uje `app.current_org_id` direktno na `::uuid` bez `NULLIF` guard-a, baca sirovi exception umjesto čistog deny kad je ta GUC zaista nepostavljena. Zaobiđeno u testu (postavlja throwaway validan UUID na agent putu, koji tu GUC nikad stvarno ne treba), NE popravljeno — van scope-a ovog taska, već praćeno.

- **Pre-existing GRANT gap na support_tickets (nađen i ispravljen u V134, ne bio prije praćen kao poseban ticket):** tabela nije imala eksplicitan GRANT SELECT/INSERT za `bilko_app` uopšte — nevidljivo u produkciji jer Flyway/API oboje konektuju kao `bilko_admin` (BYPASSRLS). V134 dodaje nedostajući GRANT, što čini i STARE V73 politike (customer/admin) stvarno dostupnim za `bilko_app` ulogu, ne samo nove.

8. Merge status (na dan pisanja ove stranice)

Branch `feat/106051-support-agent-rbac` (azdo), tip 5c06350a. NIJE merge-ovan u `azdo/main`. Faza A (Parisa adversarial verify) i Faza B (GUC-execution adversarial pass) oboje još čekaju Parisa Tabriz-ov nezavisni pregled prije merge-a — to je eksplicitni merge-gate koji su i Momjian i CodeCraft ostavili otvorenim u svojim verdiktima (vidi `~/system/evidence/106051/verdict.md` i `~/system/evidence/106055/verdict.md`).

Revision #4

Created 2026-07-20 09:07:17 UTC by John

Updated 2026-08-10 07:37:51 UTC by John