Security Architecture

Security Architecture Document

> **Project:** Bilko — Balkan Accounting SaaS **Version:** 1.0 **Date:** 2026-02-23 **Author:** Compliance Architect **Status:** Draft **Reviewers:** CTO, DPO, Engineering Lead **Classification:** Confidential

Document History

Version	Date	Author	Changes
0.1	2026-02-23	Compliance Architect	Initial draft — Bilko security architecture

---

1. Security Architecture Overview

Security Owner: Compliance Architect (security@bilko.io) **Last Security Review:** 2026-02-23 **Next Scheduled Review:** 2026-08-23 **Compliance Targets:** GDPR | Zakon o zaštiti podataka o ličnosti RS (ZZPL) | Zakon o zaštiti ličnih podataka BiH (ZZLP) | GDPR via AZOP (HR) | Zakon o računovodstvu RS/HR/BA | Zakon o PDV RS/BA/HR

Architecture Model: Bilko is a multi-tenant cloud accounting SaaS. Processes invoices, expenses, VAT returns, and financial reports for organizations in Serbia, Bosnia & Herzegovina, and Croatia. Each organization's data strictly isolated by `organizationId` at the database layer.

Defense-in-Depth Overview

```
graph TD
    CLIENT["Client Browser / PWA"]


    subgraph NETWORK["Network Layer"]
```

```
        CF["Cloudflare WAF\nDDoS Protection\nTLS 1.3 termination\nHSTS"]
    end

    subgraph APP_LAYER["Application Layer"]
        HELMET["Helmet.js\nCSP + X-Frame + HSTS\nno X-Powered-By"]
        CORS["CORS Whitelist\nbilko.io only\nno wildcard *"]
        RATE["Rate Limiter\nexpress-rate-limit\n5 req/15min auth\n100 req/15min general"]
        AUTH_MW["Auth Middleware\nJWT verify (15min access)\norg-scope injection"]
        RBAC_MW["RBAC Middleware\nowner / admin / accountant / viewer"]
        ZOD["Zod Validation\nall request bodies\ntype-safe parsing"]
    end

    subgraph DATA_LAYER["Data Layer"]
        PRISMA_ORM["Prisma ORM\nparameterized queries\nno raw SQL for user input\norg-scoped
WHERE clauses"]
        PG_ENC["PostgreSQL (Railway EU West)\nAES-256 disk encryption\nbackup encryption"]
    end

    subgraph AUDIT["Audit Layer"]
        LOG["LoggedAction table\nAPPEND-ONLY\nIP + user + timestamp\nold/new values
(changedFields)"]
    end

    CLIENT --> CF --> HELMET --> CORS --> RATE --> AUTH_MW --> RBAC_MW --> ZOD --> PRISMA_ORM
--> PG_ENC
    PRISMA_ORM --> LOG
```

---

2. Authentication

2.1 Strategy: JWT (JSON Web Tokens)

Stateless JWT, scales horizontally on Railway. Access tokens (15 min, memory-only) + refresh tokens (7 days, httpOnly cookie). Rotation on every refresh. Revocation via hashed token storage in DB.

2.2 JWT Auth Flow

```
sequenceDiagram
    actor User
    participant FE as Frontend (bilko.io — Vercel)
    participant API as Express API (api.bilko.io — Railway EU)
    participant DB as PostgreSQL (Railway EU West)

    User->>FE: Enter email + password
    FE->>API: POST /api/v1/auth/login
    API->>DB: SELECT user WHERE email = ? (parameterized)
    DB-->>API: User record (passwordHash)
    API->>API: bcrypt.compare(password, hash) — 12 rounds
    alt Password valid
        API->>API: jwt.sign({sub, org, role}, JWT_SECRET, 15m)
        API->>DB: INSERT refreshToken (hashed, expiresAt)
        API-->>FE: 200 { accessToken } + Set-Cookie: refreshToken (httpOnly, secure,
sameSite=strict)
        FE->>FE: Store accessToken in memory only
    else Password invalid
        API-->>FE: 401 Unauthorized (generic — no user enumeration)
    end

    Note over FE,API: 15 minutes later — access token expires
    FE->>API: POST /api/v1/auth/refresh (Cookie: refreshToken)
    API->>API: Rotate: delete old, issue new
    API-->>FE: 200 { newAccessToken } + Set-Cookie: newRefreshToken

    Note over User,DB: Logout
    FE->>API: POST /api/v1/auth/logout
    API->>DB: DELETE refreshToken WHERE userId = ?
    API-->>FE: 204 No Content
```

2.3 Two-Factor Authentication (2FA)

Method: TOTP (RFC 6238) — Google Authenticator, Authy, 1Password

- Setup: `POST /api/v1/auth/2fa/setup` → QR code + base32 secret
- Verify: `POST /api/v1/auth/2fa/verify { code }`
- Login: returns `{ requires2FA: true, tempToken }` → `POST /api/v1/auth/2fa/login`
- Backup: 10 single-use codes, bcrypt-hashed

---

3. Authorization (RBAC)

3.1 Role Permission Matrix

Action	owner	admin	accountant	viewer
Create invoice	✅	✅	✅	❌
Edit invoice	✅	✅	✅	❌
Delete invoice	✅	✅	❌	❌
View invoice	✅	✅	✅	✅
Approve expense	✅	✅	❌	❌
Generate report	✅	✅	✅	✅
Invite user	✅	✅	❌	❌
Edit org settings	✅	✅	❌	❌

3.2 Organization Scoping (IDOR Prevention)

```
// Injected by auth middleware on all /api/v1/* routes
app.use('/api/v1/*', (req, res, next) => {
  req.prismaWhere = { organizationId: req.user.organizationId };
  next();
});

// Applied to every Prisma query
await prisma.invoice.findMany({ where: { ...req.prismaWhere } });
```

UUID primary keys throughout — no sequential ID enumeration possible.

---

4. Encryption

4.1 In Transit: TLS 1.3

All traffic HTTPS. Cloudflare TLS 1.3 at edge, re-encrypted to Railway. HSTS: `max-age=63072000; includeSubDomains; preload`.

4.2 At Rest: AES-256

Store	Method	Location
PostgreSQL	AES-256 TDE (Railway)	Railway EU West (Frankfurt/Paris)
PostgreSQL backups	AES-256 auto-backup	Railway EU West — 30 days
Tax IDs (PIB/JMBG/OIB/JIB), IBAN	AES-256-GCM field encryption	Application layer — Railway env secret
Cloudflare R2 (receipts, PDFs)	AES-256 server-side	Cloudflare EU region

4.3 Password Security

bcrypt, 12 salt rounds. Min 8 chars. Block top 10K common passwords. Last 5 hashes retained.

4.4 Financial Data Precision

All monetary amounts: `NUMERIC(19,4)` — never float. Exchange rates locked at transaction date.

---

5. OWASP Top 10 Mitigations

OWASP Risk	Mitigation	Status
A01: Broken Access Control	RBAC + org-scoped WHERE + UUID PKs	Designed
A02: Cryptographic Failures	TLS 1.3 + AES-256 + bcrypt(12) + no PII in JWT	Designed
A03: Injection	Prisma ORM parameterized queries exclusively	Designed
A04: Insecure Design	Multi-tenant org isolation at DB layer, immutable audit	Designed
A05: Security Misconfiguration	Helmet.js, CORS whitelist (no *), sanitized errors	Designed
A06: Vulnerable Components	Dependabot + weekly npm audit + lock file	Planned
A07: Auth Failures	Rate limiting + JWT rotation + 2FA + bcrypt(12)	Designed
A08: Software Integrity	Signed commits + CI/CD + Dependabot	Planned

OWASP Risk	Mitigation	Status
A09: Logging Failures	Immutable LoggedAction table + Railway logs + Sentry	Designed
A10: SSRF	Zod validation + allowlist for SEF/HR-FISK/FINA API	Designed

6. Rate Limiting

Endpoint	Limit	Window
POST /api/v1/auth/login	5 req	15 min
POST /api/v1/auth/register	3 req	60 min
POST /api/v1/auth/refresh	10 req	15 min
GET /api/v1/reports/*	10 req	15 min
All other /api/v1/*	100 req	15 min

7. Input Validation (Zod)

```
const createInvoiceSchema = z.object({
  customerId: z.string().uuid(),
  invoiceDate: z.string().regex(/^\d{4}-\d{2}-\d{2}$/),
  dueDate: z.string().regex(/^\d{4}-\d{2}-\d{2}$/),
  currencyCode: z.enum(['EUR', 'RSD', 'BAM']),
  items: z.array(z.object({
    description: z.string().min(1).max(500),
    quantity: z.number().positive(),
    unitPrice: z.number().nonnegative(),
    taxRate: z.number().min(0).max(100),
  })),
});
```

8. File Upload Security

Allowed: JPG, PNG, PDF. Max 10 MB. MIME + extension validation. Stored in Cloudflare R2 EU. Phase 2: ClamAV scanning.

---

9. Audit Trail — LoggedAction (APPEND-ONLY)

Field	Description
eventId	Auto-incrementing
tableName	Mutated table
action	INSERT / UPDATE / DELETE
userId	Actor
actionTimestamp	UTC
rowData	Full row before mutation
changedFields	`{ field: { old: X, new: Y } }`
clientIp	Requester IP

On GDPR erasure: userId → `"deleted-user"`. Financial entries retained 11 years (law). LoggedAction never deleted.

---

10. Security Headers (Helmet.js)

Header	Value
Strict-Transport-Security	max-age=63072000; includeSubDomains; preload
Content-Security-Policy	default-src 'self'; script-src 'self' 'unsafe-inline'
X-Content-Type-Options	nosniff
X-Frame-Options	DENY
X-Powered-By	Removed

---

11. Pre-Launch Security Checklist

- ☐ JWT_SECRET generated (32+ chars, CSPRNG) — Railway env secret

- [ ] JWT_REFRESH_SECRET separate key (32+ chars)
- [ ] FIELD_ENCRYPTION_KEY generated (32 bytes hex) — for PIB/JMBG/OIB/JIB + IBAN
- [ ] HTTPS enforced
- [ ] CORS: bilko.io only
- [ ] Rate limiting tested
- [ ] Helmet.js headers verified
- [ ] bcrypt rounds = 12
- [ ] All Prisma queries use org-scoped WHERE
- [ ] Zod validation on all endpoints
- [ ] LoggedAction trigger active on all tables
- [ ] Error responses sanitized
- [ ] Dependabot alerts enabled
- [ ] Railway region = EU West confirmed
- [ ] DPAs signed (Railway, Vercel, Cloudflare, SendGrid)
- [ ] Data deletion workflow tested

Related Documents

- Compliance Framework: compliance-framework.md
- Data Encryption Policy: data-encryption-policy.md
- Key Management Policy: key-management-policy.md
- DPIA: data-protection-impact-assessment.md
- Breach Response: data-breach-response-plan.md
- Security Testing: security-testing-policy.md
- Bilko Security Docs: ../../products/Bilko/docs/security/

Approval

Role	Name	Date	Signature
Author	Compliance Architect	2026-02-23	
CTO			
DPO			

Role	Name	Date	Signature
Engineering Lead			

Revision #10
Created 2026-02-24 14:52:37 UTC by John
Updated 2026-05-25 07:32:17 UTC by John