Security Architecture Document

Security Architecture Document

> **Project:** Drop — Fintech Payment App (Remittance + QR Payments) **Version:** 1.0 **Date:** 2026-02-23 **Author:** ALAI Security Team **Status:** Draft **Reviewers:** CISO, CTO, DPO **Classification:** Confidential

Document History

Version	Date	Author	Changes
0.1	2026-02-12	Security Agent (ALAI)	Initial security audit
1.0	2026-02-23	Security Architect (ALAI)	Architecture documentation

---

1. Security Architecture Overview

Security Owner: CISO (Alem Bašić / ALAI Holding AS) **Last Security Review:** 2026-02-12 (full audit); 2026-02-13 (hardening verification) **Next Scheduled Review:** 2027-02-12 (annual) or after Phase 2 integration (BankID, Open Banking) **Compliance Targets:** GDPR | PSD2 (Betalingstjenesteloven) | AML/Hvitvaskingsloven | DORA/IKT-forskriften | Finanstilsynet license

Architecture Model: Drop operates a **PSD2 pass-through model**. Drop never holds customer funds. AISP reads bank balances via Open Banking; PISP initiates payments directly from the user's bank account. Cards are a future feature, gated behind feature flags (all default to `false`).

Security Posture Summary (post-hardening 2026-02-13):

- 0 Critical findings remaining (all 4 resolved)
- 0 High findings remaining (all resolved)
- 2 Medium findings remaining (CSP `unsafe-inline`, proxy X-Forwarded-For trust)
- 4 Low findings acknowledged (out of scope for current MVP sprint)

Defense-in-Depth Overview

```
Internet
  → WAF (planned — Phase 2 infra hardening)
  → CDN / Edge TLS termination (planned)
  → Load Balancer (TLS 1.3)
  → Application Layer (Next.js — Next.js 16.1.6)
      ├── Rate Limiting (SQLite-backed, persistent)
      ├── Origin/CSRF validation
      ├── JWT auth + session revocation
      ├── RBAC (user / merchant roles)
      ├── Input validation + sanitization
      └── Parameterized SQL queries
  → Database Layer (SQLite — MVP; PostgreSQL planned Phase 2)
      └── Stored: bcrypt(password), session token hashes, masked card tokens

Monitoring: Sentry (error tracking) — SIEM planned Phase 3
```

---

2. Authentication Flows

2.1 Current MVP Authentication (Email + Password)

```
User → POST /api/auth/login {email, password}
     → Rate limit check (10 req/60s per IP — SQLite-backed)
     → SELECT user WHERE email = ?
     → bcrypt.verify(password, hash) [cost factor 12]
     → If valid: generate JWT (HS256, jose ^6.1.3, 24h expiry)
     → INSERT into sessions table (token_hash = SHA-256(token))
     → Set httpOnly cookie (secure:true, sameSite:strict, maxAge:24h)
     → Return 200
```

Source: `src/drop-app/src/lib/auth.ts`, `src/drop-app/src/lib/middleware.ts`

2.2 Session Lifecycle

Step	Action	Source
Login	Session created, token_hash stored	`auth.ts:56-65`
Each request	Session checked for revocation	`middleware.ts:66-74`
Logout	All user sessions revoked server-side	`auth/logout/route.ts:5-14`
Password change	All sessions revoked	Planned (Phase 2)

Session table schema:

Column	Type	Purpose
`id`	TEXT PK	`ses_<hex16>` format
`user_id`	TEXT FK	References `users.id`
`token_hash`	TEXT	SHA-256 of JWT token
`expires_at`	TEXT	Expiration timestamp
`revoked`	INTEGER	0 = active, 1 = revoked

2.3 BankID OIDC Authentication (Phase 2 — Planned)

BankID is not yet integrated in the MVP codebase. Required for PSD2 SCA compliance before any live transactions. Planned integration:

```
User → Drop App → BankID OIDC (nivå høyt — eIDAS Level High)
     → BankID returns: name, fødselsnummer (national ID), verified identity
     → Drop validates: age >= 18 (from fødselsnummer), Norwegian residency
     → Dynamic linking for payment authorization (amount + payee bound to auth)
```

Regulatory requirement: PSD2 (Betalingstjenesteloven §§ 4-28, 4-29) requires SCA with two of three factors. BankID covers possession + knowledge. No live transactions without this.

Integration partner: TBD — BankID Norge AS (DPA required)

2.4 KYC Flow (Phase 2 — Planned via Sumsub)

Current state: Mock KYC with auto-approve (`kyc_status` field in `users` table). Production will use Sumsub:

```
User → Sumsub SDK → Document scan + liveness check
     → Sumsub webhook → Drop backend
```

```
→ Update users.kyc_status = 'approved'/'rejected'
→ Required for: remittance transactions
```

Source: `legal/dpa-sumsub.md`, `legal/dpia-vurdering.md`

---

3. Authorization Model

Model: RBAC (Role-Based Access Control) with resource-level user scoping

3.1 Roles

Role	Description	Access
`user`	Standard registered user	Own data only (transactions, recipients, notifications, settings)
`merchant`	Merchant with QR payment dashboard	Own data + merchant dashboard (`/api/merchant/*`)

KYC Status (enforced gate for financial operations):

Status	Meaning	Effect
`pending`	Default on registration	Cannot initiate remittance
`approved`	KYC completed	Full access to financial features
`rejected`	KYC failed or blocked	Blocked from all financial operations

3.2 Resource-Level Access Control (IDOR Prevention)

All data access queries include `AND user_id = ?` to scope data to the authenticated user. Applied to:

- `recipients` — scoped to user
- `transactions` — scoped to user
- `bank_accounts` — scoped to user
- `notifications` — scoped to user
- `settings` — scoped to user
- `cards` — scoped to user (future feature)

Merchant endpoints verify both merchant role and ownership.

Source: `src/drop-app/src/app/api/` — all route handlers

3.3 Permission Summary

Resource	`user`	`merchant`	Admin (TBD)
Own profile	CRUD	CRUD	—
Own transactions	Read	Read	—
Own recipients	CRUD	CRUD	—
Merchant dashboard	—	Read	—
All users	—	—	Admin only
AML reports	—	—	Compliance only

---

4. Data Encryption

4.1 Encryption at Rest

Data	Method	Status
Database (SQLite)	OS-level (filesystem)	MVP — migrate to PostgreSQL Phase 2
Passwords	bcrypt, cost factor 12	Implemented (`bcryptjs ^3.0.3`)
Session tokens	SHA-256 hash stored (not plaintext)	Implemented
JWT secret	`JWT_SECRET` env var (fatal if missing in prod)	Implemented
Fødselsnummer (national ID)	AES-256-GCM application layer + HSM key	Planned Phase 2
Card data	Only `last_four` + `token_ref` stored (PAN/CVV never stored)	Implemented (fix C1)
Bank account numbers	Only last 4 digits in API responses	Implemented

Note: Field-level encryption for PII (fødselsnummer) requires HSM-backed key management (planned Phase 2 with AWS KMS).

4.2 Encryption in Transit

Connection	Protocol	Status
User → Drop API	HTTPS / TLS 1.3	Production requirement

Connection	Protocol	Status
Drop → BankID	HTTPS / TLS 1.3	Phase 2
Drop → Sumsub	HTTPS / TLS 1.3	Phase 2
Drop → Neonomics (PSD2)	HTTPS / TLS 1.3	Phase 2
Drop → Swan	HTTPS / TLS 1.3	Phase 2
Internal (service-to-service)	mTLS	Phase 3 (when microservices introduced)

4.3 Cookie Security Configuration

Source: `src/drop-app/src/lib/auth.ts:48-54`

Property	Value	Purpose
`httpOnly`	`true`	Prevents JavaScript access (XSS mitigation)
`secure`	`true` (production)	HTTPS-only transport
`sameSite`	`"strict"`	CSRF prevention
`maxAge`	86400 (24h)	Session lifetime
`path`	`"/"`	Full site scope

---

5. Network Security

5.1 Current MVP Network Architecture

```
Internet → Next.js App (port 3000) → SQLite DB (local file)
```

Phase 2 target:

```
Internet
  → Cloudflare (DDoS + WAF)
  → AWS Load Balancer (TLS termination)
  → Private Subnet: Next.js App (ECS/Fargate)
  → Private Data Subnet: PostgreSQL (RDS)
  → External APIs: BankID, Sumsub, Neonomics, Swan (all HTTPS)
```

5.2 Security Groups (Phase 2 planned)

Source	Destination	Port	Action
Internet	Load Balancer	443	ALLOW
Internet	Any	80	REDIRECT → 443
Load Balancer	App servers	3000	ALLOW
App servers	PostgreSQL	5432	ALLOW
App servers	External APIs	443	ALLOW (allowlist)
Any	Data Subnet	Any	DENY (default)

5.3 Rate Limiting

Source: `src/drop-app/src/lib/middleware.ts:6-31`

Endpoint Type	Limit	Window	Implementation
Auth routes (login, register)	10 req	60 seconds	SQLite-backed (persistent across restarts)
Transaction routes (remittance, qr-payment)	10 req	60 seconds	SQLite-backed
Rate routes (`/api/rates`)	120 req	60 seconds	SQLite-backed

Rate limit table: `rate_limits` — per-IP tracking via `X-Forwarded-For` header.

Known gap: X-Forwarded-For can be spoofed. Fix requires trusted proxy validation (planned Phase 2 with load balancer).

---

6. API Security

6.1 Input Validation

Source: `src/drop-app/src/lib/middleware/validation.ts:149-203`

All API inputs validated at controller level:

- `sanitizeText()` — strips HTML tags, control characters, enforces max length
- `validateName()` — rejects XSS payloads, script tags
- `validateEmail()` — RFC format validation
- `validatePhone()` — international format

- `validateAmount()` — positive, finite, max 2 decimal places
- `validateIBAN()` — format + checksum
- `validatePIN()` — exactly 4 digits
- `validateCurrency()` — whitelist: EUR, USD, GBP, BAM, CHF, PLN, NOK, RSD, TRY, PKR
- `validateLanguage()` — whitelist: nb, en, bs, sq

Amount limits:

Endpoint	Min	Max
Remittance	100 NOK	50,000 NOK
QR Payment	1 NOK	100,000 NOK

6.2 SQL Injection Prevention

All 24 API endpoints use **parameterized queries exclusively** (`?` placeholders). No string concatenation in SQL.

Source: `src/drop-app/src/app/api/` — all route handlers; verified in security audit 2026-02-12.

6.3 CSRF Protection

Origin header validation on all authenticated requests:

- Validates `Origin` against: `NEXT_PUBLIC_APP_URL`, `http://localhost:3000`, `http://localhost:3001`
- Combined with `sameSite: "strict"` cookies for defense-in-depth

Source: `src/drop-app/src/lib/middleware.ts:44-55`

6.4 Content Security Policy

Source: `src/drop-app/next.config.ts:6-46`

```
Content-Security-Policy:
  default-src 'self';
  script-src 'self' 'unsafe-inline' 'unsafe-eval';
  style-src 'self' 'unsafe-inline';
  img-src 'self' data: https:;
  font-src 'self';
  frame-ancestors 'none';
```

Known limitation (Medium severity): `unsafe-inline` and `unsafe-eval` required for Next.js dev mode. Production should use nonce-based CSP. Planned Phase 2.

---

7. OWASP Top 10 Mitigation Matrix

OWASP Risk	Mitigation	Implementation	Status
A01: Broken Access Control	RBAC + `AND user_id = ?` on all queries	All 24 API endpoints	Implemented
A02: Cryptographic Failures	bcrypt cost 12, JWT HS256, HTTPS TLS 1.3, httpOnly cookies	`auth.ts`, `utils-server.ts`, `next.config.ts`	Implemented
A03: Injection	Parameterized queries exclusively (no string SQL concat)	All API routes	Implemented
A04: Insecure Design	Pass-through model (no fund custody), feature flags for cards	Architecture, `feature-flags.ts`	Implemented
A05: Security Misconfiguration	Security headers, demo credentials gated (`NODE_ENV !== 'production'`)	`next.config.ts`, `db.ts`	Implemented
A06: Vulnerable Components	All deps recent, no known CVEs (audit 2026-02-12)	`package.json`	Implemented
A07: Auth Failures	bcrypt hashing, session revocation, rate limiting, no SHA-256 legacy	`auth.ts`, `middleware.ts`, `utils-server.ts`	Implemented
A08: Software Integrity	TBD — signed commits planned Phase 3	—	Planned
A09: Logging Failures	Sentry (error tracking) — audit log table planned Phase 3	MVP: Sentry	Partial
A10: SSRF	Pass-through model limits outbound surface; allowlist planned Phase 2	Architecture	Partial

---

8. Security Headers Checklist

Source: `src/drop-app/next.config.ts`

Header	Value	Status

`Strict-Transport-Security`	`max-age=63072000; includeSubDomains; preload`	Implemented (fix M2)
`Content-Security-Policy`	See §6.4	Partial — `unsafe-inline` remaining
`X-Content-Type-Options`	`nosniff`	Implemented
`X-Frame-Options`	`DENY`	Implemented
`Referrer-Policy`	`strict-origin-when-cross-origin`	Implemented
`Permissions-Policy`	`camera=(self), microphone=(), geolocation=(self)`	Implemented
`Cache-Control` (auth responses)	`no-store`	TBD — Phase 2

---

9. Dependency Vulnerability Management

Last dependency review: 2026-02-12 (security audit)

Package	Version	Risk Assessment
`jose`	^6.1.3	Low — Well-maintained JWT library
`bcryptjs`	^3.0.3	Low — Pure JS bcrypt
`better-sqlite3`	^12.6.2	Low — Parameterized queries
`next`	16.1.6	Low — Recent version
`react`	19.2.3	Low — Latest major
`radix-ui`	^1.4.3	Low — UI components only

Remediation SLAs:

Severity	SLA
Critical (CVSS ≥ 9.0)	24 hours
High (CVSS 7.0-8.9)	7 days
Medium (CVSS 4.0-6.9)	30 days
Low (CVSS < 4.0)	90 days

Planned: Dependabot + Snyk integration in CI/CD pipeline (Phase 3).

---

10. Security Logging & Audit Trail

Current (MVP): Sentry for error tracking. No structured audit log table.

Planned (Phase 3): Audit log table + SIEM integration.

Event	Planned Logging	Alert?
Login success	user_id, ip, user_agent, timestamp	No
Login failure	ip, email_hash, attempt_count, timestamp	Yes (> 5 failures)
Session revocation	user_id, timestamp, reason	Yes
Transaction initiated	user_id, amount, currency, corridor, timestamp	No
KYC status change	user_id, old_status, new_status, timestamp	Yes
AML flag triggered	user_id, rule, transaction_id, timestamp	Yes
Password change	user_id, ip, timestamp	Yes

AML transaction monitoring thresholds (from `legal/hvitvaskingsrutiner.md`):

- Single transaction > NOK 50,000 → manual review
- Daily cumulative > NOK 100,000 → manual review
- Monthly cumulative > NOK 500,000 → EDD assessment
- Structuring patterns → automatic flag

---

11. Integration Security

Third-Party Services (Phase 2)

Service	Purpose	Auth Method	Data Shared	DPA
BankID Norge AS	SCA + Identity verification	OIDC	Name, fødselsnummer	Required
Sumsub	KYC/AML document verification	API key + webhook HMAC	ID documents, liveness data	Signed (`legal/dpa-sumsub.md`)
Swan	Banking / payment rails	OAuth 2.0	Transaction data	Signed (`legal/dpa-swan.md`)

Service	Purpose	Auth Method	Data Shared	DPA
Neonomics	PSD2 AISP/PISP (Open Banking)	OAuth 2.0 (PSD2)	Bank account data, payment initiation	Required
Sentry	Error monitoring	DSN	Stack traces, user IDs	Signed (`legal/dpa-sentry.md`)
AWS	Infrastructure	IAM roles + KMS	Infrastructure only	AWS DPA

Approval

Role	Name	Date	Signature
Author	ALAI Security Team	2026-02-23	
CISO / Security Lead	TBD — requires appointment		
DPO	TBD — requires appointment		
CTO	Alem Bašić		

Revision #10
Created 2026-02-23 12:05:09 UTC by John
Updated 2026-07-05 20:02:58 UTC by John