Security Architecture

Security Architecture — High-Level Design

Version: 1.0 **Date:** 2026-02-21 **Author:** Banking Architecture Team **Status:** Approved **Applies to:** Drop — Security Threat Model & Controls

---

1. Overview

Drop is a PSD2-regulated fintech application that processes financial transactions (remittance, QR payments) without holding customer funds. This document defines the security architecture: trust boundaries, threat model (STRIDE), SCA implementation, fraud detection, AML screening, data classification, and encryption strategy.

Security posture summary:

- All authentication via BankID OIDC (SCA by default)
- All payment SCA delegated to ASPSP (user's bank)
- JWT tokens in httpOnly cookies (web) or AsyncStorage (mobile)
- Parameterized SQL queries (no string concatenation)
- Input sanitization on all user-facing endpoints
- Compliance tables for audit, AML, STR, screening, consents, GDPR

---

2. Trust Boundaries

```
graph TB
    subgraph Internet["Internet (Untrusted)"]
        Browser["Web Browser"]
        Mobile["Mobile App (Expo)"]
        Attacker["Potential Attacker"]
    end
```

```
    subgraph CDN["CDN / Edge (Cloudflare)"]
        WAF["WAF + DDoS Protection"]
        TLS["TLS Termination"]
    end

    subgraph AppTier["Application Tier (AWS App Runner)"]
        subgraph NextJS["Next.js BFF"]
            WebRoutes["Web API Routes<br/>/api/auth/*, /api/transactions/*"]
            Middleware["Auth Middleware<br/>Rate Limiter<br/>CSRF Validator<br/>Input
Sanitizer"]
        end
        subgraph Hono["Hono API"]
            MobileRoutes["Mobile API Routes<br/>/v1/auth/*, /v1/transactions/*"]
            HonoMiddleware["Auth Middleware<br/>Rate Limiter"]
        end
    end

    subgraph DataTier["Data Tier (Private Subnet)"]
        SQLite["SQLite / PostgreSQL<br/>19 tables (12 core + 7 compliance)"]
    end

    subgraph ExternalServices["External Services (Trusted Partners)"]
        BankID["BankID OIDC<br/>(auth.bankid.no)"]
        ASPSP["ASPSPs<br/>(DNB, SpareBank 1, Nordea)"]
        FX["FX Rate Provider"]
        KYC["KYC Provider<br/>(Sumsub - future)"]
    end

    Browser -->|"HTTPS<br/>TB1: Internet→Edge"| WAF
    Mobile -->|"HTTPS<br/>TB1: Internet→Edge"| WAF
    Attacker -.->|"Blocked by WAF"| WAF
    WAF -->|"TB2: Edge→App"| Middleware
    WAF -->|"TB2: Edge→App"| HonoMiddleware
    Middleware --> WebRoutes
    HonoMiddleware --> MobileRoutes
    WebRoutes -->|"TB3: App→Data"| SQLite
    MobileRoutes -->|"TB3: App→Data"| SQLite
    WebRoutes -->|"TB4: App→External<br/>mTLS"| BankID
```

```
    WebRoutes -->|"TB4: App→External<br/>eIDAS cert"| ASPSP
    MobileRoutes -->|"TB4: App→External"| BankID

    style Internet fill:#ff6b6b,stroke:#333,color:#fff
    style CDN fill:#ffd93d,stroke:#333
    style AppTier fill:#6bcb77,stroke:#333
    style DataTier fill:#4d96ff,stroke:#333,color:#fff
    style ExternalServices fill:#845ec2,stroke:#333,color:#fff
```

Trust Boundary Definitions

Boundary	From	To	Protection
TB1: Internet to Edge	Browser/Mobile	Cloudflare	TLS 1.3, WAF rules, DDoS mitigation
TB2: Edge to Application	Cloudflare	Next.js/Hono	HTTPS, auth middleware, rate limiting
TB3: Application to Data	API layer	SQLite/PostgreSQL	Parameterized queries, file permissions
TB4: Application to External	API layer	BankID/ASPSP	mTLS (eIDAS QWAC), JWKS verification

---

3. STRIDE Threat Model

3.1 Threat Matrix

Component	Spoofing	Tampering	Repudiation	Info Disclosure	DoS	Elevation
BankID Auth	L: BankID handles identity	L: JWKS signature verification	L: Audit log + session tracking	M: pid hash exposure risk	M: Rate limit 10/min	L: Role check on every request
JWT Tokens	M: Token theft via XSS	L: HS256 signature	L: Session table tracks all JWTs	M: Payload contains userId	L: 7d expiry	M: Role claim in JWT
PISP Payments	L: SCA required per payment	M: Amount/payee tampering	L: Audit log + idempotency_key	L: Disclosure before payment	M: Rate limit 10/min	L: KYC check before remittance
AISP Balance	L: Consent required	L: Read-only from ASPSP	L: balance_synced_at tracking	M: Cached balance visible	L: Max 4 reads/day	N/A

Component	Spoofing	Tampering	Repudiation	Info Disclosure	DoS	Elevation
Database	L: No direct access	M: SQL injection risk	L: audit_log table	H: PII in users table	L: Rate limiting	L: User-scoped queries
API Endpoints	M: CSRF on web	M: Input manipulation	L: Audit logging	M: Error message leakage	H: Unthrottled endpoints	M: IDOR if user_id not checked

Risk levels: L = Low (mitigated), M = Medium (partial mitigation), H = High (needs attention), N/A = Not applicable

3.2 Detailed Threat Analysis

S — Spoofing

Threat	Attack Vector	Mitigation	Status
Identity spoofing	Stolen credentials	BankID OIDC (SCA: possession + knowledge)	Implemented
Session hijacking	Token theft	httpOnly + secure + sameSite=Lax cookies	Implemented
CSRF	Forged cross-origin request	State parameter (OIDC), Origin header validation	Implemented
Replay attack	Reuse old auth code	Nonce in OIDC flow, one-time code exchange	Implemented

T — Tampering

Threat	Attack Vector	Mitigation	Status
SQL injection	Malicious input in queries	Parameterized queries (all 24 endpoints)	Implemented
XSS	Script injection in fields	React auto-escaping, CSP headers, sanitizeText()	Implemented
Payment amount tampering	Modified request body	Server-side validation, SCA dynamic linking	Implemented
JWT modification	Altered token claims	HS256 signature verification	Implemented

R — Repudiation

Threat	Attack Vector	Mitigation	Status
Deny transaction	User claims they didn't authorize	BankID SCA log + audit_log table	Partial (audit_log exists, SCA tracking needed)

Threat	Attack Vector	Mitigation	Status
Deny consent	User claims no consent given	consents table with IP address + timestamp	Implemented
Admin action denial	Unauthorized changes	audit_log with user_agent and ip_address	Implemented

I — Information Disclosure

Threat	Attack Vector	Mitigation	Status
PII exposure	Database breach	Encryption at rest (planned), PID hashed with SHA-256	Partial
Card data exposure	API response leakage	Masked to last 4 digits, CVV hidden	Implemented
Bank account exposure	API response leakage	Masked to last 4 digits in recipient list	Implemented
Error message leakage	Verbose error responses	Centralized error handler, generic messages	Implemented

D — Denial of Service

Threat	Attack Vector	Mitigation	Status
API flooding	High request volume	Rate limiting (10-120/min per endpoint)	Implemented
Auth brute force	Repeated login attempts	BankID handles (locks after failures)	Implemented
Database exhaustion	Large data queries	Pagination (max 50/page), query limits	Implemented
Resource exhaustion	Large payloads	Input length limits (sanitizeText)	Implemented

E — Elevation of Privilege

Threat	Attack Vector	Mitigation	Status
IDOR	Access other user's data	`AND user_id = ?` on all queries	Implemented
Role escalation	Modify role claim	Server-side role check, role in DB not just JWT	Implemented

Threat	Attack Vector	Mitigation	Status
Merchant impersonation	Access merchant dashboard	`role = 'merchant'` check on merchant routes. **Note:** merchant role currently grants admin access (audit, screening, STR) via `isAdmin(role) === role === 'merchant'` in `admin.ts`	Implemented
KYC bypass	Skip verification	`kyc_status = 'approved'` check before remittance	Implemented

---

4. SCA Implementation

4.1 Two-Level SCA

Drop implements SCA at two levels:

Level	Purpose	Provider	Method
App Authentication	Login to Drop	BankID OIDC	BankID app (possession) + code/biometrics (knowledge/inherence)
Payment Authorization	Approve PISP payment	ASPSP via BankID	BankID at bank (dynamic linking: amount + payee)

4.2 SCA Factors

Factor Type	BankID Implementation
Knowledge	Personal code / PIN
Possession	Mobile device with BankID app / code generator
Inherence	Biometrics (fingerprint/face on mobile BankID)

PSD2 RTS Art. 4: At least 2 of 3 factors required. BankID provides 2 by default (possession + knowledge or inherence).

4.3 Dynamic Linking (PISP)

For every PISP payment, PSD2 RTS Art. 97(2) requires:

1. User sees **exact amount** and **payee name** during SCA

2. Authentication code is **cryptographically bound** to amount + payee
3. Any change to amount or payee **invalidates** the authentication

This is handled by the ASPSP's BankID integration — Drop passes `instructedAmount` and `creditorName` in the PISP API call, and the bank displays these during BankID authentication.

---

5. Fraud Detection Pipeline

```
flowchart TD
    A[Transaction Request] --> B[Pre-Transaction Checks]

    B --> C{User KYC Status}
    C -->|pending/rejected| D[REJECT: kyc_required]
    C -->|approved| E[Amount Validation]

    E --> F{Amount in range?}
    F -->|No| G[REJECT: validation_error]
    F -->|Yes| H[Velocity Check]

    H --> I{Exceeds daily/weekly limit?}
    I -->|Yes| J[FLAG: velocity_alert<br/>Insert into aml_alerts<br/>severity: medium]
    I -->|No| K[Pattern Analysis]

    K --> L{Structuring detected?<br/>Multiple txns just below threshold}
    L -->|Yes| M[FLAG: structuring_alert<br/>Insert into aml_alerts<br/>severity: high]
    L -->|No| N[Corridor Risk Check]

    N --> O{High-risk corridor?}
    O -->|Yes| P[Enhanced due diligence<br/>FLAG if first-time corridor]
    O -->|No| Q[Recipient Screening]

    Q --> R{Recipient on sanctions list?}
    R -->|Yes| S[BLOCK: sanctions_match<br/>Insert into screening_results<br/>result: match]
    R -->|No| T[APPROVE: Proceed to PISP]

    J --> T
    M --> U[Escalate to compliance officer<br/>Insert into str_reports<br/>status: draft]
    P --> T
```

```
    style D fill:#ff6b6b,color:#fff
    style G fill:#ff6b6b,color:#fff
    style S fill:#ff6b6b,color:#fff
    style T fill:#6bcb77,color:#fff
    style J fill:#ffd93d
    style M fill:#ffd93d
    style U fill:#ff9f43
```

5.1 Detection Rules

Rule	Trigger	Severity	Action
Velocity limit (`checkVelocity`)	> 5 transactions in 1 hour	Medium	`aml_alerts` record, continue with flag
Structuring detection (`checkStructuring`)	3+ transactions in 24h totaling > 50,000 NOK	High	`aml_alerts` + `str_reports` draft
High-value single (`checkHighAmount`)	Single transaction > 100,000 NOK	High	Enhanced monitoring, `aml_alerts` record
High-risk corridor (`checkHighRiskCorridor`)	Country on FATF grey/black list	High	Enhanced due diligence required
Unusual pattern (`checkUnusualPattern`)	Transaction amount > 5x user's average	Medium	`aml_alerts` record
Sanctions match	Recipient matches sanctions list	Critical	Block transaction, escalate
PEP match	User matches PEP database	High	Enhanced due diligence

These rules are implemented in `transaction-monitor.ts` and run on each remittance creation.

5.2 AML Screening Tables

Table	Purpose	Key Columns
`aml_alerts`	Transaction monitoring flags	`alert_type`, `severity`, `status` (open/investigating/resolved/escalated/filed)
`str_reports`	Suspicious Transaction Reports to authorities	`report_type`, `status` (draft/submitted/acknowledged), `reference_number`
`screening_results`	PEP/sanctions/adverse media checks	`screening_type`, `result` (clear/match/potential_match/error)

---

6. Data Classification

6.1 Classification Levels

Level	Description	Examples	Storage	Access
CRITICAL	Financial credentials, encryption keys	JWT_SECRET, BANKID_CLIENT_SECRET, eIDAS private keys	Vaultwarden only	Application runtime only
RESTRICTED	PII subject to GDPR	name, email, phone, date_of_birth, national_id_hash	Encrypted at rest (planned), DB access layer	Authenticated user (own data only)
CONFIDENTIAL	Financial data	transactions, bank balances, exchange rates, fees	DB with user-scoped access	Authenticated user (own data only)
INTERNAL	Operational data	audit_log, rate_limits, sessions	DB	System processes, compliance officers
PUBLIC	Non-sensitive	exchange rates (GET /api/rates), health check	DB / API	Unauthenticated

6.2 Data Classification by Table

Table	Classification	PII Fields	Encryption at Rest	Retention
`users`	RESTRICTED	email, first_name, last_name, phone, date_of_birth, national_id_hash	Planned	5 years post-deletion (AML)
`bank_accounts`	RESTRICTED	account_number, iban	Planned	Active + 5 years
`transactions`	CONFIDENTIAL	amount, recipient details	Planned	5 years (AML/tax)
`recipients`	RESTRICTED	name, bank_account	Planned	Active + 5 years
`sessions`	INTERNAL	token_hash	N/A (hash only)	30 days
`audit_log`	INTERNAL	ip_address, user_agent	Planned	5 years
`aml_alerts`	CONFIDENTIAL	details	Planned	5 years
`str_reports`	CONFIDENTIAL	details, reference_number	Planned	10 years
`screening_results`	CONFIDENTIAL	match_details	Planned	5 years

Table	Classification	PII Fields	Encryption at Rest	Retention
`consents`	RESTRICTED	ip_address	Planned	Until withdrawn + 5 years
`merchants`	CONFIDENTIAL	None (business data)	Planned	Active + 5 years
`cards`	RESTRICTED	last_four, token_ref	Planned	Active + 5 years
`data_access_requests`	INTERNAL	None (metadata only)	N/A	5 years
`complaints`	INTERNAL	None (user text)	Planned	5 years
`notifications`	INTERNAL	None	N/A	90 days
`settings`	INTERNAL	None (preferences)	N/A	Active
`spending_limits`	INTERNAL	None	N/A	Active
`exchange_rates`	PUBLIC	None	N/A	Indefinite
`rate_limits`	INTERNAL	None	N/A	Transient

---

7. Encryption

7.1 Encryption in Transit

Connection	Protocol	Certificate
Browser to Drop	TLS 1.3 (Cloudflare)	Cloudflare managed
Mobile to Drop	TLS 1.3	Cloudflare managed
Drop to BankID	TLS 1.2+	BankID server cert
Drop to ASPSP	mTLS (eIDAS QWAC)	Qualified Website Authentication Certificate
Drop to Database	N/A (SQLite local) / TLS (PostgreSQL)	PostgreSQL server cert

7.2 Encryption at Rest

Data	Current	Target
PostgreSQL 16 (all environments)	AWS RDS encryption (AES-256, TLS 1.3)	Active
Secrets (JWT_SECRET, etc.)	Vaultwarden	Vaultwarden + AWS Secrets Manager
Backups	Not encrypted	AES-256 encrypted backups
Logs	Plain text	Encrypted log storage

7.3 Key Management

Key	Purpose	Storage	Rotation
`JWT_SECRET`	Sign Drop JWTs	Vaultwarden / env var	Every 90 days
`BANKID_CLIENT_SECRET`	BankID OIDC client auth	Vaultwarden / env var	Per BankID policy
eIDAS QWAC private key	mTLS to ASPSPs	HSM (planned)	Per certificate lifecycle
eIDAS QSeal private key	Sign API requests	HSM (planned)	Per certificate lifecycle
`qr_hmac_key` (merchants)	HMAC for QR code verification	DB (`merchants` table)	Per merchant, on creation

7.4 Hashing

Data	Algorithm	Purpose	Source
Passwords	bcrypt (cost 12)	Password verification	`utils-server.ts:8-16`
National ID (pid)	SHA-256	User deduplication	`bankid.ts:211`
JWT tokens	SHA-256	Session lookup	`auth.ts:59`
PIN codes	bcrypt	Card PIN verification	`cards/[id]/pin/route.ts`

---

8. Security Controls Summary

8.1 Application Security

Control	Implementation	Source
Authentication	BankID OIDC (SCA)	`bankid.ts`, `auth.ts`
Authorization	JWT + role check + user_id scoping	`middleware/auth.ts`
Input validation	sanitizeText, validateName, validateAmount, etc.	`middleware/validation.ts`
SQL injection prevention	Parameterized queries (all endpoints)	`db.ts`
XSS prevention	React auto-escaping + CSP + sanitization	`next.config.ts`, `validation.ts`
CSRF prevention	Origin validation + sameSite=Lax cookies	`app.ts:23-30` (CORS)
Rate limiting	Per-IP, persistent (SQLite-backed)	`middleware/rate-limit.ts`

Control	Implementation	Source
Session management	Server-side tracking with revocation	`sessions` table, `auth.ts`

8.2 Infrastructure Security

Control	Implementation	Status
TLS 1.3	Cloudflare edge	Active (landing page)
WAF	Cloudflare WAF rules	Active (landing page)
DDoS protection	Cloudflare automatic	Active
HSTS	`max-age=63072000; includeSubDomains; preload`	Configured (`next.config.ts`)
X-Frame-Options	`DENY`	Configured
X-Content-Type-Options	`nosniff`	Configured
Referrer-Policy	`strict-origin-when-cross-origin`	Configured
Permissions-Policy	Camera (self), microphone (none), geolocation (self)	Configured

8.3 Compliance Controls

Control	Implementation	Table
Audit trail	All significant actions logged	`audit_log`
AML monitoring	Transaction pattern detection	`aml_alerts`
STR filing	Suspicious transaction reports	`str_reports`
PEP/sanctions screening	Automated list checking	`screening_results`
GDPR consent tracking	Consent grant/withdraw with IP	`consents`
Data access requests	GDPR Art. 15-17	`data_access_requests`
Complaint handling	Finansavtaleloven compliance	`complaints`

---

9. Security Audit Results

9.1 Pre-Hardening (2026-02-12)

Severity	Count

CRITICAL	4
HIGH	5
MEDIUM	6
LOW	4

9.2 Post-Hardening (2026-02-13)

Severity	Count	Details
CRITICAL	0	All resolved
HIGH	0	All resolved
MEDIUM	2	CSP tightening (nonce-based), proxy config
LOW	4	Acknowledged, out of scope for MVP

9.3 Key Remediations

Finding	Fix	Source
C1: Card data stored in plain	Now stores only `last_four` + `token_ref`	Schema change
C2: Demo credentials in production	Gated behind `NODE_ENV !== 'production'` (note: `SEED_DEMO=true` can override this check)	`db.ts:241`
C4: SHA-256 password hashes	Removed entirely, bcrypt only	`utils-server.ts`
C6/H1: No session revocation	Implemented in `sessions` table	`auth.ts:56-65`
H4: No input sanitization	sanitizeText() on all text fields	`validation.ts`
M5: Notification ID injection	Validated format + max 100 per request	`notifications/route.ts`
M6: Settings value injection	Currency/language whitelists	`settings/route.ts`

---

10. Cross-References

- **Existing Security Docs:** ../../security/SECURITY-ARCHITECTURE.md — Detailed implementation-level security
- **Compliance Status:** ../../security/COMPLIANCE.md — Regulatory readiness assessment
- **BankID OIDC:** ../integration/bankid-oidc-integration.md — Authentication flow details
- **Open Banking:** ../integration/open-banking-aisp-pisp.md — ASPSP SCA, consent security

- **Payment Processing:** ../integration/payment-processing.md — Transaction integrity, idempotency
- **Database Schema:** ../../backend/DATABASE-SCHEMA.md — All 19 tables including compliance tables
- **API Reference:** ../../backend/API-REFERENCE.md — Endpoint security requirements
- **Authentication:** ../../backend/AUTHENTICATION.md — JWT, session, rate limiting details

---

Revision #10
Created 2026-02-21 05:58:48 UTC by John
Updated 2026-05-23 10:51:40 UTC by John