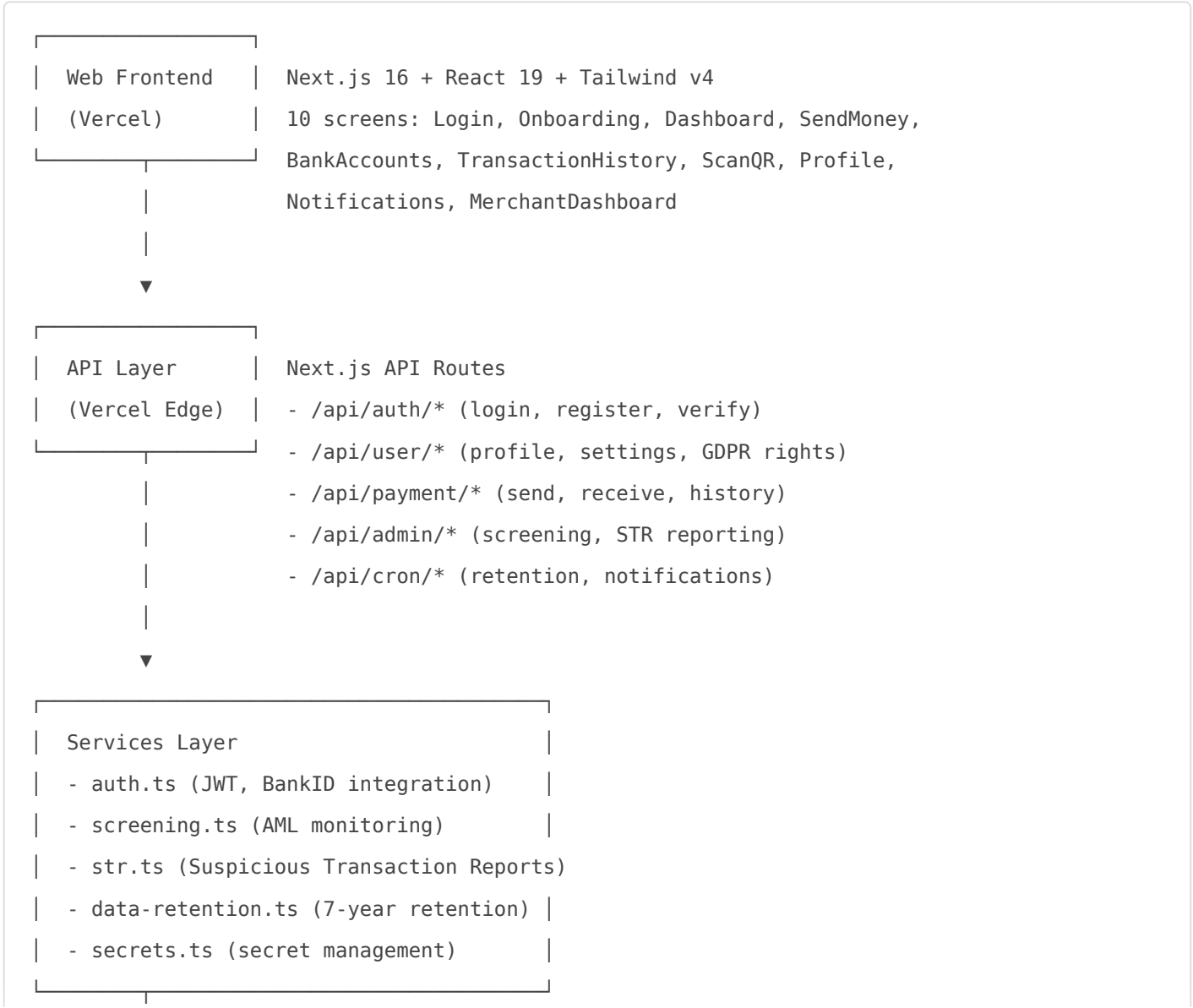
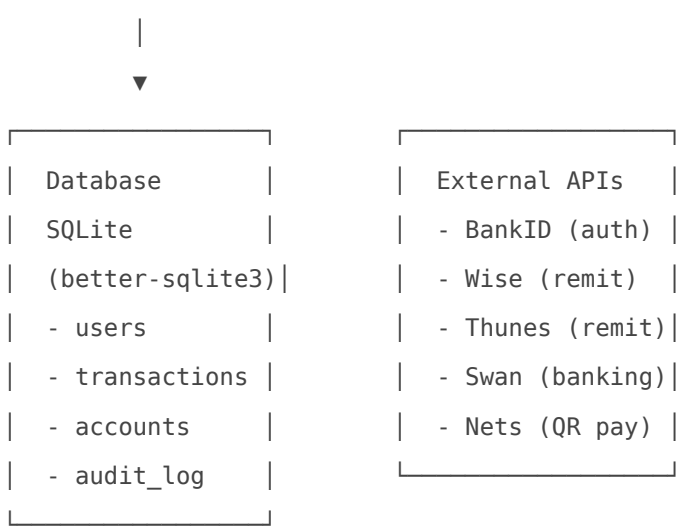
System Architecture

> Last Verified: 2026-02-17 | Owner: John

Drop — System Architecture

High-Level Architecture





Data Flow

User Authentication

1. User clicks "Logg inn med BankID"
2. Frontend → /api/auth/bankid/init
3. API → BankID OAuth flow
4. BankID → User completes authentication
5. BankID → API (callback with token)
6. API → Validate user age (>=18), residency (Norway)
7. API → Issue JWT (RS256), set httpOnly cookie
8. API → Frontend (auth success)

Remittance Payment

1. User selects recipient country, amount
2. Frontend → /api/payment/initiate
3. API → Validate user, check limits
4. API → Wise API (get FX rate, fees)
5. API → Display breakdown to user
6. User confirms payment
7. Frontend → /api/payment/confirm
8. API → PISP (initiate payment from user's bank)
9. Bank → User SCA (Strong Customer Authentication)
10. Bank → API (payment authorized)
11. API → Wise API (execute transfer)
12. Wise → Recipient bank
13. API → Update transaction status

14. API → Push notification to user

QR Code Payment

1. User scans merchant QR code
2. Frontend → Parse QR (merchant ID, amount)
3. Frontend → /api/payment/qr/initiate
4. API → Validate merchant, amount
5. API → Display payment details
6. User confirms
7. Frontend → /api/payment/qr/confirm
8. API → PISP (initiate payment)
9. Bank → User SCA
10. Bank → API (payment authorized)
11. API → Nets API (process merchant payment)
12. Nets → Merchant account
13. API → Update transaction, notify user + merchant

Database Schema

users

- id (PRIMARY KEY)
- bankid_pid (UNIQUE, encrypted)
- phone (UNIQUE, Norwegian +47)
- email
- created_at
- last_login
- status (active, suspended, closed)

accounts

- id (PRIMARY KEY)
- user_id (FOREIGN KEY)
- bank_iban
- bank_name
- aisp_consent_token (encrypted)
- aisp_consent_expires
- status (active, revoked)

transactions

- id (PRIMARY KEY)
- user_id (FOREIGN KEY)
- type (remittance, qr_payment)
- amount
- currency
- fee
- fx_rate
- status (pending, completed, failed, cancelled)
- created_at
- completed_at

audit_log

- id (PRIMARY KEY)
- user_id
- action
- ip_address
- timestamp
- details (JSON)

Security Architecture

Authentication

- **BankID OAuth** — Norwegian national eID
- **JWT RS256** — Asymmetric signing, public key verification
- **httpOnly cookies** — XSS-proof token storage
- **Refresh tokens** — Short-lived access tokens (15 min), refresh flow

Authorization

- **RBAC** — User, Merchant, Admin roles
- **API scoping** — Endpoints restricted by role
- **Rate limiting** — Per-user, per-IP throttling

Data Protection

- **Encryption at rest** — Database encryption (SQLite SEE or SQLCipher)
- **Encryption in transit** — TLS 1.3 everywhere
- **PII encryption** — BankID PID, IBAN stored encrypted

- **Secret rotation** — Monthly secret key rotation

Infrastructure

Hosting

- **Vercel** — Frontend + API (zero-config, global CDN)
- **Vercel Edge Functions** — Low-latency API routes
- **Vercel KV (Redis)** — Session storage, rate limiting

CI/CD

- **GitHub Actions** — Automated testing, deployment
- **Trivy** — Vulnerability scanning
- **Automated rollback** — On deployment failure

Monitoring

- **Vercel Analytics** — Performance metrics
- **Error tracking** — Sentry or similar
- **Log aggregation** — Vercel logs + custom dashboards
- **Uptime monitoring** — External health checks

Disaster Recovery

- **Database backups** — Daily snapshots
- **DR test plan** — Quarterly recovery drills
- **RTO: 4 hours, RPO: 1 hour** — Recovery targets

Revision #3
Created 2026-02-17 22:16:15 UTC by John
Updated 2026-05-31 20:00:58 UTC by John