Architecture Overview

Architecture Overview

System Design

Drop Srbija is a fintech payment app for the Serbian market, enabling phone-based remittance and instant transfers via NBS IPS (Narodna Banka Srbije instant payment system).

Core principle: Money never touches Drop — it stays in user's bank account. Drop facilitates transfers only.

Technology Stack

Component	Choice	Rationale
Language (Backend)	Kotlin	ALAI standard 2026-03-17; full type safety + Ktor native support
Framework	Ktor 3.1.2	Lightweight, async-first, Production-ready with netty transport
Database	PostgreSQL 16	ALAI mandate; full ACID, timezone support, JSON operators
ORM	Exposed (Kotlin DSL)	Kotlin-native, composable, maps to SQL closely
Migrations	Flyway	Version control for schema; integration with Ktor startup
Connection Pool	HikariCP	Battle-tested, fast, connection reuse
Auth	JWT (HS256)	Stateless, easy to validate on backend + frontend
Language (Frontend)	TypeScript + Next.js 15	React 19, server components, app router
Styling	Tailwind CSS 4 + shadcn/ui	ALAI standard; component library prebuilt
Icons	Lucide React	ALAI standard across all products

Component	Choice	Rationale
Testing (Backend)	Kotest + Testcontainers	Kotlin-native, containerized DB for integration tests
Testing (Frontend)	Vitest + Playwright	Unit + E2E, headless browser automation

Module Structure

```
DropSrbija/
├── backend/                    # Kotlin + Ktor 3.1.2
│   ├── src/main/kotlin/no/alai/dropsrbija/
│   │   ├── Application.kt      # Main entry point
│   │   ├── plugins/            # Ktor configuration
│   │   │   ├── Database.kt
│   │   │   ├── Routing.kt
│   │   │   ├── Authentication.kt
│   │   │   └── RateLimit.kt
│   │   ├── auth/               # JWT services
│   │   ├── models/             # Database tables (Exposed DSL)
│   │   └── modules/            # Business logic
│   │       ├── auth/           # Phone OTP flow
│   │       ├── user/           # User management
│   │       ├── transactions/   # Transaction handling
│   │       ├── recipients/     # Recipient management
│   │       └── ips/            # NBS IPS integration
│   └── src/main/resources/db/migration/  # Flyway migrations
└── frontend/                   # Next.js 15 + React 19
    └── src/app/
        ├── layout.tsx
        ├── page.tsx            # Landing page (Serbian)
        └── (app)/              # Authenticated app
```

Database Schema (Migrations V1-V9)

V1: Core Tables (Users, PhoneVerifications, Transactions, Recipients, NbsIpsLogs,

Merchants, Settings)

Users Table:

- `id` — UUID (primary key)
- `phone` — +381XXXXXXXXX (unique)
- `first_name`, `last_name` — User identity
- `email` — optional
- `kyc_status` — pending | verified | rejected
- `role` — user | merchant | admin
- `phone_verified` — boolean
- `deleted_at` — soft delete support

PhoneVerifications Table:

- `id` — UUID (primary key)
- `phone` — +381XXXXXXXXX
- `otp` — 6-digit hash (bcrypt)
- `attempts` — failed attempts counter (max 5)
- `expires_at` — 10-minute expiry
- `verified_at` — timestamp of successful verification

Transactions Table:

- `id` — UUID (primary key)
- `user_id` — references users
- `type` — phone_transfer | qr_payment | bank_transfer
- `status` — processing | completed | failed
- `amount` — integer RSD
- `currency` — RSD | EUR | USD
- `recipient_phone` — +381XXXXXXXXX (if phone-based)
- `nbs_ips_id` — NBS transaction ID
- `nbs_ips_status` — PENDING | ACCEPTED | REJECTED | SETTLED

NbsIpsLogs Table:

- `id` — UUID (primary key)
- `transaction_id` — references transactions
- `request_type` — initiate | check_status | cancel
- `request_body` — JSON payload sent to NBS
- `response_body` — JSON response from NBS
- `response_status` — HTTP status code
- `error_message` — any error text

V2: ISO 20022 Message Support

- Added `iso20022_message_id`, `iso20022_end_to_end_id` columns to `nbs_ips_logs`

V3: Linked Bank Accounts

- New table: `linked_accounts` (user_id, bank_id, iban, account_name, verified_at)

V4: Transaction Idempotency

- Added `idempotency_key_hash` (SHA-256 hash, unique constraint) to `transactions`

V5: KYC Sessions

- New table: `kyc_sessions` (user_id, status, jmbg_verified, biometric_data, completed_at)

V6: National ID Support

- Added `national_id_hash`, `national_id_encrypted` to `users` table

V7: AML Flags

- New table: `aml_flags` (user_id, flag_type, reason, severity, created_by, resolved_at)

V8: User Disclosure Acknowledgment

- Added `disclosure_acknowledged_at` to `users` table

V9: Complaints System

- New table: `complaints` (user_id, category, description, status, resolved_at)

Ports & Services

Service	Port (Local)	Description
PostgreSQL	5434	Drop Srbija database (separate from Drop Norway)
Redis	6380	Rate limiting and caching
Backend API	3003	Ktor HTTP server

Service	Port (Local)	Description
Frontend	3000	Next.js dev server

Payment Flow Diagram

```
flowchart LR
    A[User Enters Phone + OTP] -->|POST /auth/verify-otp| B{OTP Valid?}
    B -->|Yes| C[JWT Issued]
    B -->|No| D[Increment Attempts]
    C --> E[User Initiates Payment]
    E -->|POST /v1/ips/initiate| F[Create Transaction Record]
    F --> G[Call NbsIpsService]
    G -->|ISO 20022 Message| H[NBS IPS API]
    H -->|ACCP/RJCT| I[Log Response to NbsIpsLogs]
    I -->|Async| J{Status?}
    J -->|ACCP| K[Update Transaction: completed]
    J -->|RJCT| L[Update Transaction: failed]
    K --> M[Recipient Receives RSD]
    L --> N[Notify User of Failure]
```

Branch Model

Feature branches are created off the previous feature branch, following a linear progression:

```
feat/drop-srbija-models (T1)
    ↓
feat/drop-srbija-otp (T2)
    ↓
feat/drop-srbija-jwt (T3)
    ↓
feat/drop-srbija-ips (T6)
    ↓
...
feat/drop-srbija-disclosure-complaints (T13)
    ↓
feat/drop-srbija-docs (T29 — this documentation)
```

Merge strategy: TBD — will be established when merging to main/production branch.

Key Architectural Decisions

1. **Bank Partner Adapter Pattern (Task 6):** NBS IPS is ISO 20022 mTLS, not REST; single NBS direct integration impossible without bank license. Solution: Partner with licensed Serbian bank that provides IPS gateway access.
2. **Agent Model Year 1 → Own PI License Year 2:** Start as registered agent under Serbian bank (Article 24 of Payment Services Law), then pursue own Payment Institution license once proven.
3. **Idempotency via SHA-256 Hash (Task 4):** Store hash of idempotency key, not the raw key itself, to minimize PII surface area.
4. **Lock Threshold 5 vs Spec 6:** Industry-standard practice is 5 failed OTP attempts before account lock, rather than 6 mentioned in some specs.

Security Considerations

1. **PII Encryption:**
   - National ID stored encrypted (KMS key rotation)
   - Phone numbers indexed as hashes for lookups
   - Passwords NOT stored (auth via OTP only)
2. **OTP Security:**
   - 6-digit, 10-minute expiry
   - Max 5 attempts per verification
   - Rate limit: 3 requests per minute per phone
   - SMS delivery via Twilio (TLS, no logging)
3. **JWT Validation:**
   - HS256 with strong secret (64+ bytes)
   - Issuer: "dropsrbija-api", Audience: "dropsrbija"
   - httpOnly cookies (frontend sets automatic)
   - 24-hour expiry (refresh token TBD)
4. **NBS IPS Integration:**
   - mTLS for NBS API calls
   - Request signing (HMAC-SHA256)
   - Audit log for all NBS transactions
   - Retry logic with exponential backoff
5. **Rate Limiting:**
   - Global: 1000 req/min per IP
   - Auth: 10 OTP requests/hour per phone
   - Transaction: 50 transactions/hour per user
   - NBS API: Respect rate limits from NBS docs

Monitoring & Observability

Logs

- Structured JSON logs (Logback)
- Sensitive fields redacted (phone last 4 digits only)
- Log level: INFO (debug only in dev)

Metrics

- Transaction latency (p50, p95, p99)
- NBS API response time
- OTP verification success rate
- Error rates by endpoint

Alerts

- NBS IPS unavailable (down 5 min → page on-call)
- Database connection pool exhausted
- Unusual transaction volume

Last Updated: 2026-04-16
Status: Architecture scaffold complete — ready for Phase 2 (Frontend Onboarding UI)

Revision #2
Created 2026-04-16 22:35:13 UTC by John
Updated 2026-05-31 20:06:04 UTC by John