# Frontend Architecture Document

# Frontend Architecture Document

> **Project:** Drop — Fintech Payment App **Version:** 0.1.0 **Date:** 2026-02-23
> **Author:** John (AI Director, ALAI) **Status:** In Review **Reviewers:** Alem Bašić (CEO)

## Document History

| Version | Date | Author | Changes |
|---|---|---|---|
| 0.1 | 2026-02-23 | John | Initial draft from source code analysis |

---

## 1. Purpose & Scope

This document defines the frontend architecture for **Drop**, a fintech payment app for all residents of Norway/Scandinavia. It covers the Next.js web application (`src/drop-app/`) and the static marketing site (`landing/`).

**Intended readers:** Frontend engineers, mobile engineers, architects, product managers.

**In scope:**

- Next.js App Router web application (`src/drop-app/`)
- Static HTML marketing landing page (`landing/index.html`)
- Expo React Native mobile app (`src/drop-mobile/`) — details in `mobile-architecture.md`

**Out of scope:** Backend API (Hono v4), infrastructure, CI/CD pipelines.

---

## 2. Framework Choice & Rationale

| Criterion | Weight | Next.js 15 | Nuxt 3 | SvelteKit | Remix |
|---|---|---|---|---|---|
| Team expertise | 30% | 9 | 5 | 4 | 6 |
| Ecosystem maturity | 25% | 9 | 7 | 6 | 7 |
| Performance | 20% | 9 | 8 | 9 | 8 |
| DX / tooling | 15% | 9 | 7 | 7 | 7 |
| License / cost | 10% | 10 | 10 | 10 | 10 |

**Selected Framework:** `Next.js 15 (App Router) + React 19`

**Rationale:** Next.js was selected because the entire team (AI-driven) operates in TypeScript + React. The App Router supports a hybrid rendering model essential for Drop: Server Components for the marketing landing page (fast, SEO-friendly), and Client Components for the authenticated payment flows (real-time, user-specific). Code sharing between the Next.js web app and the Expo mobile app is maximized — shared hooks, types, and validation logic. The Vercel deployment model provides zero-config CDN, edge middleware for auth protection, and ISR for marketing pages without managing infrastructure.

**Node/runtime version:** `Node.js 20 LTS` **Package manager:** `pnpm`

---

# 3. Project Folder Structure

```
src/drop-app/
├── src/
│   ├── app/                        # Next.js App Router (file-based routing)
│   │   ├── page.tsx                # / — Marketing home (Server Component)
│   │   ├── layout.tsx              # Root layout — fonts, PWA, cookie consent
│   │   ├── login/page.tsx          # /login — Client Component
│   │   ├── register/page.tsx       # /register — Multi-step onboarding
│   │   ├── dashboard/page.tsx      # /dashboard — Auth required
│   │   ├── accounts/page.tsx       # /accounts — Bank account linking
│   │   ├── transactions/page.tsx   # /transactions — Transaction history
│   │   ├── scan/page.tsx           # /scan — QR payment
│   │   ├── send/page.tsx           # /send — Remittance flow
│   │   ├── profile/                # /profile and sub-routes
│   │   ├── notifications/page.tsx  # /notifications
│   │   ├── cards/page.tsx          # /cards — Feature-flagged (future)
│   │   ├── fees/page.tsx           # /fees — Public fee disclosure
│   │   ├── privacy/page.tsx        # /privacy — GDPR policy
```

```
│   │   ├── terms/page.tsx          # /terms — Terms of service
│   │   ├── complaints/page.tsx     # /complaints — Finansavtaleloven §3-53
│   │   ├── withdrawal/page.tsx     # /withdrawal — Angrerettloven
│   │   └── api/                    # API routes (BFF layer)
│   │       ├── auth/               # login, logout, me, register
│   │       ├── transactions/       # remittance, qr-payment
│   │       ├── recipients/         # CRUD recipients
│   │       ├── rates/              # Exchange rates
│   │       ├── cards/              # Feature-flagged
│   │       ├── merchants/          # Merchant dashboard
│   │       ├── settings/           # User preferences
│   │       ├── notifications/      # Push notifications
│   │       └── consents/           # Cookie consent
│   ├── components/
│   │   ├── ui/                     # shadcn/ui primitives (Radix-based)
│   │   ├── bottom-nav.tsx          # Fixed bottom navigation (5 tabs)
│   │   ├── drop-logo.tsx           # DropLogo, DropWordmark, DropLogoFull, DropAppIcon
│   │   ├── drop-icons.tsx          # 9 custom domain-specific icons
│   │   ├── cookie-consent.tsx      # GDPR cookie consent banner + modal
│   │   ├── pre-payment-disclosure.tsx  # PSD2 pre-payment disclosure modal
│   │   └── pwa-register.tsx        # Service Worker registration
│   ├── lib/
│   │   ├── use-auth.ts             # useAuth() — auth hook with redirect
│   │   ├── feature-flags.ts        # Feature flag system (env-var based)
│   │   └── middleware/             # auth-middleware, error-handler, validation
│   └── middleware.ts               # Next.js middleware (auth protection)
├── public/
│   ├── sw.js                       # Service worker (PWA)
│   └── manifest.json               # Web app manifest
├── landing/                        # Static marketing site (separate from Next.js)
│   ├── index.html                  # Main landing page (~646 lines, pure HTML/CSS/JS)
│   └── pages/                      # 12 sub-pages (priser, personvern, vilkar, etc.)
├── globals.css                     # CSS custom properties (brand tokens)
├── next.config.ts
└── package.json
```

# 4. Rendering Strategy

| Page Type | Strategy | Rationale |
|---|---|---|
| `/` — Marketing home | Server Component (SSG) | SEO-critical, no auth, static content |
| `/fees`, `/privacy`, `/terms`, `/complaints`, `/withdrawal` | Client Component (CSR) | Public pages, no auth required |
| `/login`, `/register` | Client Component (CSR) | Form state, client-side validation |
| `/dashboard`, `/accounts`, `/transactions` | Client Component (CSR) | Auth-gated, personalized, real-time data |
| `/scan`, `/send` | Client Component (CSR) | Multi-step flows, camera access, form state |
| `/profile`, `/notifications`, `/cards` | Client Component (CSR) | Auth-gated, user-specific |
| `/api/**` | API Routes | BFF layer — cookie-to-bearer proxy |

**Hydration approach:** Full hydration. Server Components used for the marketing page (`page.tsx`). All authenticated app pages are Client Components using `"use client"` directive.

**Note:** The marketing landing page (`landing/index.html`) is a completely separate pure HTML/CSS/JS file served statically. It does not use the Next.js framework.

---

# 5. Routing Architecture

## 5.1 Route Organization

```
/                          → Marketing home (Server Component)
/login                     → Login (Client)
/register                  → Registration / Onboarding (Client, 4-step)
/dashboard                 → Dashboard (Client, auth required)
/accounts                  → Bank accounts via AISP (Client, auth required)
/transactions              → Transaction history (Client, auth required)
/scan                      → QR scanner (Client, auth required)
/send                      → Remittance flow (Client, auth required)
/profile                   → Profile (Client, auth required)
/profile/personal          → Personal info — BankID-verified, read-only
/profile/security          → Security settings — 2FA, active devices
/profile/notifications     → Notification preferences
/profile/language          → Language selection (nb, en, bs, sq)
/notifications             → Notifications center (Client, auth required)
/cards                     → Card management (Client, auth required, feature-flagged)
```

```
/fees                    → Fee disclosure (Public)
/privacy                 → Privacy policy / GDPR (Public)
/terms                   → Terms of service (Public)
/complaints              → Complaint form — Finansavtaleloven §3-53 (Public)
/withdrawal              → Right of withdrawal — Angrerettloven (Public)
/api/**                  → API routes (BFF — cookie auth proxy to Hono backend)
```

# 5.2 Route Guards / Middleware

Auth is enforced at two levels:

**Level 1 — `useAuth()` hook (per-page):**

```
// src/lib/use-auth.ts
// Default: redirectIfUnauthenticated = true
const { user, loading } = useAuth();
// On 401: redirects to /login
```

**Level 2 — Middleware (future):**

```
Middleware execution order:
1. Request logging
2. Authentication check (JWT cookie validation)
3. Redirect unauthenticated → /login
4. Feature flag gate (returns 404 if flag disabled)
```

**Protected routes:** All routes under `/dashboard`, `/accounts`, `/transactions`, `/scan`, `/send`, `/profile/**`, `/notifications`, `/cards`.

**Public routes:** `/`, `/login`, `/register`, `/fees`, `/privacy`, `/terms`, `/complaints`, `/withdrawal`.

---

# 6. Build & Bundle Configuration

## 6.1 Key Config Options

| Option | Value | Reason |
|---|---|---|
| Output | `standalone` | Docker-optimized deployment |
| Image optimization | `next/image` | WebP conversion, lazy loading |

| Option | Value | Reason |
|---|---|---|
| Bundle analyzer | `ANALYZE=true` | On-demand local analysis |
| Source maps | Production: hidden | Upload to Sentry only |
| Compression | gzip + brotli | CDN-level compression |
| PWA | Service Worker at `/sw.js` | Offline capability (basic) |

## 6.2 Code Splitting Strategy

- **Route-level splitting:** Automatic per App Router segment
- **Component-level splitting:** `dynamic()` for heavy components (Dialog, QR scanner UI)
- **Feature flags:** Cards page components lazy-loaded (all flags default to `false`)

---

# 7. Performance Budget

| Metric | Target | Tool |
|---|---|---|
| LCP (Largest Contentful Paint) | < 2.5s | Lighthouse, CrUX |
| INP (Interaction to Next Paint) | < 200ms | CrUX |
| CLS (Cumulative Layout Shift) | < 0.1 | Lighthouse |
| TTFB (Time to First Byte) | < 600ms | WebPageTest |
| Total JS bundle (compressed) | < 200 KB | Bundlesize CI check |
| First page load (mobile 4G) | < 3s | WebPageTest |
| Lighthouse Performance Score | ≥ 90 | Lighthouse CI |

**Note:** The marketing page (`/`) is the primary SEO and performance target. The app pages (dashboard, send) accept slightly higher bundle sizes due to shadcn/ui + Radix dependencies.

---

# 8. Asset Management

## 8.1 Images

- Format: WebP primary, JPEG fallback, SVG for vectors (logos, icons)
- Optimization: `next/image` with defined domains
- Lazy loading: Native `loading="lazy"` via `next/image`

## 8.2 Fonts

- **Fraunces** (variable, 400-900) — loaded via `next/font/google` in `layout.tsx`
- **DM Sans** (variable, 400-700) — loaded via `next/font/google` in `layout.tsx`
- **Geist Mono** — loaded via `next/font/google` in `layout.tsx`
- CSS variables: `--font-fraunces`, `--font-dm-sans`, `--font-geist-mono`
- `font-display: swap` on all faces (handled by `next/font`)

## 8.3 Icons

- **Primary library:** `lucide-react` (inline SVG via React components)
- **Custom icons:** `components/drop-icons.tsx` — 9 domain-specific icons (IconSendMoney, IconQrScan, IconVirtualCard, IconShield, IconFastTransfer, IconCorridors, IconWallet, IconHistory, IconTopUp)
- **Social auth icons:** Inline SVG (BankID green, Vipps orange)
- Delivery: Inline SVG via component (no icon font, no sprite)

---

# 9. Internationalization (i18n) Strategy

| Item | Decision |
|---|---|
| Library | None (Phase 1 — Norwegian only) |
| Default locale | `nb` (Norwegian Bokmål) |
| Supported locales | `nb` (launch), `en`, `bs`, `sq` (Phase 2) |
| Language selection | `/profile/language` — PATCH `/api/settings` with `{ language: string }` |
| Translation format | TBD — requires i18n library selection |
| RTL support | No |

**Phase 1:** All UI text hardcoded in Norwegian Bokmål. Language setting stored in user preferences (`/api/settings`) but not yet applied to UI translation.

**Phase 2:** Introduce `next-intl` or `i18next` with per-locale JSON translation files for `nb`, `en`, `bs`, `sq`.

---

# 10. Error Boundary Strategy

| Level | Scope | Behavior |
|---|---|---|

| | | |
|---|---|---|
| Global `error.tsx` | Full page crash | Show branded error page |
| Auth failure | API 401 | `useAuth()` redirects to `/login` |
| Form submission | Mutation failure | Inline error message (`text-[#EF4444]`) + retry |
| Data fetch | `useEffect` fetch failure | Caught in `.catch()`, silent or empty state |
| Feature flag | Flag disabled | Returns 404 or "Feature not available" message |

**Error reporting:** TBD — Sentry integration planned for production.

---

# 11. Environment Configuration

| Variable | Dev | Staging | Prod | Description |
|---|---|---|---|---|
| `NEXT_PUBLIC_API_URL` | `http://localhost:3000/api` | `https://drop-app-staging.vercel.app/api` | `https://drop-app.vercel.app/api` | Backend API base URL (mobile uses this) |
| `NEXT_PUBLIC_APP_ENV` | `development` | `staging` | `production` | Runtime environment flag |
| `NEXT_PUBLIC_FF_VIRTUAL_CARDS` | `false` | `false` | `false` | Cards feature flag |
| `NEXT_PUBLIC_FF_PHYSICAL_CARDS` | `false` | `false` | `false` | Physical cards flag |
| `NEXT_PUBLIC_FF_NOTIFICATIONS` | `true` | `true` | `true` | Notifications feature flag |
| `NEXT_PUBLIC_FF_MERCHANT_DASHBOARD` | `true` | `true` | `true` | Merchant dashboard flag |
| `SENTRY_DSN` | optional | required | required | Error reporting (server-side) |

**Feature flag convention:** `NEXT_PUBLIC_FF_` + `SCREAMING_SNAKE_CASE` flag name.

**Secrets management:** All non-public secrets in Vaultwarden (`vault.basicconsulting.no`). Never committed.

---

# 12. Dependency Management Policy

| Rule | Detail |
|---|---|

| | |
|---|---|
| Allowed licenses | MIT, Apache-2.0, ISC, BSD-2, BSD-3 |
| Security audit | `pnpm audit` in CI — fail on high/critical |
| Update cadence | Monthly Renovate PRs for minor/patch; major = manual |
| Banned patterns | No `moment.js` (use `date-fns`), no `lodash` (use native) |

**Key dependencies:**

- `next@15`, `react@19` — framework
- `shadcn/ui` + `@radix-ui/*` — component primitives
- `lucide-react` — icon library
- `tailwindcss` — styling
- `class-variance-authority`, `clsx`, `tailwind-merge` — class utilities

---

# 13. Architecture Diagram

```
graph TB
    subgraph "Client Browser / Mobile"
        Browser["User Browser (PWA)"]
        Mobile["Expo Mobile App"]
    end

    subgraph "Frontend — Next.js 15 App Router"
        CDN["Vercel CDN (Static Assets + Edge)"]
        Marketing["Marketing Page (Server Component, SSG)"]
        AppPages["App Pages (Client Components, CSR)"]
        BFF["BFF API Routes (/api/*)\nCookie → Bearer proxy"]
        Middleware["Next.js Middleware\n(Auth guard)"]
        Components["shadcn/ui + Custom Components\nBottomNav, DropLogo, drop-icons"]
        AuthHook["useAuth() Hook\nGET /api/auth/me"]
        FeatureFlags["Feature Flags\nenv-var based"]
    end

    subgraph "Backend — Hono v4"
        HonoAPI["Hono REST API\n/v1/* — Bearer auth"]
        OpenBanking["Open Banking (PSD2)\nAISP + PISP"]
        BankID["BankID OIDC\nAuthentication"]
    end
```

```
    Browser --> CDN
    Browser --> Marketing
    Browser --> Middleware
    Middleware --> AppPages
    AppPages --> Components
    AppPages --> AuthHook
    AppPages --> FeatureFlags
    AuthHook --> BFF
    BFF --> HonoAPI
    HonoAPI --> OpenBanking
    HonoAPI --> BankID
    Mobile --> HonoAPI
```

# Approval

| Role | Name | Date | Signature |
|---|---|---|---|
| Author | John (AI Director) | 2026-02-23 | |
| Tech Lead | | | |
| Architect | | | |
| Engineering Manager | | | |

Revision #5
Created 2026-02-23 12:05:11 UTC by John
Updated 2026-05-31 20:03:12 UTC by John