drop-sprint1-implementation-plan

Plan: Drop Sprint 1 Implementation

> **Date:** 2026-02-26 **Author:** John (AI Director) **MC Task:** #2110+ **Status:** PENDING APPROVAL **Sprint Duration:** 5 weeks (UI prototype with mock integrations)

Research Summary

Existing Codebase (MUCH more than expected)

- **46 API routes** with complete business logic (auth, transactions, GDPR, admin, cards, merchants)
- **20 UI pages** all with real content (zero stubs)
- **936-line db.ts** with dual-driver (SQLite + PostgreSQL) — needs PostgreSQL-only refactor
- **13 database tables** via node-pg-migrate (raw SQL, NOT Drizzle) — need 12 more tables
- **BankID OIDC** fully implemented with PSD2 dynamic linking (298-line callback)
- **12 Figma Make export screens** (Vite+React) as UI source of truth
- **docker-compose.production.yml** already has PostgreSQL 16
- **2 test files** — needs major expansion
- **NO Drizzle ORM** — currently raw SQL via better-sqlite3 + pg
- **NO BullMQ/Redis** — not yet implemented
- **SHA-256 national_id_hash** — must change to AES-256-GCM
- **Email/password + PIN paths** still in code — must remove for BankID-only

Key Insight

This is NOT a greenfield build. It's a **refactor to align with architecture decisions** + **add missing infrastructure** + **extend with new FRs**. The existing code is functional but pre-dates the architecture review.

Objective

Refactor existing Drop codebase to align with all 16 ADRs, add missing database tables and infrastructure (Redis, BullMQ, Drizzle), implement BankID-only auth, and bring UI in line with Figma Make export designs. All external integrations (BankID, ZTL, FX, compliance) remain mocked.

Team Orchestration

Team Members

ID	Name	Role	Agent Type	Model
B1	db-builder	Database: PostgreSQL-only + Drizzle + 12 new tables	builder	sonnet
V1	db-validator	Validate database migration + schema	validator	sonnet
B2	auth-builder	BankID-only auth refactor + AES-256-GCM	builder	sonnet
V2	auth-validator	Validate auth + security	validator	sonnet
B3	infra-builder	Docker, Redis, BullMQ, graceful shutdown	builder	sonnet
V3	infra-validator	Validate infrastructure	validator	sonnet
B4	api-builder	New API routes for FR-073 through FR-077	builder	sonnet
V4	api-validator	Validate API routes	validator	sonnet
B5	ui-builder	Align UI with Figma Make + admin portal	builder	sonnet

ID	Name	Role	Agent Type	Model
V5	ui-validator	Validate UI against Figma Make	validator	sonnet

---

Step-by-Step Tasks

Phase 1: Foundation (Week 1) — Database + Infrastructure

Task 1: PostgreSQL-only db.ts refactor

- Owner: B1
- BlockedBy: none
- Description: Replace 936-line dual-driver `src/lib/db.ts` with PostgreSQL-only via Drizzle ORM. Remove `better-sqlite3` dependency. Remove all SQLite code paths. Install `drizzle-orm` + `drizzle-kit` + `@neondatabase/serverless` (for edge compat) or `postgres` driver. Create Drizzle schema matching existing 13 tables.
- Files: `src/lib/db.ts`, `package.json`, `drizzle.config.ts`
- Acceptance:
  - ☐ `better-sqlite3` removed from package.json
  - ☐ `drizzle-orm` and `drizzle-kit` in dependencies
  - ☐ `src/lib/db.ts` uses Drizzle with `pg` driver only
  - ☐ All existing `getOne()`, `getAll()`, `run()` calls still work (adapter layer or full migration)
  - ☐ `docker-compose.yml` updated with PostgreSQL service for dev
  - ☐ `drop.db` SQLite file removed from project

Task 2: Drizzle schema for all 25 tables

- Owner: B1
- BlockedBy: 1
- Description: Create `src/lib/schema.ts` with Drizzle table definitions for all 25 tables per database-design.md. Remove node-pg-migrate, use Drizzle Kit for migrations. Include the 12 new tables: `aml_alerts`, `str_reports`, `screening_results`, `consents`, `data_access_requests`, `complaints`, `reconciliation_reports`, `reconciliation_discrepancies`, `circuit_breaker_state`, `webhook_events`, `webhook_dlq`, `disputes`.
- Files: `src/lib/schema.ts`, `drizzle.config.ts`, `migrations/` (Drizzle format)
- Acceptance:
  - ☐ 25 Drizzle table definitions matching database-design.md

- [ ] All FKs, indexes, constraints defined
- [ ] `national_id_hash` column renamed to `national_id_encrypted` + `national_id_hmac`
- [ ] `password_hash` column still exists but default='BANKID_ONLY' (backwards compat)
- [ ] `npx drizzle-kit generate` produces valid migration
- [ ] `npx drizzle-kit push` applies to local PostgreSQL without errors

Task 3: Validate database migration

- Owner: V1
- BlockedBy: 2
- Acceptance:
  - [ ] All 25 tables created in PostgreSQL
  - [ ] Schema matches database-design.md exactly
  - [ ] Indexes from indexing-strategy.md present
  - [ ] No SQLite references remain in codebase (`grep -r "better-sqlite3\|sqlite" src/`)
  - [ ] Docker PostgreSQL starts and accepts connections

Task 4: Redis + BullMQ infrastructure

- Owner: B3
- BlockedBy: none (parallel with Task 1)
- Description: Add Redis and BullMQ per ADR-015. Create `src/lib/redis.ts` (connection), `src/lib/queues.ts` (5 queue definitions), `src/lib/workers/` (worker stubs for each queue). Add Redis to `docker-compose.yml`. Add `SIGTERM` handler per ADR-016.
- Files: `src/lib/redis.ts`, `src/lib/queues.ts`, `src/lib/workers/`, `docker-compose.yml`, `src/lib/shutdown-handlers.node.ts`
- Acceptance:
  - [ ] `bullmq` and `ioredis` in package.json
  - [ ] Redis service in docker-compose.yml (port 6379)
  - [ ] 5 queues defined: `payment-critical`, `settlement`, `compliance`, `reporting`, `notifications`
  - [ ] SIGTERM handler with 25s drain, `shutdown_interrupted` state
  - [ ] `docker compose up` starts both PostgreSQL and Redis
  - [ ] Health endpoint includes Redis connectivity check

Task 5: Validate infrastructure

- Owner: V3
- BlockedBy: 4
- Acceptance:
  - [ ] `docker compose up` starts app + PostgreSQL + Redis

- [ ] Health endpoint reports all services healthy
- [ ] BullMQ can enqueue and process a test job
- [ ] SIGTERM handler tested (send SIGTERM, verify graceful shutdown)

Phase 2: Auth + Security (Week 2)

Task 6: BankID-only authentication

- Owner: B2
- BlockedBy: 2 (needs Drizzle schema)
- Description: Remove all email/password and phone/PIN auth paths. Keep BankID OIDC as sole auth. Remove `src/app/api/auth/login/`, `src/app/api/auth/register/`, `src/app/api/auth/verify-otp/`. Update `src/app/api/auth/bankid/callback/route.ts` to use AES-256-GCM for fødselsnummer (not SHA-256). Update session policy: 30min sliding idle, 8h absolute, 5min payment SCA. Remove `bcryptjs` dependency (no passwords).
- Files: `src/app/api/auth/`, `src/lib/auth.ts`, `src/lib/services/auth-provider.ts`, `src/app/login/page.tsx`, `src/app/register/page.tsx`
- Acceptance:
  - [ ] `/api/auth/login` — REMOVED
  - [ ] `/api/auth/register` — REMOVED
  - [ ] `/api/auth/verify-otp` — REMOVED
  - [ ] `/api/auth/bankid` — sole auth entry point
  - [ ] BankID callback stores `national_id_encrypted` (AES-256-GCM) + `national_id_hmac` (HMAC-SHA256)
  - [ ] JWT has 30min expiry, refresh token server-side
  - [ ] Login page shows only "Logg inn med BankID" button
  - [ ] Registration page redirects to BankID flow
  - [ ] `bcryptjs` removed from package.json
  - [ ] Zero references to `password`, `pin`, `email+password` in auth flows

Task 7: Validate auth + security

- Owner: V2
- BlockedBy: 6
- Acceptance:
  - [ ] No email/password/PIN auth paths reachable
  - [ ] BankID mock flow works end-to-end
  - [ ] national_id stored as AES-256-GCM encrypted (verify with DB inspection)
  - [ ] Session expiry at 30min verified

- [ ] `grep -r "SHA-256\|sha256\|national_id_hash" src/` returns zero results in auth code
- [ ] `grep -r "bcrypt\|password_hash" src/` — only legacy column definition, no active auth use

---

Phase 3: API Routes (Week 2-3)

Task 8: Webhook handling API (FR-076)

- Owner: B4
- BlockedBy: 2, 4 (needs Drizzle + BullMQ)
- Description: Create `POST /api/webhooks/banking-partner` per FR-076. HMAC-SHA256 validation, IP whitelist, idempotent processing, state machine (pending→processing→completed/failed), DLQ after 3 attempts.
- Files: `src/app/api/webhooks/banking-partner/route.ts`, `src/lib/services/webhook-processor.ts`
- Acceptance:
  - [ ] HMAC-SHA256 signature validation with 5-min timestamp check
  - [ ] webhook_events table populated on each webhook
  - [ ] Duplicate webhook_id returns existing result (idempotent)
  - [ ] Failed webhooks retry 3 times then move to webhook_dlq
  - [ ] Returns 200 within 5 seconds

Task 9: Reconciliation API (FR-073)

- Owner: B4
- BlockedBy: 2, 4
- Description: Create reconciliation job and API. BullMQ cron job at 06:00 Oslo time. Compares Drop transactions with mock banking partner data. Creates reconciliation_reports and reconciliation_discrepancies records. Admin endpoint `GET /api/admin/reconciliation` to view reports.
- Files: `src/lib/workers/reconciliation-worker.ts`, `src/app/api/admin/reconciliation/route.ts`
- Acceptance:
  - [ ] BullMQ cron job scheduled at 06:00 Europe/Oslo
  - [ ] Reconciliation report created with matched/discrepancy counts
  - [ ] Discrepancies categorized by type
  - [ ] Admin API returns reports with pagination

Task 10: Circuit breaker service (FR-075)

- Owner: B4

- BlockedBy: 2, 4
- Description: Create circuit breaker service per FR-075. Shared state in PostgreSQL `circuit_breaker_state` table. 5 instances (BankID, ZTL, FX, compliance, push). Fallback behaviors per dependency.
- Files: `src/lib/services/circuit-breaker.ts`, `src/lib/workers/circuit-breaker-monitor.ts`
- Acceptance:
  - ☐ 5 circuit breaker instances initialized on startup
  - ☐ State transitions: closed→open after 5 failures, open→half-open after 30s
  - ☐ Each external service call wrapped in circuit breaker
  - ☐ Fallback behavior per dependency (read-only mode, cached rates, etc.)

Task 11: Dispute/refund API (FR-077)

- Owner: B4
- BlockedBy: 2
- Description: Create dispute endpoints per FR-077. `POST /api/transactions/:id/dispute` for user submission. `GET /api/admin/disputes` for admin queue. State machine (submitted→acknowledged→investigating→decided→closed).
- Files: `src/app/api/transactions/[id]/dispute/route.ts`, `src/app/api/admin/disputes/route.ts`, `src/lib/services/dispute.ts`
- Acceptance:
  - ☐ User can submit dispute from transaction detail
  - ☐ Auto-acknowledgement (logged, notification created)
  - ☐ Admin can view/manage dispute queue
  - ☐ State machine enforces valid transitions only
  - ☐ 5-year retention with do_not_delete flag

Task 12: Validate API routes

- Owner: V4
- BlockedBy: 8, 9, 10, 11
- Acceptance:
  - ☐ All new endpoints respond correctly (mock data)
  - ☐ Webhook idempotency verified (send same webhook twice)
  - ☐ Circuit breaker state transitions verified
  - ☐ Dispute lifecycle flows correctly
  - ☐ All endpoints require auth (except webhook which uses HMAC)

Phase 4: UI Alignment (Week 3-4)

Task 13: Align existing screens with Figma Make export

- Owner: B5
- BlockedBy: 6 (needs BankID-only auth)
- Description: Compare each of the 10 existing screens against Figma Make export (`mockups/figma-make-export/src/app/screens/`). Update layouts, colors, typography, spacing to match. Screens: Login, Onboarding, Dashboard, SendMoney, BankAccounts, TransactionHistory, ScanQR, Profile, Notifications, MerchantDashboard. Plus 2 merchant QR screens.
- Files: All page.tsx files in `src/app/`
- Acceptance:
  - ☐ Each screen visually matches Figma Make export
  - ☐ Login page: BankID-only (no email/password form)
  - ☐ Dashboard: balance card, recent transactions, quick actions
  - ☐ SendMoney: multi-step flow matching Make export
  - ☐ Brand colors match Drop brand guidelines (green gradient #0B6E35→#064E25)
  - ☐ Mobile-first responsive design

Task 14: Admin portal UI (EP-09)

- Owner: B5
- BlockedBy: 8, 9, 10, 11 (needs admin APIs)
- Description: Create admin portal pages per EP-09. MFA login (mock), AML alert dashboard, STR filing, KYC review, transaction search, settlement management, report generation, audit log viewer. Route: `/app/admin/`.
- Files: `src/app/admin/` (new directory), 8 page files
- Acceptance:
  - ☐ Admin login page with MFA (mocked)
  - ☐ AML alert queue with filter/sort/action buttons
  - ☐ STR filing form pre-filled from alert data
  - ☐ User KYC search and detail view
  - ☐ Transaction search with filters and CSV export
  - ☐ Settlement dashboard with batch status
  - ☐ Compliance report list with PDF/CSV download stubs
  - ☐ Audit log viewer with search

Task 15: Validate UI

- Owner: V5
- BlockedBy: 13, 14
- Acceptance:

- [ ] All 10 consumer screens match Figma Make export (visual comparison)
- [ ] Admin portal: all 8 pages render without errors
- [ ] No broken routes (404s)
- [ ] Mobile responsive (viewport 375px)
- [ ] Brand colors consistent
- [ ] No email/password UI anywhere

Phase 5: Integration + Testing (Week 4-5)

Task 16: Mock services for all external integrations

- Owner: B4
- BlockedBy: 6, 10 (needs auth + circuit breaker)
- Description: Create/update mock services for: BankID (already exists), ZTL banking partner (PISP/AISP mock), FX rate provider (mock with realistic NOK→RSD/BAM/PLN/EUR/PKR/TRY rates), compliance provider (mock PEP/sanctions screening), push notifications (mock). All mocks should be realistic and toggleable via feature flags.
- Files: `src/lib/services/mock-*.ts` (update existing + create new)
- Acceptance:
  - [ ] BankID mock returns valid OIDC flow
  - [ ] PISP mock returns realistic payment confirmation after delay
  - [ ] AISP mock returns balance data for linked accounts
  - [ ] FX mock returns rates for all 6 corridors
  - [ ] Compliance mock returns PEP clear/match based on test data
  - [ ] All mocks activated via `MOCK_MODE=true` env var

Task 17: End-to-end test suite

- Owner: B4
- BlockedBy: 16
- Description: Expand test suite (currently 2 files). Add Playwright E2E tests for critical user flows: BankID login → dashboard → send money → transaction history. Add API integration tests for webhook, reconciliation, dispute. Add unit tests for circuit breaker, idempotency logic.
- Files: `tests/e2e/`, `tests/integration/`, `tests/unit/`
- Acceptance:
  - [ ] E2E: Login flow → Dashboard renders
  - [ ] E2E: Send money flow (mock) completes
  - [ ] E2E: Admin portal accessible

- [ ] Integration: Webhook processing with HMAC
- [ ] Integration: Reconciliation job produces report
- [ ] Unit: Circuit breaker state transitions
- [ ] Unit: Idempotency key generation deterministic
- [ ] All tests pass in CI (docker-compose up + test)

Task 18: Final validation

- Owner: V1 + V2 + V3 + V4 + V5 (joint)
- BlockedBy: 17
- Description: Full system validation. Start docker-compose, run all tests, verify all acceptance criteria from all tasks, check for security issues.
- Acceptance:
  - [ ] `docker compose up` → app healthy within 30s
  - [ ] All 25 tables present in PostgreSQL
  - [ ] BankID-only auth (no other paths)
  - [ ] All new API routes functional
  - [ ] All UI screens render correctly
  - [ ] Zero SQLite references in codebase
  - [ ] Zero SHA-256 national_id references
  - [ ] QA-19 check passes (≥17/19 for H priority)

Validation Commands

```
# Database
docker compose up -d
npx drizzle-kit push
psql -h localhost -U drop -d drop -c "\dt" | wc -l  # Should show 25 tables

# Auth
grep -r "better-sqlite3\|bcrypt\|password.*login\|pin.*login" src/ # Should be zero
grep -r "national_id_hash" src/ # Should be zero
curl http://localhost:3000/api/auth/login # Should 404

# Infrastructure
docker compose ps # PostgreSQL + Redis + app all healthy
curl http://localhost:3000/api/health # Should include redis: ok, db: ok
```

```
# Tests
npm test
npx playwright test

# QA Gate
node ~/system/tools/qa-19.js check <task-id>
```

Risk Mitigation

Risk	Mitigation
db.ts refactor breaks all 46 API routes	Task 1 creates adapter layer first, then migrates route-by-route
BankID mock breaks existing flow	Keep `BANKID_MOCK=true` env var, test before removing legacy paths
Drizzle migration incompatible with existing data	Fresh PostgreSQL for Sprint 1 (no production data yet)
UI alignment takes longer than expected	Prioritize 5 core screens (Login, Dashboard, Send, Scan, Transactions), defer rest

---

Summary

Phase	Duration	Tasks	Builders	Validators
1: Foundation	Week 1	5	B1, B3	V1, V3
2: Auth + Security	Week 2	2	B2	V2
3: API Routes	Week 2-3	5	B4	V4
4: UI Alignment	Week 3-4	3	B5	V5
5: Integration + Testing	Week 4-5	3	B4	All
Total	**5 weeks**	**18 tasks**	**5 builders**	**5 validators**

Approve plan? Then run `/build-plan` to execute.

Revision #3
Created 2026-02-26 22:34:46 UTC by John
Updated 2026-05-25 07:24:55 UTC by John