Rollback Plan: Drop — Fintech Payment App

Rollback Plan: Drop — Fintech Payment App

> **Project:** Drop — Remittance + QR Payments **Version:** 0.5.0 **Date:** 2026-02-23
> **Author:** John (AI Director) **Status:** Approved **Reviewers:** Alem Bašić (CEO)

Document History

Version	Date	Author	Changes
0.1	2026-02-23	John	Initial draft — Fly.io deployment on Stockholm region

---

Rollback Summary

Field	Value
Deployment being rolled back	v0.5.0
Rollback target version	v0.4.x (previous stable)
Rollback image / artifact	`registry.fly.io/drop-app:v0.4.x`
DB migration reversible	Yes (Phase 0.5 adds tables only; no destructive migrations)
Estimated rollback time	2–5 minutes (Fly.io blue/green instant rollback)
Rollback owner	John (AI Director)
Backup to restore (if needed)	Fly.io volume snapshot taken before migration

1. Rollback Decision Criteria

Roll back immediately if ANY of these conditions occur:

Trigger	Threshold	Measurement	Wait Before Deciding
Error rate spike	> 1% 5xx errors	Rolling 5-min average on Fly.io metrics	5 minutes
P99 latency spike	> 2,000ms sustained	Rolling 5-min P99 on Fly.io metrics	5 minutes
Health check failures	Any instance unhealthy	Fly.io load balancer health checks	0 minutes (immediate)
Smoke test failure	Any critical Playwright test fails	user-flows E2E suite	0 minutes (immediate)
Data integrity issue	Any confirmed data corruption; `balance` column found in users table	Post-deploy verification (`db.test.ts` assertions)	0 minutes (immediate)
Security vulnerability	Critical severity confirmed (e.g., auth bypass, JWT exposure)	Security alert	0 minutes (immediate)
Pass-through model violation	Drop found to be holding customer funds in any DB column	Schema check	0 minutes (immediate)

Do NOT roll back for:

- Warning-level alerts that were present pre-deployment
- Increased error rate in non-critical paths < 0.5%
- Expected behavior changes (verify against release notes first)
- Cosmetic/visual issues that don't affect functionality
- Mock BaaS timeout errors (expected in MVP; not production-blocking)

---

2. Rollback Authority

Situation	Authority
Automated trigger (smoke test fails)	John (AI Director) — no CEO approval needed
Manual rollback (judgment call, business hours)	John (AI Director) — inform Alem post-rollback
Manual rollback involving data loss risk	Alem Bašić (CEO) approval required
Off-hours manual rollback	John (AI Director) — inform Alem immediately after

Authorization contact: John (AI Director) — Slack: #drop-deploy on alai-talk.slack.com
Emergency escalation: Alem Bašić — +47 40 47 42 51

---

3. Pre-Rollback Assessment

Data Changes Since Deployment

- **Deployment time:** Recorded at time of deployment (see deployment log)
- **Data changes since deployment:** Estimated from Fly.io metrics (transaction count in audit_logs)
- **Critical data at risk:** User registrations, completed transactions (both are financial records)
- **Acceptable to lose transaction data?** No — transactions are financial records; if loss is possible, prefer forward fix over rollback

Decision framework:

- If deployment < 30 min ago and 0 transactions completed → Proceed with rollback
- If deployment > 30 min ago or transactions completed → Escalate to Alem for decision
- Drop is a PSD2 pass-through model — no funds stored; transaction records are audit trail only

Database Migration Reversibility

Migration	Type	Reversible	Down Migration Available
`0005_security_hardening.sql` — Add `audit_logs` table	Add table	Yes (DROP TABLE is safe)	Yes
`0005_security_hardening.sql` — Add `rate_limit_requests` table	Add table	Yes	Yes
`0005_security_hardening.sql` — Add `transaction_locks` table	Add table	Yes	Yes

Phase 0.5 migrations are all additive (add-only). No column drops, no type changes. Rolling back schema is safe.

External System State

System	Events Processed Since Deploy	Reversible	Action if Rollback
Mock BaaS (PISP)	Transaction records in DB only	N/A (mocked)	No action — mock transactions stand
Mock Sumsub KYC	KYC webhook events	N/A (mocked)	No action — mock KYC status stands
Rate limiter DB	Request count records	Yes	No action needed
Audit logs	Immutable log entries	No — by design	No action — audit logs are compliance records

---

4. Rollback Procedures

4.1 Application Rollback (Step by Step)

Total estimated time: 2–5 minutes

```
# Step 1: Announce rollback (required)
# Post in #drop-deploy Slack: "ROLLBACK initiated — v0.5.0 → v0.4.x — Reason: [state reason]"

# Step 2: Trigger rollback deployment via Fly.io
# Option A — Fly.io rollback to previous release:
flyctl releases list --app drop-app  # Find the previous release number
flyctl deploy --app drop-app --image registry.fly.io/drop-app:v0.4.x

# Option B — Fly.io built-in rollback command:
flyctl machine update --app drop-app --image registry.fly.io/drop-app:v0.4.x

# Step 3: Monitor rollback progress
flyctl logs --app drop-app

# Step 4: Confirm rollback complete
curl https://getdrop.no/api/health
# Should return: {"status":"ok","db":"connected","version":"0.4.x"}
```

Verification commands:

```
# Check all instances running rollback version
flyctl status --app drop-app
```

```
# Check health
curl -i https://getdrop.no/api/health

# Run smoke tests against rolled-back version
npx playwright test --project=user-flows

# Verify error rate has dropped
flyctl metrics --app drop-app
```

4.2 Database Rollback (Migration Down)

Warning: Execute database rollback ONLY after confirming:

1. Application rollback is complete
2. Data loss from migration reversal is acceptable (see Section 3)
3. Down migration is available (it is, for Phase 0.5)

```
# Step 1: Confirm current migration state
flyctl ssh console -a drop-app -C "npm run db:migrate:status"

# Step 2: Take emergency backup BEFORE running down migration
flyctl volumes list --app drop-app  # Get volume ID
# Manual: Create volume snapshot via Fly.io dashboard

# Step 3: Run down migration (reverses Phase 0.5 security tables)
flyctl ssh console -a drop-app -C "npm run db:migrate:down"
# Drops: audit_logs, rate_limit_requests, transaction_locks tables

# Step 4: Verify migration state
flyctl ssh console -a drop-app -C "npm run db:migrate:status"
# Should show v0.4.x migrations only

# Step 5: Verify data integrity
flyctl ssh console -a drop-app -C "npm run db:verify-integrity"
```

If down migration fails: Restore from pre-deployment Fly.io volume snapshot

```
# Restore from volume snapshot (requires Fly.io support or volume recreation)
# Contact: https://community.fly.io/ or fly.io/docs/volumes/
```

4.3 Configuration Rollback

```
# Revert environment variables (if changed in this deployment)
# Phase 0.5 added: BCRYPT_ROUNDS, RATE_LIMIT_WINDOW_MS, RATE_LIMIT_MAX_AUTH,
RATE_LIMIT_MAX_GENERAL
flyctl secrets set BCRYPT_ROUNDS=10 --app drop-app  # Only if bcrypt rounds is root cause
# Note: JWT_SECRET must remain set — never remove

# Verify configuration via Fly.io secrets list
flyctl secrets list --app drop-app
```

Changed configuration to revert (if needed):

Variable	New Value (to revert FROM)	Previous Value (to revert TO)
`BCRYPT_ROUNDS`	`12`	`10` (only if bcrypt is root cause)
`NEXT_PUBLIC_SERVICE_MODE`	`mock`	`mock` (no change expected)

4.4 CDN / DNS Rollback

Drop MVP is deployed on Fly.io only (no CDN for API; static assets via Next.js on Vercel for landing page). No DNS changes are expected in Phase 0.5.

If getdrop.no DNS was changed:

```
# Verify current DNS
nslookup getdrop.no

# Revert via domain registrar (Domene.no or current registrar)
# TTL: 300s (5 min) — fast propagation
```

---

5. Verification After Rollback

Health Check Verification

- ☐ `GET https://getdrop.no/api/health` returns HTTP 200 with `{"status":"ok","db":"connected"}`
- ☐ All Fly.io instances showing previous version: `flyctl status --app drop-app`

- [ ] Load balancer health checks green for all instances (2/2 healthy in Fly.io dashboard)

Smoke Test Execution

```
npx playwright test --project=user-flows
```

- [ ] Registration → OTP → PIN flow completes successfully
- [ ] Login + dashboard access verified
- [ ] Remittance flow with mock BaaS verified

Data Integrity Verification

```
flyctl ssh console -a drop-app -C "npm run db:verify-integrity"
```

- [ ] `users` table has NO `balance` column (pass-through model invariant)
- [ ] `cards` table has NO `card_number` or `cvv` columns (PCI-DSS invariant)
- [ ] No orphaned sessions (FK constraint check passes)
- [ ] Transaction types limited to `remittance` and `qr_payment`

Monitoring Verification

- [ ] Error rate returned to pre-deployment baseline (< 0.1%)
- [ ] P99 latency returned to pre-deployment baseline (< 500ms standard, < 1,000ms bcrypt)
- [ ] No unexpected log errors (`flyctl logs --app drop-app`)
- [ ] Fly.io health check shows 2/2 healthy instances

---

6. Communication Plan

Internal Notification

Audience	Channel	When	Message
Alem Bašić (CEO)	Direct (phone: +47 40 47 42 51)	At rollback decision	"Rolling back Drop v0.5.0 — Reason: [X] — ETA: 5 min"

Audience	Channel	When	Message
Engineering (John)	#drop-deploy Slack	At rollback initiation	"ROLLBACK initiated v0.5.0 → v0.4.x"
Validator agent	Mission Control task	Post-rollback	"Verify rollback stability — run smoke tests"

External Notification

Drop MVP is pre-production (no public users). No external status page required.

For Phase 1+ production:

Audience	Channel	When	Trigger
Status page	getdrop.no/status (future)	At rollback initiation	Any production rollback
Affected users	In-app notification	If impact > 30 min	At rollback + recovery

Status page message template (Phase 1+):

```
Vi opplever for øyeblikket et problem med Drop og har startet en tilbakerulling
for å løse det. Vi forventer at tjenesten gjenopprettes innen 10 minutter.
Vi beklager ulempen og vil gi oppdateringer hvert 15. minutt.
```

(Translation: "We are currently experiencing an issue with Drop and have initiated a rollback to resolve it. We expect service to be restored within 10 minutes. We apologize for the inconvenience and will provide updates every 15 minutes.")

---

7. Post-Rollback Analysis

Post-rollback review scheduled: Within 4 hours of resolution **Post-mortem scheduled:** Within 24 hours of resolution (NFR-COMP06 / DORA incident reporting)

Analysis questions:

1. What caused the rollback? (specific code/config/migration change)
2. Could this have been detected earlier? (staging test coverage gap?)
3. Was the rollback executed correctly and within the 5-minute SLA?
4. What process change would prevent this next time?

Output: Log entry in `comms/decisions/` + lessons learned entry in lessons-learned.md

---

8. Forward Fix vs Rollback Decision Matrix

Factor	Favors Forward Fix	Favors Rollback
Time to fix	< 30 min	> 30 min
DB migration	Not included in root cause	Included (rollback simpler)
Transaction data written since deploy	Significant (> 100 records)	Minimal (< 10 records)
User impact severity	P3/P4 — cosmetic or minor	P1/P2 — auth or payment broken
Fix risk	Low — isolated change	High — cascading dependencies
Team availability	Builder agent available	Builder unavailable or offline
Off-hours	Business hours	Off-hours (02:00–06:00 CET)

Default guideline: When uncertain, **rollback**. A rollback to a known good state is safer than a rushed forward fix. Drop handles financial flows — correctness > speed.

Drop-specific rule: If any P1 issue involves the pass-through model invariant (Drop storing money), rollback immediately without waiting for forward fix analysis.

Related Documents

- Deployment Checklist
- Release Notes
- UAT Sign-Off
- Security Audit Report

Approval

Role	Name	Date	Signature
Author	John (AI Director)	2026-02-23	Approved (AI)
Tech Lead	John	2026-02-23	Approved
AI Director (John)	John	2026-02-23	Approved
CEO (Alem)	Alem Bašić	TBD	

Revision #5
Created 2026-02-23 12:06:25 UTC by John
Updated 2026-05-31 20:03:31 UTC by John