🧭 UPRISE RUNBOOK — Developer & Agent Operations Manual

Repository: `music-community-platform`
Last Updated: November 12, 2025 (America/Chicago)
Applies To: All UPRISE Agents (DeepAgent, Claude Code, Codex CLI, Cursor, etc.)

🏗️ Core Directives

File	Purpose	Status
STRATEGY_CRITICAL_INFRA_NOTE.md (./STRATEGY_CRITICAL_INFRA_NOTE.md)	Defines "DeepAgent = Foundry only" rule; production targets (Vercel, Fly, AWS, Neon).	🔴 Critical
PHASE1_COMPLETION_REPORT.md (./PHASE1_COMPLETION_REPORT.md)	Certifies monorepo foundation; establishes readiness for Phase 2.	✅ Complete

> All agents MUST read the two files above before running any task.

⚙️ Development Environment

See ENVIRONMENTS.md (./ENVIRONMENTS.md) for full setup.

Summary:
- **Web / API / Socket / Workers:** WSL 2 (Ubuntu 22.04+ recommended)
- **Mobile (RN 0.66.x):** Windows non-admin PowerShell (Hermes, Gradle 7.0.2, JDK 11)
- **Node:** v22.x (fnm or nvm)
- **Package Manager:** pnpm 9.x (corepack)
- **Database:** Postgres with **PostGIS** (dev: DeepAgent container; prod: Neon or AWS RDS)
- **Optional:** Docker for local workers/DB

🧱 Monorepo Structure

See PROJECT_STRUCTURE.md (./PROJECT_STRUCTURE.md) for details and conventions.

```
apps/
  web/       → Next.js 15 (Vercel)
  api/       → NestJS (Fly.io / App Runner)
  socket/    → Socket.IO (Fly.io / App Runner)
  workers/
    transcoder/ → FFmpeg Node worker (AWS Fargate / Fly.io)
packages/
  ui/        → Shared components
  types/     → Zod schemas → OpenAPI
  sdk/       → Generated client for web/api
infra/
  prisma/    → Prisma schema + PostGIS migrations + seeds
```

Strict Web-Tier Contract
- No DB access, no secrets, no server actions that mutate state in `apps/web`.
- All mutations go through `apps/api`.
- Realtime is subscribe-only via `apps/socket`.

🚀 Deployment Flow

Default pipeline:
DeepAgent (dev/CI) → GitHub PR → External Deploy (Vercel / Fly / AWS) → Production (Neon Postgres)

Each PR MUST include:

```
Deployment Target: [Vercel|Fly|AppRunner|Fargate|Neon]
Phase: [1|2|3]
Specs: [IDs of affected specs, e.g., 04 Community, 07 Discovery]
```

CI runs on PR:
- Lint / Typecheck / Build
- Web-tier contract guard (fail on boundary violations)
- Unit & integration tests
- Prisma PostGIS migrations on ephemeral DB
- Socket realtime smoke test

🧪 Testing Matrix

Test Type	Tool	Location	Schedule
Unit Tests	Jest/Vitest	apps/api, apps/web, apps/socket	On commit
E2E (web)	Playwright	apps/web	Nightly + on release
Realtime	Vitest + Socket test client	apps/socket	On PR
Migrations	Prisma Migrate	infra/prisma	On deploy
Contract Guard	ESLint + custom CI rule	apps/web	On PR

🔒 Web-Tier Contract Guard (T5 Implementation)

Purpose

The Web-Tier Contract Guard enforces strict architectural boundaries to prevent direct database access, server-side imports, and secret leakage in the web tier (`apps/web`). This is a critical infrastructure policy that ensures the UPRISE platform maintains proper separation of concerns.

What It Does

The guard scans all TypeScript/JavaScript files in `apps/web` and detects:

❌ **PROHIBITED Patterns:**
- Direct database imports (`@prisma/client` , `pg` , `mongodb` , `mongoose` , etc.)
- Direct imports from `apps/api/src` or `apps/socket/src`
- Server-side environment variables (`DATABASE_URL` , `JWT_SECRET` , `AWS_SECRET_ACCESS_KEY` , etc.)
- Non- `NEXT_PUBLIC_` environment variables in client components
- AWS SDK imports (`aws-sdk` , `@aws-sdk/*`)
- File system access (`fs` module)
- Server-only Node.js modules (`child_process` , etc.)

✅ **ALLOWED Patterns:**
- API client (`import { api } from "@/lib/api"`)
- Socket.IO client (`import { io } from "socket.io-client"`)
- Shared packages (`@uprise/ui` , `@uprise/types` , etc.)
- `NEXT_PUBLIC_` environment variables
- Client-safe utilities and components

Running Locally

```
# Run the guard
pnpm run infra-policy-check

# Show help and documentation
pnpm run infra-policy-check --help

# Verbose output with file counts
pnpm run infra-policy-check --verbose
```

CI Integration

The guard runs automatically on every PR and push to `main` or `develop` branches via GitHub Actions. If violations are detected, the build will fail and must be fixed before merging.

Workflow file: `.github/workflows/infra-policy-check.yml`

Error Codes

Each violation includes a specific error code for easy identification:

Code	Description
`WEB_TIER_DB_001` - `WEB_TIER_DB_007`	Database access violations
`WEB_TIER_IMPORT_001` - `WEB_TIER_IMPORT_004`	Server-side import violations
`WEB_TIER_SECRET_001` - `WEB_TIER_SECRET_006`	Environment variable/secret violations
`WEB_TIER_AWS_001` - `WEB_TIER_AWS_002`	AWS SDK violations
`WEB_TIER_FS_001` - `WEB_TIER_FS_002`	File system access violations
`WEB_TIER_SERVER_001` - `WEB_TIER_SERVER_002`	Server-only module violations

Example Violation Output

```
❌ Web-Tier Contract Violations Detected (ERRORS):
════════════════════════════════════════════════════════════════════════════════

1. apps/web/src/lib/db.ts:5:1
   Code: WEB_TIER_DB_001
   Message: Direct Prisma Client import is prohibited in web tier. Use API client instead.
   Snippet: import { PrismaClient } from '@prisma/client';

2. apps/web/src/components/user-profile.tsx:10:15
   Code: WEB_TIER_SECRET_001
   Message: DATABASE_URL must not be accessed in web tier. This is a server-side secret.
   Snippet: const db = process.env.DATABASE_URL;

════════════════════════════════════════════════════════════════════════════════

❌ Total Errors: 2
```

How to Fix Violations

Instead of direct database access:

```typescript
// ❌ WRONG
import { PrismaClient } from '@prisma/client';
const prisma = new PrismaClient();
const users = await prisma.user.findMany();

// ✅ CORRECT
import { api } from '@/lib/api';
const users = await api.get('/users');
```

Instead of server-side secrets:

```typescript
// ❌ WRONG
const secret = process.env.JWT_SECRET;

// ✅ CORRECT
const publicKey = process.env.NEXT_PUBLIC_API_KEY;
```

Instead of direct API imports:

```typescript
// ❌ WRONG
import { UserService } from '../../api/src/users/user.service';

// ✅ CORRECT
import { api } from '@/lib/api';
// Or use shared types from @uprise/types
```

Related Documentation

- `STRATEGY_CRITICAL_INFRA_NOTE.md` (./STRATEGY_CRITICAL_INFRA_NOTE.md) - Infrastructure policy
- `PROJECT_STRUCTURE.md` (./PROJECT_STRUCTURE.md) - Architectural boundaries
- `apps/web/WEB_TIER_BOUNDARY.md` (../apps/web/WEB_TIER_BOUNDARY.md) - Web-tier contract details

Script Location

- **Source:** `scripts/infra-policy-check.ts`
- **Legacy (deprecated):** `scripts/infra_check_web.js`

📚 Documentation Index

Category	File	Description
Strategy	STRATEGY_CRITICAL_INFRA_NOTE.md (./STRATEGY_CRITICAL_INFRA_NOTE.md)	Infrastructure policy
Milestones	PHASE1_COMPLETION_REPORT.md (./PHASE1_COMPLETION_REPORT.md)	Phase 1 completion
Specs	Specifications/README.md (./Specifications/README.md)	Module-by-module technical docs
Environments	ENVIRONMENTS.md (./ENVIRONMENTS.md)	Windows/WSL setup rules
Structure	PROJECT_STRUCTURE.md (./PROJECT_STRUCTURE.md)	Folder map & conventions
Changelog	CHANGELOG.md (./CHANGELOG.md)	Auto-generated PR logs

---

🧩 Agent Rules

1. **Follow the Critical Infra Note** — DeepAgent may run tests, not production workloads.
2. **Keep Docs Current** — Every merged PR must update `CHANGELOG.md` and, if scope touches architecture or ops, update this `RUNBOOK.md`.
3. **Annotate PRs** — Link to affected specification(s) in `/docs/Specifications`.
4. **Blockers** — Any CI error tagged `infra-policy-check` halts merge until fixed.

🔄 Maintenance Schedule

Interval	Task	Owner
Daily	CI: lint/type/build pass	DeepAgent
Nightly	E2E (web) + socket smoke	DeepAgent
Weekly	Update CHANGELOG from merged PRs	DeepAgent
Per Phase	Publish phase completion report	PM/Lead Agent

🏁 Quickstart Commands

```
pnpm i
pnpm -r build
pnpm -r dev
```

See ENVIRONMENTS.md for full setup and service URLs.