Contract to Cozy - App Summary

Evidence source: repository code and docs only

What it is

Contract to Cozy is a web platform for homeowners to manage properties, maintenance, risk, costs, and service bookings. It also includes a provider portal for service professionals.

Who it's for

- Primary persona: homeowners (especially EXISTING_OWNER and HOME_BUYER segments) managing one or more homes.
- Secondary persona: service providers using a separate provider dashboard and booking workflow.

What it does

- Role-based access for homeowner, provider, and admin flows.
- Property lifecycle management: create, edit, view, and track multiple properties.
- Maintenance operations: checklists, seasonal tasks, warranties, insurance, expenses, and documents.
- Risk and finance insights: risk assessment, financial efficiency, appreciation, budget, and home cost tools.
- Service marketplace: find providers, book appointments, and manage bookings.
- Property intelligence features: inventory and rooms, incidents and claims, recalls, and report export.

How it works (architecture)

- Frontend: Next.js 14 app with route-based dashboards, React Query, AuthContext, PropertyProvider, and NotificationProvider.
- Client data layer: centralized API client uses NEXT_PUBLIC_API_URL and JWT-based auth tokens.
- Backend: Express API mounts domain routes under /api/* and exposes Swagger docs at /api/docs.
- Persistence: Prisma ORM to PostgreSQL; Redis supports async queues and background processing.
- Workers: BullMQ + cron jobs process risk/FES calculations, notifications, seasonal checklist events, and recalls.
- Flow: UI action -> API client -> Express controller/service -> Prisma/Postgres; long jobs -> Redis queue -> workers -> DB -> UI refetch.

How to run (minimal)

- From repo root: run `make install`.
- Start local stack: run `make dev` (Docker Compose starts frontend, backend, postgres, redis, workers).
- Open app: http://localhost:3000 ; API docs: http://localhost:8080/api/docs.
- Seed data / default login credentials: Not found in repo.