Low Level Design – Project Ops V2

Backend Structure

Tech: Node.js (TypeScript recommended) + Express + Mongoose for MongoDB.

Folders: /config, /models, /integrations, /services, /routes, /middlewares, /utils.

Core models: Project, Service, EnvVar, Metric, optional WorkItem.

Model Updates (V2)

Project: add lifecycleStage, priority, nextAction, targetReleaseDate.

Service: add type='automation' support, provider='make', lastRunAt, lastRunStatus, expectedFrequencyMinutes.

Make Webhook Flow

Make scenario ends with HTTP module → POST /internal/make/report with X-Internal-Secret header.

Body includes: scenarioId, scenarioName, status, startedAt, finishedAt, errorMessage.

Backend finds Service by provider='make' + providerInternalId=scenarioId.

Updates service.lastRunAt, service.lastRunStatus, service.status (ok/failing).

Inserts Metric records: automation_status, automation_duration_ms.

Scheduling / Cron Jobs

Health sync: Render Cron → POST /internal/sync/health every 5 minutes.

Deploy sync: POST /internal/sync/deploys every 15 minutes.

DB health: POST /internal/sync/db-health hourly.

Automation health: POST /internal/sync/automation-health every 10–15 minutes (detect stale scenarios).

Cleanup: POST /internal/sync/cleanup nightly (old metrics).

Frontend Structure

Pages: DashboardPage (projects overview), ProjectDetailsPage (services + dev info).

API hooks: useProjects (GET /api/summary or /api/projects), useProjectDetails (GET /api/projects/:id).

Components: ProjectCard, ServiceStatusBadge, AutomationStatusList, MetricsChart (basic).