RemoteIQ — Full Stack Roadmap & Execution (ChatGPT)

Role: Act as senior full‑stack engineer and PM. Generate code, config, and docs for a production‑grade, open‑source RMM (RemoteIQ). Coordinate with Gemini outputs for UI/UX and design tokens.

# Architecture Overview

• Frontend: Next.js 15, TypeScript, Tailwind, shadcn/ui, TanStack Query, Zustand, Zod.

• Backend: FastAPI (Python) or NestJS (TypeScript) — choose one per phase; expose OpenAPI.

• Database: Postgres (primary), Redis (jobs/cache).

• Agents: Windows (.msi + PowerShell), macOS (.pkg + script), Linux (.deb/.rpm + bash).

• Messaging/Jobs: NATS or RabbitMQ; Celery/BullMQ for async.

• Realtime: WebSockets (job streams, status).

• Packaging: Docker, Compose; optional Kubernetes later.

• Security: JWT short TTL + refresh; mTLS agent channel; CSP; rate limits.

# Phase B0 — Repo Setup & CI

– Monorepo (turbo or pnpm workspaces) or split repos with shared OpenAPI.

– CI: lint, tests, build, container publish.

– Dev containers; .env.example with sane defaults.

Definition of Done: CI green; local `docker compose up` boots API + DB.

# Phase B1 — Identity & Tenancy

– Users, Orgs, Roles/Policies (RBAC).

– Tenant scoping in every query; audit trail.

– OAuth (optional), SSO ready.

DoD: CRUD + seeds; e2e auth flow; OpenAPI schemas stable.

# Phase B2 — Devices Ingest & Inventory

– Device model (org/site/department/agent type/tags).

– Ingest endpoints for agents (heartbeat, facts, metrics).

– List/filters pagination for /devices (matches frontend spec).

DoD: Generate typed client for frontend; seed scripts; fixtures.

# Phase B3 — Automation: Script Runs

– Job model, queue, and dispatcher.

– Script registry (PowerShell, Bash, Python), versioning, provenance.

– Target resolution (explicit, filters, saved views).

– Live logs via WebSocket; artifact storage (S3/MinIO).

DoD: 10k‑target enqueue < 5s; back‑pressure; retries; status transitions.

# Phase B4 — Patching & Software

– Patch metadata ingest (OS specific).

– On‑demand patch and maintenance windows.

– Software inventory ingest & uninstall hooks.

DoD: API invariants documented; idempotent ops; audit entries.

# Phase B5 — Reporting

– Scheduled reports, recipients, formats (PDF/CSV).

– Templating with pre‑built queries; signed download URLs.

DoD: Jobs queued reliably; email/webhook deliveries; retries.

# Testing & Quality

• Unit tests: >70% critical modules; e2e: happy paths.

• Contract tests for OpenAPI; client re‑gen on CI.

• Security checks: OWASP top 10; dependency audit.

• Observability: logs, traces, metrics dashboards.

# DevOps

• Dockerfiles (multi‑stage, non‑root).

• Compose with profiles (dev, demo, prod).

• Migration strategy (SQLAlchemy/Alembic or Prisma/TypeORM).

• Backup/restore; seed scripts.

# Agent Deliverables

• Windows: MSI + PowerShell bootstrap; service; signed if possible.

• Linux: systemd service; .deb/.rpm; curl|bash installer.

• macOS: launchd service; notarization path later.

• Features: heartbeat, facts (CPU/RAM/Disk/OS), metrics, remote exec, file ops.

• Security: mTLS, rotating keys; outbound‑only traffic.

# Sync With Frontend

• Generate OpenAPI client (/sdk).

• Fields naming parity; enums and error codes aligned.

• Streaming endpoints for runs/patches to power Live Run drawer.

# Phase Checklists

[ ] ADR updated

[ ] API & schema merged

[ ] Tests + coverage

[ ] Docs: README + OpenAPI

[ ] Demo script (/scripts/demo/\*.ps1|\*.sh)