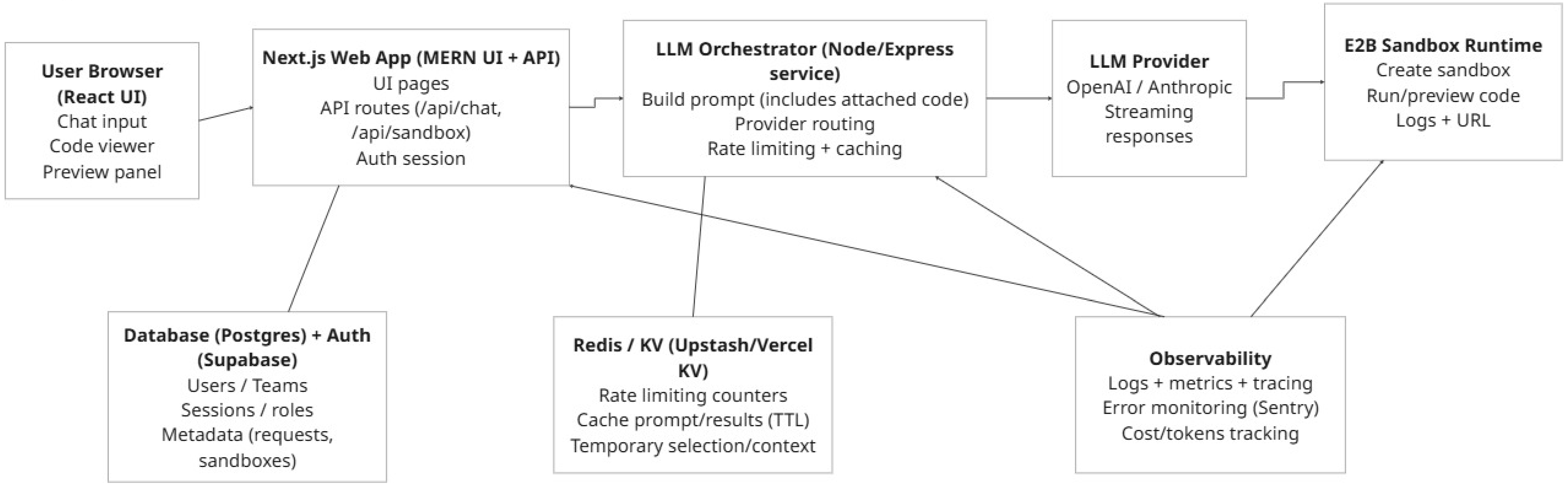

User Browser (React UI)
Chat input
Code viewer
Preview panel
Next.js Web App (MERN UI + API)
UI pages
API routes (/api/chat, /api/sandbox)
Auth session
LLM Orchestrator (Node/Express service)
Build prompt (includes attached code)
Provider routing
Rate limiting + caching
LLM Provider
OpenAI / Anthropic
Streaming responses
E2B Sandbox Runtime
Create sandbox
Run/preview code
Logs + URL
Database (Postgres) + Auth (Supabase)
Users / Teams
Sessions / roles
Metadata (requests, sandboxes)
Redis / KV (Upstash/Vercel KV)
Rate limiting counters
Cache prompt/results (TTL)
Temporary selection/context
Observability
Logs + metrics + tracing
Error monitoring (Sentry)
Cost/tokens tracking

Scalability (1k → 10k+)
Stateless services + horizontal scaling
CDN for static assets
Redis cache + rate limits per team
Queue to absorb spikes
DB: pooling + read replicas later

Monitoring & Observability
Metrics: p95 latency, error rate, cache hit, sandbox failures
Logs: requestId, userId/teamId, model, token count
Tracing: end-to-end (UI → API → provider → sandbox)
Alerts: provider errors spikes / budget overrun
Dashboard per team (usage + failures)

Bottlenecks & Mitigation
LLM latency/cost → cache + model routing
Sandbox spin-up → queue + warm pool (future)
Large code context → strict limits + truncation
DB connections → pooling + reduce sync writes

Reliability & Fault Tolerance
Timeouts + retries (with backoff)
Circuit breaker per provider
Fallback model/provider when limited
Graceful UI errors (code-only if sandbox fails)
Idempotency key for /api/chat requests

CI/CD & Deployment
CI: lint + typecheck + build + basic tests
Staging on every merge to main
Production deploy with manual approval
Canary/blue-green rollout (small % first)
Secrets via env/secret manager (no keys in repo)

Security & Multi-tenancy
Supabase JWT auth + RBAC (user/team roles)
Data isolation by teamId in DB + policies
Rate limit & quotas per team
Validate inputs + payload size limits
Sandbox isolation + TTL cleanup (no secrets inside)

LLM Cost Optimization
Token budget per request (hard limits)
Truncate attached code + chat history
Cache repeated prompts/results (TTL)
Model routing (cheap draft → expensive final)
Usage quotas per team + alerts

Go-live Plan + Cost Drivers
Start: internal demo → small beta → gradual rollout
Biggest costs: LLM tokens + sandbox runtime
Controls: quotas, caching, model routing, TTL sandboxes
Track: cost/user/day + top heavy teams
Incident plan: disable sandbox / switch provider quickly