# The Wheel – Technical Architecture

# The Wheel – Technical Architecture

## 1. Overview The Wheel is built as a modular, role-aware SaaS platform with a React + TypeScript frontend, a Supabase backend (PostgreSQL, Auth, Storage, Edge Functions), and OpenAI-powered AI integration. It follows a microservice-style design via isolated modules with shared services and event-based communication where needed.

## 2. Architecture Layers

### 2.1 Frontend (React + TypeScript)  
- Role/Mode-aware context system (via React Context + Zustand)  
- Dynamic component rendering based on current persona and company stage  
- Lazy loading and route-based code splitting  
- Responsive design system (Tailwind-based)  
- Dashboard layout engine with plug-in widgets (tasks, milestones, AI insights)

### 2.2 Backend (Supabase/PostgreSQL)  
- Multi-tenant schema with row-level security for company, user, and mode context  
- Tables for journeys, tasks, milestones, stages, templates, resources, tool evaluations, etc.  
- Event-driven updates via Supabase Edge Functions  
- JWT-based auth, scoped to active mode and user context  
- Full-text search and taxonomy via PostgreSQL indexing

### 2.3 AI Layer  
- GPT-4 based AI Cofounder integration  
- Standup bot summarization  
- Task-based recommendation engine  
- Decision support module (pros/cons evaluation, risk alerts)  
- Prompt templates stored in Supabase and updated dynamically

### 2.4 Integration Layer  
- OAuth connectors for GitHub, Google, Notion, Slack, CRMs  
- Sync service for calendar, doc, and task data  
- Integration health dashboard  
- Notification aggregator and delivery prioritizer

## 3. Security Architecture - JWT-based mode/context authentication - Role- and mode-based permissions matrix - Audit logging for sensitive actions - End-to-end encryption on PII and user data - Field-level visibility rules for templates, RFPs, and progress

## 4. Extensibility - Plugin framework (planned in Phase 4) - Custom workflow builder with schema validation - Public API with scoped rate-limiting - Developer sandbox mode

## 5. Observability - Sentry for error tracking - DataDog for performance monitoring and tracing - Event logs linked to journey context - Usage analytics for cohort-level product insights

## 6. Deployment - Vercel frontend hosting with auto-preview and CI/CD - Supabase backend with usage-based scaling - Future optional support for cloud-provider migration via Terraform templates

## 7. Key Services and Tables (Partial) - `journeys`, `stages`, `milestones`, `tasks`, `task\_dependencies` - `companies`, `users`, `user\_modes`, `mode\_preferences` - `templates`, `resources`, `tool\_evaluations`, `rfps`, `vendors` - `ai\_insights`, `activity\_logs`, `notifications`, `standup\_entries`