eWorkConnection

Design and Architecture Document

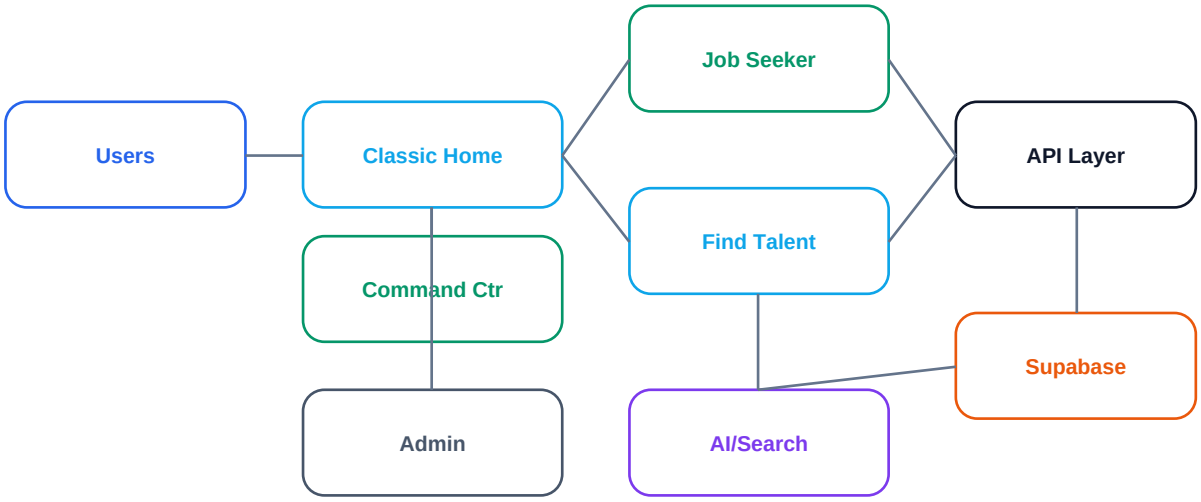
System design, component responsibilities, data architecture, security controls, and launch operations.

Document	Design and Architecture Document
Version	Soft Launch planning set
Date	June 20, 2026
Audience	Founder, admins, developers, QA employee agent, launch partners

Prepared for launch planning. This document is not legal, tax, employment, immigration, financial, or security advice; local counsel and tax advisors should review country-specific launch materials before paid scale.

1. Architecture Overview

The application is built as a Next.js app deployed through Vercel, with Supabase for authentication, database, and row-level security. Server-side API routes mediate AI calls, budget enforcement, tracking, support chat, event logging, and admin actions.



Control points: privacy acknowledgment, role guard, country launch controls, API budget guard, RLS/service role, audit/event logging.

2. Component Map

Layer	Components	Responsibilities
Frontend	Classic Home, Career Command Center, Dashboard, Recruiter, Company Workspace, Admin, Services, Event Log	Role-based UI, alternate entry points, search workflows, filters, consent, application tracking, job posting, employer verification, recruiter network, support and feedback entry.
API Routes	/api/claude, /api/career-briefing, /api/search-cache, /api/recruiter/*, /api/admin/*, /api/support/chat, /api/plans	Server-side AI proxy, cached daily briefing, cost controls, company/public job search, recruiter jobs/workspaces, event ingestion, admin updates, support automation, pricing plan lookup.
Data	Supabase tables and RLS	Profiles, applications, jobs, recruiter job posts, company workspaces, pipeline items/events, user promotions, career briefing cache, usage events, country pricing, peer/recruiter network, event logs, support tickets, budget settings.
Automation	GitHub Actions employee/QA/developer workflows	Scheduled triage, regression test, email summary, issue creation/handoff, developer fix workflow.
External Services	Anthropic/API key, Resend, Stripe/possible payment providers, public search sources	AI summarization/search generation, email notifications, payments, public data discovery.

3. Data and Regional Design

- Country is detected from profile, browser/launch settings, or user selection and stored for country-specific trial/pricing/access decisions.
- Launch availability is controlled by `country_pricing` settings: country active flag, trial start/end date, free trial days, extension days, and pricing.
- User-specific promotions live separately from country/global coupons and are selected during checkout before country/global promotions.
- Company workspace data is scoped by owner/member records and should never expose another company's internal jobs or pipeline to unrelated users.
- Future regional data residency should separate users by region: North America, Europe/UK, Latin America, India/South Asia, and other expansion zones.
- For EU/UK expansion, data minimization, consent withdrawal, deletion workflows, DPA/vendor review, and cross-border transfer review are required before enabling launch.

4. Security and Privacy Controls

Security is defined as layered controls across identity, authorization, browser exposure, server-side APIs, database policies, cost governance, and operational monitoring. The application should assume that normal users can inspect browser code and direct URLs, so privileged actions must be enforced on the server and in Supabase policies, not only hidden in the UI.

Control	Where Defined	Risk Reduced
Authentication	Supabase OAuth/session and auth callback role routing.	Prevents anonymous access to account-only pages and creates a trusted session boundary.
Authorization	Admin page guard, built-in super-admin list, profile role flags, country admin scopes.	Prevents direct URL access to admin/super-admin functions and limits country-specific control.
Database Access	Supabase RLS policies plus service-role-only server routes for privileged reads/writes.	Prevents client-side users from reading or modifying protected rows outside policy.
Secrets	Vercel environment variables for platform API keys; no platform key exposed to the browser.	Reduces key theft and prevents users from bypassing budget controls.
Same-Origin API	Security helper and API route checks for allowed origins/headers.	Reduces cross-site abuse of paid processing endpoints.
Budget Guard	API usage tables, admin budget route, call-type tracking, maintenance-mode enforcement.	Limits financial exposure from runaway usage, attack traffic, or search loops.
Consent	Service-entry privacy acknowledgment stored in cookie/localStorage; consent withdrawal requirement.	Improves notice/consent posture and reduces repeated user interruption.
Employer Verification	Admin-reviewed recruiter verification fields and trigger/policy restrictions against self-approval.	Prevents unverified recruiters from activating sensitive hiring flows.
Company Workspace	Workspace/member tables, active/verified/public availability filters, and server-side search inclusion rules.	Prevents internal company jobs from leaking to public search.
Auditability	Event log, user-reported issue metadata, QA artifacts, downloadable diagnostic traces.	Supports defect triage, abuse review, and evidence for operational decisions.

Control	Where Defined	Risk Reduced
UI Safety	Audit/trace UI hidden from normal users; feedback and support placed away from core navigation.	Reduces accidental exposure of internal diagnostics and improves usability.

5. Security Control Design Principles

- Server enforcement first: UI hiding is only a usability layer; API routes and database policies must enforce the real boundary.
- Least privilege: normal users get service flows, admins get operational controls for assigned countries, and super admins own sensitive platform-wide switches.
- No browser secrets: platform AI/API keys stay in Vercel and are called through server routes with budget checks.
- Data minimization: store only what is needed for account, search, consent, tracking, and support workflows; public data should be labeled and verified by users.
- Auditable change path: feature flags, budget updates, trial resets, user role changes, and maintenance toggles should leave an event trail.
- Country-aware compliance: country launch state, trial, pricing, and data handling should be configurable before the app is opened in that market.

6. AI and Search Flow

- The user initiates a search; the app checks launch, role, budget, maintenance, and enabled sources.
- The server API route calls the AI provider using the platform key unless a user explicitly brings their own key.
- Results are normalized into job/candidate records and stored or displayed with source, freshness, confidence, and validation messaging.
- Subsequent filter/sort actions run client-side against loaded results to reduce cost and latency.
- Completed job and talent searches are stored as browser-local search runs for guest/trial usability; signed-in server records remain the authoritative database-backed source for persisted account analytics.
- Career Command Center merges server counts with browser-local search-run counts so immediately completed searches are visible even before database persistence catches up.
- Job link validation performs deterministic URL checks, server-side page checks, closed-posting phrase checks, and AI active-posting review before results are shown.
- If exact results are empty, fallback/similar suggestions are generated where appropriate, and ordinary zero-result cases should not be logged as defects.

7. Operations Model

Operational Area	Process
Release	Developer commits to GitHub; Vercel deploys; QA agent validates public/auth/admin flows.
Employee Agent	Runs 7 AM and 5 PM America/Chicago; reviews event log, runs regression, sends email summary, and creates actionable issues.
Developer Agent	Receives approved issues or handoff items, prepares fixes, runs tests, and submits for admin/super-admin approval before production.

Operational Area	Process
Incident	Budget breach, auth failure, search failure, or high-severity event triggers maintenance for processing and admin email notification.
Country Expansion	Enable country in admin, set trial/pricing, review legal/privacy/tax, run country QA, then market to target segment.

8. Technical Risks and Mitigations

Risk	Mitigation
AI search cost spikes	Budget guard, call-type tracking, model tiering, caching, filter-only changes, and admin-maintenance fallback.
Public data accuracy	Source labels, verification warnings, confidence scores, and no guarantee language.
Role confusion	Clear role selection, role guard, direct navigation to selected role page, and dual-role support.
Mobile overcrowding	Collapsible sidebars/advanced controls, bottom-left feedback, bottom-right chat, and hidden diagnostic bars for normal users.
Compliance drift by country	Country launch controls, local policy review, consent withdrawal, tax/legal sign-off before paid launch.