

BUILT BY AN OPERATOR · ENGINEERED FOR SCALE

The technology behind VendGO.

VendGO is a multi-tenant SaaS platform for field service operators — built on a modern stack that prioritizes reliability, security, and operator workflows over engineering theater. Every tool below was chosen deliberately. Everything we deliberately don't use is on the last page.

IN THIS DOCUMENT

[01 Multi-Tenant Architecture](#)[02 Application Layer](#)[03 Database & Backend](#)[04 Communications](#)[05 Reliability](#)[06 End-User Experience](#)[07 Engineering Discipline](#)[08 The Stack at a Glance](#)

WHY THIS MATTERS

Most operational software is built by engineers who've never run a route. VendGO is built by an active operator (91 machines, 70 locations) who knows exactly where the spreadsheet model breaks. **Every architectural choice is in service of a real workflow, not engineering trend-chasing.** When you see "no Kubernetes" on the last page, that's not a gap — it's intentional restraint.

1 THE FOUNDATION

Multi-Tenant Architecture

VendGO serves every customer from a single codebase, on a single deployment, with rock-solid data isolation. New tenants onboard in minutes, not weeks. One upgrade ships to everyone — but no tenant ever sees another tenant's data.

Subdomain-Based Tenant Routing

Each customer gets their own URL — **theirname.vendgo.co** — fully branded with their own colors, logo, company name, and credit card reader naming. New tenants are provisioned in minutes through the super-admin panel. Wildcard SSL on Vercel means every new tenant gets HTTPS instantly with zero manual certificate work.

Postgres Row Level Security (RLS)

Every database table enforces tenant isolation **at the database level**, not just in application code. Every query is automatically filtered by `company_id` before any row is returned. Even if VendGO application code has a bug, the database itself refuses to return another tenant's data. Defense in depth — the kind of security posture a serious operator demands.

Per-Tenant Feature Toggles

Every tenant has a feature toggle table controlling which capabilities are enabled — Issues, Repairs, Maintenance, Inventory, Invoicing, Property List, Nayax Log, Documents. Pricing tiers map to feature sets. Custom features can be enabled for specific tenants. Kill-switch for any feature without code deploys.

Role-Based Access Control

Six standardized roles — **super_admin**, **owner**, **admin**, **manager**, **office**, **tech** — with strict permission hierarchies enforced both in the database (RLS) and the application layer. Only super-admins can promote users to owner or admin. Granular permission tables let each tenant customize what each role can do.

Temporary Cross-Route Access (TRA)

When a tech needs to cover another tech's route — vacation, sick day, equipment outage — admins grant time-limited access with feature-level granularity. Brian covers Brandon's route for two weeks, with access to Issues and Repairs only, not Invoices. Grants expire automatically. Full audit trail on every grant and revocation.

Per-Tenant Branding System

Colors, logos, company names, contact info, and even feature labels (e.g. "Nayax Log" vs "Credit Reader Log") are customer-configurable. Add a new tenant and the entire app reskins automatically — no code changes, no template duplication. One codebase, every customer sees their own brand.

2 THE APPLICATION LAYER

Next.js 16 · React 19 · TypeScript

A modern web framework powering single-page-app responsiveness, server-rendered performance, and end-to-end type safety. Fast for the operator. Reliable under the hood.

Server-Side Rendering (SSR)

Pages render on the server for fast first paint and small client bundles. A tech opening the Issues page on a slow LTE connection in a basement still gets meaningful content within a second. Better SEO, better mobile performance, lower data costs for users on metered plans.

TypeScript Strict Mode

Every variable, every function, every API request and response is typed. The compiler catches whole classes of bugs before code ever runs in production. **Zero `any` types. Zero ignored type errors.** This is the kind of discipline that lets a solo dev maintain a codebase that would normally take a five-person team.

Zod Schema Validation on Every API

Every API request is validated against a typed schema before touching the database. Bad input — wrong types, missing fields, malformed UUIDs — is rejected at the boundary with clear error messages. Catches mistakes early, makes security holes much harder to introduce.

Typed Error Handling (AppError Pattern)

Every error in the system carries a typed error code, HTTP status, and structured details. Errors are consistent across endpoints, easy to debug, and never leak internal stack traces to users. The same error code in two different parts of the system means the same thing.

App Router Architecture

Next.js App Router with server components by default, client components only when needed. Smaller JavaScript bundles, faster page loads, better Core Web Vitals. Every page is colocated with its data fetching, server actions, and TypeScript types — one folder per feature.

Tailwind CSS & Design System

A utility-first CSS framework backed by a consistent design system — neomorphic dark theme, light theme on the roadmap. CSS variables centralize colors, spacing, and shadows. Every page looks like it belongs to the same product because it actually does.

3 WHERE THE DATA LIVES

Database & Backend · Supabase + Postgres

Supabase is an enterprise-grade backend platform built on PostgreSQL — the most battle-tested open-source relational database in the world. It gives VendGO a real database (not a NoSQL toy), production-grade authentication, and global file storage out of the box.

Dedicated PostgreSQL Database

A full relational database with indexes, foreign keys, triggers, and SQL — not a key-value store. Every relationship in the business (techs to routes, machines to locations, repairs to parts) is modeled as proper relational data. Reports are fast because they're real SQL queries, not API call fan-out.

Cookie-Based Server Authentication

Sessions are stored in HTTP-only cookies, readable by the server on every request. This is the correct pattern for modern web apps — older token-in-localStorage approaches break server-side rendering and create XSS risk. Email, magic links, password reset, all handled.

S3-Compatible File Storage

Photos of broken machines, signed contracts, inventory documentation — all stored in private buckets with Row Level Security applied. A tech's photos belong to their tenant, never visible to anyone else. Signed contracts uploaded to admin.vendgo.co are accessible only to super-admins, period.

Auto-Generated REST APIs

Every new database table is queryable via REST out of the box — no boilerplate code to write. When VendGO ships a new feature, the API endpoints exist immediately and inherit Row Level Security automatically. This is what enables a solo dev to ship features at the pace of a small team.

Audit Logging

Every administrative action — company creation, feature toggle, user role change, permission grant — writes to an immutable audit log with actor identity, timestamp, and before/after state. **If a question ever arises six months later about who did what, the answer is in the database.**

Automatic Backups & Point-in-Time Recovery

Daily database backups with point-in-time recovery available. If a catastrophic bug ever deleted important data, we could roll back to a specific minute in time. Backup snapshots are stored separately from the live database — disaster-recovery ready.

4 HOW CUSTOMERS AND TECHS GET REACHED

Communications · Twilio + Resend

The platform's entire value proposition depends on reaching the right person at the right moment. Twilio handles SMS and voice. Resend handles email. Both are tenant-branded so customers see the operator's identity, not VendGO's.

Programmable SMS via Twilio

Instant tech notifications when an issue is reported. Two-way SMS conversations with customers. Self-report links sent automatically when a customer calls. Every tenant gets their own dedicated Twilio phone number — customers see the operator's brand, not VendGO's.

A2P 10DLC Registration

Application-to-Person SMS in the US requires regulatory registration to deliver reliably. VendGO handles 10DLC registration through its compliance pipeline so tenant messages aren't filtered as spam. Higher deliverability, fewer dropped texts, more recovered tickets.

Automated Voice Answering Service

Tenants forward their business phone line to a Twilio number. Inbound customer calls are answered automatically — the system prompts for issue reporting or refund request, uses geo-fencing and machine ID to identify the location, and SMS-links a self-report form. **Calls that previously went to voicemail are now fully automated.**

Transactional Email via Resend

Approval notifications, signup confirmations, password resets, invoice deliveries — all sent through Resend with high deliverability. Every email is tenant-branded: the operator's company name in the subject line, the operator's colors in the header, the operator's address in the footer.

Per-Tenant Branded Templates

Email templates pull from each tenant's branding configuration — colors, logo, company name, contact info. Add a new tenant and their emails look like theirs, automatically, with no template duplication. One template system, dynamically rendered per tenant.

Webhook-Driven Event Routing

Twilio and Resend events (delivery confirmations, bounces, replies) hit VendGO webhooks that update the database in real time. Failed deliveries are surfaced in admin dashboards. Two-way SMS replies route back to the assigned tech automatically.

5 HOW THE PLATFORM STAYS UP

Reliability · Sentry + Vercel

Software breaks. The question is how fast it gets noticed and fixed. VendGO's reliability stack catches errors the moment they happen, deploys fixes globally in seconds, and gives the team a complete picture of what every user was doing when something went wrong.

Sentry · Real-Time Error Tracking

The moment any user — tech, office staff, customer — hits a bug, Sentry captures the exact line of code, the full stack trace, the user's browser, and the context. **The bug arrives in the inbox before the user has finished typing a support email.** Most errors are fixed before the user reports them.

Sentry · Session Replay

When a tech reports "the inventory page froze," there's a video-like recording of their exact session — every click, every loaded element, every error. Diagnosis goes from "can you reproduce that?" to "I see the bug, fixing now." **Cuts support resolution time from days to minutes.**

Performance Monitoring

Every page load, every API call, every database query is measured. When something starts getting slower — even before users complain — alerts fire. Performance regressions get caught the day they ship, not three weeks later when customers churn.

Vercel · Global Edge Deployment

VendGO is deployed to a global edge network — servers in dozens of cities worldwide. A tech in Florida and a tech in California get the same sub-100ms response times. Auto-scaling handles traffic spikes without infrastructure changes.

Preview Deployments Per Code Change

Every code change gets a live preview URL before going to production. New features are tested on real infrastructure, not just a developer's laptop. Catches deployment-specific bugs that "works on my machine" never finds.

Wildcard SSL for Every Tenant

Every tenant subdomain — current and future — is served over HTTPS automatically. No certificate provisioning per customer. No expiration tracking. Security that's invisible because it just works.

6 WHAT OPERATORS AND CUSTOMERS ACTUALLY TOUCH

End-User Experience · PWA + QR Codes

The technology only matters if real people will use it. VendGO is engineered so that techs in mechanical rooms, customers at broken machines, and owners checking metrics from a couch all get a fast, native-feeling experience.

Progressive Web App (PWA)

VendGO installs to a phone or desktop home screen with one tap — no app store, no review process, no waiting. The app launches in its own window without browser chrome. Updates ship instantly to every device with zero user action.

Offline-Capable for Field Work

Techs working in mechanical rooms, basement parking, or rural areas with no signal can still log repairs, update inventory, and view their route. Data syncs automatically the moment connectivity returns. **Field service software that doesn't work offline isn't field service software.**

Push Notifications (Android)

When an issue is reported on a tech's route, their phone notifies them — even with the app closed. No app to keep open in the background, no notification setup beyond installation. iOS support tracks Apple's PWA notification rollout.

QR Code Customer Self-Reporting

Every machine gets a unique QR code sticker. A customer scans with their phone camera (no app), lands on a tenant-branded page pre-filled with location and machine ID, taps the issue type, hits submit. **The ticket logs in real time and SMS-alerts the tech in seconds.** No customer ever types a thing.

Mobile-First Design

The interface is designed for a tech's phone first, then scaled up to desktop. Big tap targets, simple flows, no nested menus. The screens techs use most are the ones that load instantly.

Owner Dashboard for the Desktop

Real-time visibility across the entire operation. Open issues per route, repairs completed this month, parts consumed, monthly submission status — all at a glance with drill-down into any data point. Built for the owner who wants the picture in 30 seconds, not 30 clicks.

7 INTENTIONAL SIMPLICITY

Engineering Discipline · What We Don't Use

A startup running 17 microservices on Kubernetes isn't sophisticated — it's drowning. VendGO is deliberately built on a minimal stack so the surface area for failure is small and a single engineer can hold the whole system in their head. Here's what we explicitly chose *not* to use, and why.

No Kubernetes · No Docker · No Microservices

VendGO is a single Next.js application deployed to Vercel. One codebase, one deployment, one database. Less complexity means fewer failure modes, faster debugging, and a system a solo engineer can actually maintain. **Microservices solve organizational problems, not technical ones** — and VendGO doesn't have those organizational problems yet.

No Redis · No Memcached

PostgreSQL handles caching needs at current scale. Adding a separate caching layer is a real cost — another service to monitor, another failure mode, another thing to keep in sync. We'll add Redis the day Postgres can't keep up, and not a day sooner.

No GraphQL

REST endpoints are simpler to debug, simpler to cache, simpler to monitor, and sufficient for everything VendGO needs to do. GraphQL is a tool with real strengths — it just doesn't solve any problem we have.

No Separate API Backend

Next.js API routes serve the frontend directly. One codebase. One deploy. One language. No "frontend team vs. backend team" handoffs because there's no backend team — Brian writes both because they're the same code.

No Custom Authentication

Authentication is the easiest thing to get wrong and the most painful to recover from. VendGO uses Supabase's battle-tested auth instead of rolling its own. Hundreds of millions of users worldwide validate this code path daily. We'd never beat it.

No NoSQL Document Store

VendGO's data is fundamentally relational — techs belong to routes, machines belong to locations, repairs reference issues, parts reference inventory. A document database would force us to either duplicate data everywhere or build joins by hand. Postgres handles this natively and correctly.

8 REFERENCE

The Stack at a Glance

Every tool VendGO uses, organized by layer. Mature, well-documented, and battle-tested by companies orders of magnitude larger than us.

FRONTEND FRAMEWORK Next.js 16	UI LIBRARY React 19	TYPE SYSTEM TypeScript	STYLING Tailwind CSS 4
STATE MANAGEMENT Zustand	FORMS React Hook Form	VALIDATION Zod	ANIMATION Motion
DATABASE PostgreSQL	BACKEND PLATFORM Supabase	HOSTING Vercel	FILE STORAGE Supabase Storage
SMS & VOICE Twilio	TRANSACTIONAL EMAIL Resend	ERROR TRACKING Sentry	PDF GENERATION WeasyPrint

THE BOTTOM LINE

A modern stack, deliberately constrained.

VendGO uses a small number of best-in-class tools, each chosen because they solve a real operator problem. **Every layer of the stack is mature, well-documented, and battle-tested by companies orders of magnitude larger than us.** When we scale, we won't outgrow the architecture — we'll fill it in. That's by design.