

Joseph A. Arrieta Chilberry

CHIEF TECHNOLOGY OFFICER · SOLUTIONS & INFRASTRUCTURE ARCHITECT · AI/LLM INTEGRATION

Quito, Ecuador

☎ (+593) 99-280-7816 | ✉ ja.chilberry@gmail.com | 🌐 chilberry.me | 📱 jac1013 | 📄 joseph-arrieta-chilberry

Summary

CTO and systems architect with 16 years of experience turning ambitious product visions into production-grade systems. Works across solutions architecture and infrastructure/DevOps: AI/LLM integration, AWS cloud architecture (certified), infrastructure as code, and SaaS platforms that scale from MVP to enterprise. Comfortable in regulated domains: architected a HIPAA-aligned healthcare platform end to end as solution architect and signed as the business associate accountable for its PHI safeguards, with earlier delivery into regulated financial communications. Currently CTO at Bixlabs and Münster Mind, leading engineering teams, defining technology strategy, and shipping AI-powered products with governance, security, and measurable ROI. Scaled teams from 5 to a peak of 30+ engineers and built architectures that have run in production for over a decade.

Work Experience

Münster Mind

AI Talent Platform

CHIEF TECHNOLOGY OFFICER

Sep. 2023 - Present

- Technology lead and founding architect of a live, revenue-generating AI-native talent platform: 3 enterprise B2B customers and 200+ end users, currently mid go-to-market execution.
- Defined the full technology strategy and system architecture from the ground up, and own the platform end to end — product architecture, AWS infrastructure, LLM orchestration, security and delivery (platform detail under Selected Projects).
- Leading a 3-engineer team across backend, frontend, and infrastructure with CI/CD automation and security-first delivery.

Bixlabs

Full-time

CHIEF TECHNOLOGY OFFICER

Sep. 2014 - Present

- Defining technology strategy and leading architecture across 12 SaaS products delivered to ~30 enterprise clients, spanning RegTech, job search, retail POS and in-store kiosks, a HIPAA-regulated medical system, and agile team management — personally architecting about half of them.
- Scaled the engineering team from 5 to a peak of 30+ developers, building the hiring processes, mentorship programs and agile culture behind it — mentoring more than 10 developers in the tech lead role and creating and delivering 100+ TechTalks, running bi-weekly from 2018 to 2023 and again since 2025.
- Owned technical hiring since 2016: ~500 interviews conducted and ~60 engineers hired across client delivery teams.
- Won client engagements as the technical closer: Theta Lake through a first-contact technical interview with their CTO, FindMyProfession and ThePeakBeyond through technical interviews with their CEO and CTO, Context Driven through a full discovery and planning process with their CEO, and Dafne by building the solution plan up front and presenting it with the execution approach across two of the three sales calls.
- Both accounts are still running years after I won them: Theta Lake with 4-5 engineers on it across 9 years, and FindMyProfession across 3 years with 5 engineers through its phases, where I was also tech lead, solutions architect and infrastructure owner.
- Turned the Context Driven discovery into the artifacts the engagement was sold on — MVP scope, release roadmap, risk register and open assumptions — after which the CEO retained a few hours of my time each month on their leadership team, citing how well I had grasped their domain; the client later returned for a second product, whose planning phase I ran before handing the documentation to the implementing team.
- Recovered 2 clients that arrived from technically failed prior engagements — ThePeakBeyond, where I ran the v1 rescue and then sold and scoped the v2 rebuild, and FindMyProfession, where I replaced the unreliable inherited platform as tech lead and solutions architect (see Selected Projects for both).
- Oversaw technical delivery across multiple concurrent client projects and ran standalone consulting engagements alongside the build work — architecture and infrastructure decisions for the delivery teams, and a technical audit of Acapedia's production system, infrastructure configuration and load behaviour (see Selected Projects).
- Expanded the role into infrastructure architecture from 2020, owning infrastructure end to end on the engagements that followed — introducing Terraform into existing client work before standardizing on it for greenfield builds.
- Directed operations initiatives for 3 years as COO, implementing new tools and training the team.

Kyoto Stores

Puerto Ordaz, Venezuela

SOFTWARE ENGINEER

2012 - 2014

- Built services to expose store operations that were locked inside a legacy system running on outdated technology.
- Developed the Android application deployed to 500+ tablets across every store in the city, consuming those services with limited on-device storage for offline functionality.
- Automated deployment to the client's on-premise servers with Jenkins.
- Technologies: Java, Spring, Hibernate, Maven, Tomcat, JSP, REST, JasperReports, MySQL, Android SDK, SQLite, jQuery, AJAX, HTML/CSS, Jenkins, Git.

- Replaced a manual XML review process with an automated validator handling thousands of documents weekly, built on a rule engine written from scratch where hard rules were updated as they were issued and soft rules resolved dynamically.
- Technologies: Java, Spring, Hibernate, JAXB, Maven, Oracle, Git.

Selected Projects

Münster Mind is my current CTO role; the remainder are client engagements delivered through Bixlabs.

Münster Mind

AI Talent Platform

CHIEF TECHNOLOGY OFFICER | PLATFORM ARCHITECTURE AND INFRASTRUCTURE

Sep. 2023 - Present

- Architected the platform that ingests unstructured hiring data (CVs, interviews, assessments) and turns subjective hiring into data-driven strategy, processing ~2,000 CVs a month across 3-6 concurrent hiring processes.
- Built the Smart Screening engine that cut client time-to-hire from 4-5 weeks to 3-5 days — an ~85% reduction — by ranking candidate shortlists out of ~100-applicant pools per opening.
- Built the Predictive Cultural Fit and Human Development modules, covering interview analysis and goal/task generation tied to each client's evaluation cycle.
- Designed a provider-agnostic LLM layer spanning OpenAI, Anthropic, Gemini, Mistral and GLM that routes each task to the best-suited provider, with a DynamoDB-backed global circuit breaker that trips per provider and reroutes traffic dynamically when one degrades under load — keeping the platform running through upstream outages.
- Evaluate LLM output against golden datasets to measure accuracy before prompt or model changes ship.
- Made breaker behaviour observable without per-function wiring: every open transition emits a CloudWatch Embedded Metric Format record dimensioned by provider and environment, so any LLM-consuming function feeds provider health metrics and alarms automatically as the fleet grows.
- Run LLM work asynchronously off SQS into Lambda and Step Functions workflows, so multi-step prompt chains complete in the background without blocking the request path.
- Track LLM token spend to the penny inside the platform, attributed per customer and per workflow, making unit economics measurable rather than estimated.
- Designed and built the platform's entire AWS infrastructure: API Gateway fronting Lambda as primary compute — 305 functions across 283 REST endpoints — alongside autoscaled ECS services and EC2, in private subnets with NAT egress, VPC endpoints, VPC peering, and an internal network load balancer.
- Solved Lambda connection exhaustion against PostgreSQL with two-layer pooling, PgBouncer in front of RDS Proxy, sized per environment against the database connection budget: production runs PgBouncer as autoscaled ECS tasks behind a network load balancer, non-production on a lower-cost EC2 container shape, with a private Route53 record keeping the database host stable across instance replacement.
- Built provisioned concurrency into the shared Lambda module as an opt-in capability — versioned aliases, Application Auto Scaling with target tracking, a business-hours baseline and a scheduled scale-to-zero outside working hours — cutting warm-capacity cost on latency-sensitive paths without hand-tuning individual functions.
- Built the whole platform from four reusable Terraform modules: 623 module instances against 2 hand-written resources, with workspace-driven staging and production environments and encrypted remote state in S3.
- Built a GitHub Actions pipeline that runs change detection across the Lambda fleet to issue targeted Terraform applies, holding deploy times down as the function count grew, and gates database migrations in CI so a bad migration fails the build instead of production.
- Built the event-driven surface: 6 Step Functions state machines, 25 EventBridge scheduled rules, SQS queues with dead-letter and redrive handling, and WebSocket APIs over API Gateway and AppSync for real-time updates.
- Built the media pipeline behind interview analysis: Amazon Transcribe jobs triggered from S3 uploads, transcription processing through Step Functions, and speech synthesis for generated assessment content.
- Built an inbound email pipeline on SES — messages delivered to S3 and parsed by Lambda producers into async workflows — plus outbound delivery-event tracking, making email a first-class product input rather than a notification channel.
- Implemented authentication on Cognito across three user pools with a custom auth challenge flow supporting passwordless magic-link sign-in.
- Instrumented the platform with 56 CloudWatch metric alarms including composite alarms, 3 operational dashboards, SNS-routed alerting through a dedicated notifier function, Sentry across the Lambda fleet, targeted tracing for performance work, and PostHog for session replay and product analytics.
- Secured customer data with least-privilege IAM, Secrets Manager for all provider and service credentials, and encryption in transit and at rest, with an export process so customers can take their data with them.
- Documented the platform for operators with an infrastructure architecture reference, deployment and operations guide, troubleshooting and maintenance guide, and a Lambda function reference.
- Technologies: NodeJS, NestJS, ReactJS, TypeScript, PostgreSQL, PgBouncer, DynamoDB, Terraform, GitHub Actions, Docker, AWS (Lambda, Step Functions, SQS, SNS, EventBridge, API Gateway REST and WebSocket, AppSync, ECS, EC2, ECR, RDS, RDS Proxy, S3, Cognito, SES, Transcribe, Amplify, VPC, VPC Endpoints, NAT Gateway, Network Load Balancer, Route53, Secrets Manager, IAM, CloudWatch, Application Auto Scaling), Sentry, PostHog, multi-provider LLM orchestration (OpenAI, Anthropic, Gemini, Mistral, GLM) with circuit-breaker failover, semantic analysis.

Dafne

Bixlabs client engagement

SOLUTION ARCHITECT

Jun. 2026 - Present

- Solution architect for a Spanish-first, HIPAA-aligned tele-therapy platform connecting clients with US state-licensed therapists — won the engagement commercially, owned the solution plan end to end, and carried it from zero to production readiness in three months.
- Made the case for replacing a Google Apps Script v1 with a real platform, then designed the replacement and scaffolded it: React 19 SPAs, a NestJS API and a NestJS BullMQ worker over Prisma and PostgreSQL, in a pnpm and Turborepo monorepo.
- Planned the therapy session around the patient rather than the software — the therapist works from the browser while the platform places a real phone call to the patient's own number, so attending a session needs no app, login or link.
- Own the platform's AWS infrastructure end to end, staging and production: designed it and built it in CDK, decomposed into network, data, compute, delivery and observability stacks with a test suite per stack, running the API and worker on ECS against RDS PostgreSQL and MemoryDB.
- Stood the staging environment up from nothing and moved the platform onto its own AWS account, documenting both so the team could repeat them.
- Signatory on the HIPAA Business Associate Agreement covering the engagement, which puts contractual accountability for the safeguards protecting PHI on me rather than only the work of implementing them.
- Built the platform's transport security end to end — verified TLS from the Prisma client through to RDS under `force_ssl` using a bundled RDS global CA, rather than leaving the connection encrypted but unverified.
- Provisioned the platform's secret material through CDK — the PHI encryption key, a MemoryDB auth token bound to a real ACL user, and third-party credentials — alongside KMS, CloudTrail, GuardDuty and Inspector for the audit and threat-detection controls the compliance posture rests on.
- Built the delivery pipeline on GitHub OIDC with an ECS migration runner and auto-deploy from main, so deployments authenticate without any long-lived AWS credentials.
- Diagnosed and fixed a Redis CROSSSLOT failure on the clustered webhook queue by hash-tagging it, then encoded the queue-prefix convention into the shared types package so the same class of bug could not recur.
- Kept running cost deliberate rather than incidental: removed 5 unused VPC interface endpoints and throttled worker error logging once it became a spend driver.
- Wrote the operational runbooks the team deploys from — infrastructure deployment and local run-and-verify.
- Working with a three-person team: two engineers plus myself as solution architect and owner of the high-level solution.
- Technologies: TypeScript, React 19, Vite, NestJS, Prisma, PostgreSQL, Redis, BullMQ, AWS CDK, AWS (ECS, RDS, MemoryDB, CloudFront, ALB, ACM, S3, ECR, VPC, Secrets Manager, KMS, CloudTrail, GuardDuty, Inspector, SES, IAM), GitHub Actions with OIDC, Clerk, DocuSign, Telnyx (Voice/PSTN and Video), Stripe, Google Calendar, Pino, Turborepo, pnpm workspaces, Vitest, Docker.

FindMyProfession

Bixlabs client engagement

TECH LEAD

Apr. 2022 - Present

- Tech lead for the ground-up rebuild of the platform behind a reverse-recruiting managed service — 500 daily users and 5,000 job entries ingested per day — replacing a system that was unreliable and missing much of what the business actually did in operations.
- Delivered the new platform in one year and executed a “live” migration with 0 downtime: all existing data moved across and clients switched by simply logging into the new portal, with nothing to do on their side; the outgoing system was decommissioned by agreement with the product owner and the new one has run in production for 3 years since.
- Eliminated a pattern of almost weekly outages inherited from the old system: the new platform ran its first full year with no customer-facing outages, and measured 99.9% availability over that year against a 99.5% design target.
- Replaced hand-managed EC2 hosts with ECS on Fargate running the backend and two asynchronous workers, CPU-based auto scaling, behind an application load balancer with the services in private subnets and NAT egress, S3 + CloudFront for static delivery, and WAF at the edge.
- Owned all infrastructure in Terraform — reusable modules, S3 remote state, and fully separated staging and production environments — taking production configuration issues to near zero and making every environment reproducible.
- Built the GitLab CI pipeline from scratch: blue/green deployments through ECS with automated rollback, replacing a manual release process and saving 2-3 hours daily.
- Made database migrations part of CI, run first against an ephemeral database so a bad migration breaks the build instead of production; RDS ran Multi-AZ with automated backups and point-in-time recovery.
- Instrumented the platform with CloudWatch logs and alarms alerting through PagerDuty, adding tracing instrumentation where performance work required it.
- Enforced least-privilege IAM roles, Secrets Manager for credentials, and ACM-issued TLS across all endpoints.
- Built job and contact search on OpenSearch, and the PayPal and Braintree checkout flows.
- Led the technical team across the life of the project — 5 engineers over 3 years, a maximum of 3 concurrently including myself as tech lead — delivering into a business of ~50 people across operations and other functions.
- Technologies: NodeJS, VueJS, ReactJS, CypressJS, StorybookJS, PayPal, Braintree, OpenSearch, Terraform, GitLab CI, Docker, PagerDuty, AWS (ECS Fargate, RDS Multi-AZ, S3, Lambda, API Gateway, ALB, VPC, NAT Gateway, CloudFront, WAF, ACM, Secrets Manager, IAM, CloudWatch).

ThePeakBeyond

Bixlabs client engagement

TECHNICAL CONSULTANT

2024 - Present

- Brought in to rescue a failing v1 of a cannabis-retail platform — POS integrations plus in-store kiosks that pair purchasing with product education — serving 50 stores and 500-1,000 users.
- Spent a year stabilizing v1 before concluding the rescue was not viable and saying so: the codebase quality was too low and the system too complex for a realistic recovery plan, with new issues surfacing faster than fixes landed.
- Built the case for a v2 rebuild instead, argued from v1's measured failures rather than preference: a \$4,000 monthly AWS bill for a workload I put under \$500, and a 40-minute POS sync — the single most important job in the system — where I committed to 5 minutes for a full sync and sub-minute deltas; v1 had no delta sync at all despite every POS vendor exposing the endpoints for it.
- Made the case commercially as well as technically, weighing the customer support load coming off the v1 UI and the deprecated surface carried by both frontend and backend, and factoring in that AI-assisted code generation had made a clean rebuild materially cheaper in 2026 than the same decision a year earlier.
- Owned the architecture and infrastructure design for v2 and delivered it as documentation to the implementing tech lead — a 5-month build with a team of 3 — rather than implementing it myself.
- The infrastructure now running against that design costs \$600 a month provisioned for 200 stores, against v1's \$4,000 for 50, and completes a full sync for the largest and highest-volume store in under 5 minutes with delta syncs at roughly 30 seconds.
- The saving came from the platform decision as much as the code: v2 replaced v1's AWS footprint with managed services — Supabase for database, auth, realtime and edge compute, a Railway-hosted Node worker for long-running catalog syncs, and Algolia for search.
- The platform integrates 5 POS vendors — Dutchie, Treez, Covasoft, Flowhub and Blaze — behind a common adapter interface, with circuit breakers, dead-letter queues and retry ladders around the vendor calls.
- First pilot store is migrating to v2 in production now, replacing a full store's v1 workload.
- Technologies: TypeScript, Next.js, React, tRPC, TanStack Query, Zustand, Tailwind, Radix UI, Storybook, Supabase (PostgreSQL, Edge Functions on Deno, Auth, Realtime, Storage, Vault), pgmq, Node.js workers on Railway, Algolia, Sentry, Turborepo, pnpm workspaces, GitHub Actions, AWS Route 53, Vitest, Playwright, NFC and serial-port kiosk hardware integration, POS APIs (Dutchie, Treez, Covasoft, Flowhub, Blaze).

Context Driven (Context.ai)

Bixlabs client engagement

TECH LEAD

Jan. 2023 - Aug. 2023

- Tech lead on an AI-assisted canvas where users built concept boards and anchored them to philosophical concepts; shipped to production and used by the client's internal users for two months before the client wound the product down.
- Provided recommendations on product direction and possible pivots, resulting in a necessary change of direction.
- Designed and implemented the integration with OpenAI's GPT-3.5 and made it model-agnostic to test functionality and performance across models, improving experimentation times from multiple weeks to a couple of days.
- Ran the LLM work asynchronously on a dedicated ECS worker consuming from SQS, because the multi-prompt workflows took minutes to complete on 2023-era models — keeping the request path responsive while long jobs ran to completion in the background.
- Evaluated and coordinated the tech stack for the system.
- Architected the whole infrastructure in Terraform — reusable modules, S3 remote state, and separate staging and production environments — with services in private subnets behind a NAT gateway, a load balancer and API Gateway at the edge, and all credentials in Secrets Manager with temporary-token access.
- Automated build and deployment with GitHub Actions.
- Technologies: NodeJS, VueJS, StorybookJS, CypressJS, MongoDB, OpenAI GPT-3.5, Terraform, GitHub Actions, Docker, AWS (ECS, SQS, API Gateway, VPC, NAT Gateway, Load Balancer, Secrets Manager, S3, CloudFront, CloudWatch).

Academia

Bixlabs client engagement

TECHNICAL CONSULTANT

Feb. 2023 - Apr. 2023

- Two-month part-time engagement (~0.25 FTE, 60-80 hours total) auditing a production system and its infrastructure, delivering documented improvements and implementing part of the configuration work directly.
- Traced degraded performance to a PHP-FPM/FastCGI misconfiguration in their Elastic Beanstalk environment; after tuning it the same workload ran on 2 containers instead of 3 — a third less compute for the same service.
- Built K6 load tests that emulated real user journeys across ~50 endpoints and complete system workflows rather than hammering a single endpoint, establishing that the system began failing at roughly 2,000 concurrent users and sizing the running container count from that data.
- Identified slow endpoints and traced them to low-level implementation problems including N+1 query patterns.
- Diagnosed a long-standing SSL problem as a redirect misconfiguration — TLS worked or failed depending on which URL a user landed on, while the certificates themselves were fine.
- Technologies: K6, PHP-FPM, AWS Elastic Beanstalk, Load Balancers, RDS, Route53, CloudWatch.

Theta Lake

Bixlabs client engagement

FULL STACK DEVELOPER

Aug. 2017 - Jan. 2020

- Built integrations for a digital communications governance and archiving platform — capture, archiving and supervision of regulated communications for financial services, healthcare and government customers under regimes such as MiFID II.
- Engineered more than 20 integrations across chat, voice and video platforms, including Zoom, Microsoft Teams, Slack, Google Meet, BlueJeans and RingCentral.
- Each integration meant the whole surface rather than a single API call: OAuth authorization flows, webhooks where the vendor offered them, bulk report extraction, incremental change capture and historical backfill, and rate-limit handling with retries.
- Worked directly with vendor engineering teams at Microsoft, BlueJeans and RingCentral to resolve API behaviour that contradicted their own documentation — Microsoft Teams was the hardest, new enough at the time that the API was still immature.
- Those integrations fed the Integrator, which pulled from every source platform and handed off to the Ingester, which normalized heterogeneous feeds into the internal data stores — both on Golang and Cassandra, chosen for the ingestion volume the product was built to absorb.
- Owned the frontend end to end because the CTO's expertise sat elsewhere: selected the stack, defined how the code was organized, and made snapshot, integration and end-to-end testing part of the development process.
- Those decisions have held for 9 years — the frontend is still the same project, never rewritten.
- Trained the frontend team on Angular quality practices, keeping the codebase in good shape.
- Coached the team on agile practices and their do's and don'ts.
- Technologies: Golang, Angular, StorybookJS, Selenium, MySQL, Cassandra, OAuth 2.0, webhooks, Zoom API, Slack API, MS Teams API, Google Meet API, BlueJeans API, RingCentral API.

Surgemaker

Bixlabs client engagement

DEVELOPER, THEN TECH LEAD

2014 - 2018

- Built the platform Agile coaches used to grow team maturity — capturing per-team delivery metrics, giving coaches visibility into how a team's practices evolved over time, and surfacing recommendations they could take back into their engagements.
- First engagement at Bixlabs and a four-year relationship: sole developer for the first two years, then tech lead for the following two.
- Owned the product end to end — frontend, backend, and the scheduled background tooling that computed metrics outside the request path; the client owned and operated the infrastructure.
- Led a team of 3 as tech lead: 2 developers and a QA engineer.
- Held the client relationship directly across all four years, owning technical decisions and expectations with the client rather than working through an account layer.
- Technologies: JavaScript, BackboneJS, NodeJS, PostgreSQL, AWS, Jenkins.

Education

UNADE American University of Europe

Online, Mexico

M.S. IN SOFTWARE ENGINEERING

Sep. 2019 - Sep. 2021

Universidad Rafael Belloso Chacín

Maracaibo, Venezuela

B.S. IN COMPUTER ENGINEERING

Sep. 2007 - Dec. 2012

Certifications & Courses

2023	AWS Certified Cloud Practitioner , Amazon Web Services	Online
2023	AWS Certified Solutions Architect - Associate , Amazon Web Services	Online
In progress	AWS Certified Solutions Architect - Professional , Amazon Web Services	Online
2016 - 2017	Data Structures & Algorithm Toolbox , University of California via Coursera	Online

Skills

Programming Languages	JavaScript/TypeScript, Golang, Python, Java, Terraform (HCL), Clojure, Haskell, SQL
AI & LLM	Multi-provider LLM integration (OpenAI, Anthropic, Gemini, Mistral, GLM), provider-agnostic routing and failover, semantic analysis / NLP, agentic AI workflows
Cloud & Infrastructure	AWS (Lambda, Step Functions, SQS, SNS, EventBridge, ECS/Fargate, EC2, ECR, Elastic Beanstalk, RDS, RDS Proxy, S3, API Gateway REST and WebSocket, AppSync, Cognito, SES, Transcribe, ALB, NLB, VPC, VPC Endpoints, NAT Gateway, CloudFront, Amplify, Route53, WAF, ACM, Secrets Manager, IAM, CloudWatch, Application Auto Scaling, KMS, CloudTrail, GuardDuty, Inspector, MemoryDB), Terraform, AWS CDK, Docker, Docker Compose, Nginx, Linux, Railway, managed-platform architecture (Supabase, Railway)
Databases & Storage	PostgreSQL, MySQL, Oracle, MongoDB, DynamoDB, DocumentDB, Elasticsearch/OpenSearch, Cassandra, Redis, AWS MemoryDB, Supabase, pgmq, BullMQ, PgBouncer / RDS Proxy connection pooling
Frameworks & Tools	NestJS, Express, Koa, React, Next.js, Vite, Vue, Angular, BackboneJS, Prisma, Tailwind, Sass, Cypress, Jest, Mocha, Storybook, Selenium, Puppeteer, Playwright, Vitest, tRPC, TanStack Query, Zustand, Radix UI, Turborepo, pnpm workspaces, Algolia, Stripe, PayPal, Braintree, Git
CI/CD & Observability	GitHub Actions, GitLab CI, Jenkins, blue/green deployments, automated rollback, CI-gated database migrations, CloudWatch logs, metric and composite alarms, dashboards and Embedded Metric Format, Sentry, PagerDuty, K6 load testing, PostHog, structured logging with Pino, GitHub OIDC keyless deploys
Architecture & Practices	System design, microservices, serverless, event-driven and asynchronous architecture, domain-driven design (DDD), infrastructure as code, CI/CD pipeline design, high availability (Multi-AZ, point-in-time recovery), circuit breakers and graceful degradation, autoscaling, zero-downtime migrations, least-privilege IAM, HIPAA-aligned system design (PHI handling boundaries, audit and threat-detection controls), Agile/Scrum
API Integrations	OAuth 2.0 and webhook-based platform integration, Slack, MS Teams, Zoom, RingCentral, Google Meet, Google Calendar, Outlook, Salesforce, MuleSoft, OpenAI, Anthropic Claude, BlueJeans, Clerk, DocuSign, Telnyx (Voice/PSTN and Video), cannabis-retail POS APIs (Dutchie, Treez, Covasoft, Flowhub, Blaze)

Languages

Spanish	Native
English	Advanced