Technical Implementation Overview

This document outlines how we'll build, host, and manage 30+ SSR client websites per month. The system integrates Salesforce, lead notifications via email/SMS, and WildJar for call tracking. We'll start with a manual process and transition to automation over time.

---

1. Core Stack & Tooling

Purpose	Tool / Platform	Notes
Frontend Framework	Next.js (SSR mode)	Great for SEO and dynamic content
API Layer	Node.js + Express or NestJS	Handles Salesforce sync, webhooks, and form routing
Hosting (Web)	Self-hosted via Docker + NGINX	Cheaper and scalable vs. Vercel; SSR-friendly
Reverse Proxy	NGINX or Traefik	Manages domain-based routing + SSL
Hosting (API)	AWS EC2 or Hetzner VPS	Runs backend services and job runners
Domains	Namecheap / Cloudflare API	For domain management and automation
Email Delivery	Amazon SES	Used for transactional lead emails
SMS Notifications	Amazon SNS	Fast, cost-effective SMS service for UK delivery
CI/CD Pipeline	GitHub Actions or custom shell scripts	Automates build, deployment, and DNS updates
Secrets Management	AWS SSM / Doppler / .env	For storing keys and environment variables
Call Tracking	WildJar	Handles call routing, recording, and analytics
Logs & Monitoring	CloudWatch / Grafana / Sentry	For observability and error tracking

---

2. Site Build Automation Flow (Post-MVP)

Once we've completed the core infrastructure, the flow to generate and deploy new client websites will look like this:

1. **Salesforce Trigger**
    - Our sales team adds a new client to Salesforce.
    - Client info is captured in a structured format (custom object).
2. **Backend Listener**
    - Our API monitors for new Salesforce records.
    - It pulls and stores client data to our internal DB for processing.
3. **Site Generator**
    - A CLI or automated worker uses a base Next.js template.

- It injects client-specific content: logos, contact info, colors, text blocks.
- A new build directory or Git branch is created per site.

4. **CI/CD Pipeline**
    - Triggers a Docker build of the Next.js SSR site.
    - Pushes the container to our server or Kubernetes cluster.
    - NGINX routing is updated to serve the new domain.
5. **Domain Setup**
    - A new domain is registered or configured via Cloudflare API.
    - DNS records are pointed to our server.
    - Let’s Encrypt or Traefik issues SSL certificates automatically.
6. **WildJar Integration**
    - Through WildJar’s API, we assign a virtual number for the client.
    - The number is routed to the client’s real phone and recorded.
    - Tracking metadata is stored in our DB.

---

3. Lead Capture: Email + SMS

All client sites will send form fills to our central API.

Email (via Amazon SES)

- Our API receives the submission, formats the content, and sends an email to the client.
- We’ll log delivery status and retry if necessary.

SMS (via Amazon SNS)

- Each form triggers an instant SMS to the client:

    > “New lead from [Website]: John Smith, 07700 900123 — ‘Looking for a quote on...’”

- We'll use templated messages for professionalism and speed.

---

4. Call Tracking with WildJar

Each site will be assigned its own WildJar number.

1. When someone calls:
    - The call is forwarded to the client.
    - A recording and metadata (duration, caller ID) is stored.
2. WildJar’s webhook sends us data for analytics and optional transcription.

3. We store this info securely in our database for optional AI analysis and future insights.

---

5. AI Feedback (Premium Tier)

For clients who opt in, we'll offer intelligent feedback reports on call performance.

- Each week, we process transcribed calls through GPT-4 Turbo.
- We'll identify missed opportunities, tone indicators, objections, and sales wins.
- Reports are sent to the client showing:
    - Lead quality
    - Conversion cues
    - Suggested improvements

We'll handle this asynchronously with background jobs and scheduled tasks.

---

6. Manual Mode (Months 1–6)

Before automation is ready, we'll manually build and deploy each site.

Steps:

- Clone the template repo and adjust client content
- Run `next build && next start` locally or in Docker
- Update NGINX with a new domain block
- Deploy manually and set DNS via Cloudflare dashboard
- Use Certbot to issue an SSL cert
- Setup WildJar via dashboard or basic script

This helps us get early traction while gradually building out automation.

---

7. Monitoring & Error Handling

Layer	Tool	Purpose
Frontend	Sentry	Tracks rendering and runtime issues
Backend	CloudWatch or Grafana	Logs and API errors
Hosting	Prometheus / Node Exporter	System health and uptime
Messaging	SNS & SES Logs	Monitors delivery success
Call Data	WildJar Analytics	Tracks call volume and lead source quality

Phase 1: Foundations & Manual Launch (Weeks 1–4)

Goal: Launch the first 10 client sites manually + prepare base infra

Task	Est. Hours	Notes
Learn DevOps basics (hosting, domains, CI/CD, Docker)	20 hrs	Focus on EC2 or VPS setup, NGINX, and SSL
Build base Next.js template	8 hrs	Generic SSR-ready template with content slots
Form + API handler	4 hrs	Handle form submissions + email/SMS
Setup AWS SES, SNS	4 hrs	Verified sender domains, SMS configs
Manual deploy for 3–5 sites	8 hrs	Practice DNS, SSL certs, deploy to VPS
Setup GitHub Actions baseline	4 hrs	For manual deploy triggers or branch builds
WildJar integration (manual setup)	2 hrs	Via dashboard or script
Testing & debugging	5 hrs	Ensure stable form → alert pipeline
Total: ~55 hrs → ~3 weeks at 20 hrs/week		

Phase 2: Semi-Automation & Scale Support (Weeks 5–8)

Goal: Streamline new site generation via templated CLI + automated deploy

Task	Est. Hours	Notes
CLI tool to scaffold client site from JSON	10 hrs	Generates new directories w/ client data
Hook up basic Salesforce sync	6 hrs	Use polling or push-style integration
Auto-build + deploy (GitHub Actions or Docker build)	10 hrs	CI pipeline handles build/push for each site
Cloudflare DNS API integration	6 hrs	Automates domain record setup
Auto-SSL (Certbot or Traefik)	5 hrs	One-click SSL per site
Automate WildJar setup (optional MVP)	4 hrs	Call number assignment via API
Testing & iteration	5 hrs	Validate full flow end-to-end
Total: ~46 hrs → ~2.5 weeks		

Phase 3: Stability, Monitoring, and Light AI Prep (Weeks 9–12)

Goal: Add observability, harden infra, and prep for AI feedback services

Task	Est. Hours	Notes
Monitoring (Sentry, logs, alerts)	6 hrs	Backend + frontend error capture
Form submission logs & DB	4 hrs	Store leads before forwarding
Build dashboard or export report (optional)	6 hrs	MVP reporting for clients
Transcription test pipeline	6 hrs	Connect WildJar output to AssemblyAI
Script AI feedback summaries (GPT-4 Turbo)	8 hrs	Proof of concept for premium tier
Call storage setup (S3 or DB)	3 hrs	Transcripts + call metadata
Total: ~33 hrs → ~1.5–2 weeks		

Summary Timeline

Phase	Focus	Weeks (20 hrs/week)	Outcome
1	Infra & manual launch	1–4 (~3 weeks)	First sites live, form → email/SMS working
2	Semi-automation	5–8 (~2.5 weeks)	Sites generate from template + deploy automatically
3	Observability + AI prep	9–12 (~2 weeks)	Monitoring + prototype feedback engine

Full MVP Infra Timeline: ~8–9 weeks (~160–180 hours)
With breathing room, we'd say **12 weeks** is ideal to stabilize, polish, and begin onboarding full automation at scale.

Ongoing Time Commitment (Post-MVP)

Once the system is live:

Role	Task	Hours/week
Infrastructure Dev	Bug fixes, new features, optimizations	3–5 hrs
Site updates (until full automation)	Minor styling/content changes	2–3 hrs
Client support + onboarding	Questions, DNS issues	1–2 hrs

---

14. Requirements for Technical Delivery & Collaboration

Immediate Technical Priorities

1. **Autobuild MVP Timeframe**
    - We need to define a *clear deadline* for when the MVP of the site autobuild system must be in place.
    - Any manual site builds done before this point will delay the automation pipeline.
    - Let's align this deadline with our first major batch of clients and sales onboarding.
2. **Salesforce Access**
    - I need access to a Salesforce developer environment or sandbox ASAP.
    - This is critical for starting work on how form fills trigger site generation and lead notifications.
    - Ideally, we configure a test workflow with real example data.
3. **Company Email Account**
    - Please provide me with a company email address (e.g., `dev@prospr.co.uk`) to register SaaS and infrastructure accounts.
    - This ensures clean separation from my personal email and makes future handover easier.

4. **Image Generation Tool**
    - If we’re planning to generate most or all site imagery using AI, I need access now.
    - Early integration into the autobuild pipeline is essential to avoid rework later.

Strategic & Planning Items

5. **Business Plan Review**
    - I’d like to review the full business plan once it’s ready.
    - Understanding go-to-market, pricing tiers, and long-term ops will help align our automation and hosting decisions.
6. **LB Involvement**
    - Can we clarify what role or involvement LB will have (if any)?
    - Just want to make sure all team dynamics and resource allocations are clear.
7. **LED Exit Strategy**
    - What’s the timeline and plan for wrapping up LED?
    - If it’s happening soon, I’d like to align that with some time off from my day job to support a focused launch.
    - I currently have 10 days’ annual leave available and would also like to reserve some time for Christmas.