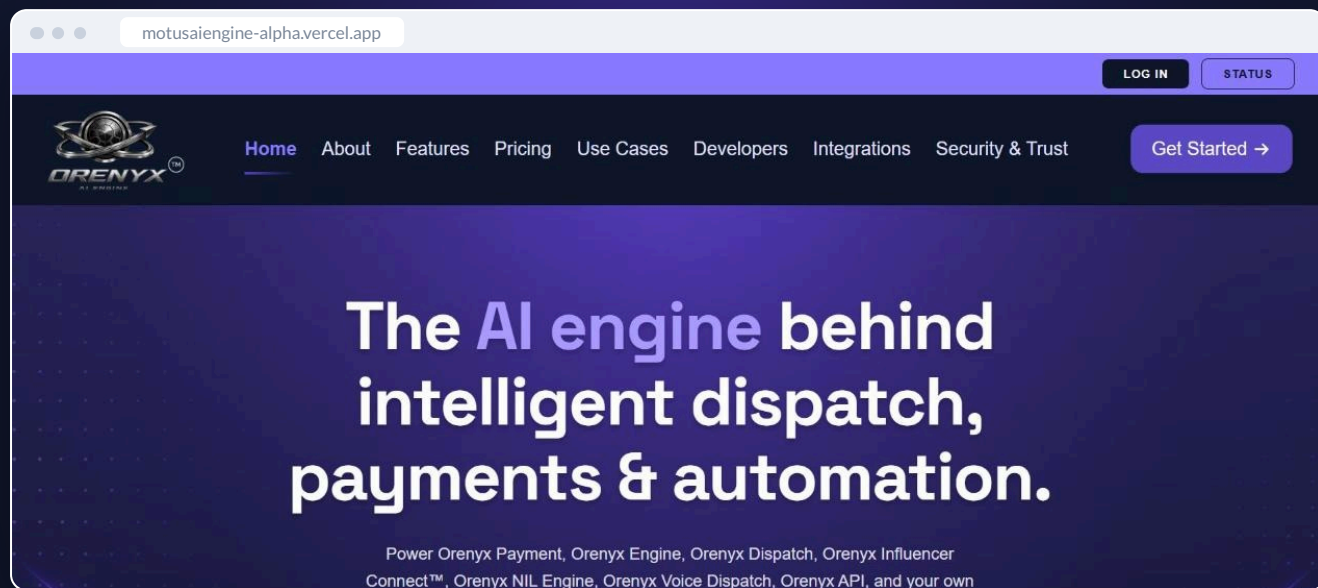




NEXT.JS 15 · REACT 19 · AI PLATFORM

Orenyx AI Engine

The website for the AI layer behind the whole Orenyx family: one engine for dispatch, payments and bots, now offered to other businesses as a platform.

CLIENT

Orenyx Labs LLC

MARKET

United States

INDUSTRY

AI infrastructure

STACK

Next.js 15, Tailwind v4

YEAR

2026

AT A GLANCE

Selling an engine you can't see

Orenyx AI Engine is the shared decision layer behind the Orenyx products: payments, dispatch, voice dispatch, Influencer Connect, NIL Engine and the API. It is also sold on its own to developers and enterprise teams.

That makes the website hard to get right. There's no single screen to show off, the buyer is technical, and the product sits under seven other brands. The site had to make the engine feel real, explain usage-based pricing, and give developers enough to start building.

27+

routes in the Next.js
App Router build

7

Orenyx products wired
into the engine diagram

3

pricing plans, with a
usage cost estimator

10

sections in the
developer docs

0

placeholders allowed
through to launch

The challenge

- An invisible product. The engine has no UI of its own to screenshot.
- A long, detailed builder packet with copy, sitemap, branding, pricing and a pre-launch checklist, and a rule to invent nothing.
- The brand changed twice during the build, and the company renamed from Motus to Orenyx.
- Animated, high-end visuals that still had to load fast and render properly in a server-rendered app.

What we delivered

- The full site in Next.js 15, React 19, Tailwind CSS v4 and TypeScript, deployed on Vercel.
- A live engine diagram and a set of custom animated components that show what the engine does.
- Pricing with plan comparison and a cost estimator, and API docs for developers.
- A launch gate that refuses to ship while any placeholder copy is left on the site.

UI design

Brand system

Next.js 15

React 19

Tailwind v4

GSAP

SVG and canvas animation

Developer docs

Vercel

BRIEF AND PROCESS

Approve it first, then build it in

The client's builder packet set out the sitemap, the copy, the pricing and a pre-launch checklist. The visual direction took a few rounds to settle, so we set up a way of working that kept the codebase clean while the design moved.

How the look settled



Round 1

A first 18-page static build in deep teal with an electric lime accent.



Round 2

A light theme from a client palette, with a full brand guide and contrast checks.



Final

Dark navy and violet, taken straight from the client's Figma, built in Next.js.

Approve, then implement

Animation is hard to judge from a description. So for the key pieces we built a standalone HTML tuner with live dials for speed, glow and timing. The client and our team agreed the feel there, then copied the final settings into the Next.js component.

Flagging, not guessing

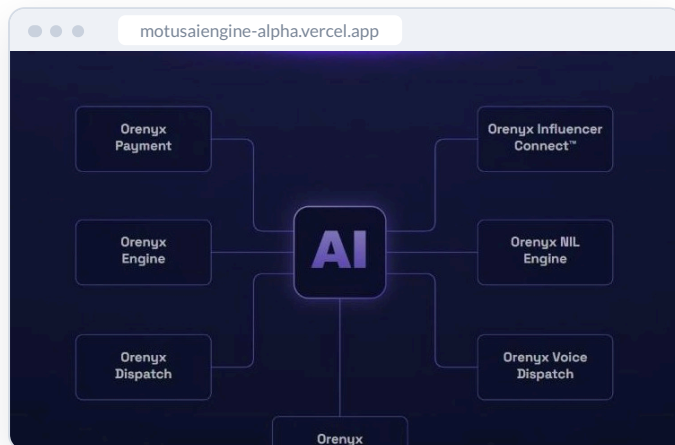
Where the packet had a problem, we said so. White text on the lighter violet failed WCAG AA contrast, so it wasn't used for body text. With the placeholder prices, the Growth plan only became cheaper than Starter above about a million events a month, and we raised that before it reached the pricing page.

No invented copy, no stock images, and every open item shown in magenta until the client filled it.

SHOWING THE ENGINE

One chip, seven products, live traffic

The engine has no screen of its own, so we drew it. Every Orenyx product sits in a box, wired to a central AI chip by circuit-board traces, with pulses travelling in along each line.



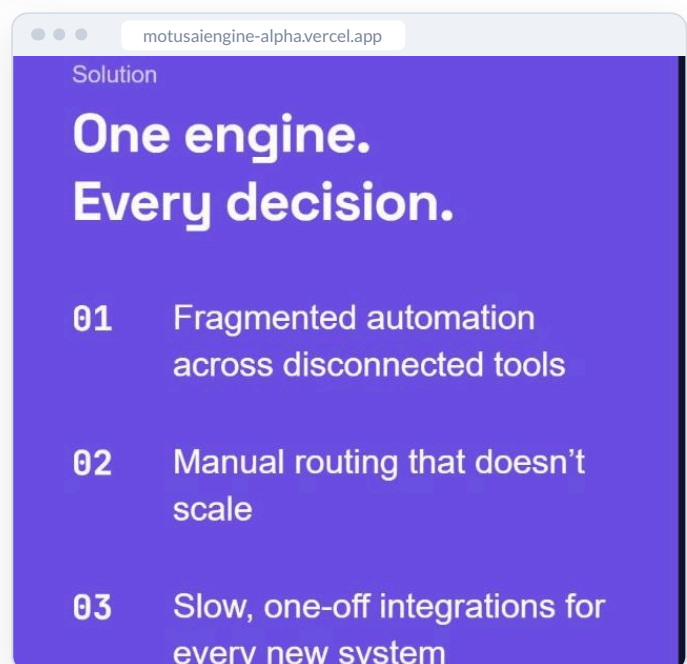
Built to render on the server

- Pure SVG and CSS. No canvas and no JavaScript, so it is in the first HTML the browser gets.
- Product names are real, selectable text, readable by search engines and screen readers.
- A seeded random generator keeps the layout identical on server and browser, so there's no flicker when the page hydrates.
- Motion turns off for visitors who ask their system for reduced motion.

The problem, told on scroll

The "One engine. Every decision." section lists the three problems the engine solves. Each line reveals as the visitor scrolls, driven by GSAP ScrollTrigger.

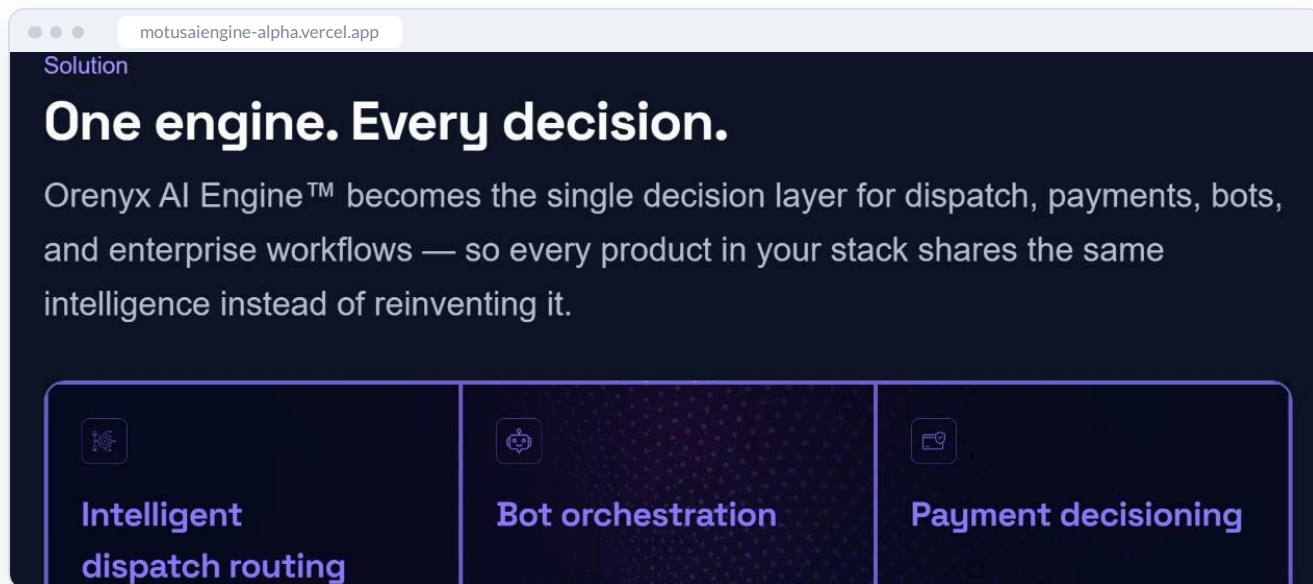
The text is rendered visible on the server first, so the content is always there for search engines and for anyone with JavaScript turned off. The animation only adds to it.



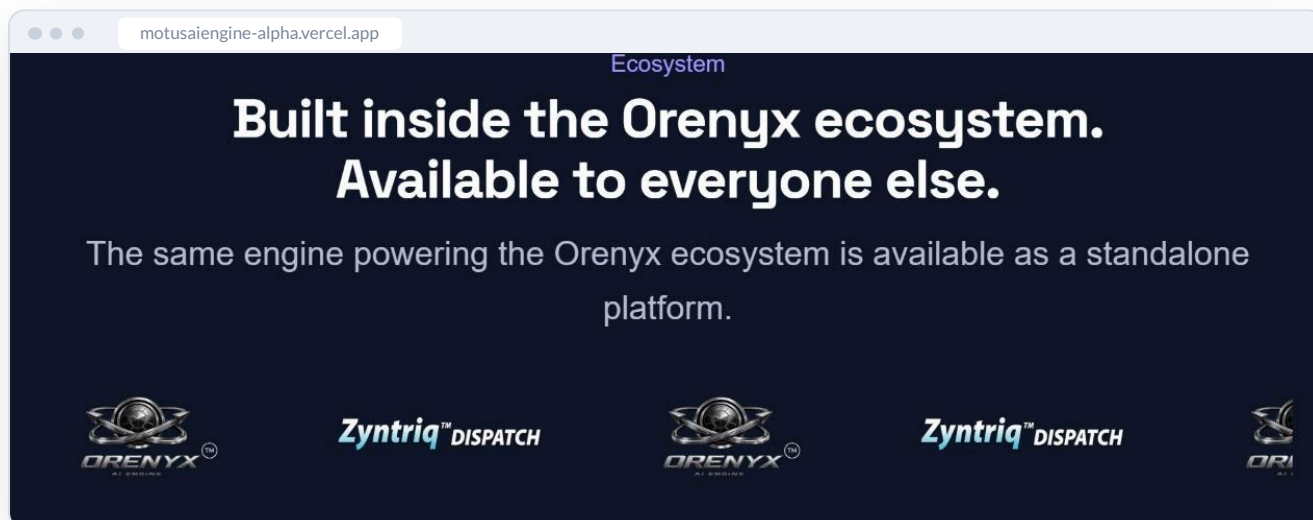
HOMEPAGE

From the problem to the business case

After the diagram, the homepage moves from what the engine is to what it does, who already runs on it, and what it's worth to a business.



The solution section breaks the engine into its three jobs: dispatch routing, bot orchestration and payment decisioning.



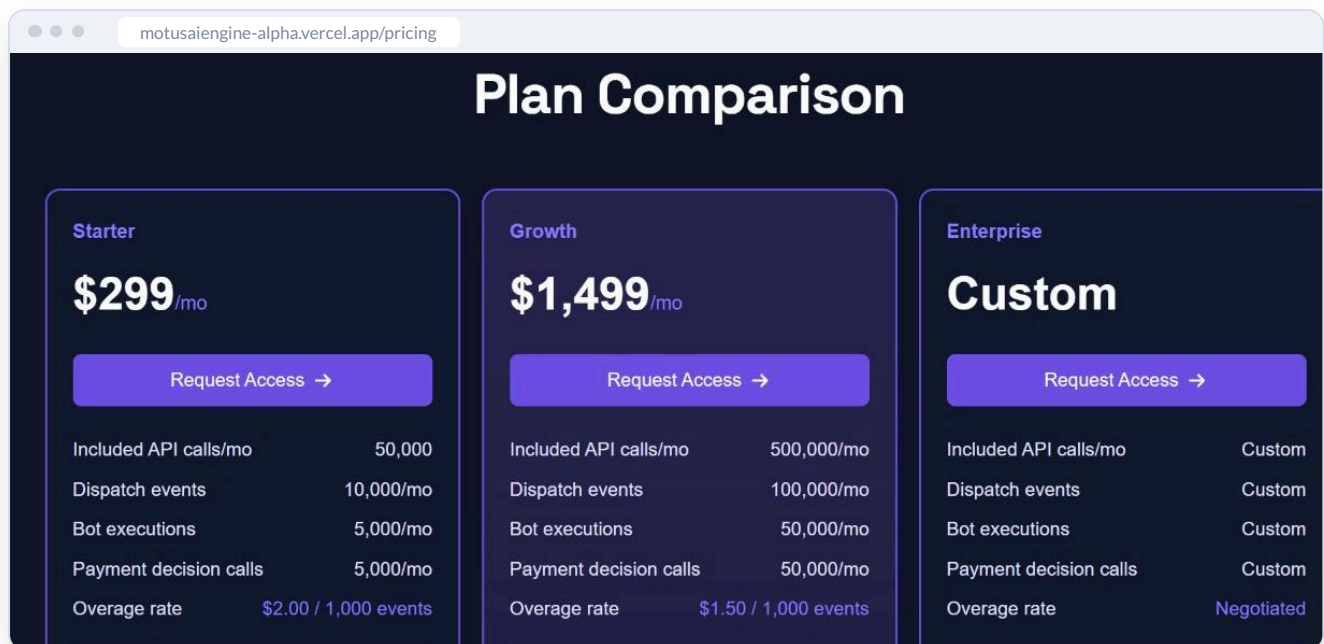
The ecosystem marquee. The Orenyx mark stays pinned while the product names scroll, kept inside the page grid.

Further down sit a business value section, a trust section and a closing call to request access. A falling-code canvas panel and a milestones timeline, with cards that slide in tilted and settle straight as the rail draws down, round out the motion on the site.

PRICING AND DEVELOPERS

Usage pricing, made easy to estimate

The engine is priced as a base platform fee plus usage. The pricing page shows three plans side by side and includes an estimator, so a buyer can work out a monthly cost from their own numbers before they ever talk to sales.



The screenshot shows a web browser window with the URL `motusaiengine-alpha.vercel.app/pricing`. The page title is "Plan Comparison". It features three pricing plans: Starter, Growth, and Enterprise. Each plan has a "Request Access" button and a table of included features and their limits.

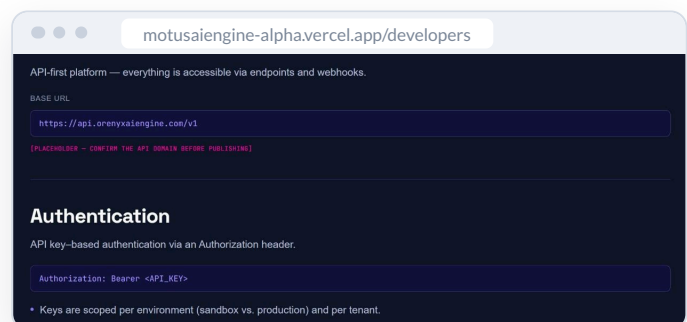
Feature	Starter	Growth	Enterprise
Included API calls/mo	50,000	500,000/mo	Custom
Dispatch events	10,000/mo	100,000/mo	Custom
Bot executions	5,000/mo	50,000/mo	Custom
Payment decision calls	5,000/mo	50,000/mo	Custom
Overage rate	\$2.00 / 1,000 events	\$1.50 / 1,000 events	Negotiated

Plan comparison. Each plan lists included API calls, dispatch events, bot executions, payment decision calls and the overage rate.

Docs that developers expect

The developer section covers the base URL, authentication, core endpoints, rate limits, error codes, webhooks, versioning, the SDK roadmap, sandbox mode and getting started, with sub-pages for authentication and the changelog.

Open items, like the final API domain, show in magenta on the preview. That is the packet's own convention, so nothing unconfirmed can slip out looking final.



ENGINEERING

A codebase the client can keep running

The site is built on Next.js 15 App Router with React 19, Tailwind CSS v4 and TypeScript, deployed on Vercel. The aim was a codebase where changing copy or colour is a one-file job.

AREA	HOW IT'S BUILT
Design tokens	Every colour, font and spacing value lives in one place, under Tailwind v4's <code>@theme</code> in <code>globals.css</code>
Content	Copy sits in its own <code>content/</code> files, apart from the components that lay it out
Launch gate	A check script fails the release with an error while any <code><Placeholder></code> is still in the code
Request Access form	A route handler sends each request by email through Resend, with a honeypot field to stop spam bots
Header and footer	Sticky header with a Log In and Status utility bar and a mobile drawer, and a two-layer footer
Fonts	Self-hosted through <code>next/font</code> , with a system font stack where a webfont licence was unclear
Pages	Home, About, Features, Pricing, Use Cases, Developers and sub-pages, Integrations, Security, Careers, FAQ, Contact, Status, Login and four legal pages

Checked with a real build

Some mistakes only show up in `next build`, not in development. Every change was verified with a full production build before it was handed back, which caught issues such as a stray export in a page file and a filename with a space in it.

Handed over cleanly

Each round started from the latest project code and kept every manual edit already made to it, so nothing was overwritten. Deployment runs from GitHub to Vercel, and we flagged that a commercial site needs Vercel's Pro plan, not the free tier.

THE OUTCOME

What Orenyx got

- A site that makes an invisible engine easy to picture.
- Pricing a technical buyer can estimate alone, and docs a developer can start from.
- Custom animation that renders on the server and respects reduced motion.
- A typed Next.js 15 codebase with tokens and copy in one place each.
- A launch gate that keeps unfinished copy off the live site.
- The benchmark later Orenyx sites were built to match, starting with Influencer Connect.

STACK

Next.js 15 · React 19 · Tailwind CSS v4 · TypeScript · GSAP · SVG and canvas · Resend · Vercel

Need a site for a technical product?

We build WordPress, Shopify, Laravel and Next.js sites for clients across Europe, Australia and the US, from developer-facing marketing sites to custom plugins.

EMAIL sales@rishfotech.com

PHONE [+91 98747 00540](tel:+919874700540)

BOOK A CALL calendly.com/rishav-rishfotech/30min

WEB rishfotech.com

