MeiuxMeiux

SOFTWARE · LOGISTICS · CONSULTING

REQUEST FOR PROPOSAL · GREENFIELD TRACK

MEIUXMEIUX · ENGINEERING PARTNERSHIP

Greenfield this repository

An invitation to bid on building Nah from scratch — a credit-only, white-label electronic-signature platform — your way, against a fixed set of requirements and rules.

PROJECT **Nah**

CONTENTS

What's inside this document

Read this top to bottom. Every section is short and written in plain language. Each one ends with what it means for you. Technical words are explained the first time they appear and again in the glossary at the end. Do not worry if English is not your first language — this document was written to be easy to translate.

How to read this document

> **This is a request for a proposal, not a contract**
> We are looking for one or more developers to **build** a software project called **Nah** from scratch. We want you to read this, understand the requirements, and then send us a written offer: what you will build, when, in what order, with what technology, and how you want to be paid. We will compare offers and reply.

There are **two** versions of this document.

- **A first one — "Complete this repository."** You take code we already wrote and finish it.
- **This one — "Greenfield this repository."** You ignore our code and build the same product fresh, **your way**, against the requirements here.

You may bid on either path. Pick the one you are most confident in. Both paths end with the **same finished product** and the **same rules** about security, ownership, and how the work is hosted. The greenfield path gives you the freedom to choose how it is built — within the frame described in Section 5.

> **One thing we will not discuss in this document**
> We do not state any prices, rates, or budgets here. Money is decided after we see your offer. Section 10 explains the *shapes* of payment we are open to. Please put your numbers in your reply, not your assumptions.

What a good reply looks like

A short written proposal that covers: what you will build, the technology you will use, the order of work, how long each part takes, when and how you expect to be paid, the people doing the work, and any questions. Section 11 lists this in full. You do not need slides or a long company profile. We value clarity over polish.

What this section means for you: read everything once, then decide whether the freedom of greenfield fits you better than continuing our code, then write us a plain offer.

SECTION 02

Nah in one minute

Nah is an **electronic-signature platform**. It lets a company send a document, have people sign it legally online, and keep a tamper-proof record of who signed and when.

Nah does not build the cryptographic signing itself. It is a **white-label wrapper** around a signing service called **Firma.dev**. "White-label" means the end customer sees *their own* brand, not ours and not Firma's. Nah owns the customer relationship, the look, the pricing, and the extra tools around the signature.

The selling point is the **pricing model**. Most signing products charge **per seat** — every employee who might ever sign pays a monthly fee. Nah charges **per document** instead, from a prepaid **credit** balance. A business buys credits, spends one credit per document, and pays nothing when nobody is signing. This is friendlier for the many firms whose signing volume is uneven.

1 repo
Four products ship from a single codebase

Credit
Prepaid, pay-per-document — no per-seat fees

BYO brand
Each reseller client keeps their own branding

Nah is also **multi-tenant**. We resell it to many client firms. Each client gets their own branded space, their own credit wallet, their own users, and — through Firma's API — their own isolated signing workspace. One running system serves all of them, kept apart by software.

What this section means for you: the product is a re-brandable, credit-billed signing portal with extra workflow and AI tools layered on top of Firma.dev. You are building all of it.

SECTION 03

The four parts you must deliver

Whichever path you bid on, the finished product must contain **four parts**, and they must all live in **one repository** (one codebase).

PART 1

The public website

The marketing landing page and the place where the desktop app is downloaded. Fast, simple, brandable.

PART 2

The web app

The signed-in product in a browser: upload a document, let AI place the fields, choose signers, send, track, and download the finished signed file.

PART 3

The desktop app

The *same* web app, wrapped as one installable program for **Windows 11 and macOS**. It must update itself when a new version exists.

PART 4

The admin panel

Our private control room: manage client firms, credits, sales, free-credit grants, run promotions, see document history, set branding.

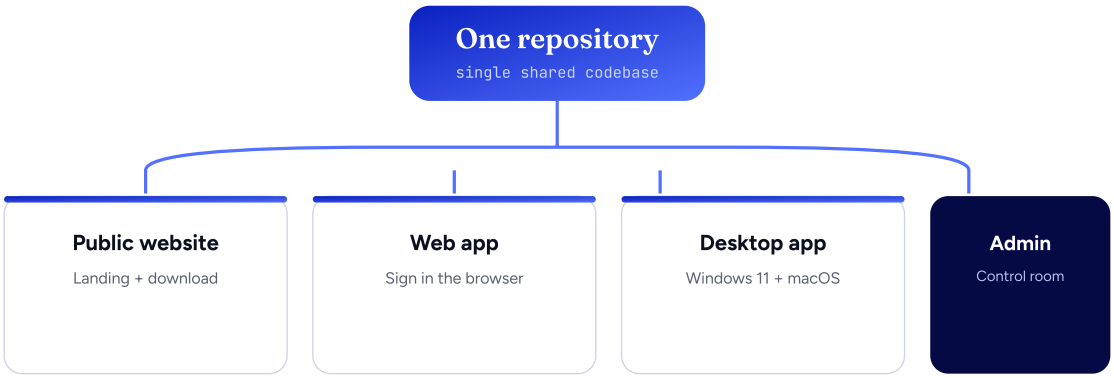

Figure 1. Four products, one codebase. The desktop app is the same web app in a native shell — it must never be a second, separate program.

The web app and the desktop app **share the same screens**. The desktop app is the web app placed inside a small native wrapper so it can open and save files on the user's computer. Building the screen twice is not allowed.

What this section means for you: your bid must deliver all four parts from one repository, and parts 2 and 3 must share one set of screens.

SECTION 04

How the system must fit together

There must be one **backend** (the server program) and one **frontend** (the screens). The frontend runs in the browser and inside the desktop app. The backend is the only part that is allowed to hold secret keys and talk to outside services.

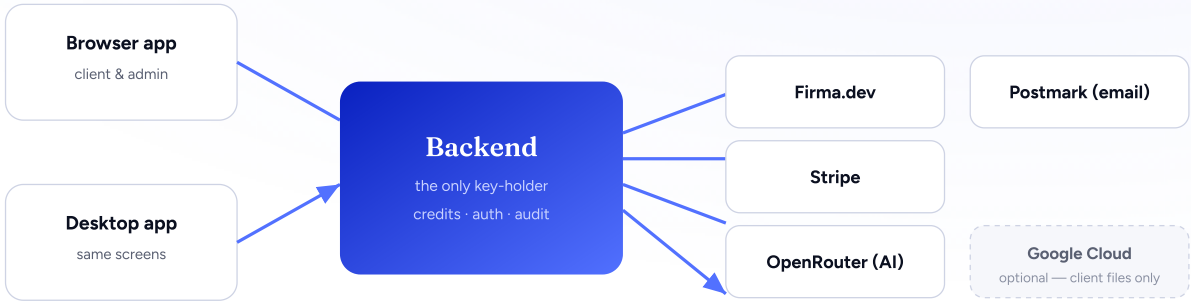

Figure 2. Outside services are only ever reached **through the backend**. The browser and desktop app never hold a Firma, Stripe, AI, or email key.

Three rules drive this picture, and they are not negotiable — they hold no matter which technology you choose:

RULE 1 **The frontend never holds a privileged key**

Only the backend talks to Firma, Stripe, the AI provider, and email. A leaked browser bundle must never expose a usable secret.

RULE 2 **Secrets are entered at runtime, never stored in the repository**

Every key is typed into the admin panel after launch and stored encrypted in the database. The codebase carries zero real secrets — ever.

RULE 3 **No cloud services, with one allowed exception**

The system runs on a plain server. The only permitted cloud dependency is **optional** Google Cloud Storage, and only for client-owned files (their templates, documents, uploaded images).

What this section means for you: whatever stack you propose, design every feature so that the only program holding a real secret is the backend, and the only place a human enters a secret is the admin panel.

SECTION 05

A clean slate — with a fixed frame

"Greenfield" means you do **not** have to use or read our existing code. You design the architecture, choose the technology, and build it the way you believe is best.

That freedom has a frame around it. The frame is small and it is firm:

FIXED	**We own and control the repository** Even on the greenfield path, the project lives in a private repository on our company account. You are invited as a collaborator. We own it from the first commit.
FIXED	**We own and control the servers** You build on a development server we provide and pay for. Production is a separate server only we control. See Section 8.
FIXED	**The four parts, one repository** Website, web app, desktop app, admin — one codebase, every time.
FIXED	**The three non-negotiable rules from Section 4** Backend-only secrets, runtime secret entry, no cloud except optional Google Cloud Storage for client files.
FIXED	**just / justfile drives build & deploy** Whatever the stack, the build, test, and deploy commands run through **just / justfile** so we can take the project over easily later.
FIXED	**Passkey authentication and portability** Passkey sign-in for clients and admins (passphrase fallback, email recovery, no SMS). The product must move cleanly between domains and environments.

We can hand you a head start — if you want it

We have already built a document editor and an AI-analysis flow for an earlier, single-customer project. If it helps your plan, we can place a copy of that code in the repository for you to draw from. On the greenfield path this is **optional** — use it, adapt it, or build your own. Tell us in your bid which you prefer.

The pitch we are making to you

This path is for a team that says: *"We already know exactly how we would build this. Give us the requirements and the freedom, and we will deliver it."* If that is you, the greenfield path is yours to win.

What this section means for you: you choose the "how"; we fix the "what", the ownership, the hosting, and the six framed items above.

SECTION 06

Everything you must deliver

This is the full product, written as **acceptance criteria** — the things that must be true for the work to be accepted. Group and order them however your plan prefers.

AUTH	**Passkey sign-in for clients and admins** Passkey-first, passphrase fallback, email-based recovery, no SMS. Roles for platform admin, client admin, and client member, enforced on every screen.
SECRETS	**Runtime secret management in the admin panel** Every provider key is entered after launch, stored encrypted, never displayed again, and testable with one click. Zero real secrets in the repository.
TENANCY	**Multi-tenant client isolation** Each reseller client has its own branding, users, wallet, and an isolated Firma workspace. No client can ever see another client's data.
BILLING	**Payments & credit management (Stripe)** Buy credit packs, optional subscription, automatic top-ups, refunds. Admin tools to grant free credits, run sales and promotions, and adjust balances. Stripe reached only through the backend.
SIGNING	**Firma.dev integration & credit accounting** Send a document, spend exactly one credit, record a tamper-proof audit entry. A clear, agreed rule for refund-on-failure.
WORKFLOW	**Complex document workflows** Multiple signers in a chosen order, CC recipients, and the ability to change participants and order while a workflow is in progress without breaking the audit trail. (See Section 7.)
AI	**AI assistance & bring-your-own-key** AI document analysis paid for with credits, backend-only. A client may supply their own AI key for a discounted (not free) rate.
AI	**AI auto-input field mapping** The system reads an uploaded PDF, proposes where fields belong, and the end client can prompt the AI in plain words to assign signers and recipients. A human confirms before sending.
DESKTOP	**One installer, Windows 11 + macOS, self-updating** One bundled installer wrapping the same web screens. It updates itself, authenticates the user, shows credits, and can buy services — all through the backend.
MIGRATION	**PandaDoc & DocuSign migration helper tools** Tools that read a new client's existing templates with their own API key and show which the AI has already mapped. (The hands-on migration *service* is ours to offer, not yours to perform.)

SITE

Public website + download + status page

A fast, brandable marketing surface, the desktop-app download, and a public technical/health page.

SHIP

Safe operations

Atomic deploys with one-step rollback, backups before data changes, no destructive database actions, portable across domains and environments.

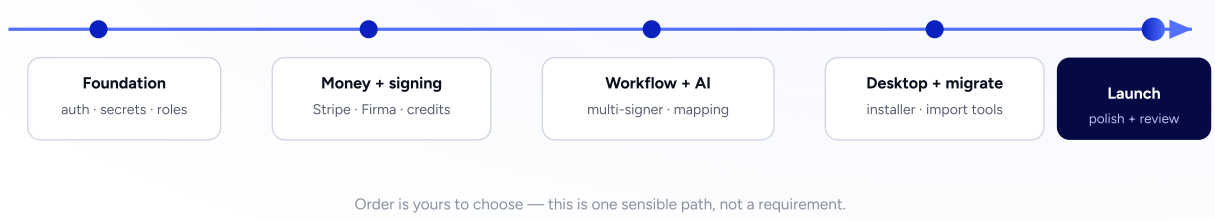

Figure 3. One sensible route from zero to launch. Your bid may propose a better order.

What this section means for you: these criteria, plus the rules in Sections 4 and 5, define "done." Estimate them in your bid.

SECTION 07

The signing workflow we need

This is the heart of the product. It must support real-world business documents, which are rarely "one person signs one box."

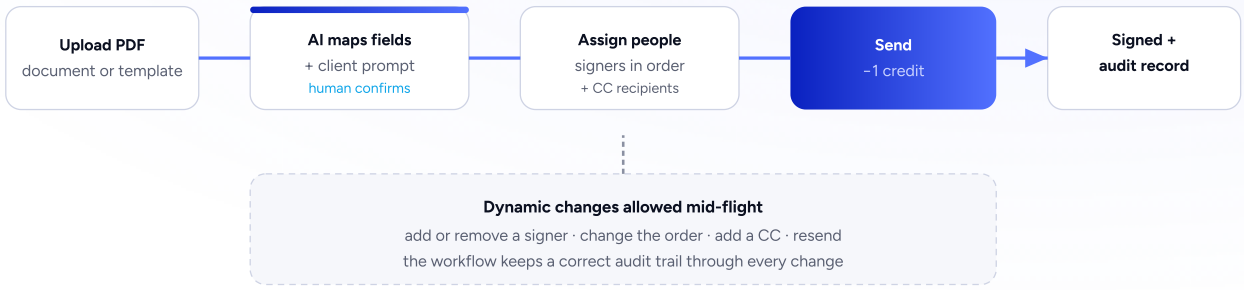

Figure 4. Upload → AI mapping (with a human check) → assign people → send (one credit) → signed file plus a tamper-proof audit record. Participants and order can change while the workflow is running.

The workflow must support, at minimum:

- **Multiple signers**, signed in a chosen order (or all at once).
- **CC recipients** who get a copy but do not sign.
- **Dynamic adjustments**: change who is involved, and in what order, after the workflow has started — without breaking the audit record.
- **AI auto-input**: the system proposes which field belongs to which signer. The end client can also type a plain-language instruction (for example, "the buyer signs page 3, the witness initials every page") and the AI assigns fields accordingly. A person always confirms before anything is sent.
- **Credit accounting**: exactly one credit per document send, with a clear, agreed rule for what happens if Firma fails after the credit is taken.

What this section means for you: plan for editable, multi-party workflows and an AI mapping step with a human confirmation — not a single-signer happy path.

SECTION 08

How you will work with us

This section protects **both sides**. It lets you work freely and fast, and it lets us keep client data and our production system safe. Please read it carefully and price it into your bid; it changes how you work day to day. It applies to the greenfield path exactly as it does to the other path.

We use a **two-server model**.

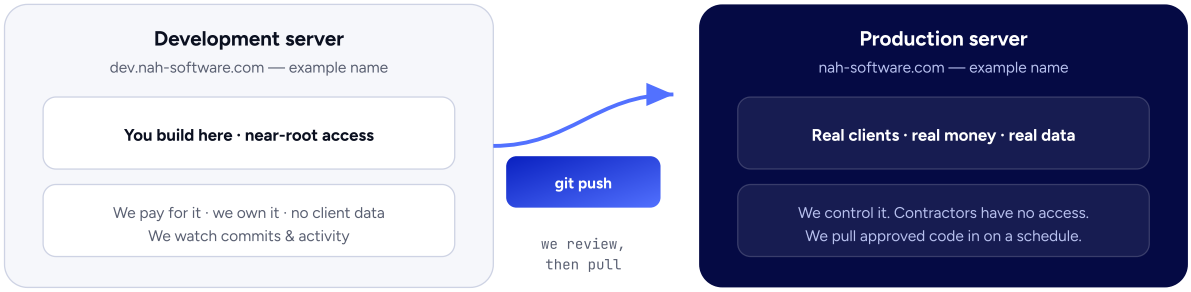

Figure 5. You work on a development server we own and pay for. We pull reviewed code into a separate production server you cannot reach.

The development server — yours to use

- We provision a dedicated **development server (a VPS)** and **we pay for it**.
- You get a working account with enough access to install the tools you need and run the project. You can treat it as your workshop.
- It carries **no real client data** and **no production secrets**. If something goes wrong on it, nothing important is lost.
- One server is fine for development and testing together. Your domain for it could be, for example, `dev.nah-software.com` (illustration only).
- We keep a few protected, owner-only tools on the box (for example, automatic database backups) that your account cannot see or change.

The production server — ours to control

- The real, paying system runs on a **separate** server that **only we** control. Contractors do not have access to it.
- You commit and push your work to the repository. **We review it and pull it into production on our own schedule.** You never deploy to production.
- This means we never have to worry about what happens on the development box. You have freedom there; we keep the real system safe.

The repository — ours to own

- We create the repository on a company account and invite you as a collaborator on a **private** repository.

- This is true even on the greenfield path. We own and control the repository in every case.
- We track progress through commit history and activity on the development server. We will not wait silently and hope a finished version appears. We expect steady, visible progress.

> ✓ **Why this is good for you**
> You get near-total freedom on a powerful machine we pay for, with no fear of breaking anything real. You are never on call for production. The boundary is simple: build freely on dev, we carry the risk on prod.

1. **We set up the dev server, domain, and repository**
   We prepare the empty repository and the development environment so your build is portable from the first commit.
2. **You build on the dev server and commit often**
   Small, frequent commits with clear messages. Your build, test, and deploy commands run through just / justfile.
3. **You push to the private repository**
   We watch the commit history and the dev activity, so progress is always visible to both sides.
4. **We review and pull into production on our schedule**
   We may also contribute our own work on our own development copy. Production is always our responsibility, never yours.

What this section means for you: plan to work on a server we own, commit and push constantly, and never expect production access. Price the workflow, not just the features.

SECTION 09

Your stack, your call — within limits

This is the section that makes greenfield different. You may choose the technology. We want to hear how you would build this.

Propose your stack — we may accept or reject it

If you can meet every requirement in this document with a technology you know well, put that proposal in your bid and explain the trade-offs. We are genuinely open to hearing it. But we will have to maintain this product for years after you finish, so we **reserve the right to reject** a stack we do not want to inherit. A rejected stack is not a rejected bidder — we may simply ask you to use ours.

For reference, the technology **we** use and are comfortable maintaining is below. Matching it makes your bid easier for us to accept; departing from it is allowed but must be justified.

Backend

Go — one compact, fast server program. SQLite to start. Plain and dependency-light by choice.

Frontend

SvelteKit + TypeScript, with Tailwind CSS. One screen set shared by web and desktop.

Desktop

Electron, packaged with electron-builder. Windows 11 first, macOS to follow. Self-updating.

Server

A plain Linux VPS behind the Apache web server. Pull-mode deploys, atomic swap, one-step rollback.

Two things are **not** negotiable regardless of stack:

ALWAYS **just / justfile drives every build and deploy command**

So we can take the project over without learning a bespoke process. This is a hard requirement on every path.

ALWAYS **The rules in Sections 4 and 5 still hold**

Backend-only secrets, runtime secret entry, one-repo-four-parts, no cloud except optional Google Cloud Storage, passkey auth, portability.

What this section means for you: bring your best stack and defend it, but accept that we may ask for ours — and that just / justfile and the core rules apply either way.

SECTION 10

How we can pay

We will not state numbers here. We will say clearly that we are **flexible** and open to several payment shapes. Tell us which one you want in your bid.

OPTION A

By the hour

You bill for time invested. Good when scope is uncertain or exploratory.

OPTION B

By feature / milestone

A fixed amount per agreed feature or work area, paid as each is delivered and accepted.

OPTION C

Whole project

One agreed amount for the entire finished product, paid across milestones.

We are happy to pay across **intervals tied to milestones** — for example, a portion when a work area is delivered and accepted. We can also mix these (for example, hourly for discovery, then per-feature for delivery).

We may also contribute

We can keep our own development copy and add our own time and skills to the project alongside you, on any of the four parts. You are a partner here, not a vendor we hand a wall to.

What this section means for you: choose the payment shape that fits how you want to work, and propose the milestone points where you expect to be paid.

SECTION 11

What we need back from you

Send a short, written proposal. Please cover every point below. Clarity matters more than length or design.

YOUR BID

The contents of a strong reply

1. **Which path** you are bidding on — "greenfield" (this document) or "complete".
2. **Your stack** — the technology you will use, and why, including any trade-offs (we may accept or reject it).
3. **Your plan** — the requirements you will deliver and the order you will deliver them in.
4. **Milestones** — how you break the work into deliverable, acceptable pieces.
5. **Timeline** — a realistic estimate for each milestone and for the whole job.
6. **Payment** — which shape from Section 10, the amounts, and at which milestones you expect to be paid.
7. **Reference code** — whether you want our existing document-editor / AI code as a starting point, or will build your own.
8. **Team, questions & assumptions** — who does the work, anything unclear, and the assumptions your estimate depends on.

We will read every serious reply and respond. A counter-proposal that improves on this plan is welcome — that is the point of asking.

What this section means for you: answer these eight points and you have a complete bid.

SECTION 12

Glossary

Plain definitions of the terms used in this document.

Electronic signature `e-signature`	A legally valid way to sign a document online instead of on paper.
Firma.dev	The outside service that performs the actual cryptographic signing. Nah wraps it; it never touches our customers directly.
White-label	A product sold under someone else's brand. Each Nah client shows their own name and look, not ours.
Credit	A prepaid unit of value. One document send costs one credit. Clients buy credits in advance.
Multi-tenant	One running system that serves many client firms while keeping each one's data and branding separate.
Backend	The server program. It holds the secrets and talks to outside services. Users never see it directly.
Frontend	The screens a user sees and clicks, in the browser or the desktop app.
Greenfield	Building fresh from nothing, choosing your own design and technology, instead of continuing existing code.
Acceptance criteria	The specific things that must be true for a piece of work to be considered finished and accepted.
Passkey	A modern, password-free login using the security built into a phone or laptop.
VPS	Virtual Private Server — a rented computer in a data center that we control.
Repository `repo`	The place where all the project's code is stored and its history tracked.
Commit / push	Saving a unit of work to the project history, then sending it to the shared repository.
Atomic deploy	Releasing new code in a way that switches over instantly and can be undone in one step.
Pull-mode	The server fetches approved code itself. Nobody pushes files onto production by hand.
CC recipient	Someone who receives a copy of a document but does not need to sign it.
Audit record	A tamper-proof log of who did what and when — essential for legal signatures.
BYOK	"Bring your own key." A client supplies their own AI provider key to lower their own cost.

just / justfile	A simple task runner. One short command runs the right build, test, or deploy steps.
PandaDoc / DocuSign	Competing signature products. New clients may want to move their templates over from them.

Ready to bid?

Send your written proposal covering the eight points in Section 11 — including the stack you would choose and why. We read every serious reply and respond. Honest questions and counter-proposals are welcome — a better plan than ours is exactly what we are hoping to receive.

PROJECT	REFERENCE	REACH US
Nah — Greenfield track	NAH-RFP-GREENFIELD	meiuxmeiux.com/contact