THIS WEEK'S FILE · FOUNDER FILE #004

THE MAN WHO LAMINATED EVERY CARD AT HIS KITCHEN TABLE

A Dutch founder with a day job, a mortgage and two kids ran his business on Google Forms for five years. Then he rebuilt it with Claude Code, took a month on a boat, and says the hard part was understanding the process first.

THE FILE

FOUNDER	Maarten (surname withheld), Founder and only employee
COMPANY	Withheld at his request
FOUNDED	2020
HQ	Netherlands
SECTOR	Road-traffic exemption admin for infrastructure repair firms
STAGE	Bootstrapped, one-person company
FUNDING	Undisclosed (started as an evenings-and-weekends side business)
MODEL	Charged per vehicle, invoiced monthly

Hello again, and welcome to File #004. Thank you for staying with us.

This week is a little different, and we think you'll like it. There's no billion-dollar valuation and no lab. There's a man at a kitchen table in the Netherlands, with a full-time job, a mortgage and two small children, laminating plastic cards one at a time.

He did that for five years. Every card had to be right, because a wrong one is worse than useless. At the end of it he rebuilt the whole business with an AI coding tool, took four weeks off on a boat, and, by his own account, didn't open the tool once.

We'll call him Maarten, because that's all we're allowed to. He has asked us to withhold his surname and his company's name, and we've respected that. (All figures are in euros, and they are his.)

THE SHORT VERSION

- **Who:** Maarten, founder and only employee of a Dutch company he started in 2020.

- **What:** he handles road-traffic exemption paperwork for infrastructure repair firms: the gas, water and telecoms crews who have to park where nobody is normally allowed to.
- **Why it matters:** it is a small, unglamorous, very real business, and it shows what changes when a non-developer can build custom software for a process he already understands.
- **The number to remember:** four weeks. That's how long he was away on a boat last summer, on about three calls a week and an hour of email, without opening Claude Code once.

THE BUSINESS, IN PLAIN LANGUAGE

In the Netherlands, gas pipes, water mains and telecom cables run underground. When one breaks, a repair crew has to get a van or a small excavator as close as it can, which often means parking on a footpath, up a kerb, or past a "no entry" sign. The authorities allow it by granting an exemption. In return, every vehicle and every mechanic using the exemption has to be registered and traceable.

What the repair firm actually wants is simple: a card for the dashboard that it can show to a police officer. That card, with the right plate and the right name, is the product. Everything else is paperwork to make that card correct.

And the work never stops. Mechanics join and leave, vans are replaced, leases end. Every one of those changes has to be processed, checked and reprinted.

(For context, exemptions of this general kind in the Netherlands are usually granted under the national traffic regulations, the RVV 1990, often by municipalities and for limited periods. We haven't verified the exact scheme Maarten works with, and the case study doesn't spell it out.)

THE FOUNDER

Maarten is not a developer, and he is clear about that. He spent about a decade of his working life under ISO 27001, the international standard for information security, which means he spent years working to written security policies. He calls himself a goldfish when it comes to memory.

The idea came from his day job. While working for a contractor, he watched the firm pay an outside company to handle its road-traffic exemption admin, and watched that company keep getting it wrong. He asked whether he could take it over. His employer said yes, as long as he could do it better for the same price.

So the company began in 2020 as a side business, built in evenings and weekends around a full-time job, a mortgage and two children.

FIVE YEARS ON FORMS AND A LAMINATOR

Here is the part that surprised us. He kept it manual on purpose. His view, and he comes back to it again and again, is that if you automate a process you don't understand yet, you automate the wrong thing.

So for five years the business ran on seven Google Forms, one Google Sheet, Stripe and Gmail. The forms covered company onboarding, driver and vehicle onboarding, plate changes, offboarding, renewals, objections and an eligibility check. The six steps looked like this:

STEP	HOW IT WORKED	WHO DID IT
1. Data in	The customer types into a Google Form	The customer
2. Checking	He reads the sheet line by line and corrects by hand	Him
3. Invoicing	At month end he counts each company's vehicles and adjusts the quantity in Stripe	Him
4. Customer changes	He processes every driver and vehicle change as it arrives	Him
5. Printing	He emails a CSV to a printing company	Him
6. Final check	When the cards come back he checks and laminates every one	Him

He was, as the case study puts it, the validation layer and the safety net. A form accepts whatever you type, so the errors came through.

A typo in a licence plate gave a card that didn't match the vehicle, which is the one thing the card exists to do. A wrong address sent a card somewhere that didn't exist. Once he overcharged a customer because he had counted the vehicles wrong, and he says it felt stupid.

Form validation helped a little, for example forcing a phone number to ten digits. But a form can't tell whether a plate belongs to a real vehicle or an address to a real building. And checking got less reliable as the volume grew, because the more you check, the more you get used to everything being right and the faster you go. That is a very human failure, and he knew it.

He also answered every request on the day it arrived, seven days a week. On one family holiday he told his family he would only stay at the house for the morning, then worked through the afternoon too.

He knew the set-up was wrong and had a growing backlog. He tried a no-code tool, did two courses, and struggled for months. Learning a new platform at ten at night after a full day's work, he found, didn't work.

THE THURSDAY

Then came Claude Code, Anthropic's coding assistant. He installed it on a Thursday. By the end of that day, it had built the things he had been stuck on for months. Within a week, everything on his backlog board was working code.

A quick note on dates, because this is where the story needs care. The case study says Claude Code "arrived at the start of 2026". It didn't: Claude Code launched as a research preview in February 2025 and became generally available in May 2025. Our best guess is that this is when Maarten came to it, not when it was released. The case study also says the system went live in January, but a week of building plus six weeks of data migration doesn't fit comfortably into January if he started at the beginning of the year, and a shorter summary elsewhere gave a different go-live date. We can't reconcile these from here. What's solid is the sequence: one Thursday, a week of building, about six weeks of migration and testing, then customers.

And it was not easy. This is a full custom application with logins, payments, per-role permissions, two government data connections and a supplier integration, and it holds other people's personal data. Maarten is explicit that this was hard. It wasn't a weekend project.

What made it possible is worth listing, because it's the part most readers will overlook:

- Five years of running the process by hand, so he understood every step.
- An existing base of paying customers, so he had a known specification, not a guess.
- Free public Dutch data for plates and addresses.
- A print supplier willing to offer an API.

BEFORE AND AFTER

The core design is one idea: move the checking to the front. The system now refuses data it can't verify, so bad data never gets in.

STEP	BEFORE	AFTER
1. Data in	Google Form accepts anything	Plates checked against the RDW, the Dutch vehicle authority. Type a plate and the app shows the make and model for the customer to confirm. Addresses checked against a Dutch address database. Formats enforced on entry
2. Checking	Him, line by line	Gone as a separate job
3. Invoicing	Him, at month end, in Stripe	Generated from the live vehicle count. Payments through Mollie, matched automatically

STEP	BEFORE	AFTER
4. Customer changes	Him, one by one	Each company gets a dashboard to manage its own drivers and vehicles, with permissions per role. The 800-vehicle client got a CSV import. Local authorities have their own view
5. Printing	A CSV, emailed	Sent by API to a new supplier, ADDI, which sends back a status: printed, in progress or dispatched
6. Final check	Him, at the kitchen table	Gone. The supplier laminates, and validation caught the errors earlier

After the rebuild, he personally does none of the six. The two that vanished completely, the checking and the final check, were the two where he had been the safety net.

One detail we liked. A client with a leased fleet asked him to stop drivers editing their own vehicle details. The feature didn't exist. He told them he could turn it on anyway, put the phone down, prompted Claude Code, and called back fifteen minutes later to say it was done.

A note on that plate lookup, because it's smaller than it sounds. The Dutch vehicle authority publishes vehicle data as free open data, and the lookup returns make, model and similar details but not who owns the vehicle. That is exactly why the app asks the customer to confirm the car on screen: it can't tell you whose it is, but it can tell you whether the plate is real and what it belongs to.

WHAT IT RUNS ON, AND WHAT IT COSTS

TOOL	JOB	WHAT IT REPLACED
Claude Code	Building and maintaining the app	Years of not getting round to it
Supabase	Database and logins	The Google Sheet
Vercel	Hosting	Google Forms
Mollie	Payments and reconciliation	Stripe, and his month-end checks
Resend	Transactional email	Gmail, partly
Sentry	Error monitoring	Him noticing
RDW open data	Licence plate lookup	Manual retyping
Dutch address database	Address validation	Nothing; addresses went unchecked
ADDI	Printing, laminating, dispatch status	A CSV and his laminator
MailShield	Email deliverability	Nothing
Squarespace	Marketing site	Nothing

The running cost is about €354 a month, plus a Claude Max plan. That comes from his own billing, and he thought it sounded high and offered to check it, so treat it as an upper estimate. He had also bought a customer-support tool, Intercom, on a startup plan at about €65 a month, which would have been over €900 a month at standard prices after three years. He replaced it with custom in-app support.

WHAT CHANGED

- **Time:** operations take about four hours a week, down from about eight. These are his estimates, and nobody was timing either period. The difference that matters is that the old eight had to happen every day as requests came in. The new four can wait.
- **The boat:** last summer he took four weeks off. He had about three calls a week and roughly an hour of email, and didn't open Claude Code once. Anything that occurred to him went on a list for when he got back.
- **Revenue:** about €300,000 a year. Please read that carefully. The business was already growing before the platform, so the platform doesn't get all the credit. What he does credit it with is helping him win bigger, more established clients, because a self-service product looks professional, and freeing time for sales and marketing.
- **The one thing left:** selling. When he paused outbound over the summer holiday, that was the part that slowed. Sales is the only part of the business that still needs him.

THE PART HE IS BLUNT ABOUT

"It's an easy tool, it easily fools you," Maarten says.

The risk is real. The code was written by an AI that he can't fully judge himself, and the app holds personal data, so a leak would end the business. His answer is a routine, not a one-off. Every week he has Claude go back over the codebase and check that it's still solid, still secure and that it still scales. He runs the review again whenever a new model comes out. He usually finds something to improve, and fixes it. He also runs Sentry, so errors reach him instead of waiting for a customer to notice.

He isn't guessing, either. A decade of ISO 27001 means he knows what good practice looks like, which is an advantage most people building this way won't have. His advice to anyone holding sensitive data is to make the review a habit, and the first step is simple: point the AI at your own code and ask it to review for security problems. It takes an afternoon, and it's better done before you have customers than after.

HIS RULES

1. **Get the process right before you automate it.** He ran on Google Forms for five years on purpose. Automate a process you don't understand and you just make the wrong thing faster.

2. **Don't build the spaceship.** If you keep adding features, it never flies. Start with the one process that costs you most.
3. **Solve the customer's problem, not yours.** The customer wanted a correct card on the dashboard. They didn't care that he laminated it in his kitchen. Making life easier for himself came later.

And one more, which he gives as his answer to the obvious question: if he started the same business today, he'd go straight to Claude Code on day one.

> **Try this**
>
> **The lookup at the front.** Look at the last ten corrections you made by hand. Count how many were a value that could have been looked up: a plate, an address, a postcode, a company number. That count is the size of your problem. Then find the one free or public lookup for the most expensive field, and put the check at the very front, so bad data never gets in.
>
> **The weekly review.** Once a week, ask your AI tool to go back over your code for security, stability and scale. Run it again when a new model arrives. Add error monitoring so problems come to you.
>
> **Two quick counts.** Have an AI draft replies to your last 20 emails and count how many you could send unchanged. Write the data rules for your spreadsheet first, then count how many rows fail them. That's your migration estimate.

WHAT COULD GO WRONG

One person. Founder and only employee means that every outage, every question and every sale still runs through one man. He has taken the humans out of the ops. He hasn't taken himself out of the business.

AI-written code he can't fully judge. His weekly reviews are sensible, but they're an AI reviewing an AI's work, with a non-developer deciding what to do about it. A serious flaw in a system holding other people's personal data would be very hard to recover from, and Maarten says so himself.

Dependencies. The whole thing leans on a public data feed, one print supplier, and a stack of hosted services. Any of them can change a price, a term or a format.

The sales problem. Selling is the one job the software can't do, and it's the part that slows the moment he stops.

One company's result. He had domain knowledge, paying customers, free national data and a decade of security training. Those are advantages, not universal conditions. The case study is also his own telling, and the figures for revenue, hours and cost are his, not independently audited.

WHAT TO WATCH

1. **Whether sales gets a system.** Can he do for outbound what he did for operations, or does growth always depend on his own hours?
2. **The first real incident.** Every system has one. How fast Sentry and the weekly review catch it, and how he handles a customer, will say more than any claim.
3. **Scale.** The business was already growing. Does it keep growing without him hiring, or does the single-person structure become the ceiling?
4. **The suppliers.** Does ADDI, and the public data he leans on, stay reliable and priced as they are?

He was a man at a kitchen table with a laminator. What changed isn't that he stopped caring about the cards. It's that he finally got the checking to happen before the data came in, and not after.

That's File #004. Thank you for reading it. If it made you think of someone, forward it along, and if there's a founder you'd like to see opened next, tell us at info@aifounderfiles.com. File #005 lands next Sunday.

Where this came from: the account here is based on an interview published by Create With as Case Study 01, with figures supplied by the founder. The company name and surname are withheld at his request, and we haven't invented them. A note on disclosure: Anthropic makes Claude and Claude Code, and Claude helped research and draft this file. Spotted a mistake? Email info@aifounderfiles.com and we'll correct it.

SOURCES

- Case Study 01: Maarten, road exemptions, Create With (interview of 4 September 2026). createwith.com
- Create With podcast, Create With. pod.wave.co
- Open data, RDW (Dutch vehicle authority). opendata.rdw.nl
- Claude (language model), Wikipedia. en.wikipedia.org
- Anthropic funding announcement quoted by Simon Willison (12 February 2026). simonwillison.net