

Sites That Differ by Data, Not Code

A five-minute contract for trade-business websites, the gates that make it honest, and what the first live deploys got wrong.

Michael Kaminski

2026-09-23

michael-kaminski.io/papers

White paper 01

Summary

A trade-business website can be generated from one structured intake file, deployed to the edge, and verified live in under a minute of machine time. The content is the part that does not fit in five minutes yet. This paper describes a private template and generator for lead-generation sites (lighting, roofing, HVAC, and two non-trade verticals) where every site differs by data, not code. It records the measured time budget, the quality gates that refuse a deploy, the three things the first real deploys got wrong, and the one rule that never bends: the generator does not invent trust content.

Sources of record: the `trade-site-template` repository (private, created 2026-09-21), its twelve architecture decision records, and two client sites generated from it on 2026-09-21 and 2026-09-22.

1. The problem

A small trade business needs a website that does four things: say what the business sells, say where it works, take a lead, and be found. Most of them have a site that does one of the four. The ones that have all four paid an agency, and the site froze on the day the invoice cleared.

The economics are the problem, not the design. Building each site as a project means every site carries the full cost of a project. The way out is the way out of every build-once problem: **make the sites differ by data, not by code**. One kit renders every site. One validated JSON file describes each business. The kit improves for every client at once, and a client site is a fork of a template with one data file committed.

That is a claim about economics, so it needs a number. On 2026-09-21 the scaffold created a private client repository from the template, provisioned a database and a bot-check widget, built, deployed and verified a full-depth site in 46.5 seconds. Creating the template repository itself and pushing 101 files took 5.2 seconds. Those are measured. The content generation for a new business is not yet inside the five-minute budget, and section 4 says why.

2. The contract

The owner's definition, recorded as ADR-0010 on 2026-09-21: provide a structured input (company name, website URL, phone, email, what they sell, and free-form context), and get a deployed website with every

required media asset created, within five minutes.

Three parts of that definition do work.

- **Structured input.** One file, `intake.json`, validated by a schema. Required fields: company name, website (or an explicit `null`), phone, email, offering. Free-form context steers copy and is never treated as a factual claim. Address, hours and founding year exist only if the client supplied them.
- **Every media asset created.** An earlier plan used pre-generated image packs per trade. That meets "deploy in five minutes" but not "all media created", so the media policy became a priority order: the client's own photos, logo and colours harvested from their existing site; then the trade's pack; then freshly generated images capped by a budget; and gradient placeholders are never acceptable on a live site. A live deploy must refuse if any slot would render a placeholder.
- **Definition of done.** Live on the account's `workers.dev` address with the lead form working. Custom domain and DNS are a separate asynchronous step, because they wait on a registrar and a certificate, not on the generator.

3. What is measured and what is estimated

The time budget is a table, and the column that matters is the basis. Half of the rows are measured. Half are estimates. The paper does not blur them, because the contract is unproven until a full run is timed end to end.

Step	Time	Basis
Validate intake and site data	0.1 s	measured
Harvest the client's existing website	20–40 s	estimate
Write <code>site.json</code> (model, in session)	60–120 s	estimate
Generate 9 images: $43 + 8 \times 15.2$	165 s	measured; runs in parallel with the two rows above
Build	5 s	measured
Bundle	1 s	measured
D1, Turnstile, upload, secret, repo create and push	30–90 s	estimate; repo create and push measured at 5.2 s

The critical path is max(harvest + writing, images) plus build and deploy: $165 + 5 + 1 + (30 \text{ to } 90) = \mathbf{200 \text{ to } 260 \text{ seconds}}$. It fits, with little margin.

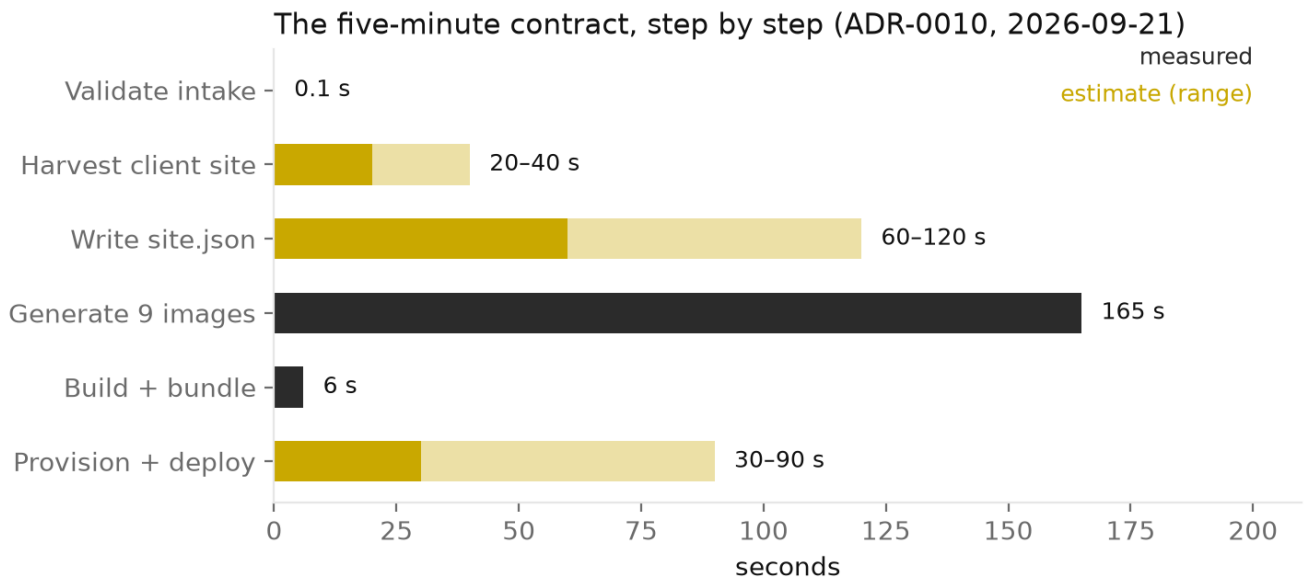


Figure 1. The five-minute budget. Dark bars are measured; gold bars are estimates with their range. Image generation dominates and runs in parallel with harvesting and writing.

The image numbers deserve their own line because they were the decision. Measured on 2026-09-21, per process including model load, on a local image model: 1376×768 at 9 steps took 43.1 s; 1024×576 at 9 steps took 25.8 s; 1024×576 at 5 steps took 17.9 s; 1024×576 at 4 steps quantized took 15.2 s. All four were judged usable by eye. So the hero gets the 43-second setting and every other image gets the 15-second one, and the slot count drops from 13 to 9 when generating, which is the schema minimum.

The deploy path is separately measured at 46.5 s for a full-depth build: database 2.2 s, bot-check widget 0.3 s, build 1.3 s, repository from template 17.6 s, deploy 11.2 s, verify 13.8 s. The first-ever deploy under a new subdomain waited one to two minutes for a TLS certificate; a second deploy under the same subdomain was reachable in seconds. That is a platform cost paid once per account, not per site.

4. The content problem

The first site launched from the kit was destroyed the same day. Measured against a reference site in the same trade, it had 21% of the reference's content: 1,613 words against 7,722. Service pages were at 17%. Structured data was thinner and there were no AI-readiness files. The fix belongs in the scaffold, so the launched site was destroyed rather than patched, and every future site benefits.

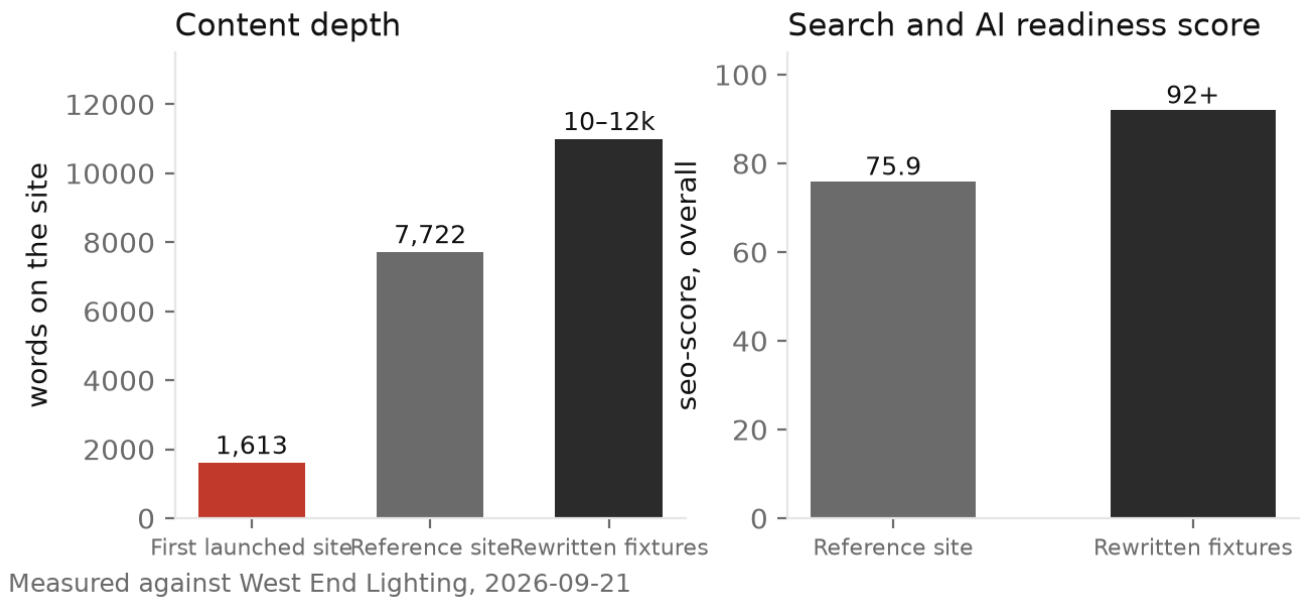


Figure 2. Parity-plus, measured on 2026-09-21. The first launch was at 21% of the reference's word count. The rewritten fixtures carry 10,000 to 12,000 words, 45 to 66 validated internal prose links, and score 92 or more on the same crawler that scores the reference 75.9.

"Parity-plus" became a contract, ADR-0011. Depth targets per page type live in code: home 1,500 words; services index 500; each service 650; FAQ 1,500 with at least 16 questions plus 4 per service; about 500; contact 300; at least three guides of 700 or more; privacy 900 or more; total 9,000. A checker prints the table and fails below target, and the scaffold refuses a live deploy that misses it. The rendered truth is a crawler that scores the built site on six categories and compares it against any reference URL. Below-parity metrics are findings fixed in the scaffold, never waived per site. Where the reference has stub pages, the kit omits the page instead of shipping a stub.

Two things follow, and both are costs. About 8,000 words of new content per site is more than the five-minute budget has room for today, and ADR-0011 says so plainly: the real timing of a full-depth run is still to be measured. And more content raises the risk of confident, wrong statements, which is where the next section comes from.

5. The rule that does not bend

A generated lead-generation site is tempting to fill with invented reviews, ratings, licence numbers, awards, hours, addresses and "projects in your suburb". That is deceptive to the visitor and it invites structured-data penalties. ADR-0002, accepted 2026-09-20, puts the rule in the schema rather than in a style guide.

- Reviews and credentials exist only with `provenance: "client-supplied"` and a named source. The reviews page renders only when they exist.

- Unsupplied facts (hours, address, coordinates, price range, founding year) are omitted from the page and from the JSON-LD, never inferred. Area served is emitted only when the client confirmed it.
- A gallery of generated images is labelled illustrative and may not name a place.
- Copy may not state specific warranty lengths, insurance or licence claims. It says the terms are provided in the written estimate.
- Palettes must meet WCAG AA text contrast, and the check is in the fixture validator.

The consequence is stated in the record and worth repeating: **sites launch honest but thinner until the client supplies real photos, reviews and credentials**. Depth to reach parity comes from general educational content and process, never from invented facts. A copy linter enforces the vocabulary. Tests fail if a rule regresses.

Harvesting the client's own website is where the facts are supposed to come from, and it carries its own rule: it needs the owner's confirmation that the client has agreed, and harvested facts count as client-supplied only then.

6. The gates

A kit that builds many sites needs the checks to live in the kit, and a gate failing on one site is a kit finding, not a per-site waiver. The gates, in the order they run:

Gate	What it refuses
Schema and copy lint	Any trust content without provenance; any specific warranty, licence or insurance claim; palette below AA contrast
Content depth	Any page type below its word target; fewer than 16 FAQ entries; fewer than 3 guides
Prose links	An internal link to a route that does not exist
Accessibility (axe)	Contrast, labelling and landmark failures. A 0.7-opacity index label failed on the editorial palette during the first client build; the fix and a regression test went into the kit the same day
Overflow, no-JS, reduced motion, focus ring	Horizontal scroll at any width; a page that needs JavaScript to read; motion without a reduced-motion fallback; a focus ring the design hides
JS budget and CLS	Bundle over budget; layout shift. A real shift was fixed by switching to <code>font-display: optional</code> with preloads
CTA above the fold	A home page whose call to action is below the first viewport
Visitor-first UI (ADR-0012)	Bands, imagery, ambient light, disclosure nav and read-more: the DOM keeps full depth for crawlers while the visitor sees about 900 words with the rest behind a disclosure
Local end-to-end	The real edge worker against a local database: routes, lead validation, honeypot and stored rows, in about 3.5 s, with no account
Verify (live)	Every sitemap URL serving 200; no internal link redirecting; a tokenless lead blocked. About 14 s, before the scaffold reports LIVE

7. What the first live deploys got wrong

The gates in the last two rows exist because of three things found on real deploys on 2026-09-21. None of them was visible in the local build.

Every internal link took a 307

Directory-style output made the platform redirect every internal link, and the canonical tag pointed at a URL that redirected. The fix was slashless URLs (`about.html` with `trailingSlash: never`), and the canonical, breadcrumbs and sitemap now agree. A test and the local end-to-end run, which replays the platform's routing, both fail if any internal link would redirect. **A redirect on every internal link is a crawl-budget tax on every page you own, paid on every visit, and it does not show up in a screenshot.**

Pages returned 404 briefly while the home page was up

Assets reach the edge seconds apart. The scaffold used to report success when the deploy command returned. Now a verify step polls every sitemap URL until it serves 200.

The lead endpoint accepted unverified leads for a window

The bot-check secret was set in a separate step after the code deployed, which left a window where the worker accepted leads with no verification. Three changes closed it. The worker now refuses leads without a secret (a local-only flag is the exception, and it is off in production). The secret ships in the same worker version as the code, through a secrets file with mode 600 that is removed after. And the verify step proves a tokenless lead is blocked before the scaffold says LIVE. The general rule is one this site has written about before in another context: **a control that depends on a second step is a control with a hole exactly the width of that step.**

And one on the way down

The first real teardown printed ok for a bot-check widget that was still there, because it judged the deletion by the shell exit code instead of the API's answer. Deletes now go through the API, count only an explicit success or a 404 as done, and surface the API's message on failure.

8. The two client sites

Two client sites were generated from the template, on 2026-09-21 and 2026-09-22. Each is a private repository created from the template with the client's data file and image pack committed, and its own edge worker and leads database. Client data reaches the client repository only, never the template; the scaffold copies an explicit allowlist into the clone and patches the worker configuration in place for the run, restoring it on exit so a client's database id never lands in the template's tracked file.

The second site's build surfaced the contrast failure in section 6. That is the economics working as designed: the finding went into the kit, with a regression test, and the first site gets the fix on its next build.

9. What is not proven

Three things, stated so the reader can hold the paper to them.

1. The five-minute contract is measured for the deploy path (under one minute) and for image generation (165 s). It is not yet measured for a full run that writes 10,000 words for a new business. The ADR records the levers if it does not fit: fewer generated images, harvested photos in place of generated ones, and keeping the image model loaded in one process to save 5 to 8 s per image (not measured).
2. Lead volume from the generated sites is not yet reported. Both client sites are days old. The leads table exists and the lead endpoint is verified; the count is the next paper's job.
3. The template is private by decision, so the code cannot be inspected by the reader. The decision records and the measurements quoted here are the checkable artifacts, and the two generated sites

are live.

10. What to copy

- **Put the rule in the schema, not in the style guide.** A field that requires provenance cannot be fabricated by a tired generator at 11pm.
- **Measure the steps you own and label the ones you estimate.** The contract stays honest even while it is unproven.
- **Verify live, not on exit code.** Poll every URL, replay the platform's routing locally, and prove the negative case (a blocked lead, a deleted widget) before reporting success.
- **Destroy the first launch.** Fix the kit instead, so the second and every later site inherit the fix.