DESIGNME

FREE GUIDE · NOVEMBER 2026

A design system Claude can actually use

Tokens, components and written rules that keep Claude's pages on brand.



Adrian Kuleszo
Co-founder, DesignMe

designme.agency

INSIDE

Seven chapters

01 **Why Claude’s pages drift**

What a model falls back on without your system, and why small mistakes add up.

02 **Tokens with names an agent understands**

How 191 colours became one file, the two tiers of tokens, and the names that matter.

03 **One component per job**

Why five card styles become six, and the rule that stopped it on our site.

04 **Rules, written where Claude reads first**

What goes in CLAUDE.md or AGENTS.md, what stays out, and real lines from ours.

05 **A way in: files and MCP servers**

How Claude reaches your tokens and components, from the repo to Figma and Storybook.

06 **Checks, then a person**

Why rules in a file aren’t enough, and the review loop that keeps every change honest.

07 **Is your system ready? The self-check**

The guide in five lines, seven questions to score your system, and how we close the gaps.

SOUND FAMILIAR?

Every page looks generic

Ask for a pricing page and you get a pricing page. Just not one that looks like the rest of your product.

A sixth card style appears

There are five in the codebase already, so the agent picks one at random or writes a new one.

Prompts keep growing

Every request repeats the same colours, spacing and “please match the other pages”, and it still drifts.

Same fixes, every review

Wrong blue, wrong radius, a shadow nobody asked for. Fixed by hand, again, on every pull request.

CHAPTER 01

Why Claude's pages drift

Here's the thing nobody says out loud: Claude is very good at building a page. It's just building the page it has seen most often.

Without context, a model falls back on the average. Anthropic says as much in its own guidance on frontend design: Claude tends to converge toward generic output unless it's steered. Vercel's v0 is trained on default shadcn/ui, and its docs point to a design system registry as the way to make output match your brand.

There's a maths reason too. Small error rates add up across steps. An agent that's right 90% of the time per step is right 59% of the time after five steps.

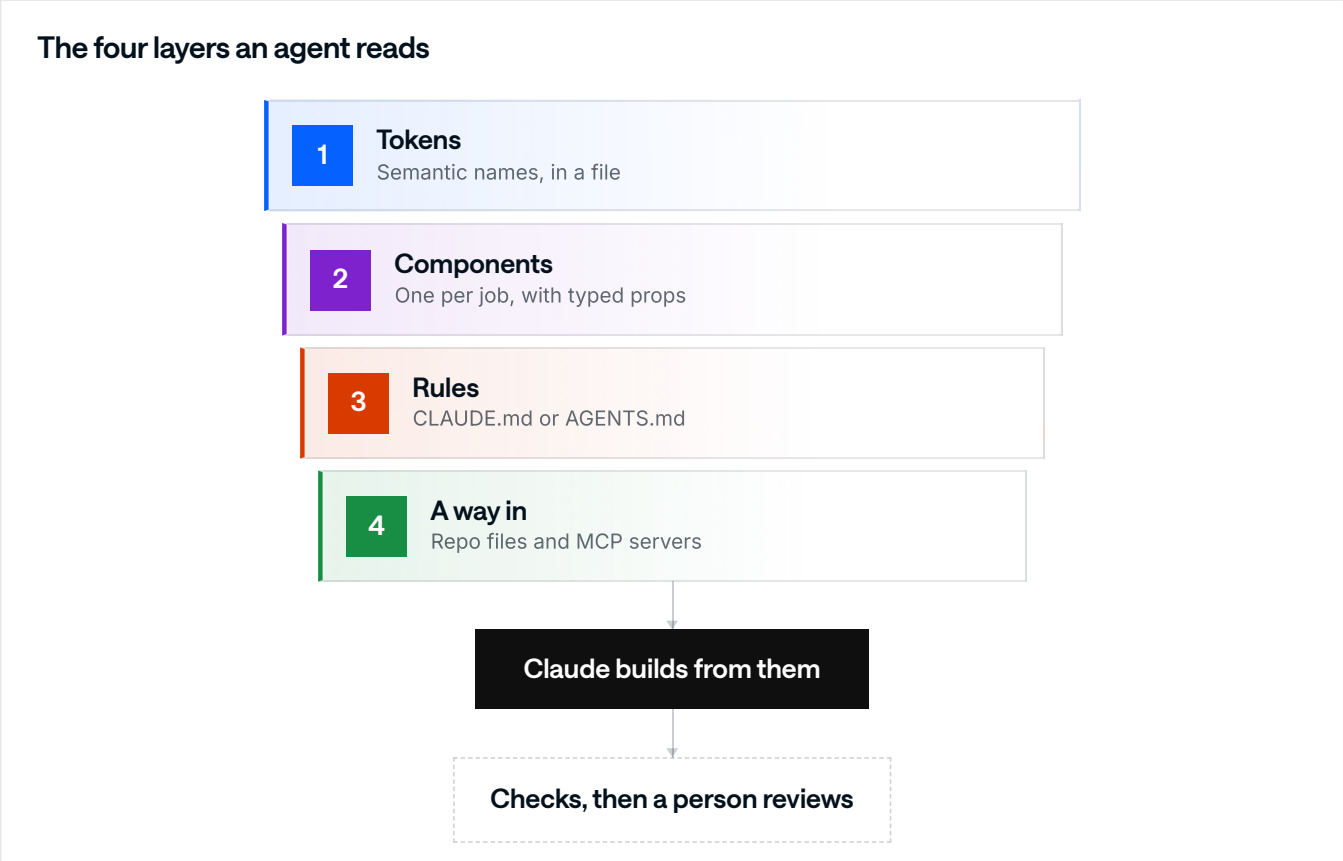
90%

right on each step

59%

right after five steps in a row

So every decision you take out of the agent's hands (which blue, which spacing, which card) is a step it can't get wrong. That's all a design system is, for an agent: decisions made once, written where it can read them.



CHAPTER 02

Tokens with names an agent understands

Our own site ran on Framer for three years. When we moved it to code in October 2026, we counted what had piled up one section at a time: 191 colour values, 37 font sizes and 53 font families. No agent can stay on brand in that. Neither can a person.

They all collapsed into one tokens file, and that file is the first thing Claude builds from.

Design tokens are the named values behind the interface: colours, type sizes, spacing, radii, motion. For an agent, the name is everything.

A token called `--color-text` tells Claude what it's for. A token called `--blue-500` only tells it what it looks like, and it still has to guess where it belongs.

Use two tiers

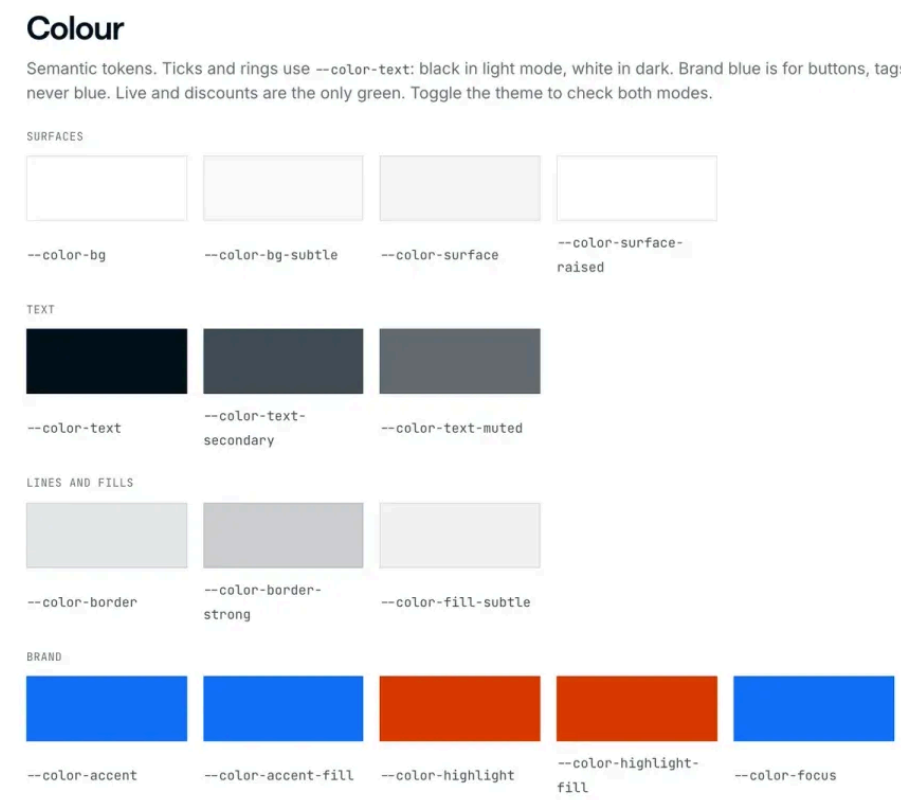
- **Primitives** hold the raw palette and scale. They exist only to define the next tier.
- **Semantic tokens** say what each value is for (text, surface, border, accent) and point at the primitives. Components use these, and only these.

Here's an excerpt from ours. Every colour is written once with `light-dark()`, so light and dark themes work without the agent thinking about them:

src/styles/tokens.css (excerpt)

```
/* Primitives */
--blue-500: #0561ff;
--navy-950: #03111c;

/* Semantic: what each value is for */
--color-text: light-dark(var(--navy-950), var(--paper));
--color-border: light-dark(rgb(3 17 28 / 0.1), rgb(245 245 245 / 0.12));
--color-accent: light-dark(var(--blue-500), var(--blue-400));
```



Colour tokens designme.agency

Our semantic colour tokens on the site's styleguide page. Claude picks a colour by its job, never by hex.

Put them in a file machines can read

Tokens that live only in Figma styles are invisible to a coding agent. Keep them in code (CSS variables work fine), and if they need to feed design tools, iOS and Android too, use the W3C Design Tokens Community Group format. Its first stable version, 2025.10, came out on 28 October 2025: each token is a JSON object with a `$value` and an optional `$type`.

> Name tokens by job, not by look. If you can't say what a token is for, the agent can't either.

CHAPTER 03

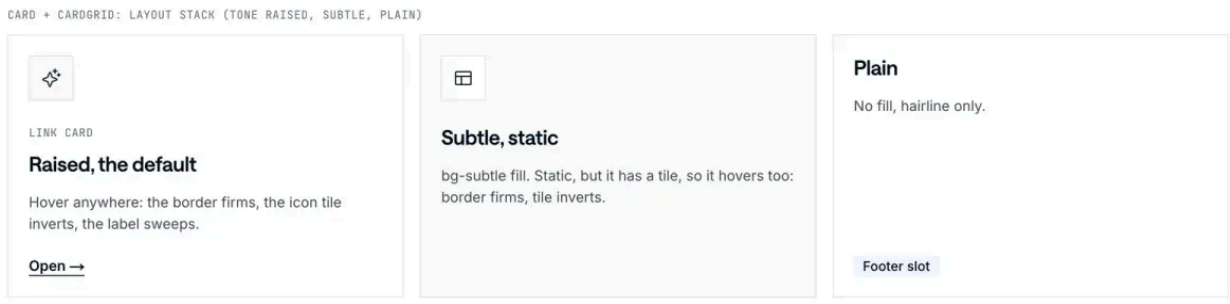
One component per job

For anything an agent ships, the components in your repository are the source of truth. Claude assembles pages from what's there, so what's there has to be right.

Keep exactly one component per job. If your codebase has five card styles, the agent picks one at random, or writes a sixth. (It'll write a sixth.)

When we built our component library, my feedback to the team was blunt: "We don't want to have 50 different styles for cards or 50 different styles for lists." So the site has one primitive for each job, 23 of them in all:

THE JOB	THE ONE COMPONENT
Any card or tile	`Card` inside a `CardGrid`
Rows and lists	`List` and `ListRow`
Every chip, badge, status or filter	`Tag`
A bordered visual or a table	`Panel`, with `DataTable`
A big number	`Stat`
Eyebrow, heading and sub	`SectionHead`



Card tones designme.agency

One Card primitive with three tones on our styleguide, so Claude picks a variant instead of inventing a card.

Extend, never invent

The rule that does the most work is the plainest one: no new card, list or chip styles. If a primitive can't do what a page needs, Claude adds a prop or a variant to the primitive and shows it on the styleguide page. It never writes one-off CSS on the page.

Then give each component what an agent needs to choose it: a clear name, typed props, its variants and one line on when to use it.

CHAPTER 04

Rules, written where Claude reads first

Tokens and components say what exists. Rules say when to use what, and they're the layer most teams skip.

Claude Code reads a CLAUDE.md file in the project at the start of every session. AGENTS.md is the open equivalent for other agents, and Claude Code can read that too. Anthropic's docs suggest keeping each CLAUDE.md under 200 lines, so keep the file short and link out for the long version.

Ours is 100 lines. Here are three real ones from its design system part:

CLAUDE.md (three lines from ours)

```
- Use semantic tokens (`--color-text`, `--text-2xl`, `--space-6`) in components.
  Never hard-code a hex value, px font size or Framer value.
- Radius: nothing visible above 4px.
- Don't add new card, list or chip styles. Extend the primitive
  (a prop or variant), then add it to `/styleguide`.
```

What good rules cover

- **Tokens.** Semantic tokens only. Never a raw hex value, pixel font size or one-off shadow.
- **Components.** Which one to use for each job.
- **What never to create.** New card, list or chip styles, gradients behind heroes, corners above the radius scale.
- **Content.** Tone, spelling, banned punctuation, how headings are phrased.
- **How to check work.** Build it, both themes, a phone width, contrast.

The full reasoning lives in a separate `docs/design-system.md`, with the component inventory and the decisions behind it. CLAUDE.md points there.

Decisions go in the file, not just the page

Here's the habit that matters most. When someone makes a call in a review ("no drop shadows on cards"), it goes into the rules with a date, the same day. Several of ours read like that. Once it's written down, every later change follows it, whoever asks for it, and nobody has to say it twice.

When a page drifts off brand on our site, the gap is usually in the rules, not the model. The fix is a line in CLAUDE.md, not a longer prompt.

CHAPTER 05

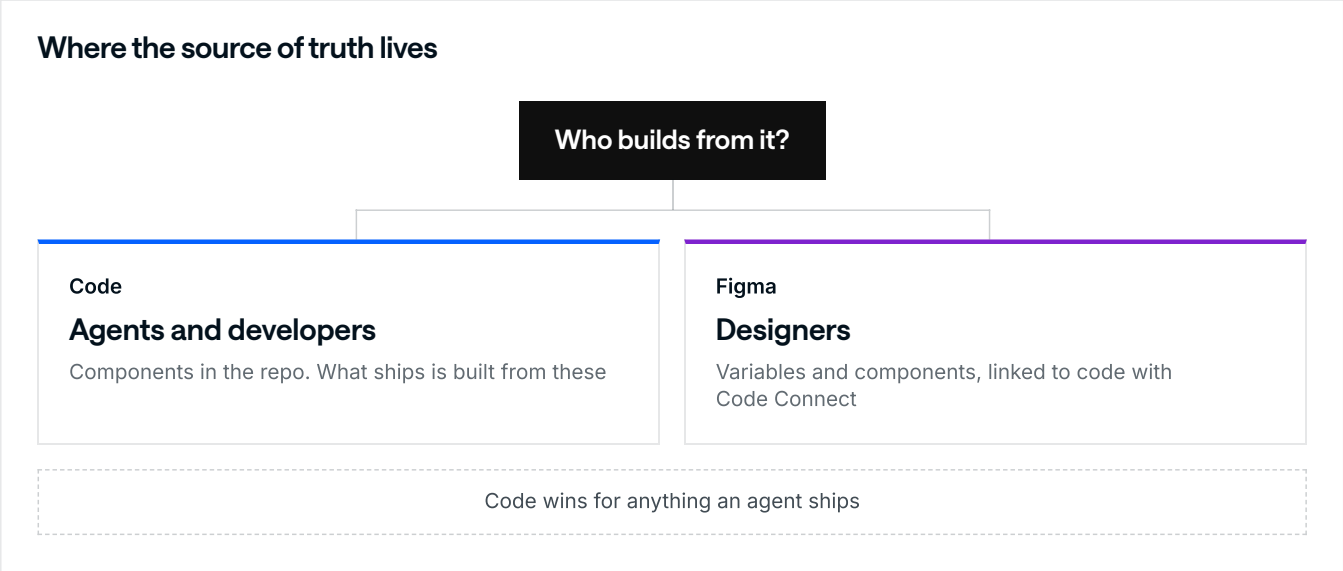
A way in: files and MCP servers

An agent can only use what it can reach. There are two ways in.

Files in the repository. The tokens, the components and the rules, as plain files. This is the way that works everywhere, and it's the one our site runs on.

MCP servers, the protocol Anthropic open-sourced in November 2024. The ones that matter for a design system:

- **Figma.** Figma's MCP server (launched in June 2025 as the Dev Mode MCP server) passes variables, components and layout to coding agents. With Code Connect linking each Figma component to its code component, it can point the agent at the exact file to use.
- **Storybook.** Storybook 10.3 added an MCP server for React that lets agents look up your components, build from them and run component and accessibility tests. Storybook marks it as a preview.
- **Component registries.** The shadcn CLI 3.0 and its MCP server let agents browse, search and install components from a registry in plain language.



Keep both in sync, but don't argue about it: for anything an agent ships, code is the source of truth.

Want us to look at your system?

Bring your repo or your Figma file to a 30-minute call. You'll leave knowing what Claude can and can't read in it today, and what to fix first.

Book a 30-minute call

CHAPTER 06

Checks, then a person

Rules in a file are context, not enforcement. Anthropic's docs say it plainly: Claude treats CLAUDE.md as context, not as enforced configuration.

So anything that must never happen needs a check in code too: a lint rule that rejects raw hex values, contrast and accessibility checks on every pull request, and a styleguide page that shows every token and component in both themes, to compare against.

How a change ships on our site

01 **A branch**
Each task gets its own branch. Claude reads CLAUDE.md first.

02 **Built from the system**
Tokens and primitives only. A missing piece becomes a variant.

03 **A preview**
The branch is built and deployed to a private, noindex address.

04 **A person checks it**
Both themes, a phone width, next to the styleguide page.

05 **Then it ships**
Only after review, and only when the owner deploys.

The system makes Claude's output consistent. It doesn't make it right. That's why the last step is always a person, and I'd keep it even if the agent never drifted again.

The first drafts are fast. The review is where the time goes, and it's where the quality comes from.

CHAPTER 07

Is your system ready? The self-check

Here’s the whole guide in five lines, before you score yourself:

1. **Tokens** with names that say what they’re for, in a file an agent can read.
2. **One component** per job, with typed props and a line on when to use it.
3. **Rules** in CLAUDE.md or AGENTS.md: short, specific, and dated when they change.
4. **A way in** through the repository, plus MCP servers for Figma or your library.
5. **Checks in code**, then a person who reviews every preview before it ships.

Now go through the checklist with your team. Be honest, nobody sees the score but you.

Is your design system agent-ready?

Seven questions. Five or more ticks and Claude can build on it today.

TOKENS

- [ ] Tokens have semantic names, and components use only those
- [ ] Tokens live in a file an agent can read, not only in Figma styles

COMPONENTS

- [ ] There's exactly one component for each job
- [ ] Every component has typed props and a line on when to use it

RULES AND ACCESS

- [ ] The rules are written in a file agents read at the start of a session
- [ ] Agents can reach Figma or your component library through MCP

CHECKS

- [ ] Automated checks catch raw values, contrast failures and broken pages before review

Fewer than five? The gaps are usually quicker to close than another round of fixing generic pages by hand.

How we build them

We design systems in Figma and in code together, so your designers and your agents build from the same parts. It's part of our product design work:

$10,000
the starting price for a focused UX and UI cleanup, design system included

$25,000
the starting price for a full product, design system included

8 weeks
is about how long a full product takes, from the audit to handoff

For a marketing site, our Astro builds ship with the tokens, the components and the agent docs from day one, the same way ours does.

Want to know where your system stands? Book a 30-minute call and bring the repo or the Figma file. You'll leave knowing what Claude can read today and what to fix first.

RATHER TALK IT THROUGH?

Want your system ready for agents?

Bring your repo or Figma file to a 30-minute call with Adrian. You'll leave knowing what Claude can read in it today and what to fix first.

Book a 30-minute call at designme.agency/intro

01

What Claude can read

Your tokens, components and rules, as an agent sees them today.

02

What to fix first

The two or three gaps that cause most of the drift.

03

What it costs

A fixed price from $10,000, confirmed before anything starts.

DESIGNME

DesignMe designs and builds websites for funded startups and B2B teams, in Framer, Webflow, Astro and Next.js.