IT SUMMIT

One Team: Powering Technology Together

University of Arizona
Student Union Memorial Center Ballroom

Code as Design, Design as Code

Next-Gen Workflows

Presented by:
Mario Vasquez – Experience Designer · UITS:Trellis

Tuesday, November 10, 2026

Exploring the next CatCloud

CatCloud

What is CatCloud

A digital hub for students and staff

Team exploring next CatCloud

Me today + one developer soon

Moving to Next.js and APIs

Standard web tools that developers, and AI, already know.

Today: a hosted platform

Salesforce Experience Cloud. The platform controls the frontend and its tools.



Future: Next.js and APIs

Our own frontend, with data from API integrations.

The CatCloud Design System

Figma components, coded components, design tokens and guidelines for university apps.

Figma components

Designed in Figma

Design tokens

U of A colors, type and spacing

Guidelines

Usage, accessibility and specs

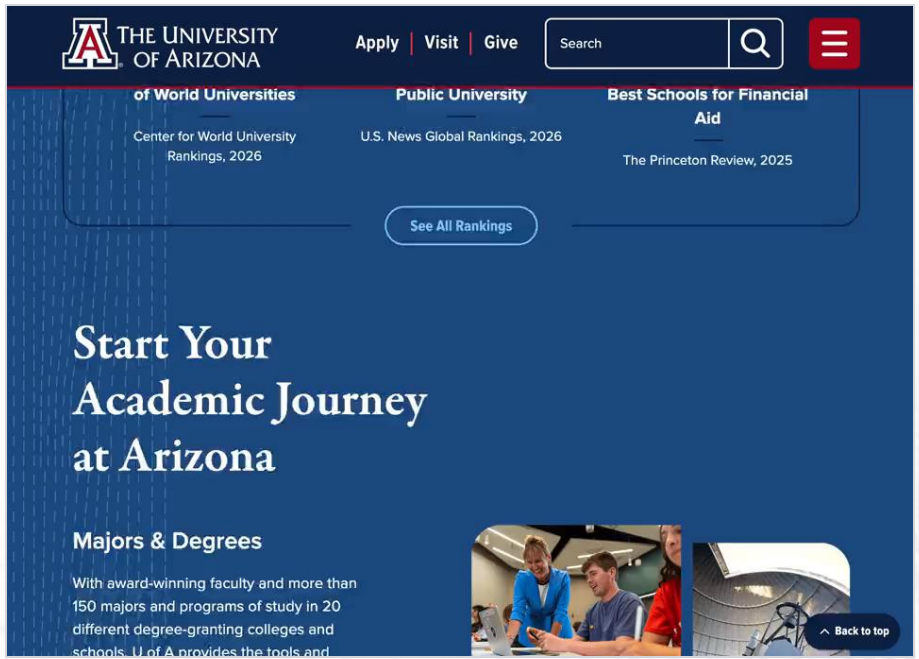
Coded components

Web components for any framework

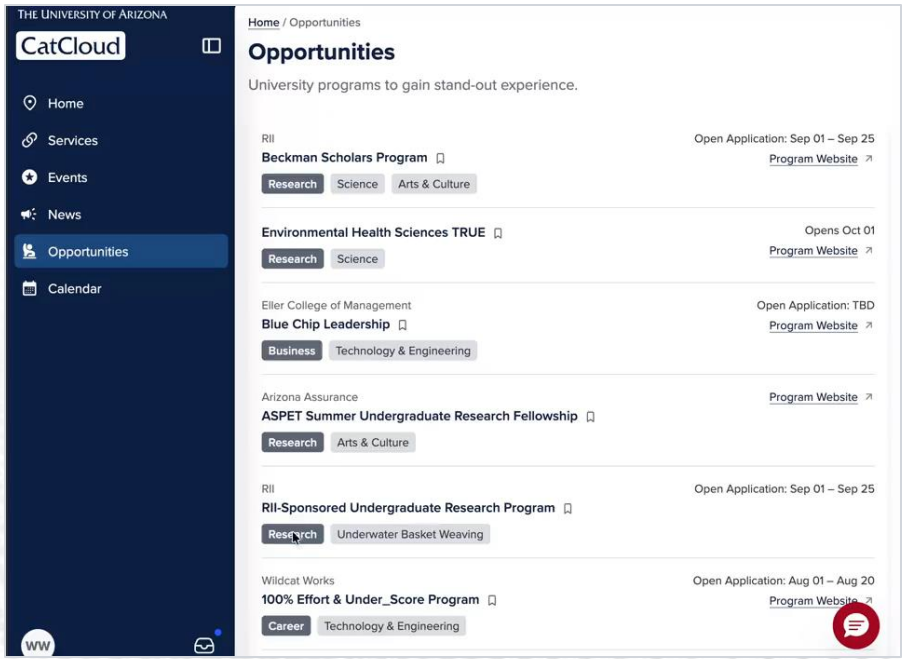
Apps are not websites

Websites bring people in. Apps help people get things done.

Website: browse and read

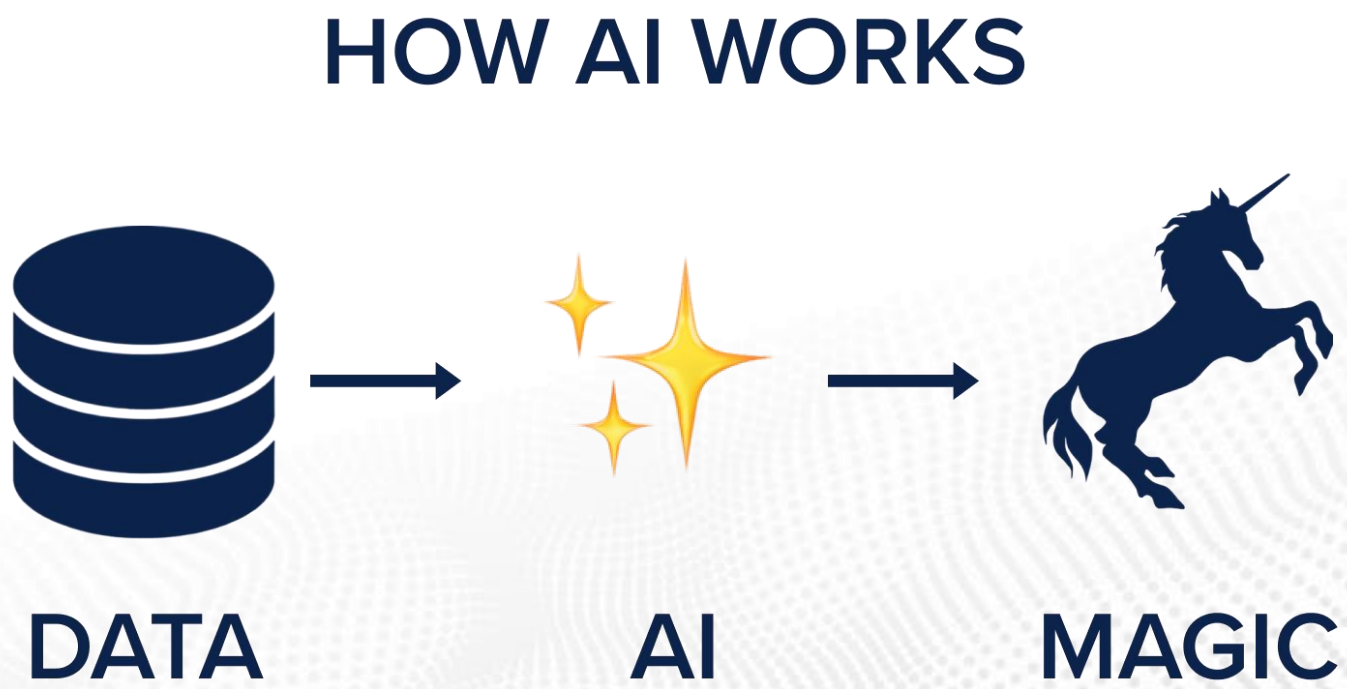


App: find and act



AI changes what users expect

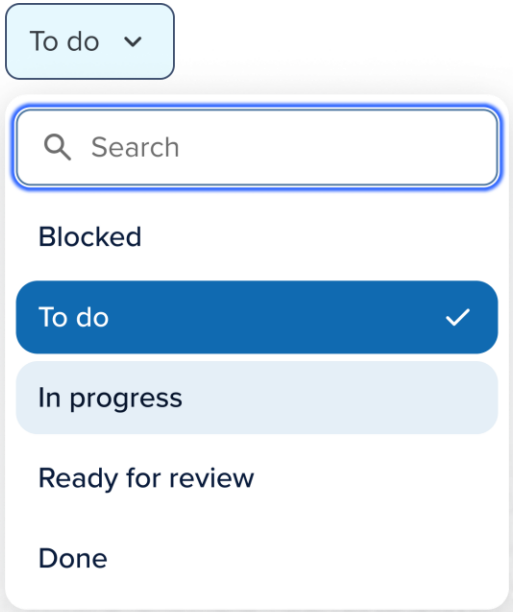
Less browsing. More speed. Suggestions, not searches.



Decisions I make

Component Props, Design tokens, and Guidelines.

Component



Design tokens

`--cc-color-bg-option-selected`
#106AB1

`--cc-color-bg-option-hover`
#E5EFF7

`--cc-color-border-input`
#6C757D

`--cc-radius-md`
8px

`--cc-spacing-100`
8px

`--cc-size-width-popover-option-max`
320px

`--cc-motion-duration-xs`
0.15s

Guidelines

When to use something else

The overlay shows content, a heading or controls beyond a simple list. Use a **Popup**.

The action needs confirmation or more input. Use a **Modal**.

Only 1–2 high-priority actions that should always be visible. Use inline **Buttons**.

Picking an answer in a form. Use **Select, Radio Group or Checkbox Group**.

2–4 options with short labels that should be visible right away. Use a **Filter Group**.

AI is fast, but it drifts

Southleft test: the same AI coded the same five screens with the same components. Only the contracts changed.

Drift at AI speed

Design system, no contracts	Design system + contracts
69 / 100	**100** / 100
90 of 283 checks failed	0 of 253 checks failed

The 90 failures: invented options · hard-coded colors · restyled components

Source of truth

The JSON tells design tools and code how a component is built.

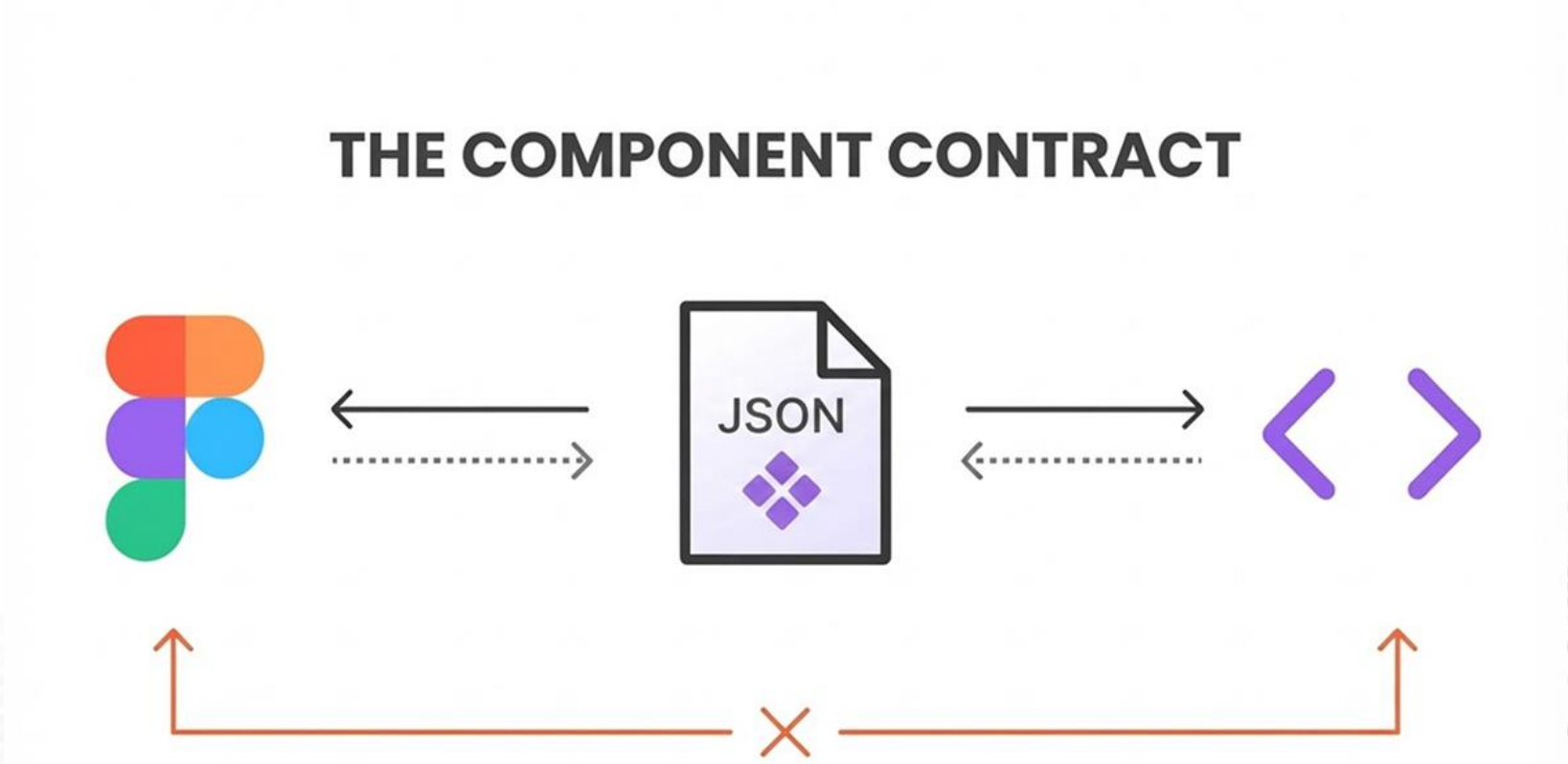


Diagram: Christine Vallaure, moonlearning
christinevallaure.substack.com/p/design-system-contracts-the-component

One contract, two places

Figma, the spec and the code share the same decisions.

From design to release

I decide when the design is ready. The checks decide when it ships.

DESIGN — Figma and Notion

1 Design the component
Use the shared colors, spacing and type, never one-off values.

2 Label each part
Say what each part is, so screen readers can announce it.

3 Write the guidelines
When to use it, and what to use instead.

→

Mark it "Ready for dev"
The design is done. Code can't start before this.

The designer decides when.

→

BUILD — design-system

4 Claude builds the code
It reads the design, the guidelines and our written rules.

5 Claude checks its own work
An accessibility scan runs before anyone looks at it.

6 A person tests it
Keyboard only, then a screen reader. Tools can't do this part.

7 Log the change
Record what changed. Figma shows the component as done.

→

AUTOMATIC CHECKS — every change

They run on every change, and again before each release.

- **Uses only shared styles**
  No hard-coded colors, sizes or timing.
- **Docs match the design**
  Docs and specs are up to date with Figma and the code.
- **Every example works**
  Clicks and keys behave, and each example passes an accessibility scan (WCAG 2.2 AA).

⊗ **One failure stops the release.**

→

RELEASE — catcloud-kit

8 Release the kit
Give it a version number and release notes. The checks run one more time, then apps can install the new version.

○ Automated or done by Claude ◌ Done by a person

From design to release

I decide when the design is ready. The checks decide when it ships.

DESIGN — Figma and Notion

1 Design the component
Use the shared colors, spacing and type, never one-off values.

2 Label each part
Say what each part is, so screen readers can announce it.

3 Write the guidelines
When to use it, and what to use instead.

→

Mark it "Ready for dev"
The design is done. Code can't start before this.

The designer decides when.

→

BUILD — design-system

4 Claude builds the code
It reads the design, the guidelines and our written rules.

5 Claude checks its own work
An accessibility scan runs before anyone looks at it.

6 A person tests it
Keyboard only, then a screen reader. Tools can't do this part.

7 Log the change
Record what changed. Figma shows the component as done.

→

AUTOMATIC CHECKS — every change

They run on every change, and again before each release.

- **Uses only shared styles**
  No hard-coded colors, sizes or timing.
- **Docs match the design**
  Docs and specs are up to date with Figma and the code.
- **Every example works**
  Clicks and keys behave, and each example passes an accessibility scan (WCAG 2.2 AA).

⊗ **One failure stops the release.**

→

RELEASE — catcloud-kit

8 Release the kit
Give it a version number and release notes. The checks run one more time, then apps can install the new version.

○ Automated or done by Claude ◌ Done by a person

The AI follows our rules

Same question, three answers. The guidelines decide.

Request 1

A status filter above a task list, with 8 statuses.

Request 2

A filter with 3 options: Open, Closed and All.

Request 3

A Status field in the Edit task form.

Which component should be used?

How the pieces fit together

Each layer works on its own, in any framework.

1. Guidelines

Accessibility contracts, test steps and design decisions

2. Tokens and styles

UA colors, type and spacing as CSS variables

3. Components

Built, tested and kept up to date

Collaboration and data

We build with partners and app teams

Arizona Digital

One brand foundation for websites and apps

MarCom

The university brand and voice

Disability Resource Center

Accessibility requirements and review

App teams

Context, pain points and prototypes

What comes next

Design system improvements.

- **Component contracts**
- **More components**
- **Layout patterns**
- **Refinements**

Questions