Agentic AI Foundation
AGNTCon + MCPCon
Japan

Built for Two

What design systems are missing when AI agents become the third user.

Karen Ng Principal UX Designer Endor Labs

What if **Figma** isn't relevant in six months?

Documentation looks complete

Prototype came out wrong

Karen Ng カレン

Principal UX Designer | ENDOR LABS

20+ years across enterprise, healthcare, and cyber security

hp DELL Roche tenable

0→1→100

products
built

6+

design system
built

12th (?)

trip to Japan

BUTTONS

Save | Cancel | Disabled

STATUS

Active | Paused | Not active | Critical

CONTROLS

COLOR

TYPE

Heading

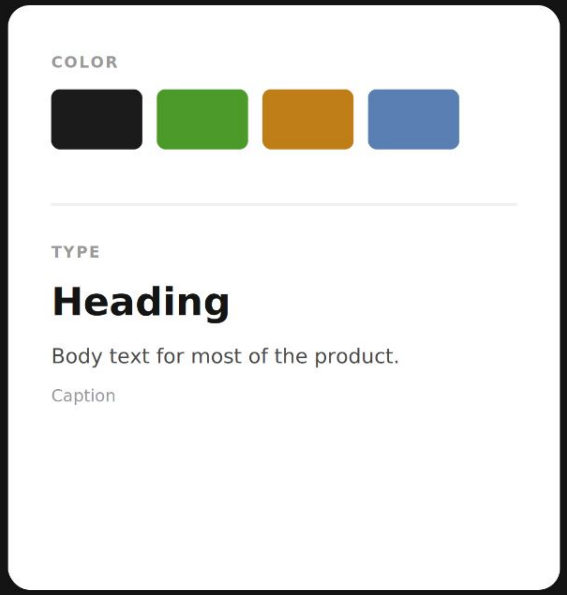
Body text, the size most of the product is set in.

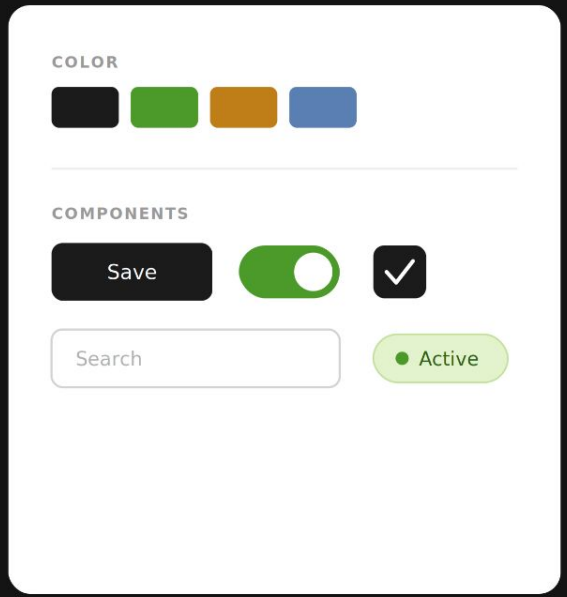
Caption

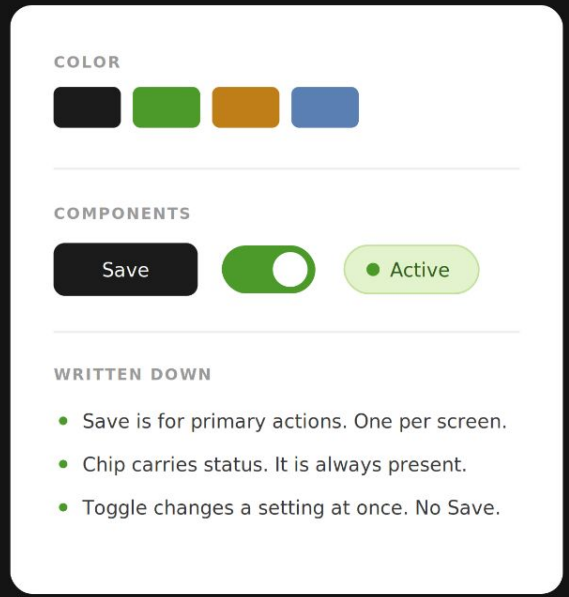
TABLE

Service	Status	Updated
web-client	Active	14h ago
invoice-service	Paused	3d ago


Agentic AI Foundation
AGNTCon + MCPCon
Japan
Karen Ng ©2026

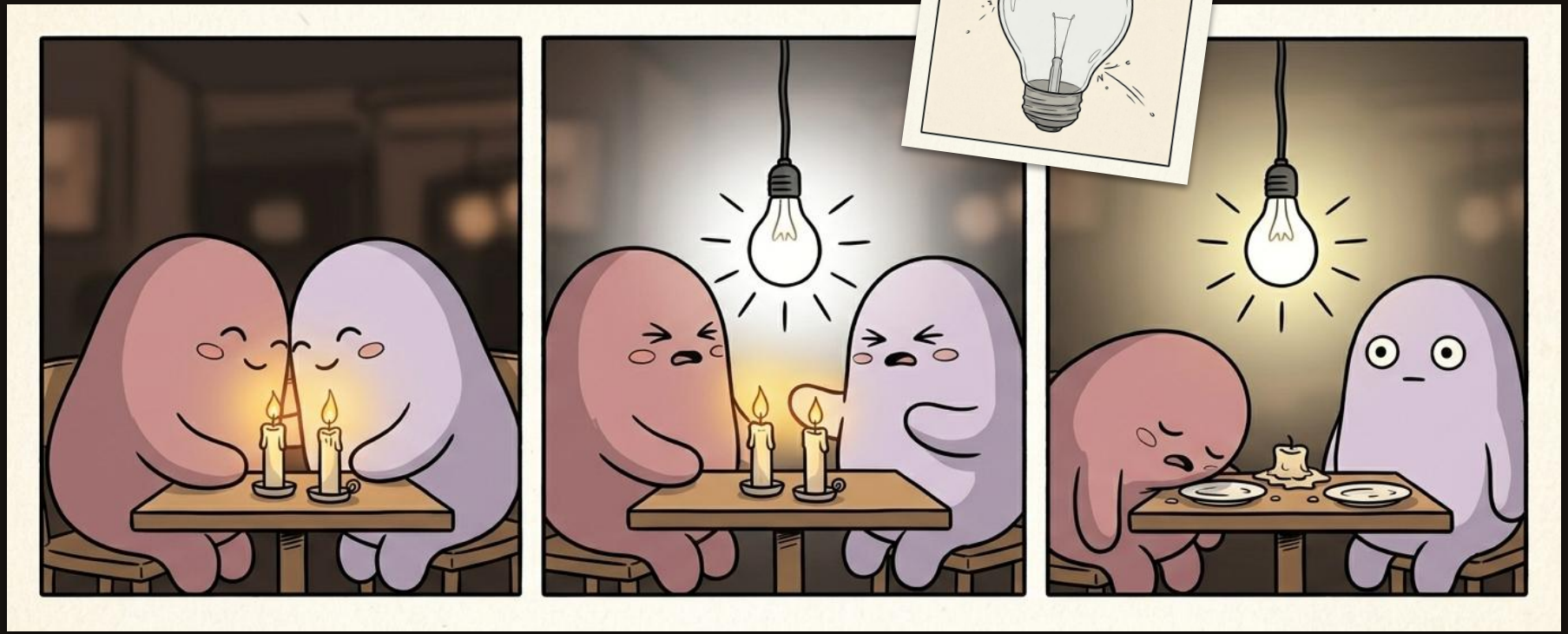




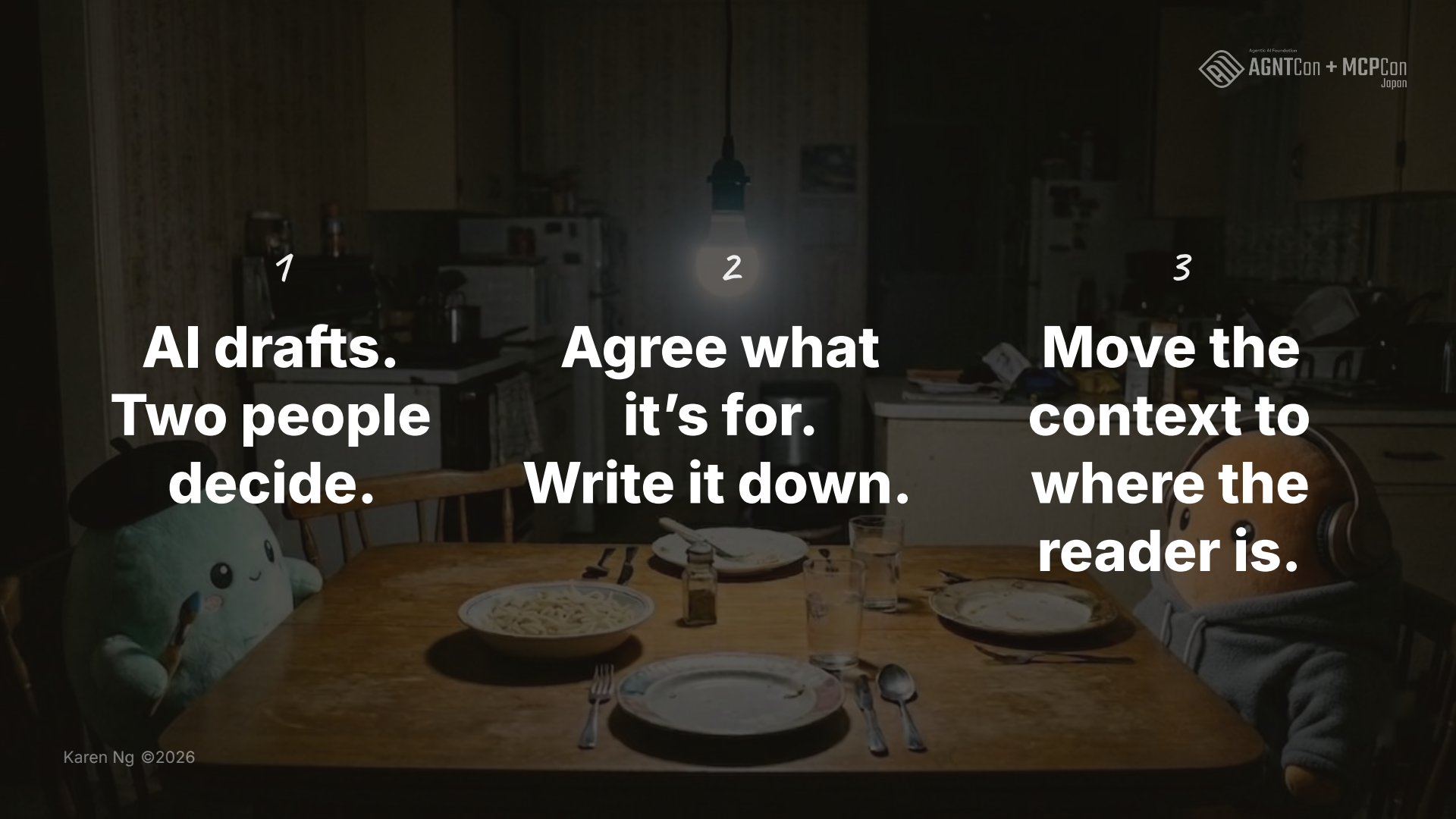


Style guide **Component library** **Design system**


Agentic AI Foundation
AGNTCon + MCPCon
Japan
Karen Ng ©2026

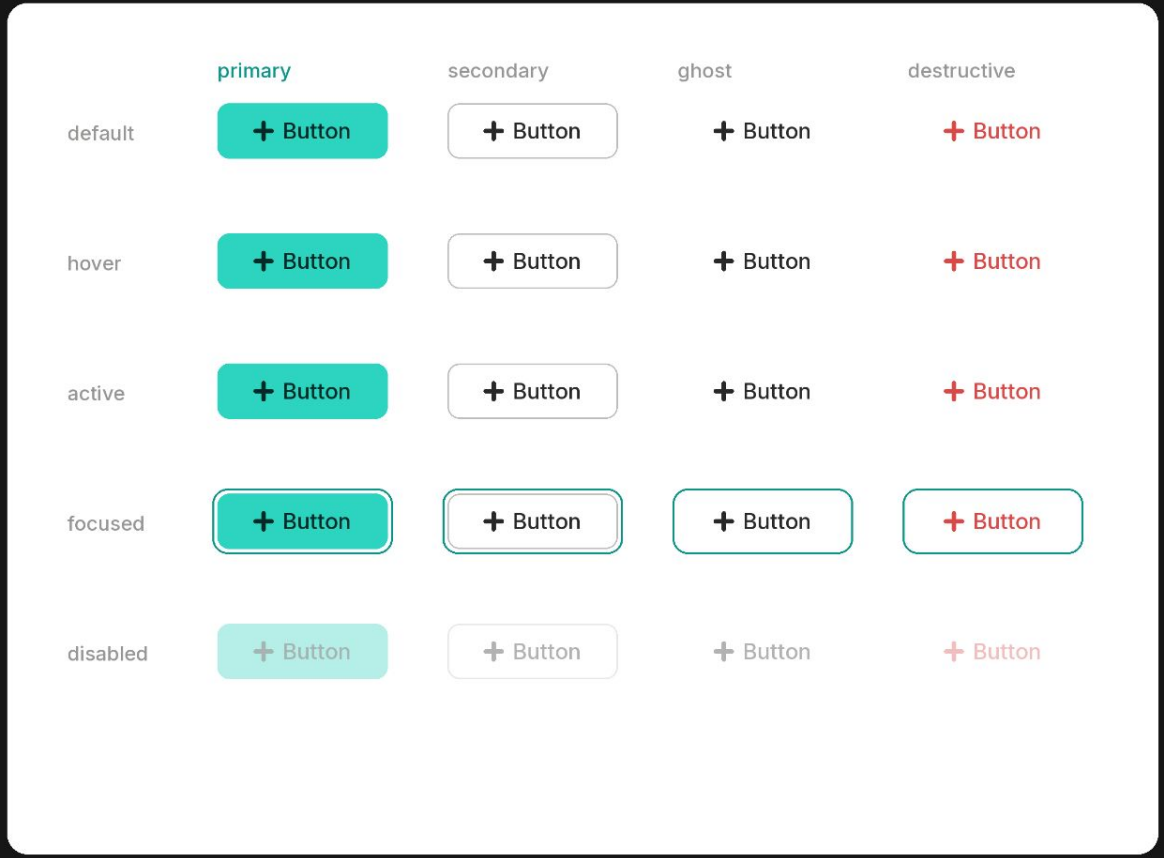


Third Reader Era


Agentic AI Foundation
AGNTCon + MCPCon
Japan
1
AI drafts.
Two people
decide.
2
Agree what
it's for.
Write it down.
3
Move the
context to
where the
reader is.
Karen Ng ©2026

An agent can draft.

It cannot **decide** .



Button

Sub-types

API Spec

Structural Spec

Typography

Color Assignment

Behavioral Spec

Disabled state

Voice / Accessibility

Usage Notes

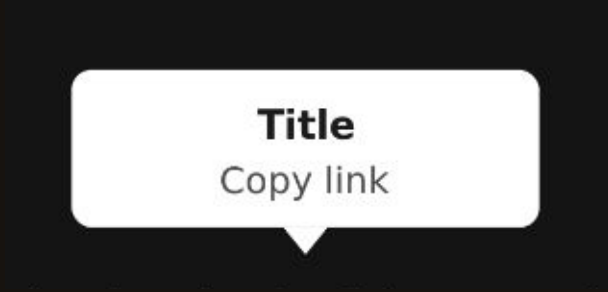
```
interface ButtonActionProps {
  type: 'primary' | 'secondary' | 'ghost' | 'destructive';
  size: 'sm' | 'md';
  prefix?: boolean;        // show leading icon. default: true
  prefixIcon?: ReactNode;  // default: icon/plus
  suffixIcon?: ReactNode;  // default: icon/plus
  text?: string;           // label. default: 'Button'
  disabled?: boolean;
}
```

What the agent wrote

label	"Button"
icon	plus

What was actually true

label	*— not set*
icon	*— not set*




```
<Tooltip title="Copy link">
  <IconButton />
</Tooltip>
```

Tooltip.md 150 lines · 5.8 KB

Tooltip

Sub-types

API Spec

Replaces Radix's **sideOffset**:

Structural Spec

tooltip.tsx shadcn/ui docs

```tsx
<Tooltip>
  <TooltipContent sideOffset={4}>
    ...
</Tooltip>
```

AI drafts.
Two people decide.

An agent can name.

It cannot **choose** .

Tag

C **Deserialization of untrusted data**

critical · fix available

DEPENDENCY	SEVERITY SCORE
`yaml-parser@5.1`	9.8

ATTRIBUTES

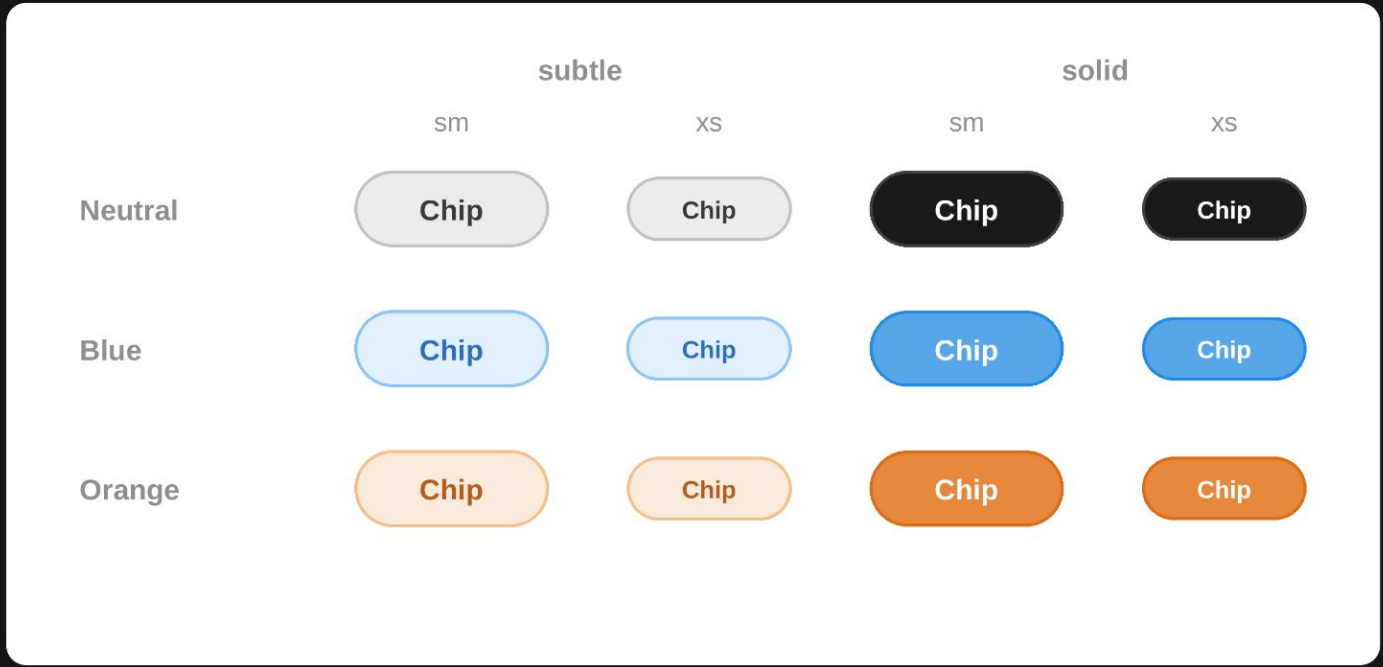
Direct · Normal · Reachable Dependency · Fix Available

CATEGORIES

Dependencies · Vulnerabilities · Security

Chip

	Service	Status	Updated
☐	web-client	● Active	14h ago
☐	api-gateway	● Active	2d ago
☐	invoice-service	● Paused	3d ago



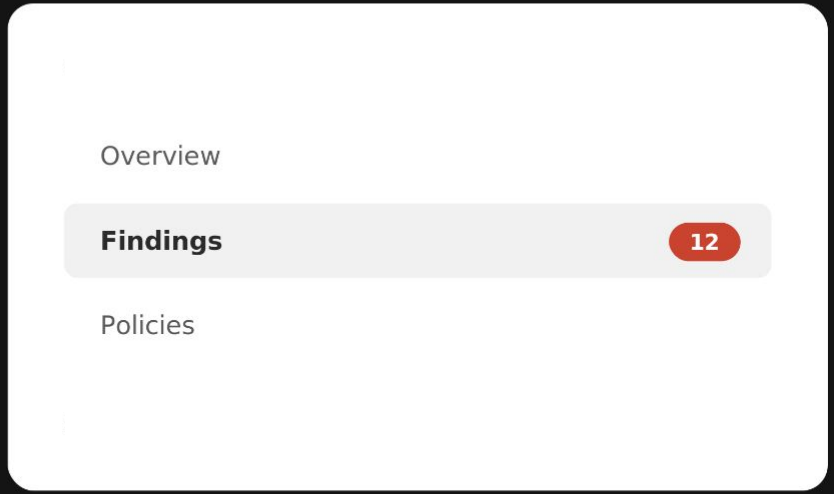
What the agent built


```
<div className="tag">
  Vulnerabilities
</div>
```

What should have been rendered

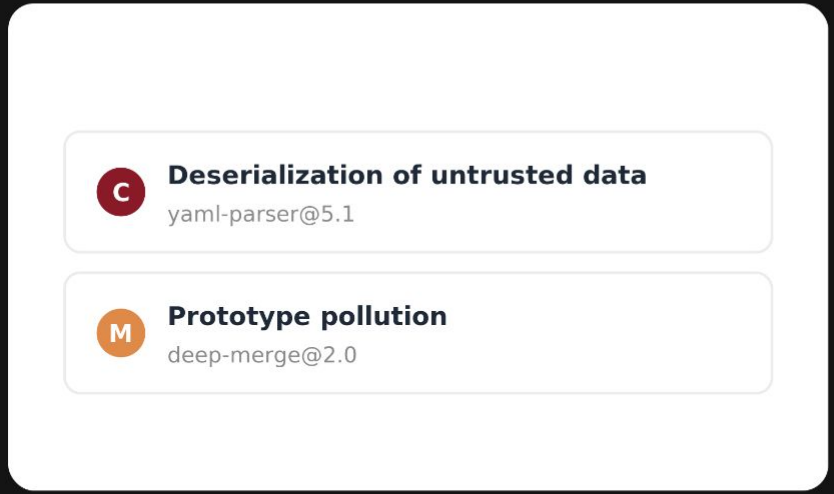

```
<Chip label="Vulnerabilities" />
```

Counter Badge

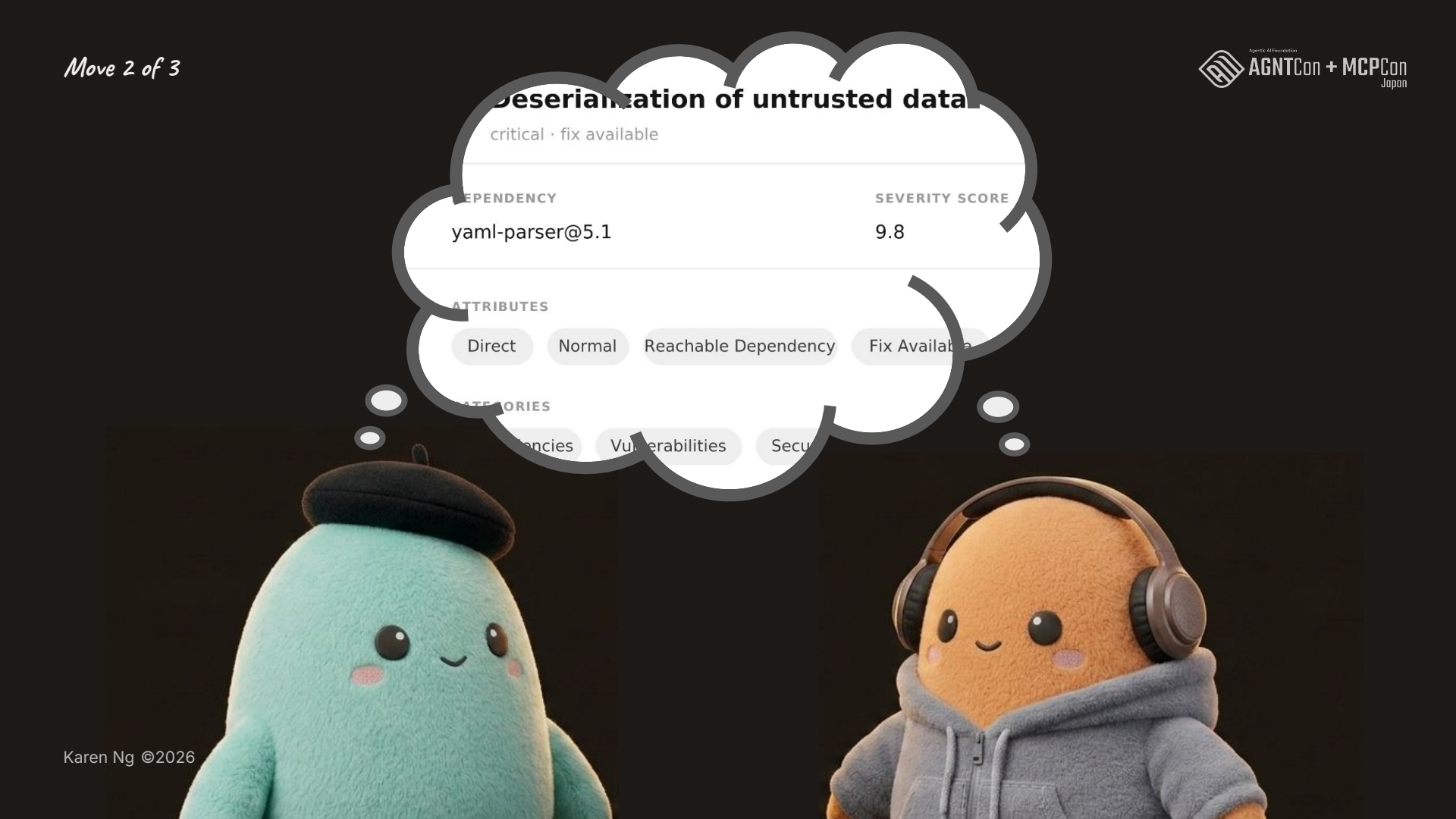


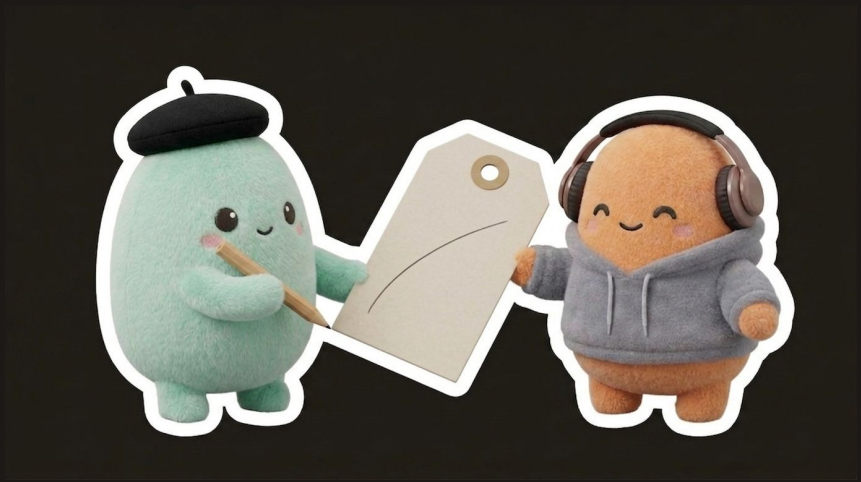

```
<Badge count={12} />
```

Severity Badge




```
<Badge severity="critical" />
```





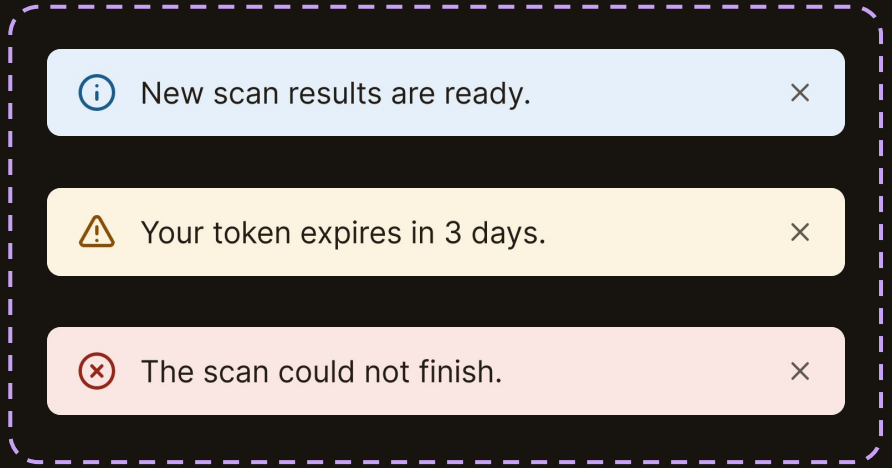
Agree what it's for.
Write it down.

An agent can read.

It cannot **know**.

What do we actually have?

❖ Alert / Light




```
scans/Legacy.tsx
1304  <Box className="scan-alert"> ⚠ …
```

```
policies/Form.tsx
212  <Alert severity="warning" sx={{...}}>
```

```
findings/Row.tsx
63  <FindingAlert severity="high" />
```

```
settings/Saved.tsx
87  {saved && (
88    <Snackbar><Alert severity="success"/>
```

```
scans/StatusBar.tsx
46  return (
47    <Alert type="warning">
```

+ 84 more

Detect Triage Investigate Remediate

Background

Goals

The flow

Requirements

Constraints

Edge cases

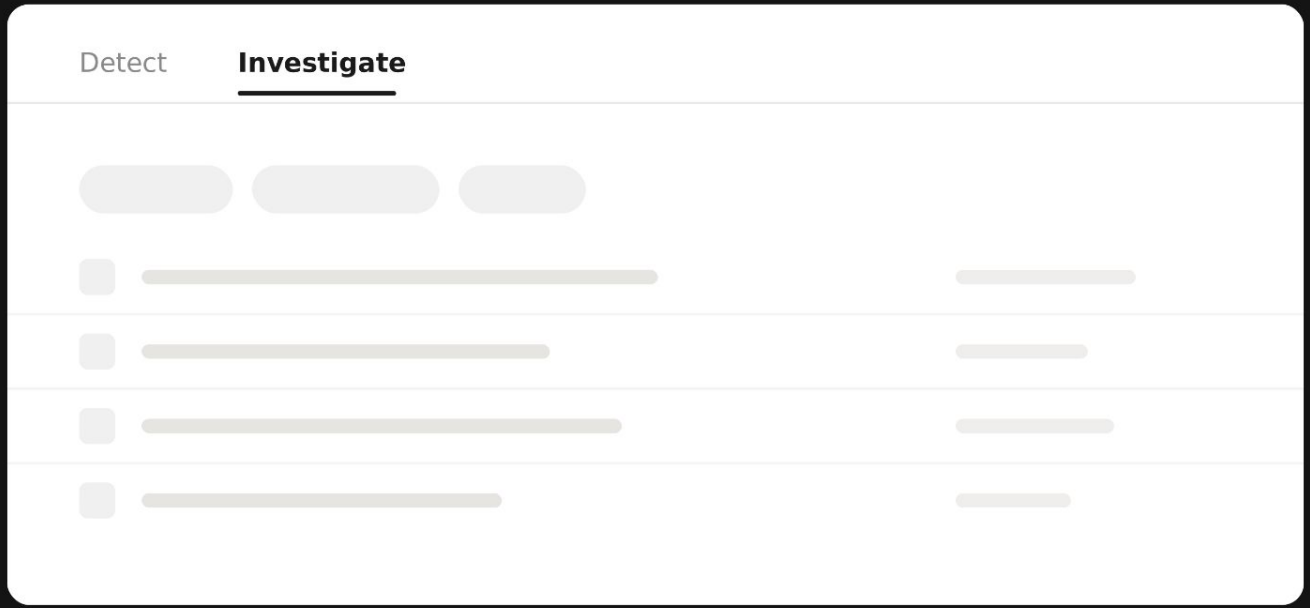
Similar screens

Out of scope

design-system/

Meeting note

design-system/



monorepo / docs / component-index.md

Preview | Code | Blame | 248 lines · 12.1 KB | Raw

Component index

Generated from source on every build · 248 components · do not edit

Component	Defined in	Also called
Alert	ui-theme/src/Alert.tsx	banner, callout
Chip	ui-theme/src/Chip.tsx	tag, label, pill
DataTable	ui-theme/src/DataTable.tsx	grid, table
Drawer	ui-theme/src/Drawer.tsx	sheet, panel
Tabs	ui-theme/src/Tabs.tsx	segmented
Toast	ui-theme/src/Toast.tsx	snackbar
Tooltip	ui-theme/src/Tooltip.tsx	hint

… 241 more

monorepo / docs / product-shape.md

Preview | Code | Blame | 41 lines · 2.3 KB | Raw

How this product is shaped

Written by hand · last reviewed 2026-07-14 · design + eng

Workflows

- A workflow is one page, however many steps it has
- Detect and investigate are the same surface:

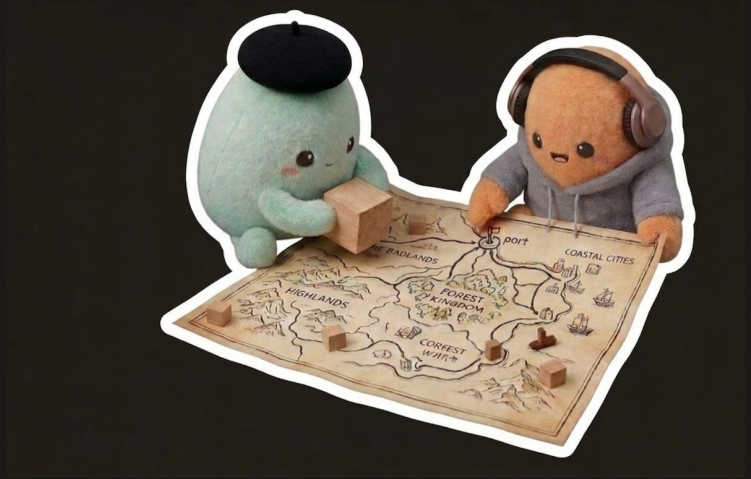
  dashboard → table → filter → detail panel

Not surfaces

- Triage is a decision with no screen of its own
- Remediate happens outside this product

Naming

- Tags label. Chips carry state
- A verb is never a tab



Move the context to where the reader is.

Files

main

Go to file T

LICENSE
README.md
alert.md
component-doc.template.md
component-index.template.md
tiers.md
two-minute-test.md
workflow.md

ng-karen The Third User Kit 2fdcd84 · 1 minute ago History

Preview Code Blame 133 lines (98 loc) · 5.27 KB Raw

Alert

Status: Core · **Since:** 2.1 · **Owner:** Design Systems

An Alert communicates a status, result, or consequence that is relevant to the content it appears beside. It stays on screen until the situation changes or the user dismisses it.

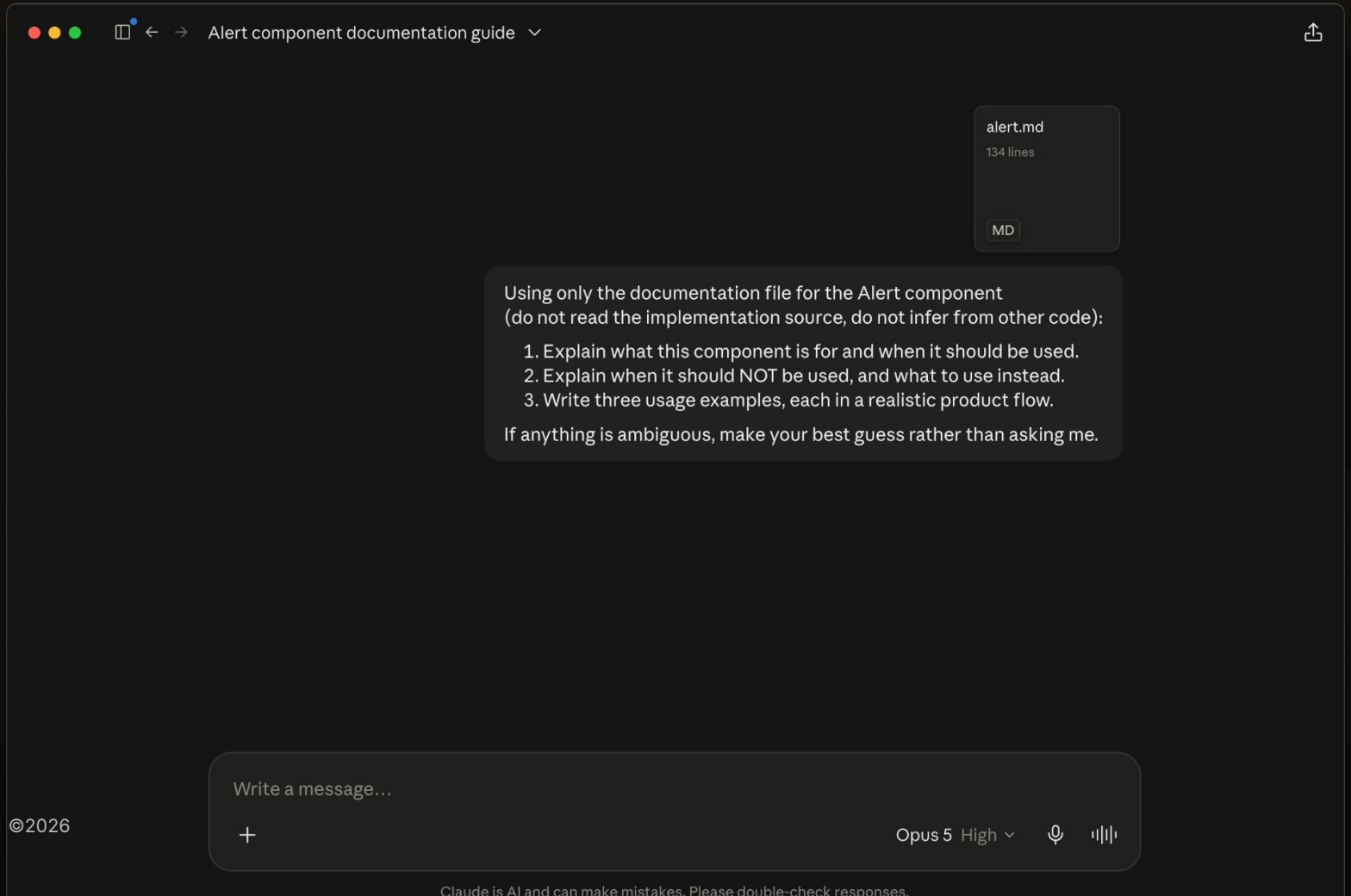
Anatomy

1. **Icon** — derived from `intent`, communicates severity at a glance
2. **Title** — required, one line, carries the message on its own
3. **Body** — optional, supporting detail
4. **Actions** — optional, up to two
5. **Dismiss** — optional close control

The component fills the width of its container. It does not set its own margins; spacing is the responsibility of the layout that places it.

Props

Prop	Type	Default	Notes
`intent`	`'info' | 'success' | 'warning'`	`'info'`	Sets colour, icon, and default ARIA role



If anything is ambiguous, make your best guess rather than asking me.

1. What it's for

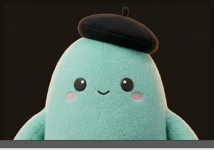
An Alert is **inline, persistent status attached to a place in the UI**. Two properties define it:

- **Contextual** — it communicates a status, result, or consequence relevant to the content it sits beside, not to the app as a whole. It fills its container's width and sets no margins of its own; the surrounding layout owns the spacing.
- **Persistent** — it stays until the situation changes or the user dismisses it. There's no timeout.

Use it when the reader needs to know something in order to act correctly on the thing in front of them:

- `info` — neutral context they need before acting ("Changes apply to new tickets only")
- `success` — a completed operation with a durable result, i.e. something still true after the message is read
- `warning` — a condition that will cause a problem if left alone
- `danger` — a failure, or an action that can't be undone

The intent choice isn't just colour: it selects the icon and the announcement behaviour. `info`



What designer stated in the doc

Alert component documentation guide

2. When not to use it

The doc is explicit on a few of these and leaves the rest to inference — I've marked which is which.

Stated in the doc:

- **Zero-data views** → use `EmptyState`. An empty list isn't a status change, it's a state of the view.
- **Narrowing a list** → use `FilterBar`. Don't use an Alert as a filter-status readout.
- **Reporting several unrelated conditions at once.** One Alert per region; summarise into one rather than stacking.
- **Anything needing three or more actions** → "it needs a different surface." The Alert caps at `action` plus `secondaryAction`, and `secondaryAction` is invalid without `action`.
- `compact` **with** `danger`. A failure gets the full component; compact drops the body slot and allows only a title and one action.
- **Overriding** `role`**, colour, or typography.** `className` is for layout and spacing only, and `role` should be overridden only where the default announcement behaviour is demonstrably wrong.

Inferred from the definition (the doc names no substitute component for these, so treat the alternatives as my suggestion, not the design system's):

- **Transient confirmations** — "Copied", "Saved" with nothing durable behind them. Persistence is the whole point of an Alert; a message meant to disappear on its own belongs in whatever ephemeral notification surface the system provides.
- **App-wide or account-wide announcements** — outages, billing lockouts, maintenance windows. These aren't relevant to the content beside them, and putting one inline misrepresents its scope.
- **Blocking a decision.** An Alert has no trigger and doesn't take focus on appearance; dismissing it moves focus forward rather than back. If the user must answer before proceeding, that's a dialog.
- **Field-level validation.** An Alert's title is a standalone sentence about the region, not an error bound to a single input.

Write a message...

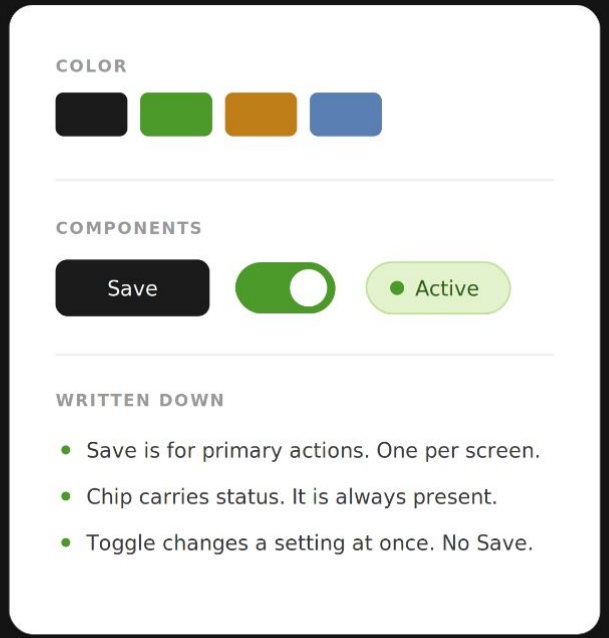
Opus 5 High

What agent reinterpreted on its own

Wrote down something nobody ever chose

Built something new when the real one was already sitting in your library

Got the right pieces and put them in the wrong shape




```
"name": "git_diff",
"description": "Shows differences
            between branches or commits"

"name": "compare_versions",
"description": ""

"name": "list_branches",
"description": ""

"name": "get_commit",
"description": ""
```

git/src/mcp_server_git/server.py

```
{
  "name": "git_diff",
  "description":
    "Shows differences between
    branches or commits",
  "inputSchema": {
    "type": "object",
    "properties": {
      "repo_path": { "type": "string" },
      "target": { "type": "string" }
    },
    "required": ["repo_path", "target"]
  }
}
```

review/src/mcp_server_review/server.py

```
{
  "name": "compare_versions",
  "description":
    "Compares two versions and
    returns the differences",
  "inputSchema": {
    "type": "object",
    "properties": {
      "file_path": { "type": "string" },
      "version_a": { "type": "string" },
      "version_b": { "type": "string" }
    },
    "required": ["file_path"]
  }
```

git/src/mcp_server_git/server.py

```
{
  "name": "git_diff",
  "description":
    "Shows differences between
    branches or commits",
  "inputSchema": {
    "type": "object",
    "properties": {
      "repo_path": { "type": "string" },
      "target": { "type": "string" }
    },
    "required": ["repo_path", "target"]
  }
}
```

review/src/mcp_server_review/server.py

```
{
  "name": "compare_versions",
  "description":
    "Compares two versions and
    returns the differences",
  "inputSchema": {
    "type": "object",
    "properties": {
      "file_path": { "type": "string" },
      "version_a": { "type": "string" },
      "version_b": { "type": "string" }
    },
    "required": ["file_path"]
  }
```

A name?

git/src/mcp_server_git/server.py

```
{
  "name": "git_diff",
  "description":
    "Shows differences between
    branches or commits",
  "inputSchema": {
    "type": "object",
    "properties": {
      "repo_path": { "type": "string" },
      "target": { "type": "string" }
    },
    "required": ["repo_path", "target"]
  }
}
```

review/src/mcp_server_review/server.py

```
{
  "name": "compare_versions",
  "description":
    "Compares two versions and
    returns the differences",
  "inputSchema": {
    "type": "object",
    "properties": {
      "file_path": { "type": "string" },
      "version_a": { "type": "string" },
      "version_b": { "type": "string" }
    },
    "required": ["file_path"]
  }
```

A reader?

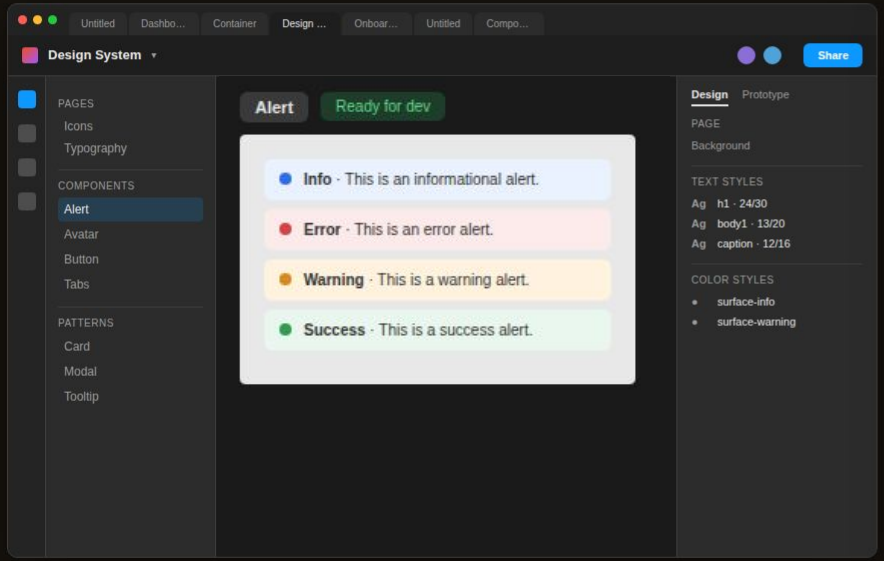
Style guide

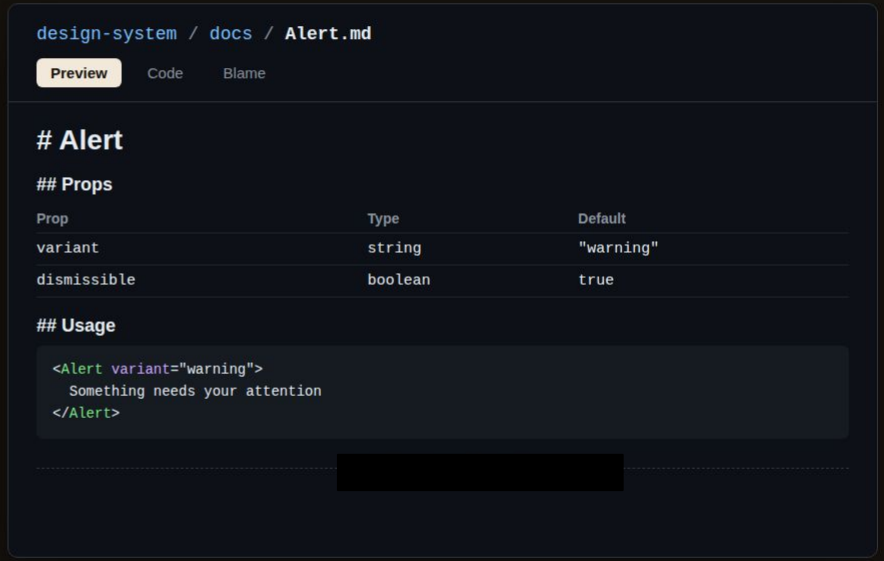
Component library

Design systems

Agentic design system


Agentic AI Foundation
AGNTCon + MCPCon
Japan
Karen Ng ©2026





github.com/ng-karen/third-user-kit

The Third User Kit

A component library has always had two users: the product designer who uses it and the engineer who ships it. It now has a third — the AI agent reading the same files to generate UI.

This kit is the workflow from the talk **"Built for Two: What Design Systems Are Missing When AI Agents Become the Third User"** — AGNTCon + MCPCon Japan 2026, September 11, Tokyo. Everything here is deliberately small: one test, one index, one set of tiers, one doc template.

What's in the kit

File	What it is	Time to adopt
workflow.md	The workflow the other files support. Five steps here; grouped into the three moves used on stage, mapped at the top of the file	A read
two-minute-test.md	The test: what does an agent get wrong about your best-documented component?	2 minutes
alert.md	The document from the talk — the one that looks finished. Run the test on it and see what it can't answer	2 minutes
component-index.template.md	The index: one file, every component, one line each	An afternoon
tiers.md	Core / Extended / Experimental — status criteria and promotion rules	A team discussion
component-	Component doc template with multi-author review built in	Per component

Get another

reader

Agree on the

name

Give it a

map


Agentic AI Foundation
AGNTCon + MCPCon
Japan
Karen Ng ©2026


Agentic AI Foundation
AGNTCon + MCPCon
Japan
Karen Ng ©2026


Agentic AI Foundation
AGNTCon + MCPCon
Japan
Karen Ng ©2026

AGNTCon + MCPCon Japan

Karen Ng カレン

Principal UX Designer | ENDOR LABS

karen@endor.ai

LinkedIn

linkedin.com/in/fivepluszero

The Third User Kit

github.com/ng-karen/third-user-kit