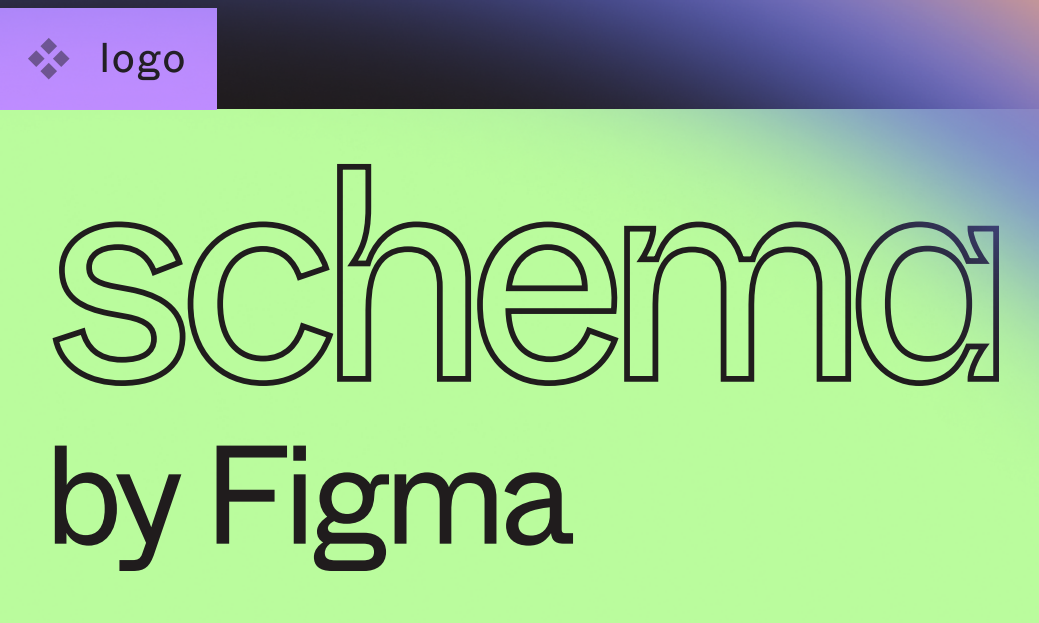


Mastering the art of crafting code-aligned UI kits



Jan Toman

Design System Lead,
Productboard





Jan Toman

Design System Lead

 @HonzaTmn



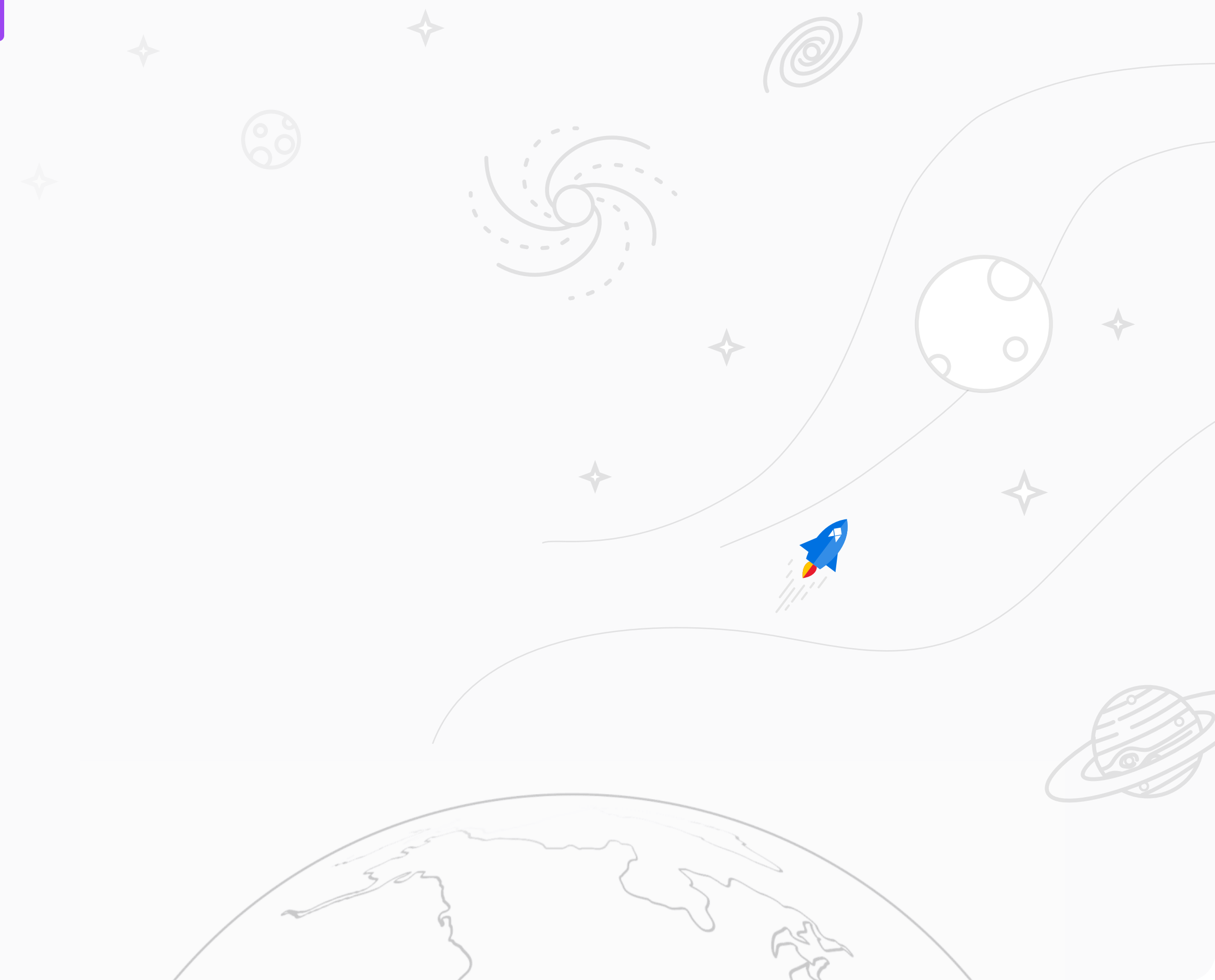
We're hiring!

Productboard helps product managers understand what customers need, prioritize what to build next, and rally everyone around the roadmap.

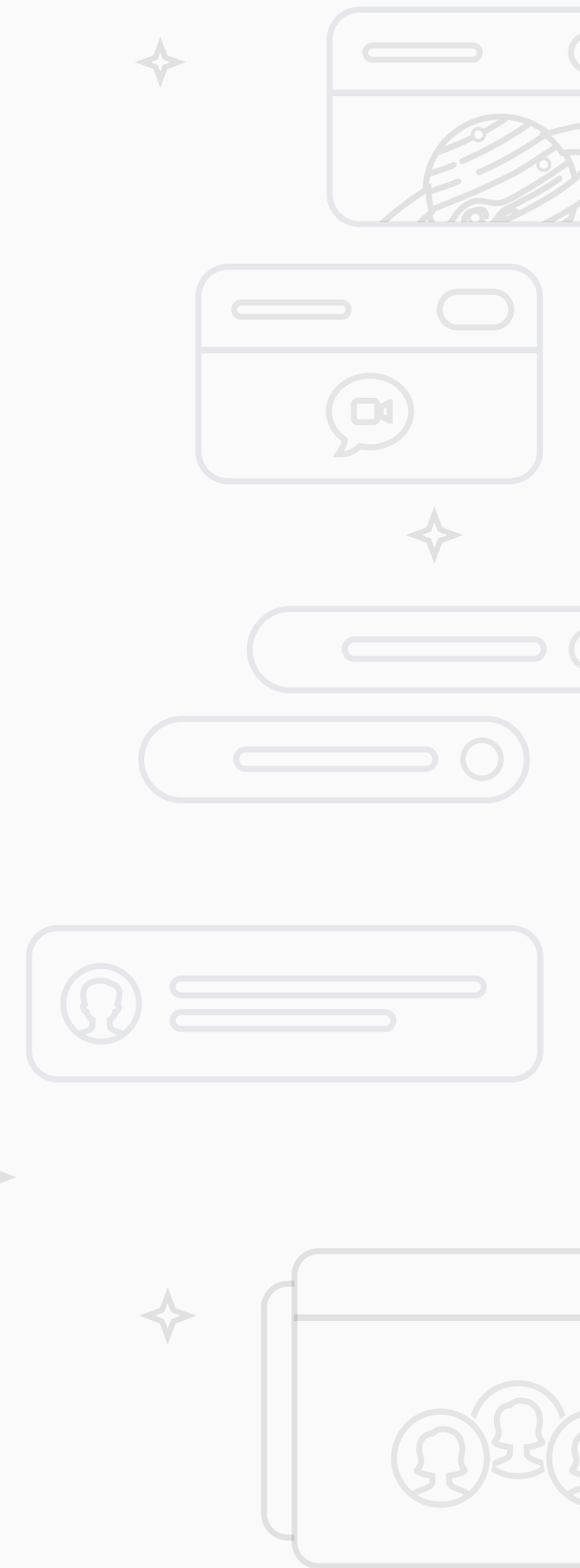
Previously working at



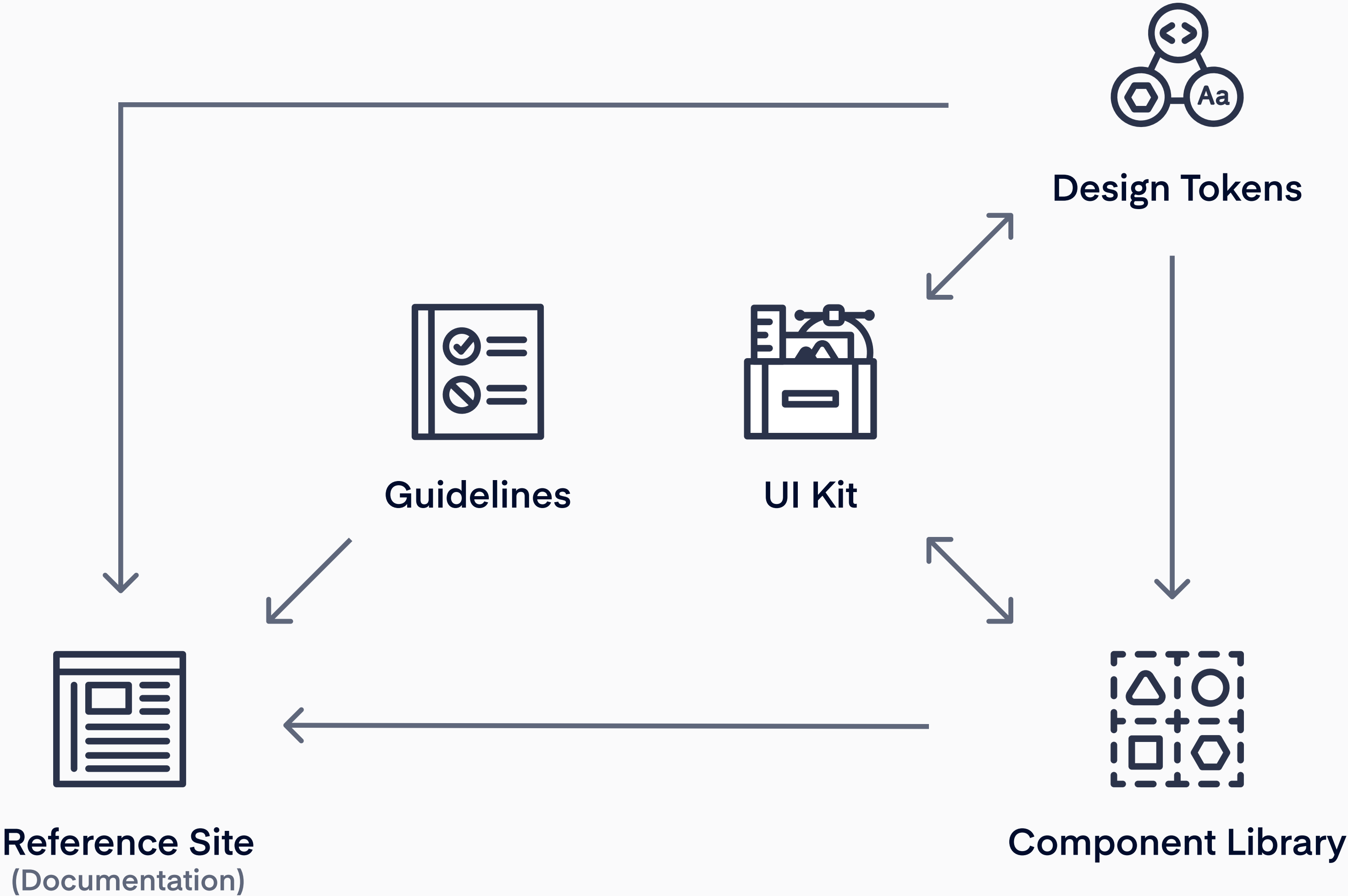
Hello!

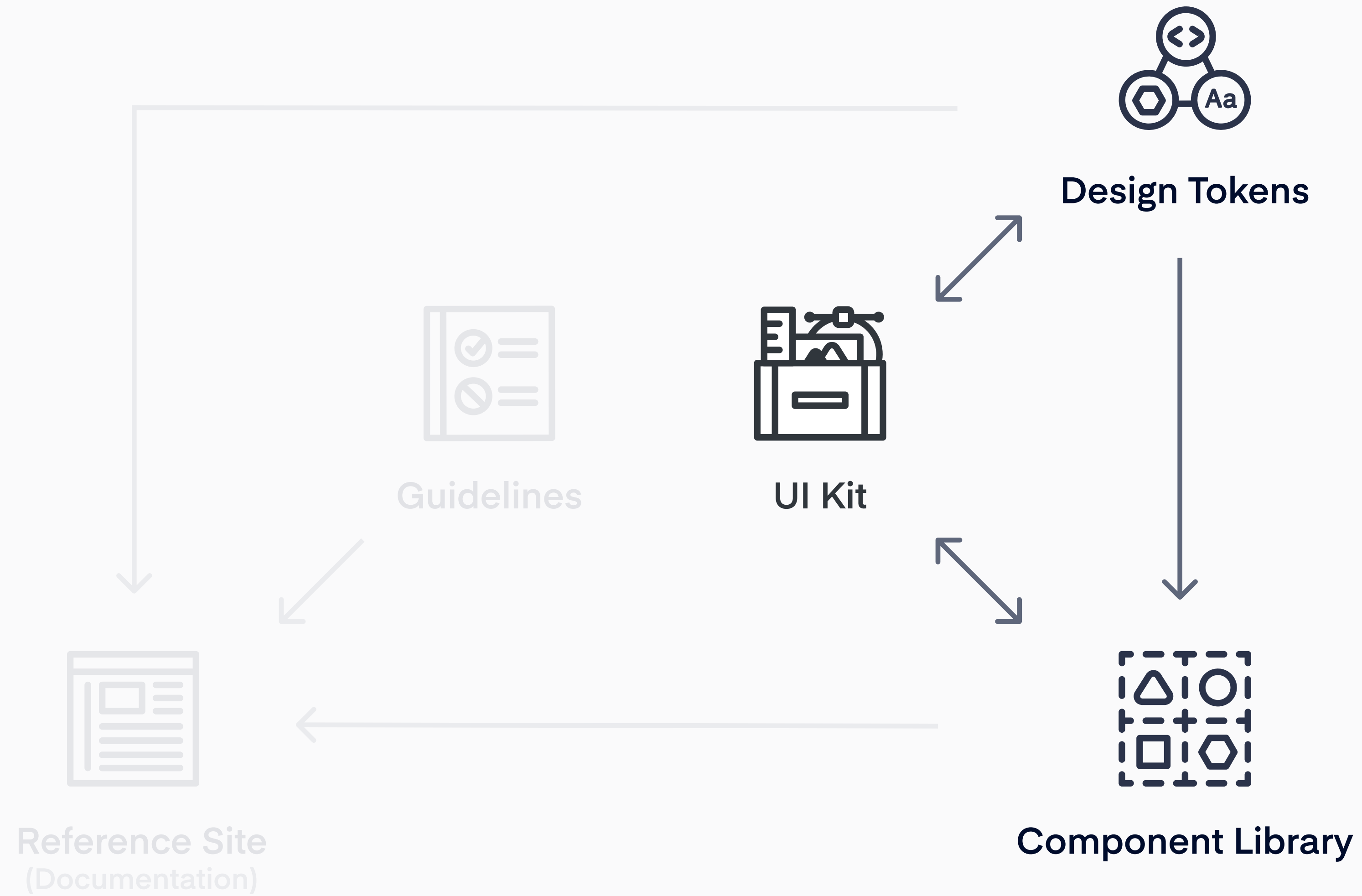


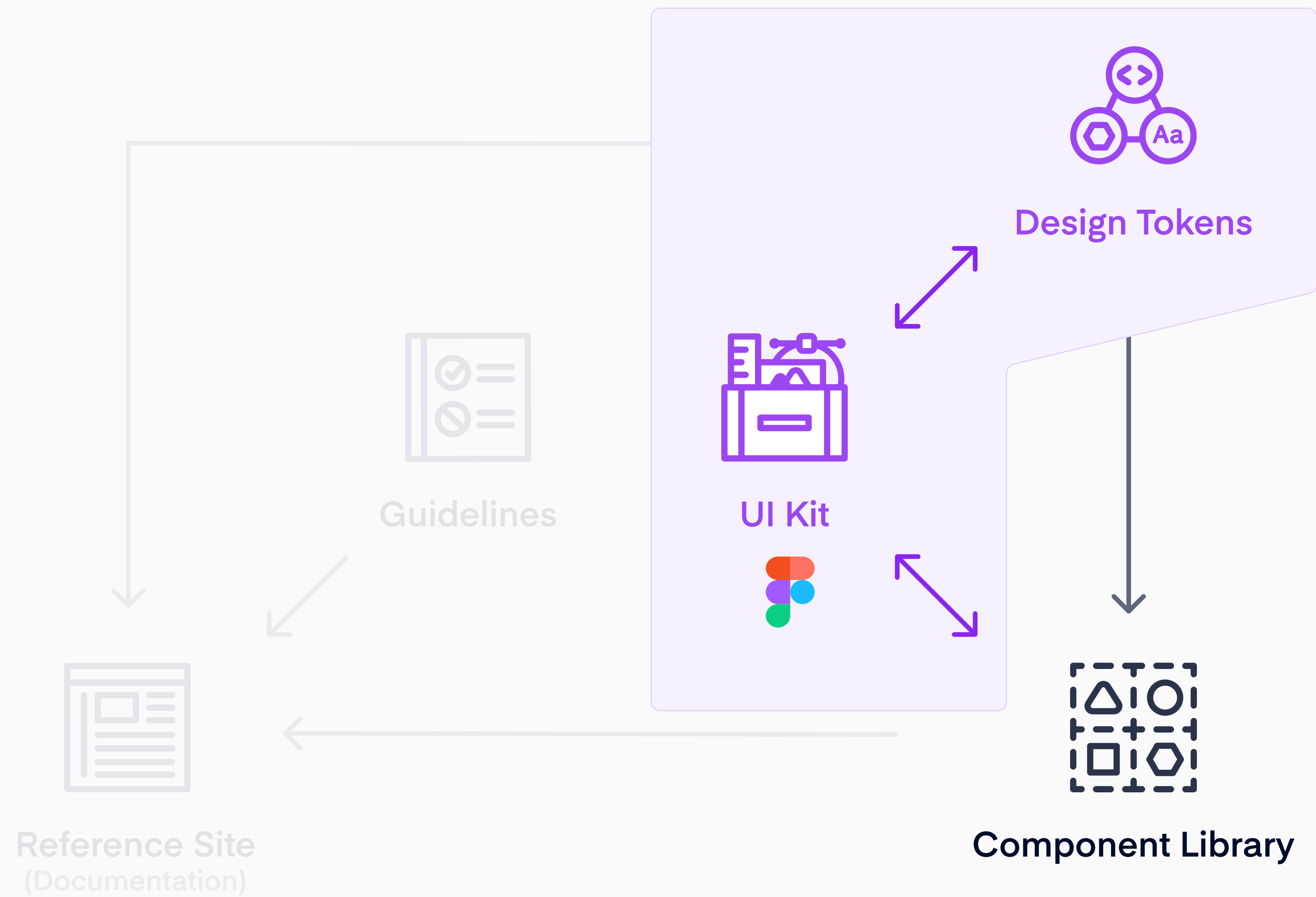
Let's talk about design systems



Design systems







What will we learn today?



How component API
choices affect usability



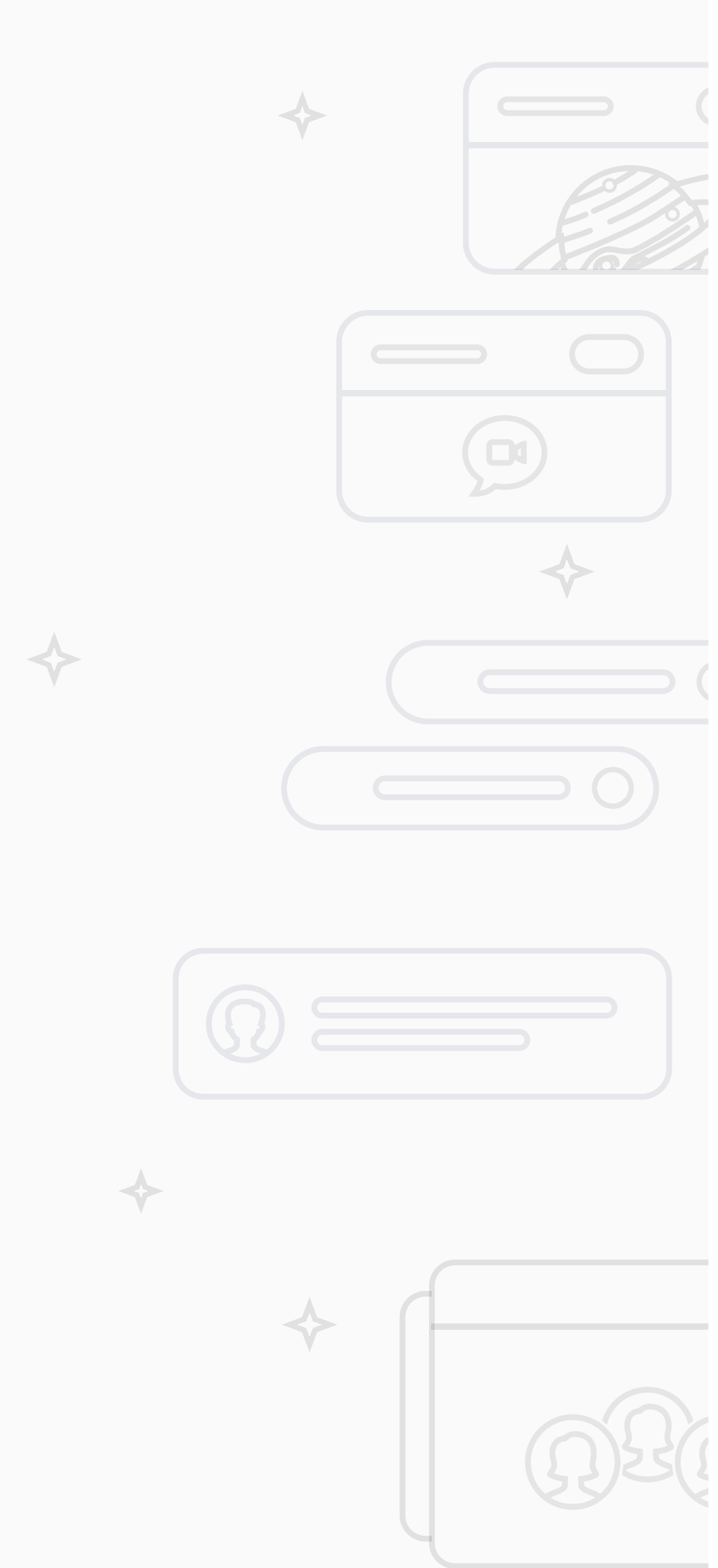
Making changes
(and not breaking things)



Aligning Figma
with code

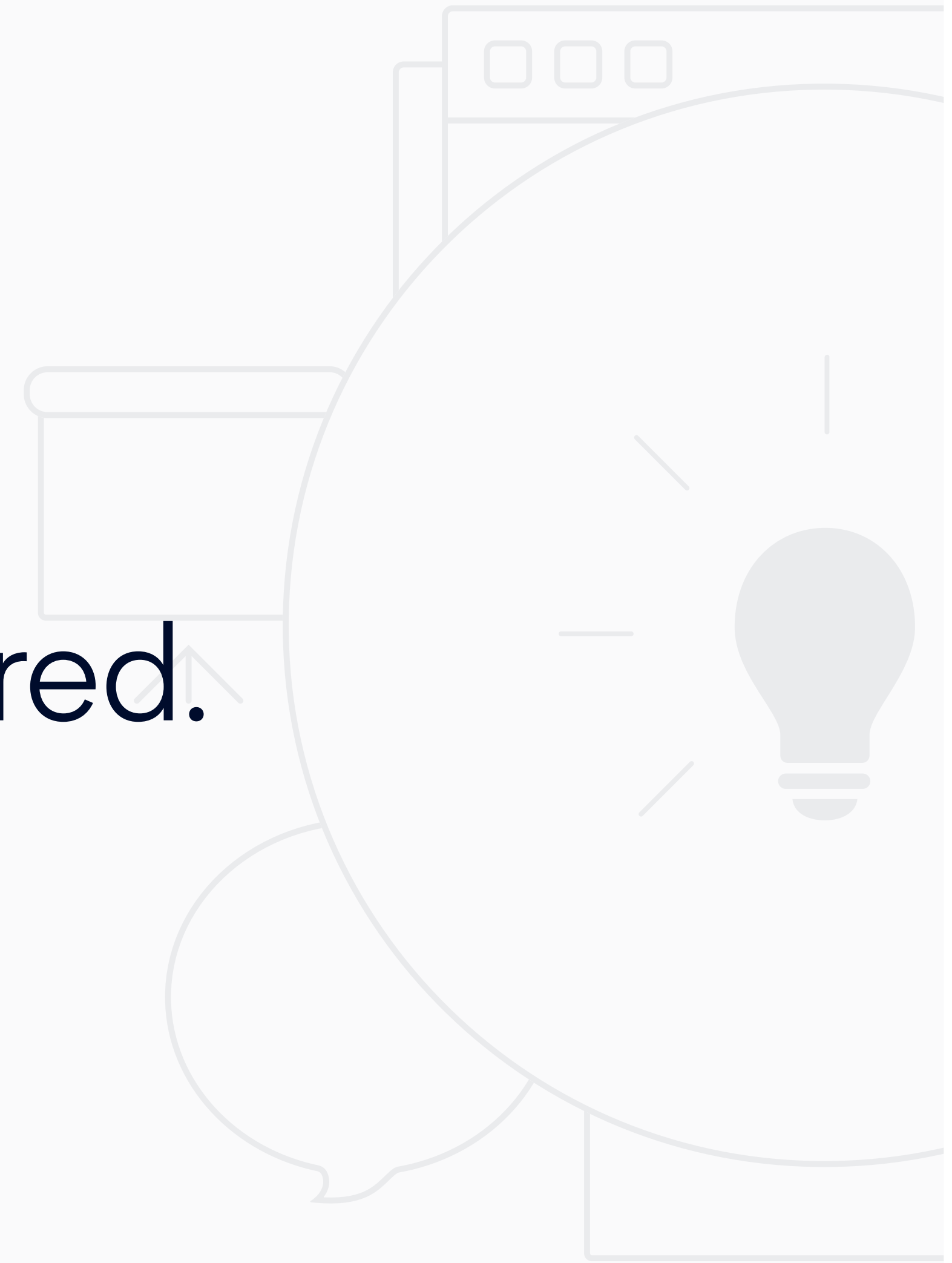


How component API choices affect usability



What's component API?

The way each component can be controlled and configured.



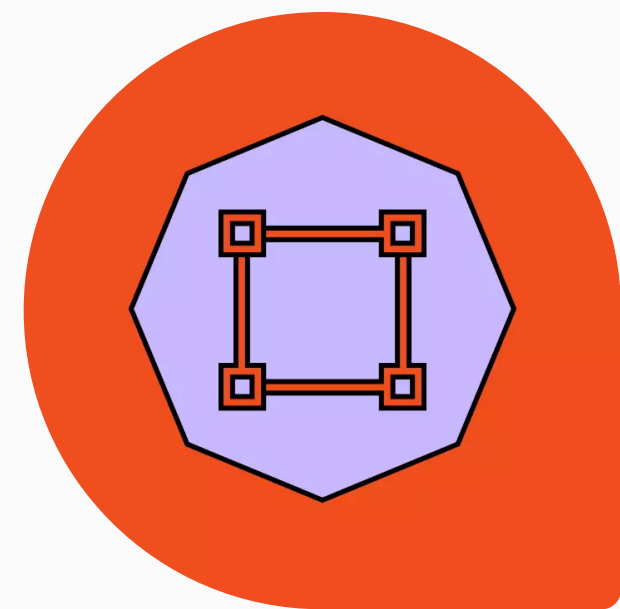
Component API in React



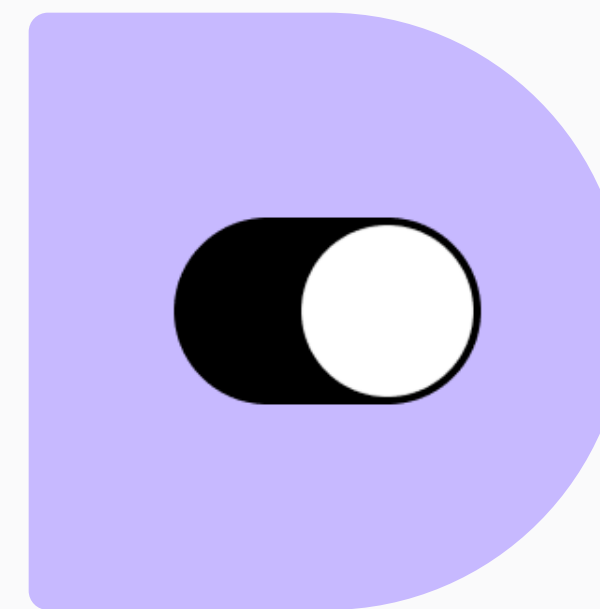
```
<ButtonPrimary  
  size="large"  
  icon={<FloppyDiskIcon />}  
  isSubmit  
  onClick={saveFormCallback}  
>  
  Save form  
</ButtonPrimary>
```


Component API in Figma

The way each component can be controlled and configured.



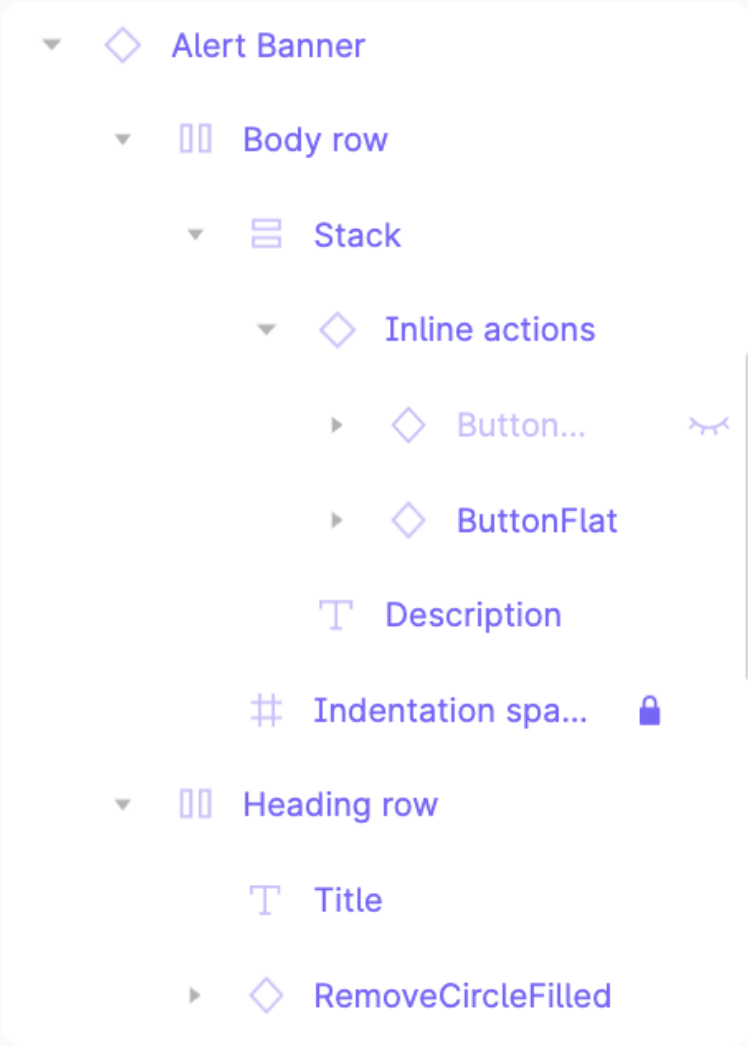
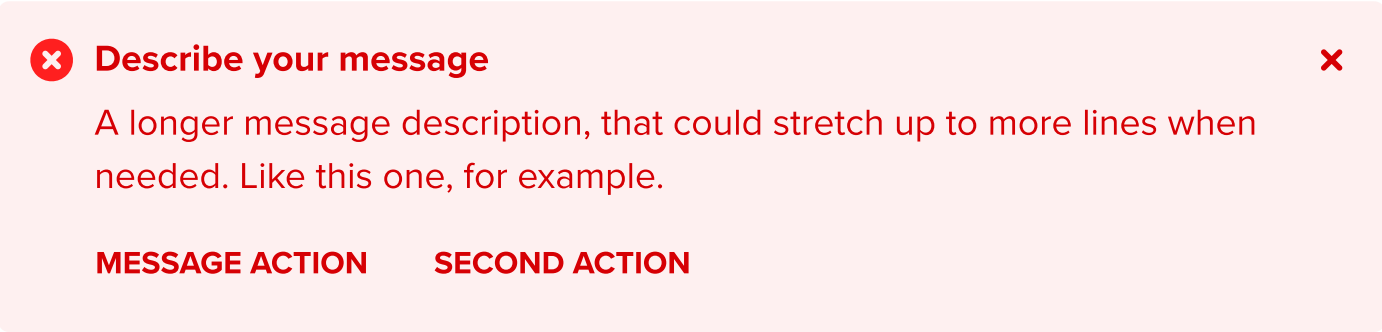
Layers



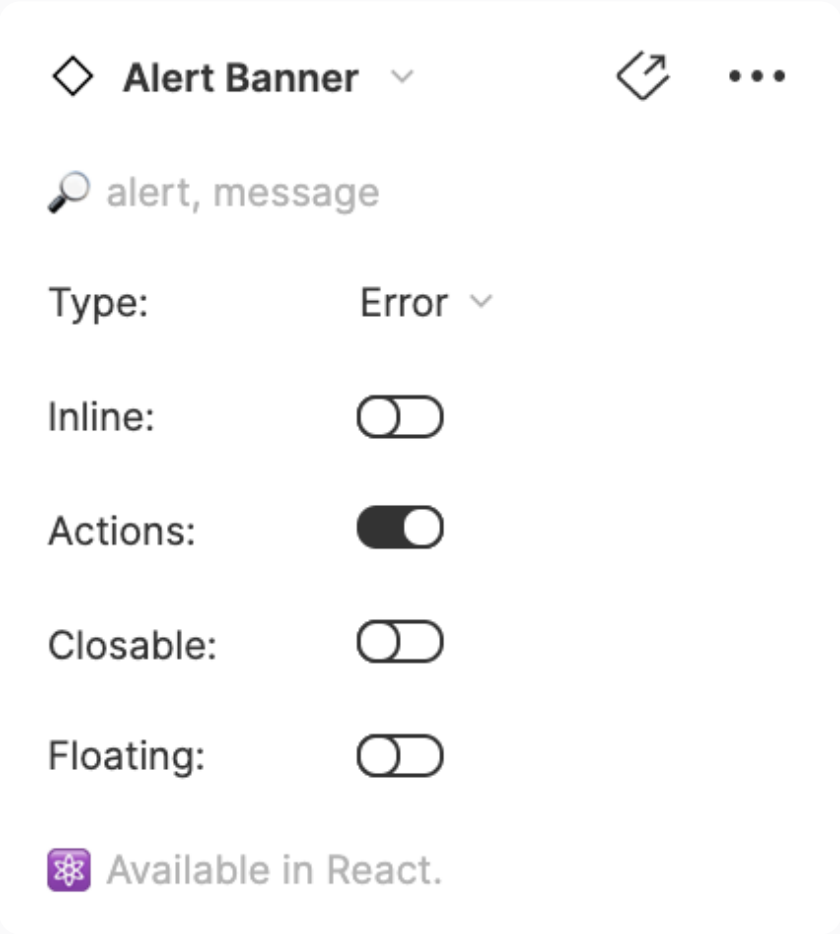
Variant controls

⚠ Not to confuse with Figma REST or Plugin API!

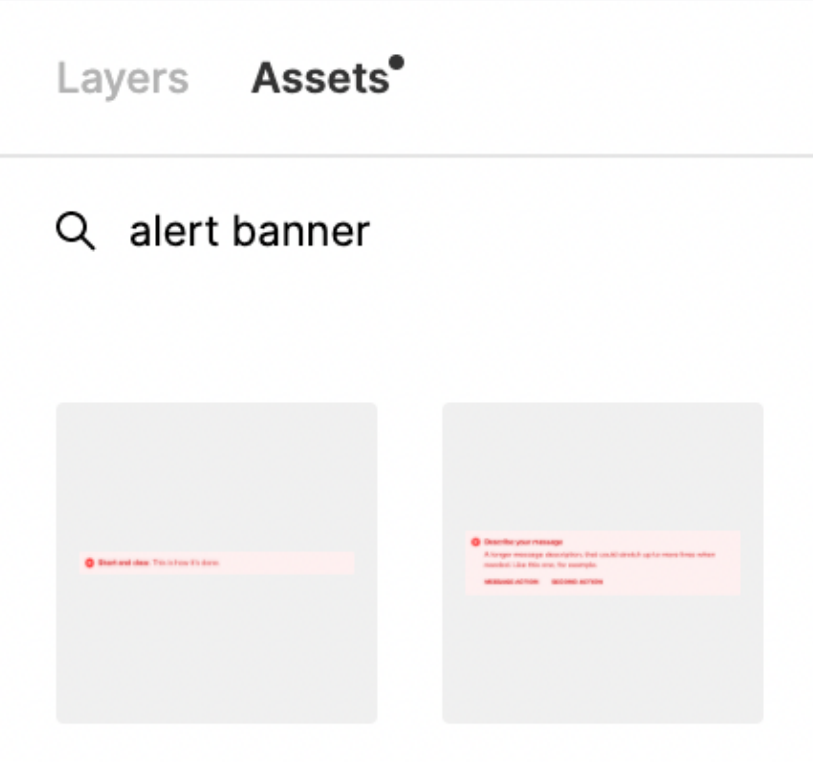
Component API example: AlertBanner



Layers

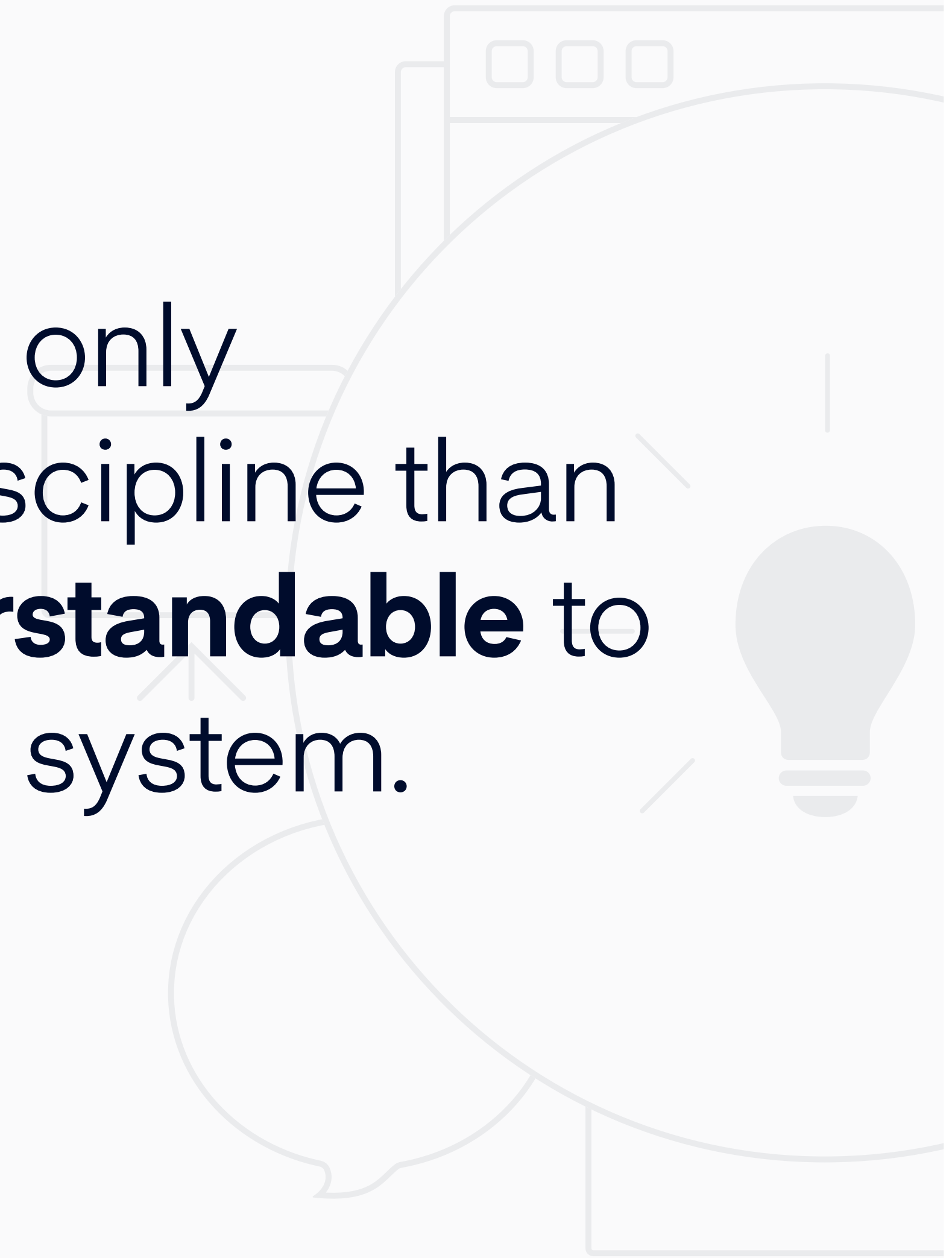


Variant controls



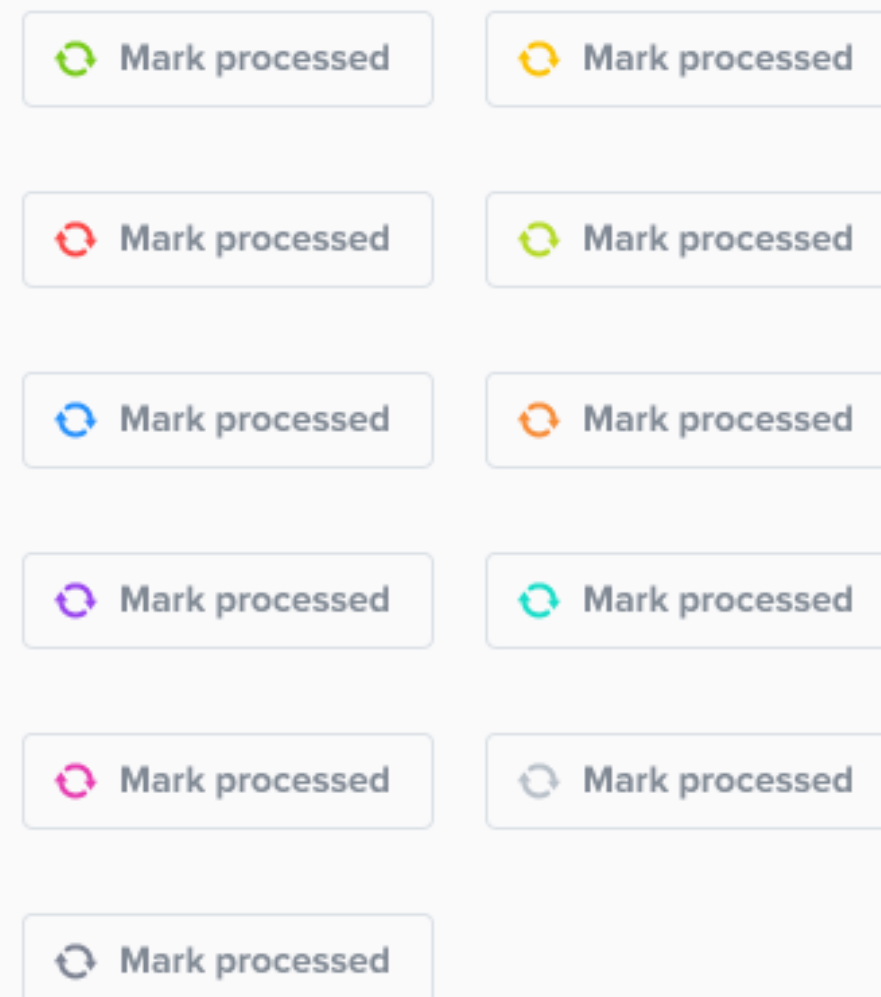
Previews in assets panel

Designing APIs for components that only I will use is a completely different discipline than **creating components that are understandable** to people who will only use our design system.



API approach #1 – Optimized for me

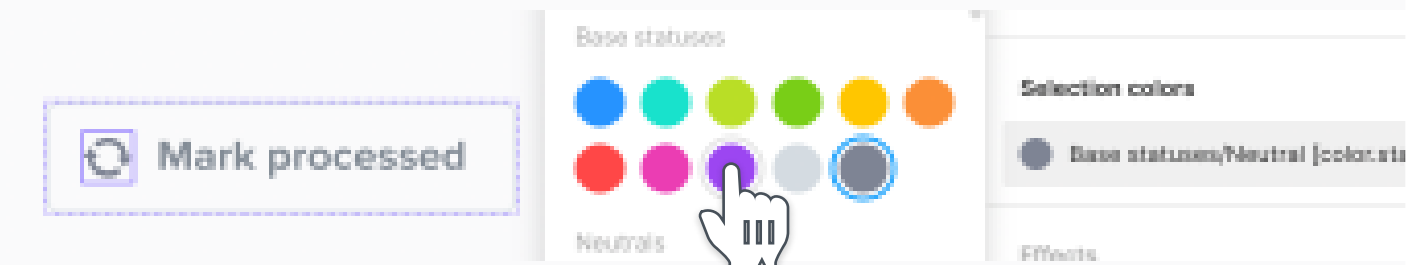
All variants we need



Our component definition



Consumer using a component



Benefits 👍

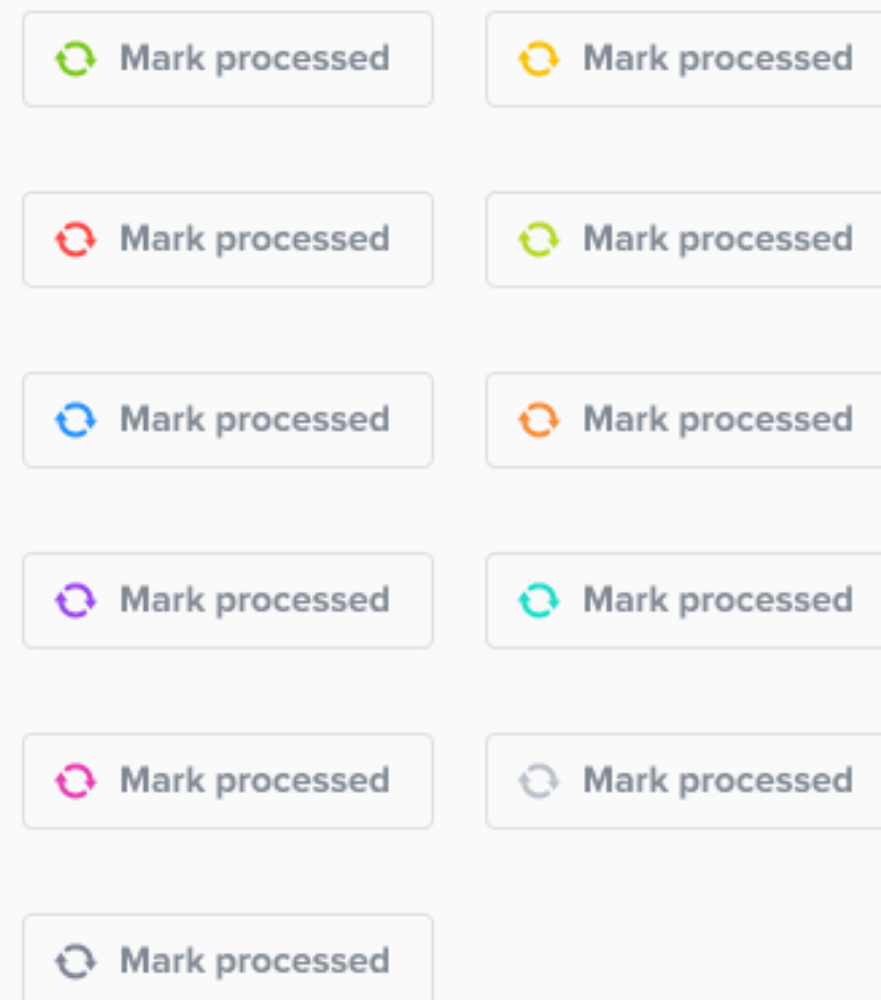
- Simple for maintenance
- More visual flexibility

Cons 👎

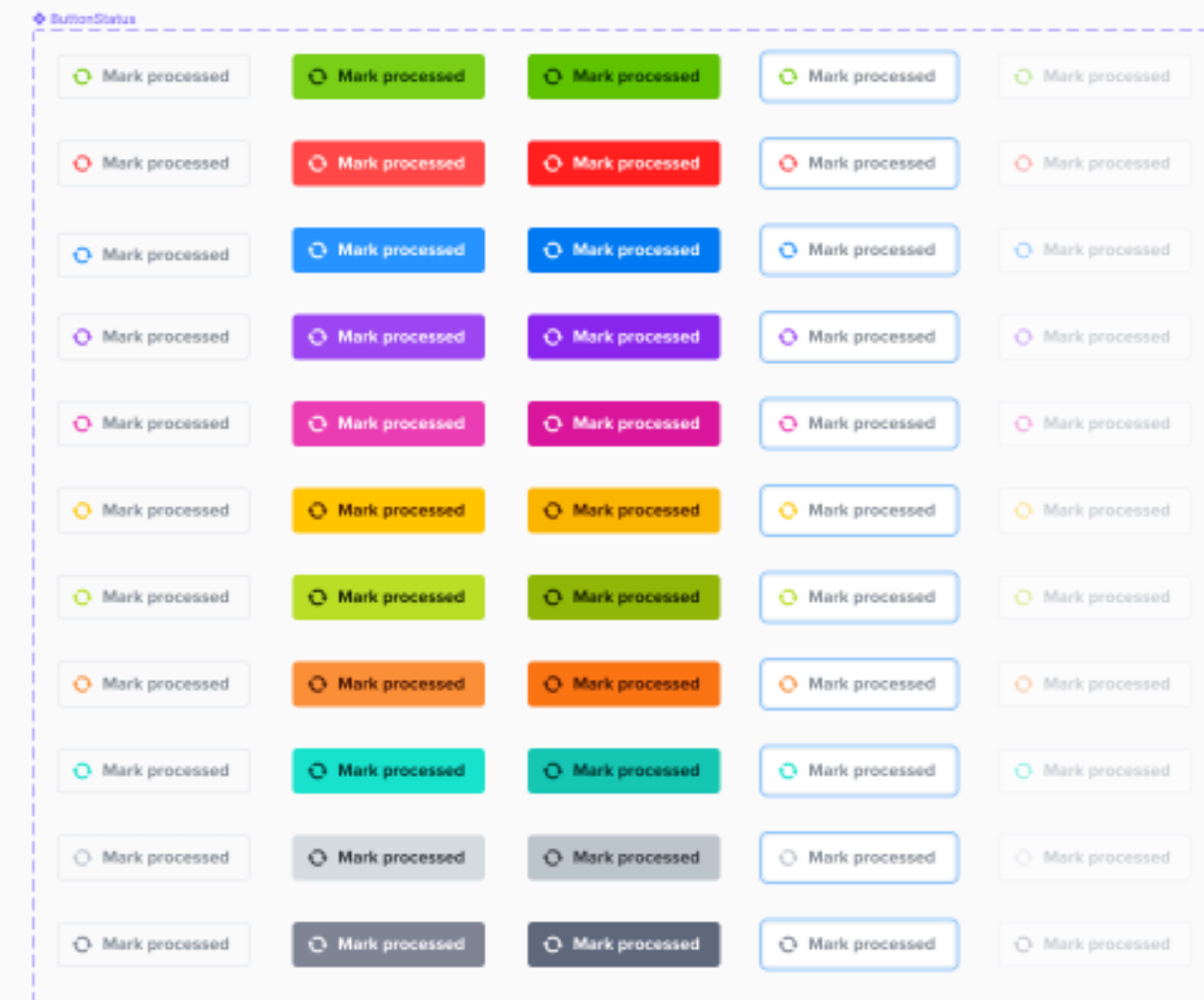
- Need to know aspects of visual language
- Need for “how to use” guideline
- More cognitive load for change
- It will lead to inconsistencies (eventually)

API approach #2 – Optimized for others

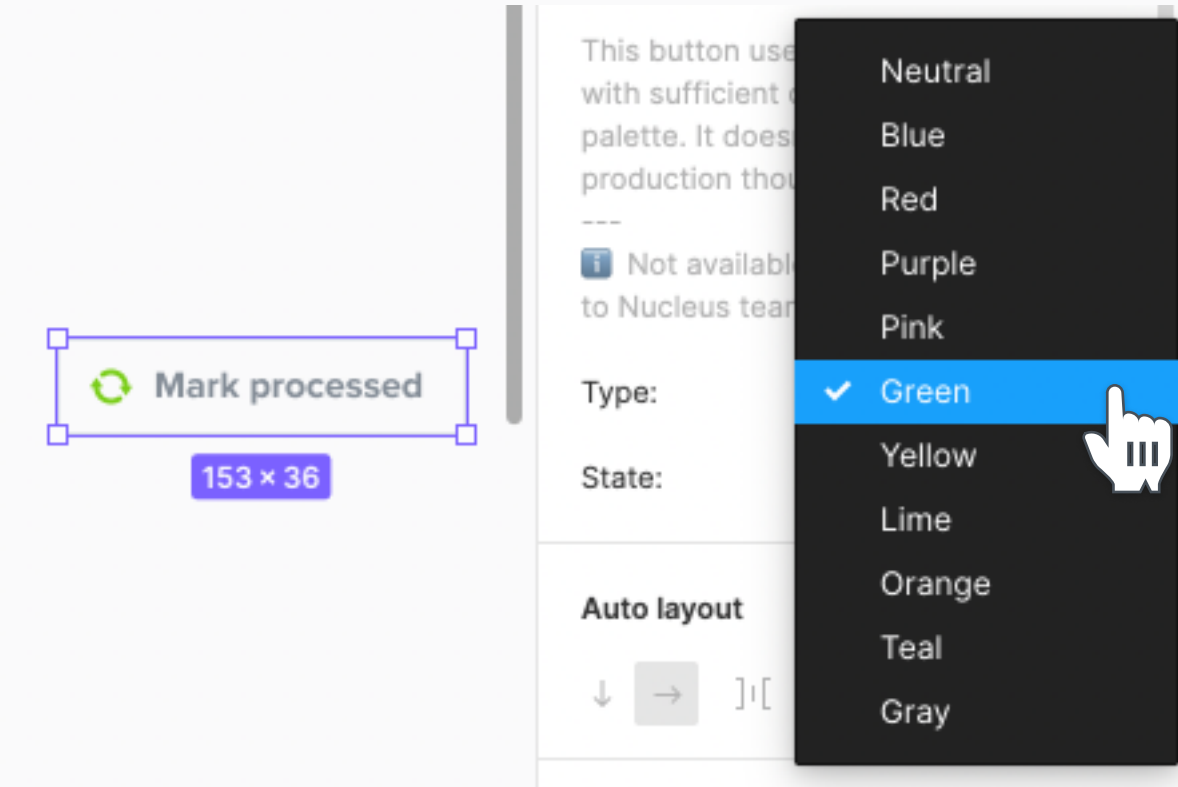
All variants we need



Our component definition



Consumer using a component



Benefits 🙌

- It's clear what's supported
- Intuitive component API
- No need to know design language
- Less cognitive load
- No room for inconsistency
- Centralized control over visual look

Cons 🙄

- Harder for maintenance
- Possibly larger library size

UI kit maintainer

UI kit consumer

 Knowing all components and variants in the system

Yes

No

 The inner structure of each component

Yes

No

 Deep knowledge of visual language

Yes

Some

 Mastery in all the Figma quirks & hacks

Yes

Some

UI kit maintainer

UI kit consumer

Now

After a while

 Knowing all components and variants in the system

Yes

→

Fading

No

 The inner structure of each component

Yes

→

No

No

 Deep knowledge of visual language

Yes

→

Fading

Some

 Mastery in all the Figma quirks & hacks

Yes

→

Yes

Some

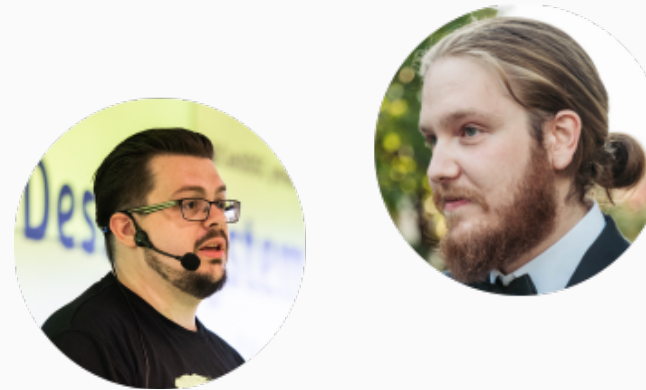
How should we design API for our UI kit components in Figma?

Should it be flexible or constrained?

Will consumers need to know details of product visual language?

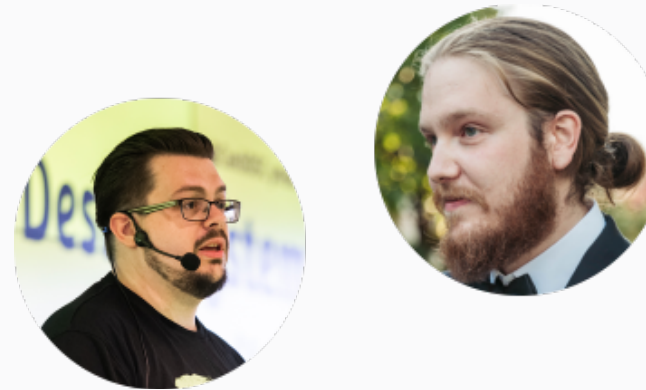
What is API approach for our code components?

Will it be used only by me, or also by others?



2 Design Systems Designers

(one until August 2021)

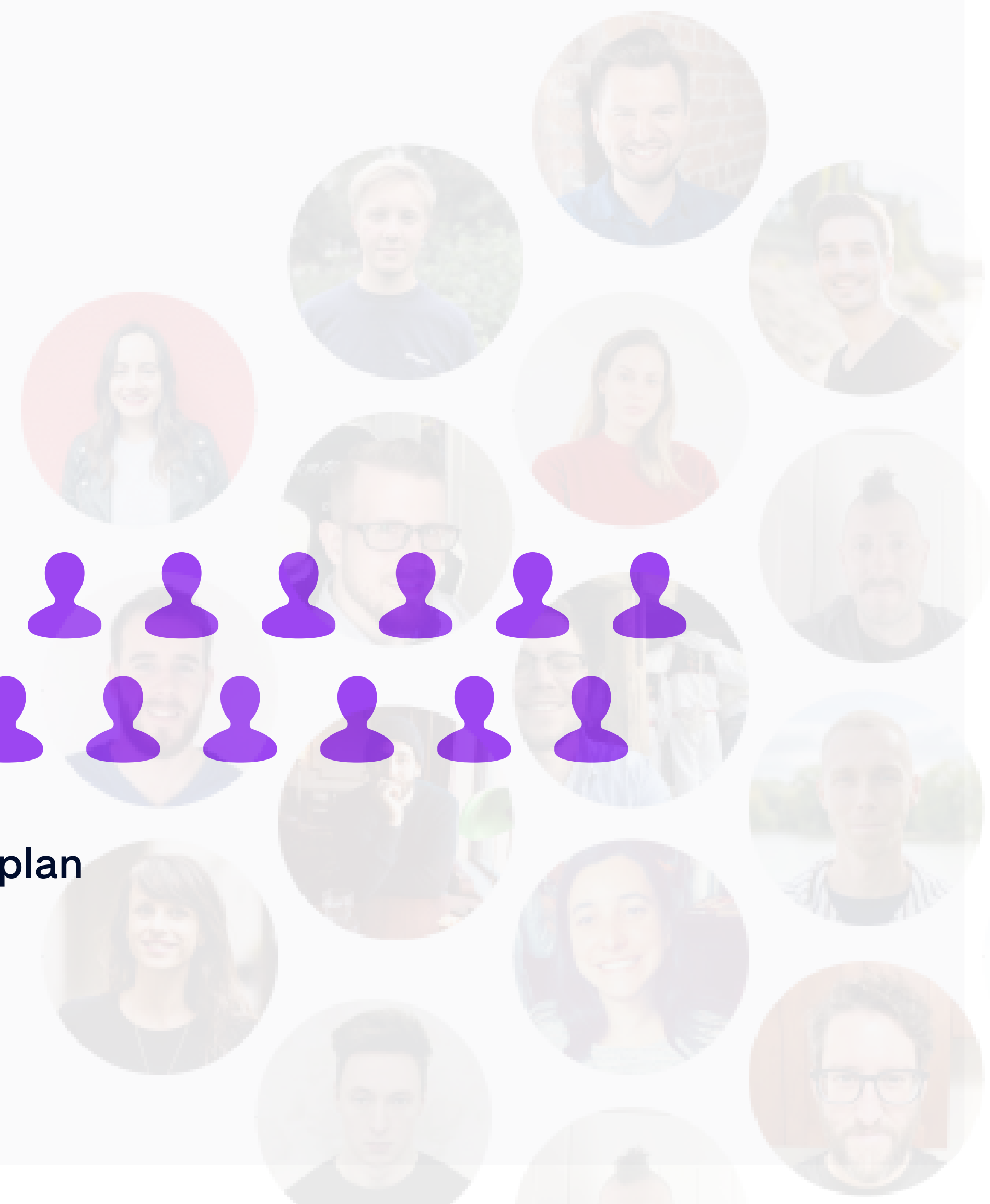


2 Design Systems Designers

(one until August 2021)



35 weekly active editors on our Figma plan



UI kit maintainers vs. consumers

1 : ~18↑

Design System Designer

Product Designers
UX Writers
Design Managers
Product Managers
Content Editors
...

If we're able to **save at least 10 seconds everytime a consumer uses a component** from our UI kit, it's more important to **optimize our UI kit for usability** rather than maintainability.

How might we create UI kits that...

...designers will love using?

...won't break with future changes?

...will be aligned with code?

How might we create UI kits that...

...designers will love using?

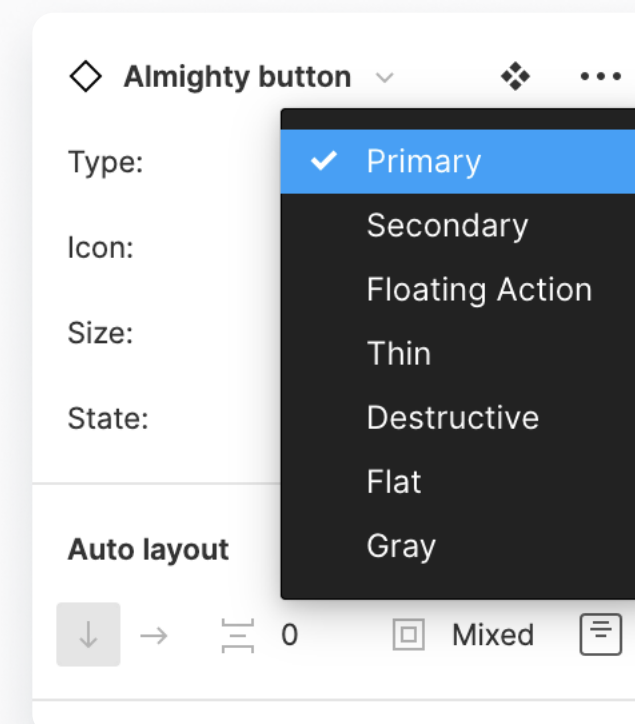
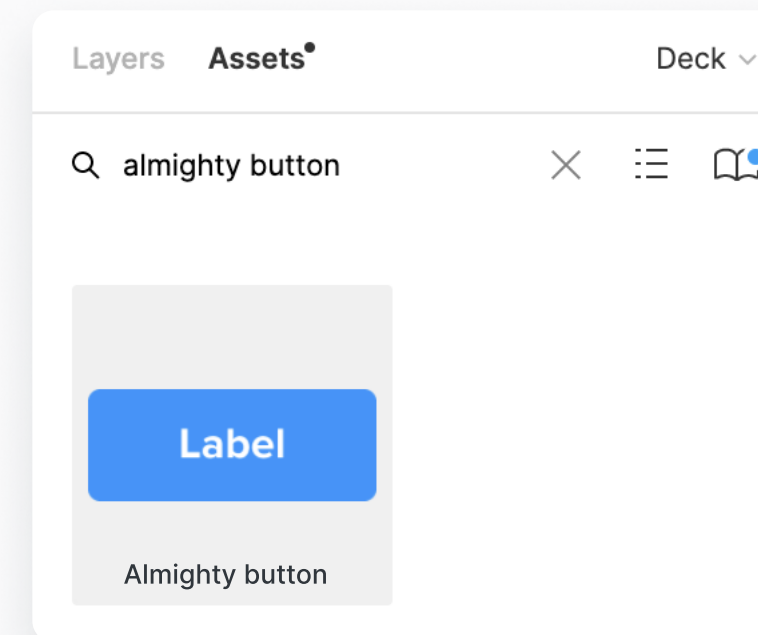
...won't break with future changes?

...will be aligned with code?

Have each distinctive variant as a separate component

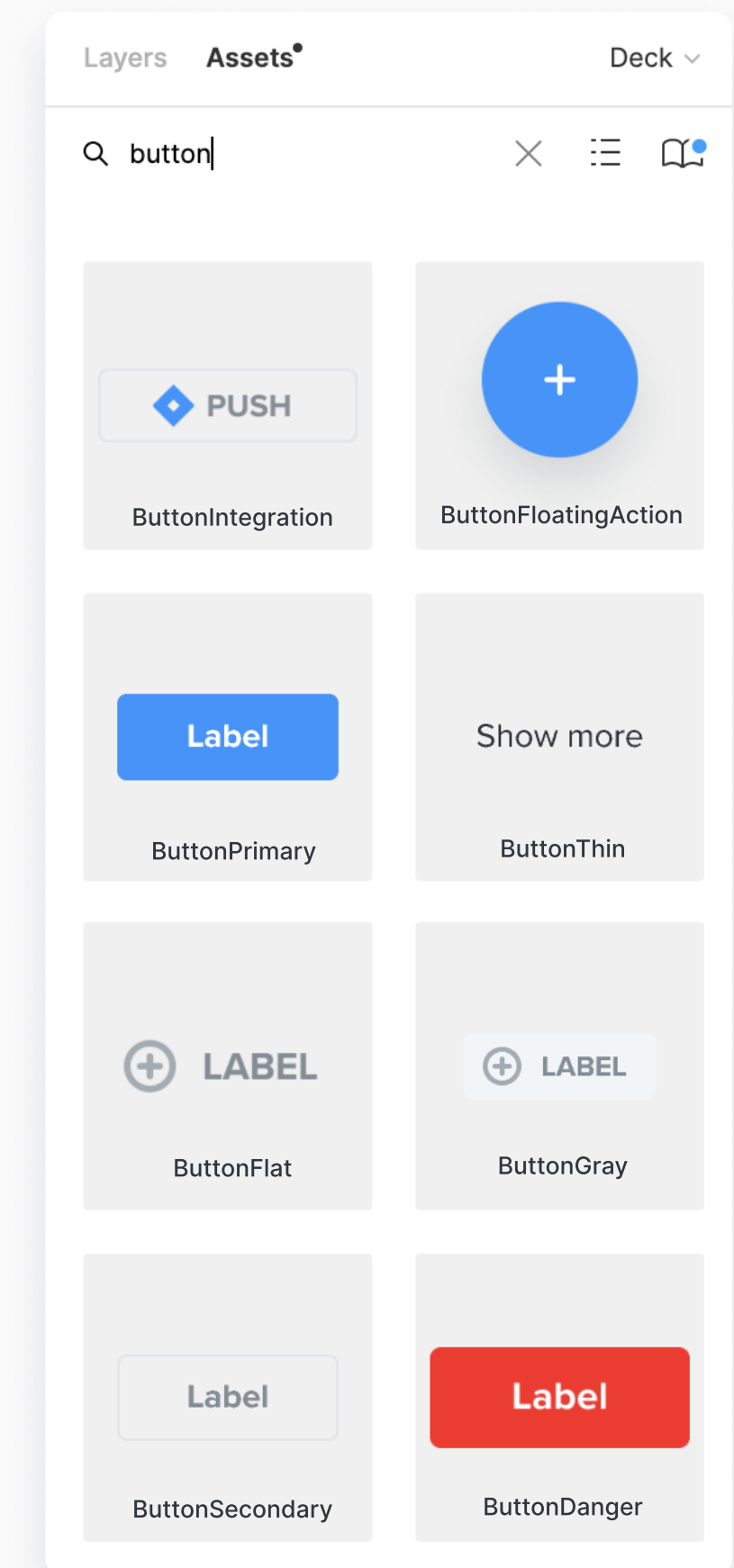
Having button types as variants would require more cognitive load for designers **when looking for a specific button.**

It will improve discoverability of all distinctive variants in the system and speeds up the work with UI kit.



DON'T

It's slow to select the right button 🐌

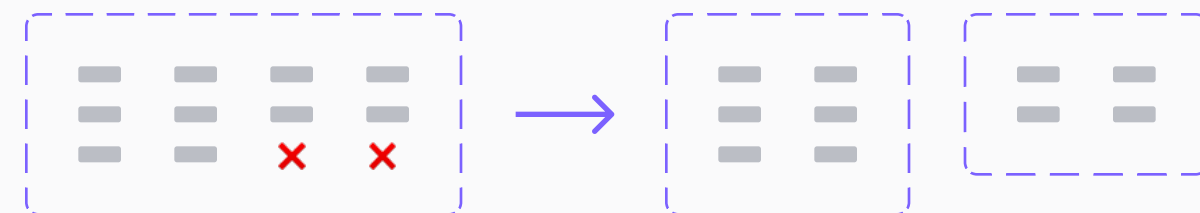


THE BETTER WAY

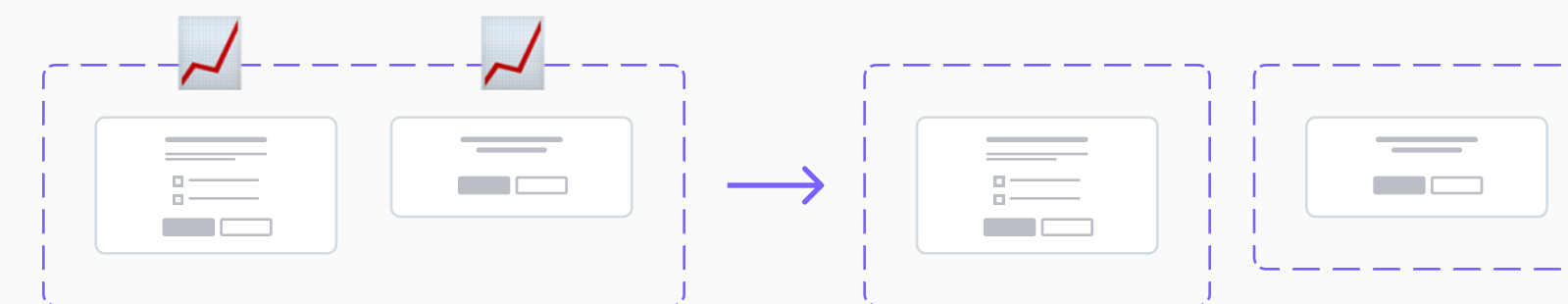
It's much faster for users 🚀



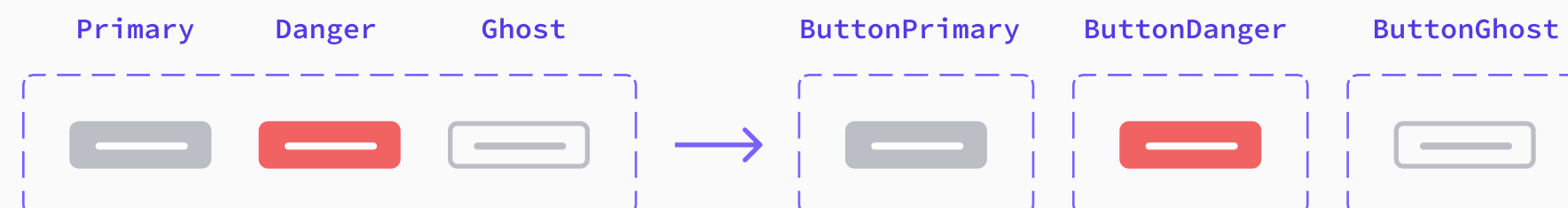
When to consider splitting variants into more components?



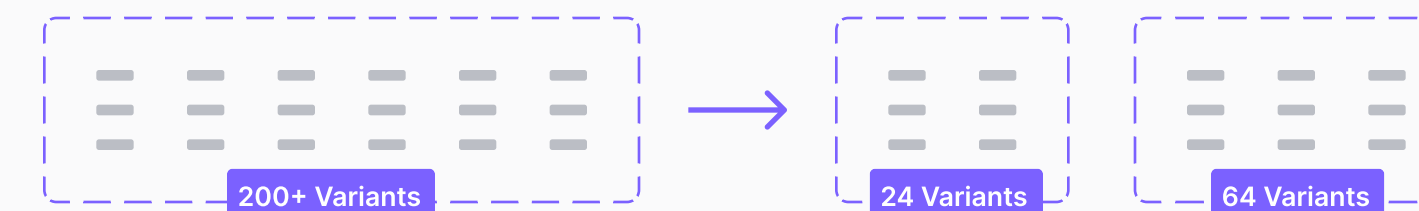
Some properties are not applicable to all component variants



Some key variants are more commonly used by consumers



Some common variants have visually distinctive types or different use cases

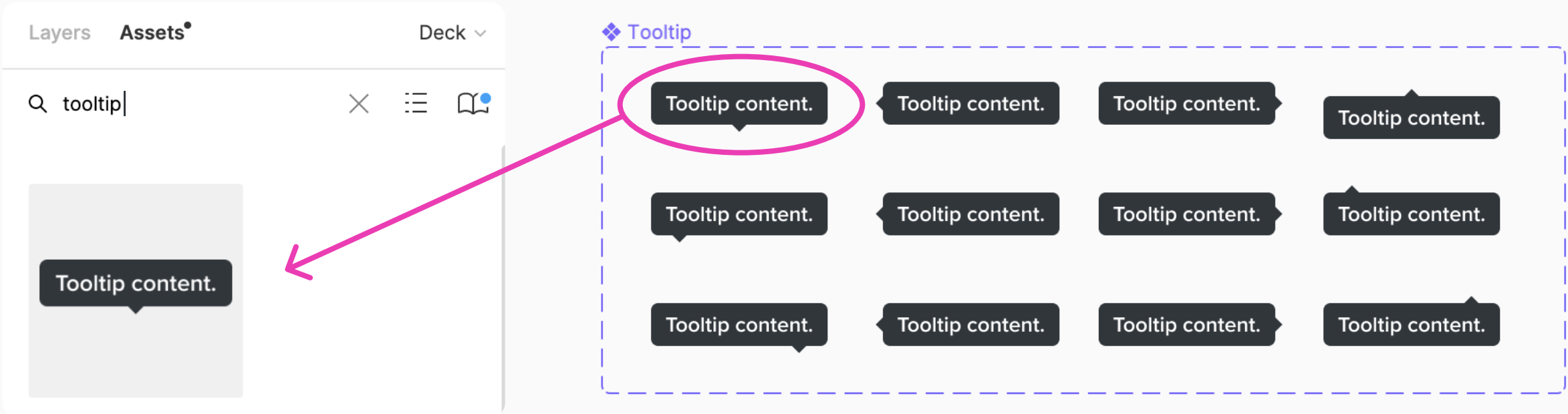









Some variant property causes large number of permutations

Set the most frequent variant as default

Is centered tooltip the most commonly used?
Are people using the small buttons the most often?

Having it as a default variant saves clicks everytime
when a designer needs to use it.

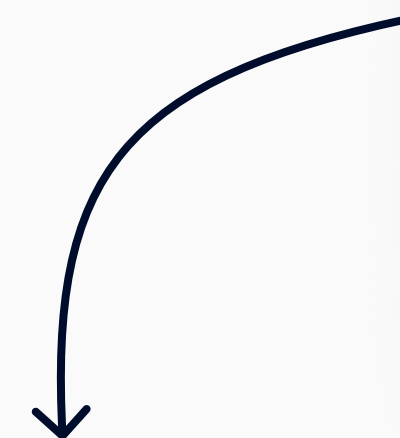


Libraries					×
< Tooltip					
	Description  Available in React.	Total 646 instances	Used by 3 teams	Used in 31 files	
All variants					
Variant name		Instances (total) ↓ Detaches (last 30 Days)			
 Top, Center		362	0	>	
 Right, Center		175	0	>	
 Left, Center		43	0	>	
 Bottom, Center		30	0	>	
 Top, Start		12	0	>	

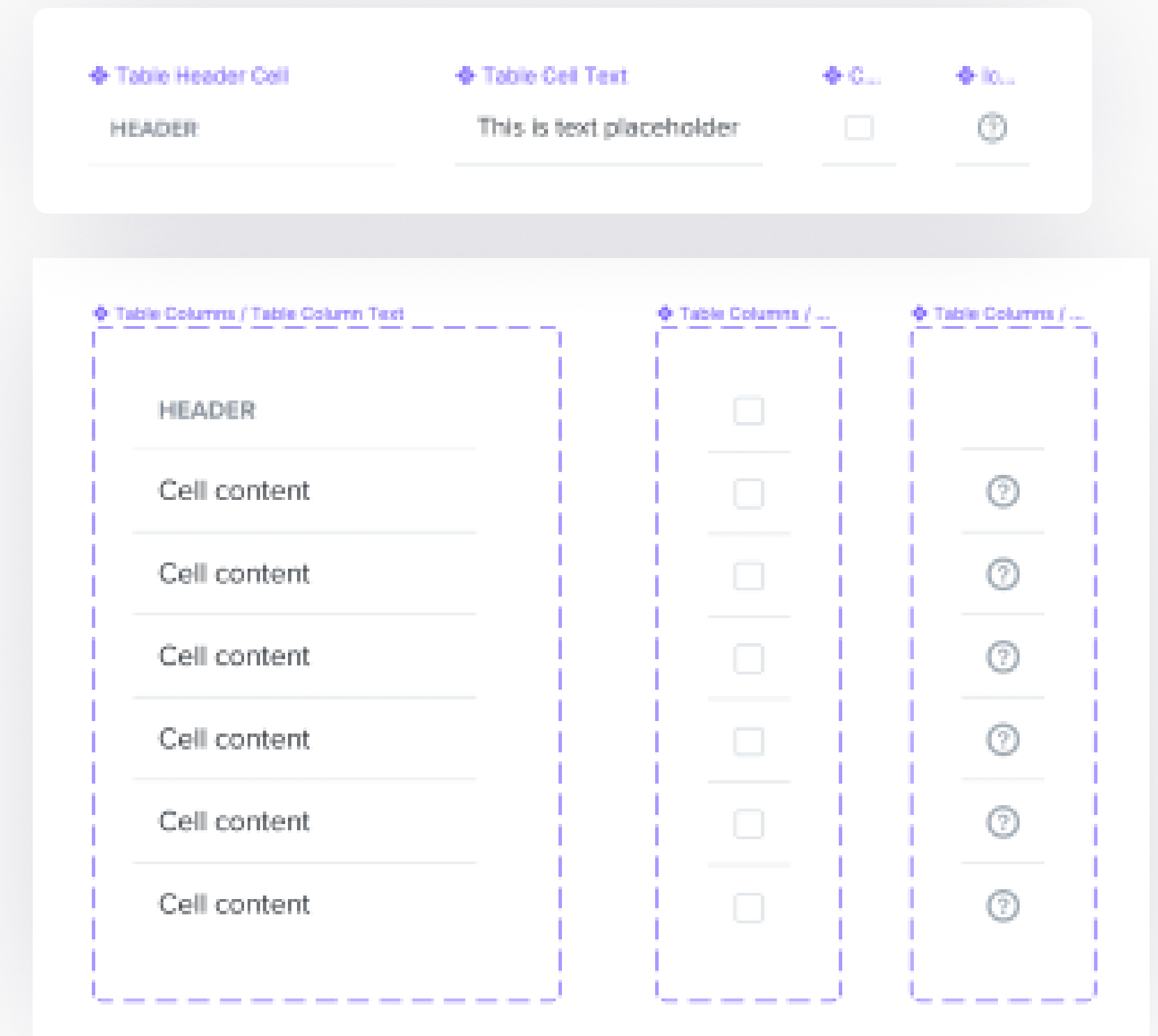
Provide common presets for complex components

Tables are great example of very complex component. It's hard to use for designers, plus there is no good support from Figma to handle tables.

How might we make it easier for designers to compose tables?



Defined in UI kit: different Table Cells + Table Columns



When used by consumers: effective way to compose tables



How might we create UI kits that...

...designers will love using?

...won't break with future changes?

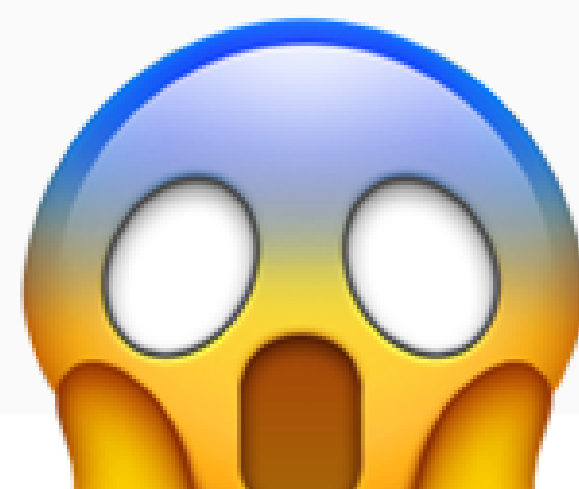
...will bridge code and design?



Component updates available

Review



Dismiss





Designers often don't update the library because they're afraid that their designs and prototypes will break with the update.





💔 Resetting overrides

 **Sugar Rush** | Vanellope von Schweetz | 1 



○○○ So that would be a nice feature - to be able to revert back to old views - to have just in case... similar to how google drive works

 **Hogwarts** | Rowena Ravenclaw | 6 

●●● Hi is there the ability to do a multi-select dropdown field? For example, I want to assign engineering teams to the feature...

 **Berk Village** | Toothless | 1 

●●● Hey, I just forgot to mention. Another thing that bothers me is that I need to look to the right side of the screen to process the notes.

 **Avengers Tower** | Tony Stark | 1 


●●● UI is absolutely beautiful, but sometimes the grey-on-white is somewhat hard to read.



Before update






Update all






Default text 


 **User name** Company name | ●●● | 1 



Default text 

 **User name** Company name | ●●● | 1 

Default text 

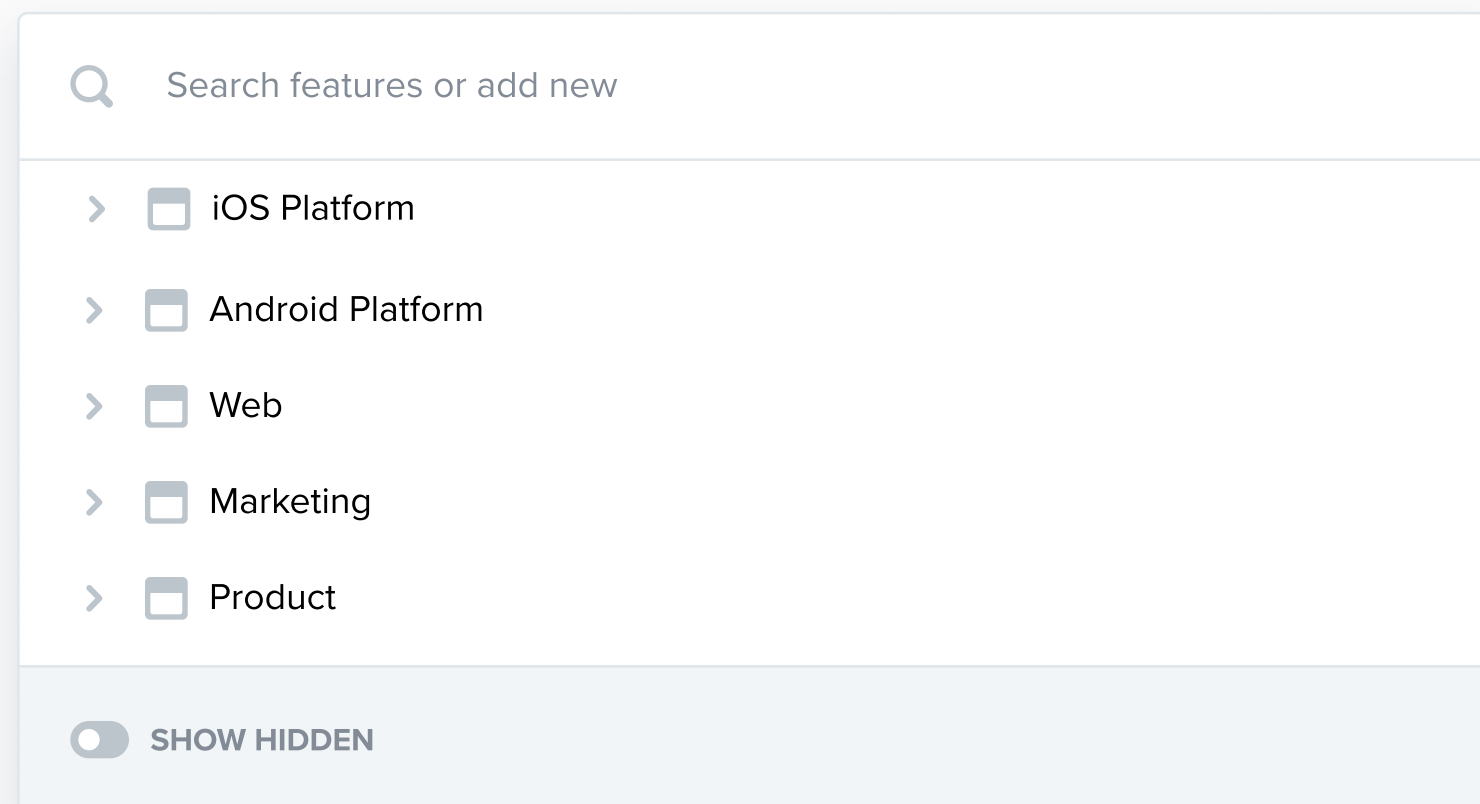
 **User name** Company name | ●●● | 1 

Default text 

 **User name** Company name | ●●● | 1 

After update

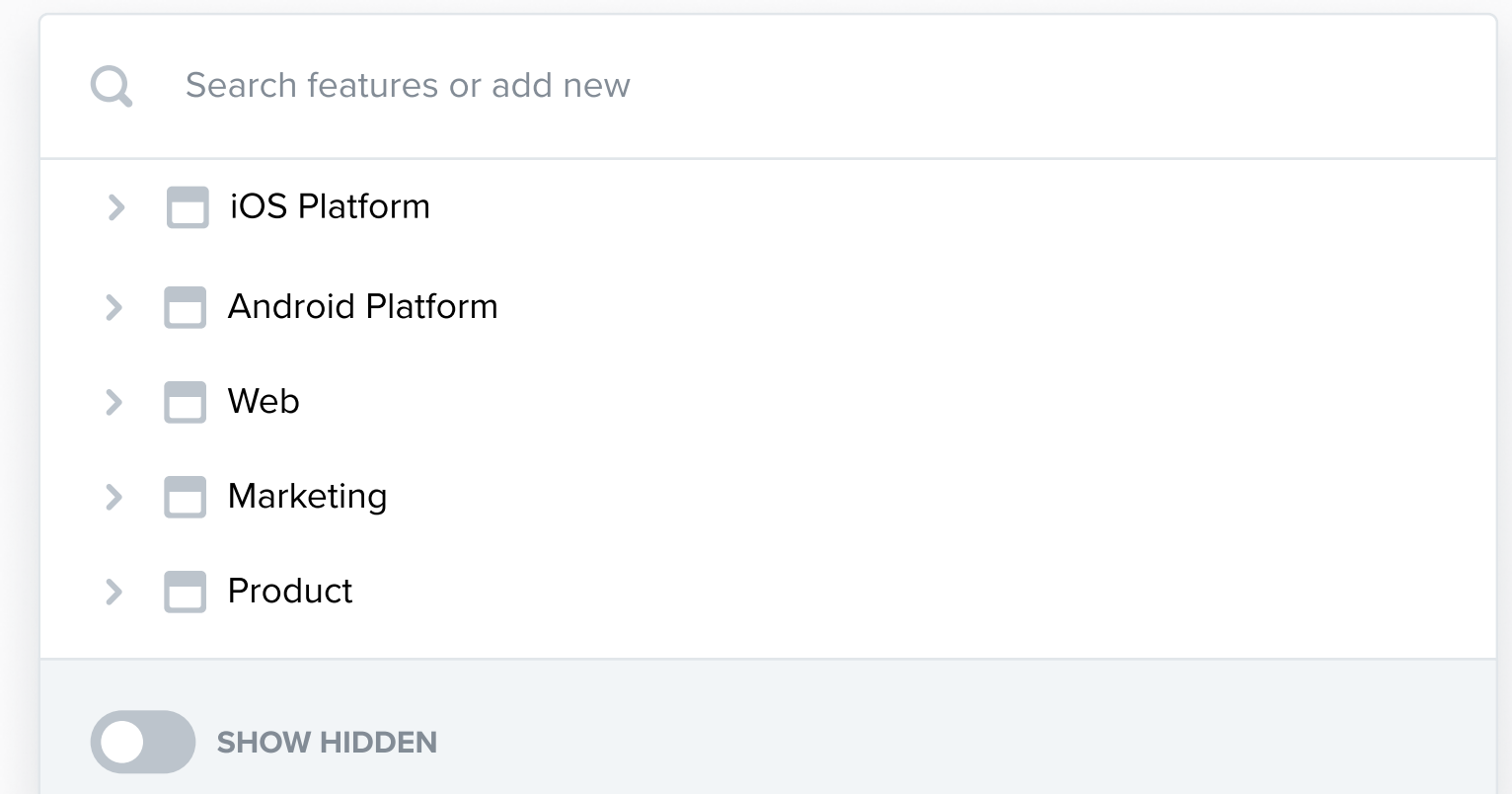
💔 Visual breaking change



Before update



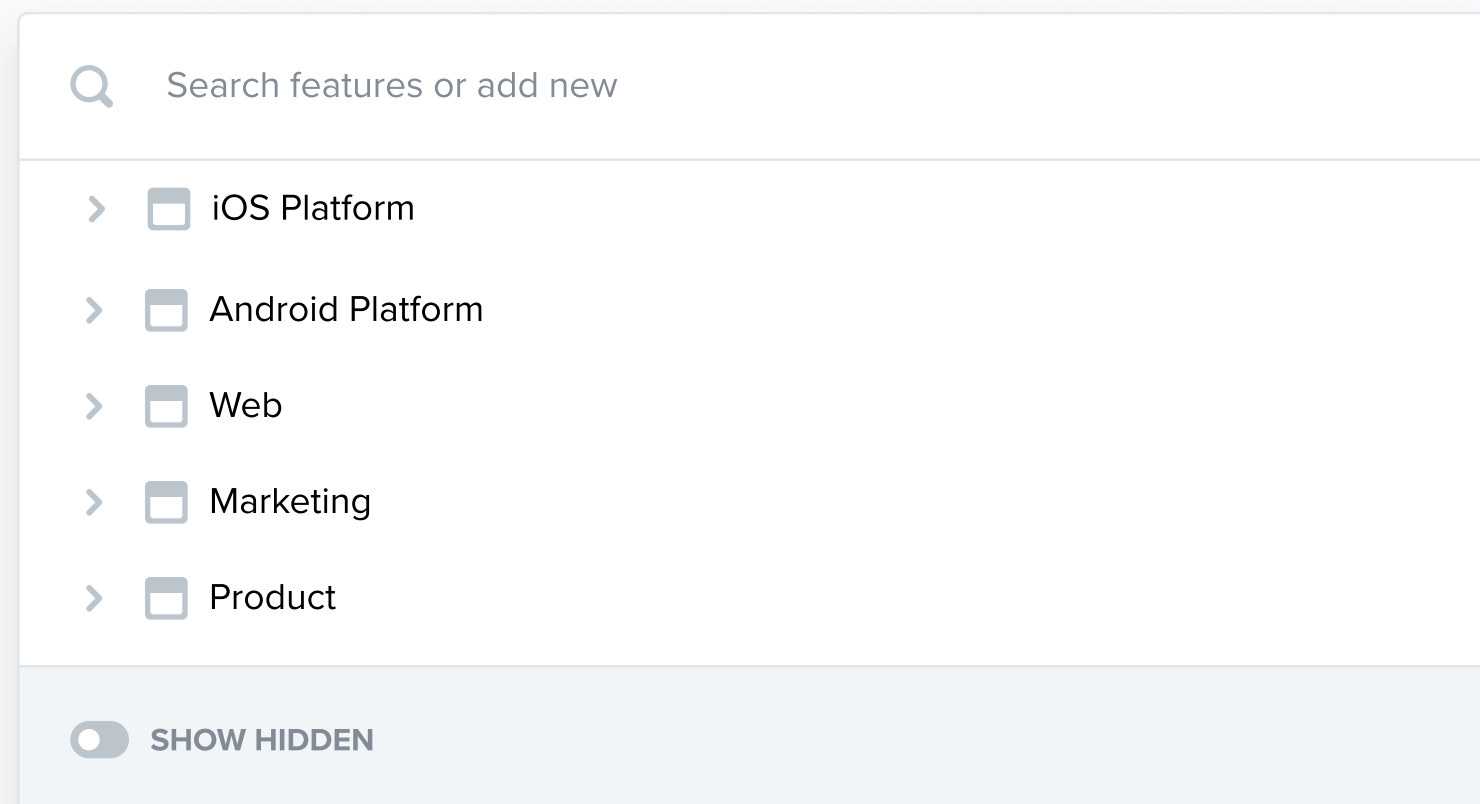
Update all



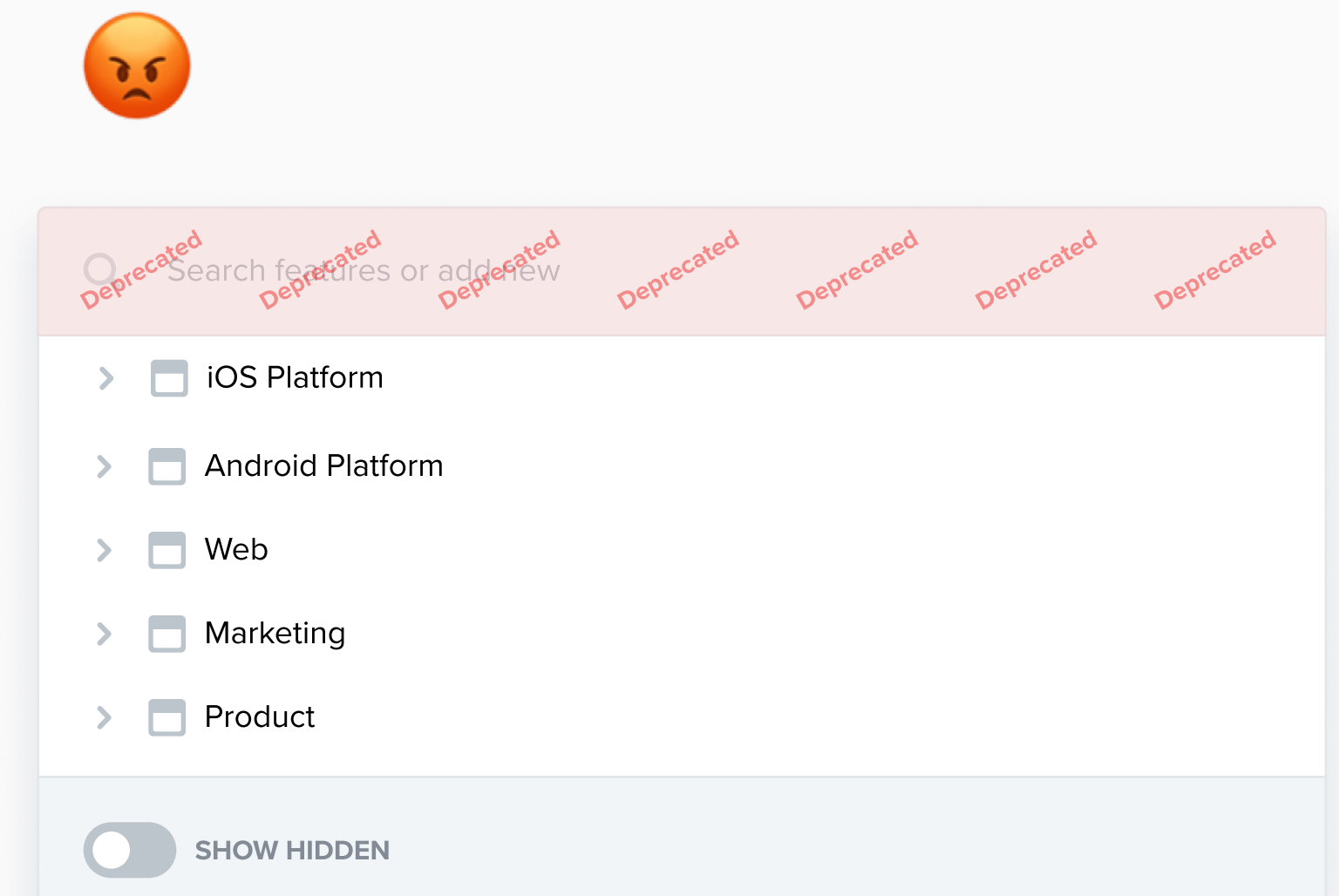
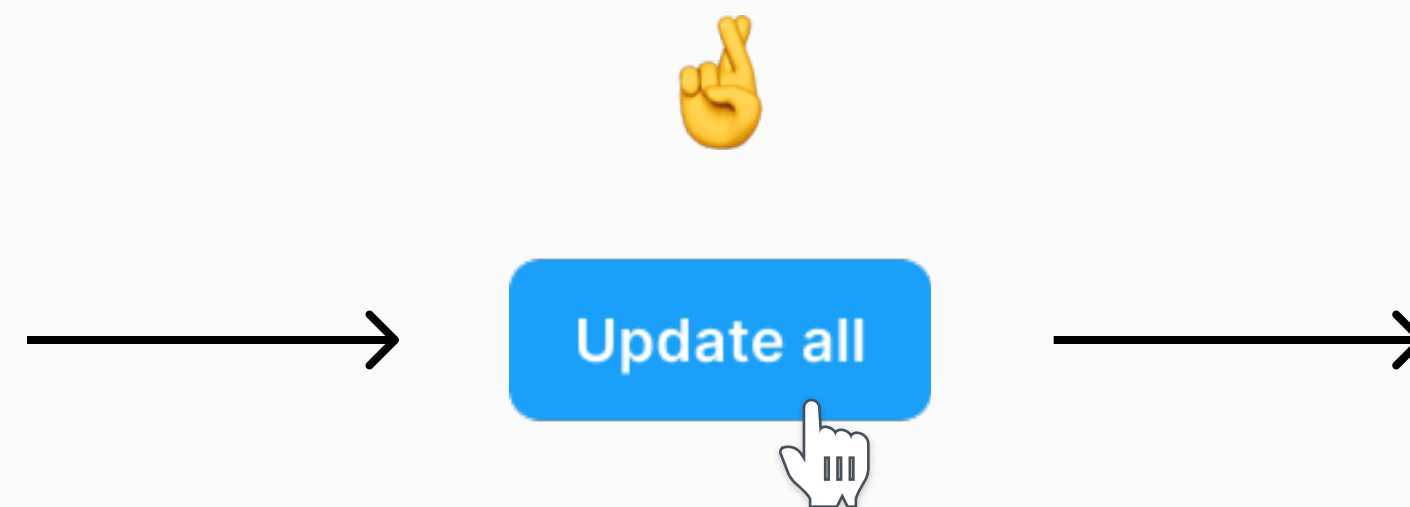
↑
Different size

After update

💔 Also visual breaking change



Before update



After update

Let's do some demos 🎮



Preventing breaking changes



Base components

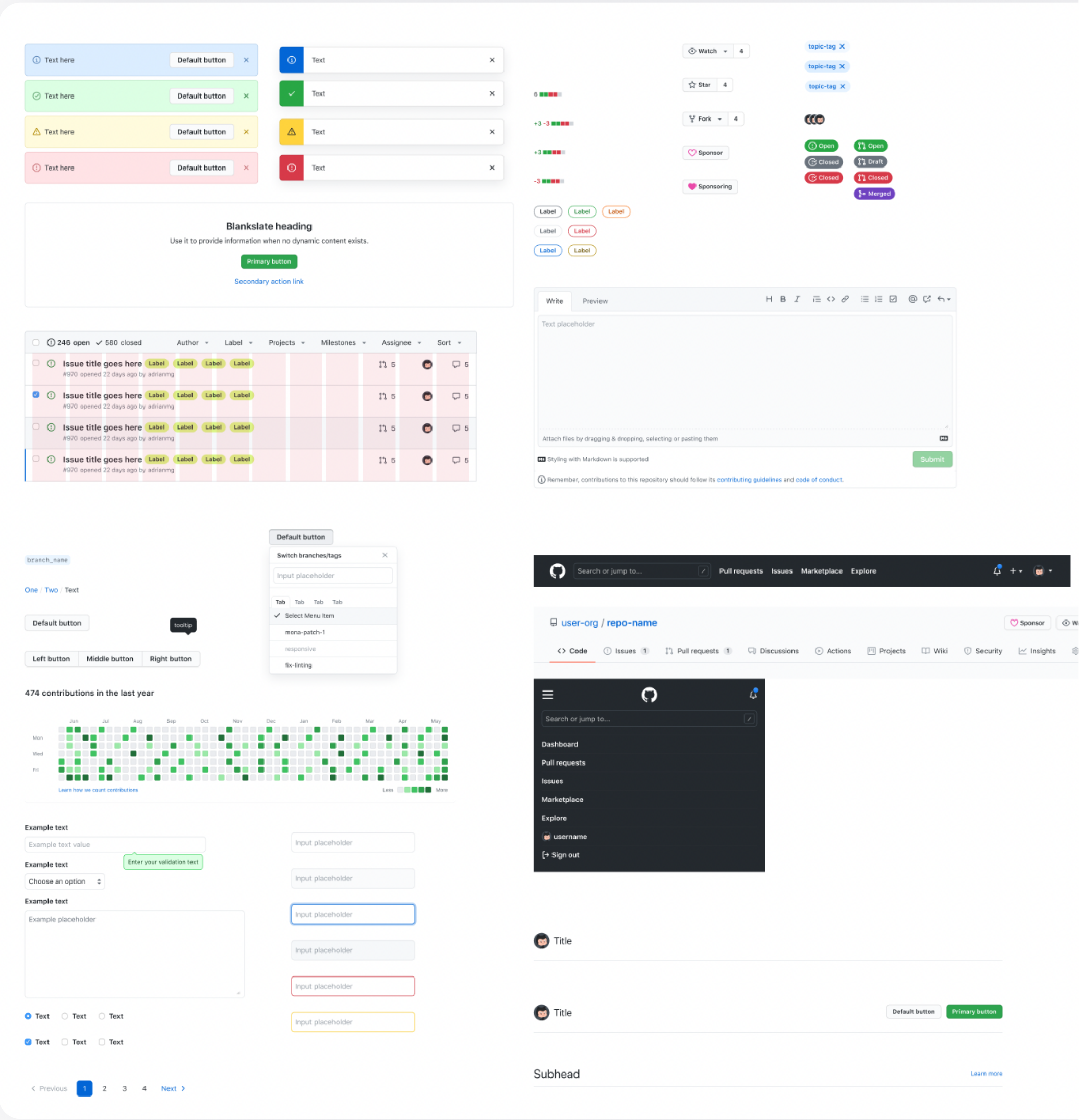


Canvas organization

Setup page with components examples

Great way to not just see all components in one place, but also seeing how they affect each other.

Bonus: It's prepared for possible automated visual regression testing in the future (e.g. via Figma REST API).



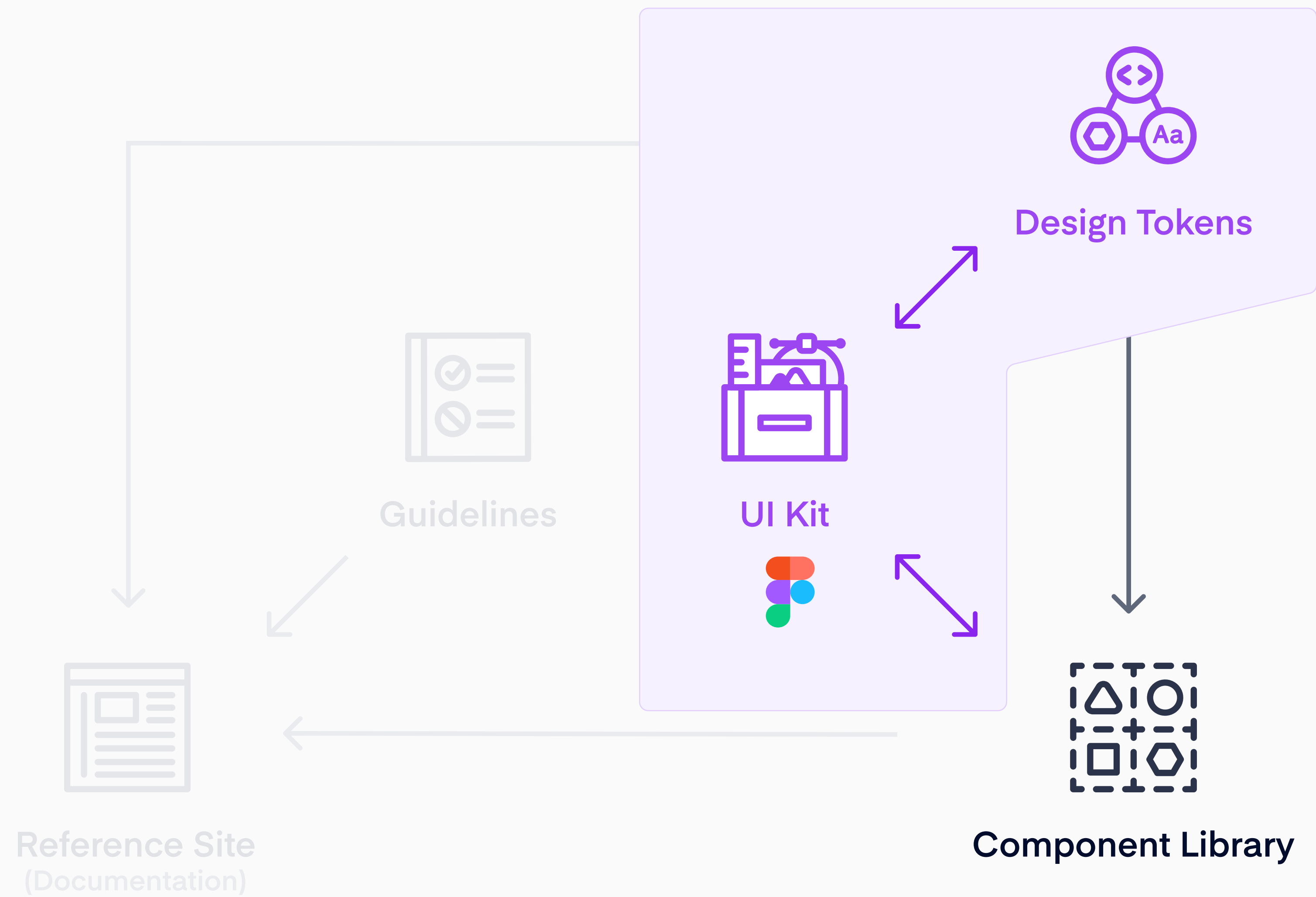
Source: [Github Primer Web UI kit](#)

How might we create UI kits that...

...designers will love using?

...won't break with future changes?

...will be aligned with code?

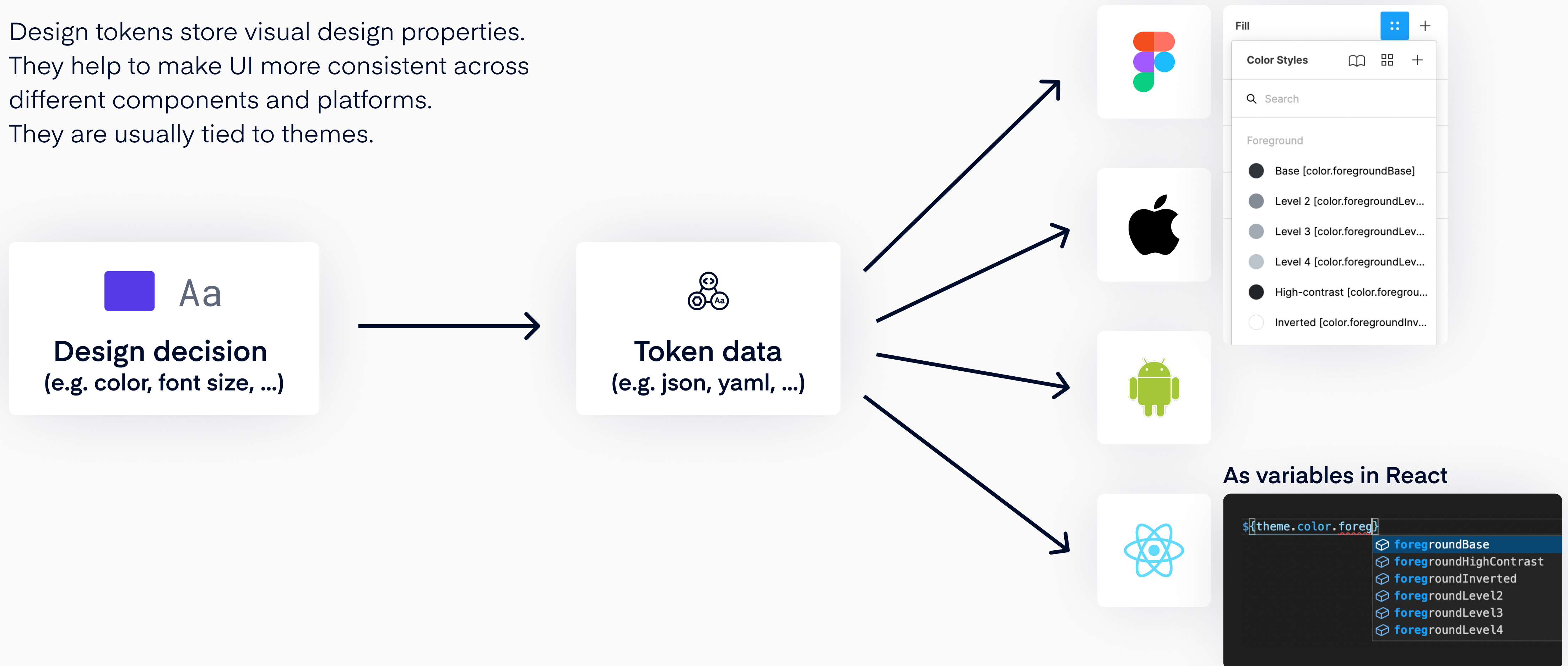


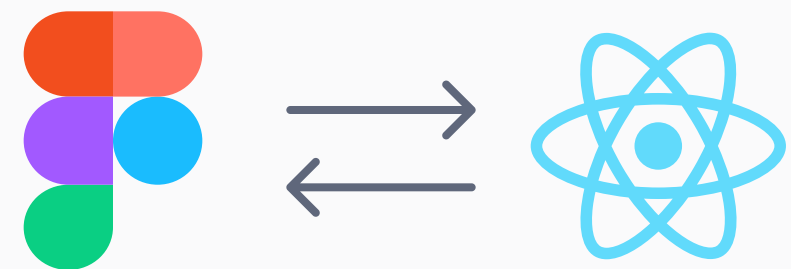
Design tokens



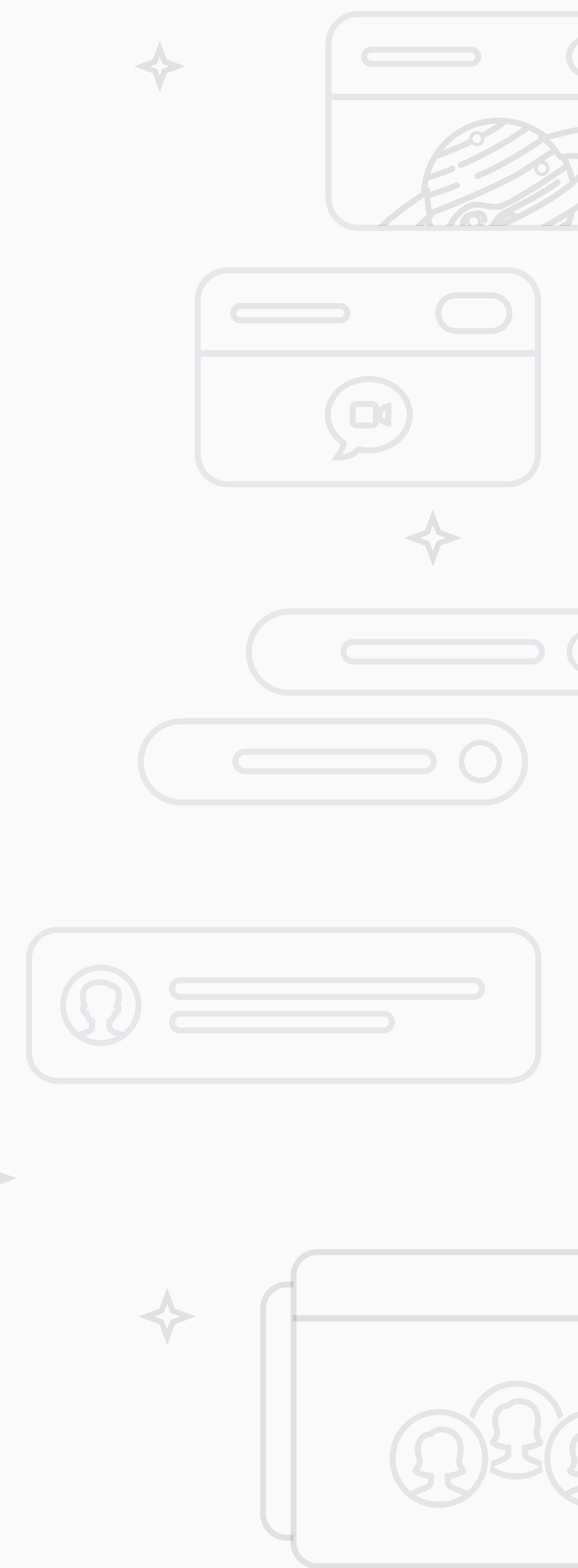
What are design tokens?

Design tokens store visual design properties.
They help to make UI more consistent across
different components and platforms.
They are usually tied to themes.





Tips and trick for aligning design tokens in Figma and in code

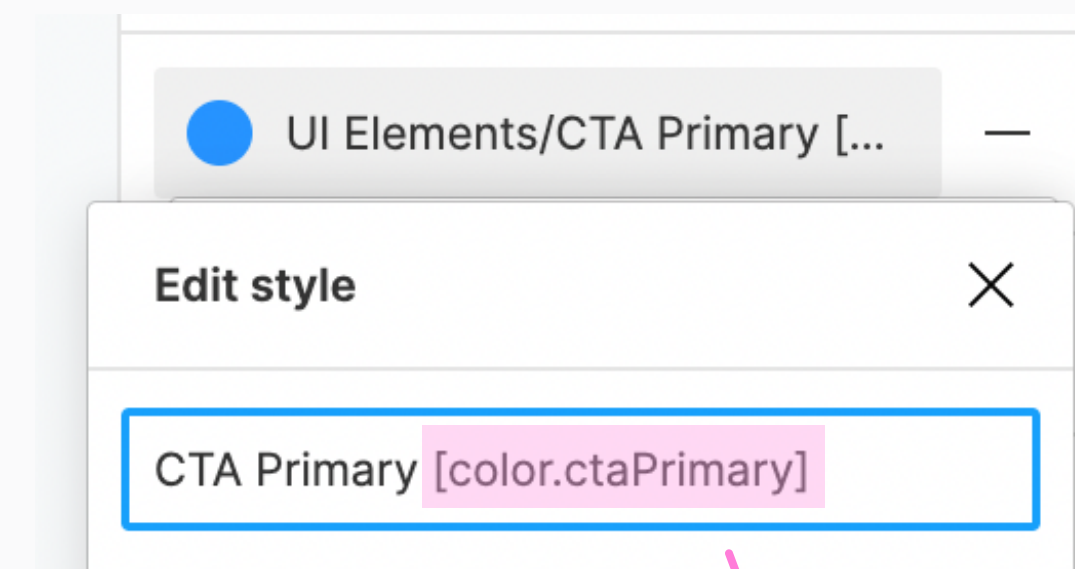


Always add name of the token to Figma style name

It will help developers to quickly get a name of the **code variable directly from the Figma Inspect panel**.

If you need to support more formats of code variables, pick the one that has the most developers in the company.


Note: Adding code variable name won't work, because descriptions are not visible in Figma Inspect panel.



```
/* UI Elements/CTA Primary  
[color.ctaPrimary] */  
background: #2693FF; 
```

Other examples

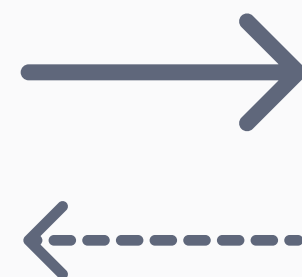
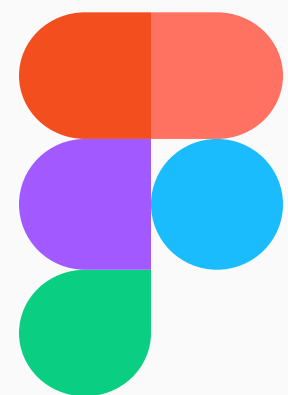
```
/* Elevations / Level 2 -  
Medium [boxShadow.level2] */  
box-shadow: 0px 3px 10px  
           rgba(63, 71, 79, 0.15);
```

```
/* Foreground/Base  
[color.foregroundBase] */  
color: #30363C; 
```

```
/* New brand / Midnight 900  
[color.midnight900] */  
color: #000C2C; 
```

```
/* Fields/Outline  
[boxShadow.fieldsOutline] */  
box-shadow: inset 0px 0px 0px  
           1px #D4DBE1;
```


Leverage tooling to keep design tokens in sync automatically



Figma tokens



Supernova



Design Tokens



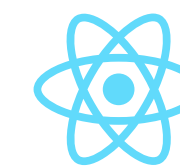
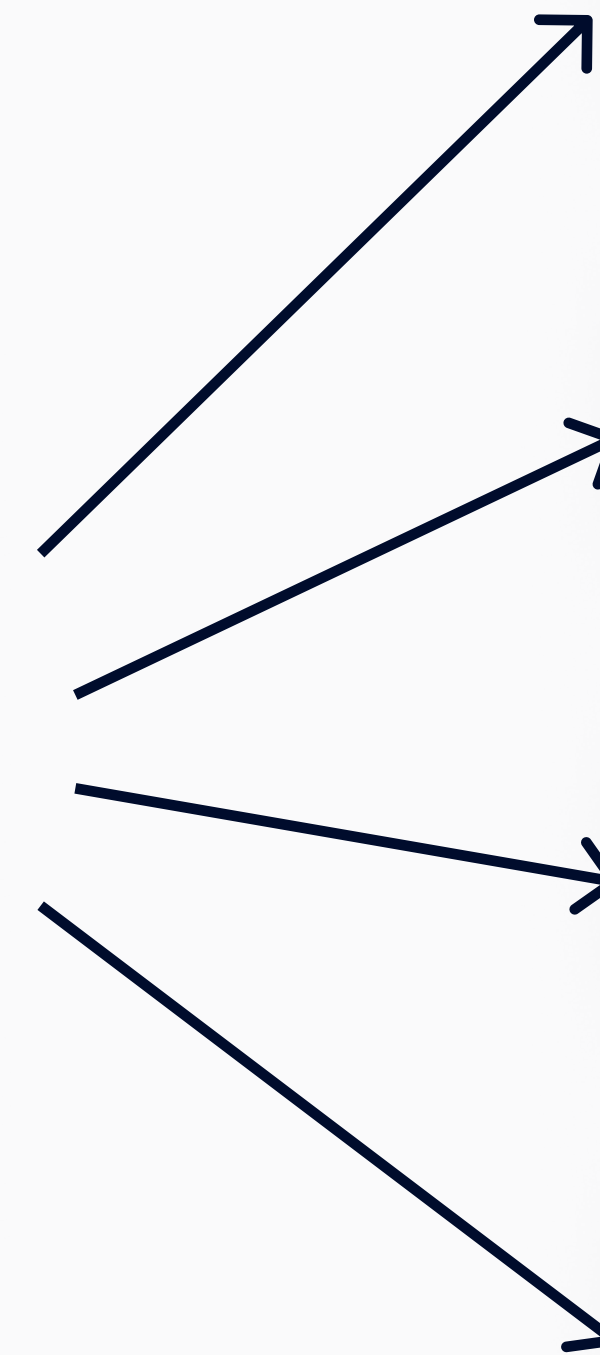
Specify



Knapsack

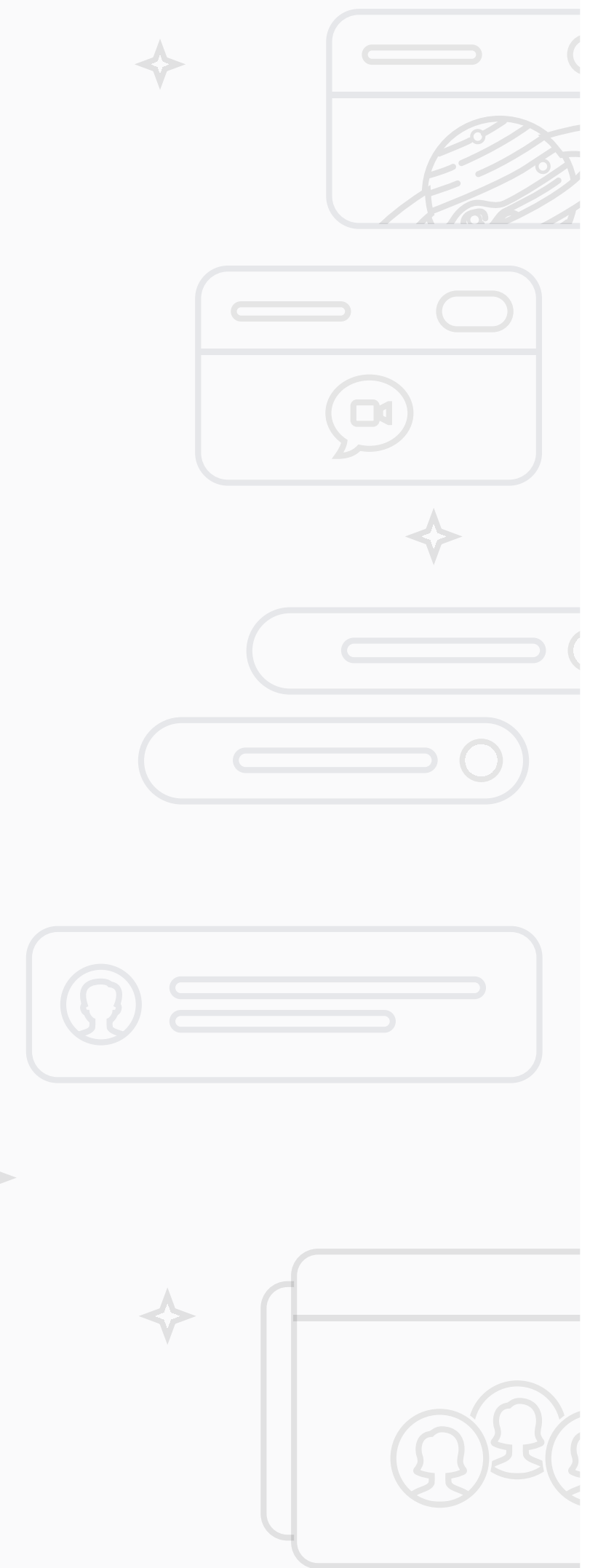


Arcade



...

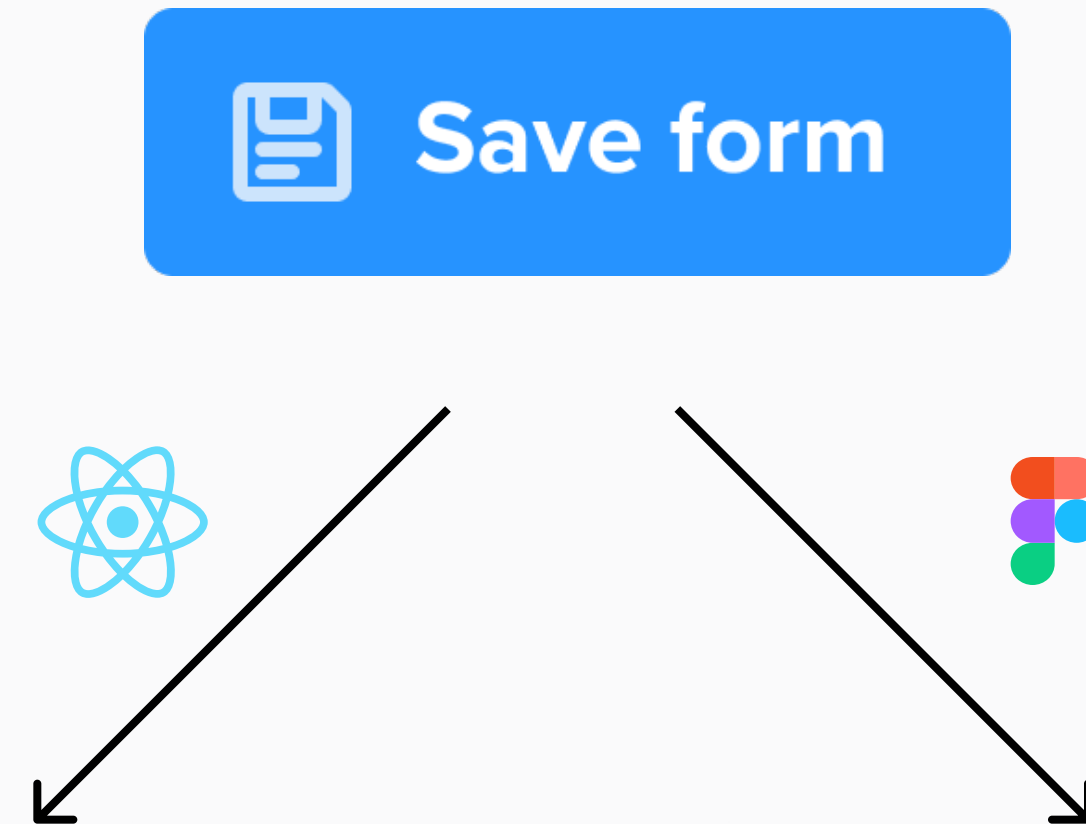
What about components?



Align component API across different implementations

It won't be perfectly aligned because of tooling gap and imperfections, but there is no need to have "type" in React and "variant" in Figma.

Make the connection clear whenever possible.



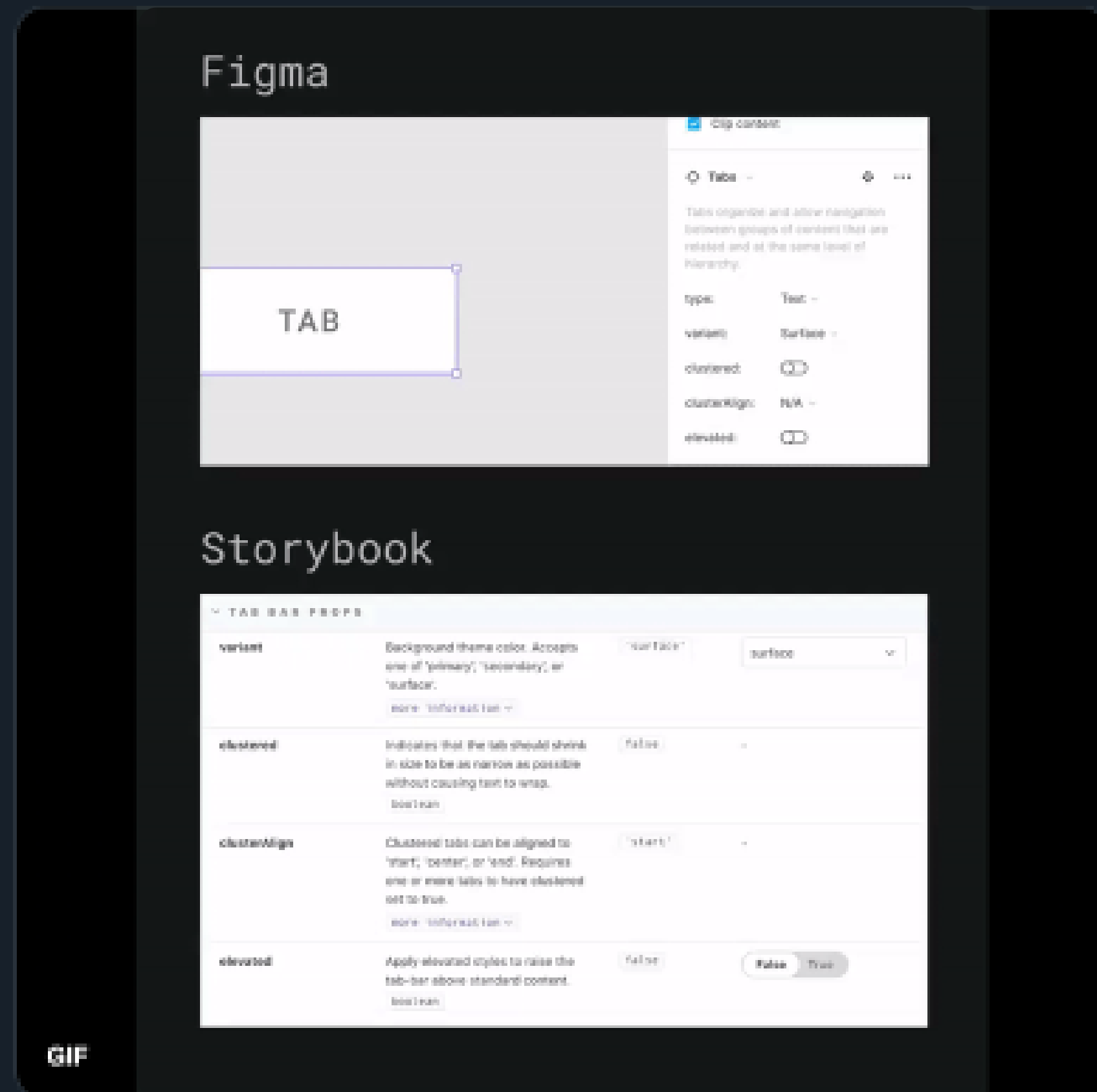
```
<ButtonPrimary  
  icon="save"  
  size="large"  
  ...  
>  
  Save form  
</ButtonPrimary>
```

◇ ButtonPrimary ▾	◆ ...
✿ Available in React.	
✿ Icon:	<input type="checkbox"/>
✿ Size:	Default ▾
State:	Default ▾



Chris Doner
@chrisd008

I absolutely love how [@figmadesign](#)'s variants functionality has allowed us to better align with code.



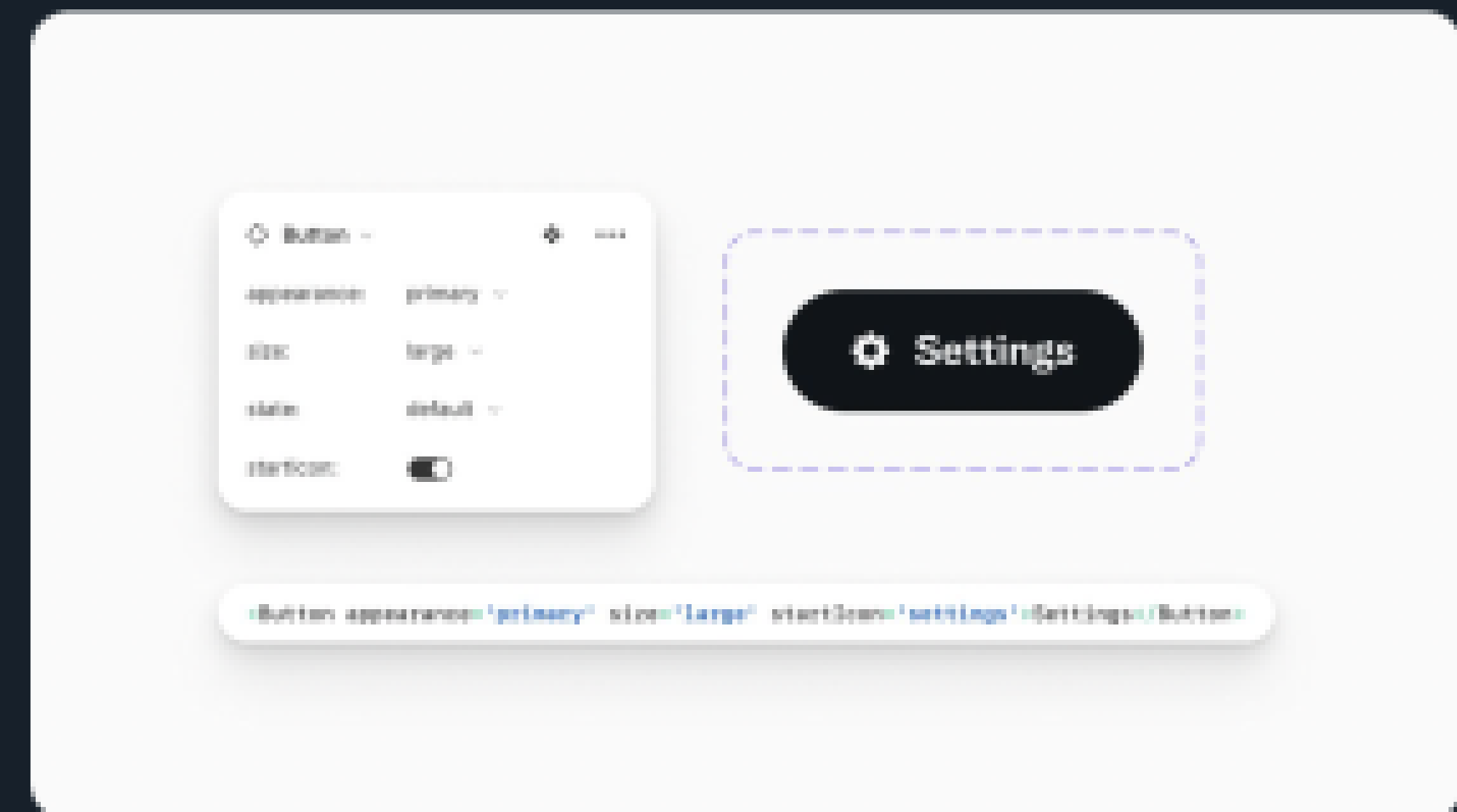
9:46 PM · Aug 2, 2021 · Twitter Web App

Source: <https://twitter.com/chrisd008/status/1422282553545957384>



Joey Banks
@joeyabanks

When it comes to working on design systems, one thing I've learned is how much of a positive difference it can make to match variant properties & names in Figma with what's reflected in code. Any opportunity to bring designers & engineers closer always feels like such a big win.



6:51 PM · Oct 1, 2021 · Twitter Web App

Source: <https://twitter.com/joeyabanks/status/1443981959533891586>



EightShapes

DISCOVERY

DESIGN

SYSTEMS

Crafting Component API, Together

How to unify anatomy and props across code and design tools



Nathan Curtis

Follow



Jul 17 · 8 min read



G

F

T C

T R



<D>

<E>

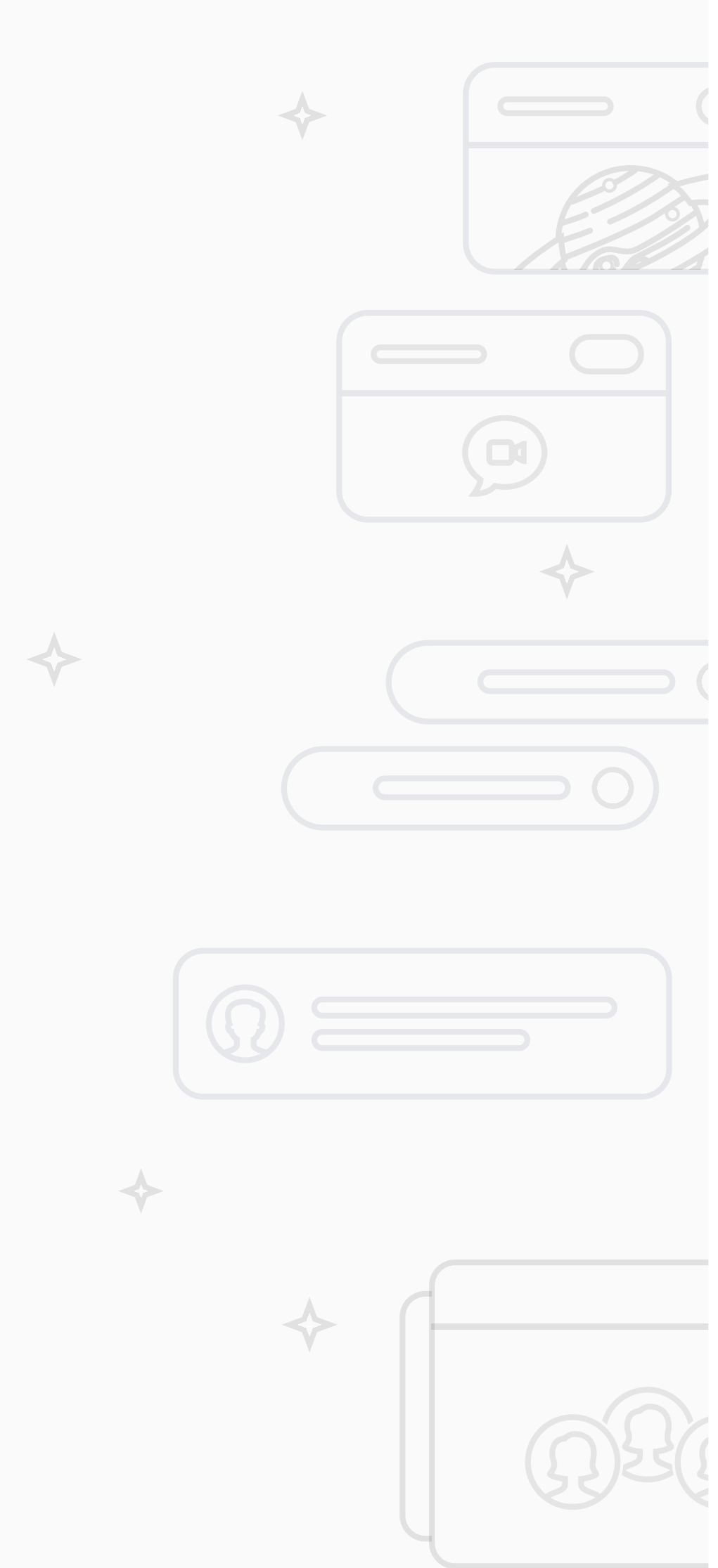
<F>

<G>

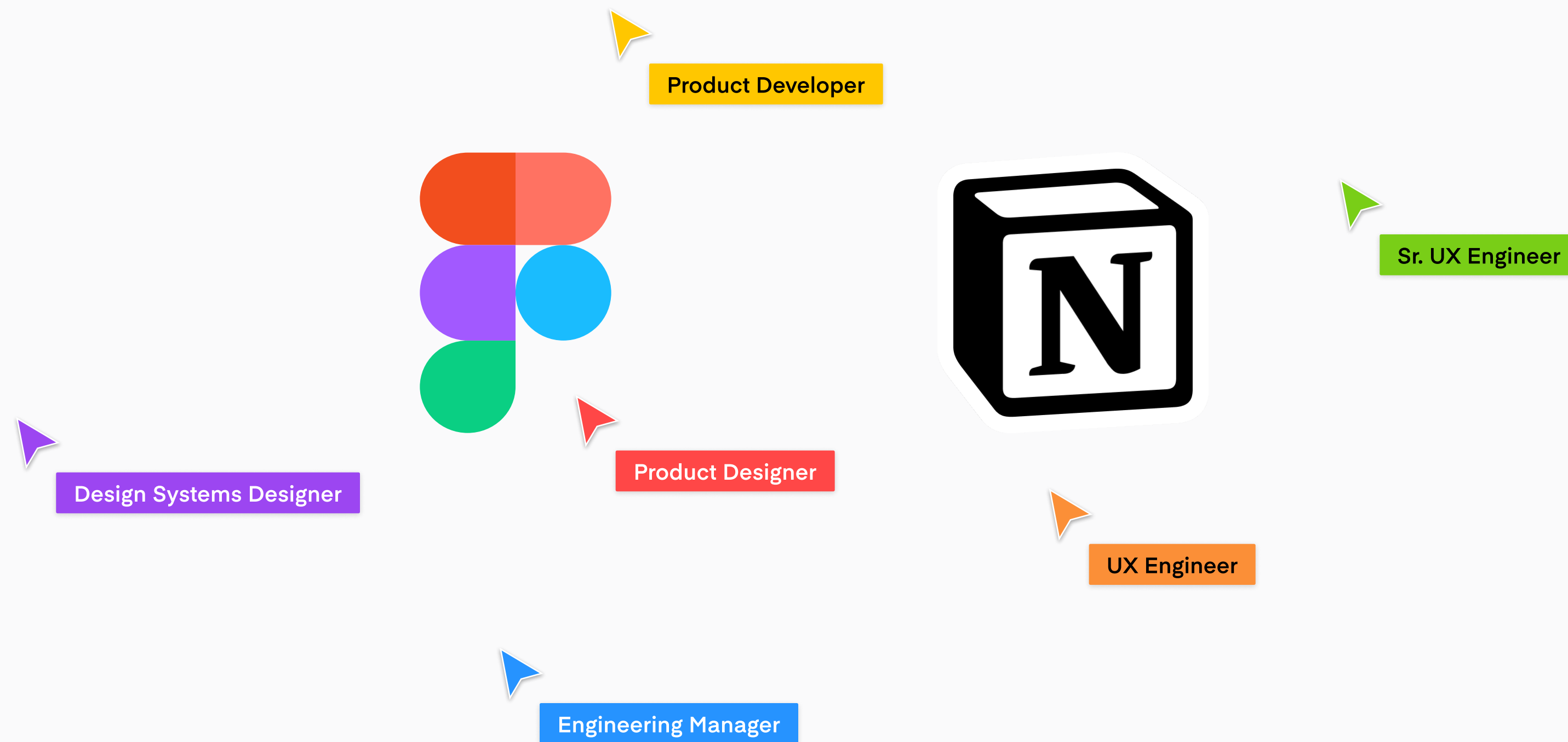
<https://medium.com/eightshapes-llc/crafting-ui-component-api-together-81946d140371>



What's our process for bringing component API in Figma closer to our code?



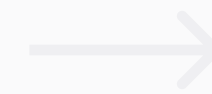
Designers collaborate on component API with developers



STEP 1



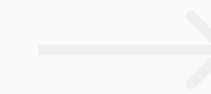
Design in Figma



STEP 2



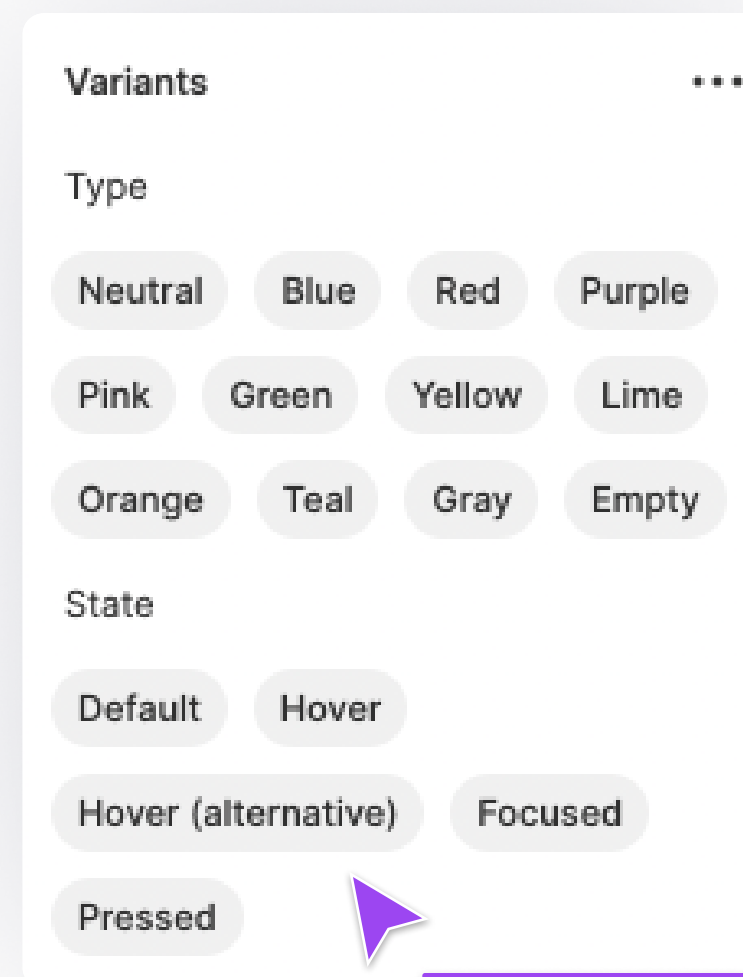
Redlines in Figma



STEP 3



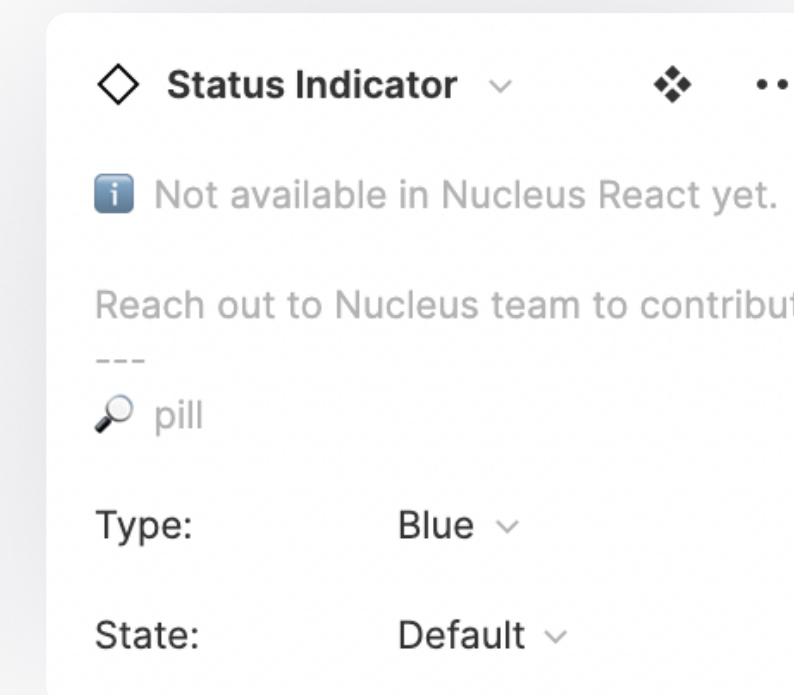
Specs in Notion



Design Systems Designer

Product Designer

Product Designer



STEP 1



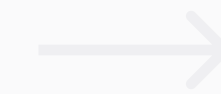
Design in Figma



STEP 2



Redlines in Figma



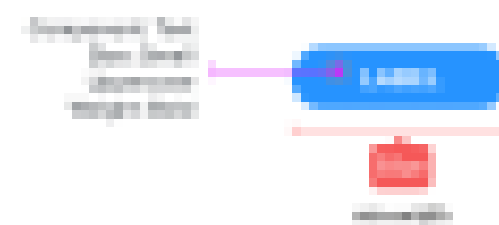
STEP 3



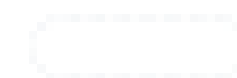
Specs in Notion

Status Indicator

Common properties

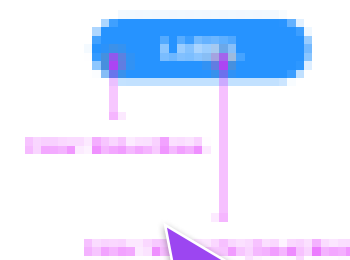


Empty state

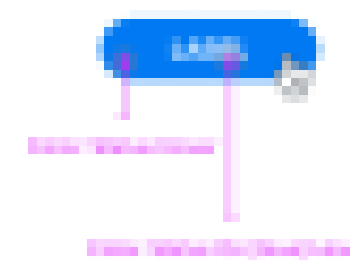


Colors

Default



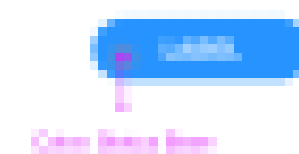
Hover



Focus



Pressed



Design Systems Designer

Engineering Manager

Alternative hover label

Status Indicator is used also for indicating a status from third party tools (e.g. Jira or Azure DevOps). In those cases, it is a link for the default of focus to that tool query tool. On focus, we show an alternative label (if other status). If the link is set to external, it should display the icon too.

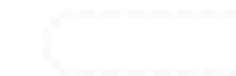
Default



Hover



Default

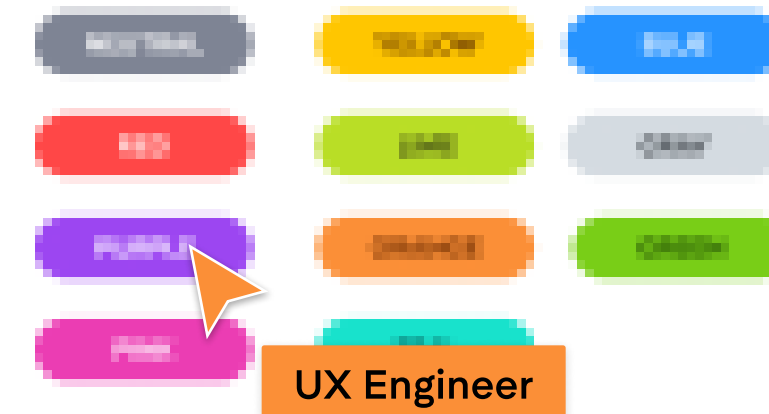


Hover



Sr. UX Engineer

Types



UX Engineer

With tooltip



STEP 1



Design in Figma



STEP 2



Redlines in Figma



STEP 3



Specs in Notion

Design specs

Design



Design Systems Designer

API proposed

```
type Props = Pick<
  ButtonPrimitiveProps,
  'external' | 'href' | 'onClick' | 'onBlur' | 'title';
> & {
  children: string; // differs from ButtonPrimitiveProps children type.
  isEmpty?: boolean;
  alternativeLabel?: string; // Label that is displayed on hover
  type: "neutral" | "yellow" | "blue" | "red" | "lime" | "gray" | "purple" | "orange"
} & GlobalProps & LayoutProps;
```

System Props

- color
- font

Additional props

Engineering Manager

```
type Props = Pick<
  ButtonPrimitiveProps,
  'external' | 'href' | 'onClick' | 'onBlur' | 'title';
> & {
  children: string; // differs from ButtonPrimitiveProps children type.
  isEmpty?: boolean;
  alternativeLabel?: string; // Label that is displayed on hover
  type: "neutral" | "yellow" | "blue" | "red" | "lime" | "gray" | "purple" | "orange"
} & GlobalProps & LayoutProps;
```

UX Engineer

Sr. UX Engineer

Product Developer

STEP 1



Design in Figma



STEP 2



Redlines in Figma



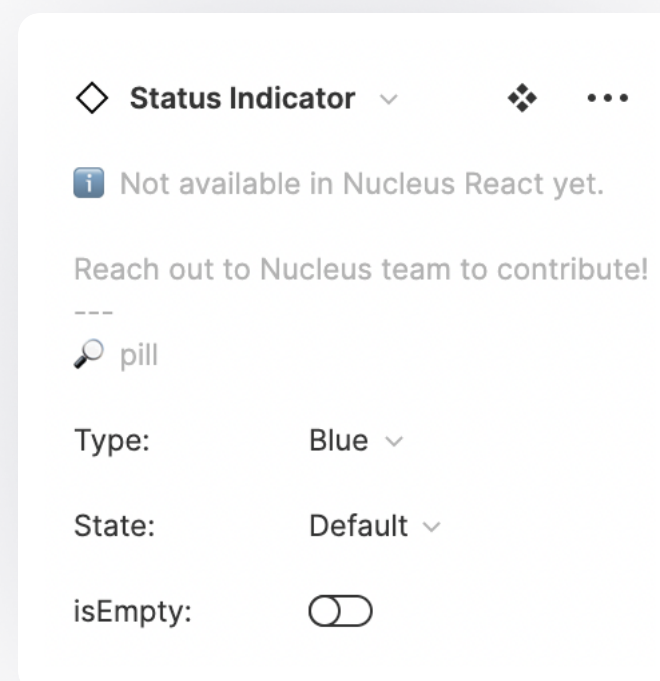
STEP 3



Specs in Notion

STEP 4

Adjust Figma variant properties
to match agreed component API.



Design Systems Designer

```
type Props = Pick<
  ButtonPrimitiveProps,
  'external' | 'href' | 'onClick' | 'onBlur' | 'title';
> & {
  children: string; // differs from ButtonPrimitive children type.
  isEmpty?: boolean;
  alternativeLabel?: string; // Label that is displayed on hover
  type: "neutral" | "yellow" | "blue" | "red" | "lime" | "gray" | "purple" | "orange";
} & GlobalProps & LayoutProps;
```



API consistency is great, but...

When not to align API in Figma with code?

- When it would make a component less usable for designers
- When it causes large amount of variant permutations (e.g. boolean/toggle properties)
- When it's not effectively doable in Figma (e.g. nesting content like in HTML)

Key takeaways



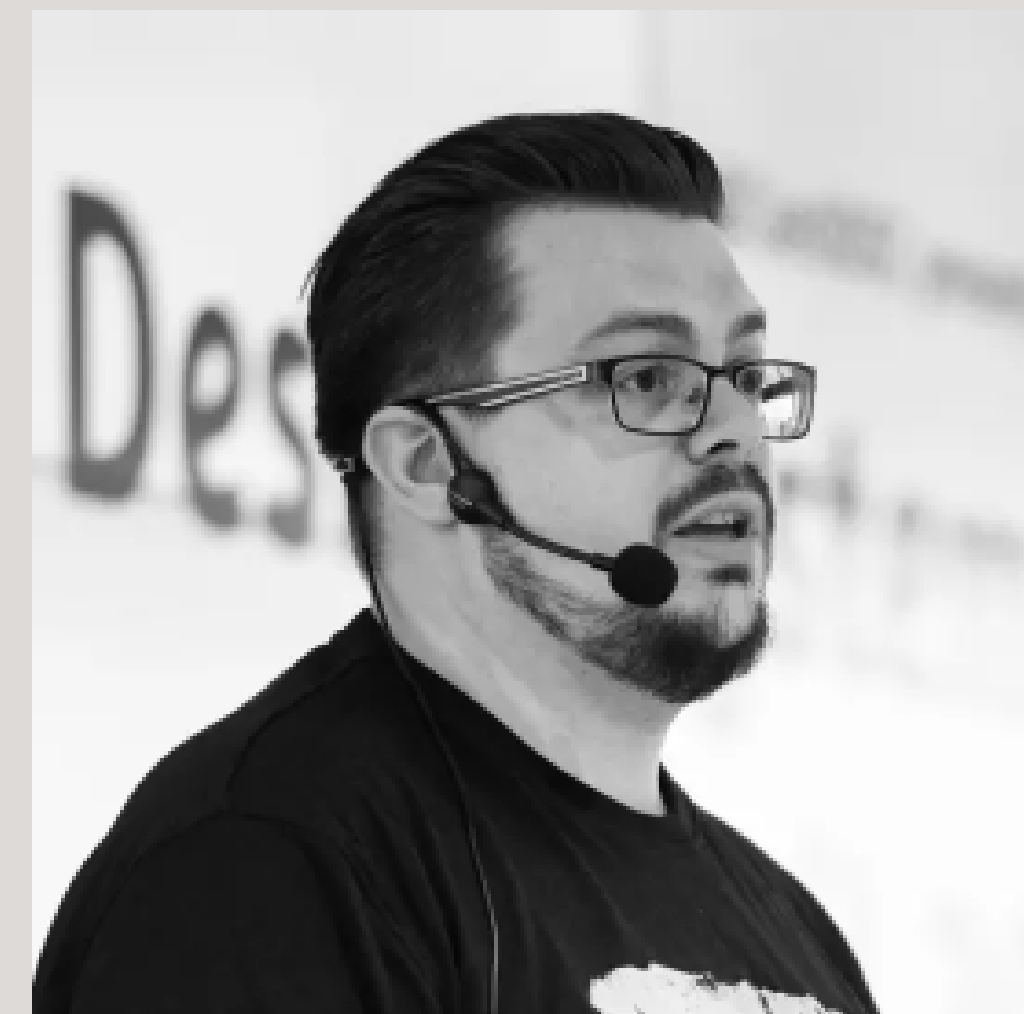
**Provide great UI kit experience
by preventing breaking changes**



**Optimize UI kit for usability
rather than for maintainability**



**Co-design component API
together with developers**



Jan Toman

Design System Lead
Productboard

 @HonzaTmn

schema
by Figma