



This is about Alva.
And how to integrate it.



Mario



Tilman

Alexander Peschel, Engineering

Julian Cebulla, Engineering

Julius Walther, Engineering

Lasse Küchler, Engineering

Mario Nebl, Engineering

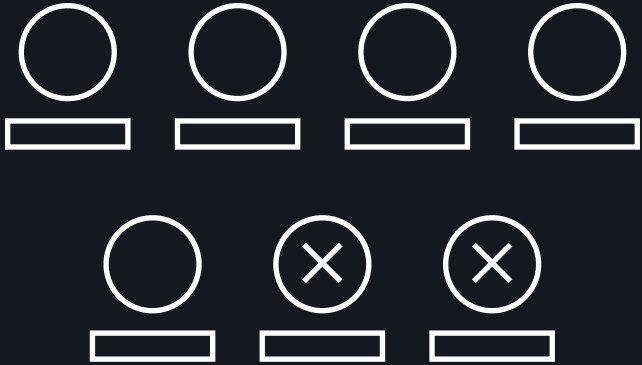
Markus Ölhafen, Design

Mauricio Palma, Engineering

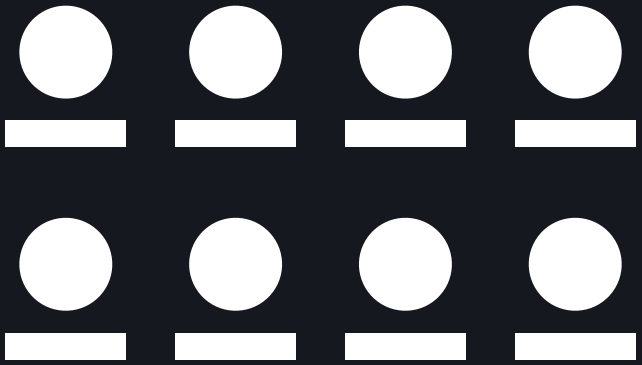
Thomas Jacob, Engineering

Tilman Frick, Lead

Static Design Library
for layouts & mockups



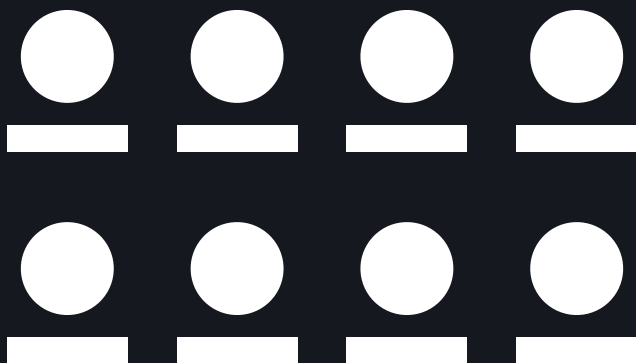
Interactive Code Library
for production websites



Interactive Code Library
for layouts & production websites



Design with Alva

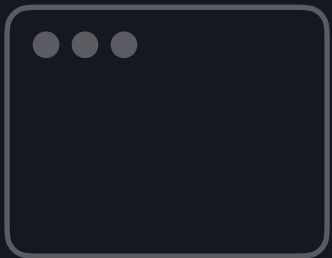
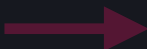
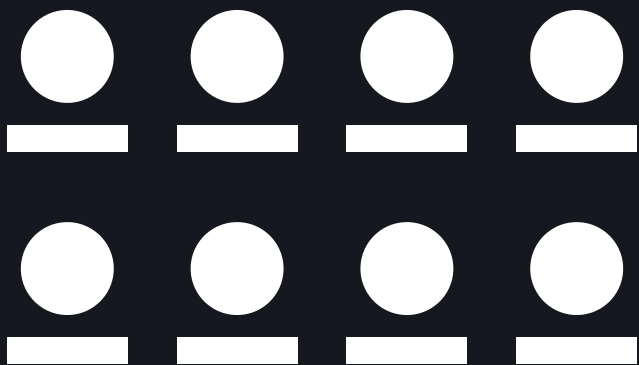


Production Websites

Interactive Code Library
for layouts & production websites



Design with Alva





Production Websites

WHAT ALVA DOES

Create living **prototypes** based on the same **components** engineers are using for **production** websites.

fn alva = components → data → app



fn alva =  components →  data →  app

REQUIREMENTS

Typescript

Language

+

React

As View Library

+

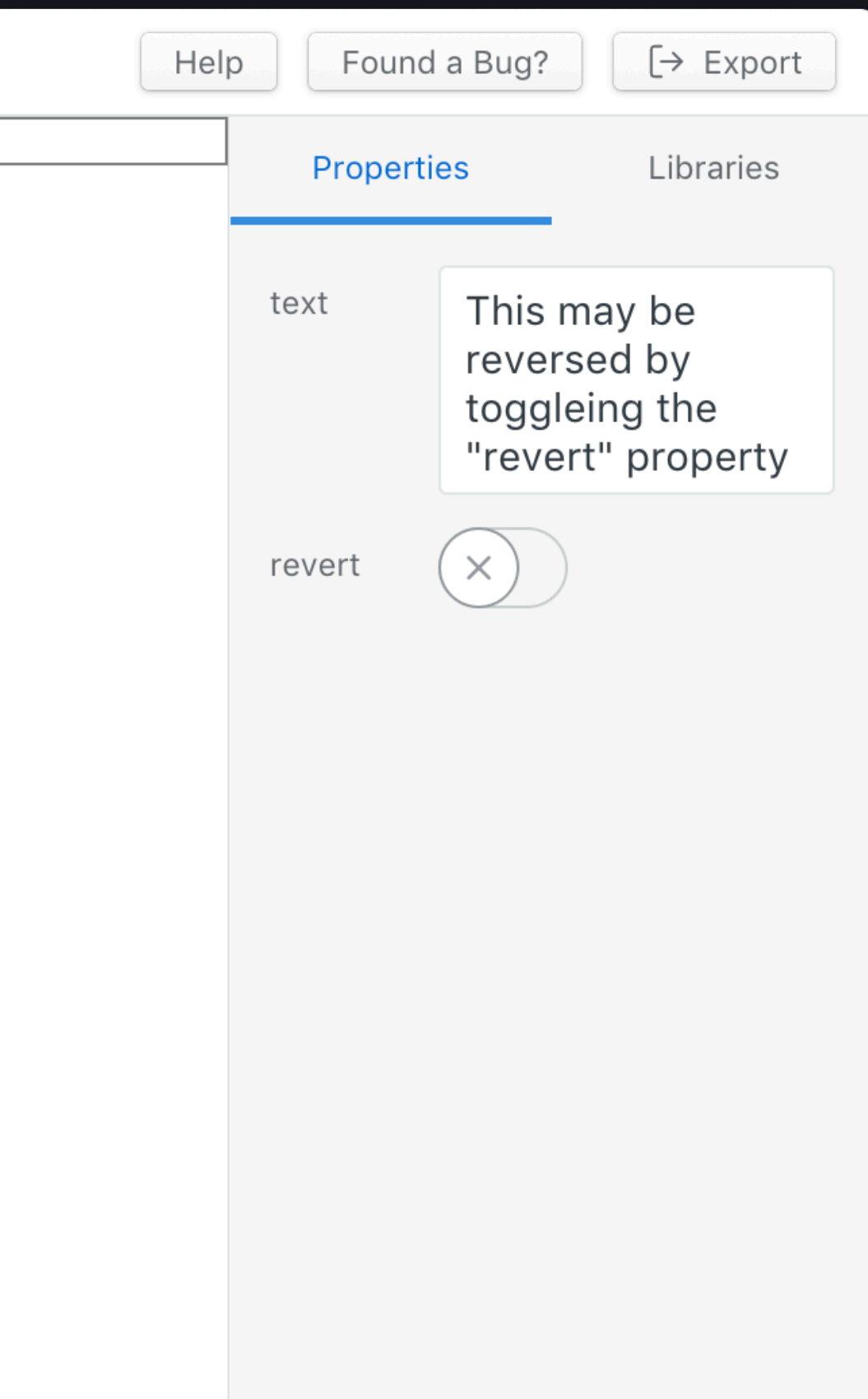
npm

Package

```
import * as React from "react";

export interface ReverserProps {
  text: string;
  /** @default false */
  revert: boolean;
}

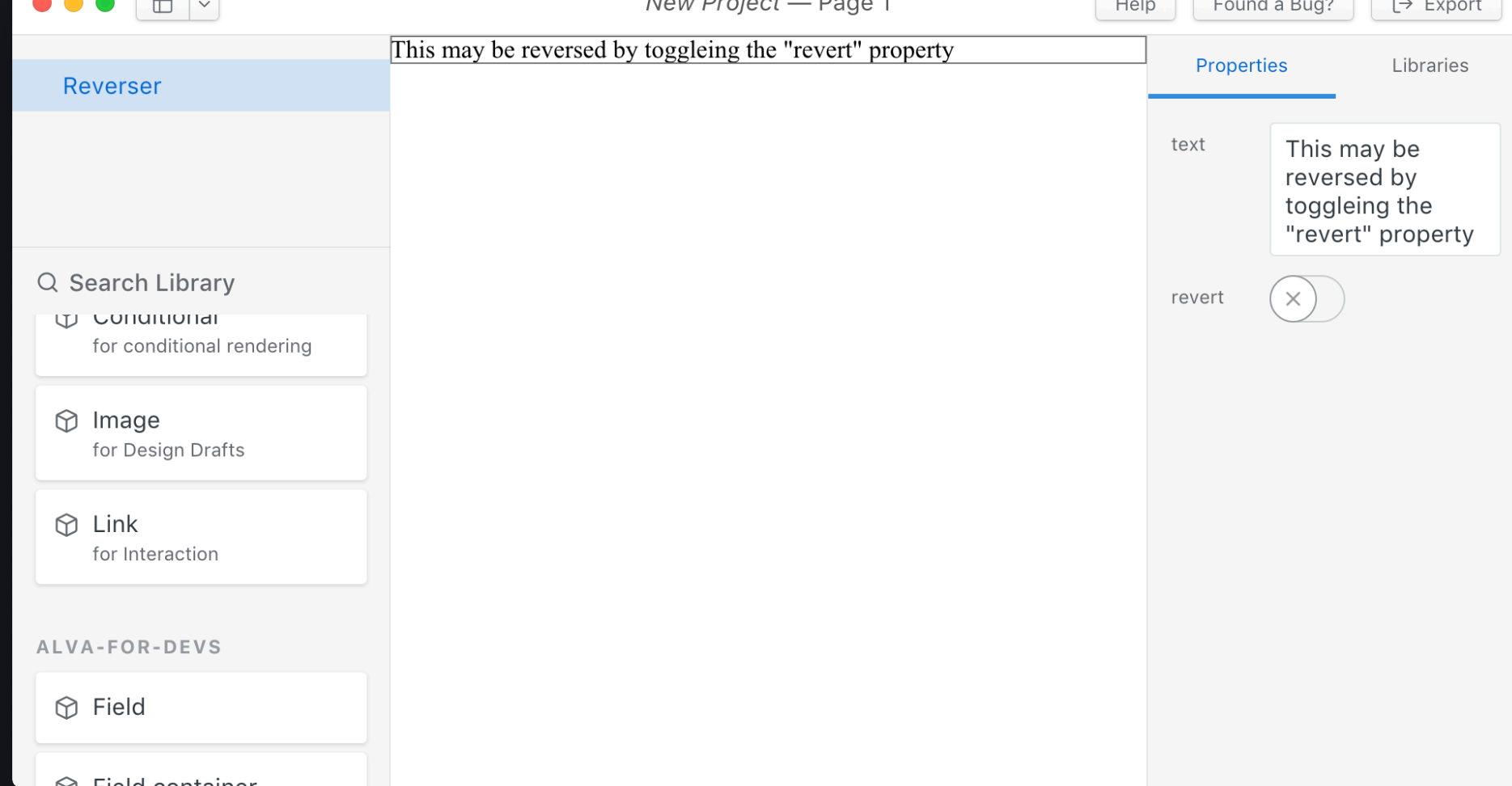
export const Reverser: React.SFC<ReverserProps> = props => {
  return (
    <div>
      {props.revert
        ? props.text
          .split("")
          .reverse()
          .join("")
        : props.text}
    </div>
  );
};
```



```
import * as React from "react";
```

```
export interface ReverserProps {  
  text: string;  
  /** @default false */  
  revert: boolean;  
}
```

```
export const Reverser: React.SFC<ReverserProps> = props => {  
  return (  
    <div>  
      {props.revert  
        ? props.text  
          .split("")  
          .reverse()  
          .join("")  
        : props.text}  
    </div>  
  );  
};
```



```
export const Reverser: React.SFC<ReverserProps> = props => {
  return (
    <div>
      {props.revert
        ? props.text
          .split("")
          .reverse()
          .join("")
        : props.text}
    </div>
  );
};
```



Demo

Use a Component

PROPERTIES

```
export interface ReverserProps {  
    /** @default false */  
    revert: boolean;  
}
```

PROPERTIES

```
export interface ReverserProps {  
    /** @default false */  
    revert: boolean;  
}
```


STRING

revert: string;

NUMBER

count: number;

BOOLEAN

active: boolean;

ENUM

state: State;

SLOT

left: React.ReactNode;

ANNOTATIONS

```
export interface ReverserProps {  
    /** @default false */  
    revert: boolean;  
}
```

```
NAME  
/** @name Revert */
```

NAME

@name Revert

DESCRIPTION

/** @description Reverts the text */

NAME

@name Revert

DESCRIPTION

@description Reverts the text

DEFAULT

/** @default false */

DESCRIPTION

@description Reverts the text

DEFAULT

@default false

EXAMPLE

```
/** @example Hello */
```

DEFAULT

```
@default false
```

EXAMPLE

```
@example Hello
```

SLOT

```
/**
```

```
@slot
```

```
*/
```

EXAMPLE

@example Hello

SLOT

@slot

ASSET

/**

@asset

*/

SLOT
@slot

ASSET
@asset

IGNORE
/** @ignore */

```
import * as React from "react";

export interface ReverserProps {
  text: string;
  /** @default false */
  revert: boolean;
}

export const Reverser: React.SFC<ReverserProps> = props => {
  return (
    <div>
      {props.revert
        ? props.text
          .split("")
          .reverse()
          .join("")
        : props.text}
    </div>
  );
};
```

SUPPORTED COMPONENT TYPES


React.SFC

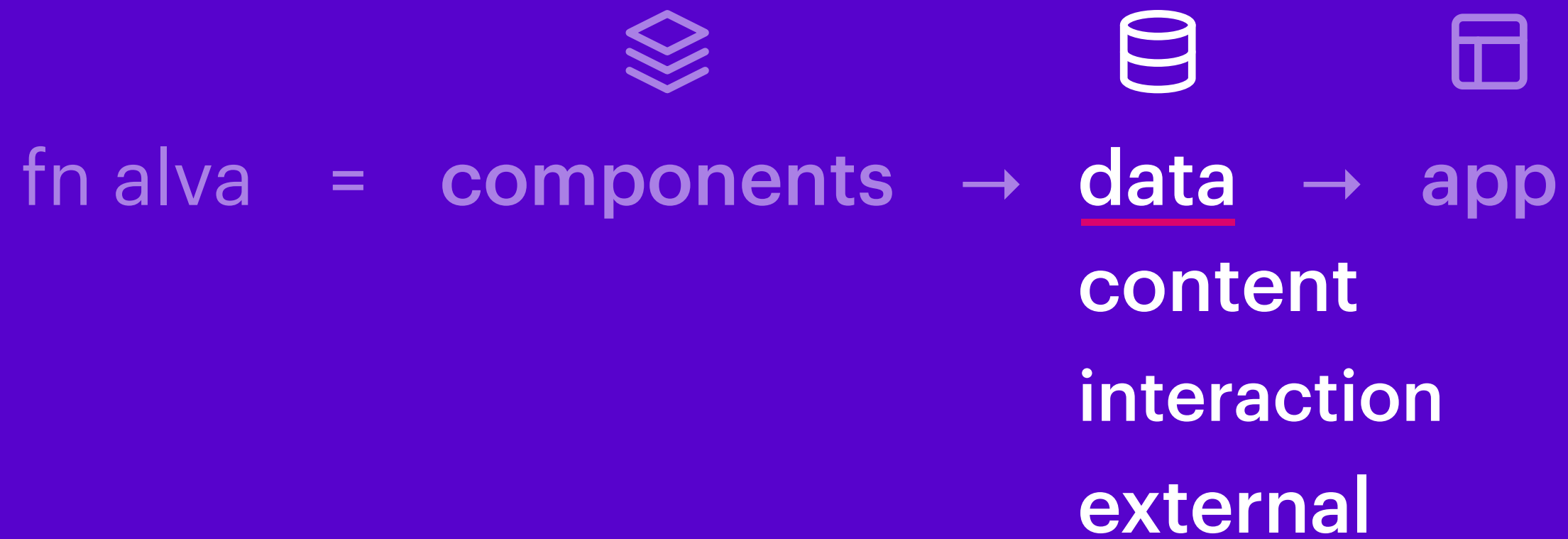
React.ComponentClass

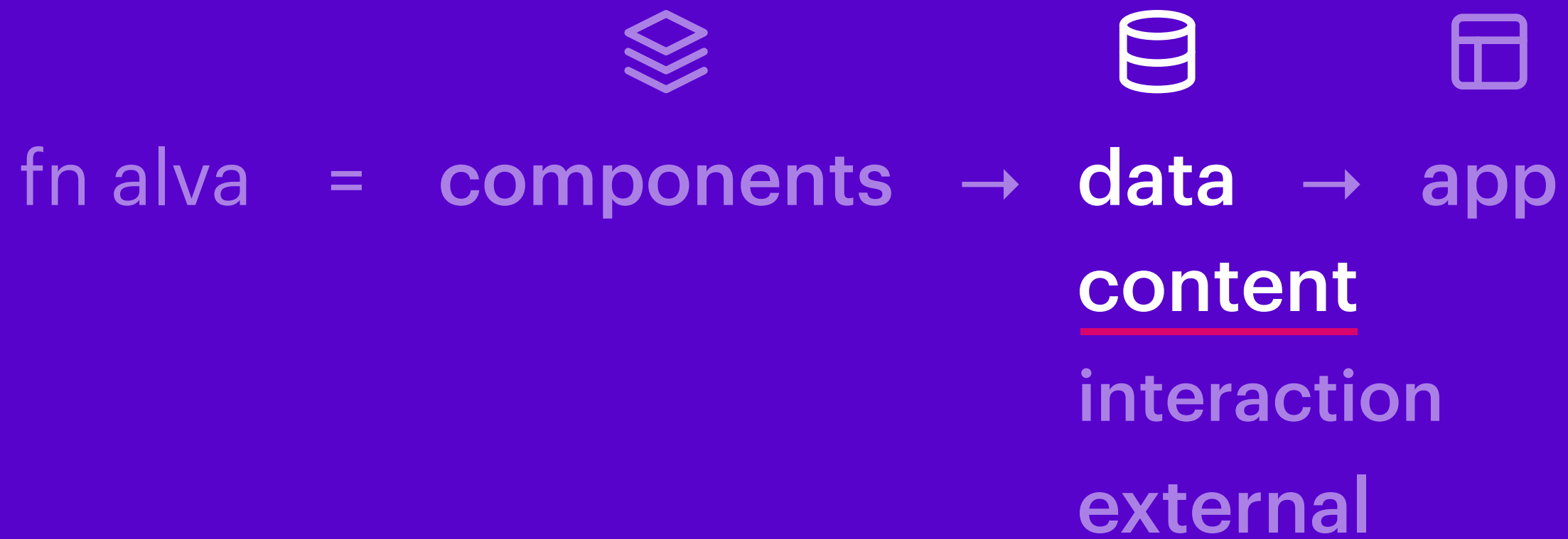
React.ComponentType

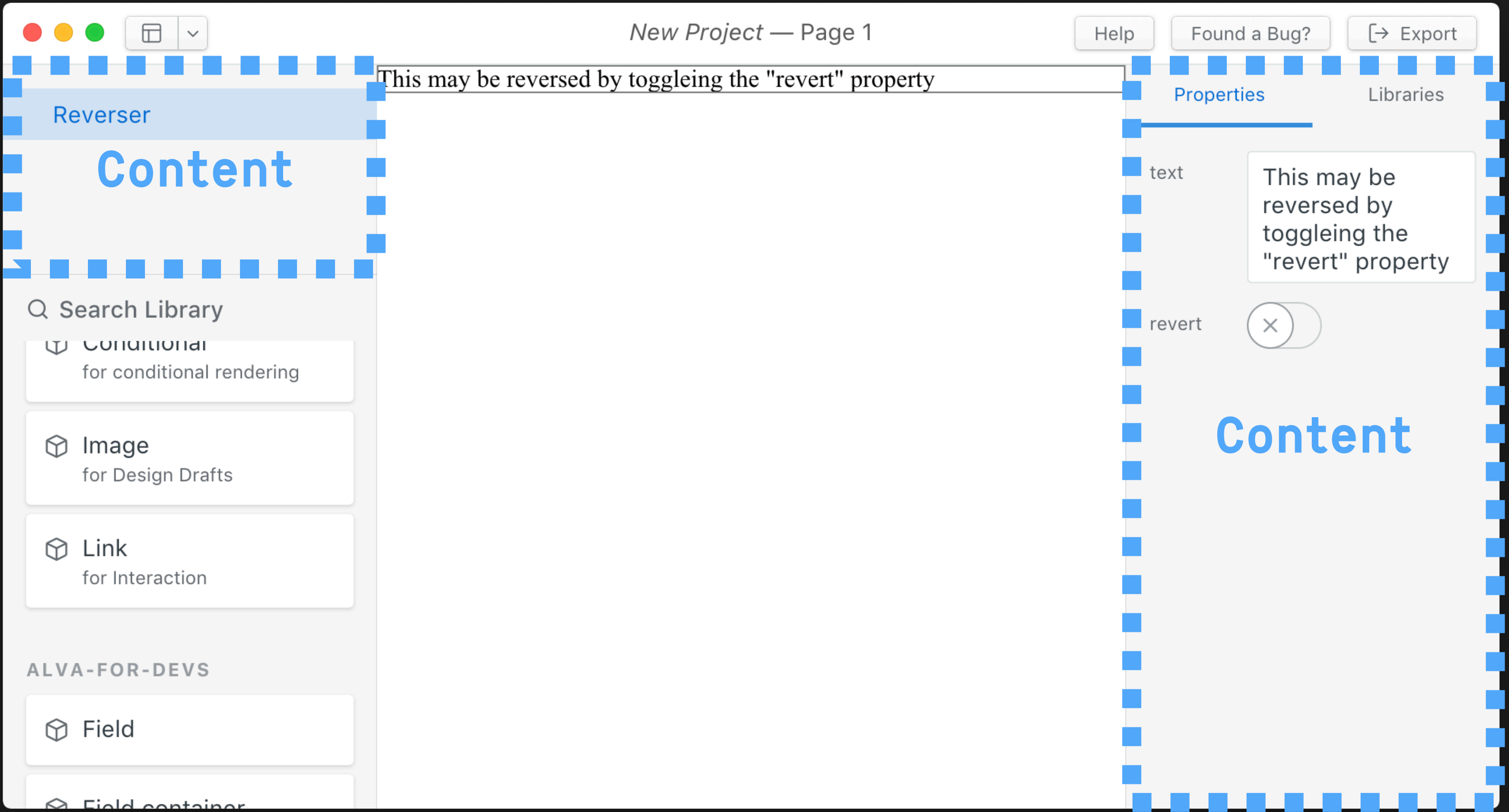
React.PureComponent

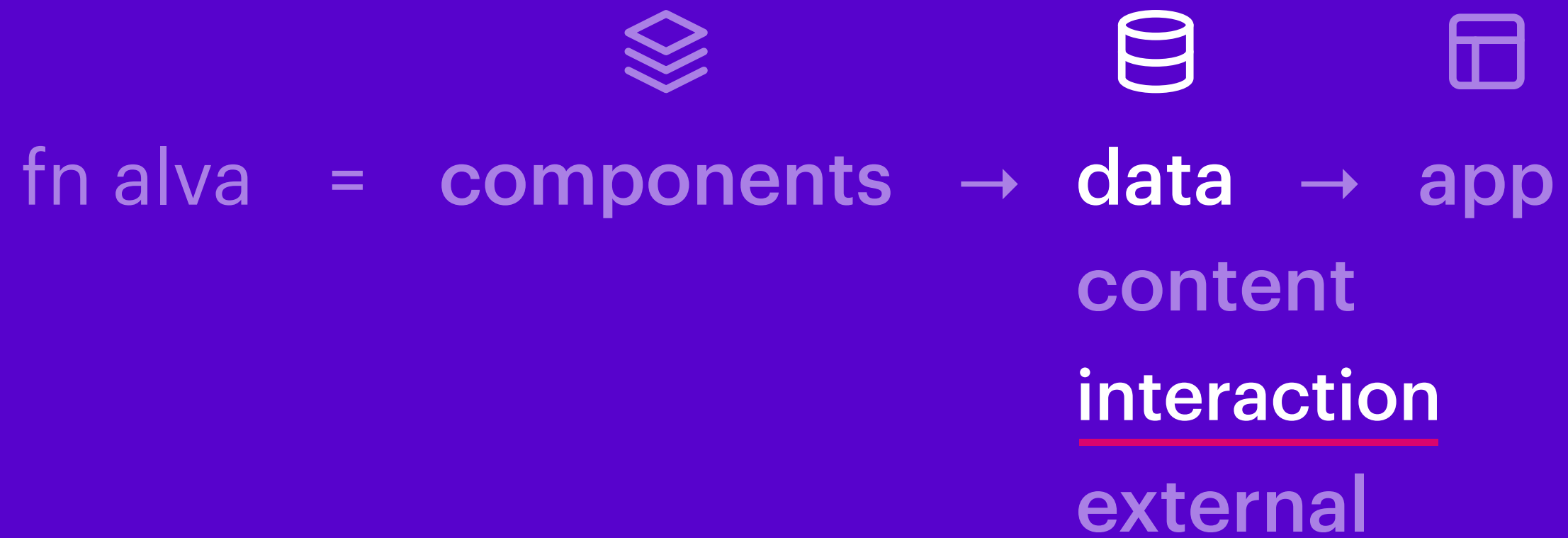
React.StatelessComponent

fn alva =  components → data  → app 





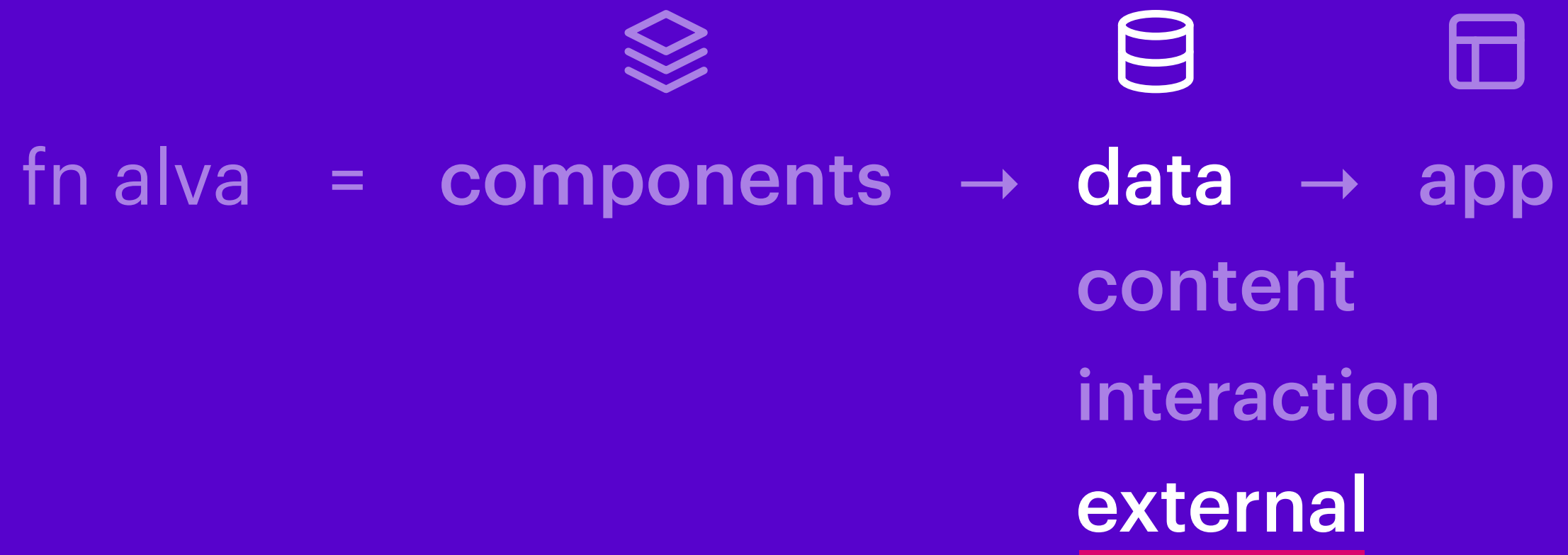






Demo

Work with User Data





Demo

Pull External Data

fn alva =  components →  data →  app



Demo

Prototype Export

Electron 2

as a wrapper for all major platforms

Web Technology

React App, Webpack, MobX, Styled Components

Q&A

Let's talk.

- ① Download it.
- ② Try it out.
- ③ Give us feedback.

meetalva.io

#s2-alva-feedback

Presentation + Alva Examples

github.com/meetalva/alva-for-devs



meetalva.io

github.com/meetalva

@meetalva