

# COMPOSING COMPONENTS

## IN ANGULAR

**Renee Vrantsidis**  
**@renvrant**



HELLO!



WE KINDA TAKE  
COMPONENTS FOR GRANTED



# COMPONENTS ARE COMPLEX

- API Design
- UI Design
- Architectural Design



# COMPONENTS ARE HIGH IMPACT

- Maintainability
- Scalability
- Performance

# ! COMPONENTS ARE IMPORTANT

- First point of entry for developers
- Greatest need for flexibility
- First impression for your user

NEGLECTING COMPONENTS  
MEANS PROJECTS SUFFER



HOW CAN WE MAKE THIS BETTER?





DESIGN



BUILD



INTEGRATE



# DESIGN THE COMPONENT



DESIGN IS ABOUT CHOICES  
AND TRADE-OFFS



# COMPONENT DESIGN QUESTIONS

- Inputs and Outputs?
- Content Projection?
- Flexibility?
- Variations?
- Responsibilities?
- NOT responsibilities?
- Is this a new component?



## WHAT MAKES A GOOD COMPONENT?

- Encapsulates a single concern
- Usable in many contexts
- Easy use with minimal internal knowledge

# COMPONENT WARNING SIGNS

- There are too many responsibilities
- You have to copy and paste
- Using the component is complicated

# EXAMPLE: STAR RATING COMPONENT



- Rounds to nearest star
- Users can click a star to add their rating





```
<star-rating>
  <star-button></star-button>
  <star-button></star-button>
  <star-button></star-button>
  <star-button></star-button>
  <star-button></star-button>
</star-rating>
```





HOW CAN WE MAKE THIS BETTER?



WORK BACKWARDS FROM HOW  
YOU'LL USE THE COMPONENT

```
<star-rating [rating]="3.5"  
            [max]="5"  
            (onRate)="handleRating($event)">  
</star-rating>
```

```
<star-rating [rating]="3.5"  
             [max]="5"  
             (onRate)="handleRating($event)">  
</star-rating>
```

```
<star-rating [rating]="3.5"  
             [max]="5"  
             (onRate)="handleRating($event)">  
</star-rating>
```

```
<star-rating [rating]="3.5"  
            [max]="5"  
            (onRate)="handleRating($event)">  
</star-rating>
```



```
<star-button *ngFor="let starNumber of max"  
    [isFull]="starIsFull(starNumber)"  
    (click)="onRate.emit(starNumber)">  
</star-button>
```

```
<star-button *ngFor="let starNumber of max"  
  [isFull]="starIsFull(starNumber)"  
  (click)="onRate.emit(starNumber)">  
</star-button>
```

```
<star-button *ngFor="let starNumber of max"  
  [isFull]="starIsFull(starNumber)"  
  (click)="onRate.emit(starNumber)">  
</star-button>
```

```
<star-button *ngFor="let starNumber of max"  
    [isFull]="starIsFull(starNumber)"  
    (click)="onRate.emit(starNumber)">  
</star-button>
```

## ✨ BENEFITS

- Less code for the user
- More encapsulated
- Responsibility has been delegated



# BUILD MARKUP & STYLES



## GOOD MARKUP MATTERS

- Smaller page weight
- Accessible components
- Easier to read and refactor

# GOOD MARKUP HAS ARMS

- Accessible
- Readable
- Minimal
- Semantic



```
<div class="border">
  <div (click)="handleClick()" class="margin">
    <div class="bold">
      Click me
    </div>
  </div>
</div>
```

```
<div class="border">  
  <div (click)="handleClick()" class="margin">  
    <div class="bold">  
      Click me  
    </div>  
  </div>  
</div>
```

```
<div class="border">  
  <div (click)="handleClick()" class="margin">  
    <div class="bold">  
      Click me  
    </div>  
  </div>  
</div>
```

```
<div class="border">  
  <button (click)="handleClick()" class="padding">  
    <strong>Click Me</strong>  
  </button>  
</div>
```

```
<div class="border">  
  <button (click)="handleClick()" class="padding">  
    <strong>Click Me</strong>  
  </button>  
</div>
```

```
<div class="border">  
  <button (click)="handleClick()" class="padding">  
    <strong>Click Me</strong>  
  </button>  
</div>
```

```
<button (click)="handleClick()" class="padding">  
  <strong>Click Me</strong>  
</button>
```

THE COMPONENT HOST IS  
RENDERED IN THE DOM

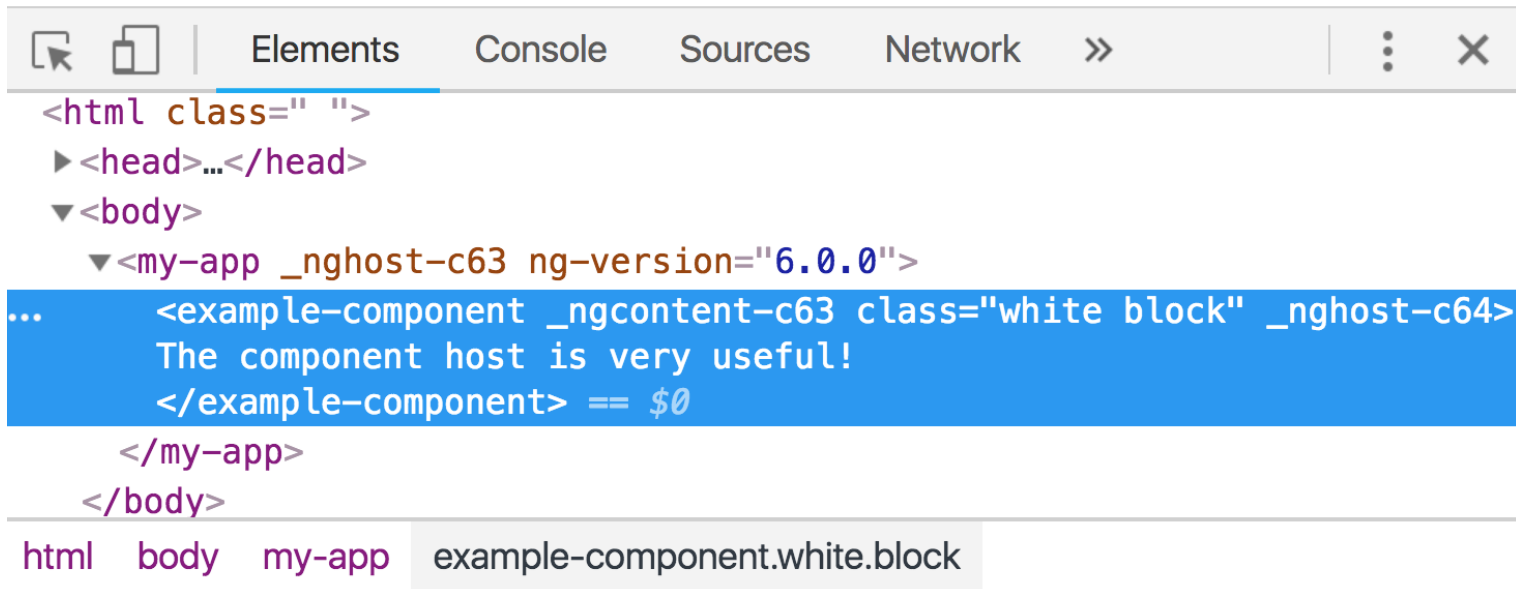


```
@Component({
  selector: 'example-component',
  template: `<ng-content></ng-content>`,
  styles: [`:host {
    background: slateblue;
    padding: 1rem; }`
  ],
  host: { class: 'white block' },
})
```

```
@Component({  
  selector: 'example-component',  
  template: `<ng-content></ng-content>`,  
  styles: [`:host {  
    background: slateblue;  
    padding: 1rem; }`  
  ],  
  host: { class: 'white block' },  
})
```

```
@Component({  
  selector: 'example-component',  
  template: `<ng-content></ng-content>`,  
  styles: [`:host {  
    background: slateblue;  
    padding: 1rem; }`  
  ],  
  host: { class: 'white block' },  
})
```

The component host is very useful!



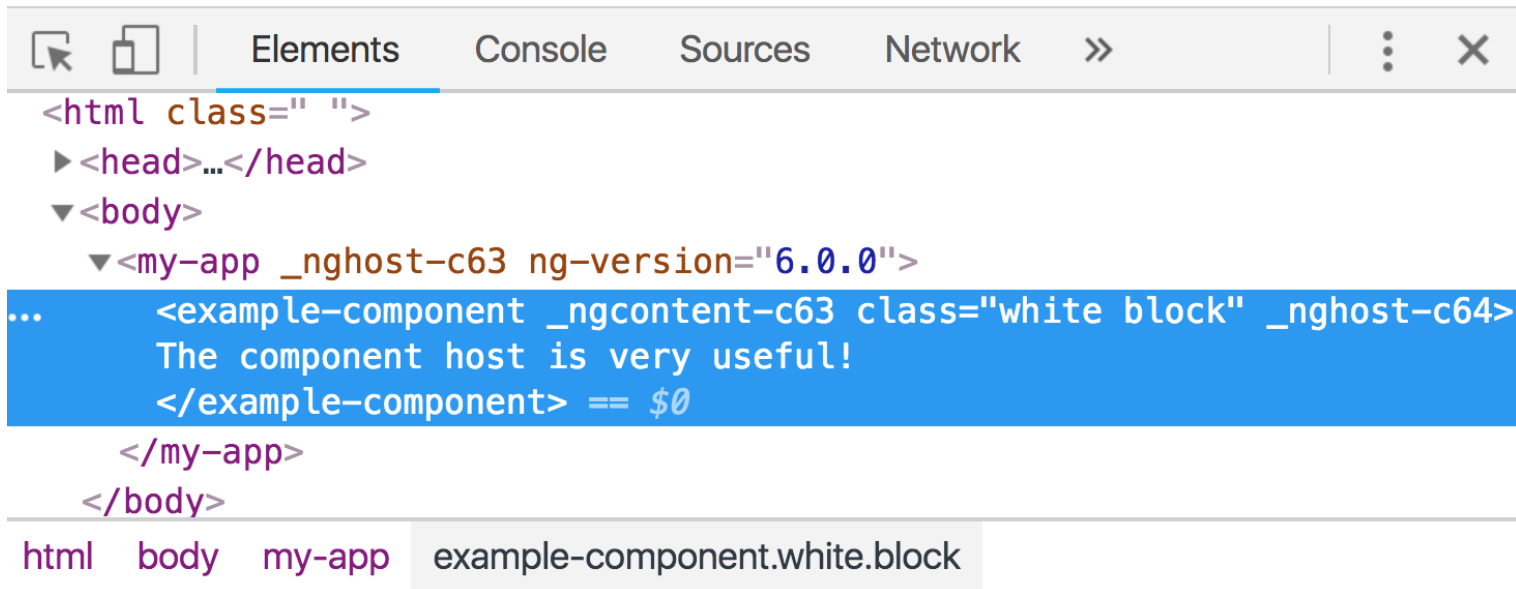
```
<html class="">
  <head>...</head>
  <body>
    <my-app _ngghost-c63 ng-version="6.0.0">
      ... <example-component _ngcontent-c63 class="white block" _ngghost-c64>
        The component host is very useful!
      </example-component> == $0
    </my-app>
  </body>
```

html body my-app example-component.white.block

```
@Component({  
  selector: 'example-component',  
  template: `<ng-content></ng-content>`,  
  styles: [`:host {  
    background: slateblue;  
    padding: 1rem; }`  
  ],  
  host: { class: 'white block' },  
})
```

```
@Component({
  selector: 'example-component',
  template: `<ng-content></ng-content>`,
  styles: [`:host {
    background: slateblue;
    padding: 1rem; }`
  ],
  host: { class: 'white block' },
})
```

The component host is very useful!



```
<html class="">
  <head>...</head>
  <body>
    <my-app _ngghost-c63 ng-version="6.0.0">
      ... <example-component _ngcontent-c63 class="white block" _ngghost-c64>
        The component host is very useful!
      </example-component> == $0
    </my-app>
  </body>
```

html body my-app example-component.white.block

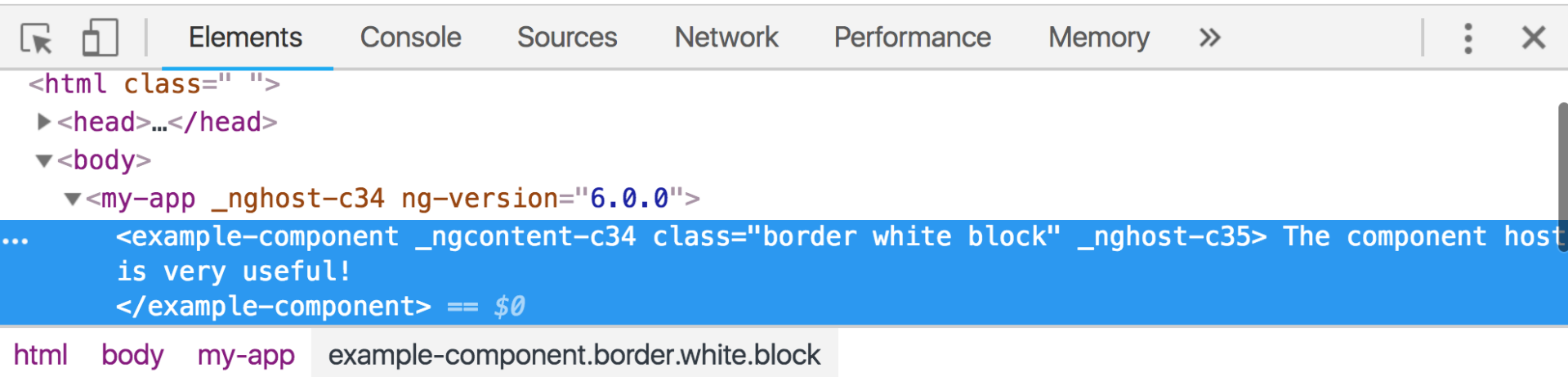
```
<example-component class="border">
```

```
  The component host is very useful!
```

```
</example-component>
```



The component host is very useful!



The screenshot shows a web browser's developer tools interface. The 'Elements' tab is selected, displaying a DOM tree. The tree structure is as follows:

- `<html class="" ">`
  - `<head>...</head>`
  - `<body>`
    - `<my-app _ngghost-c34 ng-version="6.0.0">`
      - `<example-component _ngcontent-c34 class="border white block" _ngghost-c35> The component host is very useful!`
      - `</example-component> == $0`

The rendered text 'The component host is very useful!' is highlighted in blue. The breadcrumb at the bottom of the developer tools shows the path: `html > body > my-app > example-component.border.white.block`.

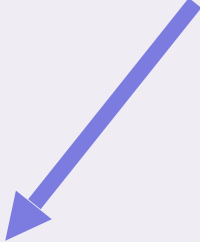
HOST STYLING MAKES FOR  
FLEXIBLE COMPONENTS







## No Direct Access



```
<star-rating>  
  <star-button></star-button>  
</star-rating>
```

```
<star-rating [rating]="3.5"  
             [max]="5"  
             [color]="red"  
             [size]="large"  
             (onRate)="handleRating($event)">  
</star-rating>
```



HOW CAN WE MAKE THIS BETTER?



```
// star-button
:host-context(.large-stars) {
  font-size: 3rem;
  color: red;
}
```

```
// star-button
:host-context(.large-stars) {
  font-size: 3rem;
  color: red;
}
```

```
// star-button
:host-context(.large-stars) {
  font-size: 3rem;
  color: red;
}
```



```

Elements Console Sources Network Performance Memory Ap
<!doctype html>
<html class=" ">
  <head>...</head>
  <body>
    <my-app _ngghost-c0 ng-version="6.0.0">
...  <star-rating _ngcontent-c0 class="large-stars" _ngghost-c1> == $0
      <star-button _ngcontent-c1 class _ngghost-c2>★ ★ ★ ★ ☆</star-button>
    </star-rating>

```

```
<red-star-context>  
  <star-rating ...></star-rating>  
</red-star-context>
```

```
// star-button
:host-context(red-star-context) {
  font-size: 3rem;
  color: red;
}
```

## ✨ BENEFITS

- Style variants stay in the component
- Avoids cluttering the API
- Preserves CSS scope



# INTEGRATE APP CONTEXT





COMPONENTS ARE PART OF  
A LARGER SYSTEM



## COMPONENTS INTERACT WITH

- Application state
- Application context
- Other components



## COMPONENTS INTERACT WITH

- Application state
- Application context
- Other components



# A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.



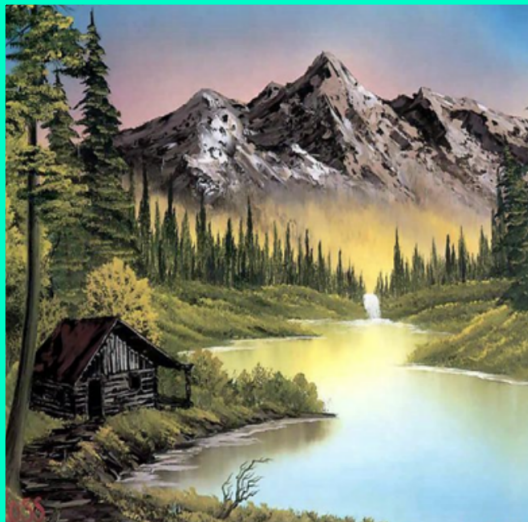
INPUT IS STRICT,  
PROJECTION IS FLEXIBLE

## // Using Location Card

```
<location-card imgSrc="path/to/image">  
  <ng-container Title>A Mountain Cabin</ng-container>  
  <ng-container Description>Lorem Ipsum</ng-container>  
  <star-rating ...></star-rating>  
</location-card>
```

## // Location Card

```
<card-title>
  <ng-content select="[Title]"></ng-content>
</card-title>
<card-copy>
  <ng-content select="[Description]"></ng-content>
</card-copy>
<img [attr.src]="imgSrc" alt="">
<ng-content select="star-rating"></ng-content>
```



# A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.





## // Using Location Card

```
<location-card imgSrc="path/to/image">  
  <ng-container Title>A Mountain Cabin</ng-container>  
  <ng-container Description>Lorem Ipsum</ng-container>  
  <star-rating ...></star-rating>  
</location-card>
```

## // Location Card

```
<card-title>
  <ng-content select="[Title]"></ng-content>
</card-title>
<card-copy>
  <ng-content select="[Description]"></ng-content>
</card-copy>
<img [attr.src]="imgSrc" alt="">
<ng-content select="star-rating"></ng-content>
```



# A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.



## // Using Location Card

```
<location-card imgSrc="path/to/image">  
  <ng-container Title>A Mountain Cabin</ng-container>  
  <ng-container Description>Lorem Ipsum</ng-container>  
  <star-rating ...></star-rating>  
</location-card>
```

## // Location Card

```
<card-title>
```

```
  <ng-content select="[Title]"></ng-content>
```

```
</card-title>
```

```
<card-copy>
```

```
  <ng-content select="[Description]"></ng-content>
```

```
</card-copy>
```

```
<img [attr.src]="imgSrc" alt="">
```

```
<ng-content select="star-rating"></ng-content>
```



# A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.



## // Location Card

```
<card-title>
  <ng-content select="[Title]"></ng-content>
</card-title>
<card-copy>
  <ng-content select="[Description]"></ng-content>
</card-copy>
<img [attr.src]="imgSrc" alt="">
<ng-content select="star-rating"></ng-content>
```

## // Using Location Card

```
<location-card imgSrc="path/to/image">  
  <ng-container Title>A Mountain Cabin</ng-container>  
  <ng-container Description>Lorem Ipsum</ng-container>  
  <star-rating ...></star-rating>  
</location-card>
```



## // Location Card

```
<card-title>
  <ng-content select="[Title]"></ng-content>
</card-title>
<card-copy>
  <ng-content select="[Description]"></ng-content>
</card-copy>
<img [attr.src]="imgSrc" alt="">
<ng-content select="star-rating"></ng-content>
```

## ✨ BENEFITS

- Enforce markup where appropriate
- Allows for formatting
- Works well with atomic design



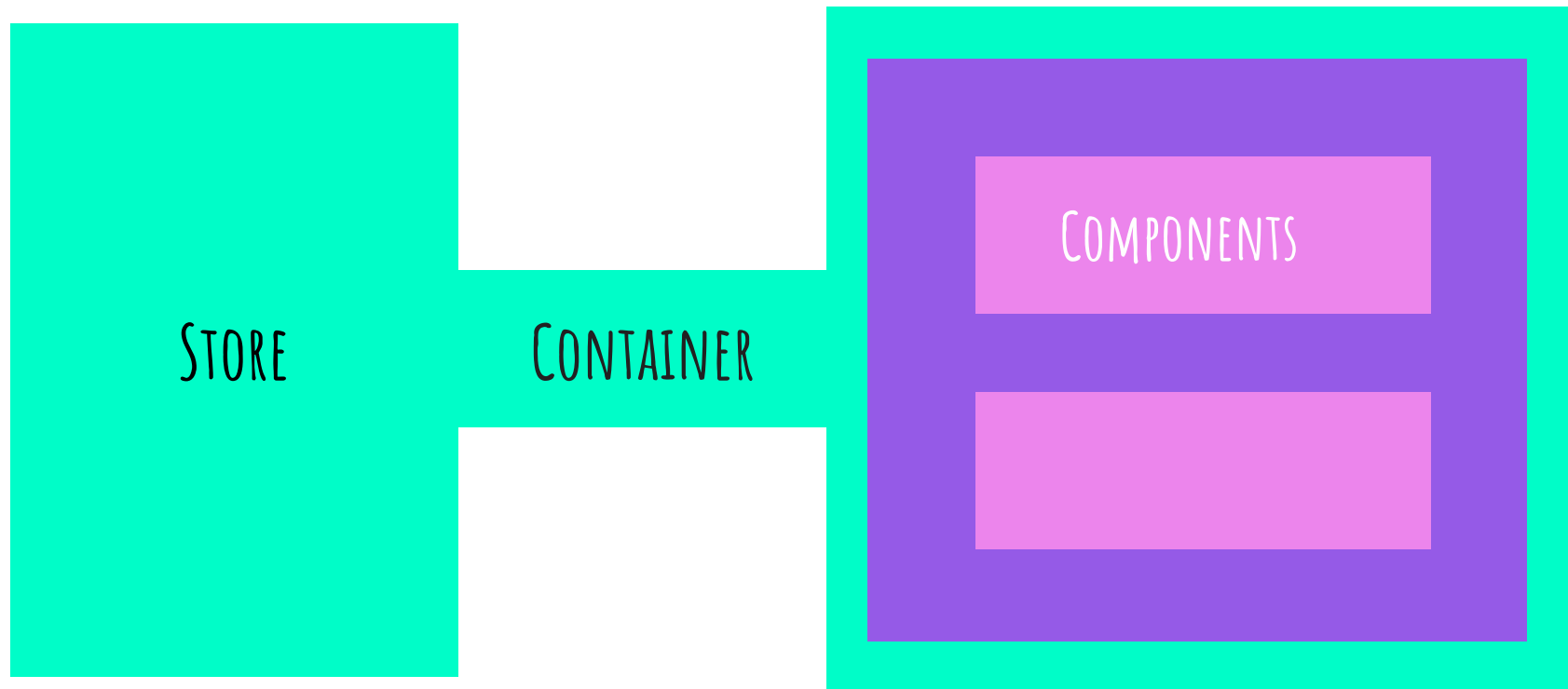
## COMPONENTS INTERACT WITH

- Application state
- Application context
- Other components



# CONTAINERS

- Insulate components from state
- Make components easier to reuse
- Let business logic be defined elsewhere





# A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.





## A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.



## A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.



## A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.



## A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.



## A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.



## A Mountain Cabin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean vel aliquam ex, vel ultricies velit. Curabitur ultricies eros ex, id blandit sapien gravida in. Mauris laoreet volutpat elit, nec interdum dui vestibulum quis. Nam sagittis elit et erat molestie, quis dignissim purus cursus.



<location-container>

<location-list>

<location-card>

<star-rating>

<star-button>



<location-container> (**Action Dispatch**)

<location-list>

<location-card>

<star-rating>

<star-button> (**Event**)





HOW CAN WE MAKE THIS BETTER?



COMPONENTS

<location-container>

<location-list>

<location-card>

<rating-container>

<star-rating>

<star-button>

<location-container>

<location-list>

<location-card>

<rating-container> (**Dispatch**)

<star-rating>

<star-button> (**Event**)



## ✨ BENEFITS

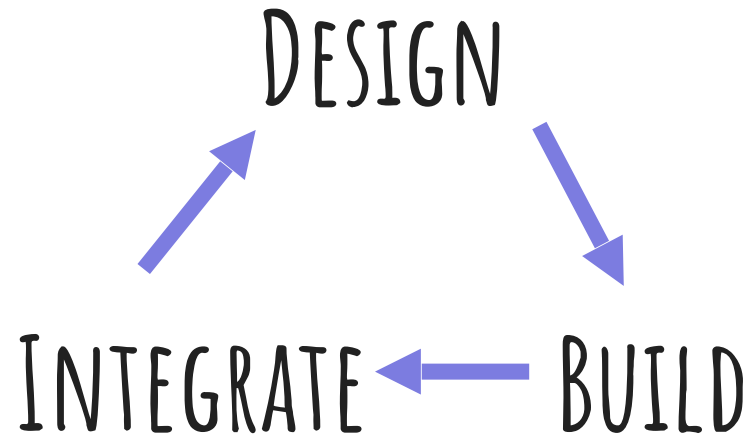
- Component tree stays loosely coupled
- No data “just passing through”
- Easier to maintain and debug



# FINAL THOUGHTS

DESIGN → BUILD → INTEGRATE





DESIGN  
BUILD  
INTEGRATE



KEEP REFACTORING



HOW CAN WE MAKE THIS BETTER?



THANK YOU!

@RENVRANT