Structuring Components

Dan Wahlin
WAHLIN CONSULTING

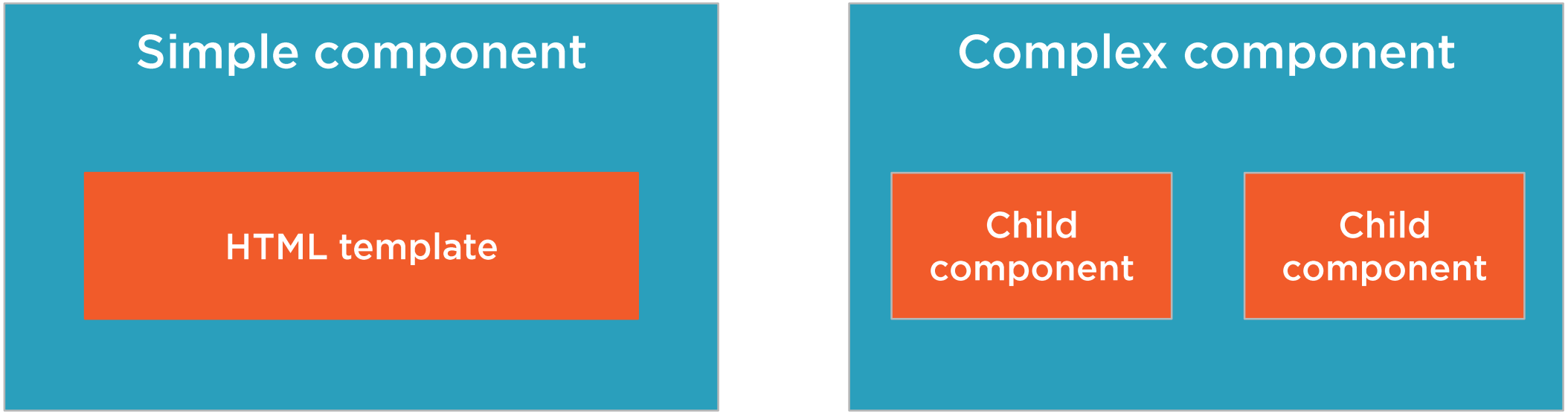
@danwahlin www.codewithdan.com

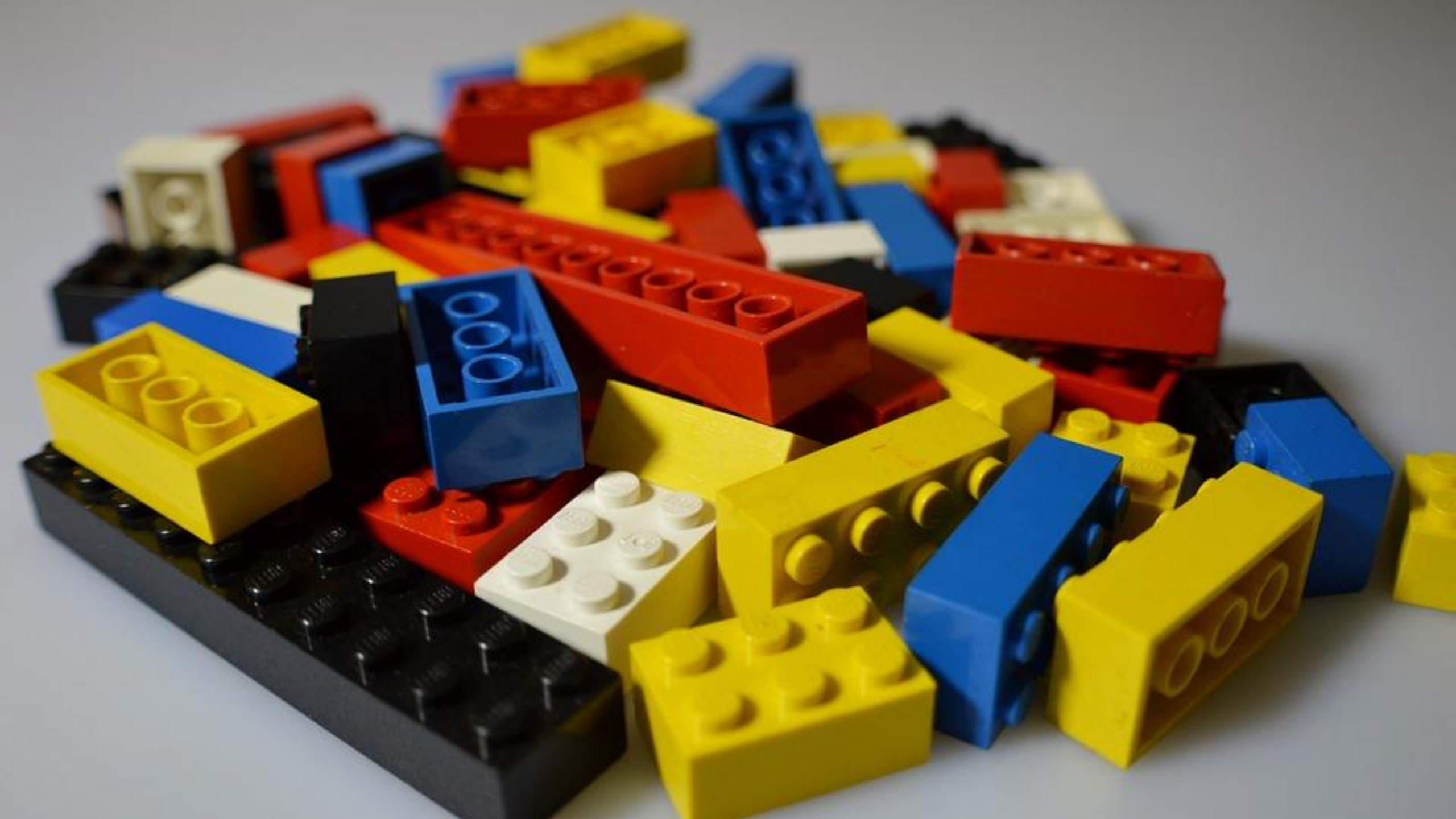
Module Overview

- Container and presentation components
- Passing state with input and output properties
- Change detection strategies
- ngOnChanges: reference vs. value
- Cloning techniques
- Component inheritance

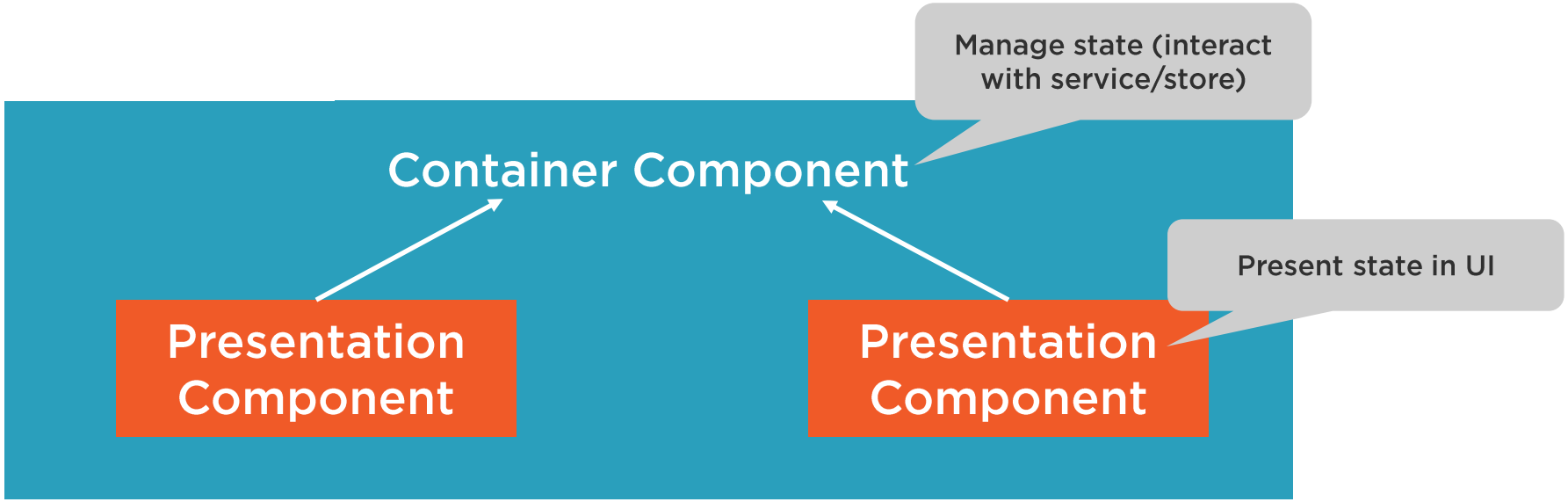
Container and Presentation Components

Structuring Components





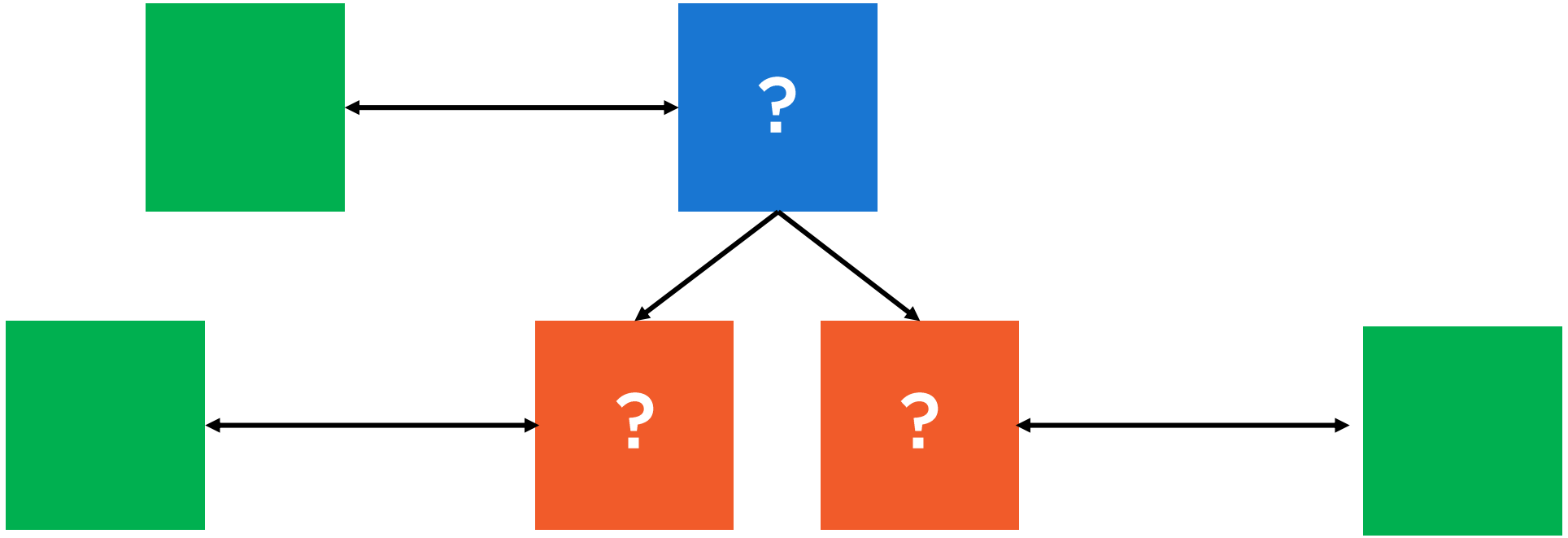
Container and Presentation Components



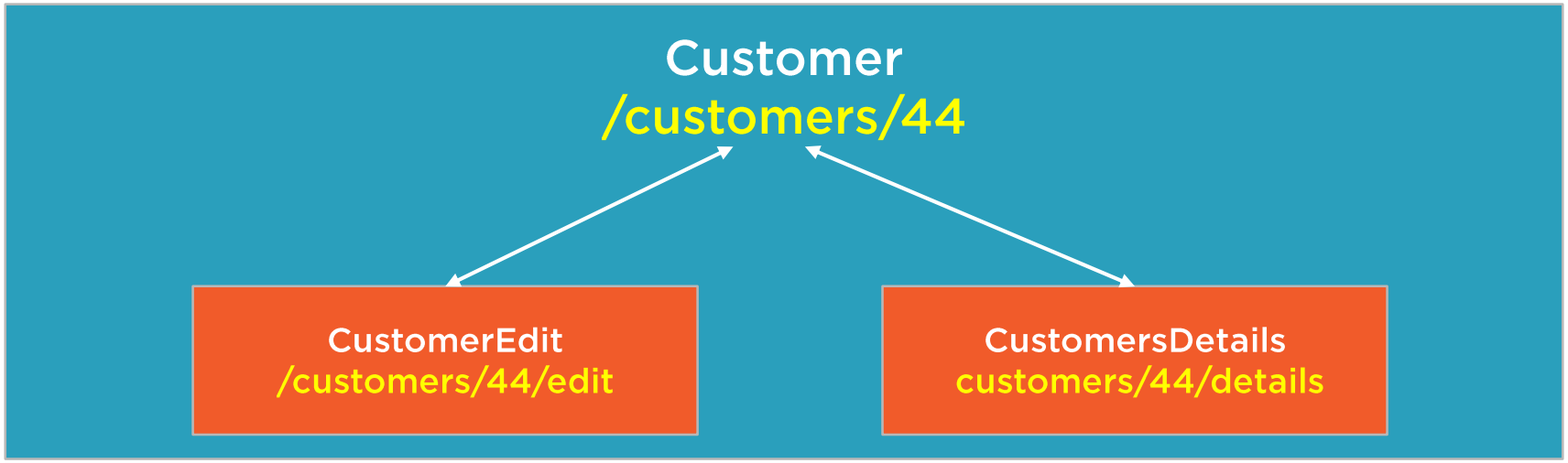
Container component, retrieves state

Presentation component, presents/renders state

Without a Container Component

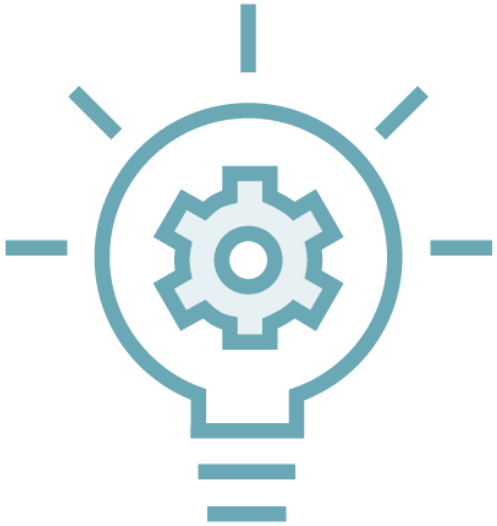


Child Routes and Components




```
const routes: Route = [
  {
    path: 'customers/:id',
    component: CustomerComponent,
    children: [
      { path: 'edit', component: CustomerEditComponent },
      { path: 'details", component: CustomerDetailsComponent }
    ]
  }
];
```

Working with Container/Presentation Components

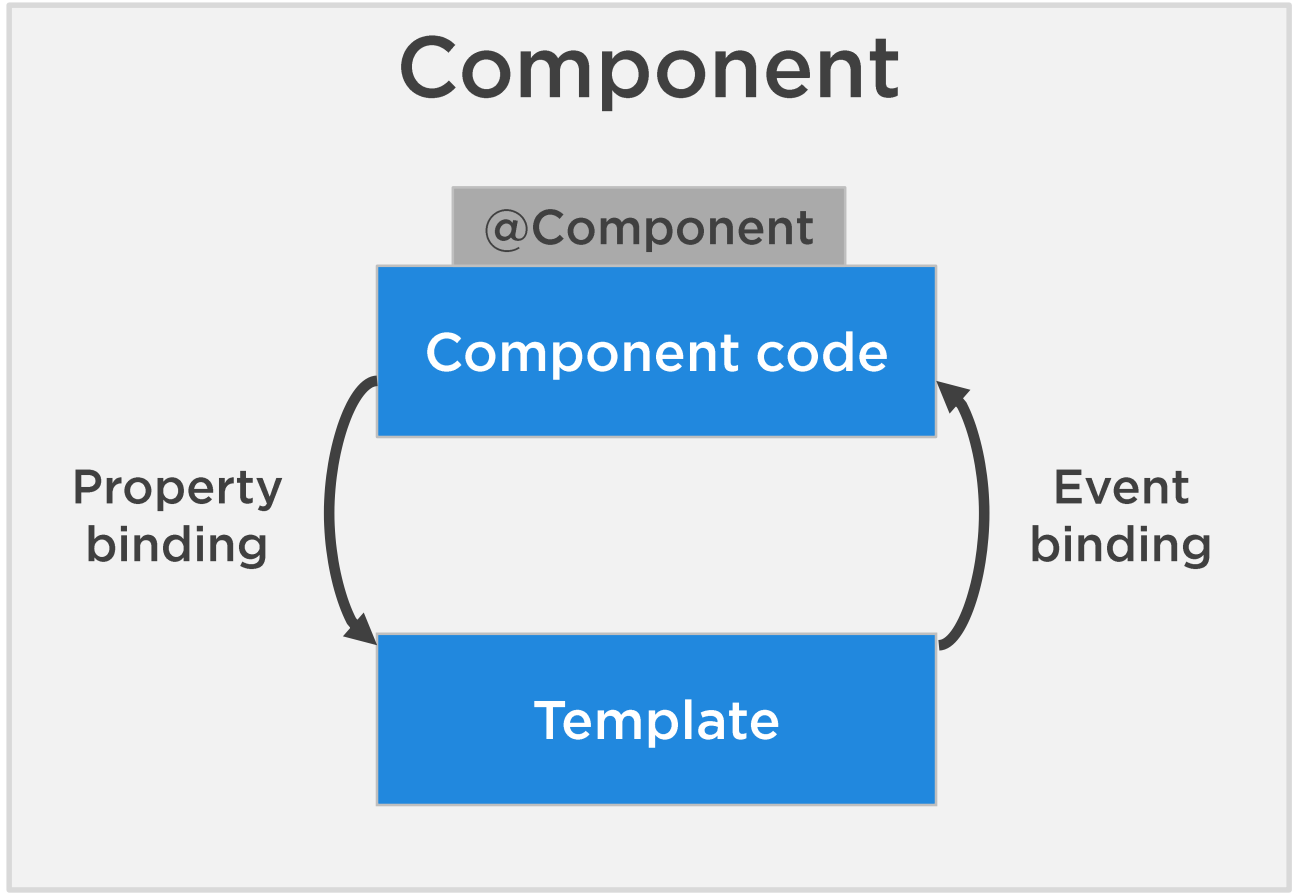


- Pass state from container → presentation using input properties
- Communicate from presentation → container using output properties
- Set change detection strategy to OnPush in presentation components
- Consider cloning complex types
- Use child routes as appropriate

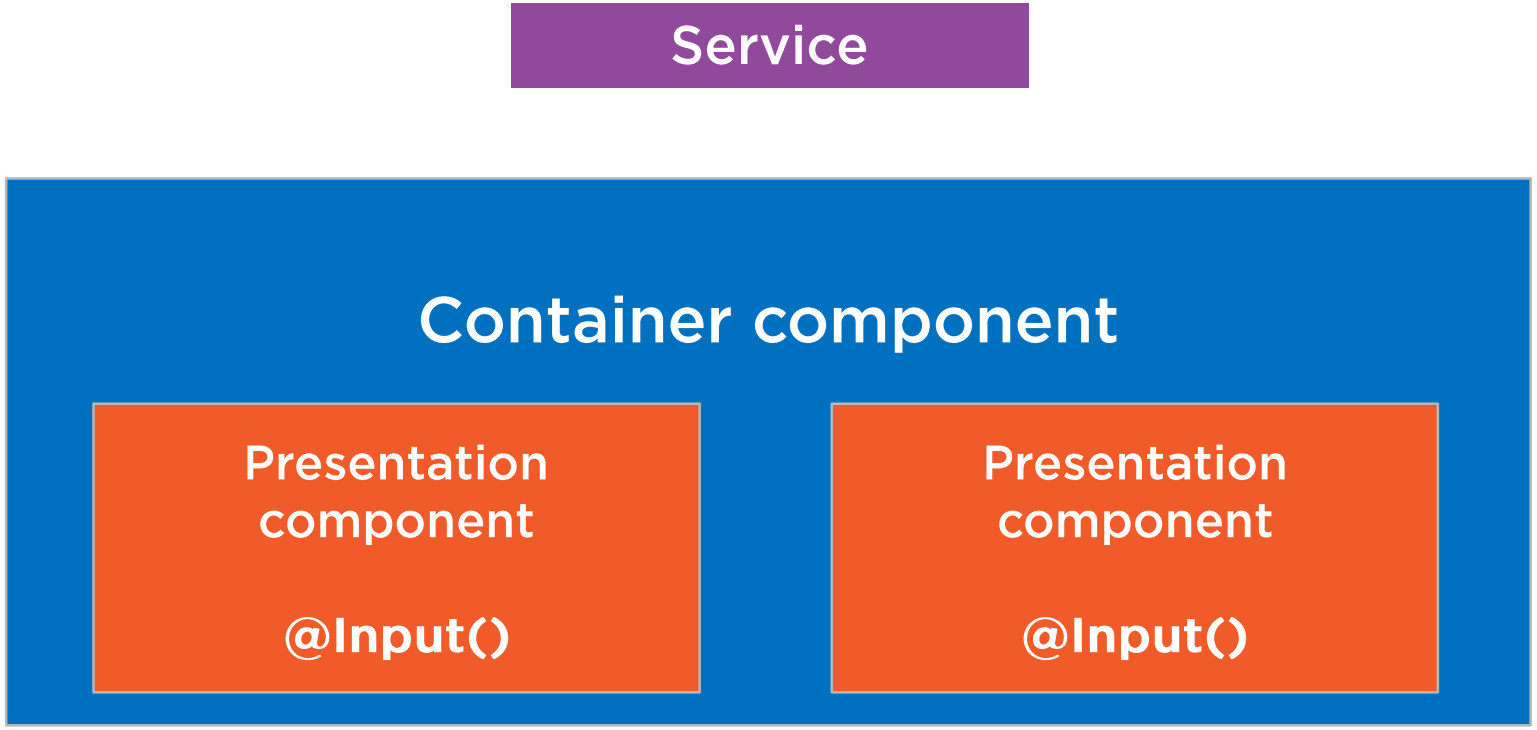
Container and Presentation Components in Action

Passing State with Input and Output Properties

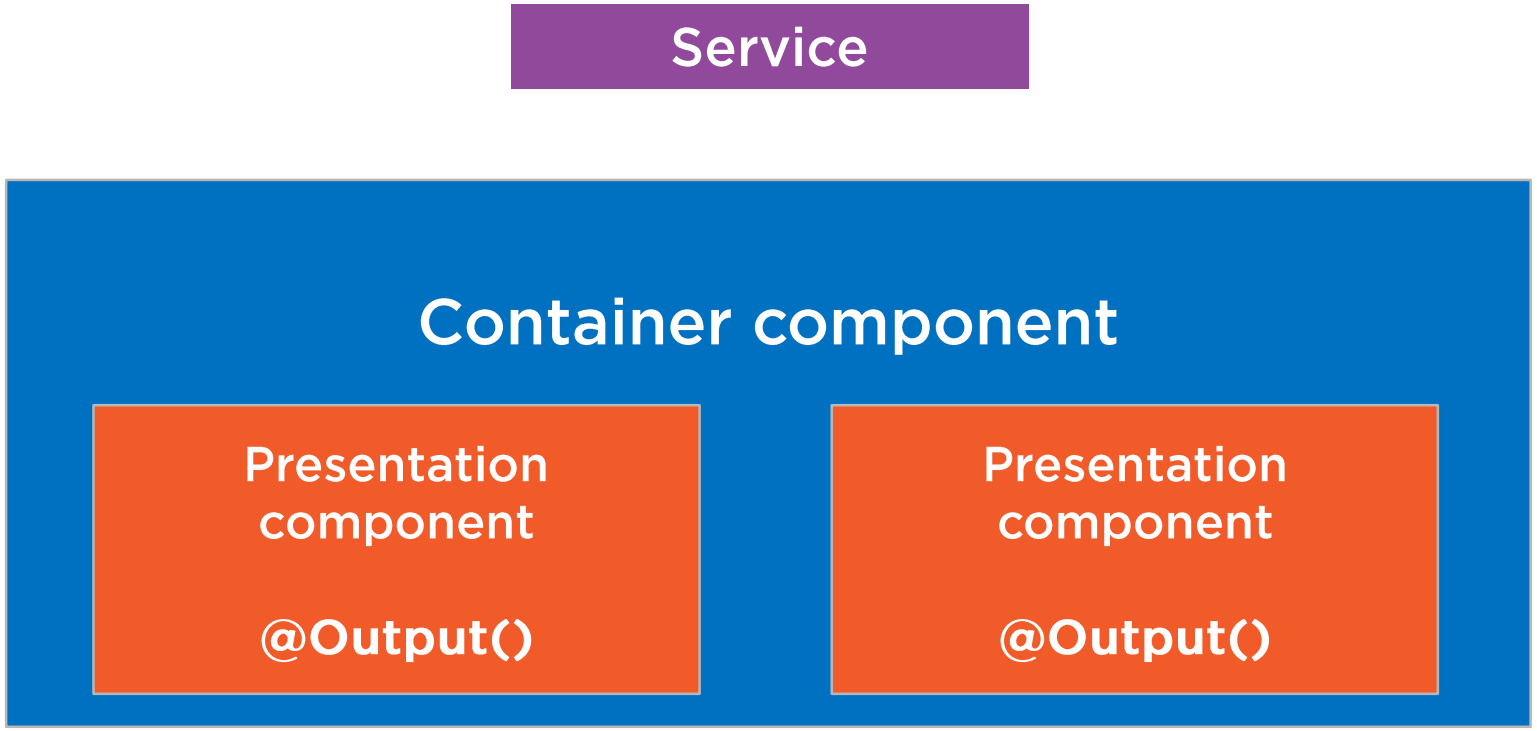
Component Templates



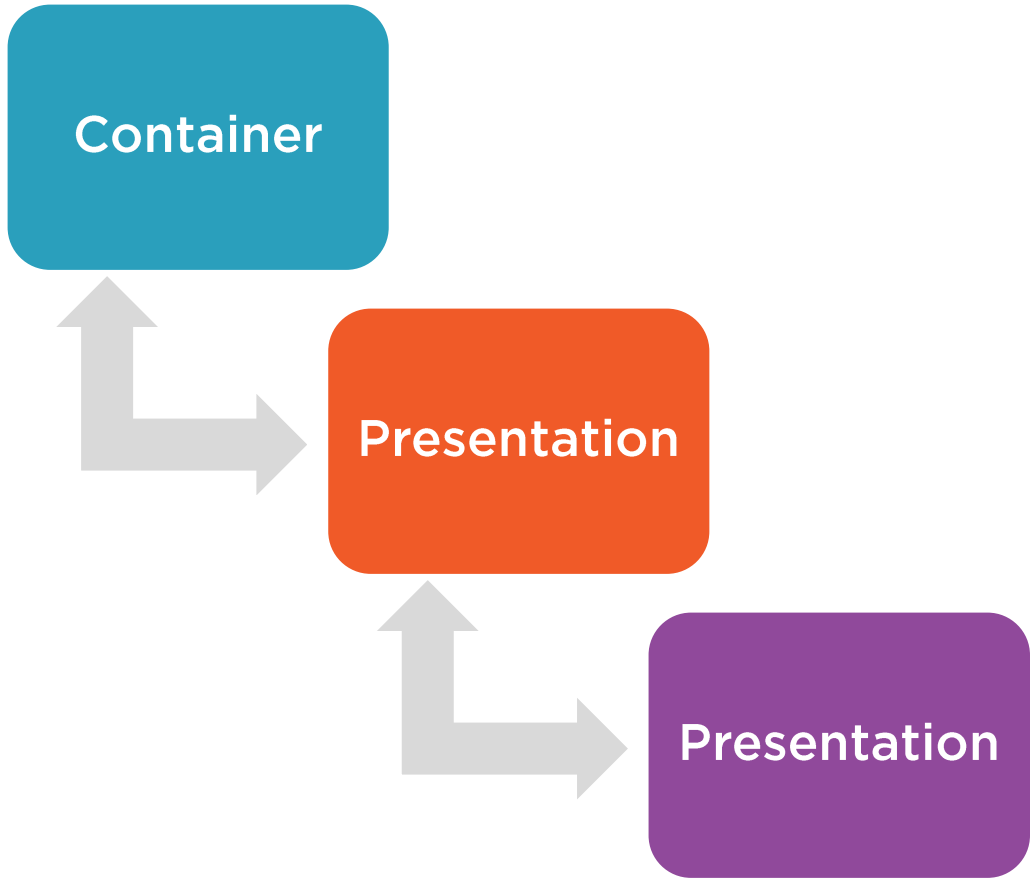
Input Properties



Output Properties



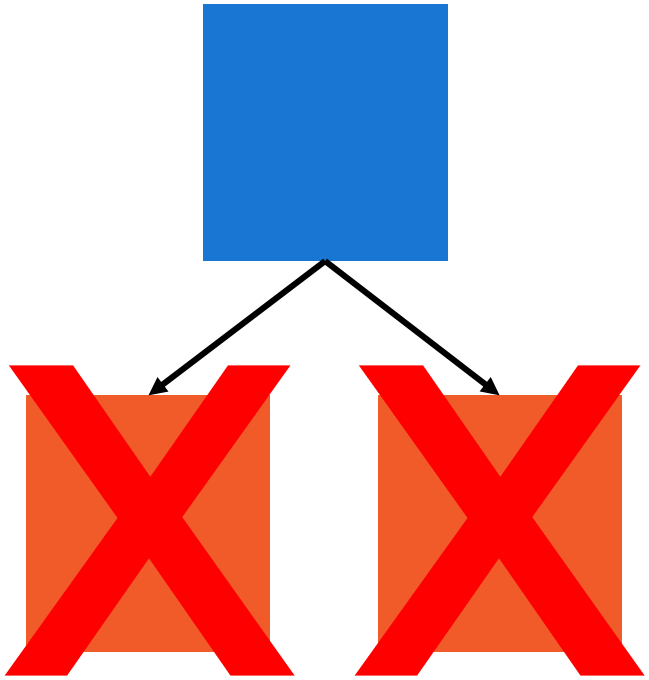
Input/Output Properties and Hierarchies



Rarely is there *one right way* that applies to all scenarios

Input and Output Properties in Action

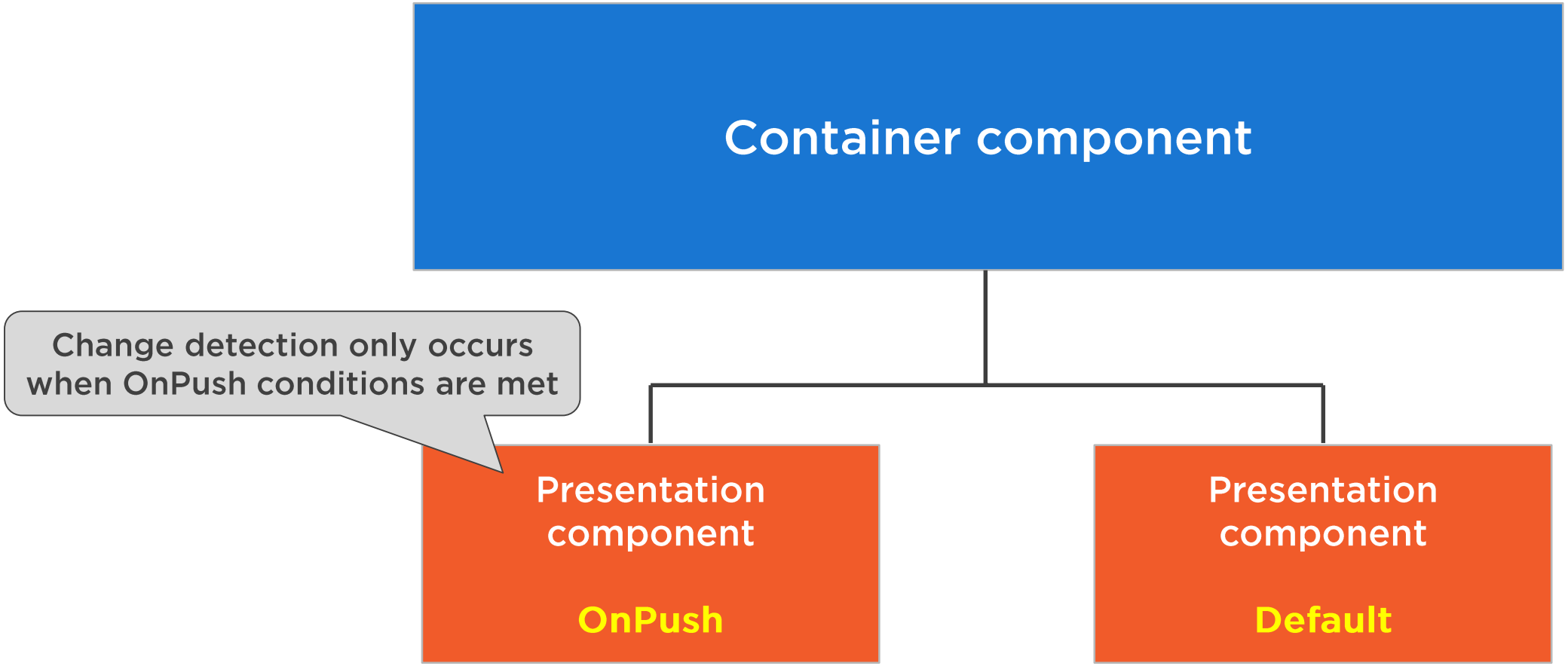
Change Detection Strategies



Child/presentation component should not change data

That’s the job of the parent/container component

OnPush and Change Detectors



Reference vs. Value Types

Cloning Techniques

Cloning is the process of making an exact copy of an object

A Few Cloning Techniques

JSON.parse()

Custom

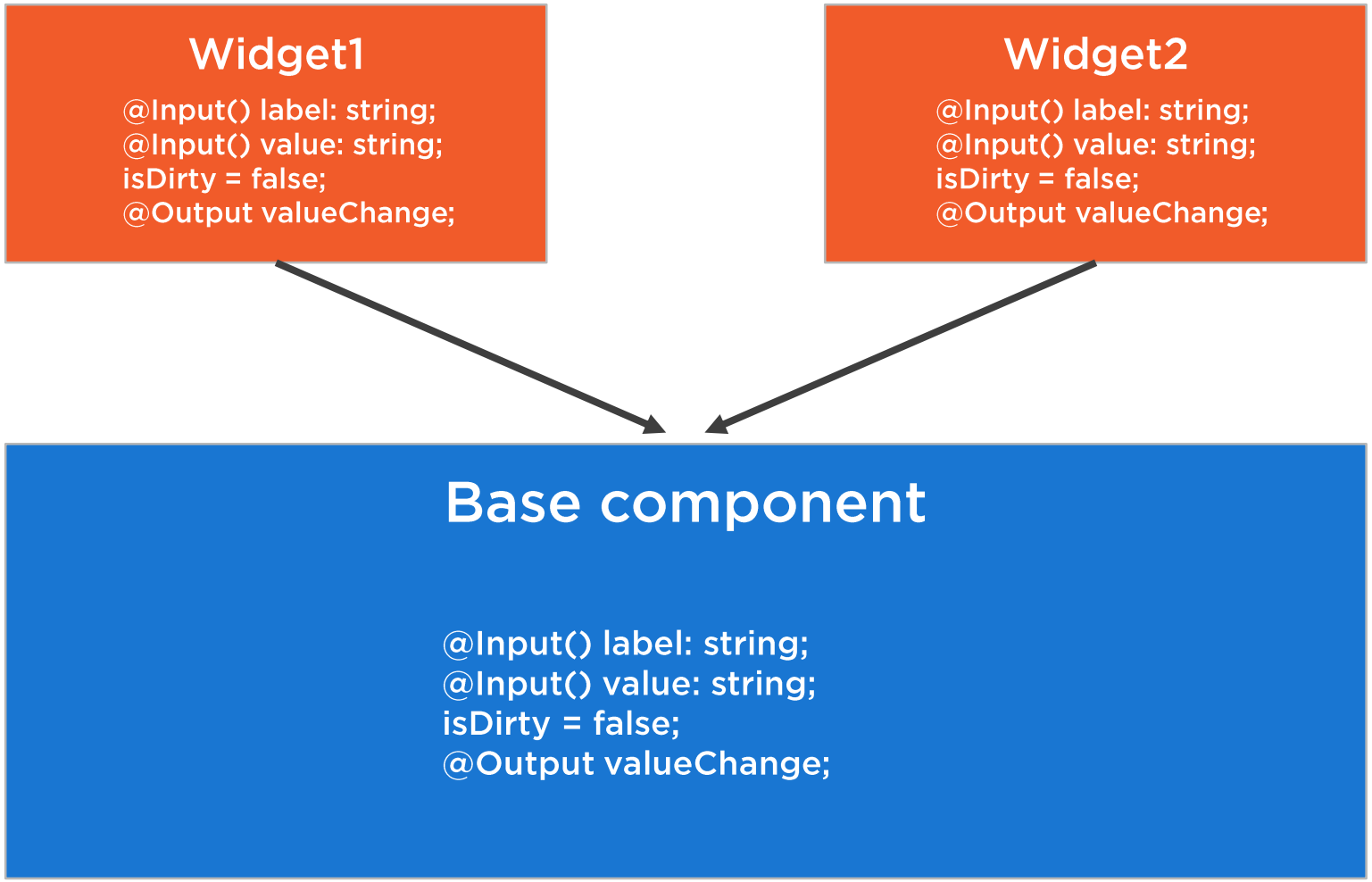
Immutable.js

Cloning in Action

Cloning with Immutable.js

Component Inheritance

Component Inheritance



Component Inheritance in Action

Summary

Break complex components into child components

Use the container → presentation pattern where possible

Container retrieves state and presentation components render

Use OnPush on presentation components

Leverage cloning/immutable data when appropriate

Use component inheritance sparingly (when it makes sense)