

REACT组件化通用模式



关于我

卢林 & 小撸

国双前端工程师

[GitHub](#) | [Blog](#) | [ZhiHu](#) | [Sina](#)

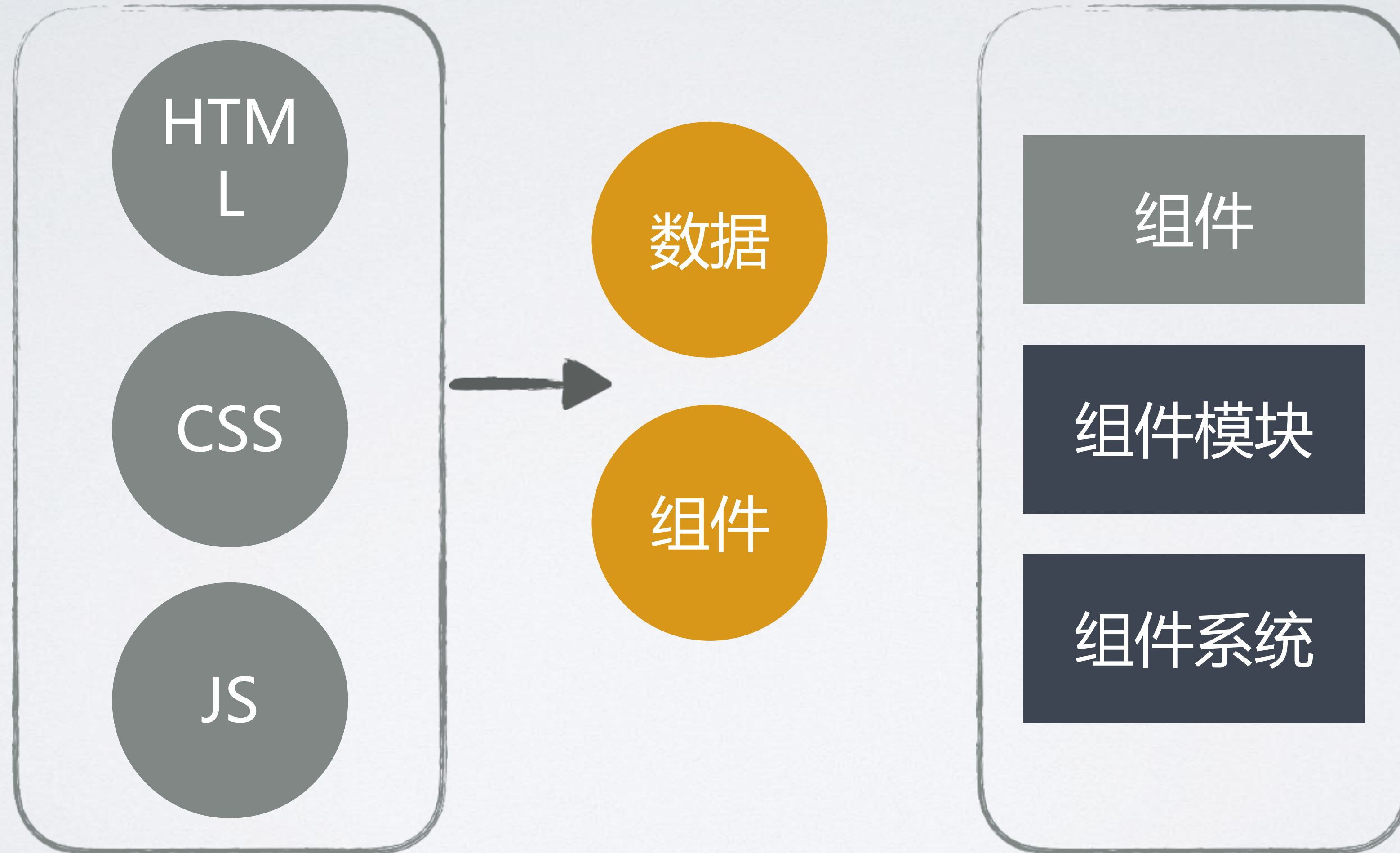
为什么要讲模式？

框架越来越多



需要找到规律或者有效应对的方法

关注点



为什么是数据？

```
{
  selected: [ 1 ]
  items: [{
    id: 1,
    name: 'A'
  }, {
    id: 2,
    name: 'B'
  }]
}
```

```
{
  items: [{
    id: 1,
    name: 'A',
    selected: true
  }, {
    id: 2,
    name: 'B',
    selected: false
  }]
}
```

什么是组件？

iOS的故事板

The screenshot displays the Xcode interface for editing an iOS storyboard. On the left, the 'Login Scene' hierarchy is shown, with 'Login Button' selected. The central canvas shows a login form with fields for '用户名' (Username) and '密码' (Password), a red error message '密码至少6位数' (Password must be at least 6 digits), and a blue '登录' (Login) button. The right-hand 'Properties Inspector' is open to the 'Button' section. A red arrow points to the 'State Conf' dropdown menu, which is expanded to show options: 'Default' (selected), 'Highlighted', 'Selected', and 'Disabled'. Other visible settings include 'Type: System', 'Font: 系统字体 15.0', 'Text Color', 'Shadow Color: Default', 'Image: Default Image', 'Background: Default Background Image', 'Shadow Offset' (Width: 0, Height: 0), and 'Line Break: Truncate Middle'. The 'Control' section at the bottom shows alignment options.

```
<scenes>
  <scene sceneID="tne-QT-ifu">
    <objects>
      <viewController title="Login" customClass="ViewController">
        <view key="view" contentMode="scaleToFill"></view>
        <connections>
          <outlet property="passwordTextField" />
          <outlet property="tipValidLabel" />
        </connections>
      </viewController>
    </objects>
  </scene>
</scenes>
```

结构描述

```
<button>
  <state key="normal" title="Login">
    <color key="titleColor" red="1" green="1" blue="1" alpha="1" />
  </state>
</button>
```

状态&样式描述


```
@IBOutlet private weak var passwordTextField: UITextField!  
@IBOutlet private weak var tipValidLabel: UILabel!  
  
let passwordValid: Observable<Bool> = passwordTextField.rx.text.orEmpty  
    .map { newPassword in newPassword.characters.count > 5 }  
  
passwordValid  
    .bind(to: tipValidLabel.rx.isHidden)  
    .disposed(by: disposeBag)  
  
passwordValid  
    .bind(to: loginButton.rx.isEnabled)  
    .disposed(by: disposeBag)
```

数据行为描述



跟 REACT 有撒关系？

结构描述



样式描述



数据描述



在 REACT 中如何体现？

```
export default class Button extends Component {
  render() {
    const {props} = this;
    const buttonClass = this.getClassName(['lv-button', 'primary']);

    return (
      <button onClick={props.onClick} className={buttonClass}>
        {props.children}
      </button>
    );
  }
}
```


在 REACT 中如何体现？

```
export default function Button(props) {  
  const buttonClass = getClassNames(['lv-button', 'primary']);  
  
  return (  
    <button onClick={props.onClick} className={buttonClass}>  
      {props.children}  
    </button>  
  );  
}
```

如何使用这些组件？

```
import {Dog} from 'dog.component'  
  
export class Cat extends Component {  
  render() {  
    return (  
      <div>  
        bully => <Dog />  
      </div>  
    );  
  }  
}
```


如何关联理解？

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.css']  
})  
export class AppComponent { }
```

Angular

```
@NgModule ({
  declarations: [
    AppComponent,
    TabComponent,
    GridComponent
  ],
  imports: [
    BrowserModule,
    HttpClientModule
  ],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Angular


```
export default class SimpleModal extends Component {
  render() {
    const {props} = this;
    if(!props.isOpen) {
      return null;
    }

    return (
      <div className={styles['lv-modal']} style={props.mainStyle}>
        <div className={styles['lv-modal__header']}>
          <h3>{props.title}</h3>
          <span onClick={this.handleClick}><i>X</i></span>
        </div>
        <div className={styles['lv-modal__main']}>{props.children}</div>
      </div>
    );
  }
}

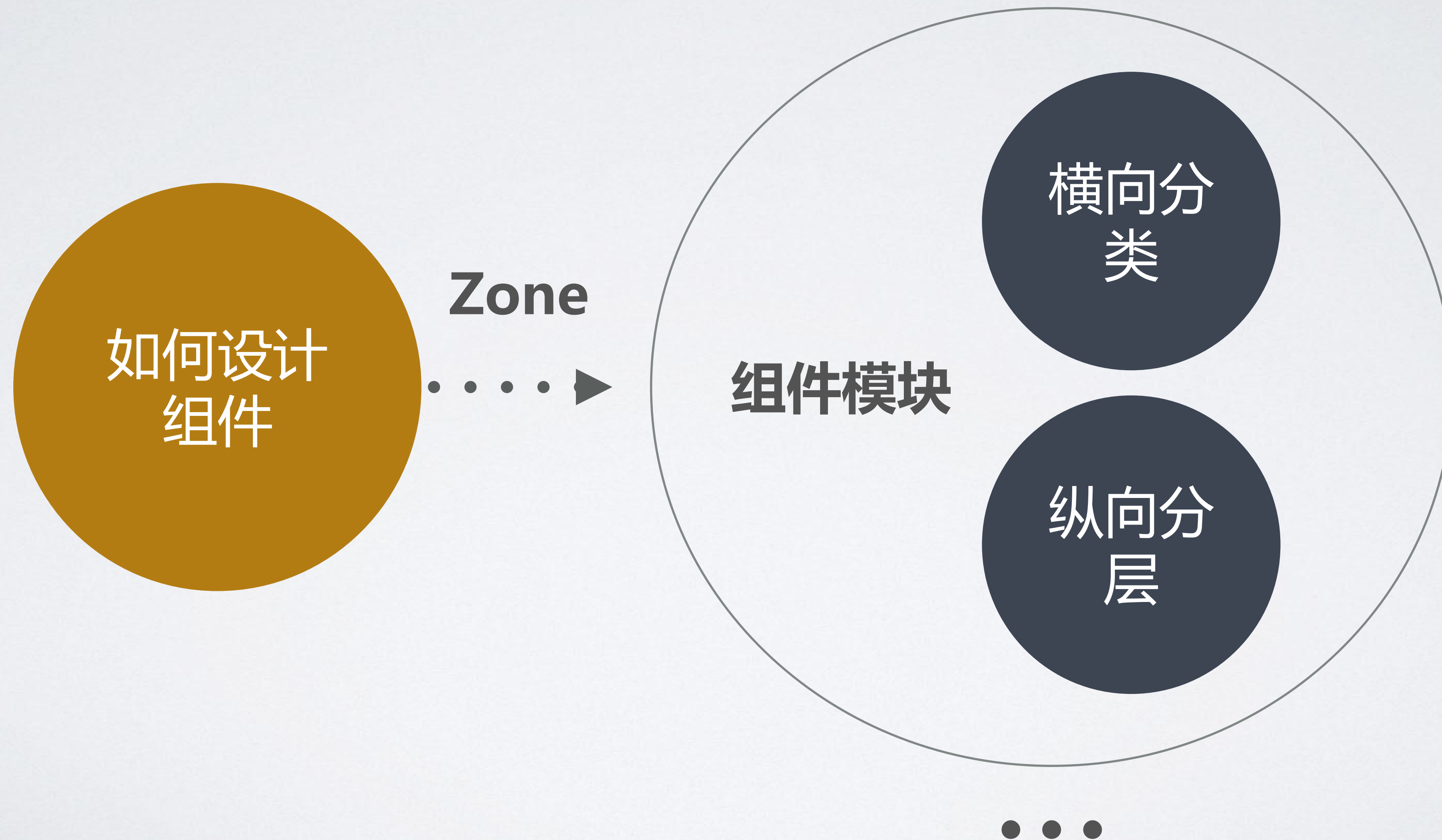
this.setState((state, props) => {
  return {
    isOpen: !state.isOpen
  }
});
```

实际操作呢？

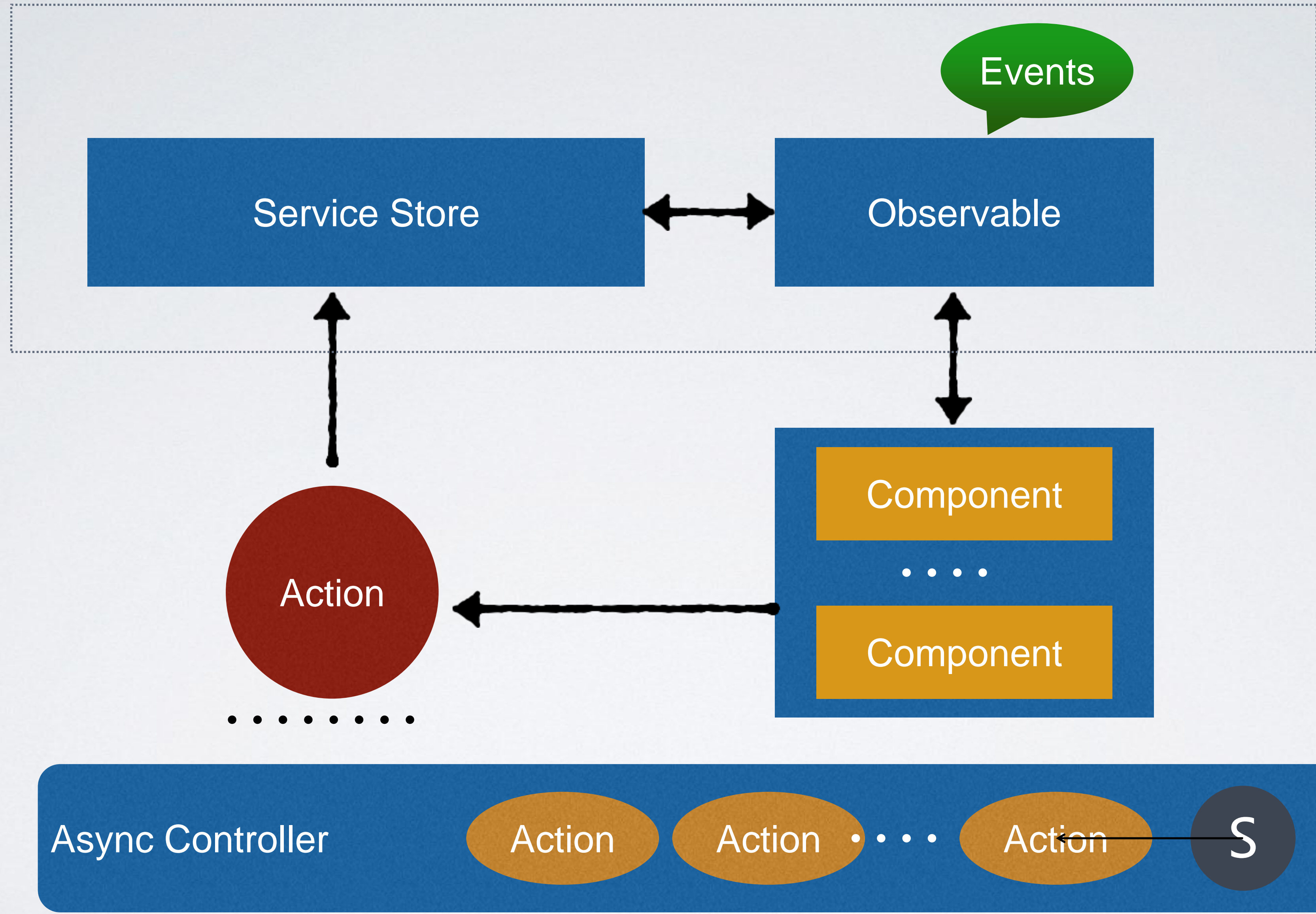
使用他们的场景？

- 我就只是想要一个临时用的静态弹框
- 我想要在表格每行触发弹框
- 我是多个页面都需要差不多一样的弹框
- 我担心这个需求改怎么办？

不能帮你解决的问题



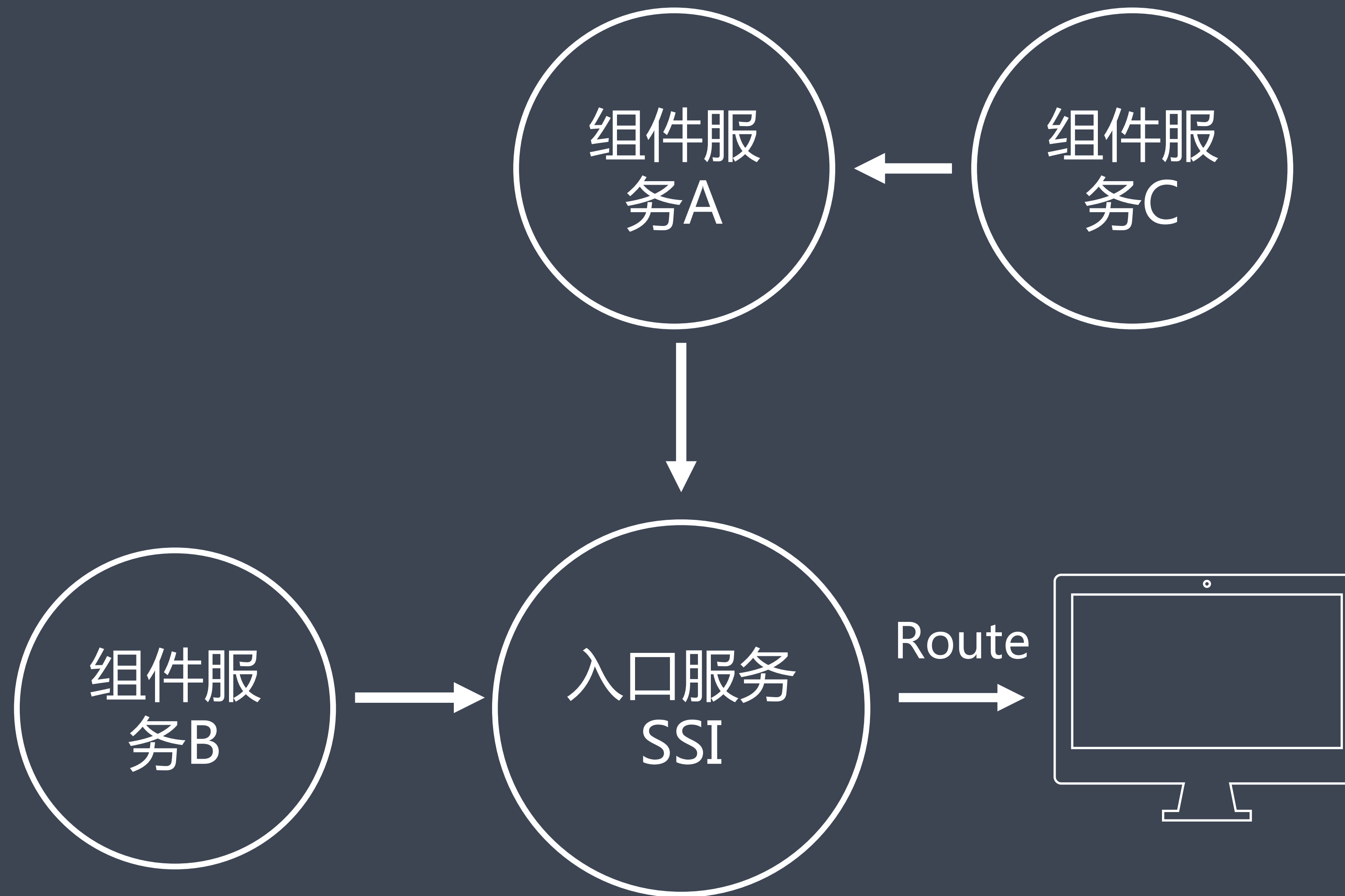
模块怎么“活”起来呢？



开发模块技术不约束



前端微服务化 => FAAS



<https://github.com/neuland/micro-frontends>

$O(n \cdot n)$ 谢谢

Q&A