组件化体系在NOW直播中的实践

React+Redux

IMWEB-willliang

介绍

- 腾讯高级工程师—梁伟盛（大圣）
- IMWeb团队成员
- 先后参与花样，交友，NOW直播等业务的核心开发和架构设计
- 现在负责互动视频业务前端架构设计与开发

NOW直播

素人直播

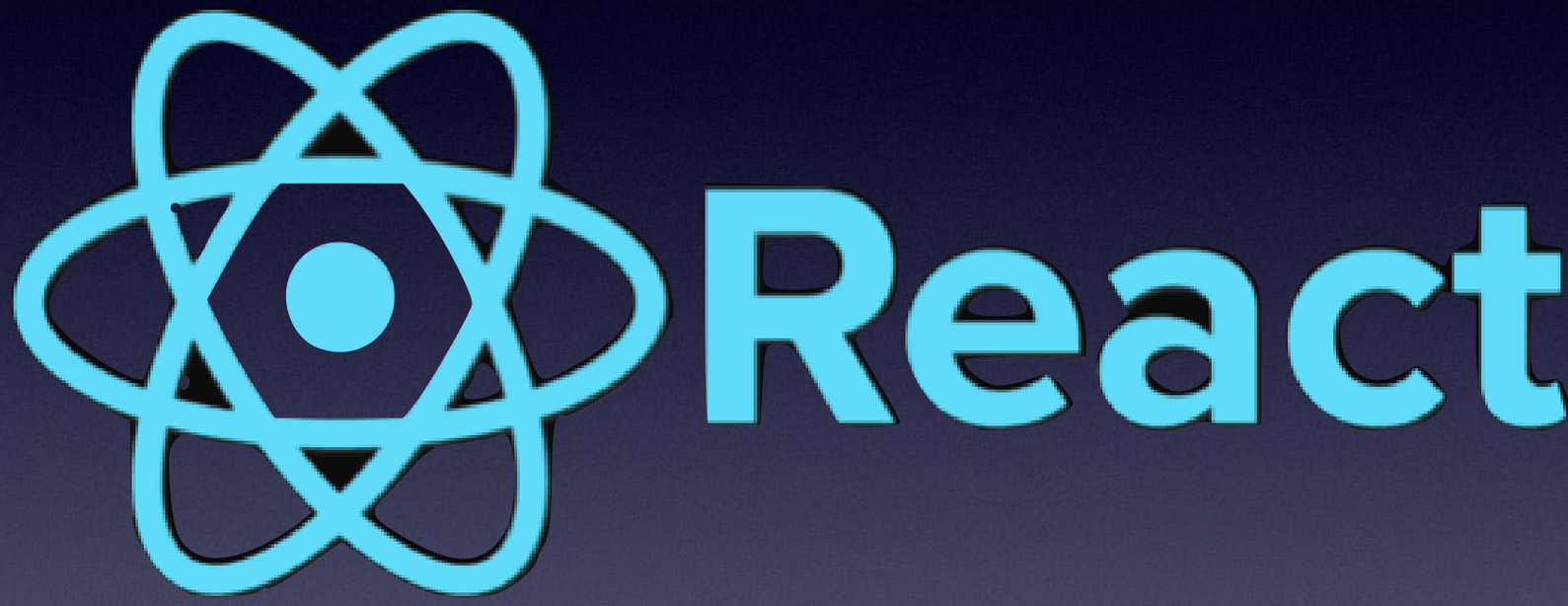

年糕呀
132人
+ 关注
魅力 727272
one
送一个爱你哦
x1
聪明的我❤
送一个爱你哦
x1
一个人世界有没有陪: 那么厉害
吓惹初♡ 来了
xuer: cf，dnf
陌浅离 来了
纯属“虚构”: 你说的我都成大神了
聪明的我❤ 送了 x 1个

花样直播

秀场直播

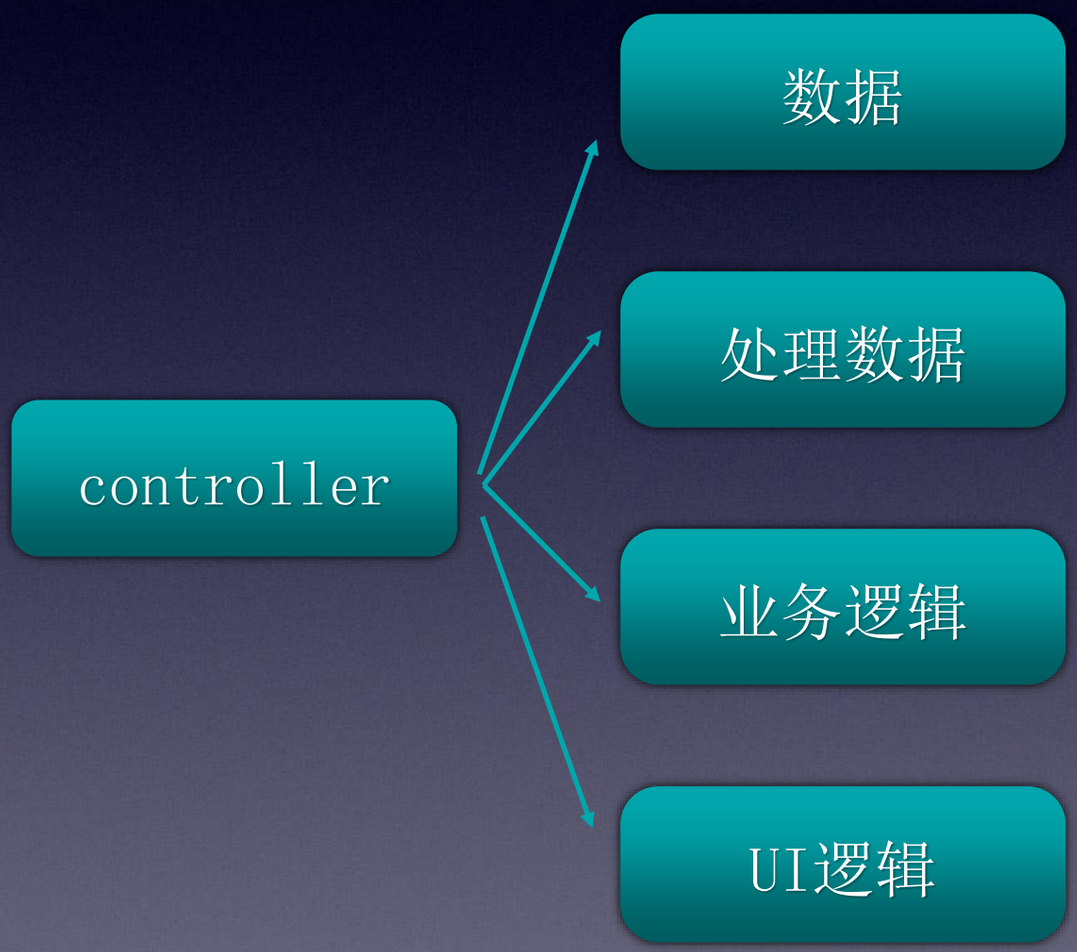

返回
子岩是个男孩...
8688人观看
APP
大厅
关注
子岩是个男孩子耶_____ 乖
子岩是个男孩子耶_____ 继续
流浪汉丨安静听歌丨
簧片百度云加QQ120... 送给主播 学习宝典 x 1
子岩是个男孩子耶_____ 继续啊
恋染到底丶醉酒 为毛我没有子岩徽章？？？
子岩是个男孩子耶_____ 刷10个
曾经我也开心过...隐形人 希望别揉耳朵
快与主播互动吧~

CONTENTS

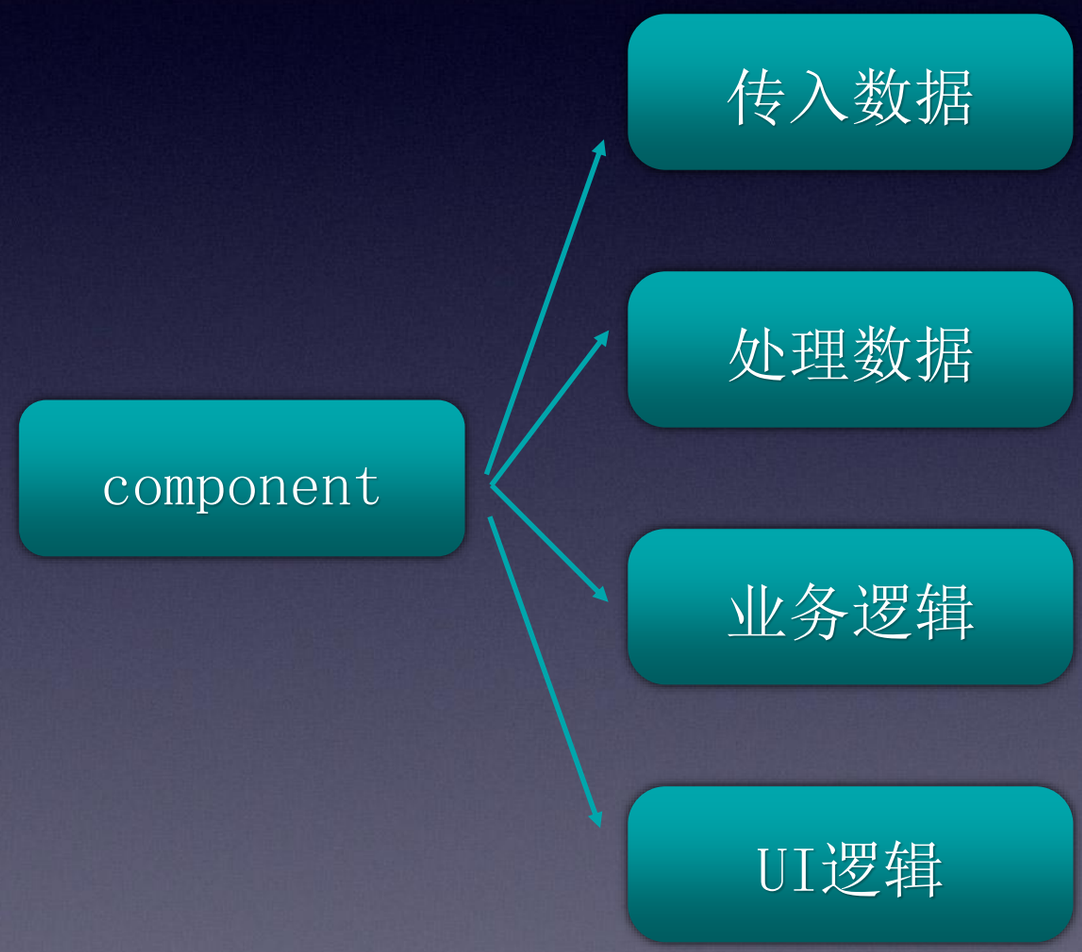

React

组件

angular



react



数据流向

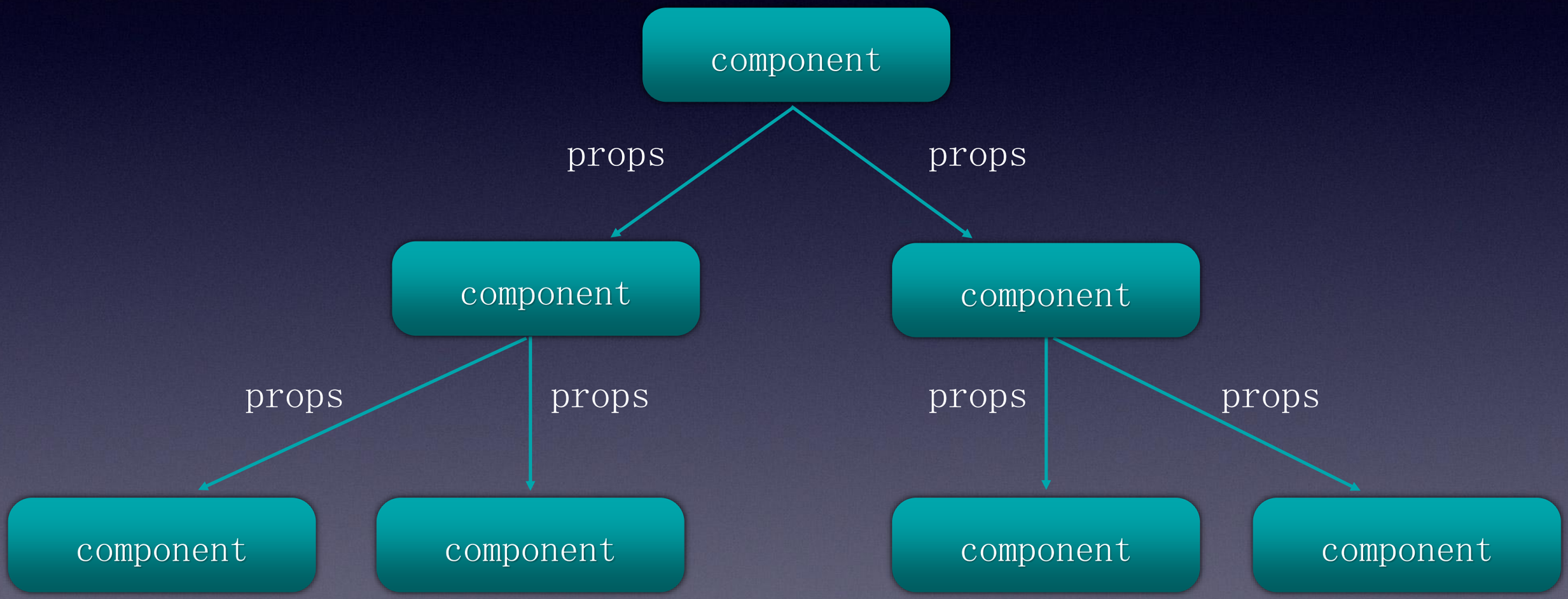
angular

- 没有规范数据流向

react

- 规范单向数据流

单向数据流


component
props
props
component
component
props
props
component
component
props
props
component
component

数据交互

数据交互

angular

- 通过observer做数据的监听交互？
- 父controller定义model传给子controller，互相监听交互？
- 定义model后，DI注入到controller中？

react

- 通过observer做数据的监听交互？
- 子component调用父component，由父component交互数据
- 使用context上下文

组件化

angular

- directive实现？

react

- 天生组件化

生命周期

angular

- 没有完善的生命周期

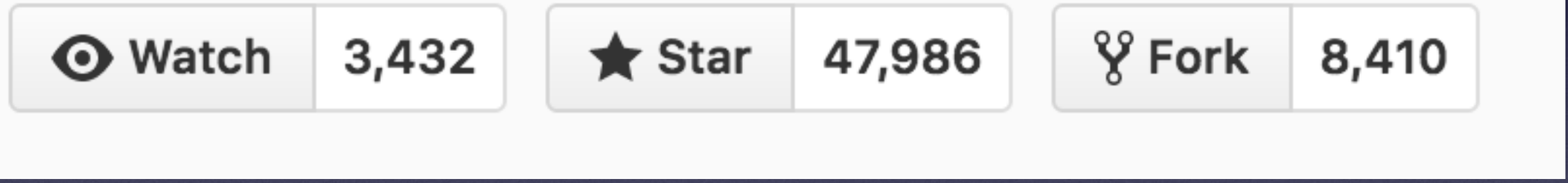
react

- componentWillMount
- componentDidMount
- componentWillUpdate
- componentDidUpdate
- 。。。

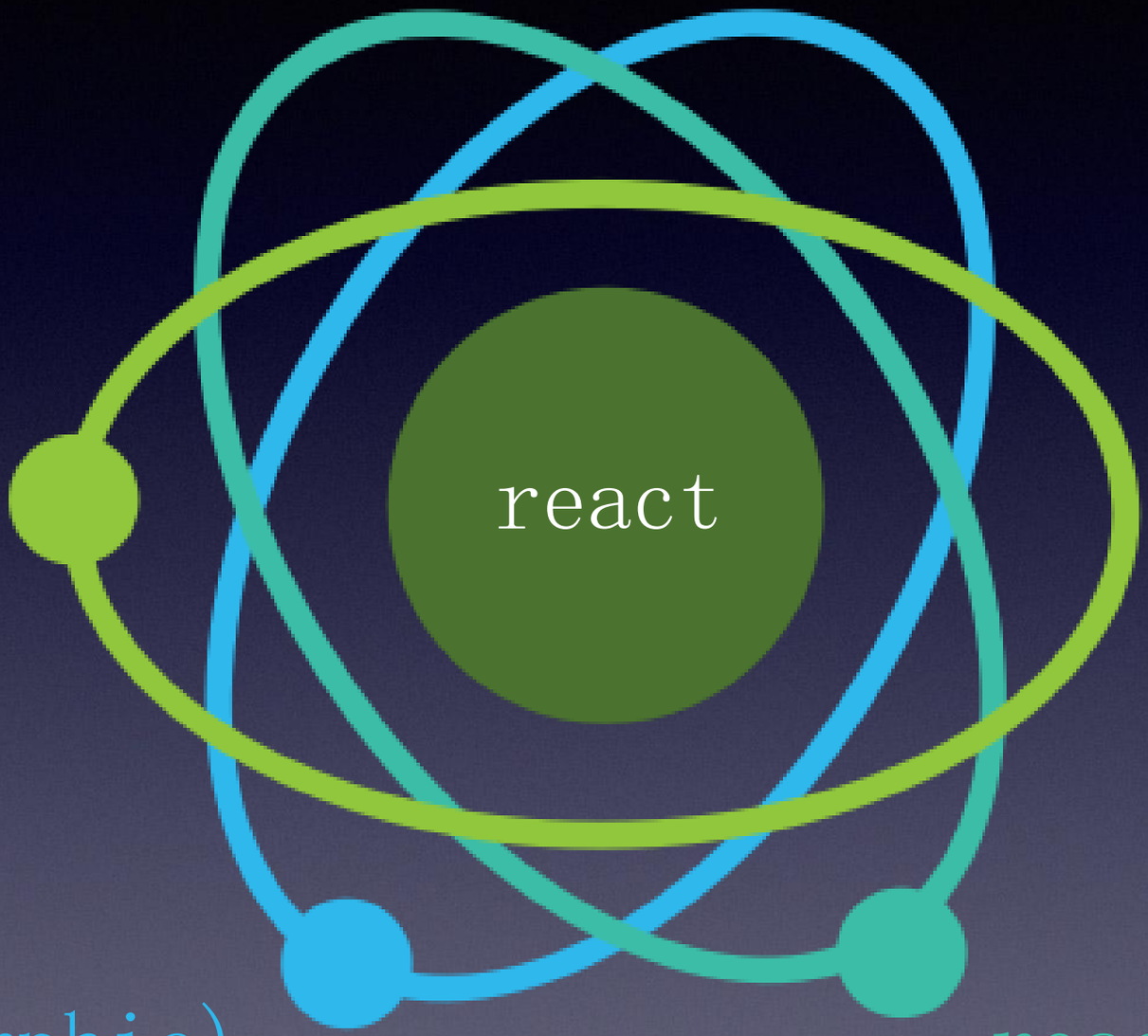
Why React?

- 非常完整的生命周期
- 天生组件化
- 单向数据流
- Virtual DOM
- JSX（同构）

强大的生态圈



丰富的周边



react-native

同构(Isomorphic)

react-redux

react缺点

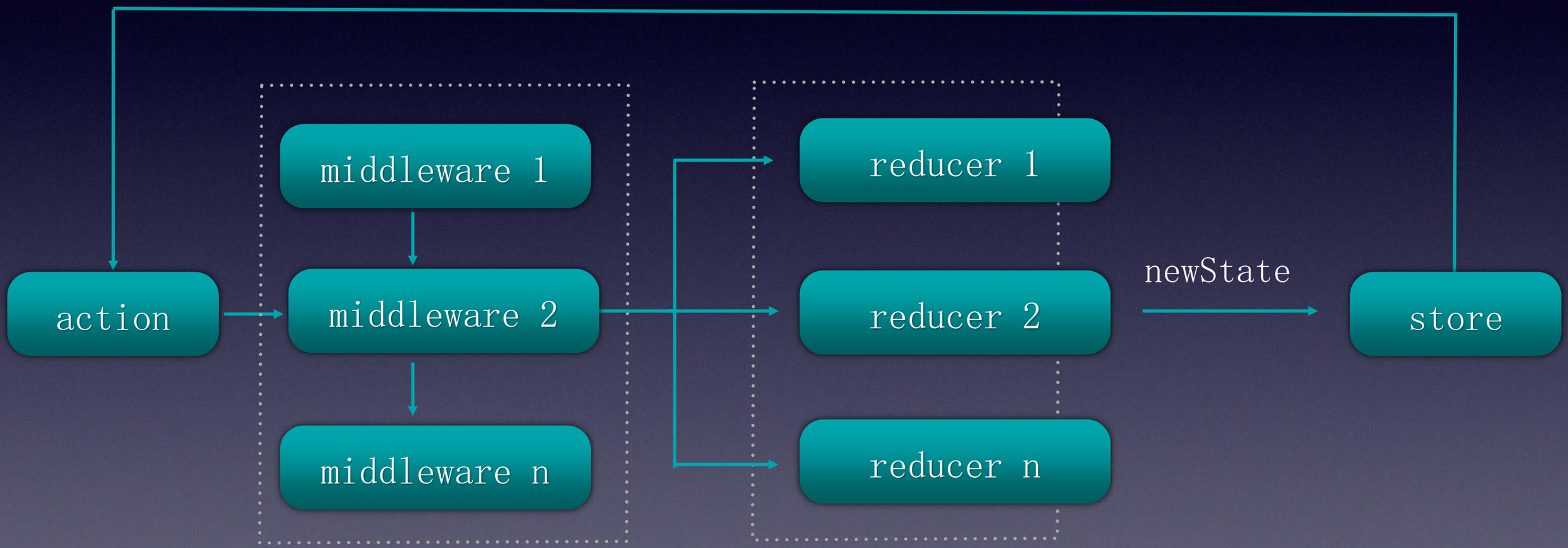
- component依然很重
- 单向数据流，一旦组件层次变深，传递数据会变得异常复杂
- 依然没解决数据交互问题

CONTENTS

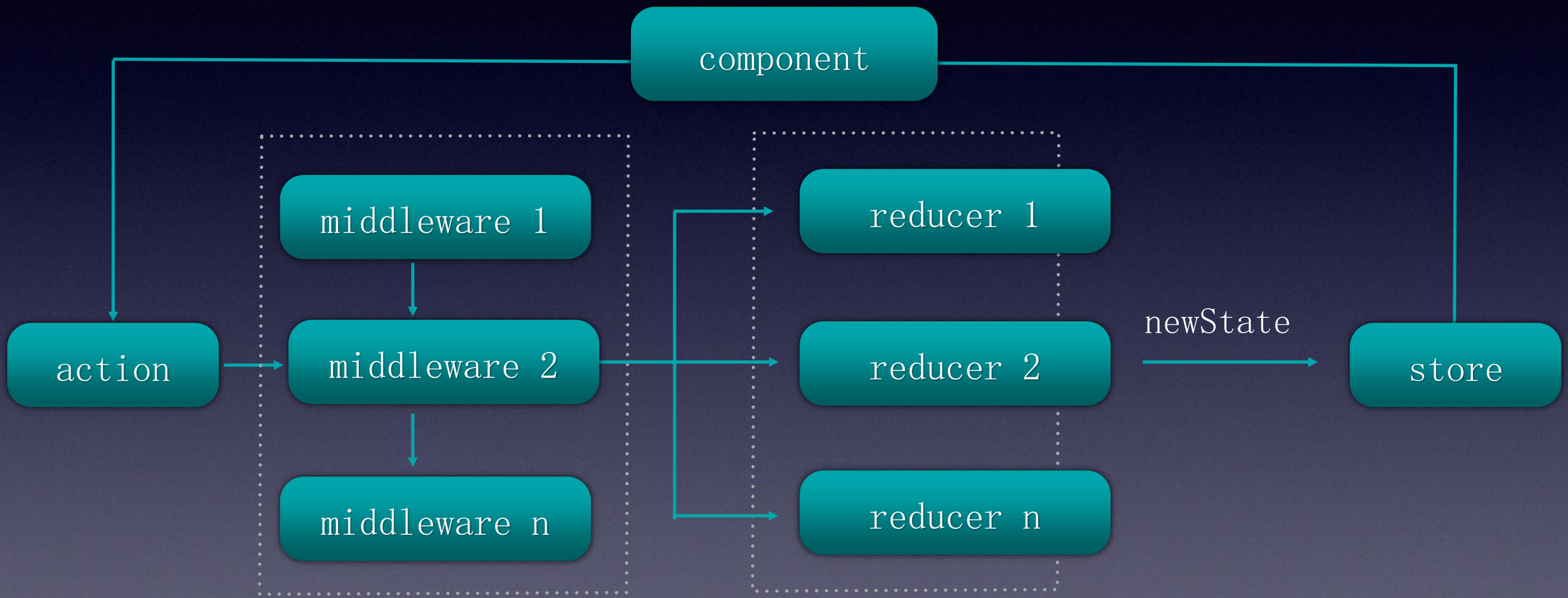
Why Redux?

- 单一数据源store
- action解耦
- react-redux提供了Provider和connect

单一数据源store


middleware 1
middleware 2
middleware n
action
reducer 1
reducer 2
reducer n
newState
store

React+Redux


component
middleware 1
middleware 2
middleware n
action
reducer 1
reducer 2
reducer n
newState
store

单向数据流?

react-redux提供了Provider和connect

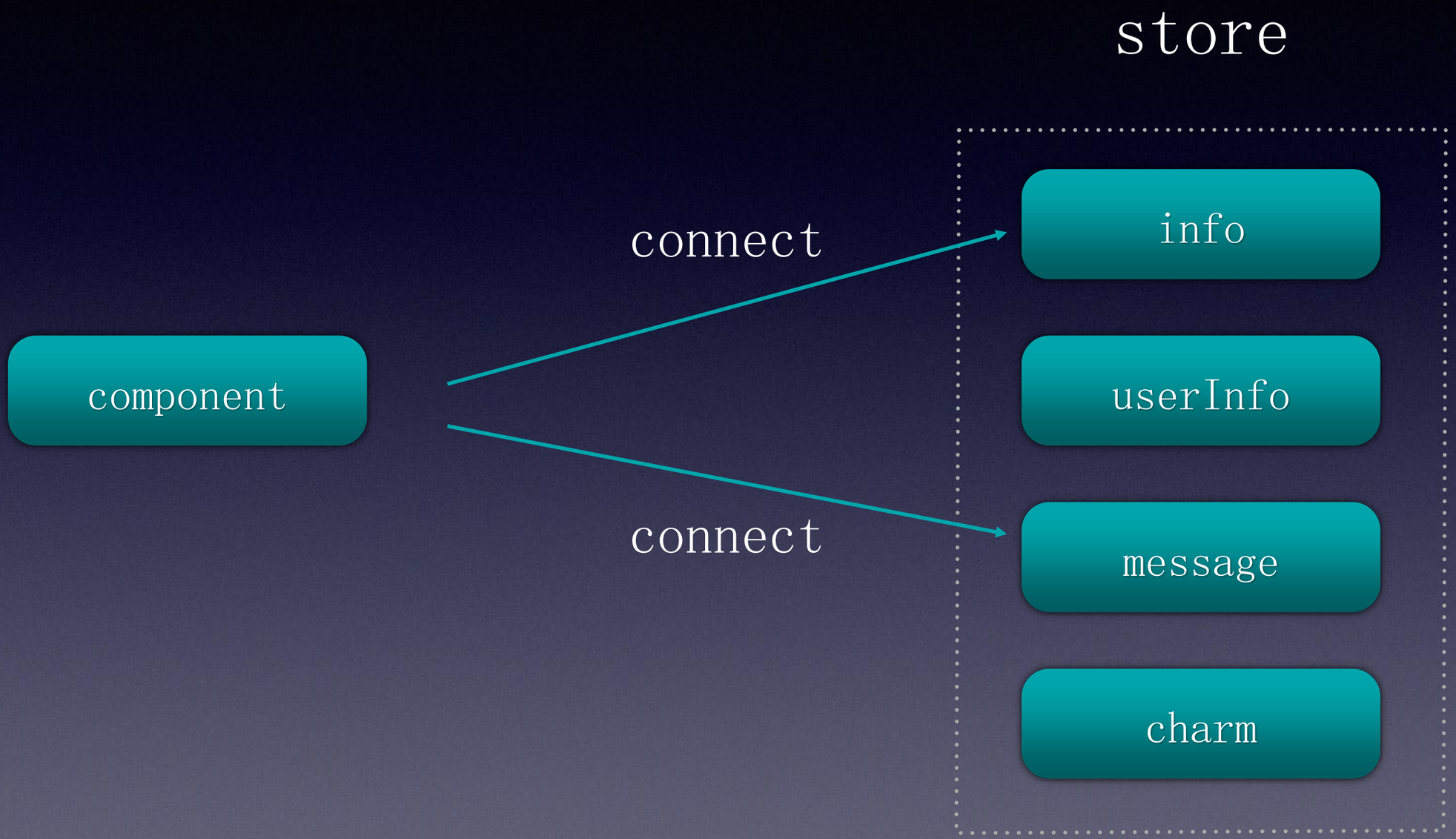
Provider

```
// connect
const Root = connect(function(state) {
    return state;
})(PageContainer);
```

```
ReactDOM.render(
    <Provider store={store}>
        <Root />
    </Provider>,
    document.getElementById('container')
);
```

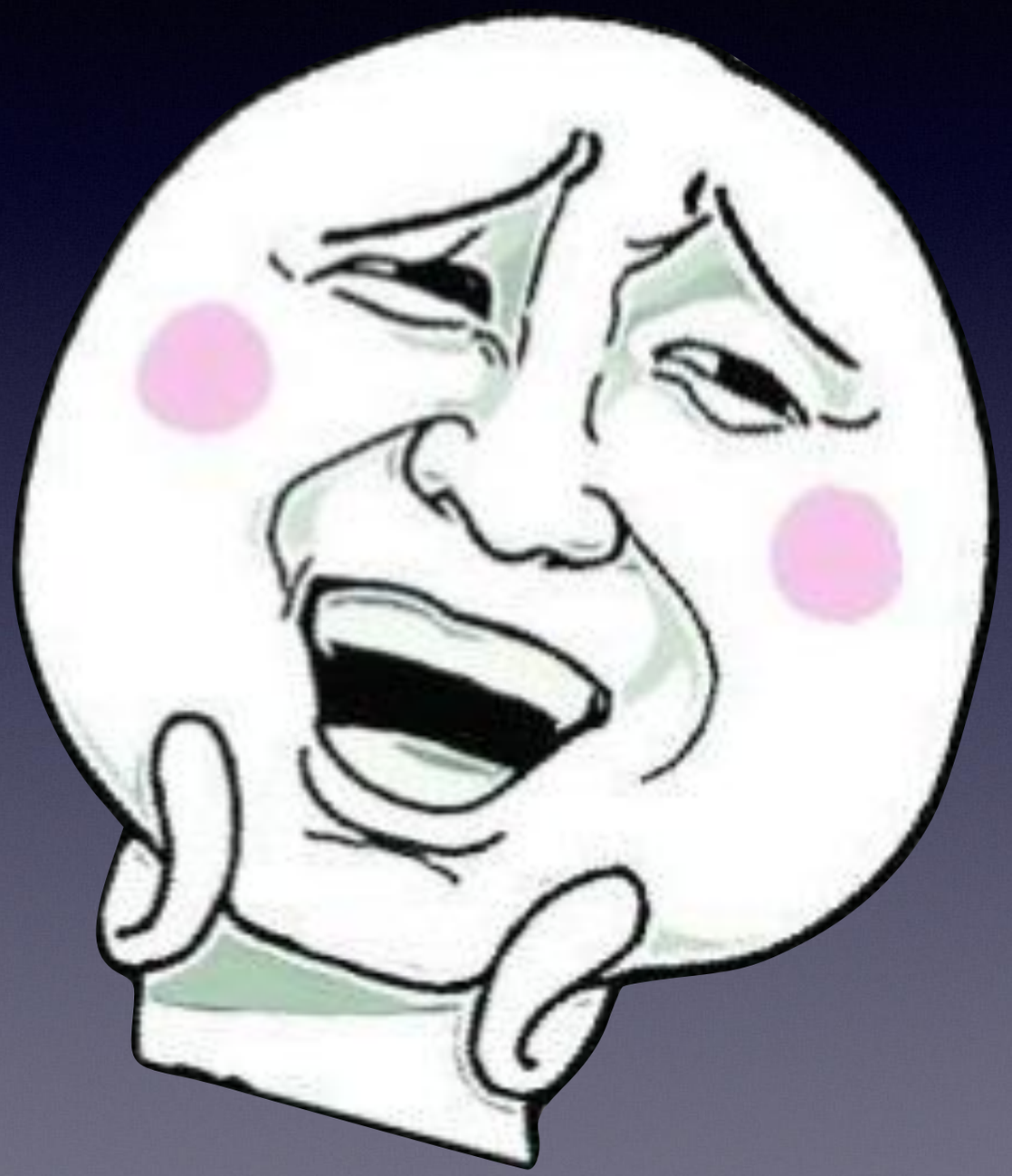
```
class Header extends React.component {
    render() {
        return (
            ...
        )
    }
}

export default connect((state) => {
    const {
        info
    } = state;

    return {
        info
    }
}, (dispath) => {
    return {
        ...
    }
})(Header)
```

connect


store
connect
info
component
userInfo
connect
message
charm

this.props拥有

妈妈不用担心我的数据从哪获取了

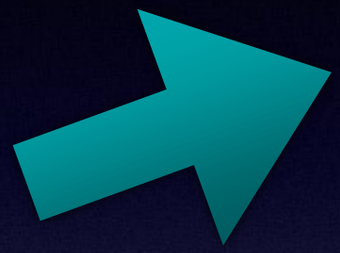


NOW直播

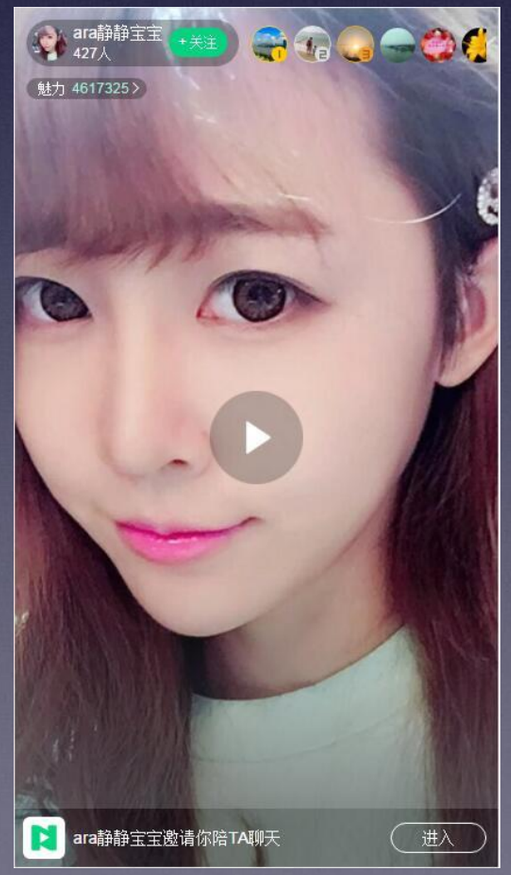

年糕呀
132人
+ 关注
魅力 727272
one
送一个爱你哦
x1
聪明的我❤️
送一个爱你哦
x1
一个人世界有没有陪：那么厉害
吓惹初♡ 来了
xuer：cf，dnf
陌浅离 来了
纯属“虚构”：你说的我都成大神了
聪明的我❤️ 送了 x 1个

CONTENTS


年糕呀
132人
+ 关注
魅力 727272
one
送一个爱你哦
×1
聪明的我❤️
送一个爱你哦
×1
一个人世界有没有陪：那么厉害
吓惹初♡ 来了
💋xuer👑：cf，dnf
陌浅离🍃 来了
纯属“虚构”：你说的我都成大神了
聪明的我❤️ 送了 x 1个




中国移动
20:57
18%
薇安bibi
924人
魅力 147626
回放
0:12
6:30


ara静静宝宝
427人
+关注
魅力 4617325
ara静静宝宝邀请你陪TA聊天
进入

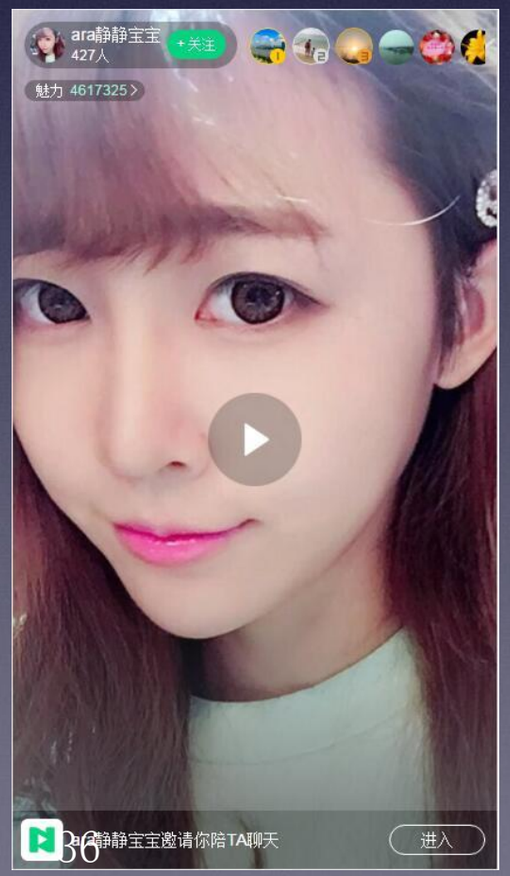
ctrl+c、ctrl+v大法好

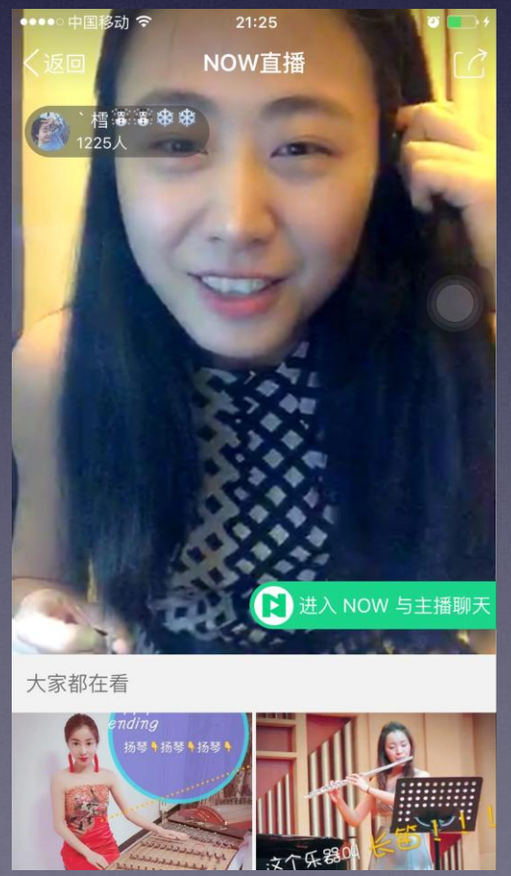
但这真的OK了么

业务越来越多，迭代越来越快


年糕呀
132人
+ 关注
魅力 727272
one
送一个爱你哦
x1
聪明的我❤️
送一个爱你哦
x1
一个人世界有没有陪: 那么厉害
吓惹初♡ 来了
xuer: cf，dnf
陌浅离 来了
纯属“虚构”: 你说的我都成大神了
聪明的我❤️ 送了 x 1个


中国移动
20:57
18%
薇安bibi
924人
魅力 147626
回放
0:12
6:30


ara静静宝宝
427人
+关注
魅力 4617325
ara静静宝宝邀请你陪TA聊天
进入


中国移动
21:25
返回
NOW直播
1225人
进入 NOW 与主播聊天
大家都在看

你的代码真的符合规范？
能快速切换不同的场景？
不同的业务？

无规矩不成方圆

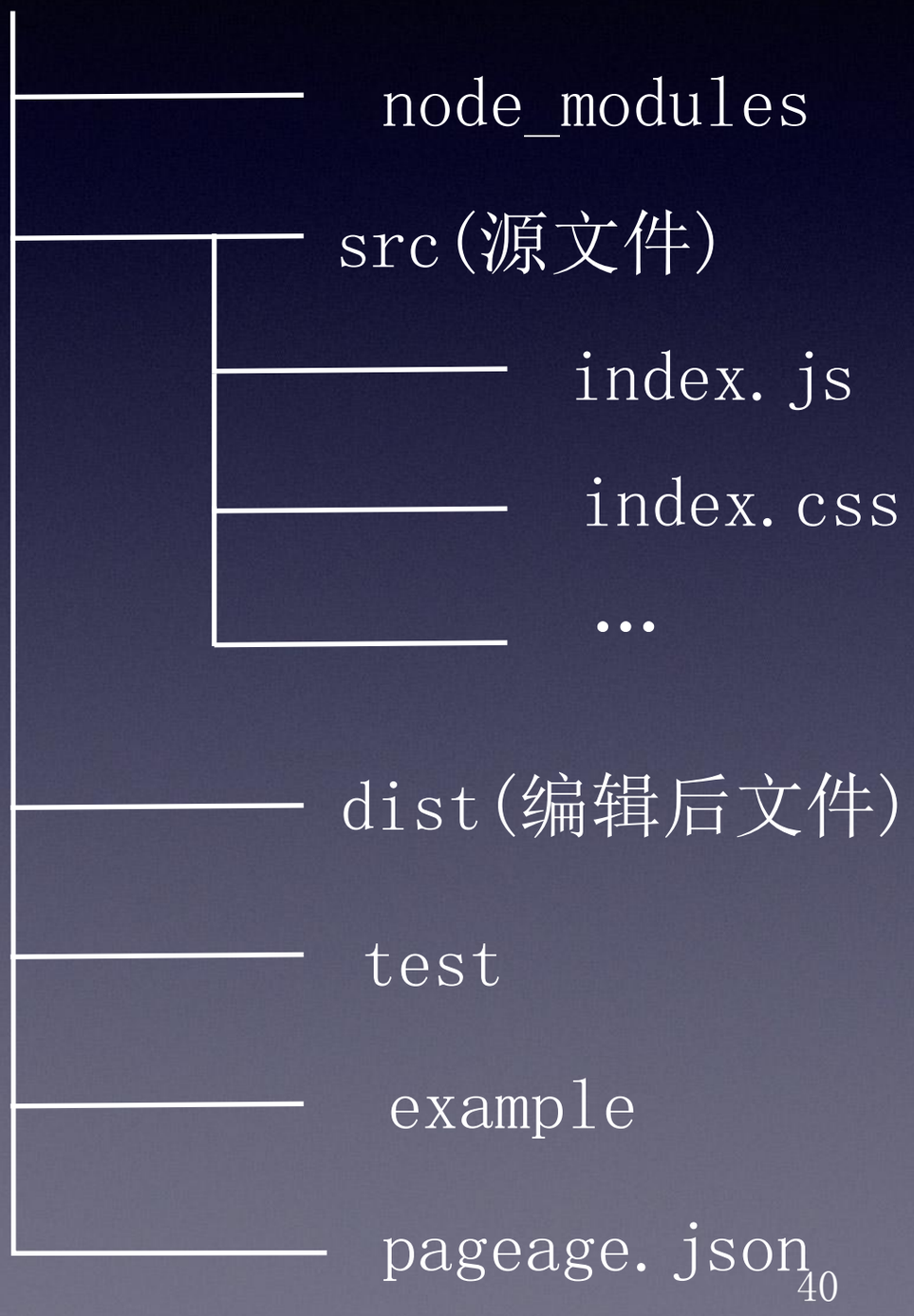
数据中心

展示组件

高阶组件

数据组件

NPM(包管理器)


node_modules
src(源文件)
index.js
index.css
...
dist(编辑后文件)
test
example
pageage.json

数据中心

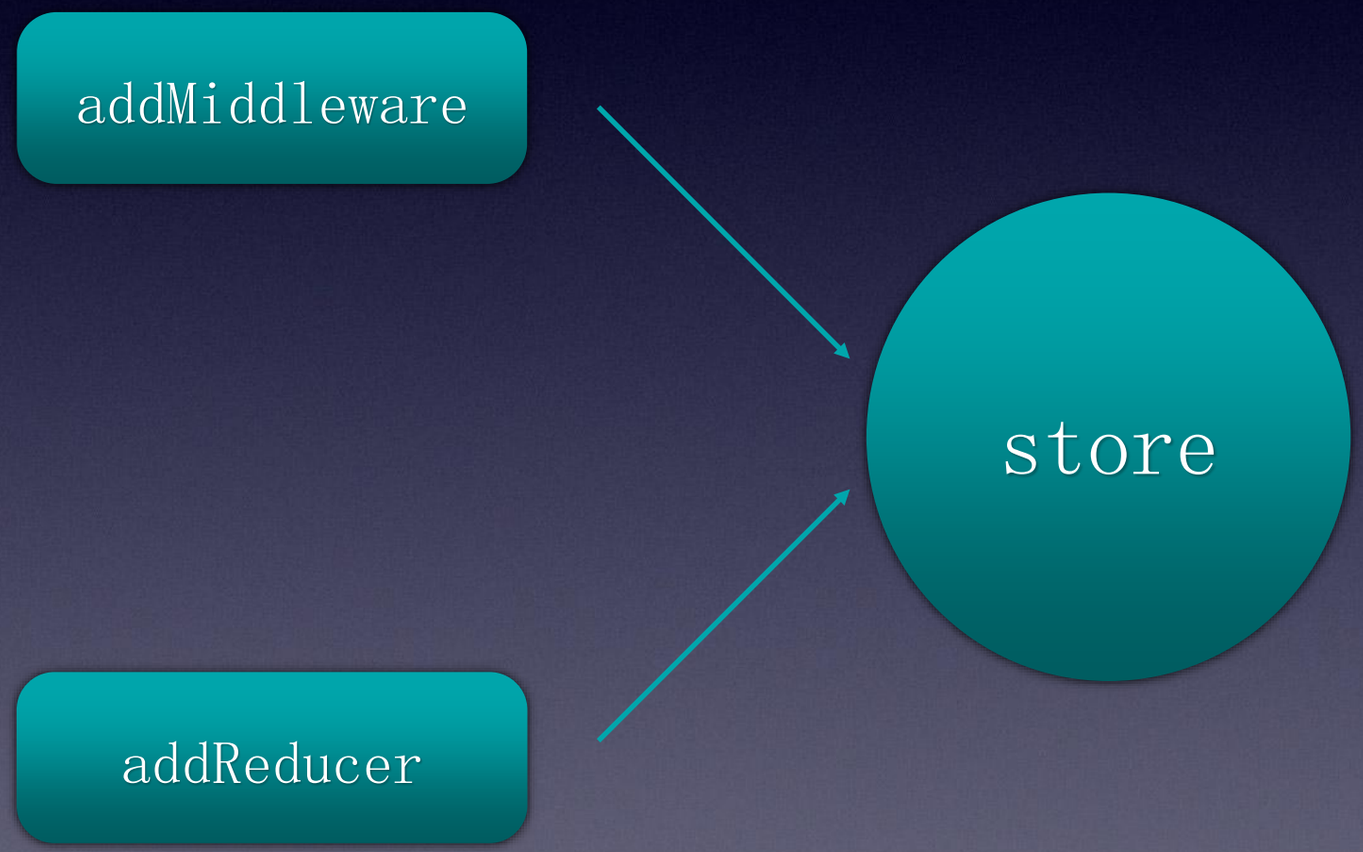
- 单一数据源store
- 暴露注入reducer与middleware的函数
- 暴露Provider与connect函数

静态写入

- store静态写入middleware与reducer

动态注入

- 新增addMiddleware函数负责动态注入middleware
- 新增addReducer函数负责动态注入reducer



以下哪些是业务代码？

A. 用户点击关注主播，修改数据变更关注状态

B. 用户点击头像，调用资料卡组件展示用户资料卡

C. 用户点击资料卡关闭按钮，关闭资料卡

以下哪些是业务代码？

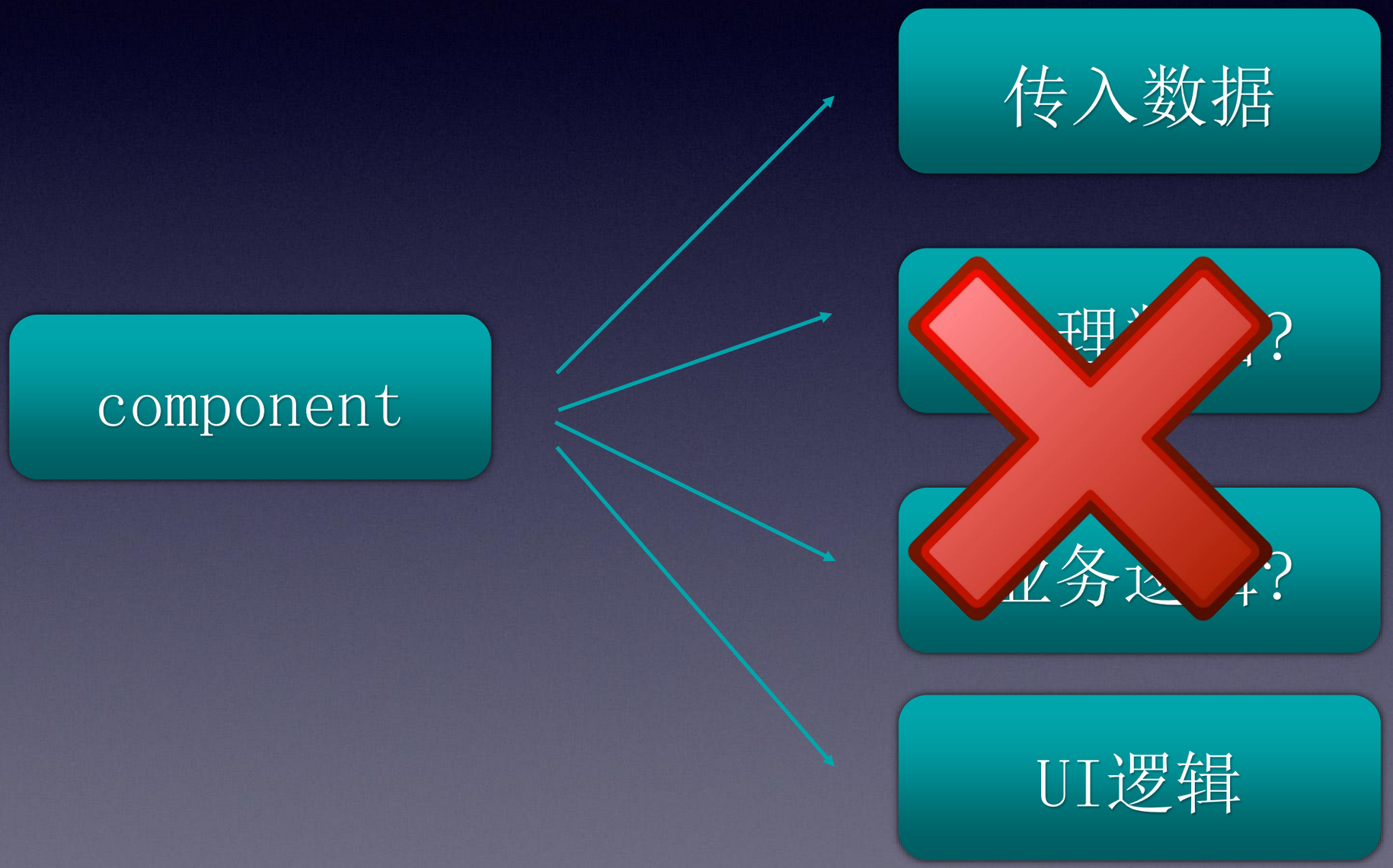
A. 用户点击关注主播，修改数据变更关注状态

B. 用户点击头像，调用资料卡组件展示用户资料卡

C. 用户点击资料卡关闭按钮，关闭资料卡

展示组件

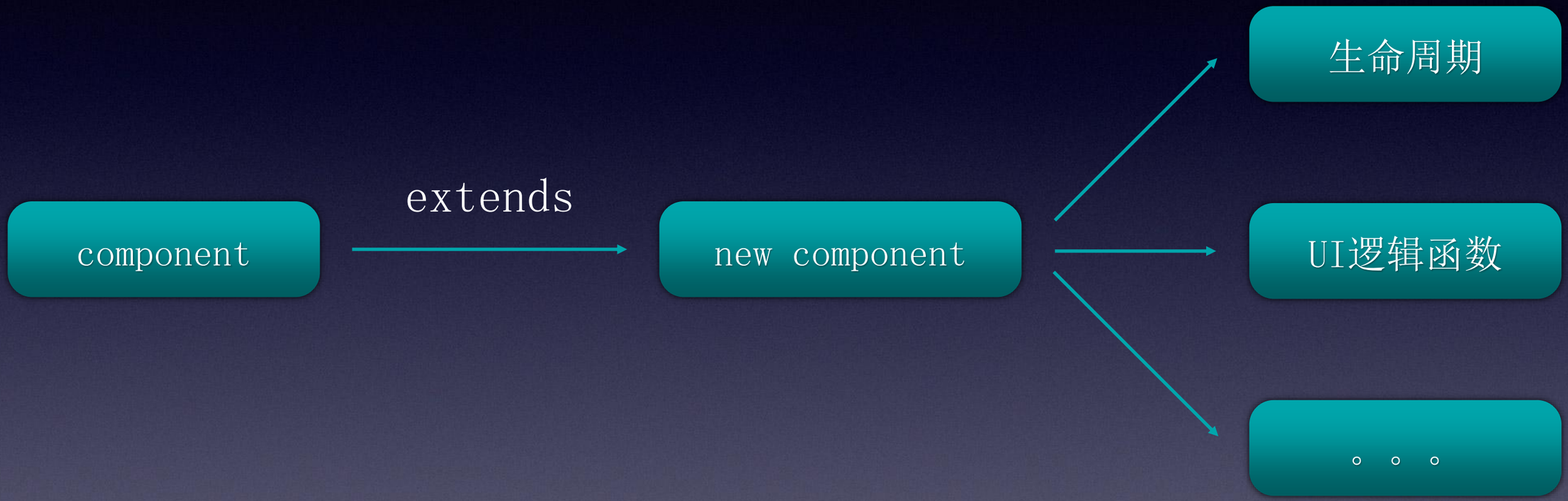
component



展示组件

```
class Header extends React.Component {
    render() {
        return (
            //...
        )
    }
}

Header.PropTypes = {
    name: PropTypes.string.isRequired
}
```

复用展示组件



展示组件

- 和正常react组件写法完全一致
- 不包含业务逻辑，只包含展示和交互逻辑
- 必须使用PropTypes校验字段
- 复用方式extends
- 不与数据组件耦合

数组组件

action

```
//请求状态
export const REQUEST = 'REQUEST';
//请求成功状态
export const REQUEST_SUCCESS = 'REQUEST_SUCCESS';
//请求失败状态
export const REQUEST_FAIL = 'REQUEST_FAIL';

function fetch() {
    return {
        ...
    }
}

export function load(arg) {
    return (dispath, getState) => {
        return dispath(fetch(arg));
    }
}
```

reducer

```
import {
    REQUEST,
    REQUEST_SUCCESS,
    REQUEST_FAIL
} from './action'

export default function info(state = {
    name: ''
}, action) {
    switch(action.type) {
        ...
    }
}

addReducer({
    info
});
```

数据组件

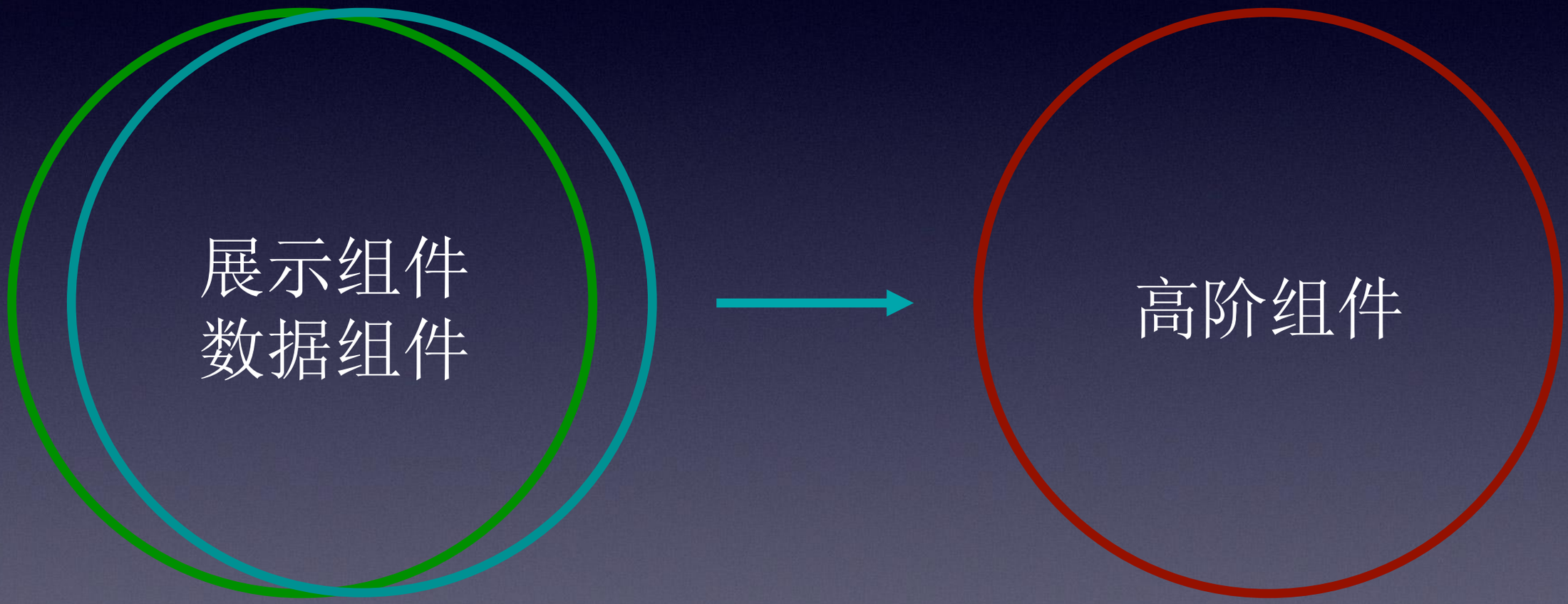
- redux为基础框架，包含action与reducer
- 依赖数据源组件，使用时，将reducer注入到数据源组件中
- 不与展示组件耦合

高阶组件

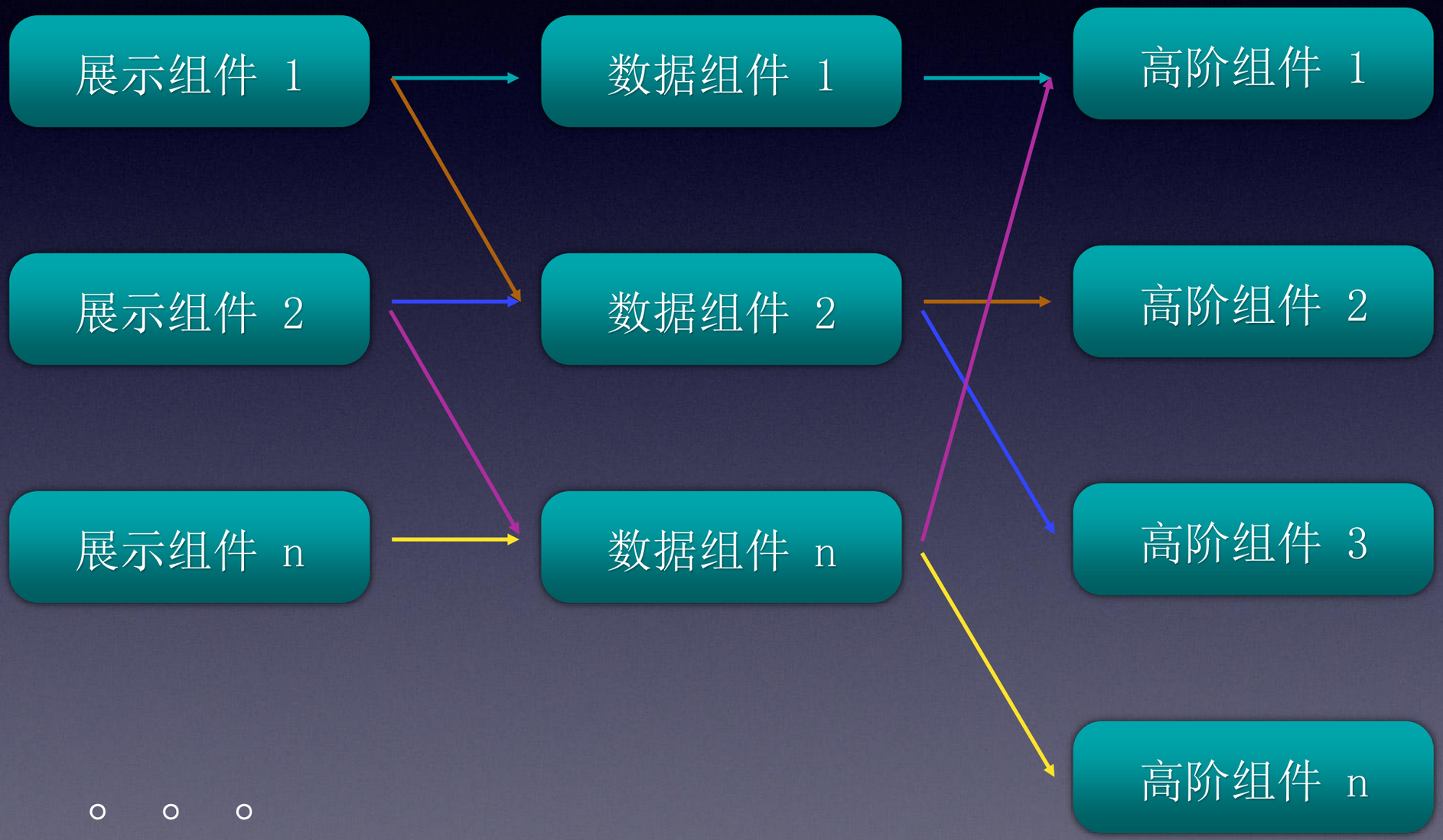
展示组件connect数据组件

```
//引入展示组件
import Header from 'header';

//引入数据组件
import info from 'info';

export default connect((state) => {
    const {
        info
    } = state;

    return {
        info
    }
}, (dispath) => {
    return {
        //编写业务逻辑代码
    }
})(Header)
```

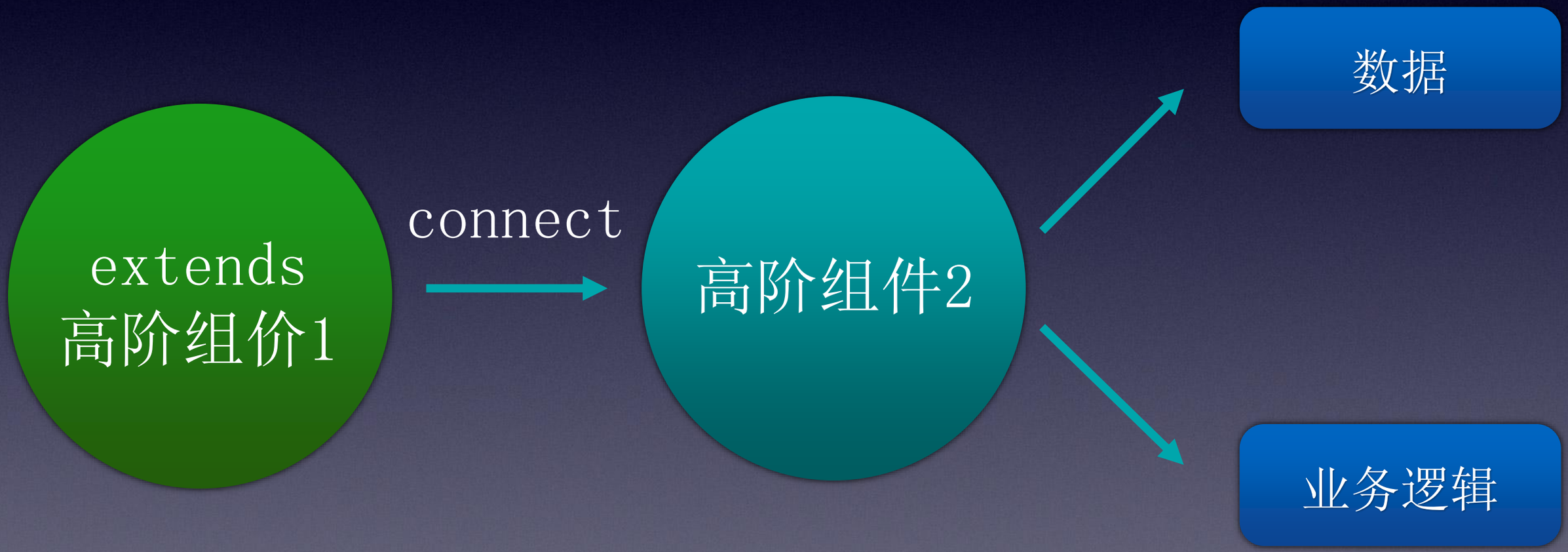
connect



组件关系(N对N)

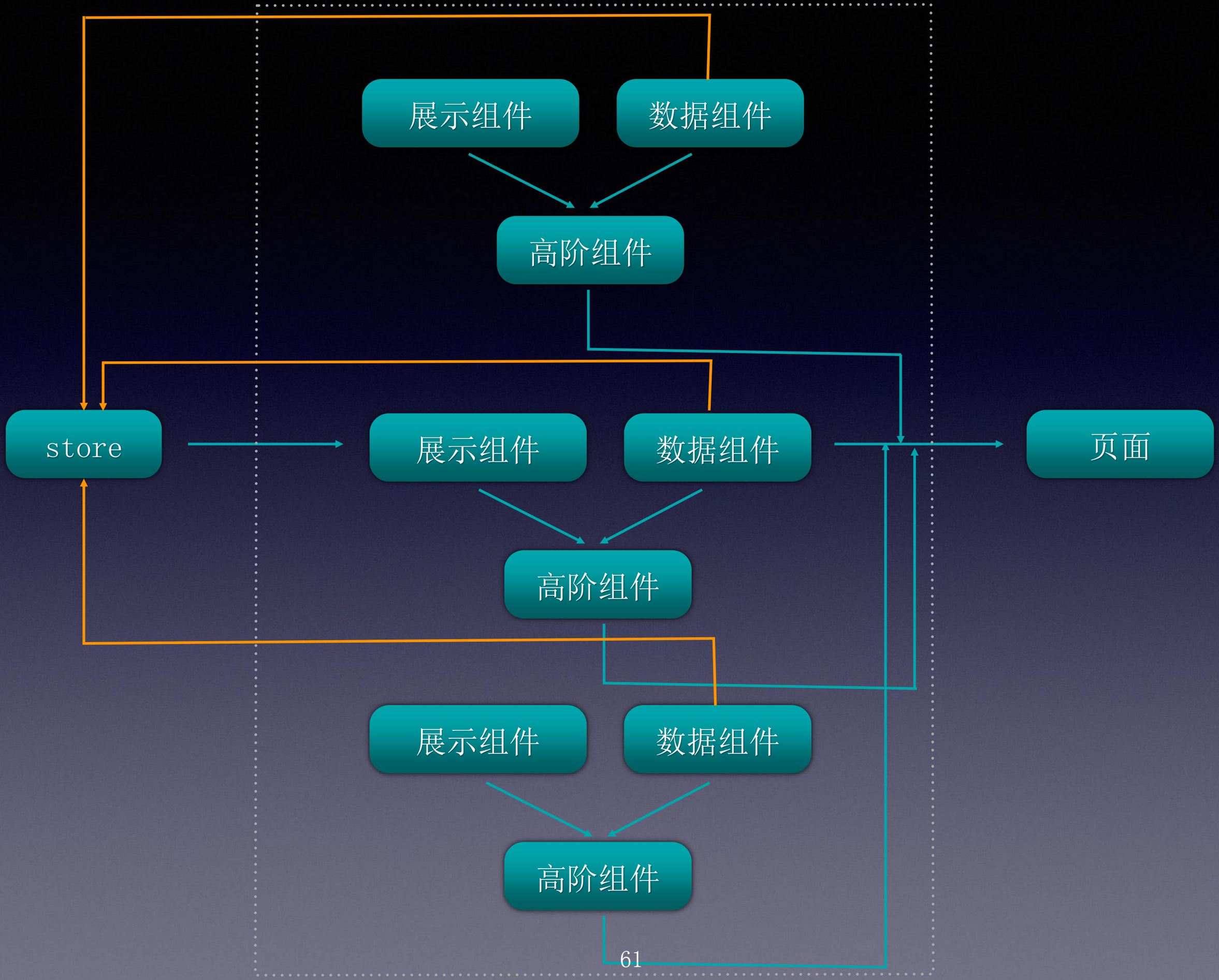

展示组件 1
展示组件 2
展示组件 n
数据组件 1
数据组件 2
数据组件 n
高阶组件 1
高阶组件 2
高阶组件 3
高阶组件 n
。 。 。

extends+connect



高阶组件

- 展示组件与数据组件组成高阶组件
- 在connect编写业务逻辑
- 复用使用extends+connect
- 展示组件与数据组件是多对多关系
- 高阶组件高度聚合，而展示组件和数据组件间又充分解耦


展示组件
数据组件
高阶组件
store
展示组件
数据组件
页面
高阶组件
展示组件
数据组件
高阶组件

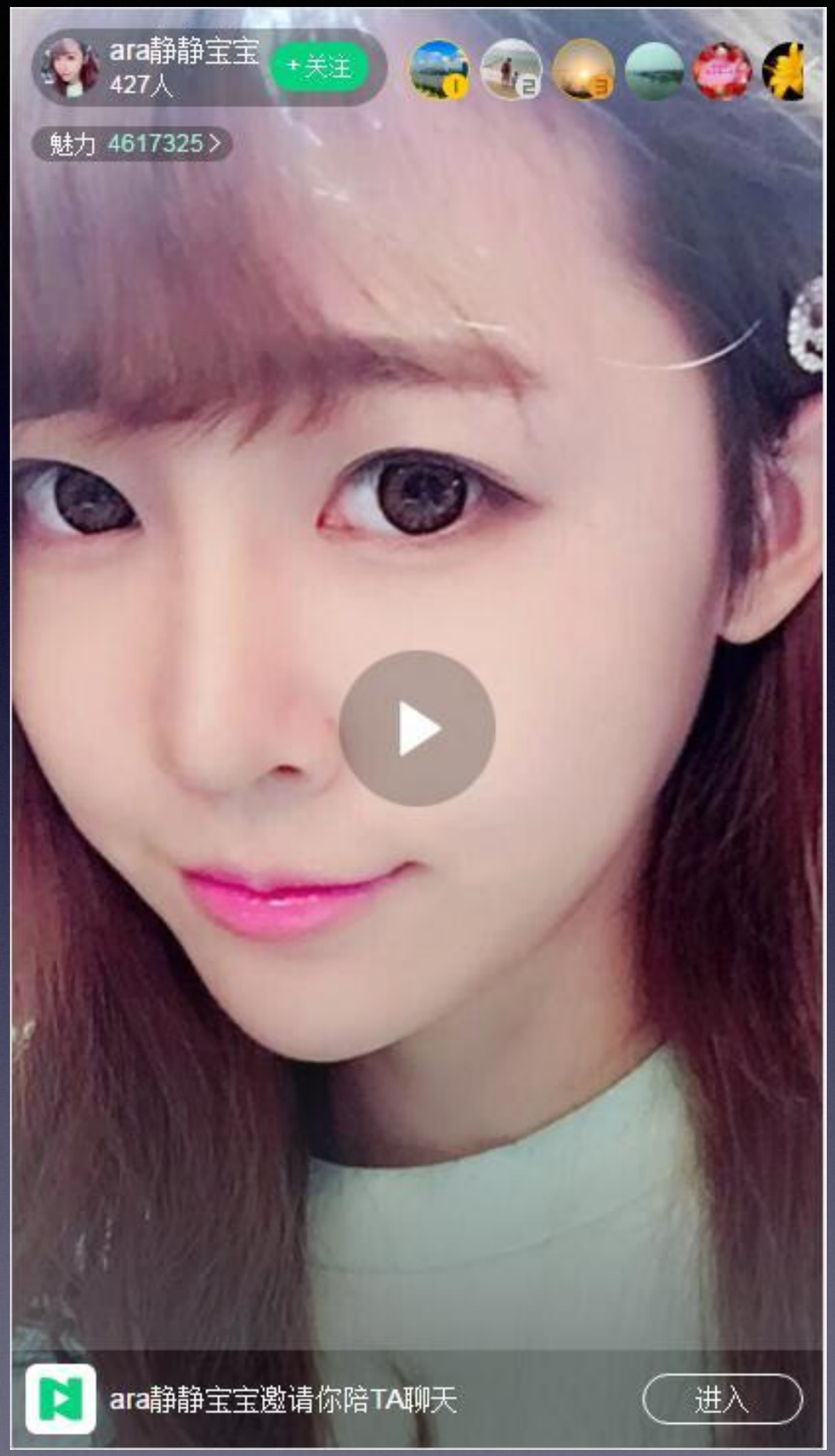
CONTENTS

组件分解

- 头部展示(highorder-header)
- 聊天展示(highorder-message)
- 礼物展示(highorder-giftmsg)
- 点赞飘心(highorder-bubble)
- 视频(highorder-video)
- 。。。

组件分解

- 头部展示(nowjs-highorder-header)
- 聊天展示(nowjs-highorder-message)
- 视频(nowjs-highorder-video)



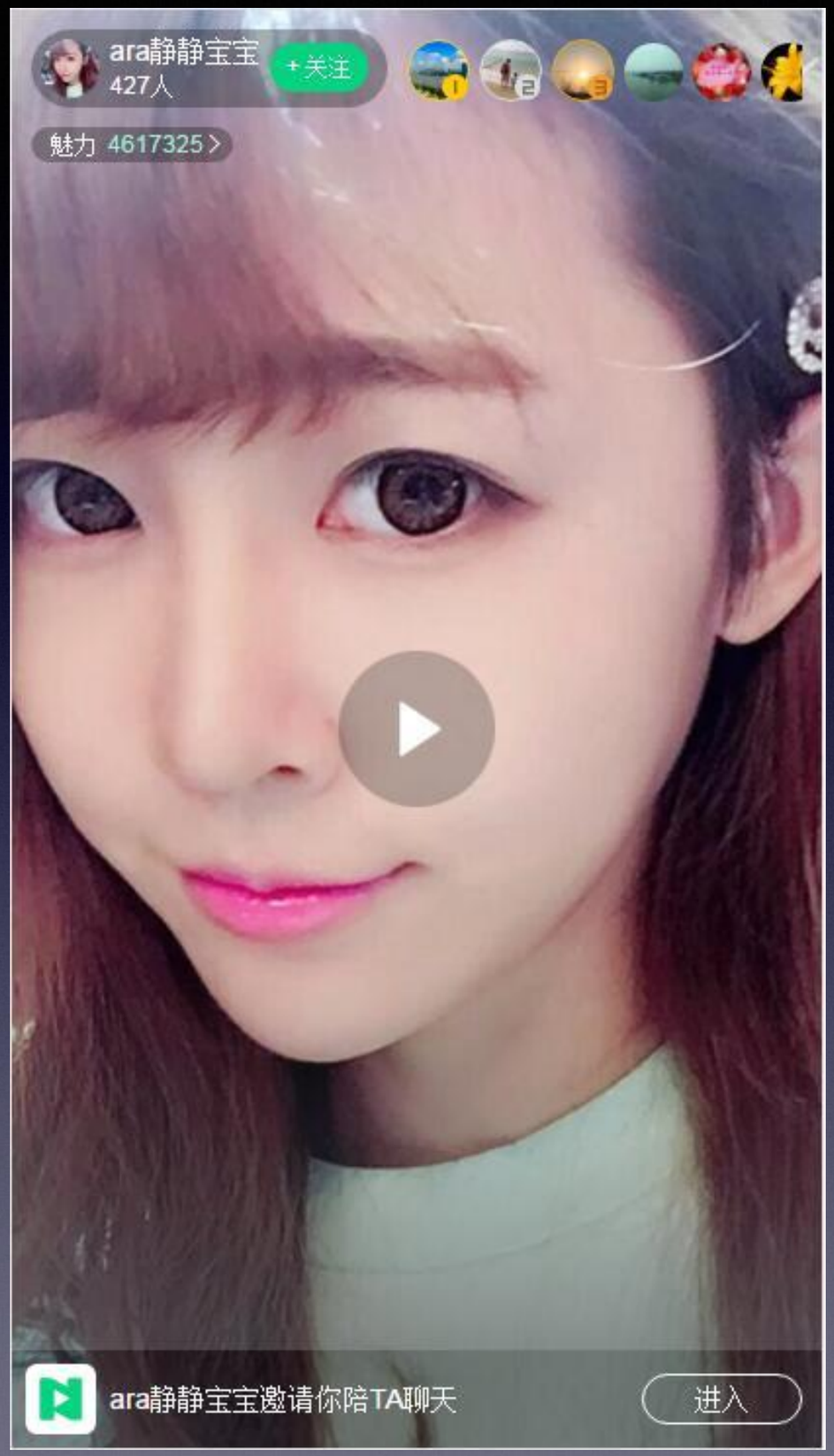
可复用的高阶组件

```
install highorder-header highorder-message highorder-v
```

搭建高阶组件

```
import Header from 'highorder-header';
import Message from 'highorder-Message';
import Video from 'highorder-video';

class PageContainer extends React.component {
    render() {
        return (
            <div id="root">
                <Header />
                <Video />
                <Message />
            </div>
        )
    }
}

const Root = connect((state) => {
    return state;
})(PageContainer);

ReactDOM.render(
    <Provider store={store}>
        <Root />
    </Provider>,
    document.getElementById('container')
)
```


年糕呀
132人
+ 关注
魅力 727272 >
one
送一个爱你哦
×1
聪明的我❤️
送一个爱你哦
×1
一个人世界有没有陪: 那么厉害
呀惹初♡ 来了
💋xuer👑: cf，dnf
陌浅离🍃 来了
纯属“虚构”: 你说的我都成大神了
聪明的我❤️ 送了 x 1个


ara静静宝宝
427人
+关注
魅力 4617325 >
ara静静宝宝邀请你陪TA聊天
进入

架构的优势

- 组件的引用简单(npm install)
- 快速搭建项目与快速切换不同的场景
- 展示组件与数据组件之间实现的低耦合，而连接两者的高阶组件实现了高内聚

Future

- react+redux开源组件化平台
- 组件平台可视化编辑组件
- 。。。


明天有个活动，全部人都排下期


没问题
没问题
不行

谢谢