

搜狗前端工程化探索之路

段振兴 2016.11.24

ABOUT ME

盛大网络

2010

腾讯(QQ空间)

2011

百度云

2012

腾讯/搜狗

2013

AGENDA

石器时代

工程化时代



石器时代

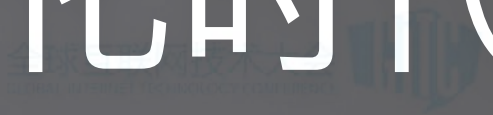
痛点

无法本地调试

静态资源文件
缺乏组织

复用、维护性较差

基本无前端
自动化工具



工程化时代

第一次工业革命

本地可调试

目录规范 + 模块化开发

自动化构建 + 一键发布

web安全加固



本地可调试

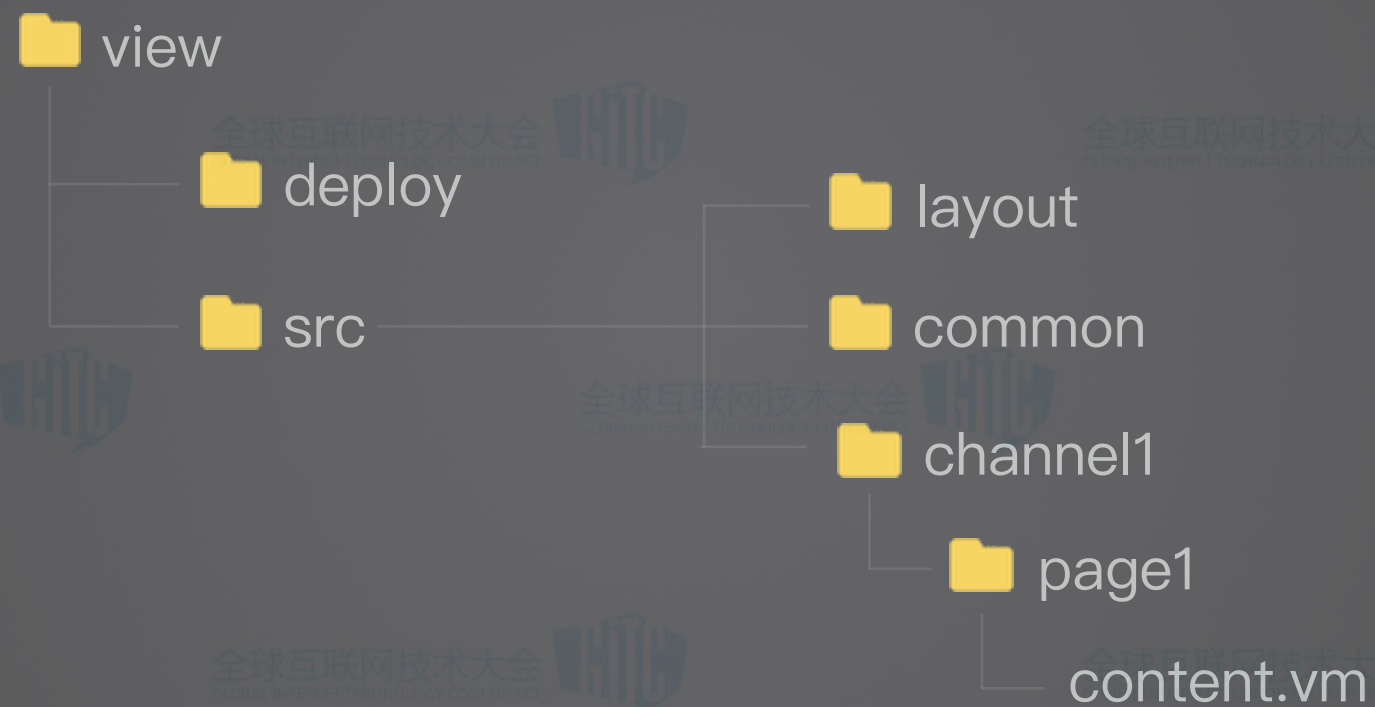


SpringMVC

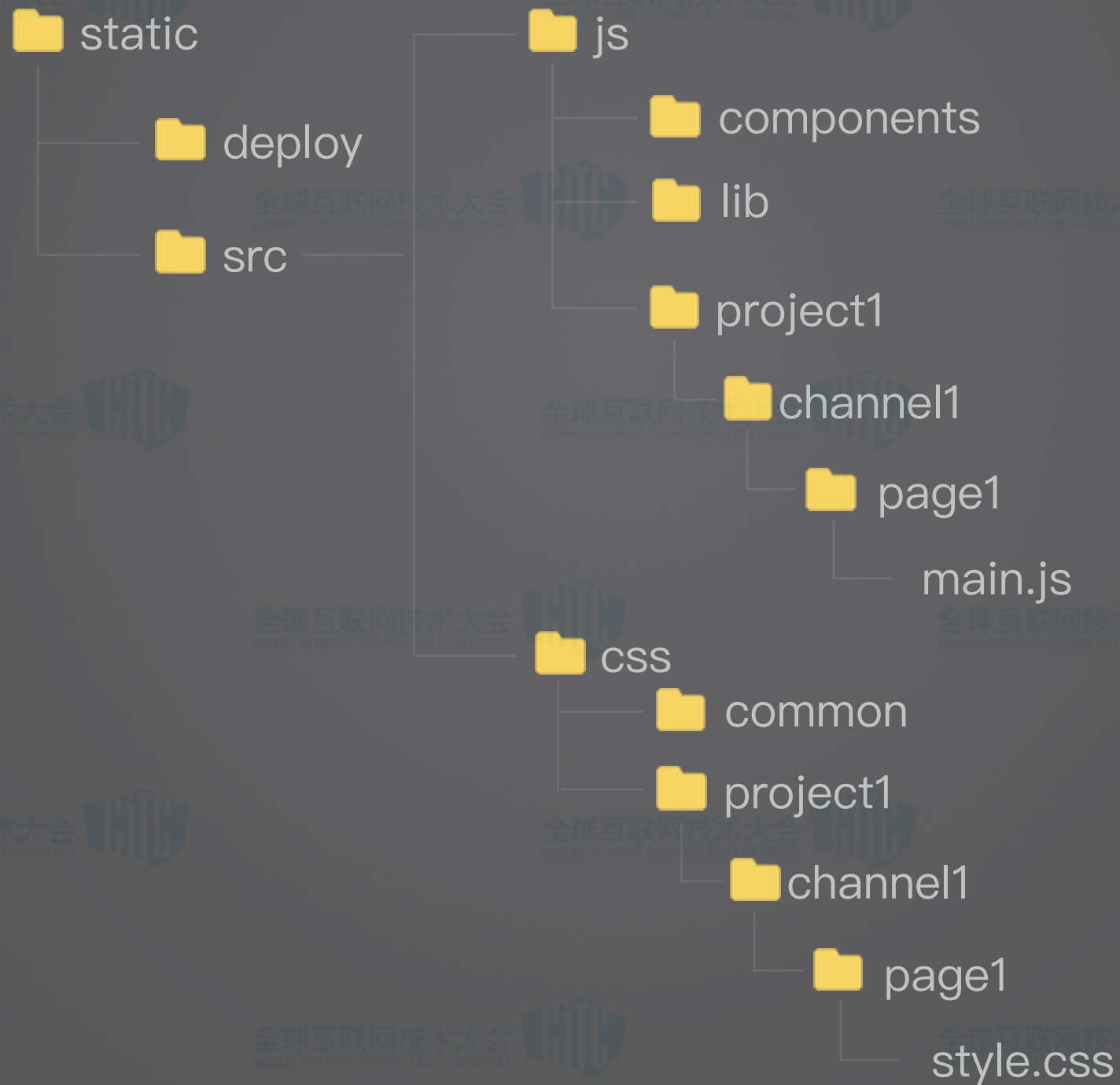
Velocity模板



Velocity模板目录规范



静态资源目录规范



JS模块化

Sea.js

```
define(function(require, exports, module){  
    //code  
});
```



CSS模块化

@import

SASS

自动化构建工具wenke

v1.0

Sea.js

Grunt

配置繁琐
编译效率不高

自动化构建工具wenke

启动
wenke

扫描分析Velocity模板，提取需
要构建的静态资源列表

处理静态资源文
件并编译模板

生成静态
资源映射表

对比静态资源映射表增量上传
CDN，校验文件指纹

你想愉快的发布吗？

一键发布

发布系统

首页 / [redacted] / 发布

机器管理 发布历史 打基线 拉分支

主干最新代码 [dropdown] 从主干打Tag

test preview deploy 全部

选择	部署路径	<input type="checkbox"/> 最近一次发布	<input type="checkbox"/> 当前版本	<input type="checkbox"/> 操作
<input type="checkbox"/>	[redacted]	2016-11-11 16:04:58	trunk	锁定 导出覆盖率

全选 备注: [text area]

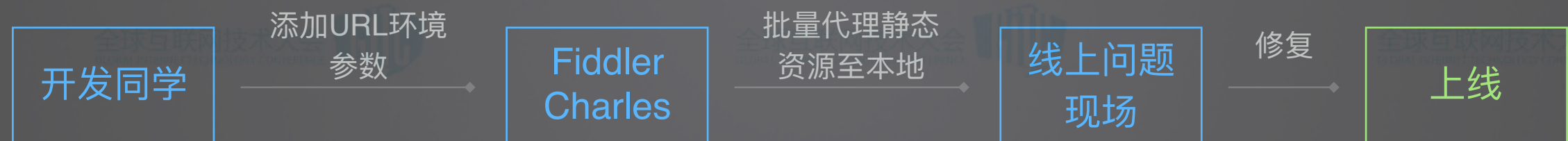
行行好，写点备注吧，避免忘记干了啥

☒ 串行发布 ☐ 并行发布 (慎选)

☐ 前后端联合发布 ☒ 前端独立发布

发布

快速定位线上问题



危险字符 <、>、&、'、" 过滤

安全相关Cookie HttpOnly化

Content Security Policy

Web安全加固

CSRF防御

CGI接口安全扫描

XSS前端主动防御

静态代码质量检查工具

JSLint

调试工具

Chrome Inspect

weinre & vConsole

Fiddler & Charles

待优化

只想写CommonJS模块，不想写define函数

想写ES6，用class更好地组织代码

提升构建性能

有没有比Grunt更好的构建工具？

升级wenke

v2.0

Browserify

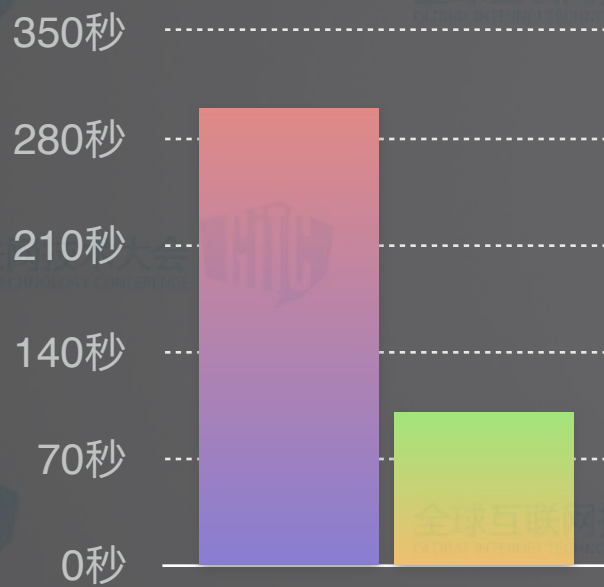
Gulp

引入Babel
支持ES6

别忘了
兼容IE8

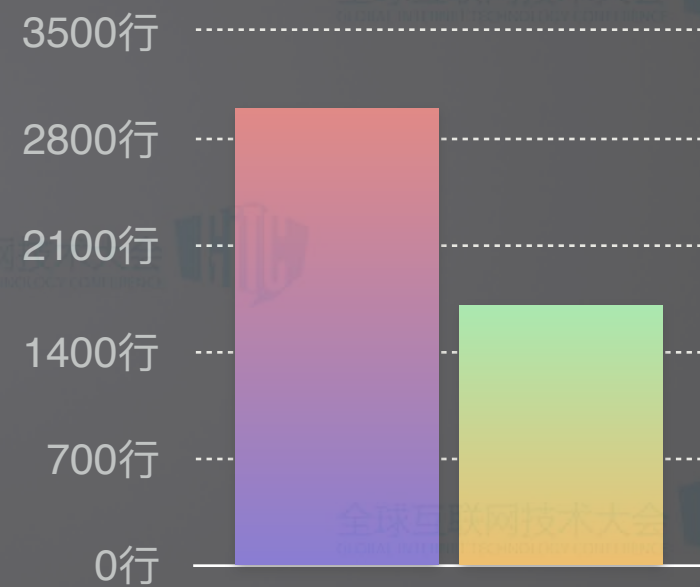
重构后

构建时间峰值



■ 优化前 ■ 优化后

构建工具代码量



■ 优化前 ■ 优化后

wenke-dev

实时开发辅助工具

引入Babel, 转换ES6

实时编译Browserify模块

Gulp.watch + socket.io = LiveReload

wenke-dev 工作流

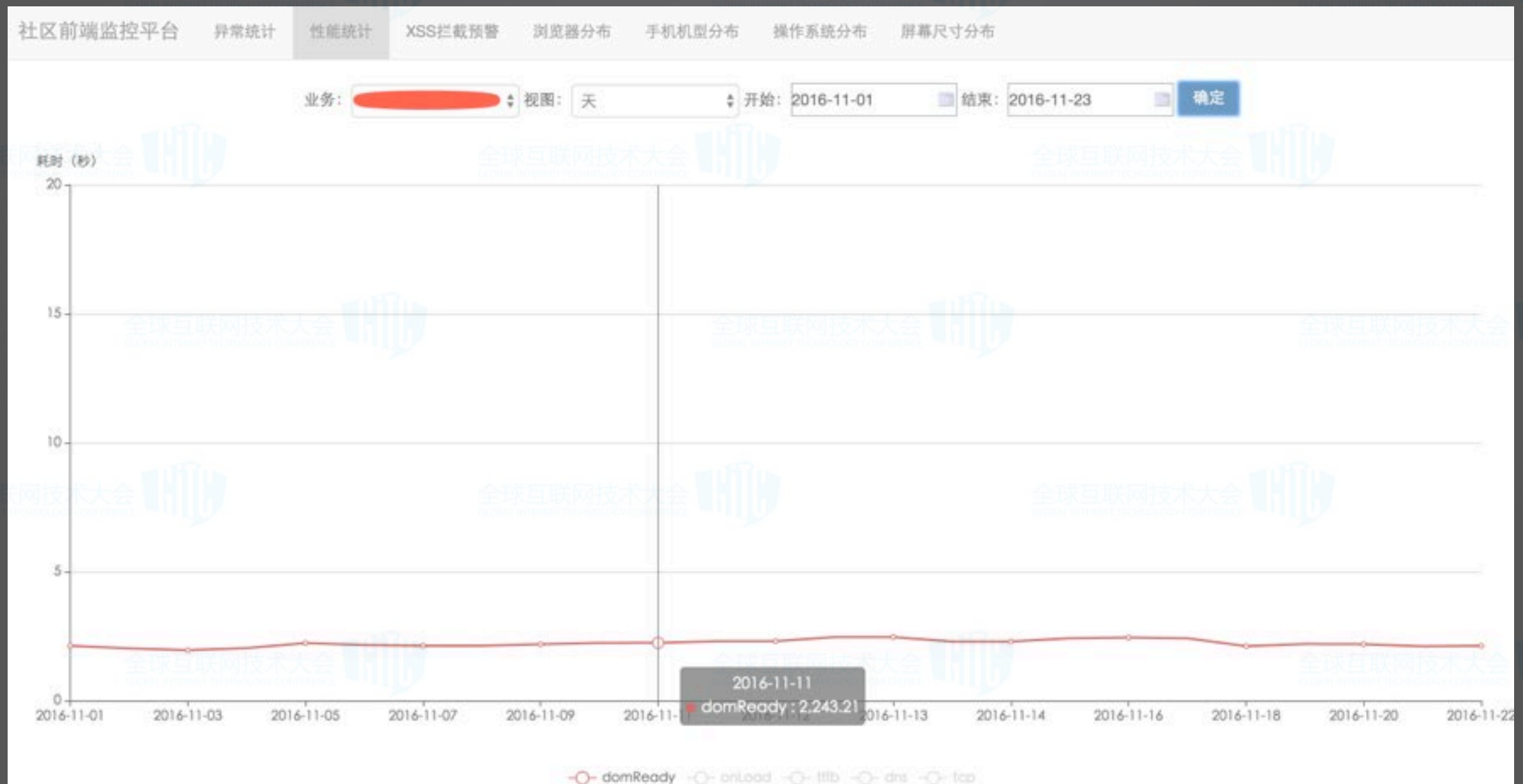


升级成本

一周完成wenke-dev开发

3天完成全部代码的改造

前端监控平台



待优化

迫切需要更犀利的按需加载

前端模板开发调试效率低

胶水代码多

需要更强大的构建和扩展

第二次工业革命

webpack + Gulp

React + Vue.js + ES7/8

webpack

强大的Loader生态

Code Splitting

图片动态地址

热加载

Code Splitting

The screenshot illustrates code splitting in a web application. It shows three source files:

- index/main.js**:

```
1 require.ensure(['./moduleA', './moduleB'], function(require){  
2   require('./moduleA');  
3 });
```
- moduleA.js**:

```
1 console.log('moduleA say hi!');
```
- moduleB.js**:

```
1 console.log('moduleB say hi!');
```

The browser's developer console shows the output of the code execution:

- The **Console** tab displays the message `moduleA say hi!`, which is highlighted with a red box.
- The **Network** tab shows a request for `5.bundle.js` (200 bytes, script type), with the corresponding HTML snippet:

```
<script type="text/javascript" charset="utf-8" async src=  
"./5.bundle.js"></script>
```


Code Splitting

The image illustrates code splitting in a web application. It shows three source files:

- index/main.js**:

```
1 require(['./moduleA', './moduleB'], function(moduleA, moduleB){  
2  
3 });
```
- moduleA.js**:

```
1 console.log('moduleA say hi!');
```
- moduleB.js**:

```
1 console.log('moduleB say hi!');
```

The browser console shows the output of these modules:

```
moduleA say hi!  
moduleB say hi!
```

The generated bundle file, **5.bundle.js**, is shown with a size of 200 bytes and type 'script'. Its content is:

```
<script type="text/javascript" charset="utf-8" async src=  
./5.bundle.js"></script>
```


图片动态地址



```
<i className="ct">  
  <img data-src={require("../images/task/" + task.id + ".png")} alt=""/>  
</i>
```

```
var map = {  
  "./mrw2.png": 3,  
  "./rw-libao.png": 4,  
  "./rw-s1.png": 5,  
  "./rw-s10.png": 6,  
  "./rw-s11.png": 7,  
  "./rw-s2.png": 8,  
  "./rw-s3.png": 9,  
  "./rw-s4.png": 10,  
  "./rw-s5.png": 11,  
  "./rw-s6.png": 12,  
  "./rw-s7.png": 13,  
  "./rw-s8.png": 14,  
  "./rw-s9.png": 15  
};
```

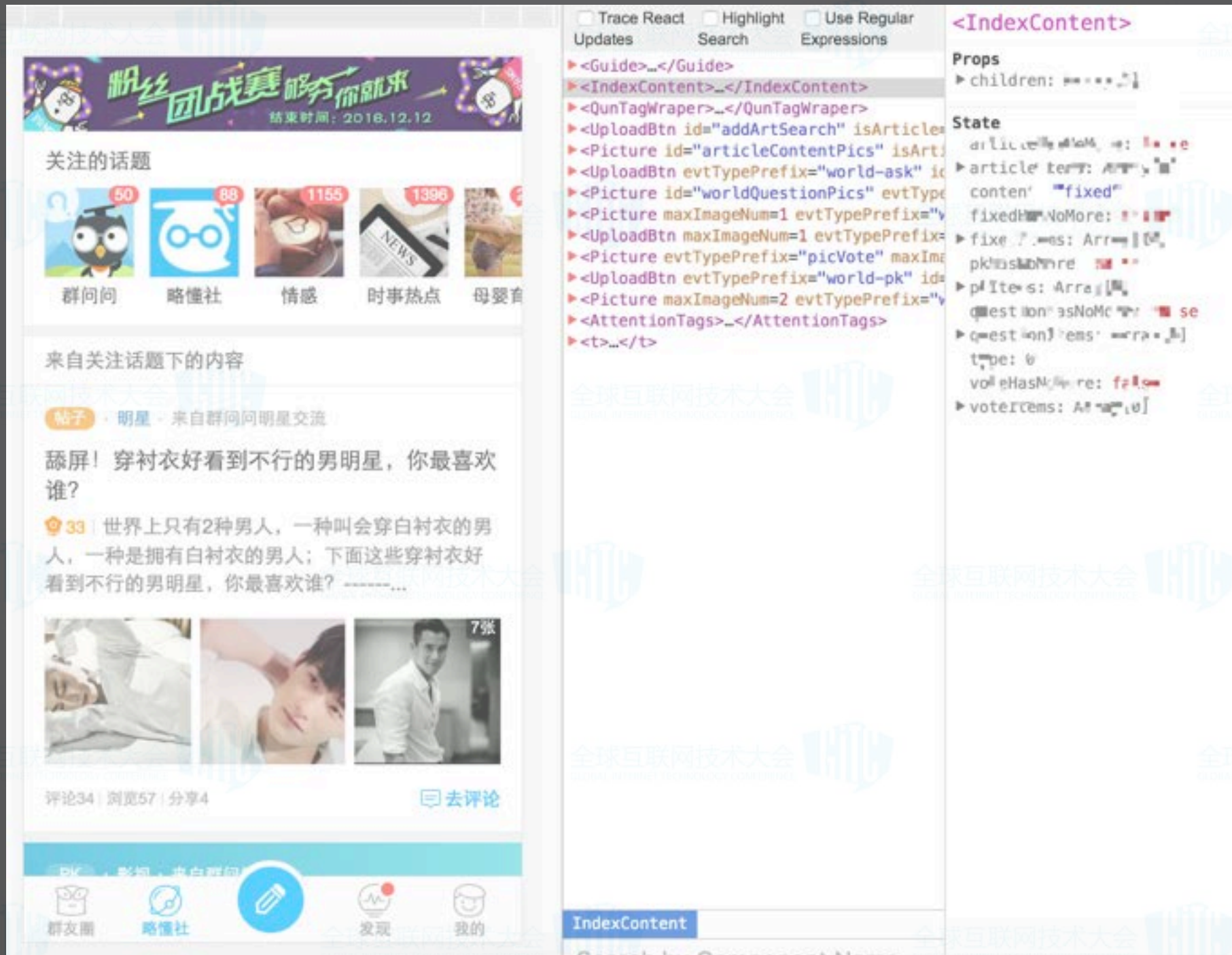
开发框架选择

React
+
Vue.js

react-router
vue-router
react-redux
vuex

别忘了兼容IE8

React调试插件



The image shows a mobile application interface on the left and the React DevTools component inspector on the right. The app interface displays a topic page for '群问问' (Group Questions) with a post about male celebrities. The component inspector shows the component tree and the props/state for the selected `<IndexContent>` component.

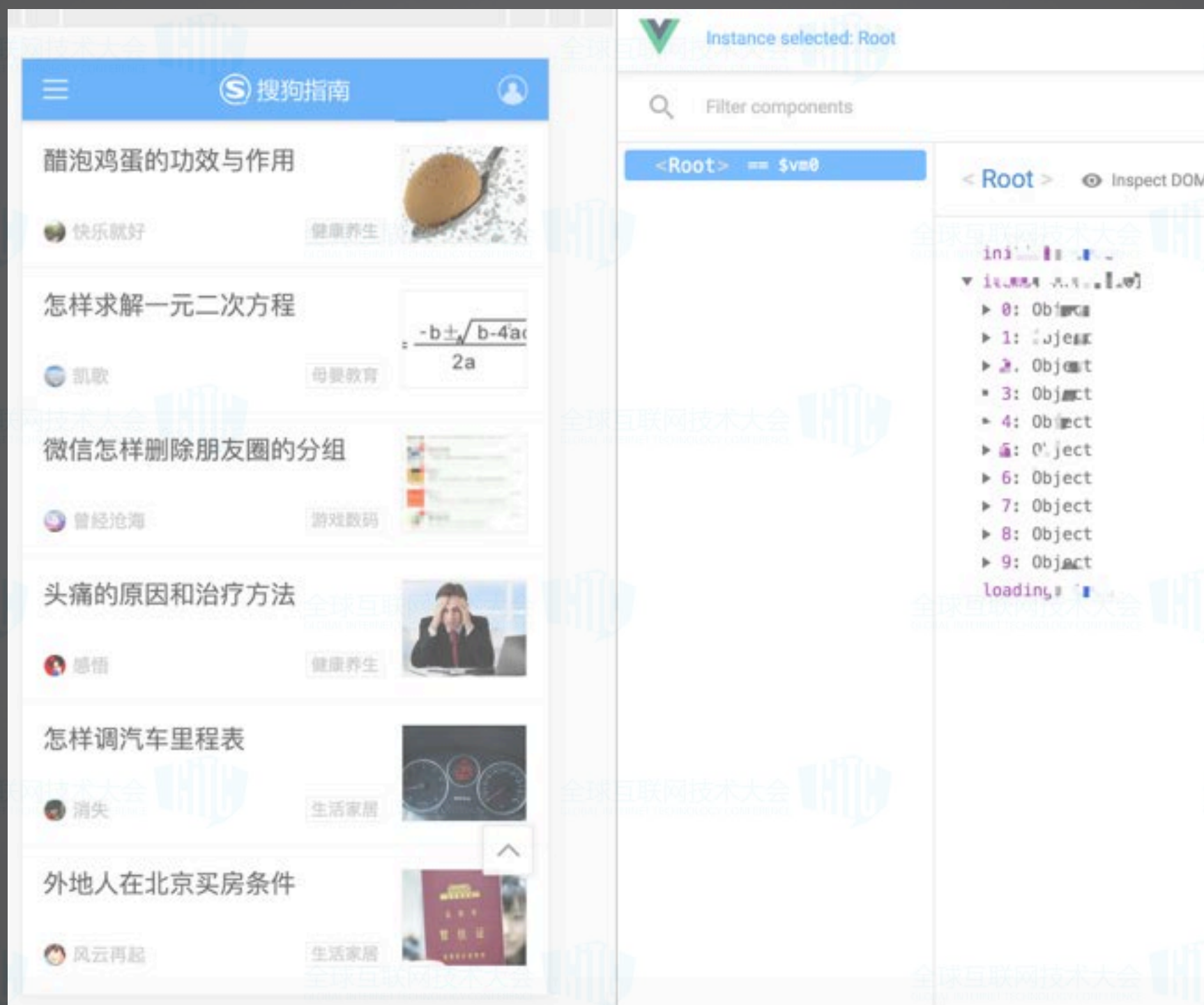
App Interface (Left):

- Header: 粉丝团战赛 够劲你就来 (End time: 2018.12.12)
- Section: 关注的话题 (Topics I Follow)
- Grid of topics: 群问问 (50), 略懂社 (88), 情感 (1155), 时事热点 (1396), 母婴育 (1396)
- Section: 来自关注话题下的内容 (Content from followed topics)
- Post: 舔屏! 穿衬衣好看到不行的男明星, 你最喜欢谁? (33 likes)
- Text: 世界上只有2种男人, 一种叫会穿白衬衣的男人, 一种是拥有白衬衣的男人; 下面这些穿衬衣好看看到不行的男明星, 你最喜欢谁? -----...
- Images: 3 images of male celebrities (7 total)
- Footer: 评论34 | 浏览57 | 分享4 | 去评论
- Bottom Bar: 群友圈, 略懂社, 发现, 我的

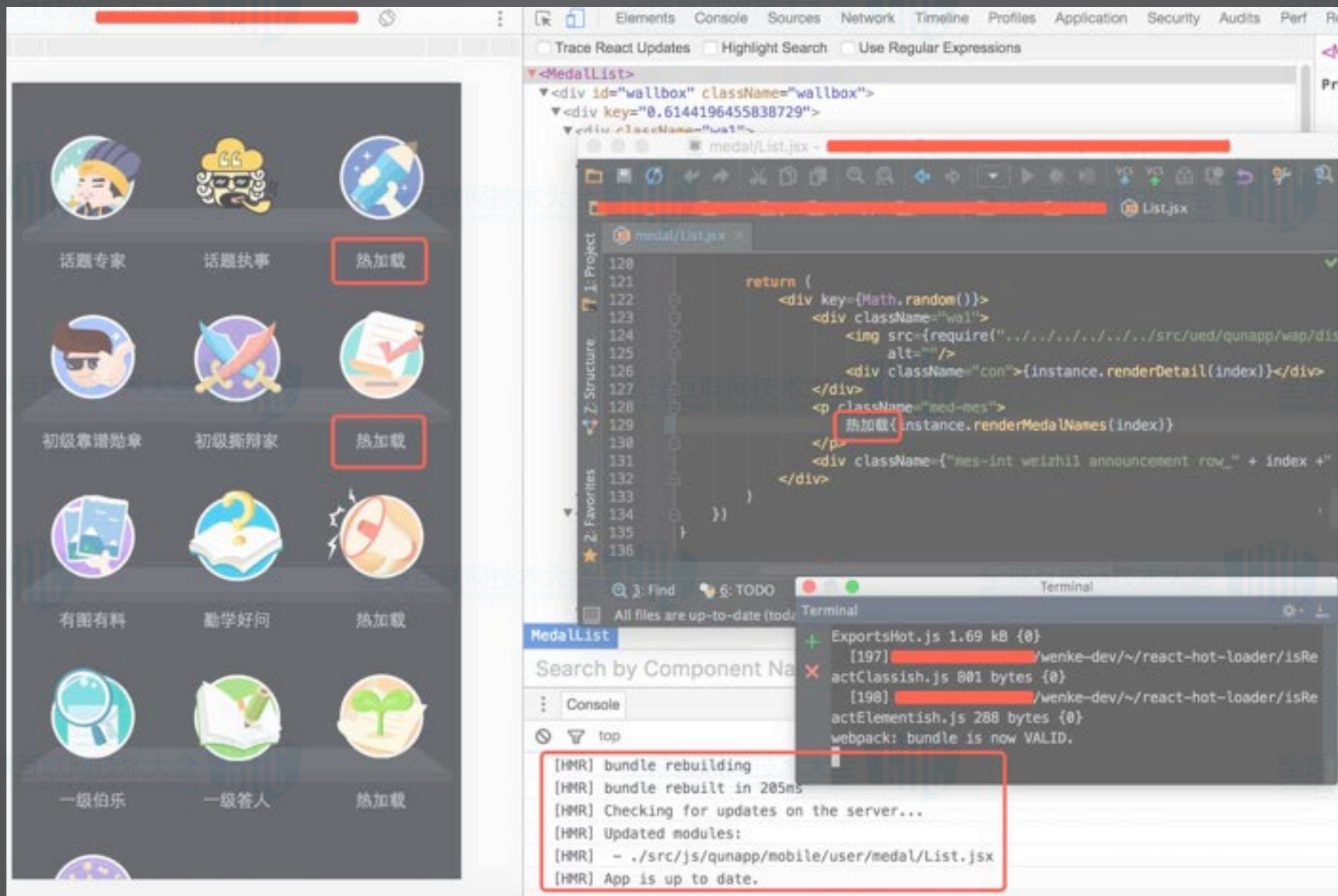
React DevTools (Right):

- Trace React Updates: ☐ Highlight Search: ☐ Use Regular Expressions: ☐
- Component Tree:
 - `<Guide>_</Guide>`
 - `<IndexContent>_</IndexContent>` (Selected)
 - `<QunTagWrapper>_</QunTagWrapper>`
 - `<UploadBtn id="addArtSearch" isArticle>`
 - `<Picture id="articleContentPics" isArticle>`
 - `<UploadBtn evtTypePrefix="world-ask" id>`
 - `<Picture id="worldQuestionPics" evtTypePrefix>`
 - `<Picture maxImageNum=1 evtTypePrefix>`
 - `<UploadBtn maxImageNum=1 evtTypePrefix>`
 - `<Picture evtTypePrefix="picVote" maxImageNum>`
 - `<UploadBtn evtTypePrefix="world-pk" id>`
 - `<Picture maxImageNum=2 evtTypePrefix>`
 - `<AttentionTags>_</AttentionTags>`
 - `<t>_</t>`
- Props:
 - `children: ...`
- State:
 - `articleContentPics: ...`
 - `articleText: ...`
 - `content: "fixed"`
 - `fixedNoMore: ...`
 - `fixedTimes: Array[0]`
 - `pkHasMore: ...`
 - `pictures: Array[0]`
 - `questionAsNoMore: ...`
 - `questionItems: ...`
 - `type: 0`
 - `voteHasMore: false`
 - `voteItems: Array[0]`

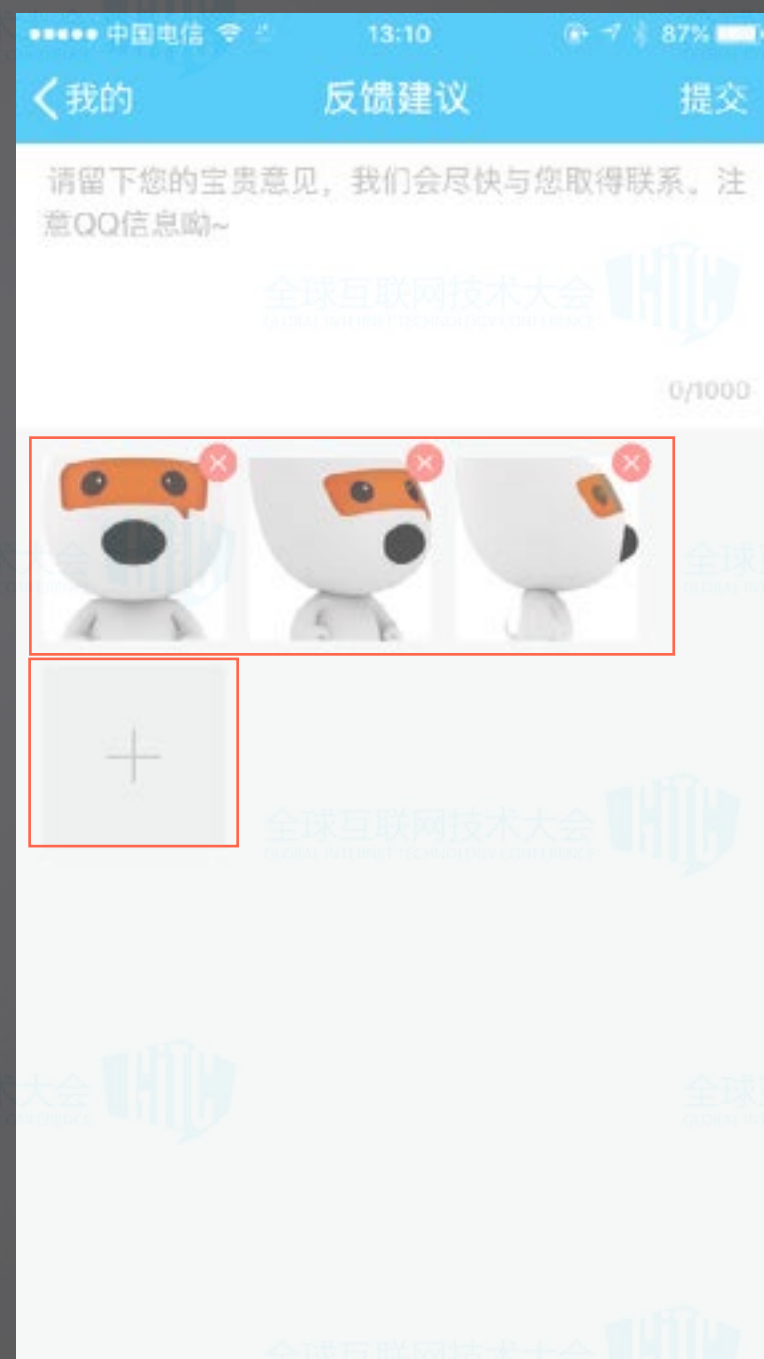
Vue.js调试插件



热加载



组件化开发



静态代码质量检查工具

ESLint

eslint-plugin-react

eslint-plugin-vue

一点体会

视图数据绑定，胶水代码减少

JSX + DSL = 模板开发调试高效

事件绑定更清晰

组件化开发

升级wenke-dev

webpack + Gulp

React + Vue.js

ES6/7/8

react-hot-loader

Gulp.watch + socket.io = LiveReload

升级wenke

v3.0

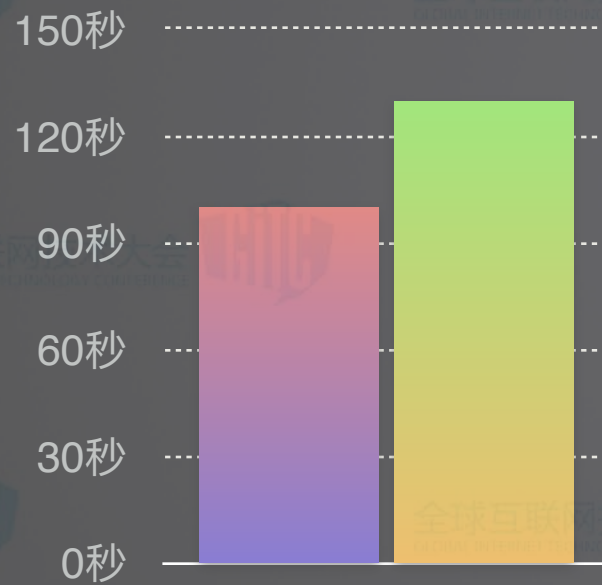
webpack
+
Gulp

React
+
Vue.js

ES6/7/8

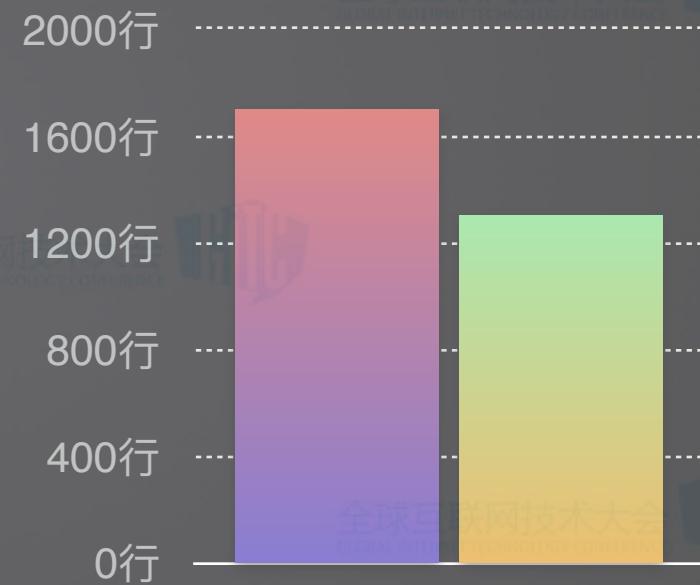
重构后

构建时间峰值



优化前 优化后

构建工具代码量



优化前 优化后

总结

React、Vue.js、ES6/7/8、模块化、
组件化、基础库

实时开发辅助工具：wenke-dev

一键发布系统（集成wenke）

前端监控（性能、异常等）

一点经验

- ① 尽可能建设比较完善的前端基础设施
- ② 善用提升效率的工具和方法
- ③ 与时俱进，稳步推进团队技术升级



欢迎大家加入搜狗

THANKS

Q&A

段振兴