

ThoughtWorks®

JimmyLv's

JSCONF 2016

*What's better Developer Experiences &
How to build FullStack Application by feature?*

FORMAL SESSIONS

单页应用“联邦制”实践 —— 孙坤鹏 (UCloud)

Building a Unified Frontend and Mobile Team —— 郭达峰 (Strikingly)

Learning design patterns from modern JavaScript frameworks —— Fraser Xu (Envato)

移动海量服务下基于React的高性能同构实践 —— 梁伟盛 (Tencent Now 直播)

Using nodejs to count 30 billion requests per day —— Guillaume (Goyoo Networks)

DevTools for the Progressive Web —— Kenneth Auchenberg (Microsoft)

单页应用“联邦制”实践

基于Angular的SPA应用发布优化，单页应用不同产品线希望以不同节奏支持灰度发布。问题的新的解决方案 —— 联邦制

按模块发布并做记录，运行时每个用户从服务端读取特定的模块版本信息来加载。

产品拆分：单页面、多应用 模块单独治理 分模块加载机制，按需加载、资源优化

可能复杂的主要在于整个流程的工程化和自动化，多个产品线公共的模块如果需要升级似乎还是避不开提到的痛点的。

（对比阿里的“诸侯制”） => 升华，不管如何， Feature 化和 组件化 都是为了分治，FED MircoService

BUILDING A UNIFIED FRONT-END AND MOBILE TEAM

由技术变革引发的团队架构变革 | 产品最难的就是沟通，软件工程解决的是人的问题

React Native / Weex => 前端/全端团队结构、合作方式

跨端的工程化，代码复用率 90%

跨端组件差异 => 原生模块的补充（使用 TypeScript 定义组件接口，分别实现 => 语言所带来的更高层次的抽象）

打造全端团队：具备native开发经验，iOS 学 RN

历史包袱，现有的组织结构与人员分配，（实践检验真理，五年后）

LEARNING DESIGN PATTERNS FROM MODERN JAVASCRIPT FRAMEWORKS

来源于建筑设计模式 => 现代流行框架中的设计模式

FP(Pure Function, HOF)/Declarative and functional view render/
unidirectional data flow/immutable data structure/type system

view: Model -> HTML

Elm、yo-yo、MobX

设计模式不关心实现细节，有几种模式不重要，六大原则才是代码设计的关键，充分考虑业务场景的需要。

移动海量服务下基于REACT的高性能同构实践

腾讯的极致用户体验 | 腾讯直播APP的直播页面秒开优化

React 所支持的服务器端同构渲染

以RTT为切入点，减少RTT请求（服务端渲染）

页面首屏直出，其他使用异步加载（先显示框架）

使用UDP替换 TCP，节省字节，加快数据的返回

不常变更的数据cache（服务端 + localStorage）



USING NODEJS TO COUNT 30 BILLION REQUESTS PER DAY

法国人，中文，打广告

数据分析

质量保证

异常发现与报警

应用异常监控、应用性能分析、用户数据采集分析等等

DEVTOOLS FOR THE PROGRESSIVE WEB

Web 开发模式的变迁 => 在 Web 逐渐渗透 Native 的背景下重新反思前端

前端调试工作流的重复、不流畅 => 浏览器 DevTools 和编辑器

VS Code 直接调试 Chrome、Edge、Node、甚至 iOS 移动端

编辑器和 DevTools 紧密结合在一起 => IntelliJ? VS Code 是穷人版的

而浏览器和DevTools则会分开，普通用户友好（Edge）

LIGHTNING TALK

- bamei

一个帮助快速搭建 Node.js 项目的工具，对模块，目录，中间件，库，配置文件，程序初始化的统一管理，包括一个bamei的生成器，我理解是一种拼装项目的技术概念，名字叫bamei...

- npminstall

<https://github.com/cnpm/npminstall>

安装模块优化，拿自己流量装，一个cnode论坛不到一分钟，速度提升6-60倍。

纯属炫技，装完逼就跑。🏃

UNIVERSAL UI COMPONENTS

Universal Component

推荐的一个是 Universal UI Components，另一个是由此引出的为什么要这么来写 components（只用这些 view 渲染库的子集），是为了让自己不要被绑定在一个框架下。

```
const Button = (props) => (
```

```
  <button {...props} />
```

```
)
```

不想每隔半年就用最新潮框架重写一次我的前端代码，也并不想被绑定在一个框架上之后就面对更新的思想无法应用上。

做 jsx 的意义也就是希望像 element ui 之类的这些基础组建能够不必站队，写一次，前后做点工作，就能一套代码跑在两个框架下该多好。

开发框架的相互促进、借鉴、合作和相互弥补从而共建技术生态社区也是一个可能的趋势。

PPT 下载

Building a Unified Frontend and Mobile Team.pdf

DevTools for the Progressive Web.pdf

Learning design patterns from modern JavaScript frameworks.pdf

Using nodejs to count 30 billion requests per day.pdf

单页应用“联邦制”实践.pdf

移动海量服务下基于React的高性能同构实践.pdf



Hypermedia As The Engine Of Application State!



THANK YOU

Any Question or Feedback?

Jimmy Lv | 吕靖
jinglv@thoughtworks.com
<http://blog.jimmylv.info>

ThoughtWorks®