

LiveVideoStackCon

# 高品质互动在线课堂开发实践

和君 Daniel He

2017年10月20日

## 和君 (Daniel He)

- 10余年前后端研发及架构经验
- 擅长大型前端项目架构，前端技术体系搭建
- 曾就职于沪江网、途牛网等互联网公司
- 现任tutorabc前端负责人，致力于打造互联网教育领域的高品质在线课堂解决方案。



*tutorabc*

*vipjr*

01

**选型** 为什么选择WebRTC

02

**优化** 极致用户体验

03

**互动性** 丰富线课堂互动性

04

**持续交付** 快速发布部署

05

**前端APM** 提供稳定保证



TutorMeet +

The screenshot displays a TutorMe online classroom interface. At the top, the 'vipabc' logo is visible on the left, and the session title 'Session 935' is on the right. A central banner reads 'Vocabulary' with a pizza image. Below this, a red-bordered window titled 'vipabc' shows a lesson on 'Planet'. The main content area features the word 'Planet' in large orange letters, surrounded by colorful illustrations of a planet, a sun, a rocket, and a moon. A toolbar on the left contains drawing tools and text options. On the right, a list of '在线学员' (Online Students) includes names like Susan Mi., Sandy, Luna, Larry, Jessie, Vic, and Mia, each with a volume control slider. A chat window on the left shows messages from Josh, Jessie, Susan, and Sandy. At the bottom, a row of student avatars includes Josephine, Luna, and Paul. The interface is clean and user-friendly, designed for interactive learning.

# Why WebRTC ?

- 1 内置于浏览器，无需下载，无需装插件
- 2 开发成本低，简单的 JS-API 即可实现音视频通讯
- 3 开源、安全
- 4 浏览器支持越来越好
- 5 Flash 将于 2020 彻底退役

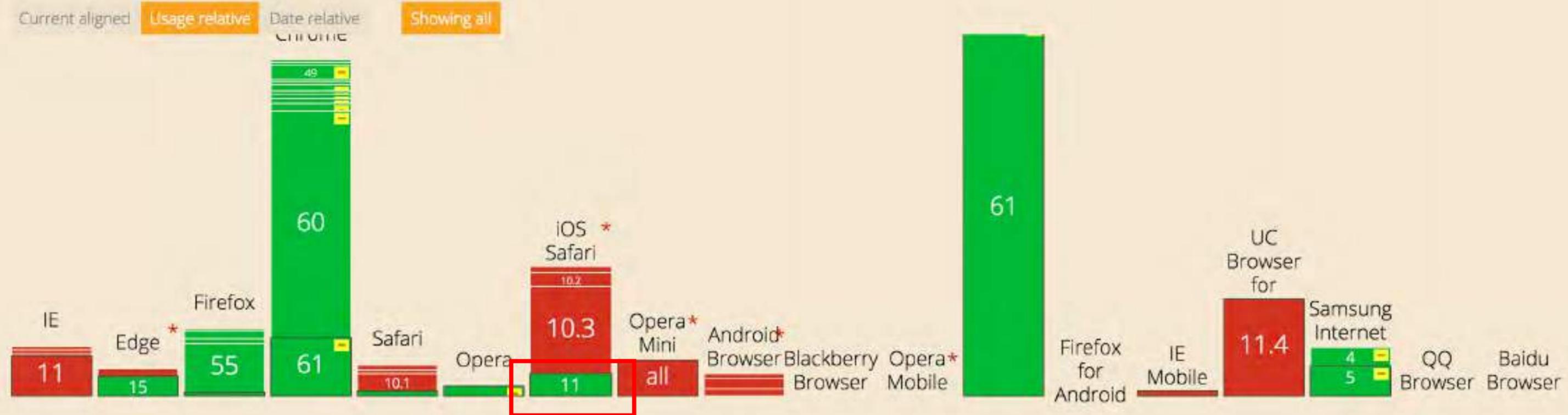


## By Sept 2017

### WebRTC Peer-to-peer connections - WD

Global 70.68%  
unprefixed: 9.04%

Method of allowing two users to communicate directly, browser to browser using the RTCPeerConnection API.



Safari 在 11版本做了支持

MediaStream

RTCPeerConnection

RTCDataChannel

icecandidate  
AudioContext  
getLocalStreams()  
getRemoteStreams()  
getVideoTracks  
getUserStream()

<chrome://webrtc-internals>

signalingstatechange  
RTCSessionDescription  
MediaDevices  
getAudioTracks

01

Adapter @ WebRTC

<https://github.com/webRTC/adapter>

02

AdapterJS @ Terasys

<https://github.com/Terasys/AdapterJS>

03

SimpleWebRTC @ andyet

<https://github.com/andyet/SimpleWebRTC>

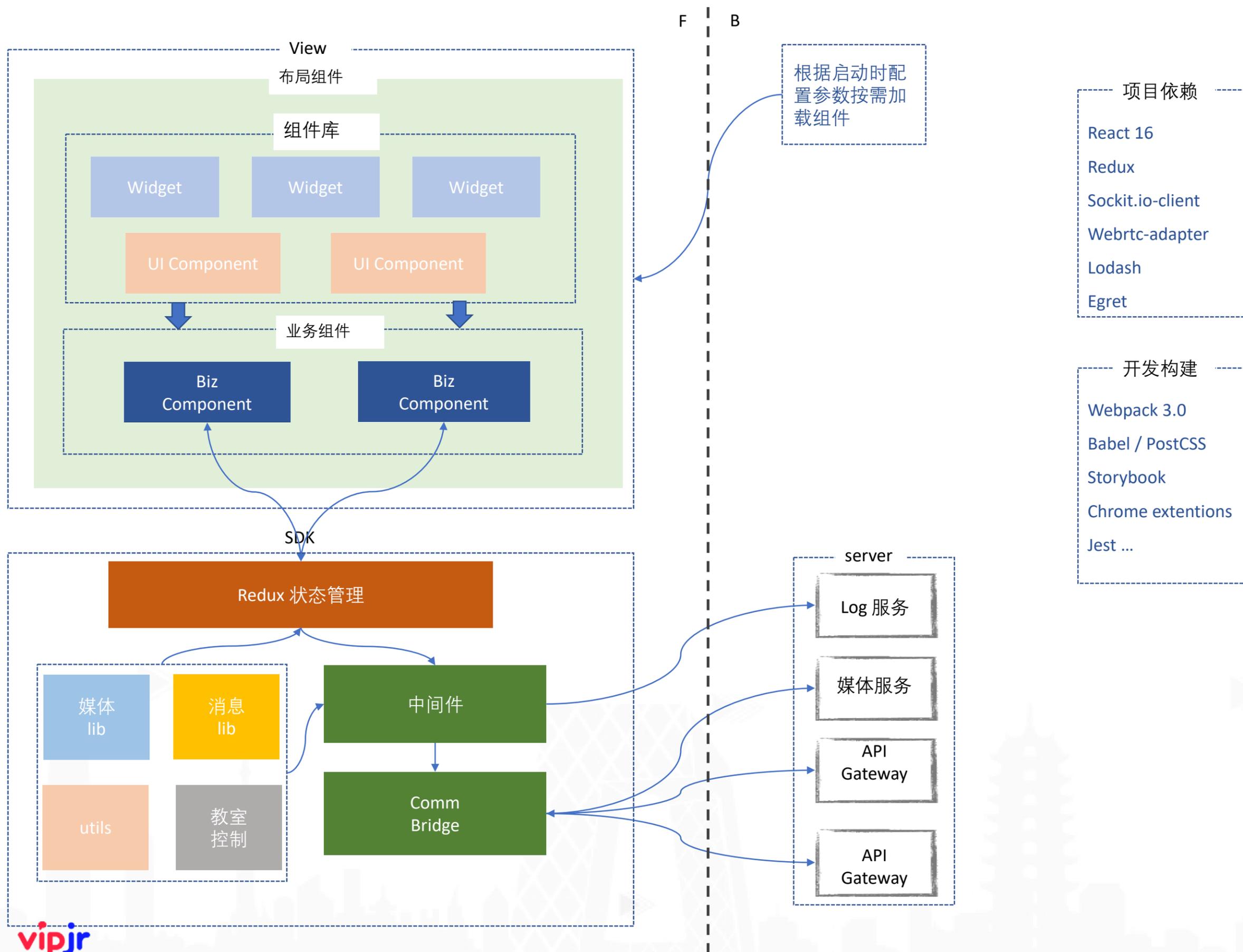


## 公开课，大班课场景

- WebRTC + 推流技术
- 支持10000人同时在线
- 支持RTMP, HLS

## 小班课场景

- WebRTC 会议模式
- 智能服务切换



相较一般SPA交互性更强

用户的页面滞留时间长

需要尽可能的避免页面刷新

功能繁复，导致静态资源体积很大

...

# Optimizations

构建时优化  
Build-time  
Optimization

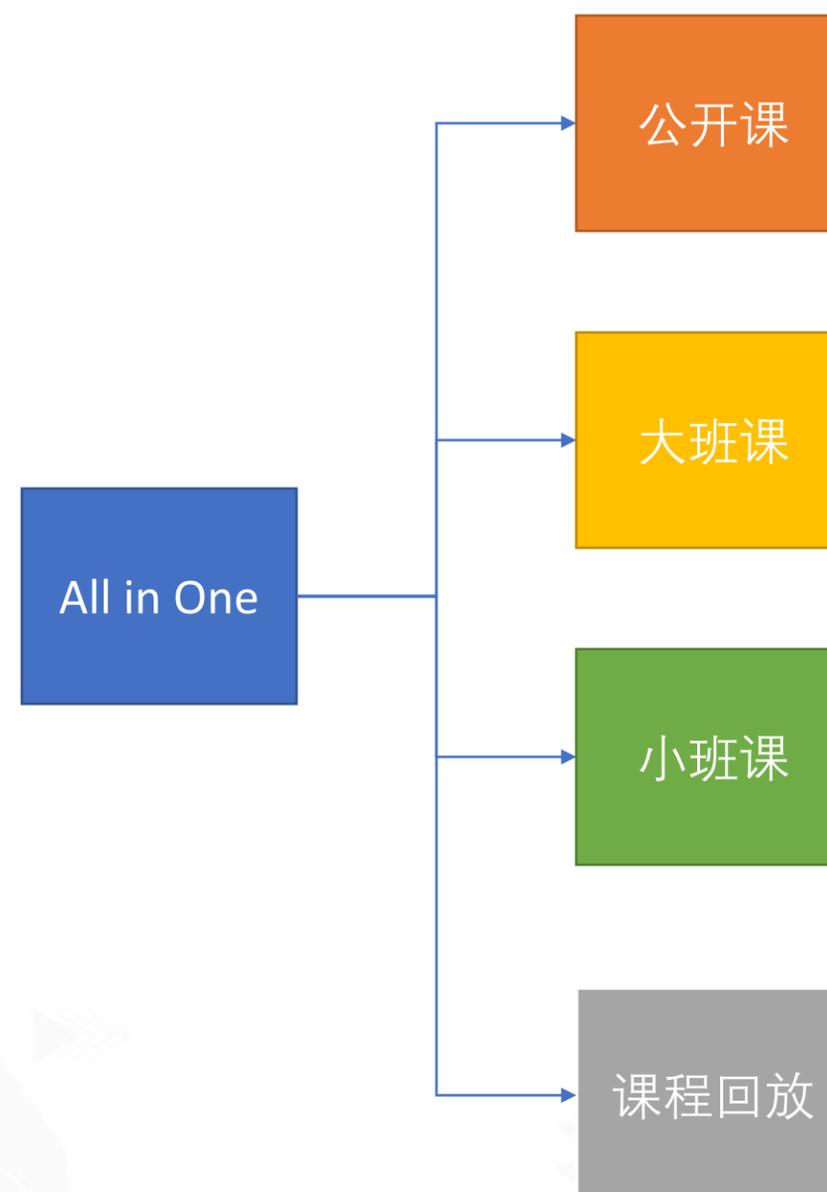
运行时优化  
Run-time  
Optimization

用户体验优化  
UX  
Optimization

# 基于webpack



- 1 按业务逻辑拆分多入口
- 2 Code Splitting
- 3 本地化语言包按需加载



## 利用Webpack 3.0 的 **Tree-shaking / Scope Hoisting** 等特性

```
// foo.js
function foo () {
// ...
}
// bar.js
const bar = 123
// main.js
// foo(bar)
```

Tree-Shaking

```
// nothing left!
```

```
import { foo } from './foo.js'
import { bar } from './bar.js'
foo(bar)
```

```
// foo.js
export function foo () {
// ...
}
```

```
//bar.js
export const bar
```

Scope Hoisting

```
// foo.js
function foo () {
// ...
}
// bar.js
const bar = 123
// main.js
foo(bar)
```

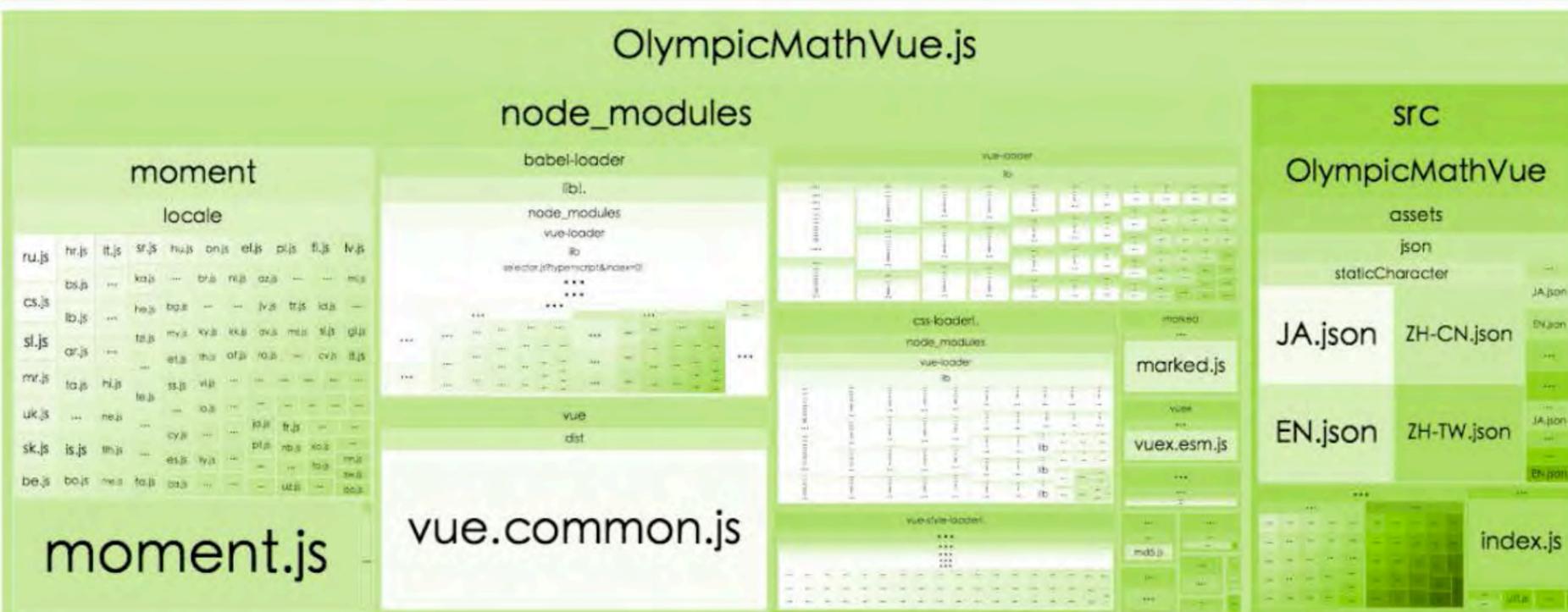
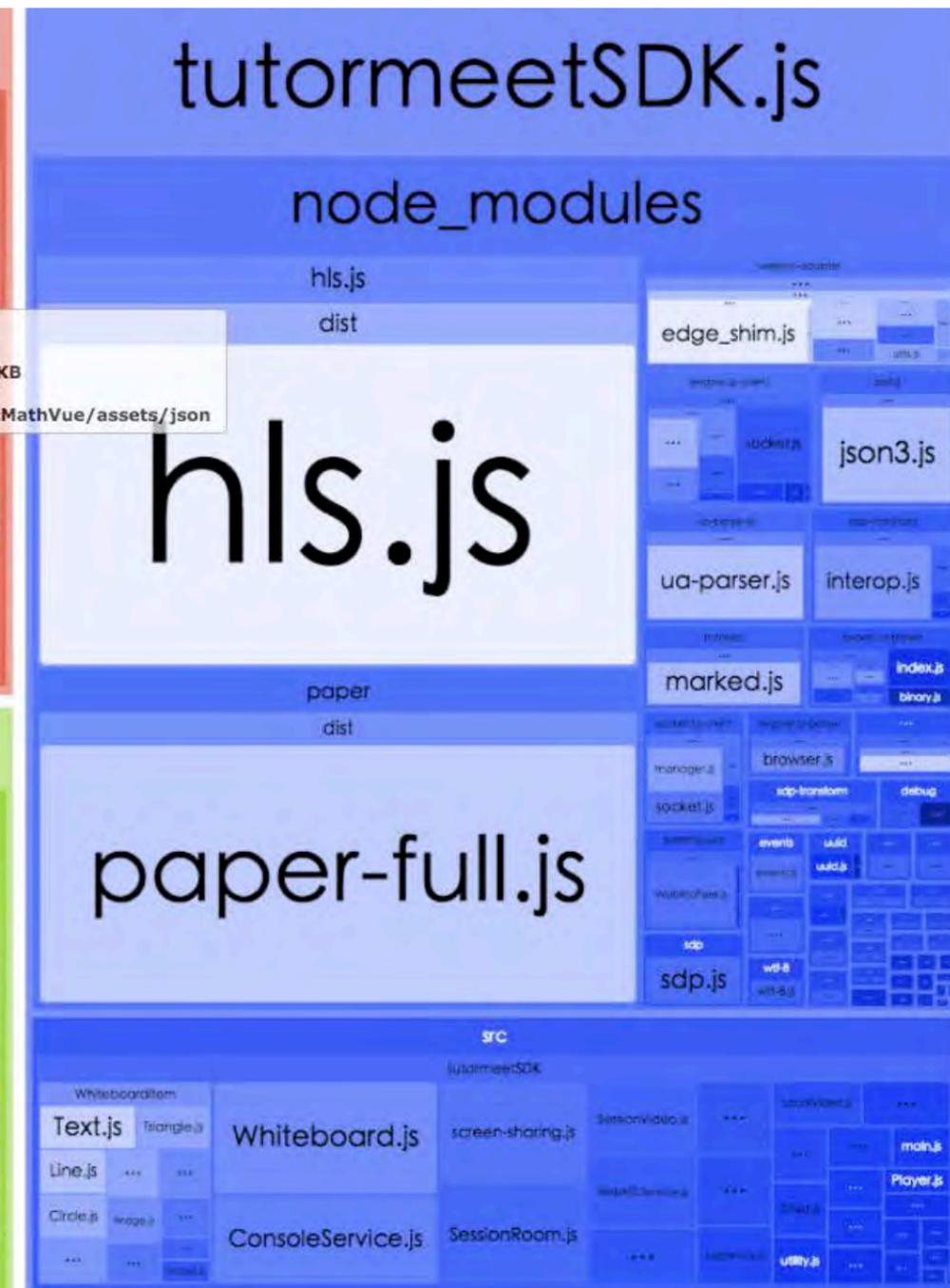
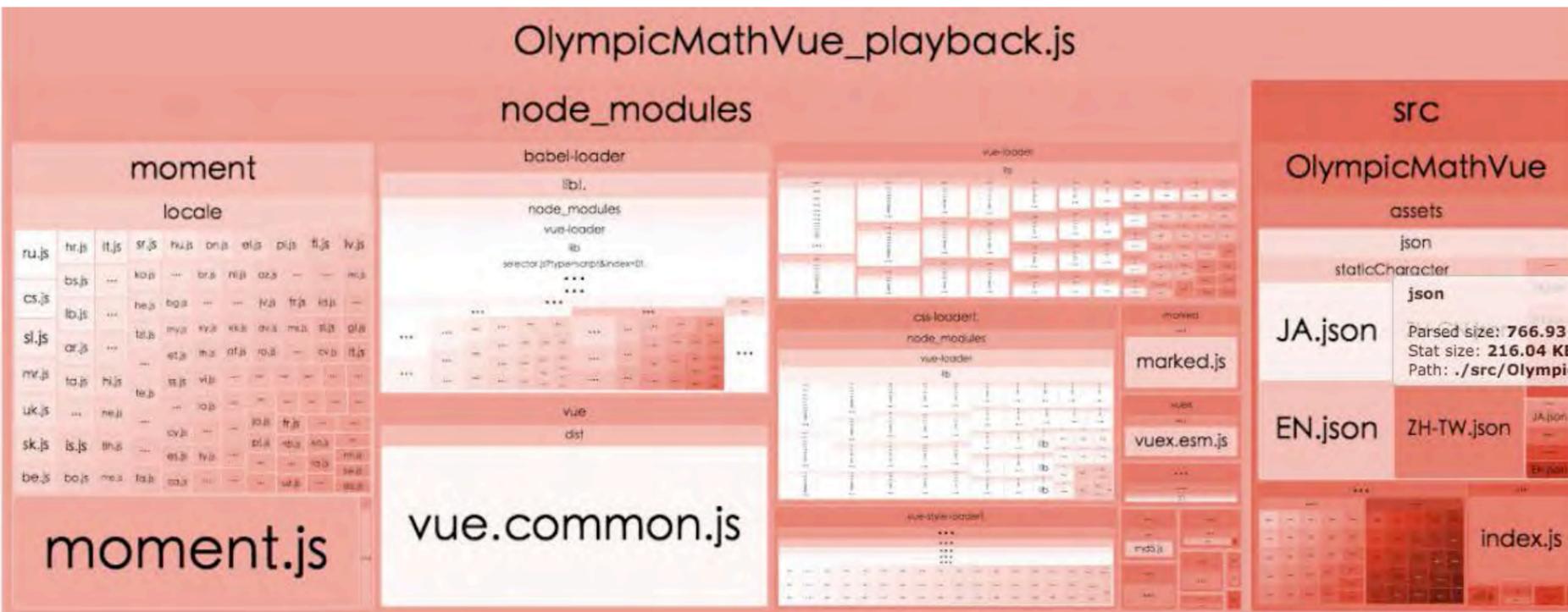
利用Prepack (partial evaluator), **！未用到生产环境**

```
(function () {  
  function fibonacci(x) {  
    return x <= 1 ? x : fibonacci(x - 1) + fibonacci(x - 2);  
  }  
  global.x = fibonacci(10);  
})();
```

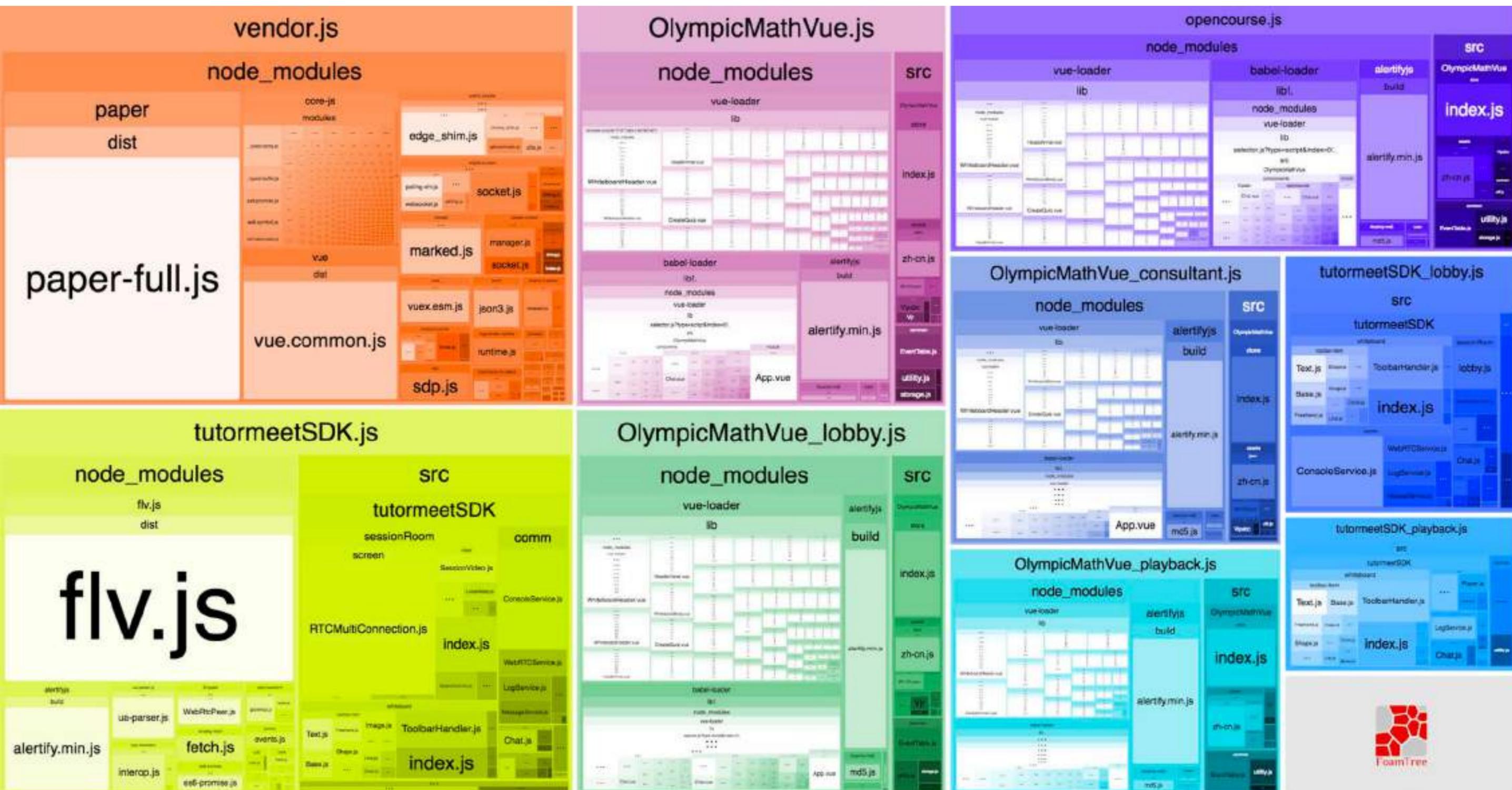
Partial evaluation

```
x = 55;
```

平均每个入口JS总体积大小**1060k** ( gzip )



平均每个入口JS总体积大小**442k** (gzip) , 减少 **60%**

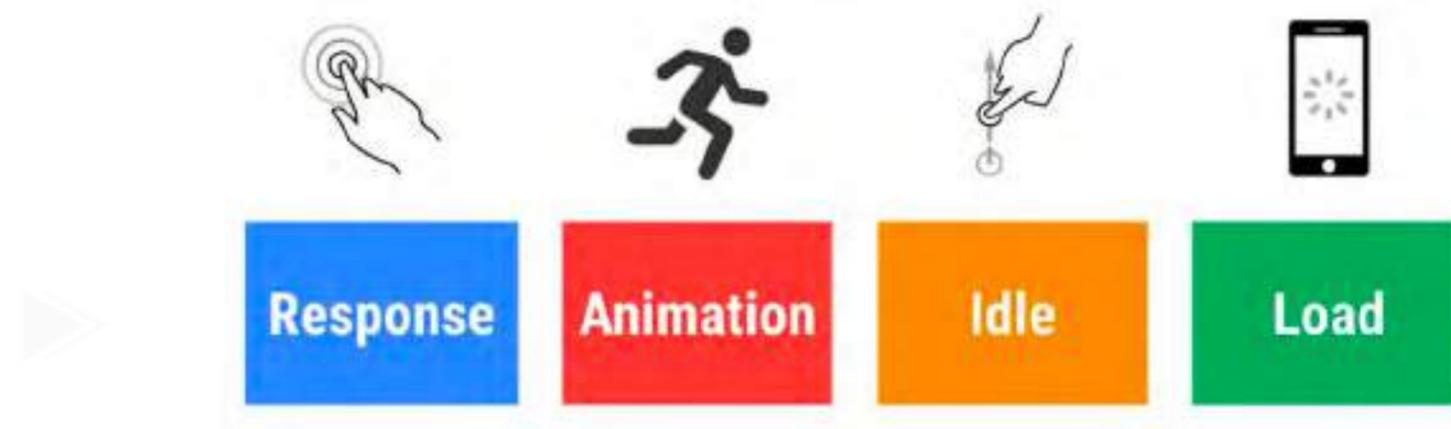


响应 (Response) : 100ms内做出响应

动画 (Animation) : 10ms内产生一帧

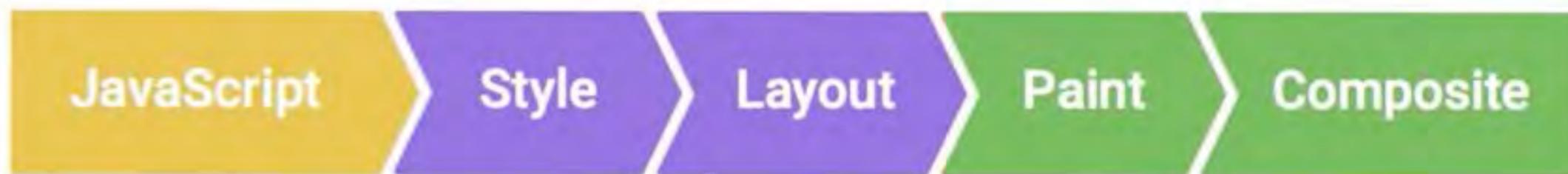
空闲 (Idle) : 最大化空闲时间

加载 (Load) : 1000ms内提供内容

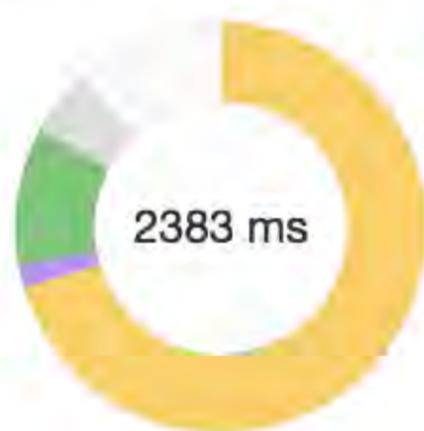


<https://developers.google.com/web/fundamentals/performance/rail>

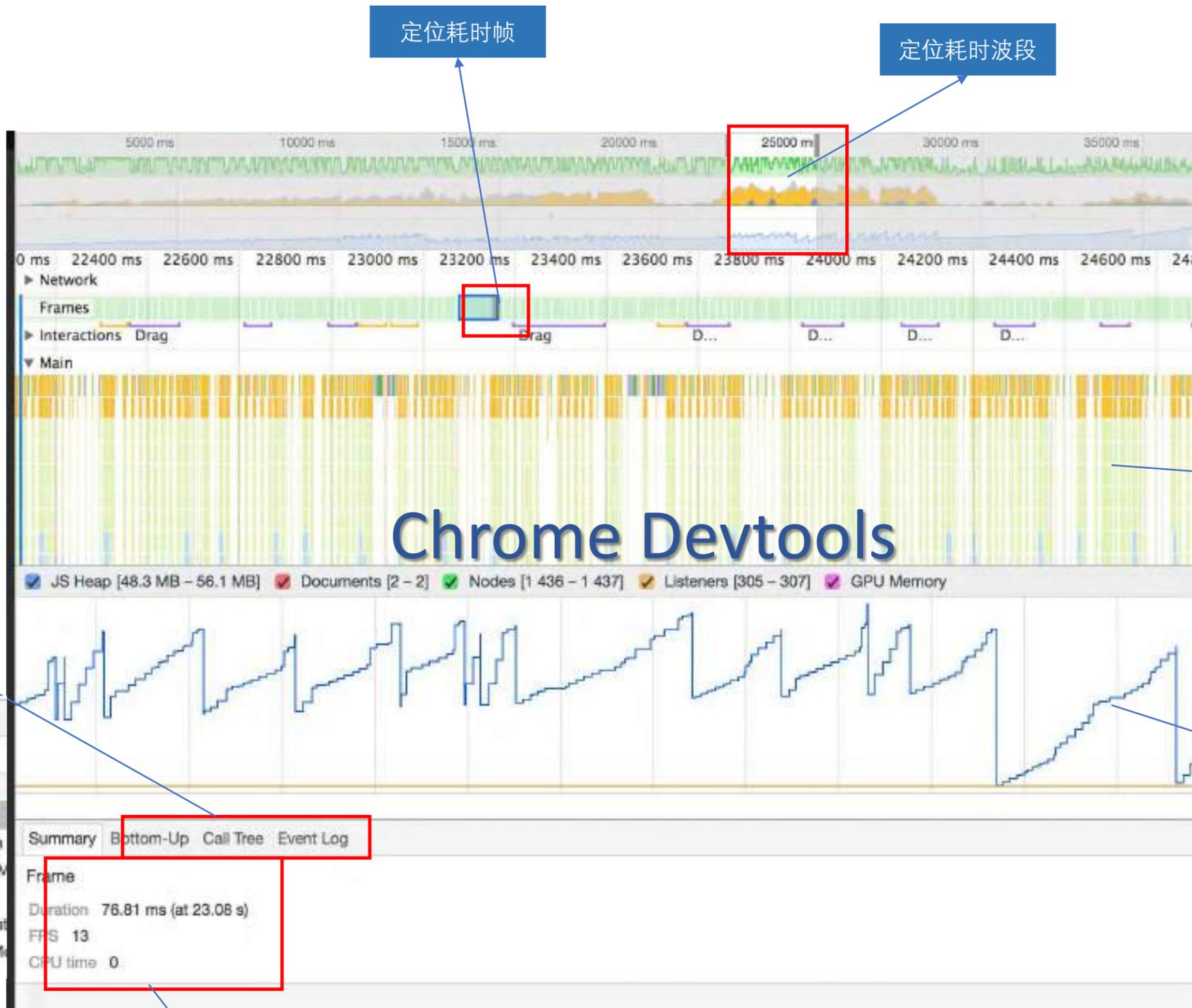
## Pixel Pipeline



Range: 23.80 s – 26.18 s

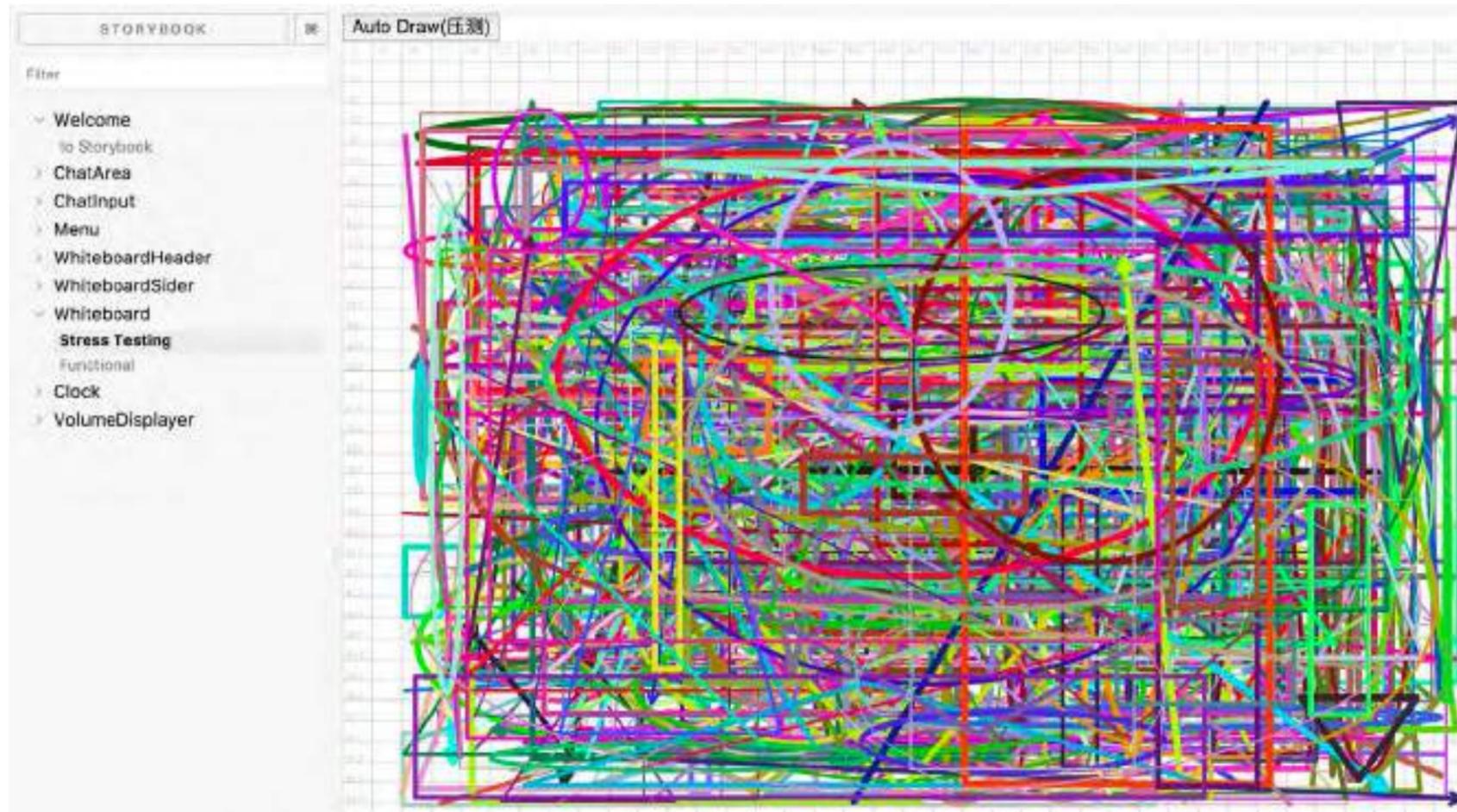


|           |           |   |                   |
|-----------|-----------|---|-------------------|
| 1674.8 ms | Scripting | → | Javascript        |
| 42.1 ms   | Rendering | → | Style / Layout    |
| 258.7 ms  | Painting  | → | Paint / Composite |
| 123.1 ms  | Other     |   |                   |
| 284.5 ms  | Idle      | → | 空闲时间              |



分析三项

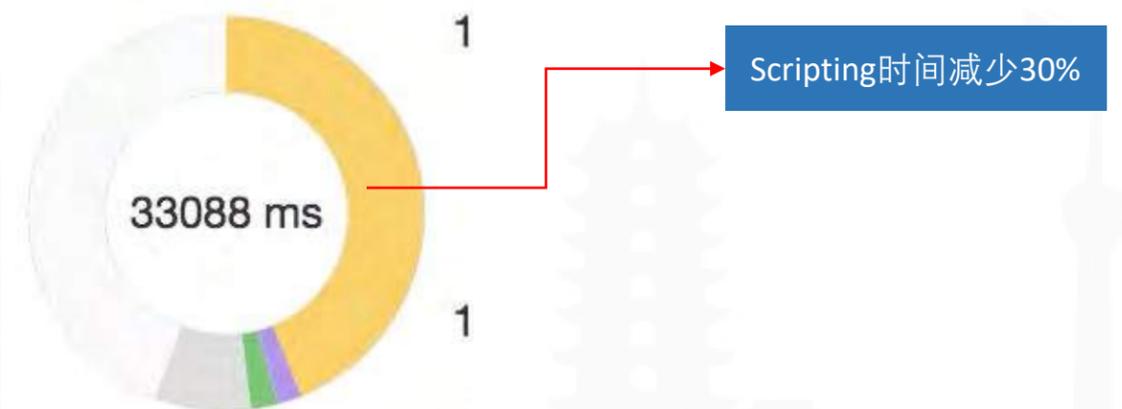
耗时过长!



## 优化点

- 对象复用
- 绘制路径优化
- 节流控制 (throttle)
- 拖动、缩放等交互优化
- ....

优化后



01 预加载 / 懒加载

02 响应式布局

03 渐进式用户体验

04 层级管理 ( z-index )

▶ 05 Web安全色、安全字体

# 03 互动性

授课过程互动性不强，比较枯燥无味

青少儿教学，对内容趣味性的要求

青少儿注意力容易分散

内容不够个性化

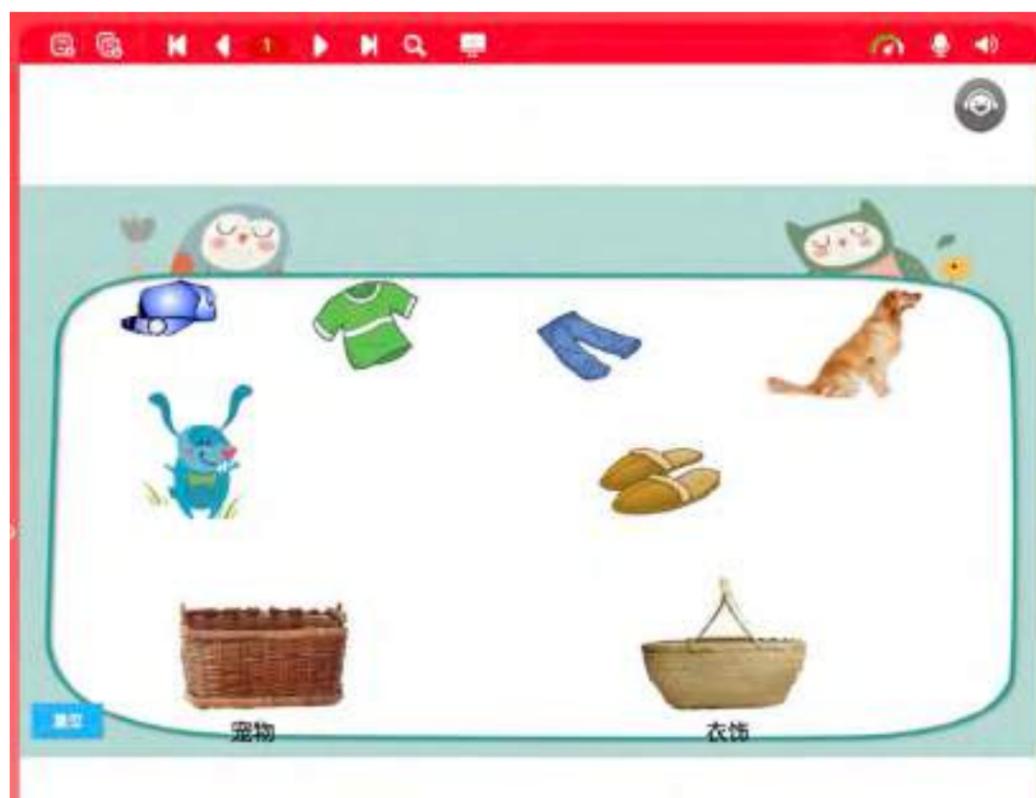
...



# 游戏化白板

老师学生及学生之间的互动

趣味性 + 教育性 相结合





## 屏幕分享

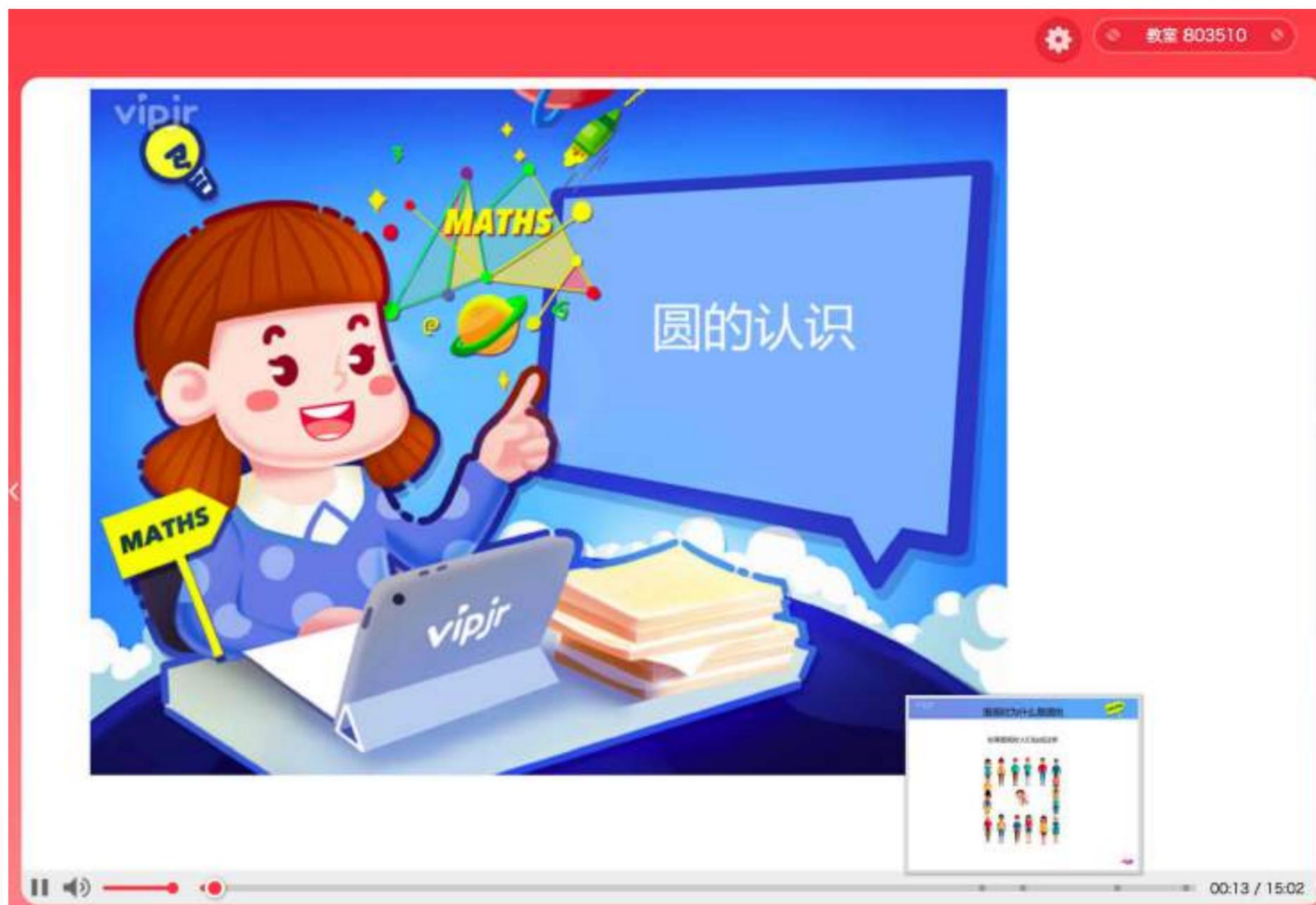
增加课程教育的多样性

展示丰富的教学软件



# 录影档结构化

增加翻页节点，方便课程回顾



# 持续交付 Continuous Delivery

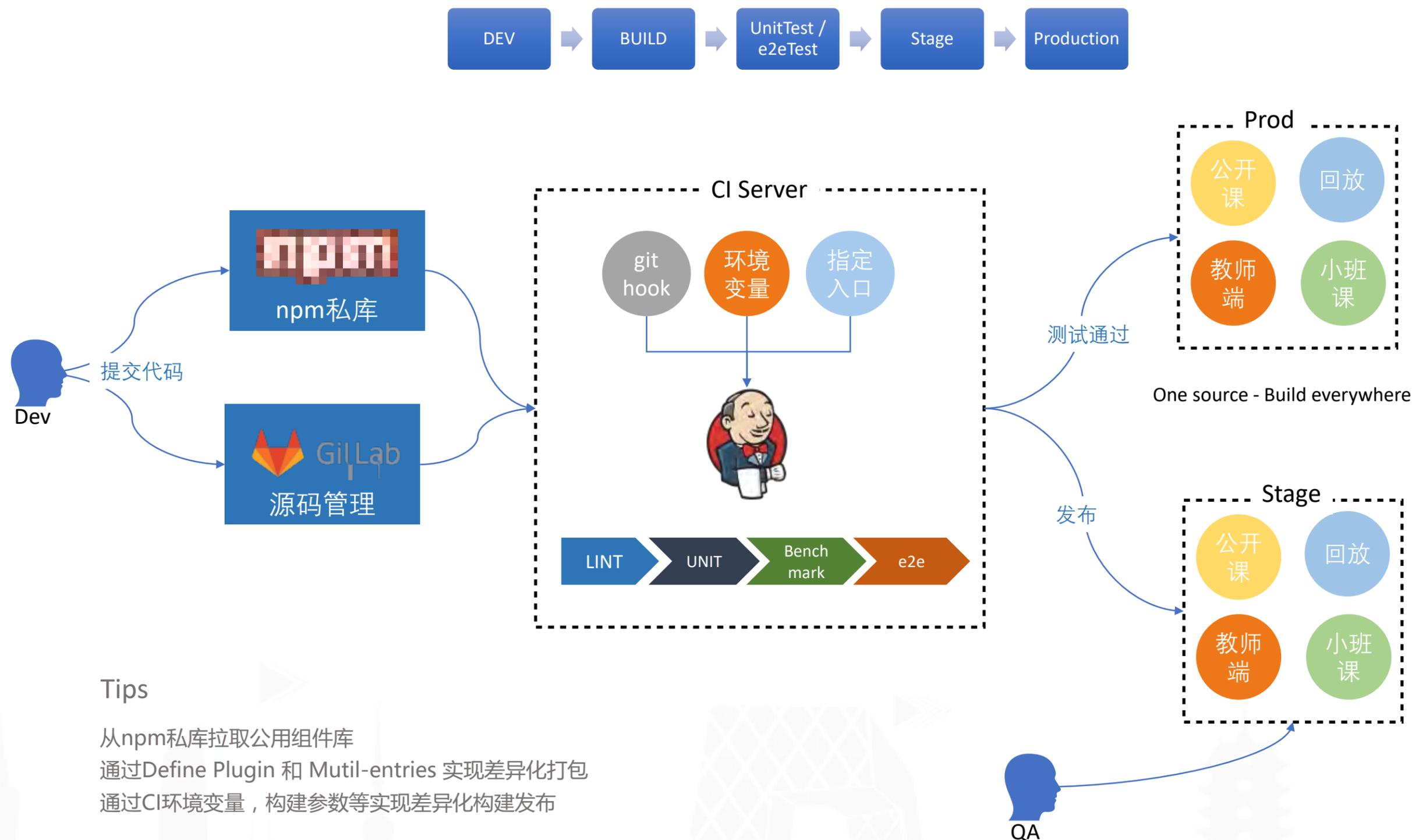
## 目标

- 适应互联网快速迭代的需求
- 自动化、降低人工成本
- 开发、测试、运维更紧密合作
- 保证生产环境质量



## 达成

- ✓ 清晰的分支管理和发布策略
- ✓ 保证核心模块的高覆盖率测试
- ✓ 非覆盖增量 发布到CDN
- ✓ 快速回滚



## Tips

- 从npm私库拉取公用组件库
- 通过Define Plugin 和 Mutil-entries 实现差异化打包
- 通过CI环境变量，构建参数等实现差异化构建发布

01

私有包管理库

02

webpack / node.js / npm script

03

Shell Script

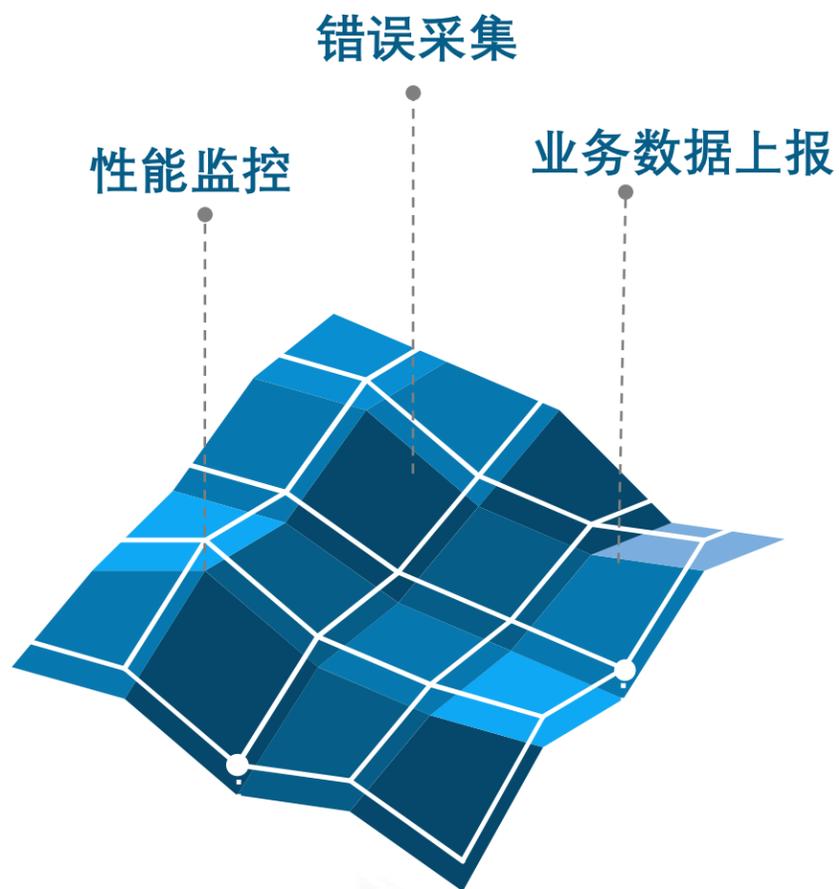
04

Jenkins / git webhook

05

eslint / stylelint / jest / benchmark / Nightwatch

# 05 前端 APM



1

## 性能监控

首屏加载：针对TTFB，Content Download 等关键数据的采集  
可预期的耗时操作

2

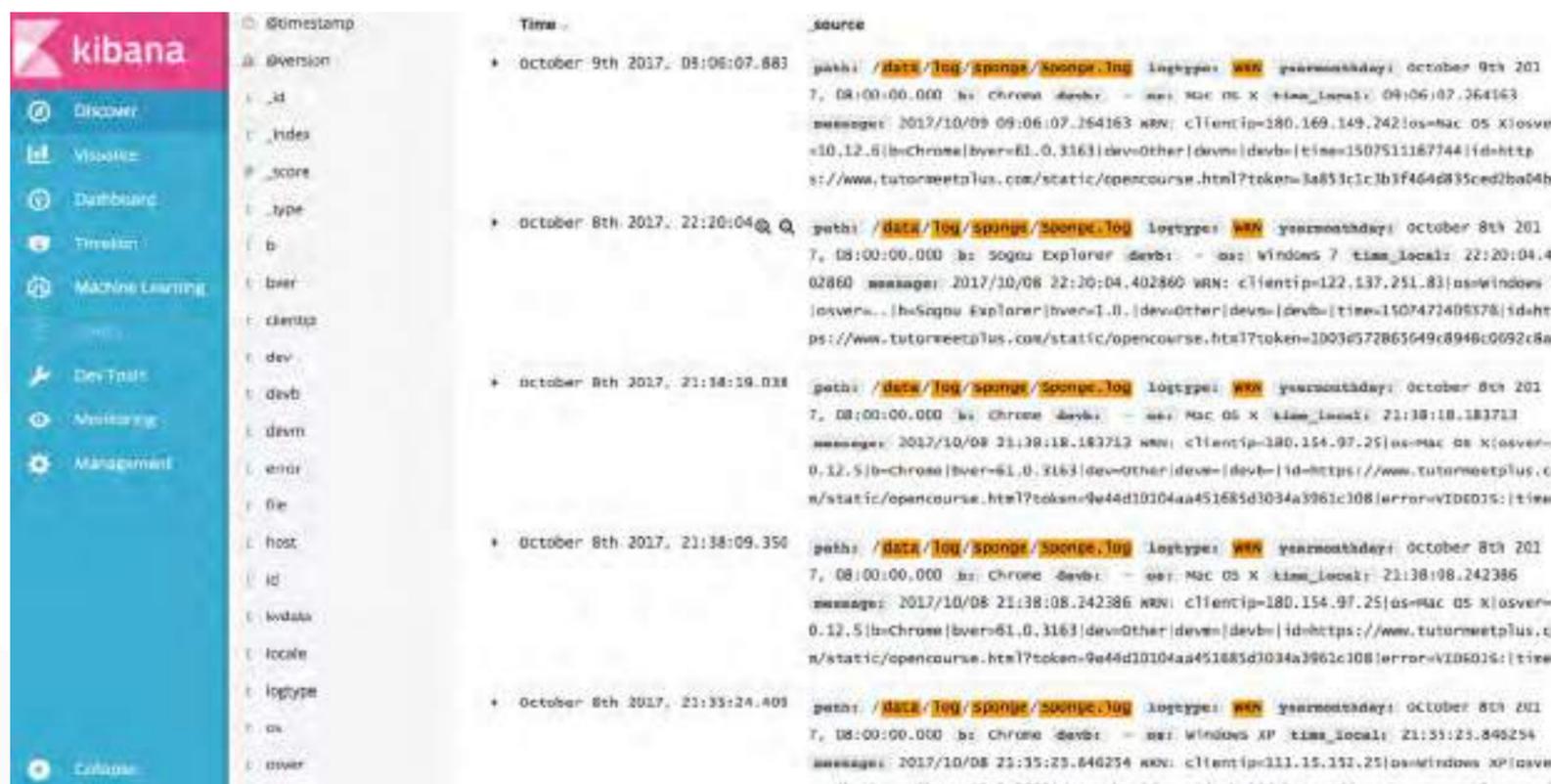
## 错误采集

全量采集 “uncaught error”，资源加载失败等  
按需采集 “caught errors”

3

## 业务数据上报展示

周期性上报客户端“丢包率”，“网络延时”等实时状态  
客户端类型，用户行为、关键流程节点



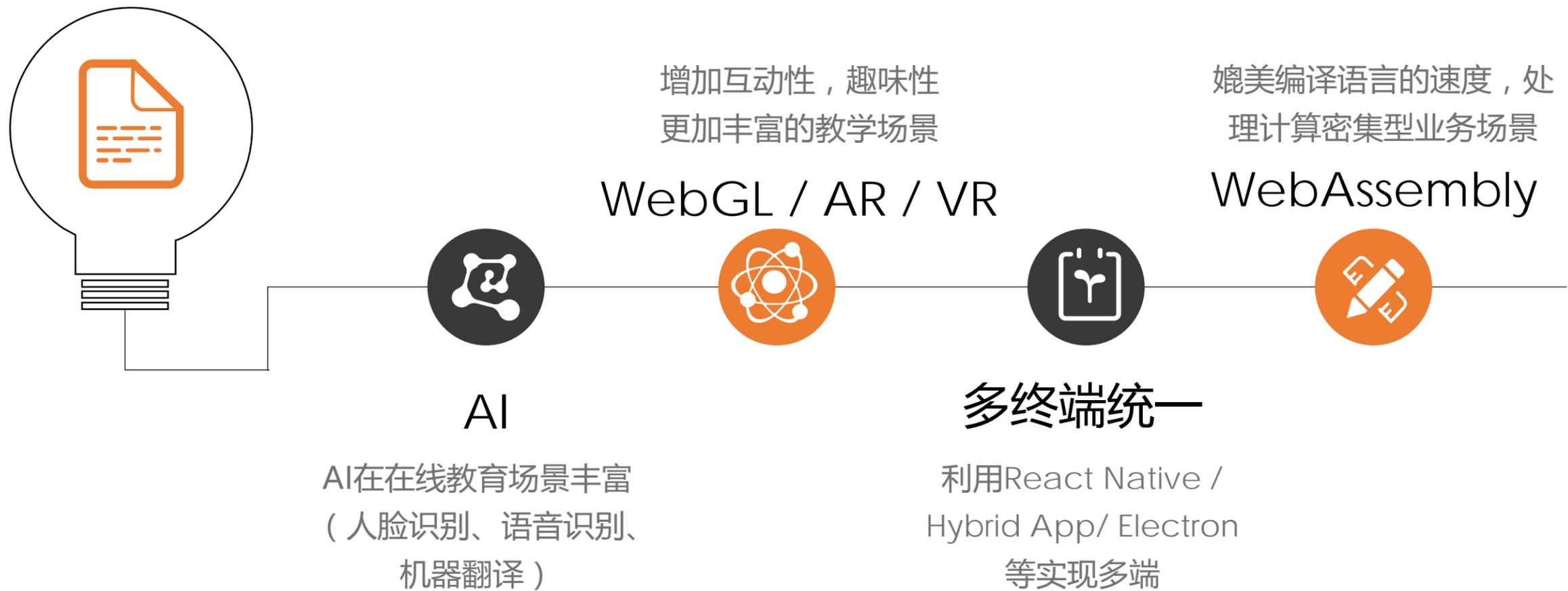
01 对上报数据分类、分级

02 尽量做到“无埋点”

03 声明式埋点 替代 命令式埋点

04 尽量做到按需采集，最小化分析时的“噪音”

kibana Webpack jest PWA  
React.js ES2015+ WebRTC gRPC  
RTMP storybook Javascript Elastic Search  
HLS **Tutormeeet Plus** Sass  
WebSocket npm Jenkins Electron  
Babel PostCSS Logstash Headless browser  
Egret Engine



# Thank You

和君：18021077662

wdzr\_826@163.com



LiveVideoStackCon

聚音视 研修不止于形



关注LiveVideoStack公众号

回复 **和君** 为讲师评分