



# Vue.js

## 渐进式前端解决方案



[vuejs.org](https://vuejs.org)



[@vuejs](https://twitter.com/vuejs)



[vuejs/vue](https://github.com/vuejs/vue)



尤雨溪  
Evan You



@尤小右



@youyuxi



@yyx990803

现在:Vue.js

过去:Meteor, Google Creative Lab

**| 为什么要有框架？**

**框架的存在是为了  
帮助我们应对复杂度。**



...但是框架自身也有复杂度。

A solid red circle.

应用复杂度

**VS.**

A solid purple circle.

框架复杂度




内在复杂度  
Inherent  
Complexity

**VS.**



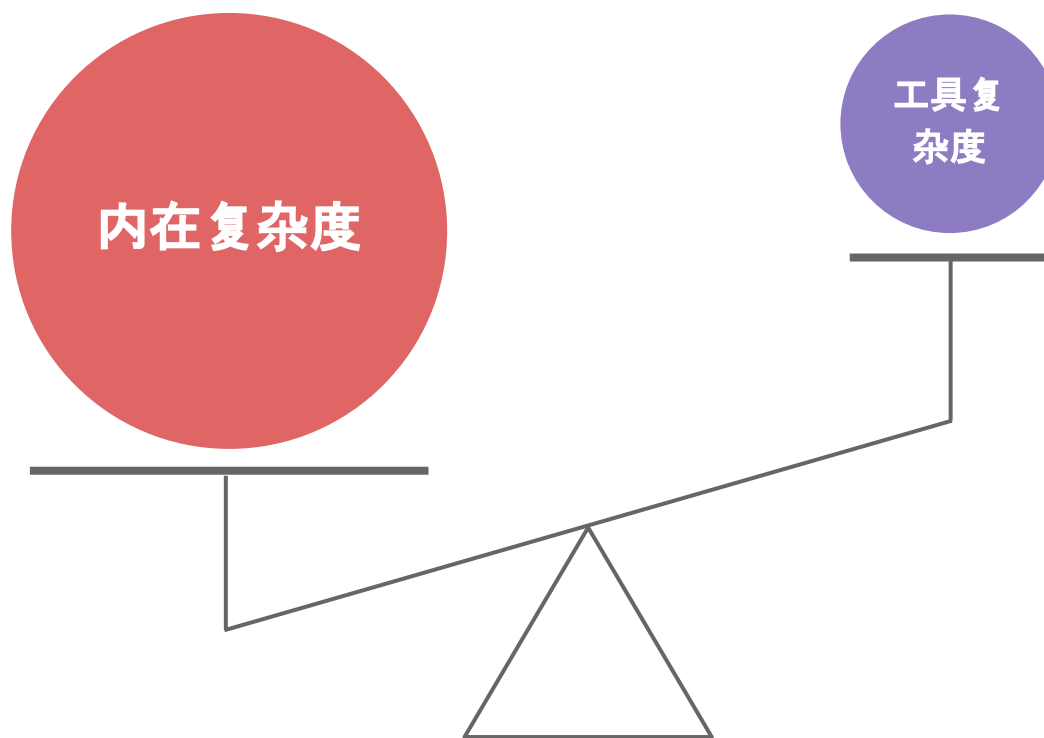
工具复杂度  
Tooling  
Complexity



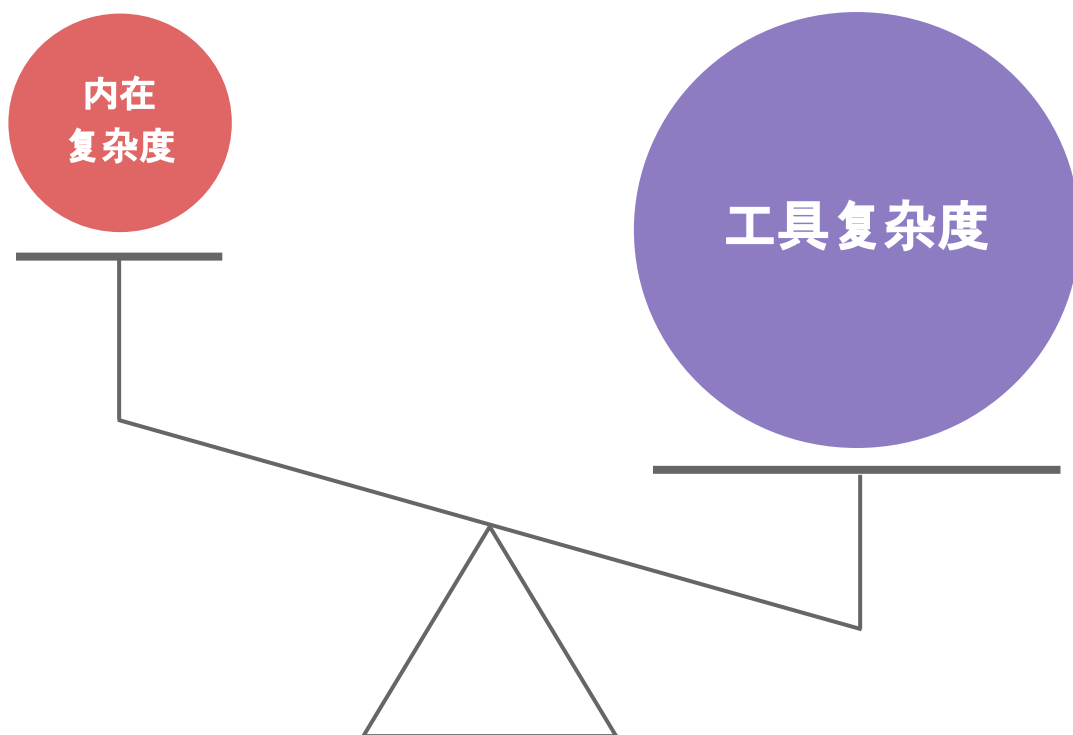
**工具复杂度是我们为了  
处理内在复杂度所做的投资。**



# 工具太弱



# 杀鸡用牛刀



**| “Pick the right tool for the job.”**



声明式渲染

Declarative  
Rendering

组件系统

Component  
System

客户端路由

Client-Side  
Routing

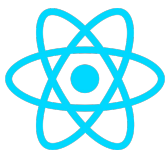
大规模状态管理

Large Scale  
State  
Management

构建工具

Build  
System

纯模板引擎



React



Vue



Backbone



Angular



Ember



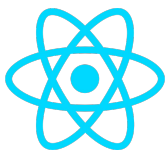
Meteor



Less

More

纯模板引擎



React



Vue



Backbone



Angular

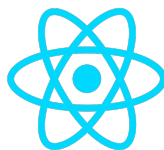


Ember



Meteor

配合  
生态圈的库  
和工具



React+



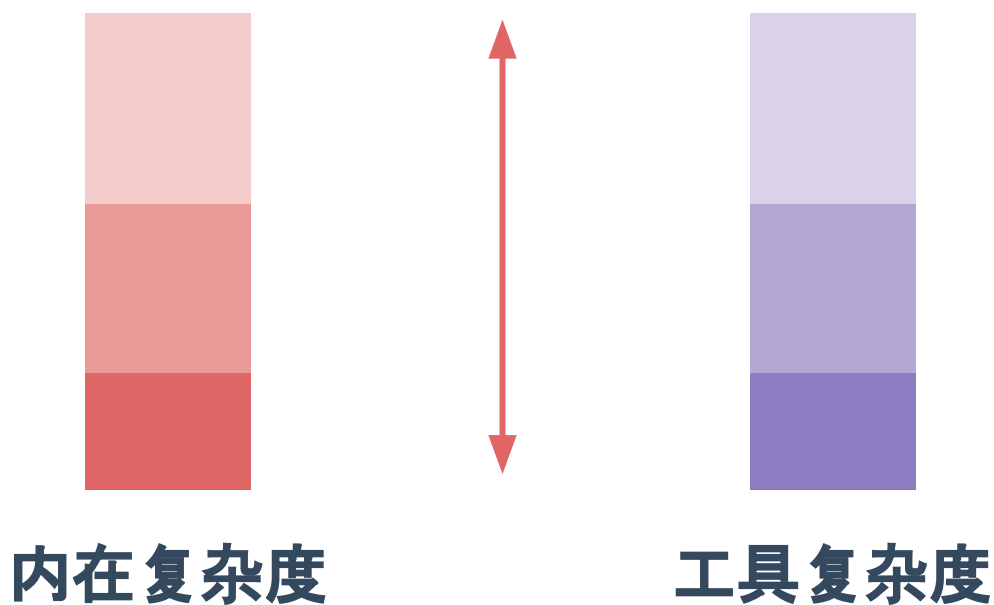
Vue+

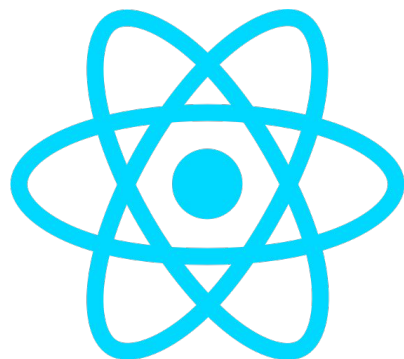


Less

More

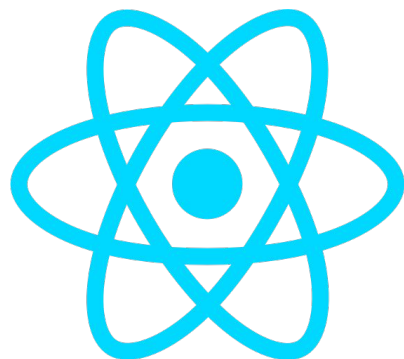
# 可弹性伸缩的工具复杂度 应对不同复杂度场景





**视图层核心**  
**+**  
**可选的附加库/工具链**





视图层核心

+

可选的附加库/工具链



Eric Clemmons

[Follow](#)

Creator of React Resolver, Genesis/Evolution for WordPress. Purveyor of a better Developer Experience...

Dec 26, 2015 · 4 min read

# Javascript Fatigue

A few days ago, I met up with a friend & peer over coffee.

*Saul: "How's it going?"*

*Me: "Fatigued."*

*Saul: "Family?"*

*Me: "No, Javascript."*



**Dan Abramov**

@dan\_abramov



Following

Extreme modularity makes it hard to build a cohesive experience.

RETWEETS

76

LIKES

190



8:09 AM - 7 Aug 2016



76



190



# Vue.js

## 渐进式框架

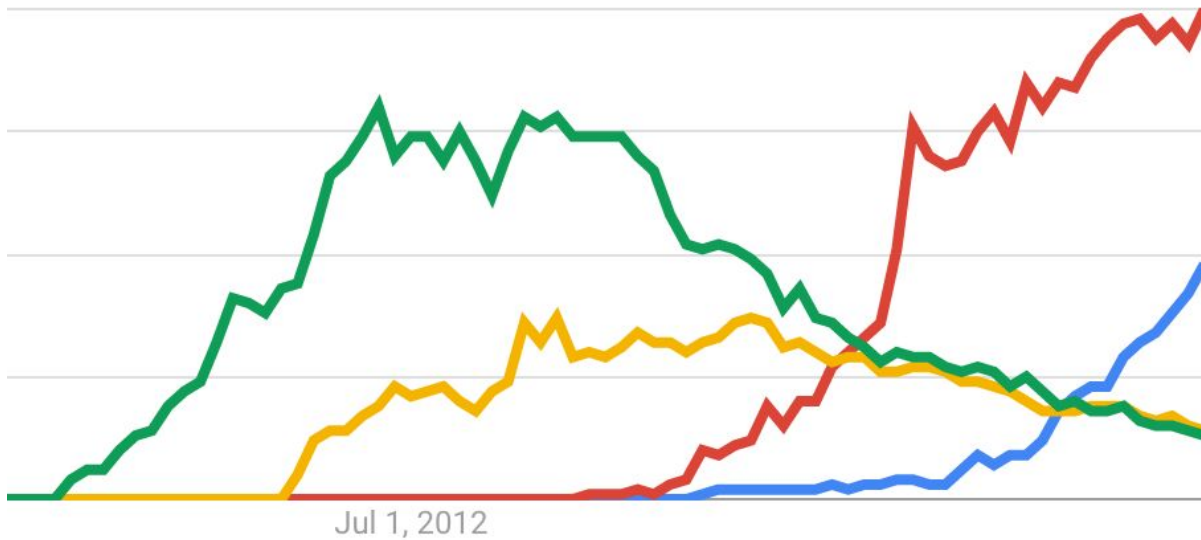
### Progressive Framework

# 现状

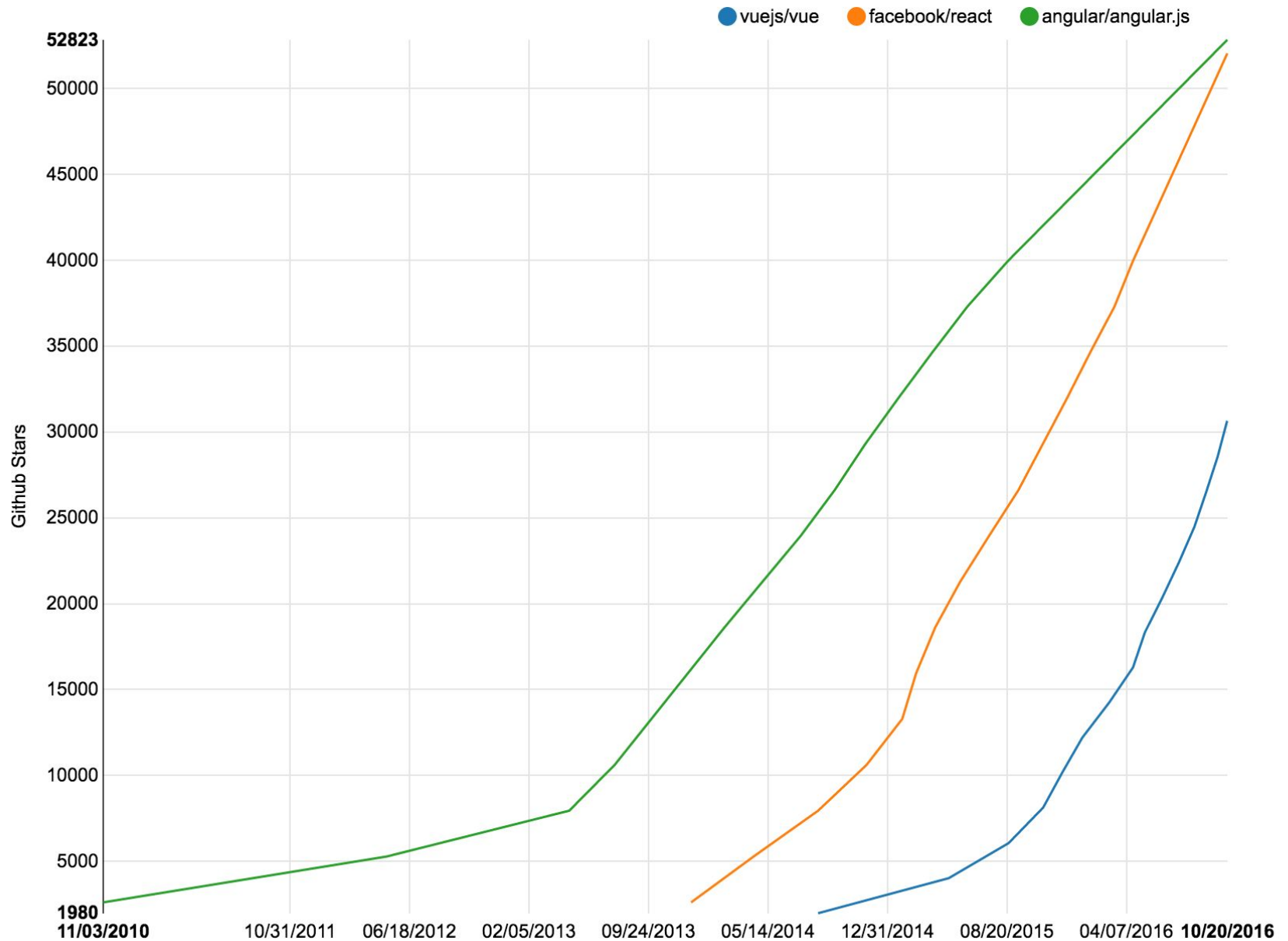
- ~30k stars on GitHub
- ~185k/mo downloads on NPM
- ~264k/mo unique visitors to vuejs.org
- ~55k weekly active Chrome extension users

# Google Trends

● vuejs + vue.js   ● reactjs + react.js   ● emberjs + ember.js  
● backbonejs + backbone.js



# GitHub Stars



# pro·gres·sive

/prə'gresiv/

*adjective*

1. happening or developing gradually or in stages; proceeding step by step.

"a progressive decline in popularity"

*synonyms:* continuing, [continuous](#), increasing, growing, developing, [ongoing](#), accelerating, escalating; [More](#)





声明式渲染

Declarative  
Rendering

组件系统

Component  
System

客户端路由

Client-Side  
Routing

大规模状态管理

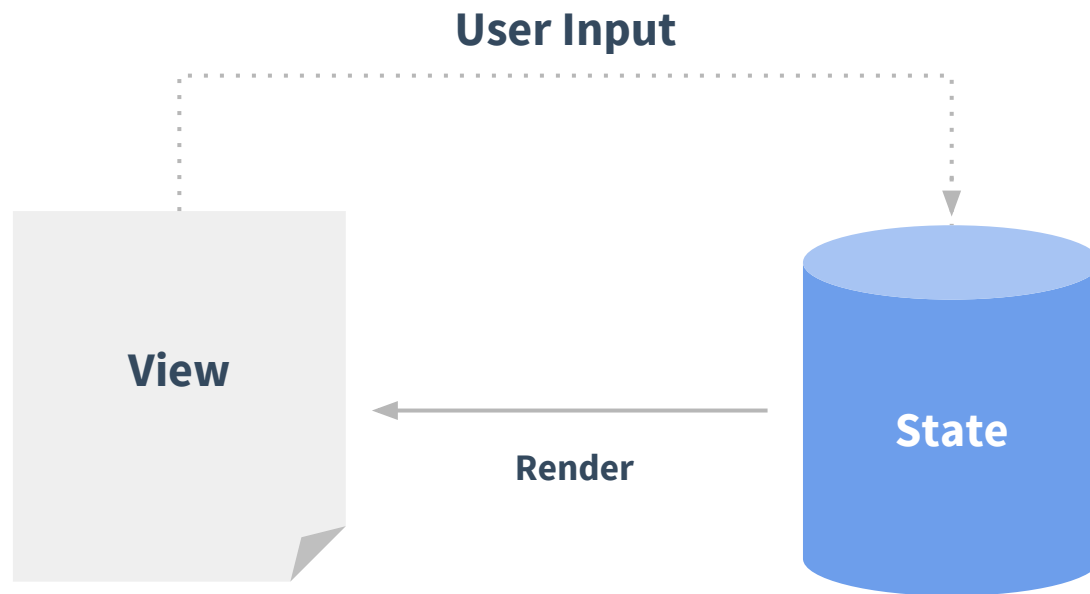
Large Scale  
State  
Management

构建工具

Build  
System

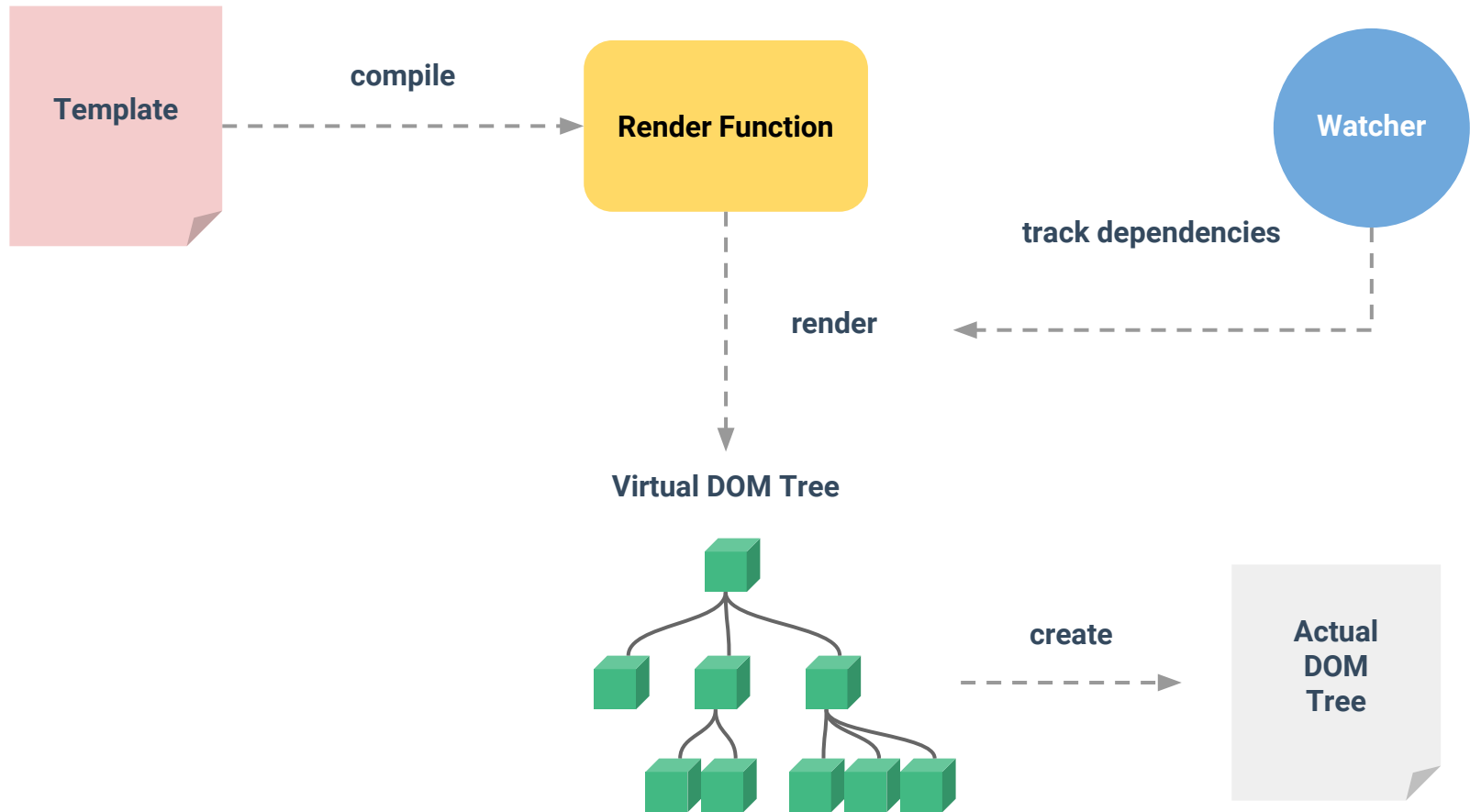
# 声明式渲染

## Declarative Rendering

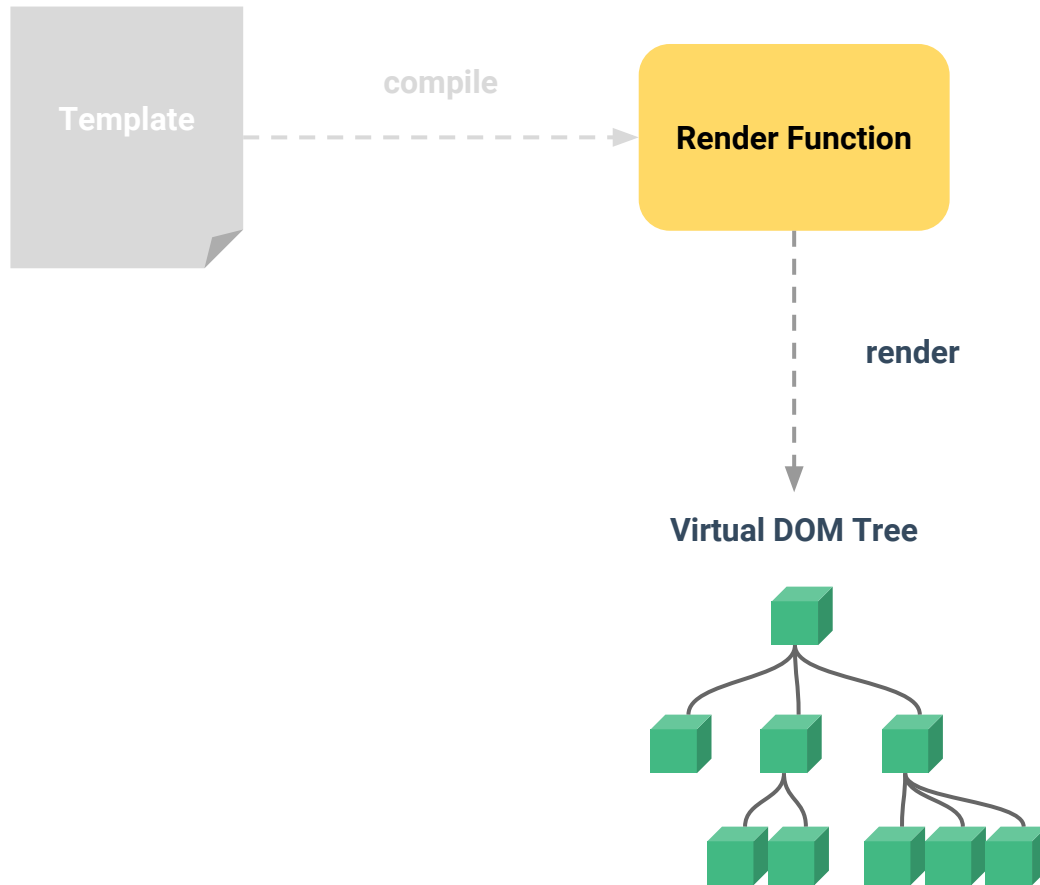


```
<div id="app">
  <ul>
    <li
      v-for="todo in todos"
      v-on:click="todo.done = !todo.done">
      {{ todo.title }} ({{ todo.done ? 'done' : 'nah' }})
    </li>
  </ul>
</div>
```

# Virtual DOM



# Render Function



# Render Function

```
<div class="message">  
  Hello World  
</div>
```



```
render (h) {  
  return h(  
    'div',  
    { class: 'message' },  
    ['Hello World']  
  )  
}
```

# Render Function

```
<div class="message">  
  Hello World  
</div>
```



```
render (h) {  
  return h(  
    'div',  
    { class: 'message' },  
    ['Hello World']  
  )  
}
```

## 模板

书写便利、易读  
更接近文档结构和 CSS

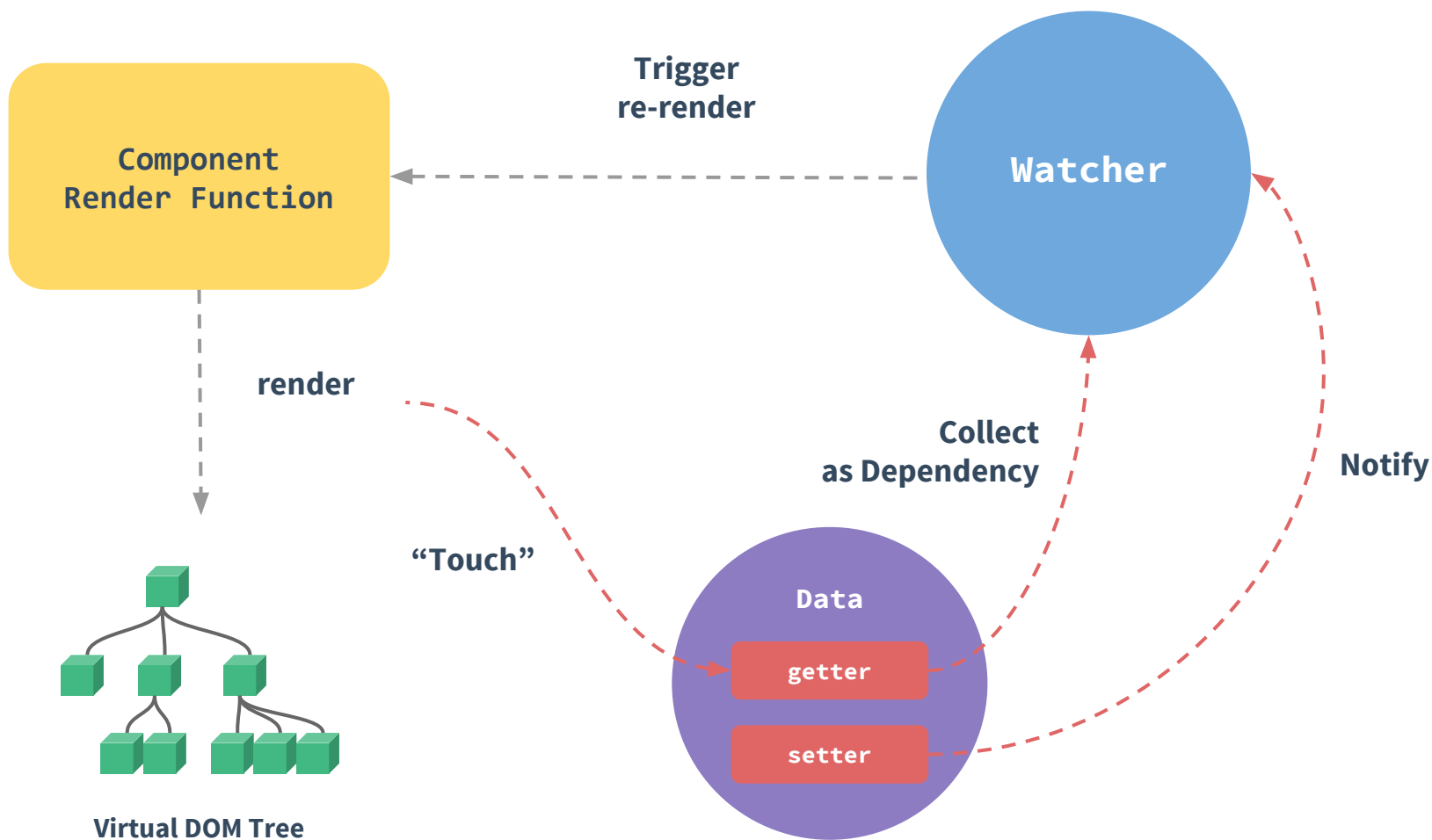
## 函数/JSX

更灵活、强大  
能够表达更复杂的逻辑



# 高效的依赖追踪系统

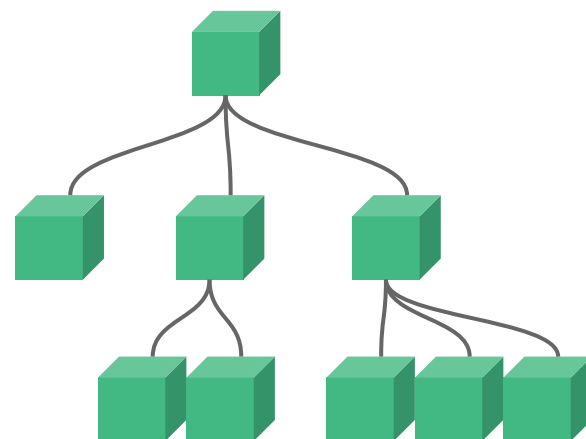
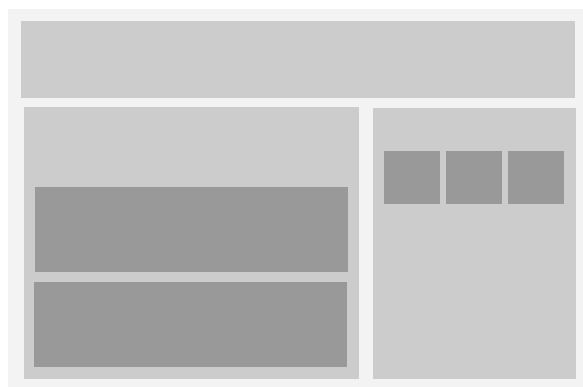
精确侦测需要重绘的组件, 无需手动优化



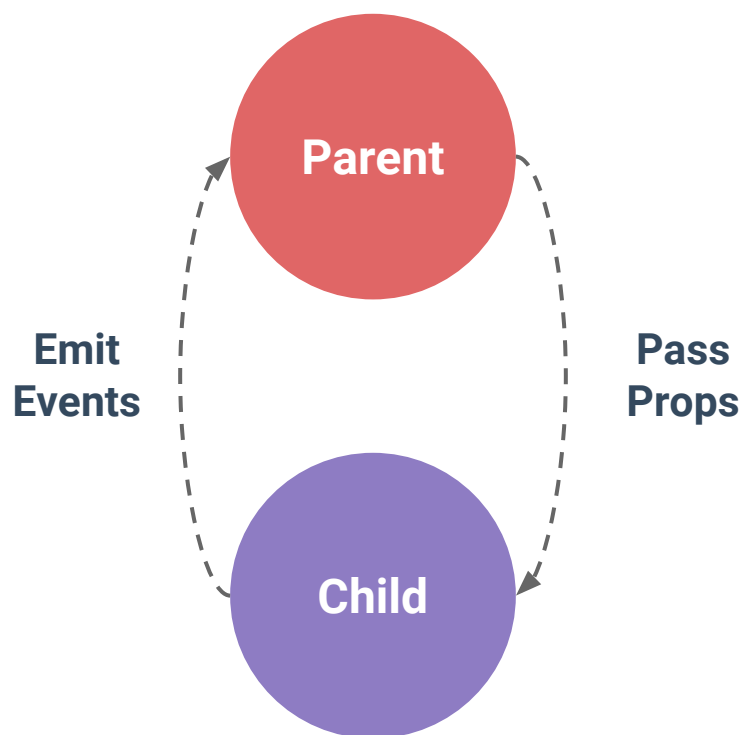
# 组件系统

## Component System

UI 结构 → 组件树



**Props: 父 -> 子 单向数据流**  
**Events: 子 -> 父 触发副作用**



```
example.vue
<template lang="jade">
  div.my-component
    h1 {{ msg }}
    other-component
</template>

<script>
import OtherComponent from './other-component.vue'

export default {
  components: { OtherComponent },
  data() {
    return {
      msg: 'Hello Vue.js!'
    }
  }
}
</script>

<style lang="sass" scoped>
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

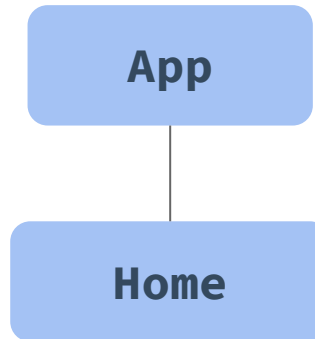
.my-component {
```

\*.vue  
单文件组件

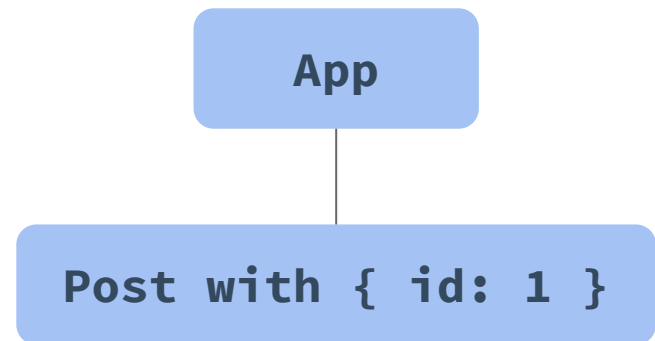
# 客户端路由

## Client-side Routing

**/app/**



**/app/post/:id**



# 客户端路由的各种问题

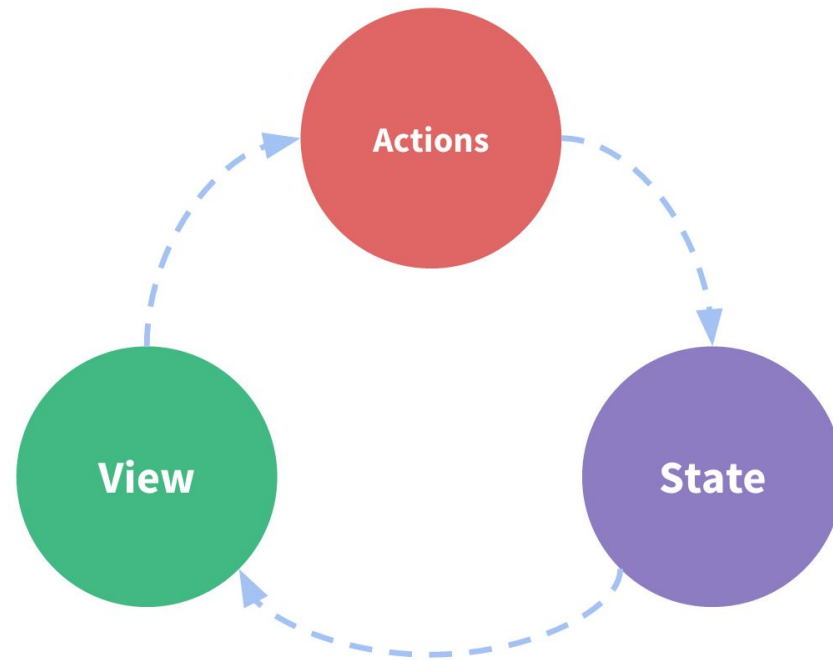
- 嵌套路由
- 具名路由
- 多个平级路由出口
- 复杂匹配规则
- 当前活跃链接
- 重定向 / 别名
- 跳转动画
- 异步数据处理
- 跳转规则限制
- 滚动条行为
- 懒加载

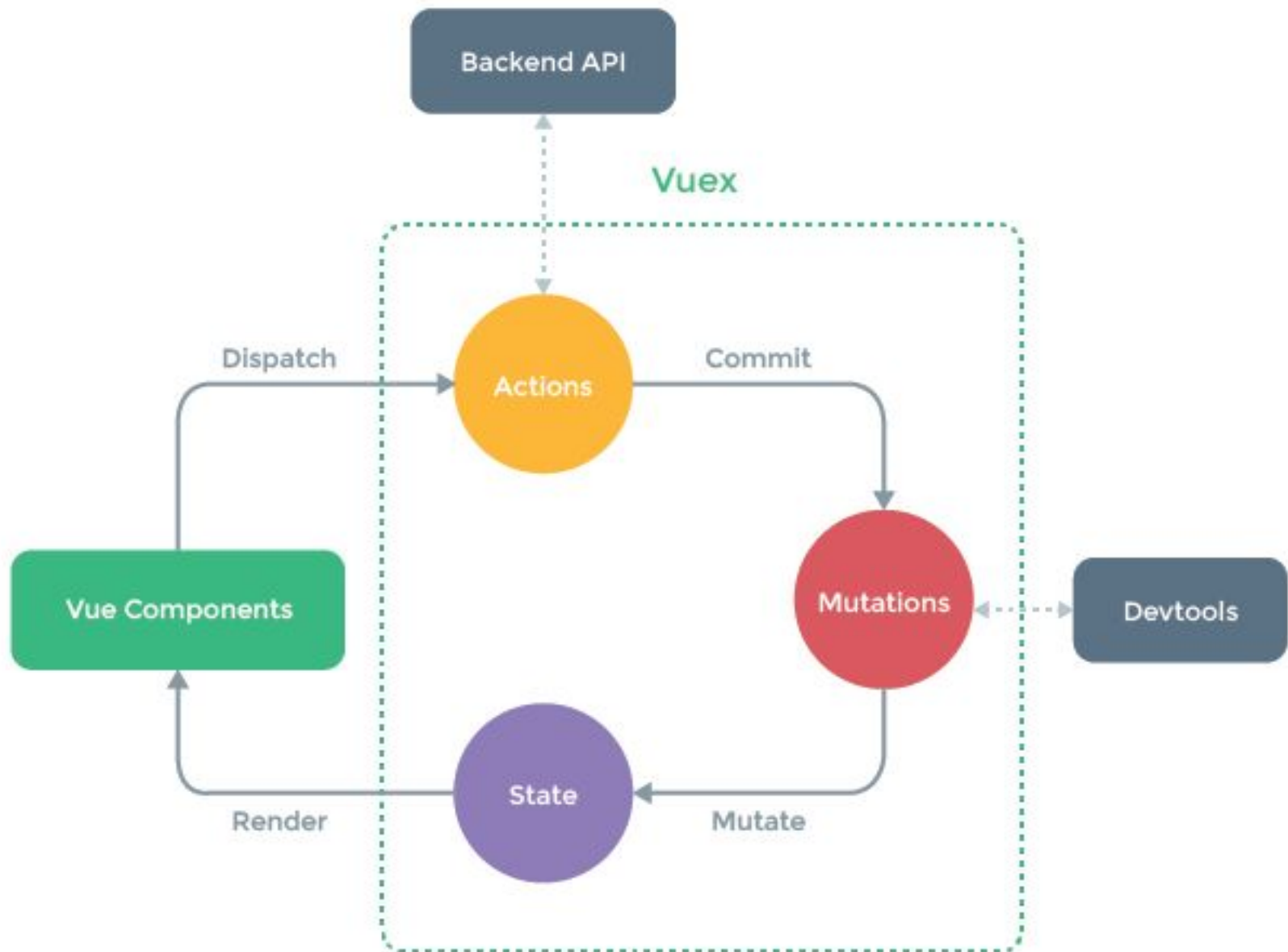


**| router.vuejs.org**

# 状态管理

## State Management

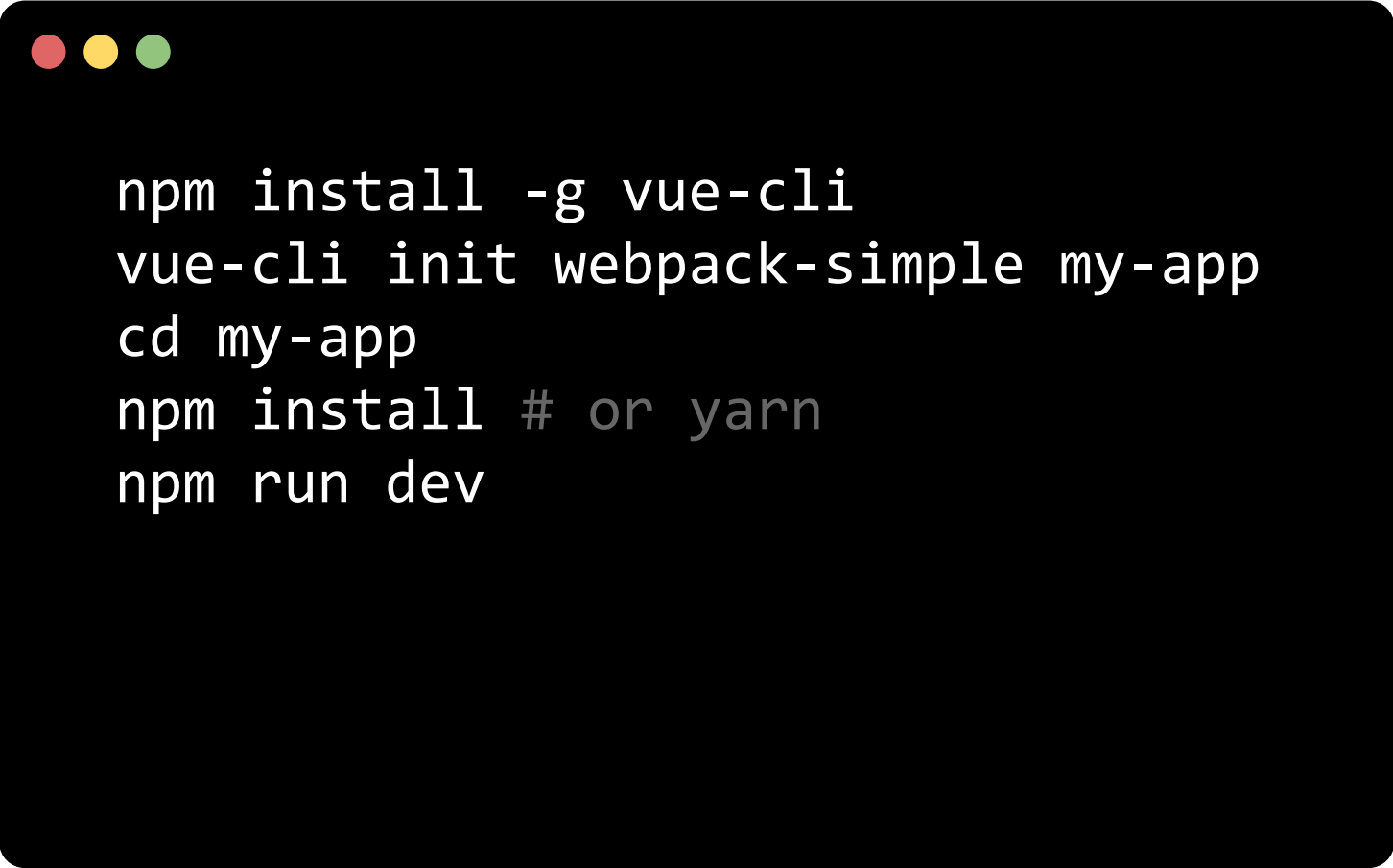




 **[vuex.vuejs.org](https://vuex.vuejs.org)**

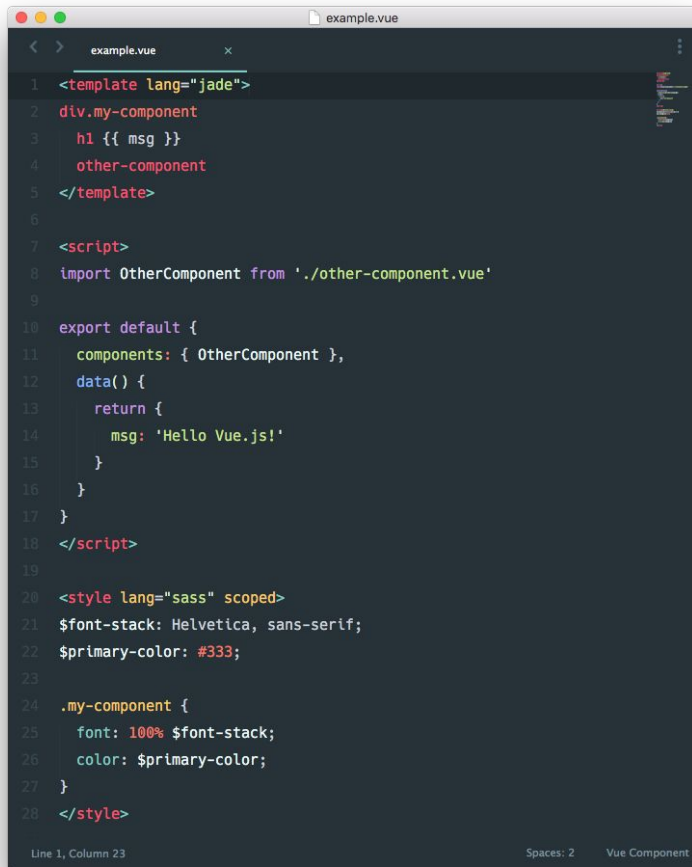
# 构建工具 / 开发体验

## Build Tools / Dev Experience



```
npm install -g vue-cli  
vue-cli init webpack-simple my-app  
cd my-app  
npm install # or yarn  
npm run dev
```

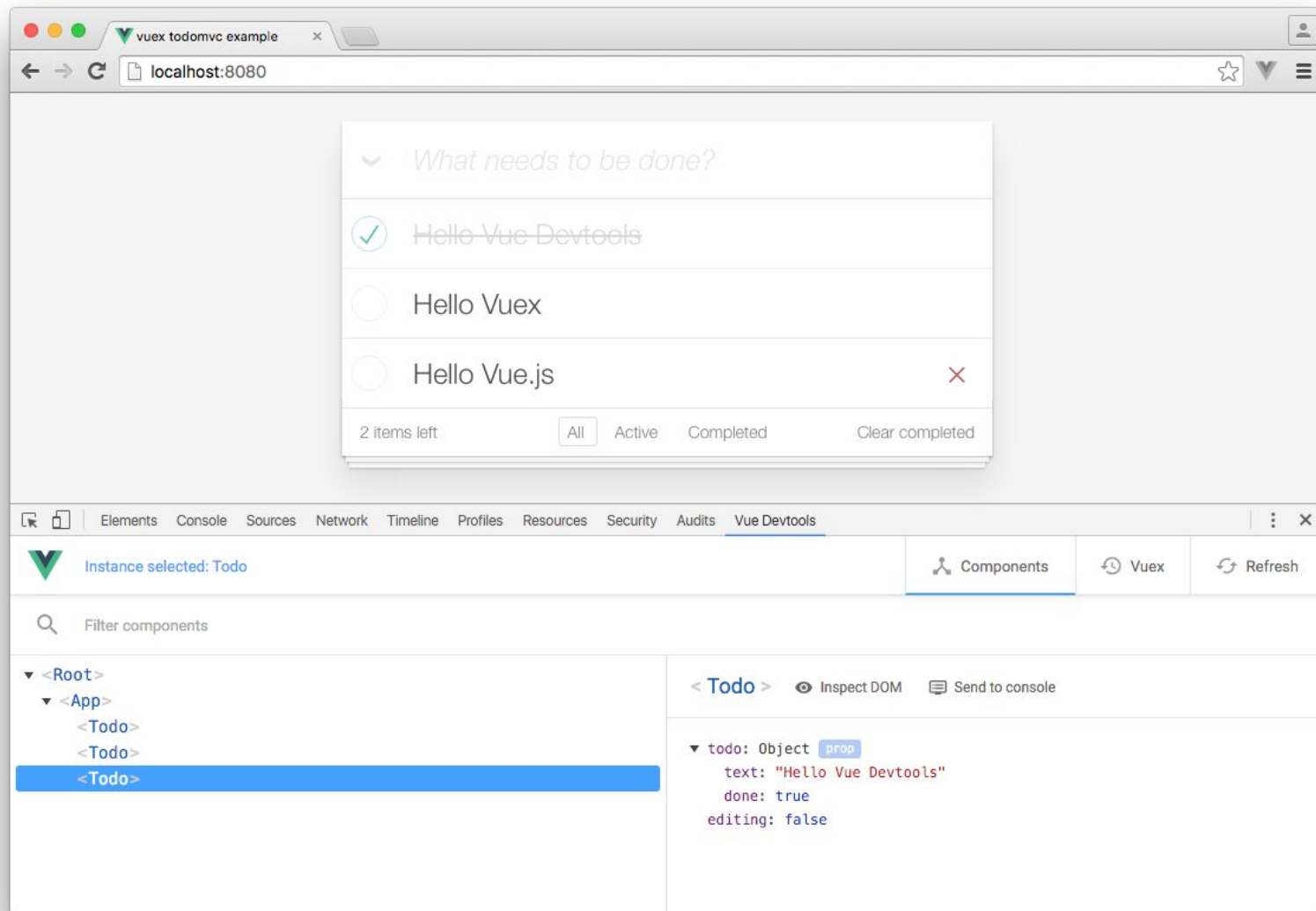
# Webpack 支持下的单文件组件



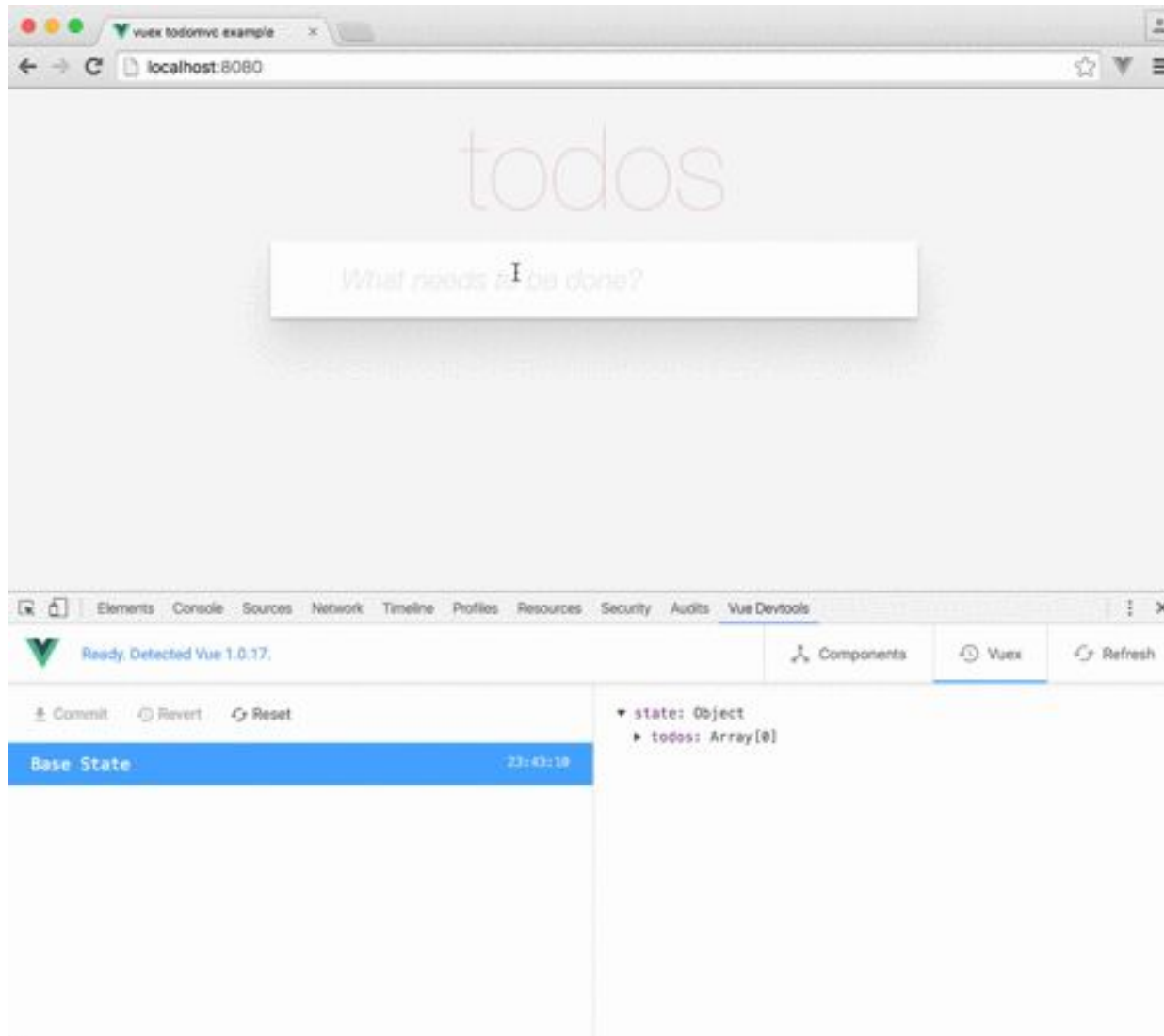
- 集中组件相关的模板, 逻辑和样式
- 编译为 JavaScript 模块
- 各语言块支持任意编译 / 预处理器 : Babel, TypeScript, LESS, Sass, PostCSS, Pug...
- 开箱即用的热重载, 保存即更新
- Scoped CSS 支持
- 配合 Webpack 和 Vue 的异步组件特性 轻松实现懒加载



# Chrome 开发者工具



# Vuex Time-Travel Debugging



# Vuex 状态快照导出 / 导入

Components

Vuex

Refresh

Export

Import

▼ state: Object

▶ todos: Array[1]

```
{
  "todos": [
    {
      "text": "Hello
      QCon",
      "done": false
    }
  ]
}
```

# Vue 2.0

# 更轻



**2.0 Runtime Build**  
16kb min+gzip

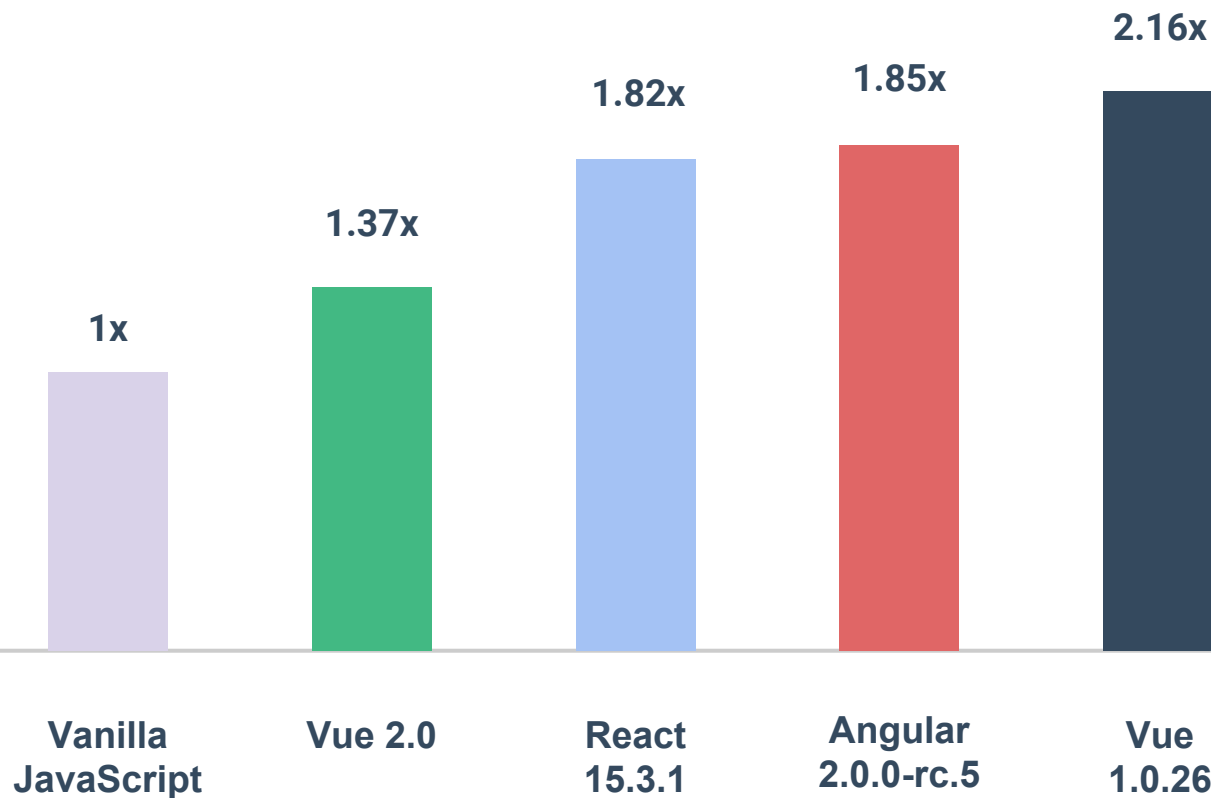


**Vue + vue-router + vuex 2.0**  
26kb min+gzip



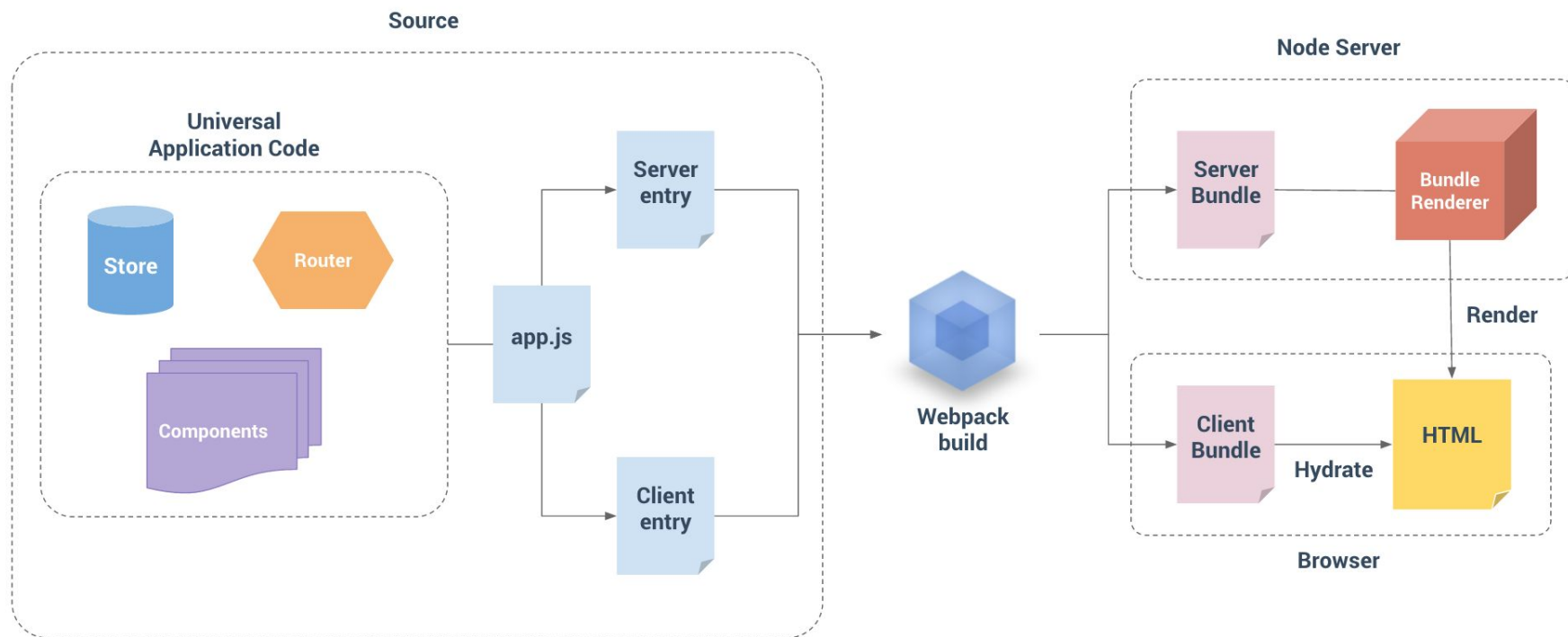
**1.0.26**  
26kb min+gzip

# 更快

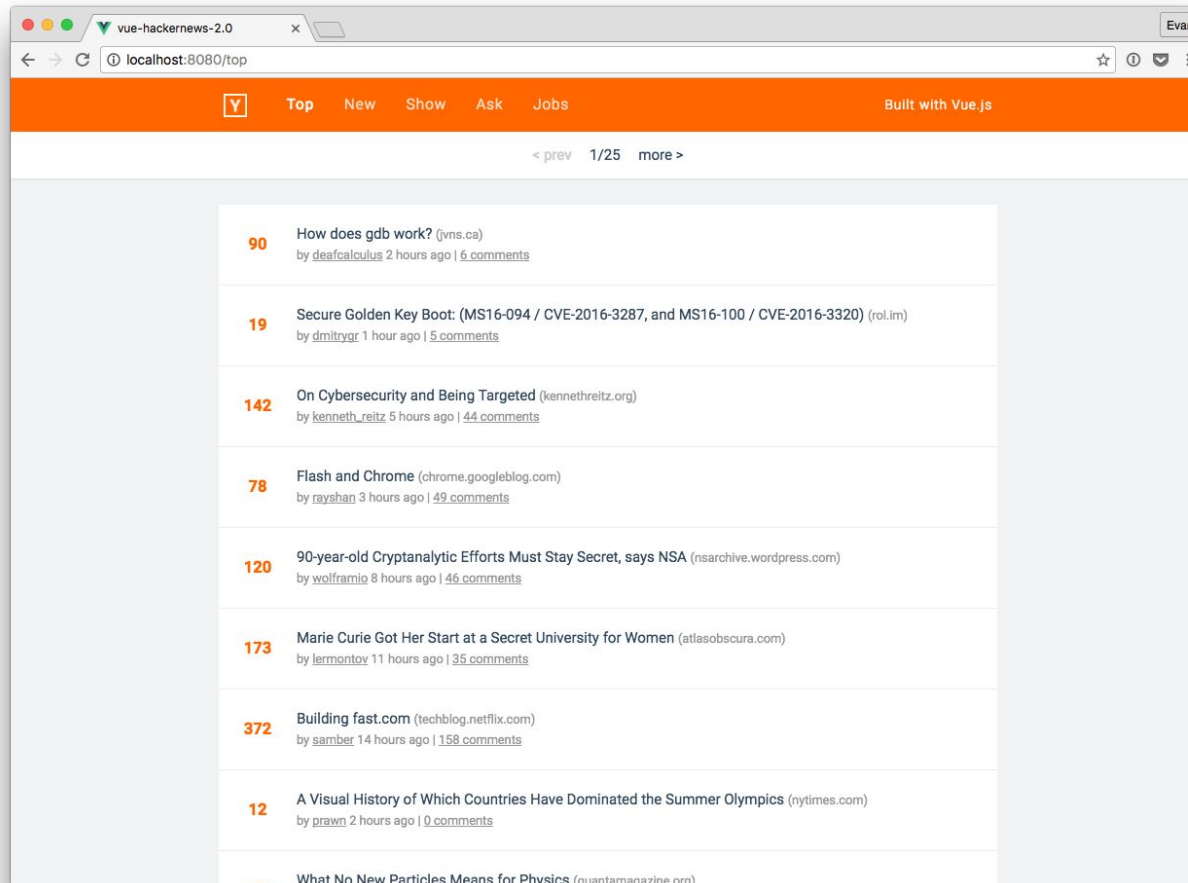


# 服务端渲染

支持流式渲染 + 组件级别缓存



# vue-hackernews 2.0



<https://github.com/vuejs/vue-hackernews-2.0>

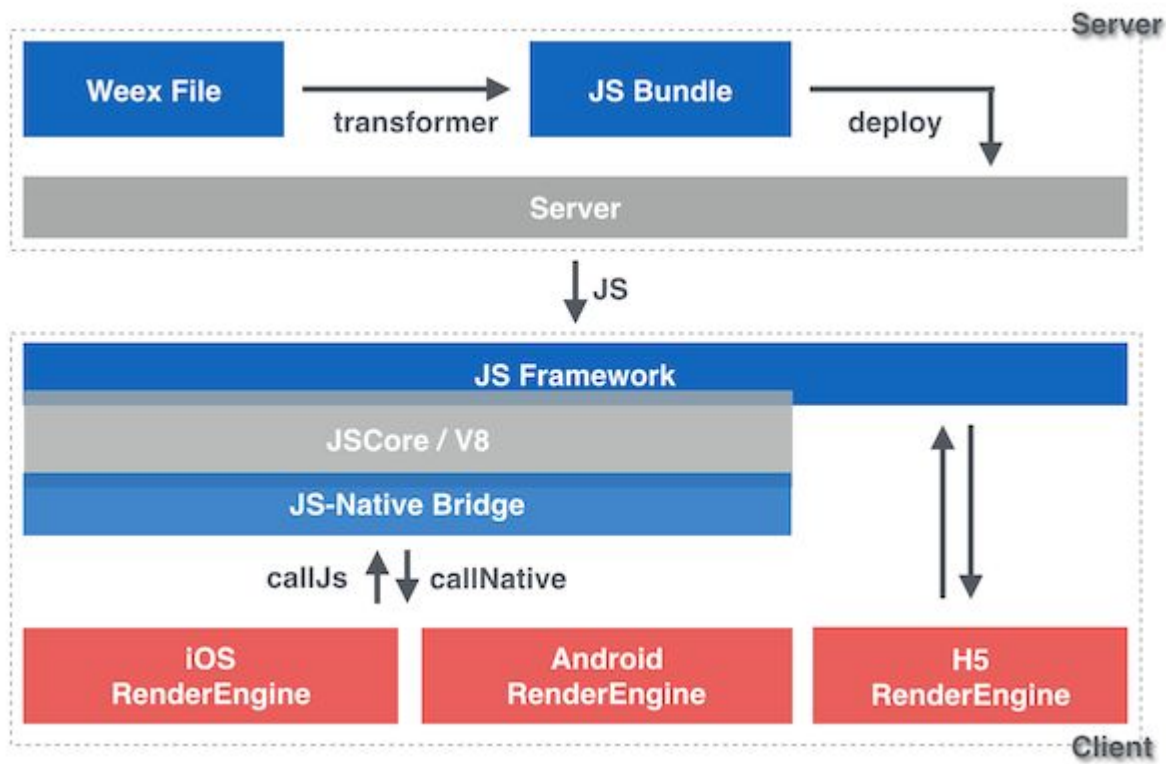


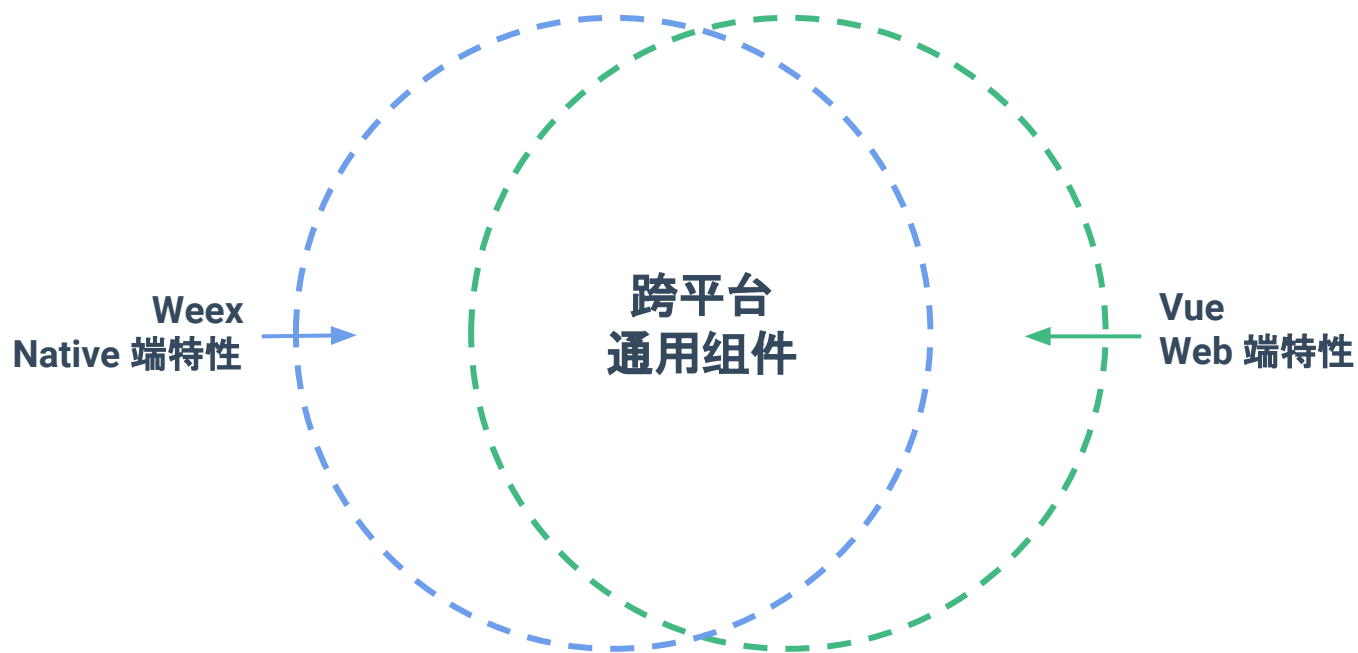
# 原生渲染

## Native Rendering



<https://alibaba.github.io/weex/>







# Thanks!