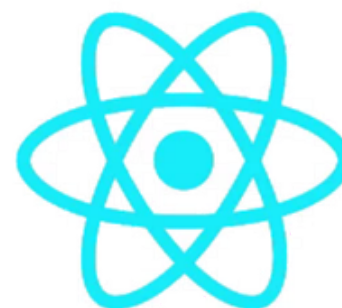


Meteor+react+redux

全栈式开发方案



Meteor 是什么？

Meteor 是一个全栈型 JS 开发平台，基于 Nodejs 技术。所谓全栈，就是我们可以用一种语言 JS，同时开发前端和后端程序，用 Meteor 技术开发的 App 不仅仅可以运行在各个浏览器中，包括微信，也可以编译成原生应用，运行在 ios 和 andriod 系统之上。meteor 差别于一般的框架，meteor官方的说法是：





















Meteor is Unix , not Rails 。

为什么讲meteor？

- 全栈随动**基于实时数据订阅的，是以实时性为默认的框架**
- spa单页面应用，Meteor 是一个全栈 SPA 框架，这个其实是很特别的。
- 类原生应用
- 拥抱各种先进的主流生态系统(跨平台开发 iOS Android Browser Mobile)
- Javascript 全平台单一语言开发
- 时刻拥抱变化（热部署）
- 代码压缩混淆部署都由Meteor透明完成
- 可以使用npm扩展

Full-stack frameworks

That's where Node.js really shines. The full-stack MVC frameworks are bundled with scaffolding, template engines, websocket and persistence libraries to allow you build real-time scalable web apps.

- ✓ AllcountJS  Star 278 
- ✓ Derby  Star 4,105 
- ✓ Feathers  Star 4,224 
- ✓ SocketStream  Star 3,424 
- ✓ MEAN.js  Star 3,637 
- ✓ MEAN.io  Star 9,324 
- ✓ Meteor  Star 
- ✓ Meatier  Star 2,585 
- ✓ TWEE.IO  Star 118 
- ✓ Mojito  Star 1,628 

全栈随动 (Full Stack Ractivity)

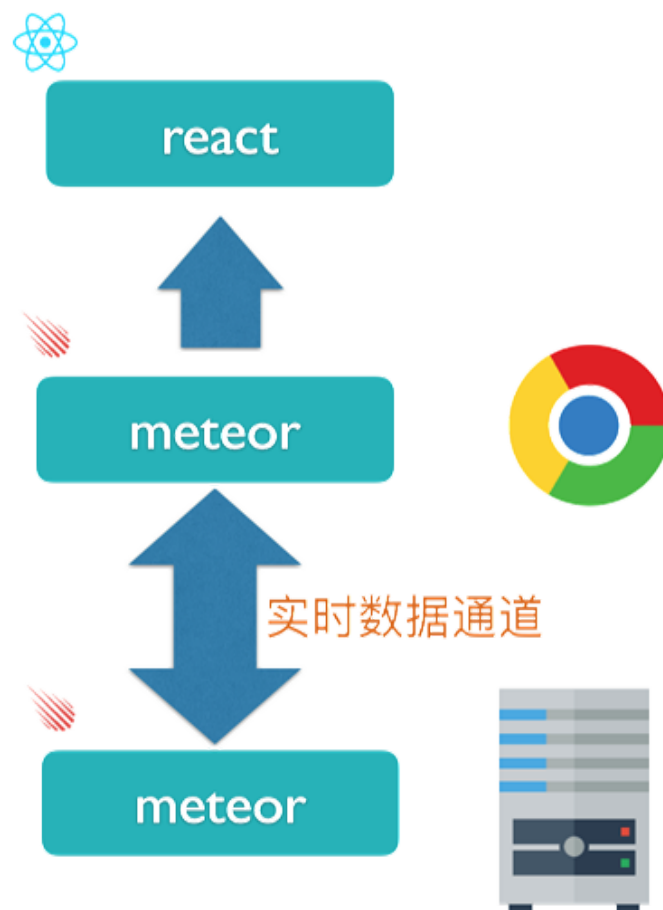
那么 meteor 的优势体现在哪里呢？应该说体现在很多方面，但是这里我想强调一点：全栈随动 (Full Stack Ractivity)

全栈随动 full stack reactivity

数据一旦变，界面自动变

后台数据变，前台自动变

前台数据变，后台自动变



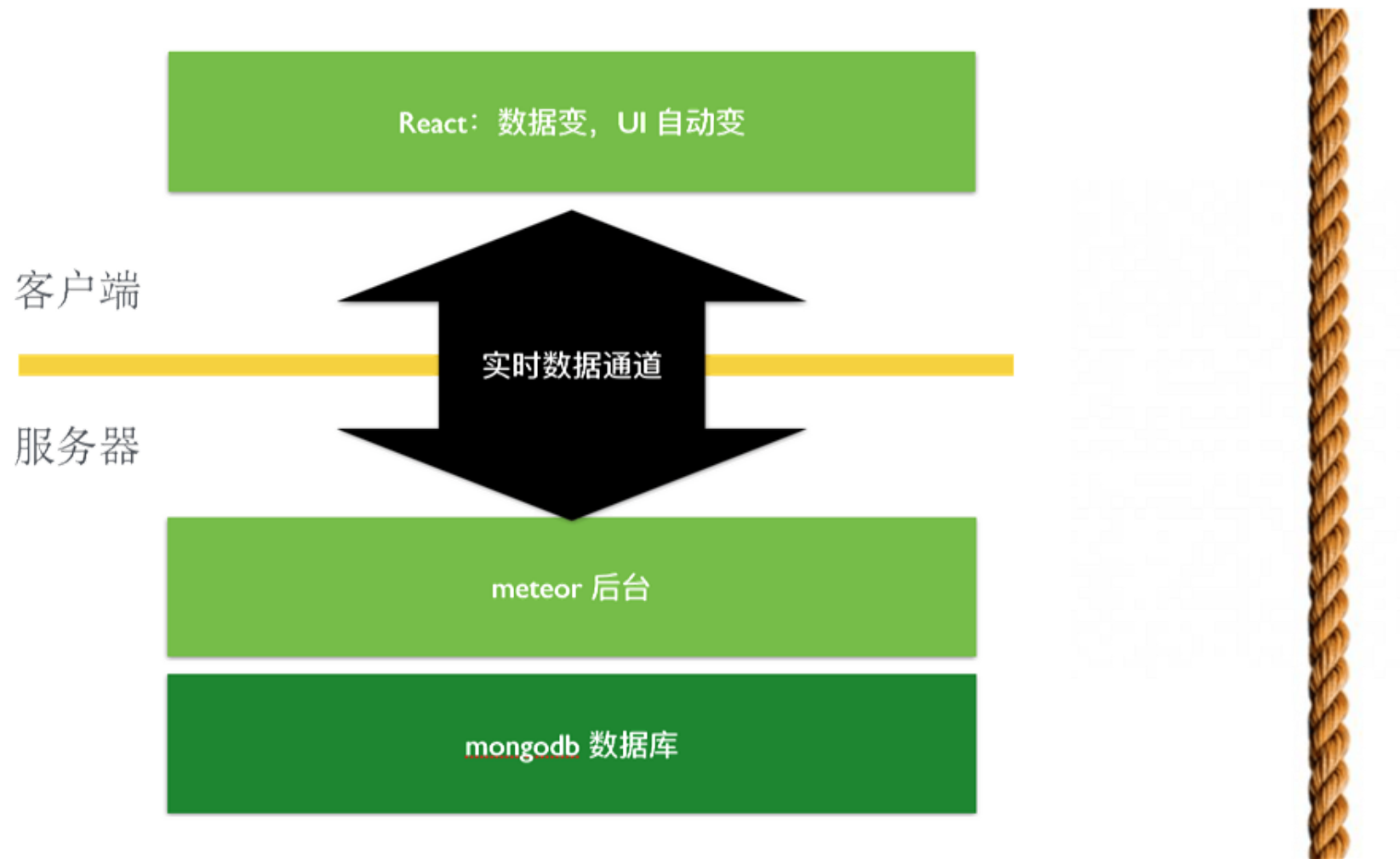
基本的意思就是说，整个应用的数据已经成了“一根筋”：
后端和前端数据通过实时订阅通道，自动同步

Meteor的工作原理

大家都知道 [Socket.io](#) 是当前写实时应用的最佳解决方案了，底层基于的技术就是 Websocket。Meteor 把 Websocket 技术融入到了自己的核心，因为 Meteor 是以实时性为默认的平台。传统的 Web 应用都是走 请求/应答 模式来从服务器端获取数据的，但是 Meteor 采用的却是一套不同的思路。Meteor 通过让客户端代码订阅服务器数据的形式，实际上打通了客户端和服务器的实时数据通道。

React 和 Meteor是天生一对

Meteor 是有实时数据通道的，也就是说后端数据如果变化了，前端自动跟着变，反之亦然。而 React 的特点是前端数据如果变了，那界面自动就更新。这两点加到一起结论就是：应用的任何一个位置有了数据修改，界面上马上就会自动体现出来。Meteor 和 React 一起用是非常棒的。Meteor 提供了简单易用的跨客户端和服务器的数据管理，React 提供了组织复杂 UI 的组件化的思路。由于 React 只是一个库，不是框架，它只负责 MVC 的 V，也就是视图层的功能。而 Meteor 提供了一条简单的路。首先 Meteor 在 View 层可以直接支持 React，这是官方保证的。Meteor 提供了前面我们提到的各种 React 构建完整的 App 所必须的部分，好处是都有高手对这些部分进行的精心的选择和集成。



多平台通吃

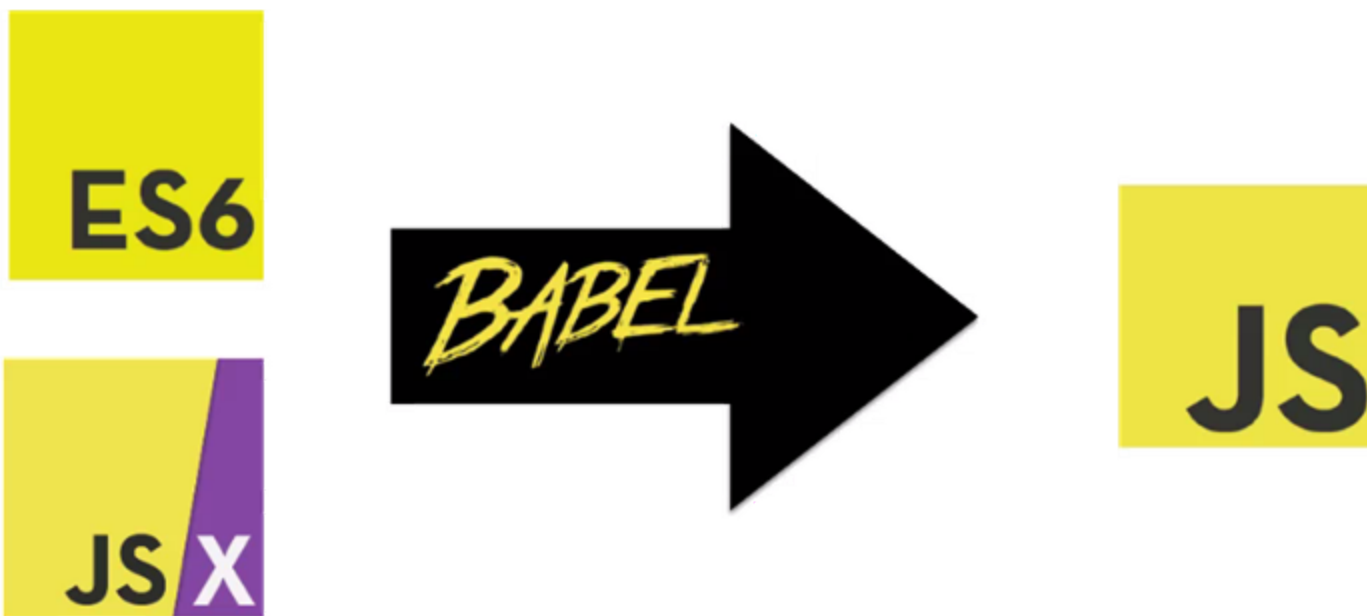
另外，利用 <http://cordova.apache.org/>，meteor 写成的 Web 应用，可以直接编译成 ios 或者 andriod 的应用。另外一个方案就是使用 react-native)，也可以让我们用 JS 就写成手机 App 。

API 抽离增加了工作量

API 抽离增大工作量

第一点，API 抽离增大工作量。前后端都是他自己一个人弄，不管是同步开发，还是一个人开发，效率都很高。但是现在是 React + API 的前后端分离形式，发现真是一个双刃剑。其实原则应该是这样，自己项目的业务逻辑还是一体完成比较好，硬性分离出 API 是愚蠢的。只有当使用第三方功能，或者自己有高度重复的业务的时候，抽出 API 才是值得的。因为抽出 API 是有成本的：第一，Meteor 为我们做的那些事情，基本上现在都要自己手动做了，例如 [Socket.io](https://socket.io/) 实时通讯。第二，每次实现一个小功能，都要两个人碰一下才行，不利于开发效率

Meteor-随时调整



METEOR

React

npm

Meteor



webpack
MODULE BUNDLER

Meteor 也拥抱 JS 社区的各个其他主流生态系统。

Meteor 拥抱 NPM，meteor 自带的功能少得很。可以安装 meteor 的包来进行扩展。Meteor 自己的包仓库，目前已经有近一万个包了。另外，meteor 背靠整个 npm 生态，安装普通的 npm 包进 meteor 也是非常方便的。

选择 React。Meteor 和 React-Native 也很容易在网上搜到资料。

很多工具例如支持 ES6 和 JSX 编译的 Babel，如果在 Webpack 或者 gulp 条件下自己去配置，很多时候可能就会出问题，但是 Meteor 这里都内置了，新手不用配置就能用。

Meteor 有自己的构建系统（提供类似于 Webpack 的功能），可以进行代码转翻，页面自动刷新等各种功能。同时后台的各种功能，服务器，数据库适配，实时数据传输也都为我们做好了。

总结，简单意味着更少的时间花在工具配置上，更多的时间用于实现功能。Meteor 可以让我们快速成型项目。


```
function cc(){  
  var dfd= kfj;  
}
```

Meteor 是以实时为默认

由于有了实时数据的订阅，一旦服务器上的 comments 列表有了变化。那么客户端的 comments 数据就会自动变化，而 React 组件的特性是，一旦 state 值（对应当前情况，就是 comments ）有了变化，界面就能自动刷新(内置的 [socket.io](#)) 。

所以最终的结果是：一旦有其他人发布了评论，我自己的浏览器中是会自动显示出来的，也就是 Meteor-React 实现的这个评论框，就是一个聊天室。