百度APP移动搜索落地页体验白皮书5.0 <https://ziyuan.baidu.com/college/documentinfo?id=2925>快手内部文档请勿外传

# 单页应用、多页应用、客户端渲染、服务器渲染

## 单页应用

1、单页应用并不是说用户最终看到的就只有一个页面，而是指在开发层面上只有一个 html 文件，至于用户看到的多个页面均是在这一个 html 文件中对应生成。

2、由于单页应用中“多页”的生成是在同一个 html 中部分渲染完成的，所以不需要浏览器重新下载、解析、渲染另外的 html 文件，响应速度更快，用户的体验更好，这应该是最大的优势。

3、由于单页应用中的“多页”是通过 js 生成的，而传统的爬虫是需要分析完整的 html 文件，所以不利于 SEO （搜索引擎优化），至于 SEO 就是为了让你输入某个关键词后能够更好更靠前的搜索到你的应用。

4、单页应用在首屏渲染的时候需要加载的东西过多，往往会导致首屏速度很慢。

## 多页应用

1、与单页应用相反，多页应用指的是每个页面独立对应一个自己的 html 文件。

2、每个页面的跳转都需要重新下载、解析、渲染 html 文件，响应速度较慢。最大的劣势吧。

3、每个页面都是完整的 html ，便于 SEO 。

4、每个页面相互独立，可以分别升级维护，但涉及到不同页面之间的数据交互就比较麻烦。

## 服务器渲染

1、服务器渲染，也叫后端渲染，就是指在后台生成 html 并在相应的地方插入好数据然后传递给前端（浏览器）。

2、后台输出的是完整的 html ，利于 SEO ，但也很死板。

3、服务器一般计算速度更快，但会占用了相应的服务器资源。

4、传递给前端的数据量更大，更加占用网络传输资源。

## 客户端渲染

1、客户端渲染，也叫前端渲染，兴起于前后端分离，前端通过 API 接口从后台获取数据然后自行处理。

2、相比于服务器渲染更灵活，但也不便于 SEO 。

3、减轻了服务器的压力，网络传输数据量小，前端专注于用户界面和交互，后台专注于业务逻辑和数据持久化，分工更明确。

# 服务端渲染详解

## 什么是服务端渲染

简单理解是将组件或页面通过服务器生成html字符串，再发送到浏览器，最后将静态标记"混合"为客户端上完全交互的应用程序

如下图所示，

左图页面没使用服务渲染，当请求user页面时，返回的body里为空，之后执行js将html结构注入到body里，结合css显示出来；

右图页面使用了服务端渲染，当请求user页面时，返回的body里已经有了首屏的html结构，之后结合css显示出来

## 使用SSR的利弊

### SSR的优势

1. 更利于SEO。

不同爬虫工作原理类似，只会爬取源码，不会执行网站的任何脚本（Google除外，据说Googlebot可以运行javaScript）。使用了React或者其它MVVM框架之后，页面大多数DOM元素都是在客户端根据js动态生成，可供爬虫抓取分析的内容大大减少(如图一)。另外，浏览器爬虫不会等待我们的数据完成之后再去抓取我们的页面数据。服务端渲染返回给客户端的是已经获取了异步数据并执行JavaScript脚本的最终HTML，网络爬中就可以抓取到完整页面的信息。

2. 更利于首屏渲染

首屏的渲染是node发送过来的html字符串，并不依赖于js文件了，这就会使用户更快的看到页面的内容。尤其是针对大型单页应用，打包后文件体积比较大，普通客户端渲染加载所有所需文件时间较长，首页就会有一个很长的白屏等待时间。

### SSR的局限

1. 服务端压力较大

本来是通过客户端完成渲染，现在统一到服务端node服务去做。尤其是高并发访问的情况，会大量占用服务端CPU资源；

1. 开发条件受限

在服务端渲染中，只会执行到componentDidMount之前的生命周期钩子，因此项目引用的第三方的库也不可用其它生命周期钩子，这对引用库的选择产生了很大的限制；

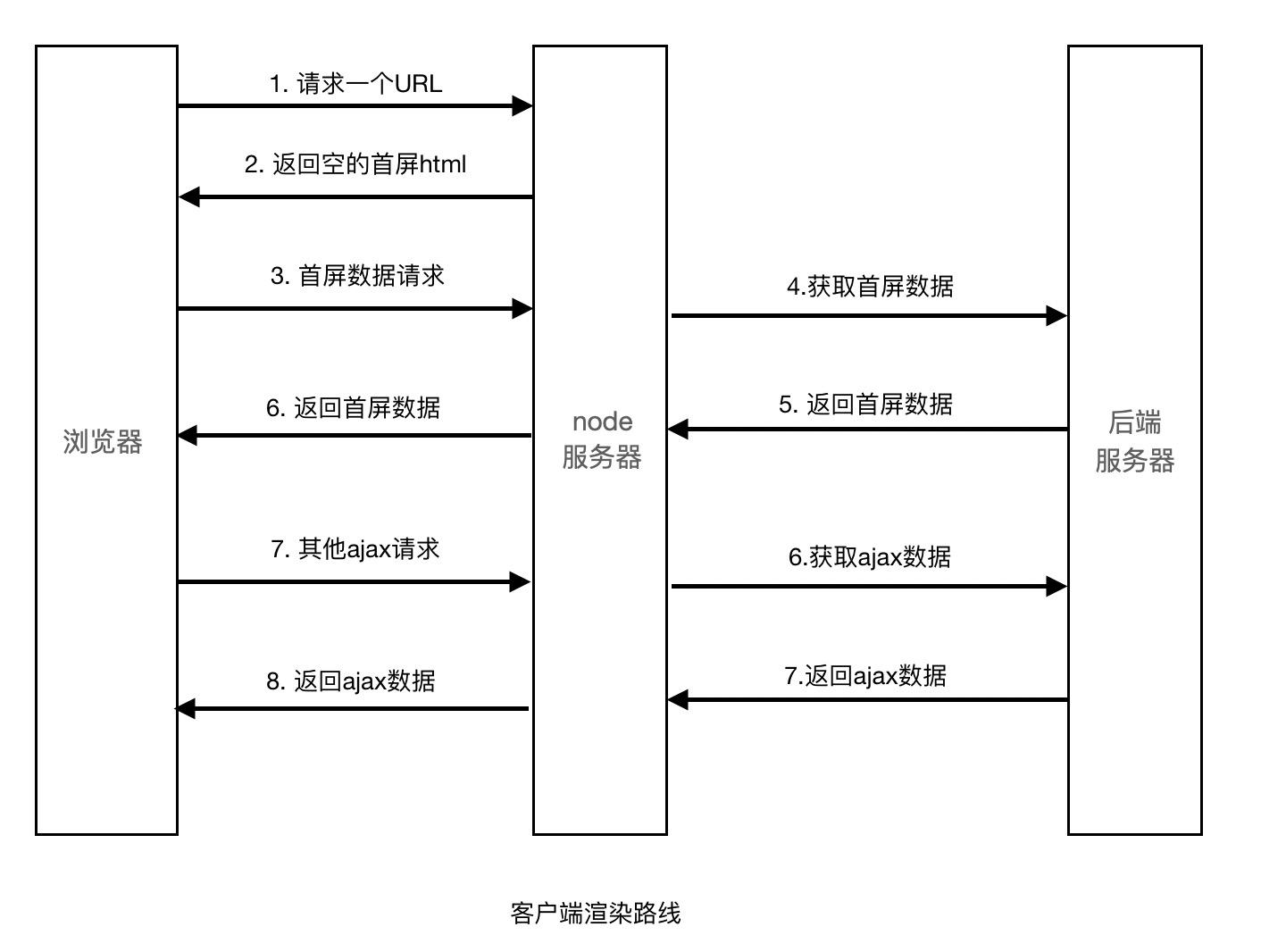
1. 学习成本相对较高

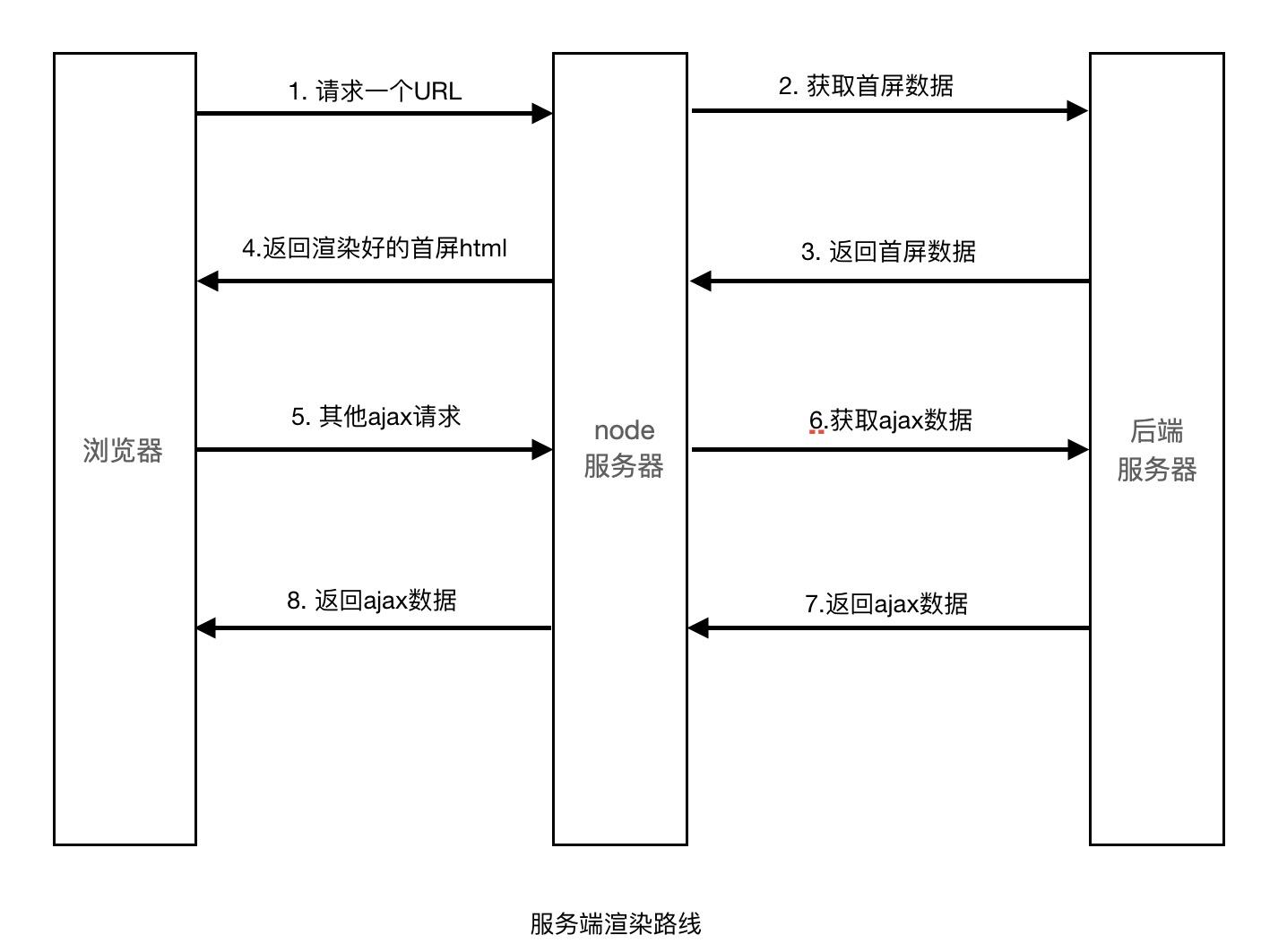
除了对webpack、React要熟悉，还需要掌握node、Koa2等相关技术。相对于客户端渲染，项目构建、部署过程更加复杂。

## 时间耗时比较

1. 数据请求

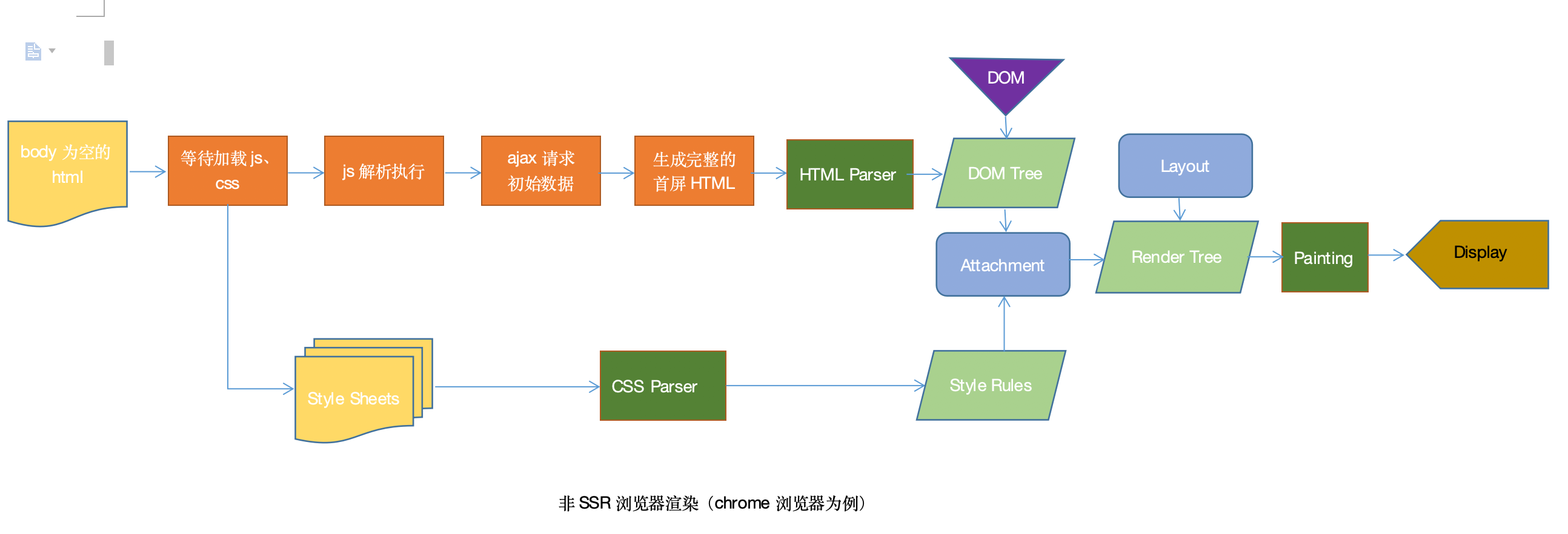
由服务端请求首屏数据，而不是客户端请求首屏数据，这是“快”的一个主要原因。服务端在内网进行请求，数据响应速度快。客户端在不同网络环境进行数据请求，且外网http请求开销大，导致时间差。 下图为服务端渲染的数据请求路线和客户端渲染的数据请求路线图

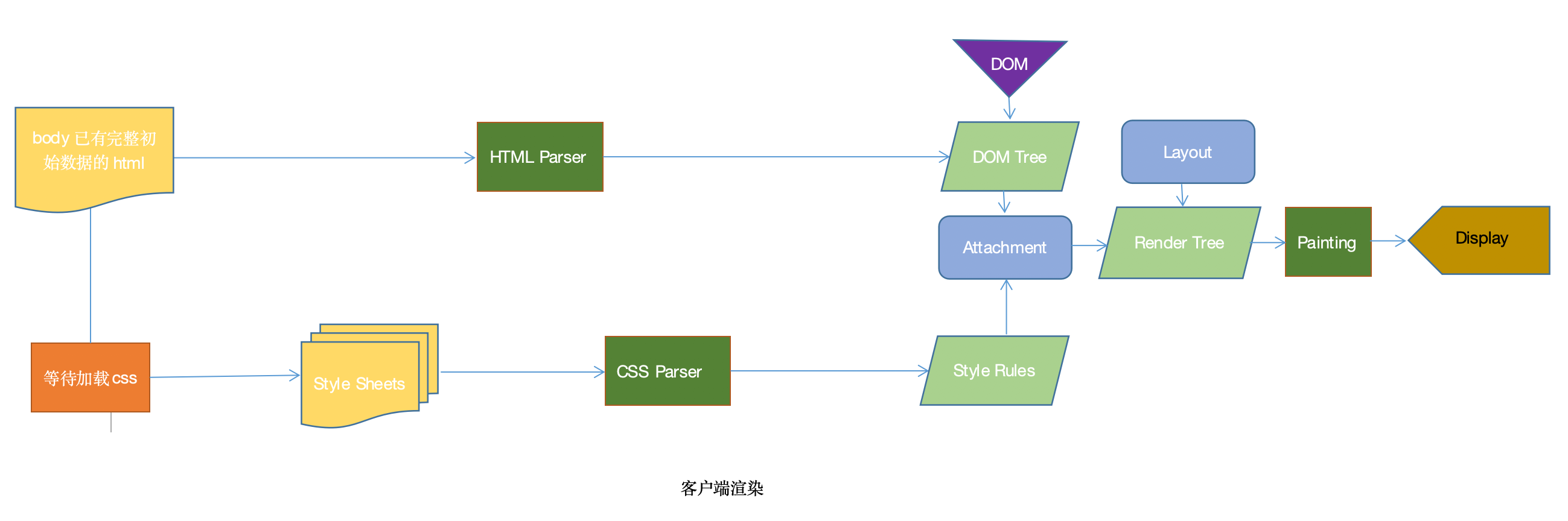




2.html渲染

服务端渲染是先向后端服务器请求数据，然后生成完整首屏html返回给浏览器；而客户端渲染是等js代码下载、加载、解析完成后再请求数据渲染，等待的过程页面是什么都没有的，就是用户看到的白屏。就是服务端渲染不需要等待js代码下载完成并请求数据，就可以返回一个已有完整数据的首屏页面。



快手内部文档请勿外传