Serverless SSR 实践

阿里巴巴·淘系技术部·水澜 (陈俊)

SSR

Server Side Rendering

服务器端渲染

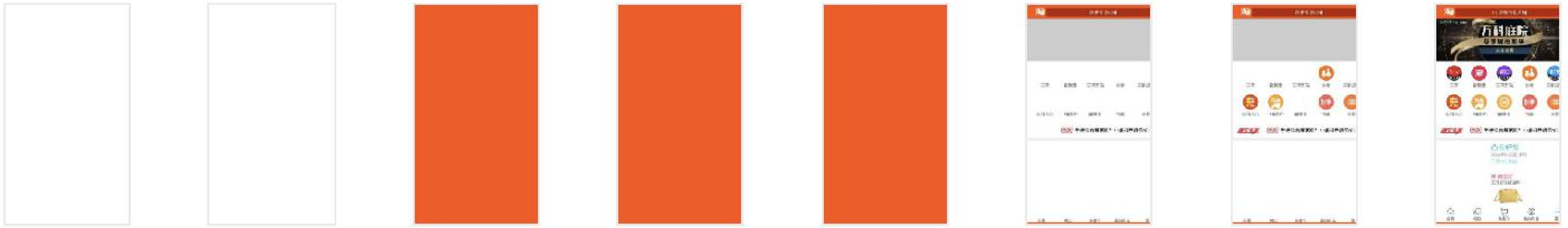
VS

CSR

Client Side Rendering

客户器端渲染

CSR



HTML | Assets | Data | Render

SSR

HTML | Assets | Hydarte

SSR 可以带来什么 ？

更快的内容达到时间

最大程度地减少网络和设备的影响

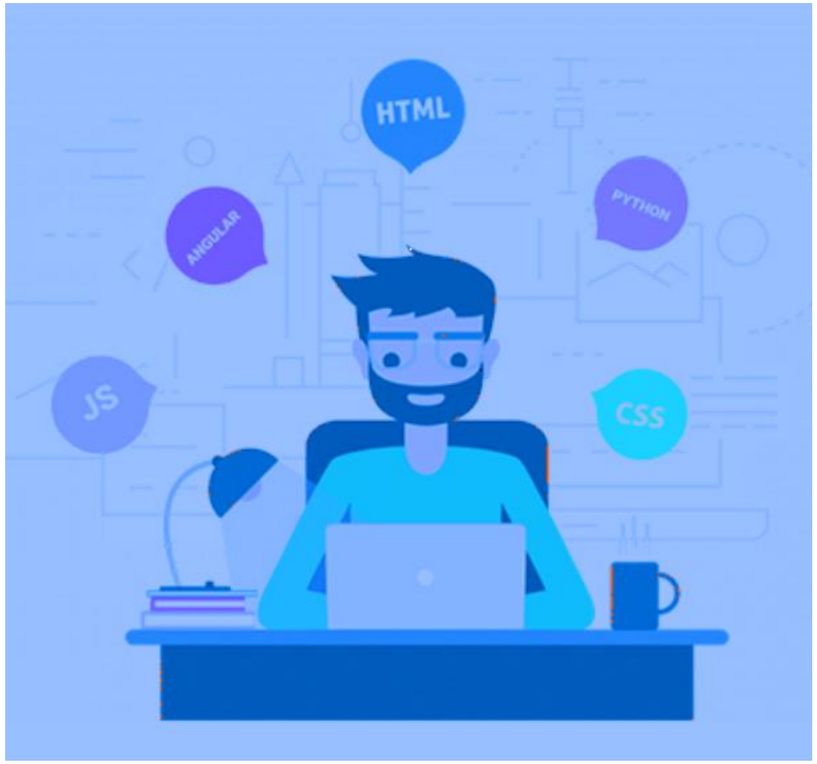
更好的 SEO

内容可以更好的被搜索引擎所理解

SSR VS CSR

但却没有成为主流的 Web 开发模式

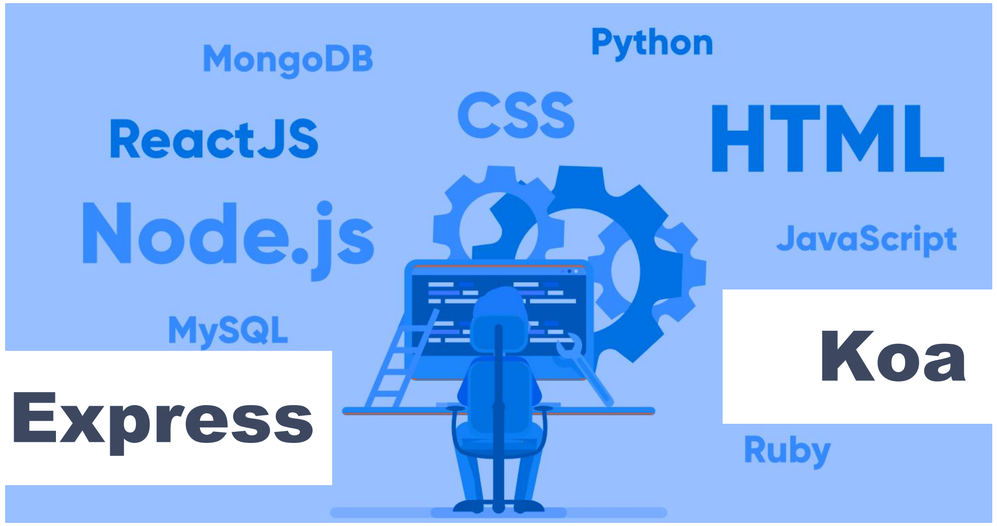
构建 SSR 应用并不容易



从前端变身全栈工程师

JS Bundle → Node 应用

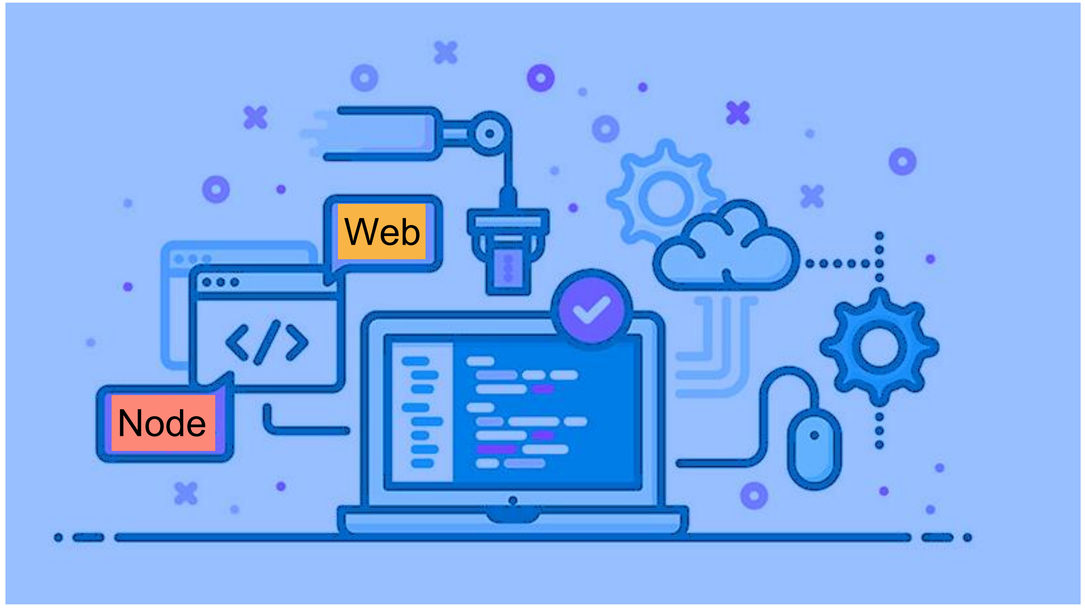
- 选型 Node 框架
- 应对性能开销
- 保障应用稳定性
- …



不再只是枚前端工程师

JS Bundle → Node 应用

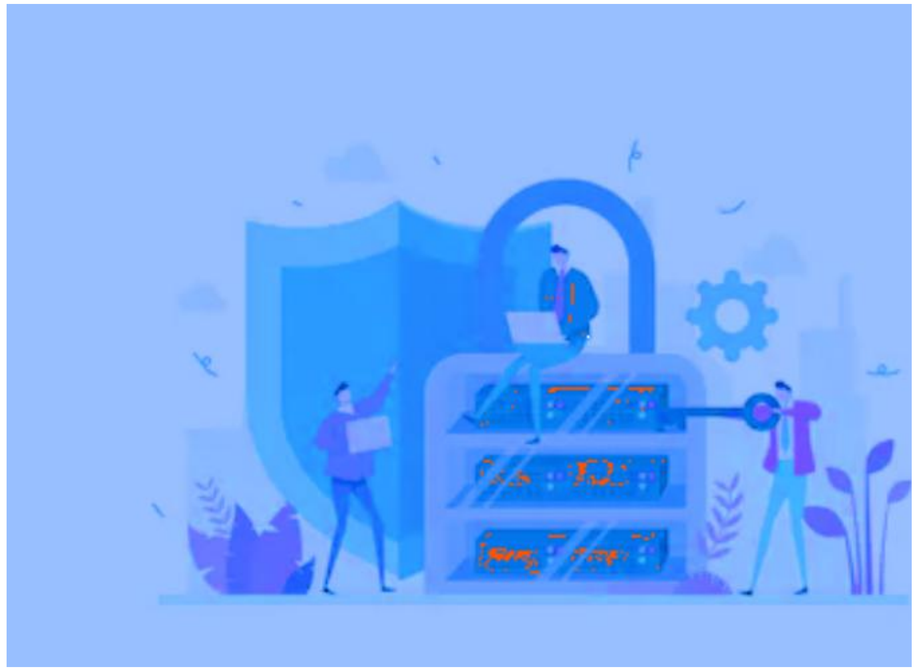
- 选型 Node 框架
- 应对性能开销
- 保障应用稳定性
- …

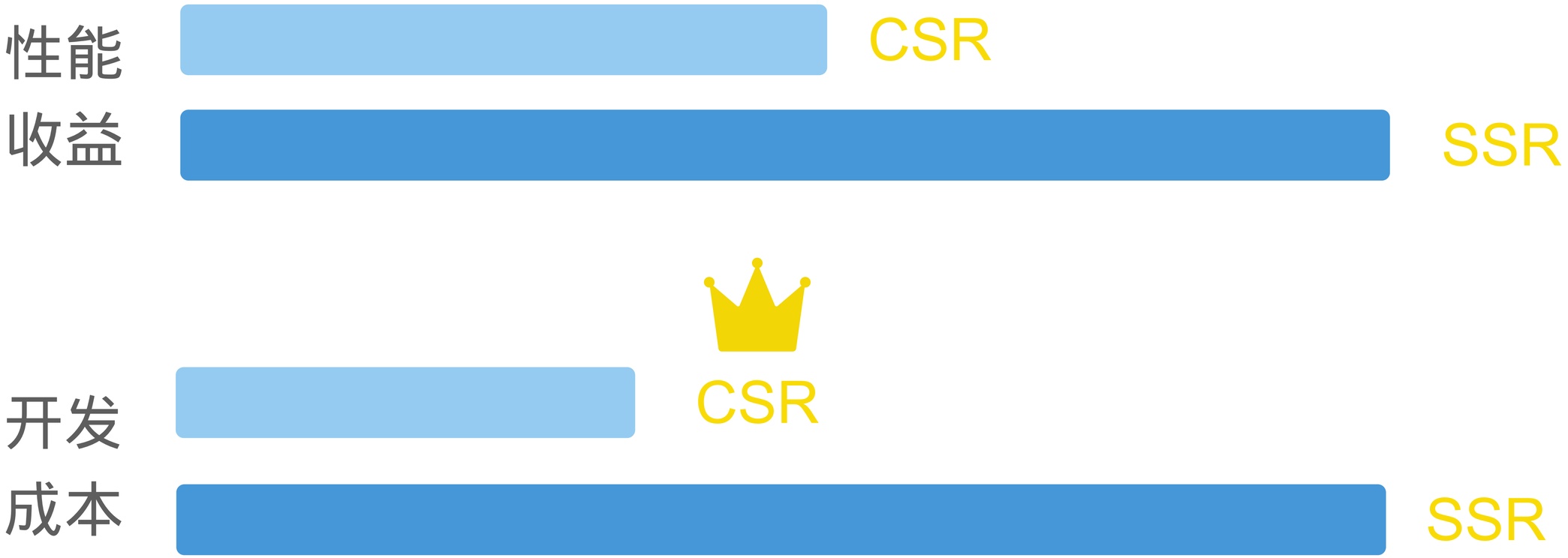


如何让一套代码两端共用

- 渲染机制的差异
- 端上环境的限制
- 如何处理数据请求
- 如何避免状态污染
- 开发调试环境的打通
- …

上线应用前还需要考虑




性能
CSR
收益
SSR
CSR
开发
成本
SSR

直到 SSR 遇上 Serverless

函数即服务（FaaS）

开发者可以更关注业务逻辑本身

天然的隔离性

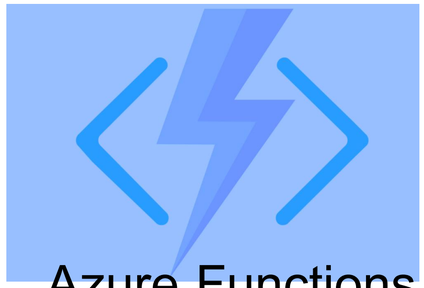
动态修复，函数间相互独立

无需运维

全托管服务、按需执行、弹性伸缩

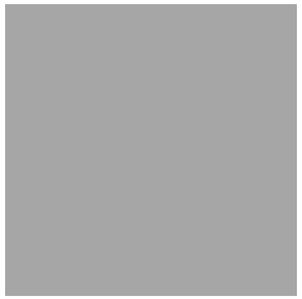
生态的不断完善

AWS Lambda



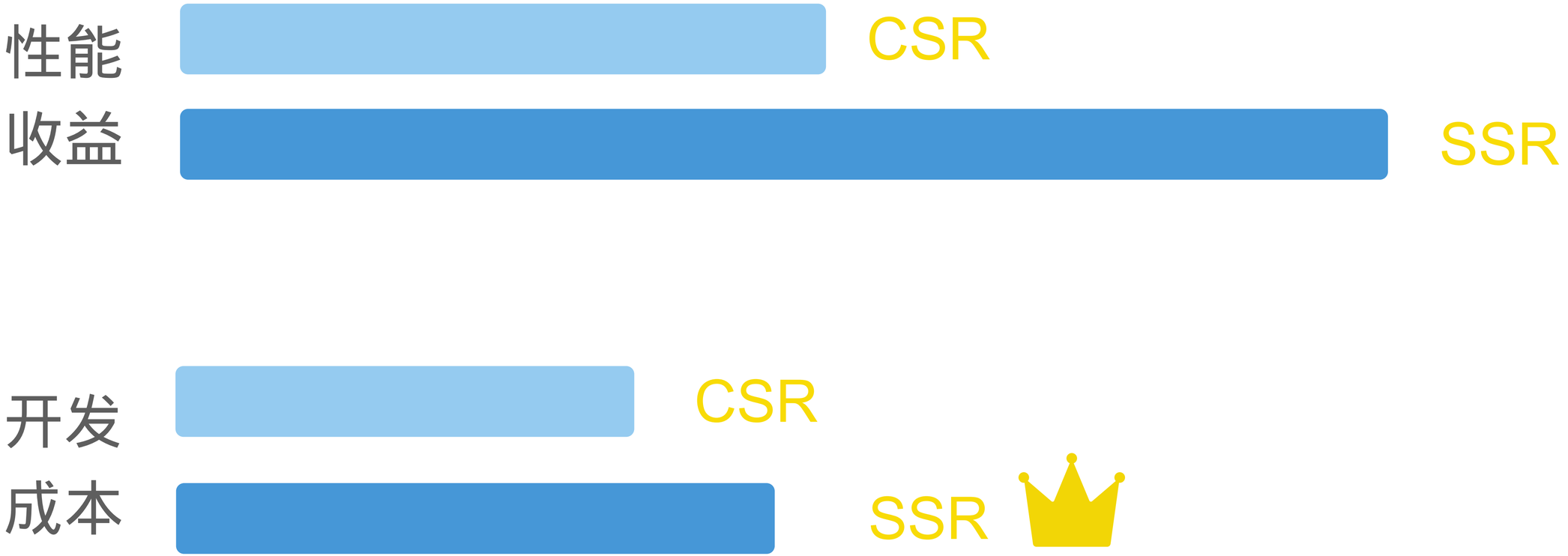
Azure Functions

阿里云 FC



Google Functions

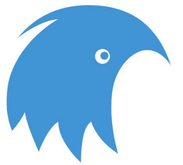
腾讯云 SCF


性能
CSR
收益
SSR
开发
CSR
成本
SSR

Rax 中的实践

Rax 是一套遵循 React 标准的跨端解决方案

```
import {render, createElement} from 'rax';

function HelloMessage(props) {
  render() {
    return (
      <div>
        Hello {props.name}
      </div>
    );
  }
}

render(
  <HelloMessage name="Taylor" />,
  document.getElementById('hello-example')
);
```

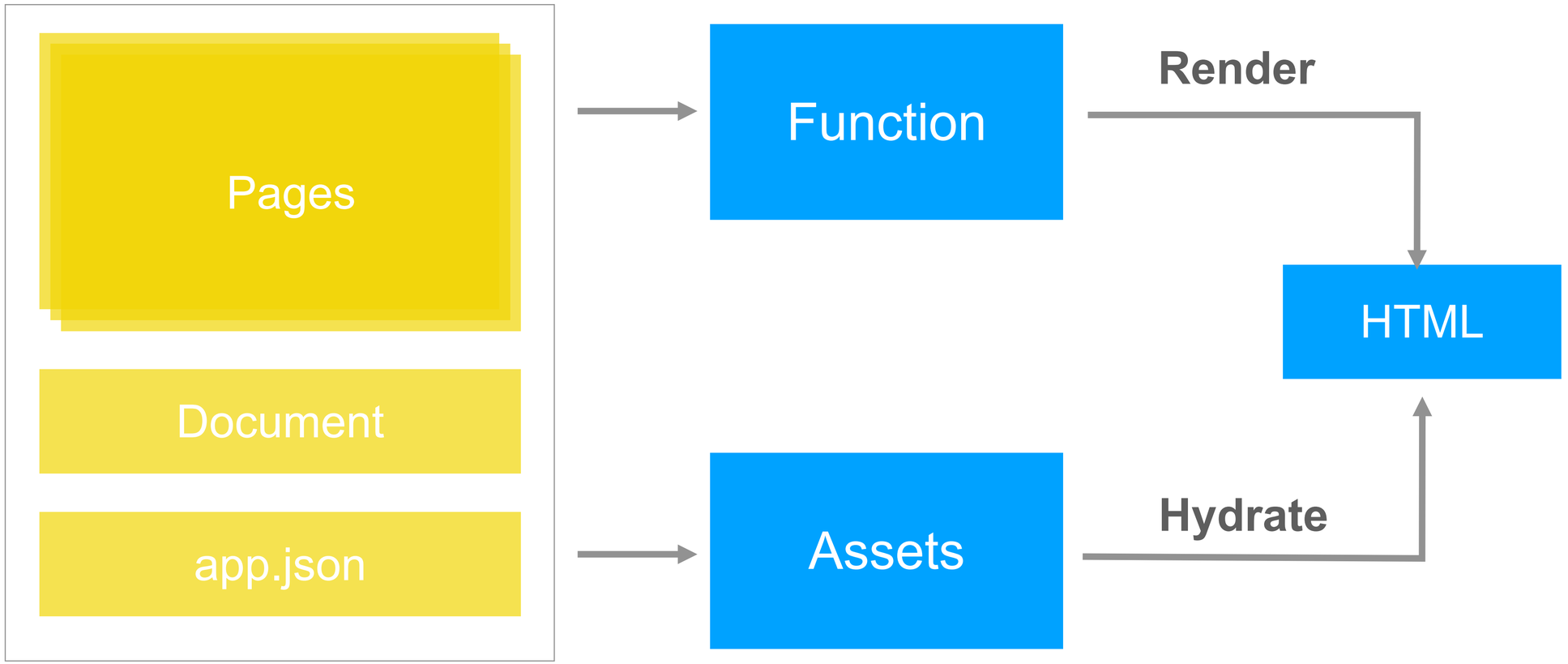



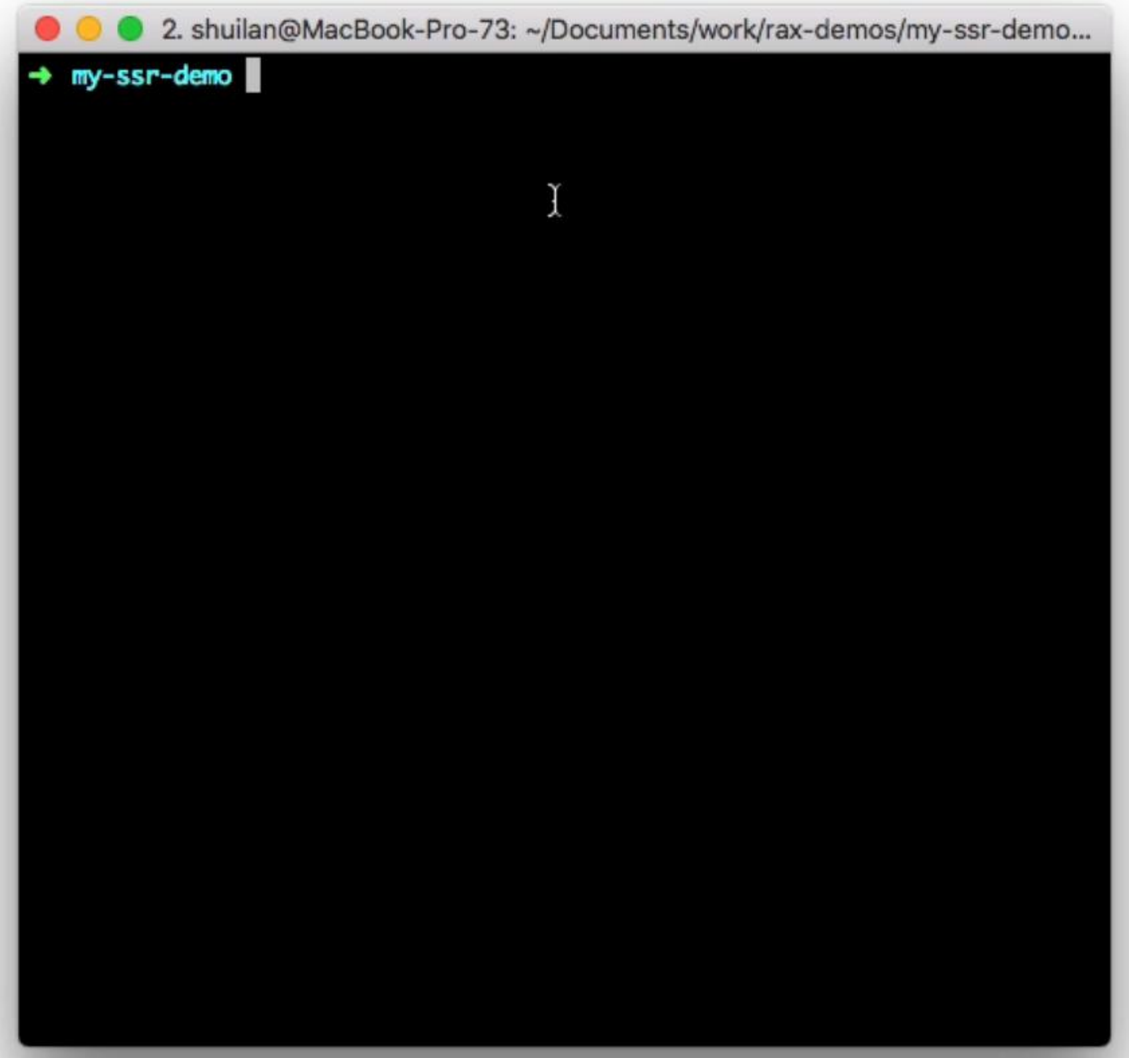
Web Weex

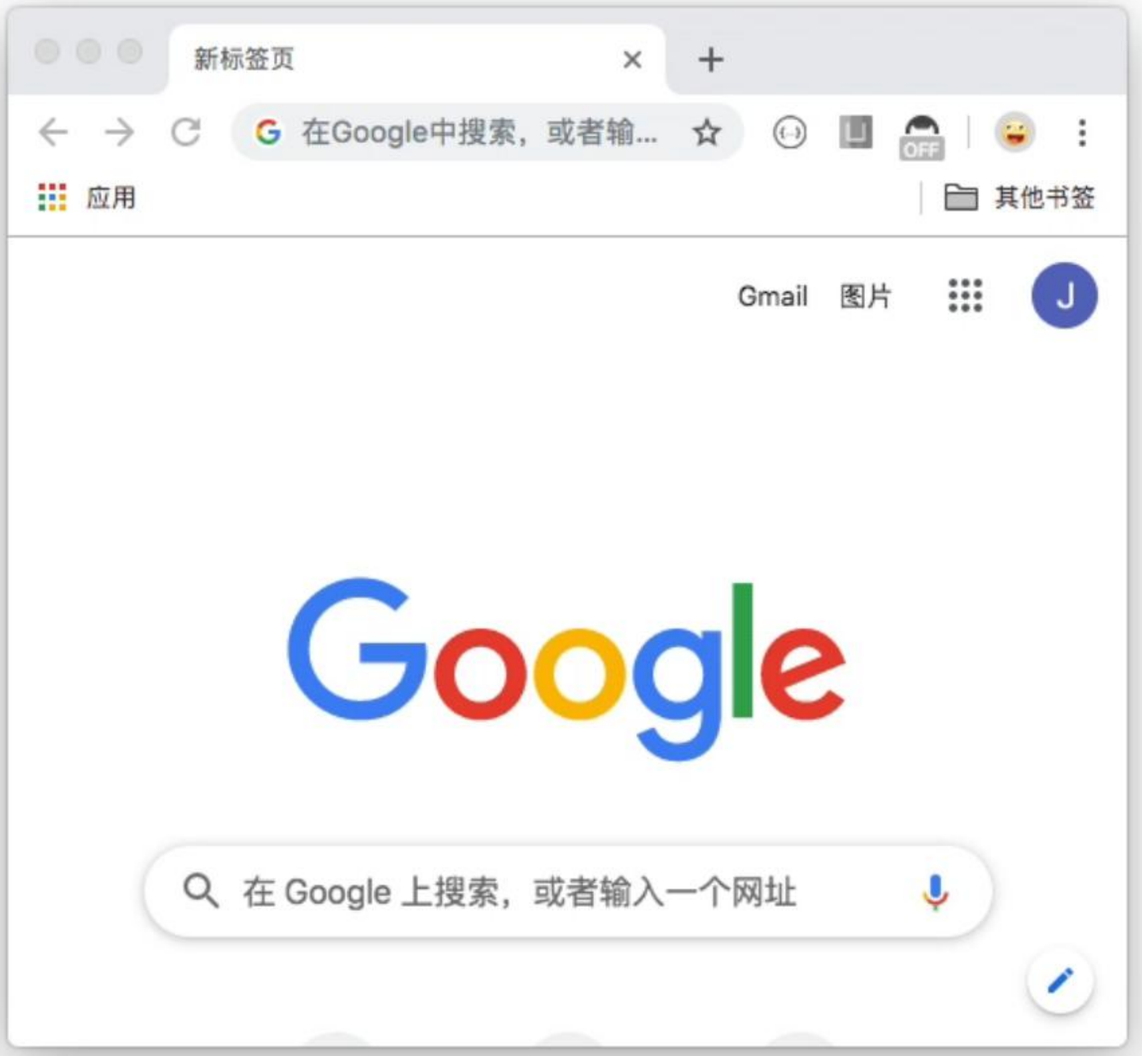


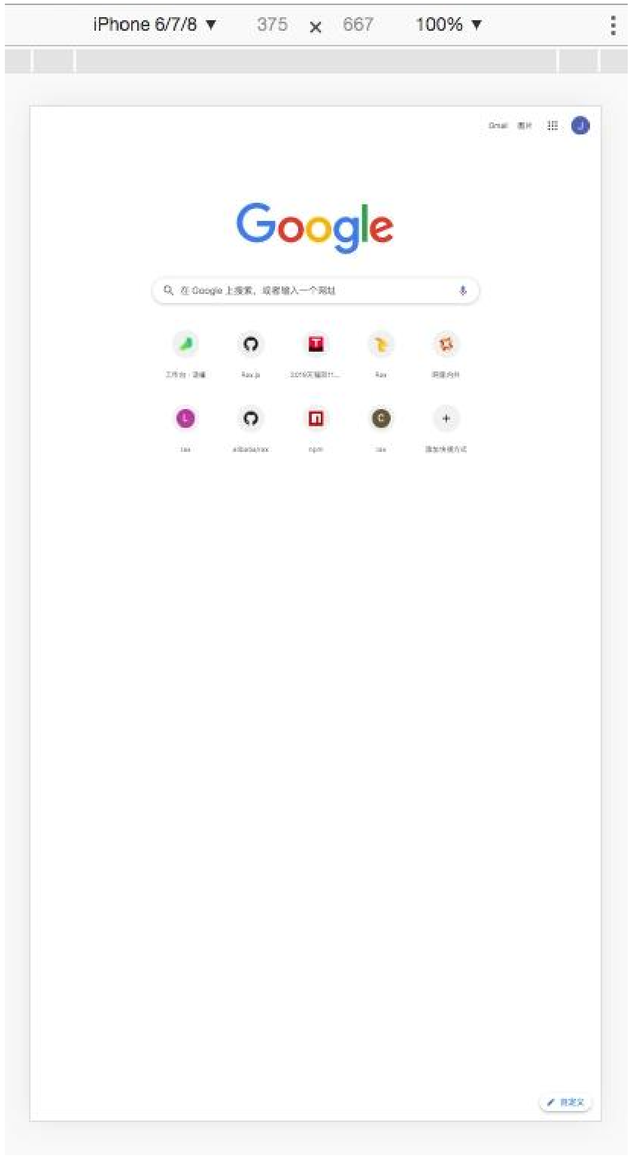
MiniApp Flutter

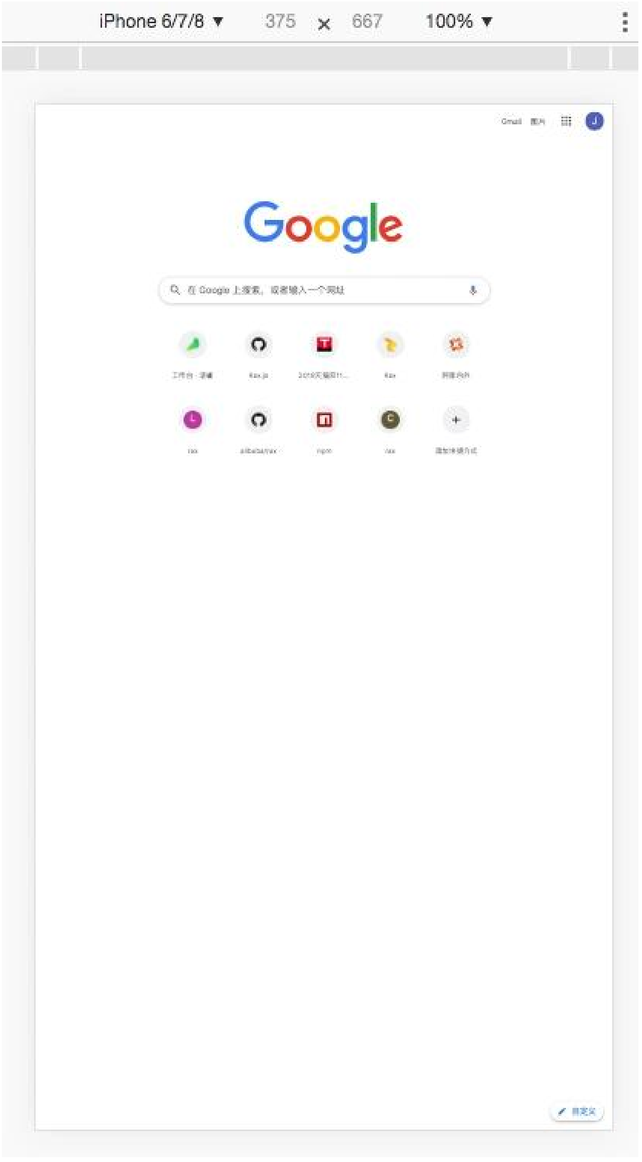
Rax SSR 工作流程


Pages
Document
app.json
Function
Assets
Render
Hydrate
HTML


2. shuilan@MacBook-Pro-73: ~/Documents/work/rax-demos/my-ssr-demo...
my-ssr-demo


新标签页
在Google中搜索，或者输...
应用
其他书签
Gmail
图片
J
Google
在 Google 上搜索，或者输入一个网址


iPhone 6/7/8 ▼ 375 × 667 100% ▼
Google


iPhone 6/7/8 ▼ 375 × 667 100% ▼
Google

组件如何完成到服务的转变

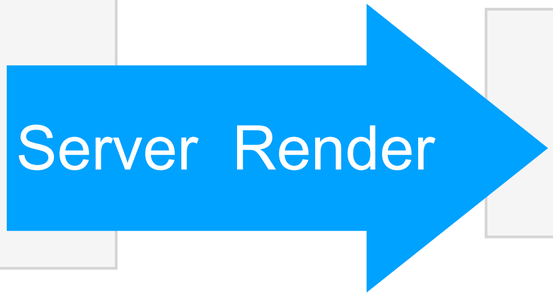
可行性分析 — SSR 工作原理

Component

```
class HelloMessage extends Component {
  render() {
    return (
      <div>
        Hello {this.props.name}
      </div>
    );
  }
}
```

Data

```
{
  data: 'rax'
}
```



HTML

```
<div>Hello Rax</div>
```

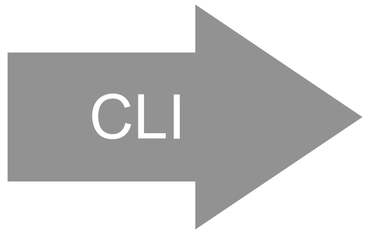
可行性分析 — FaaS 工程

```
exports.handler = (req, res) => {
  res.send('a simple function project');
};
```

```
ROSTemplateFormatVersion: '2015-09-01'
Transform: 'Aliyun::Serverless-2018-04-03'
Resources:
  simplegroup:
    Type: 'Aliyun::Serverless::Service'
    Properties:
      Description: 'simple group'
      Role: acs:ram::1647796581073291:role/aoneserverlesstestrole
    simplefunction:
      Type: 'Aliyun::Serverless::Function'
      Properties:
        Description: 'simple function'
        Initializer: index.initializer
        Handler: index.handler
        Runtime: nodejs8
        CodeUri: ./build/serverless
        Timeout: 30
        MemorySize: 1024
      Events:
        run-test:
          Type: HTTP
          Properties:
            AuthType: ANONYMOUS
            Methods: ['GET']
```

函数

描述文件



服务

可行性分析 — HTTP 触发器

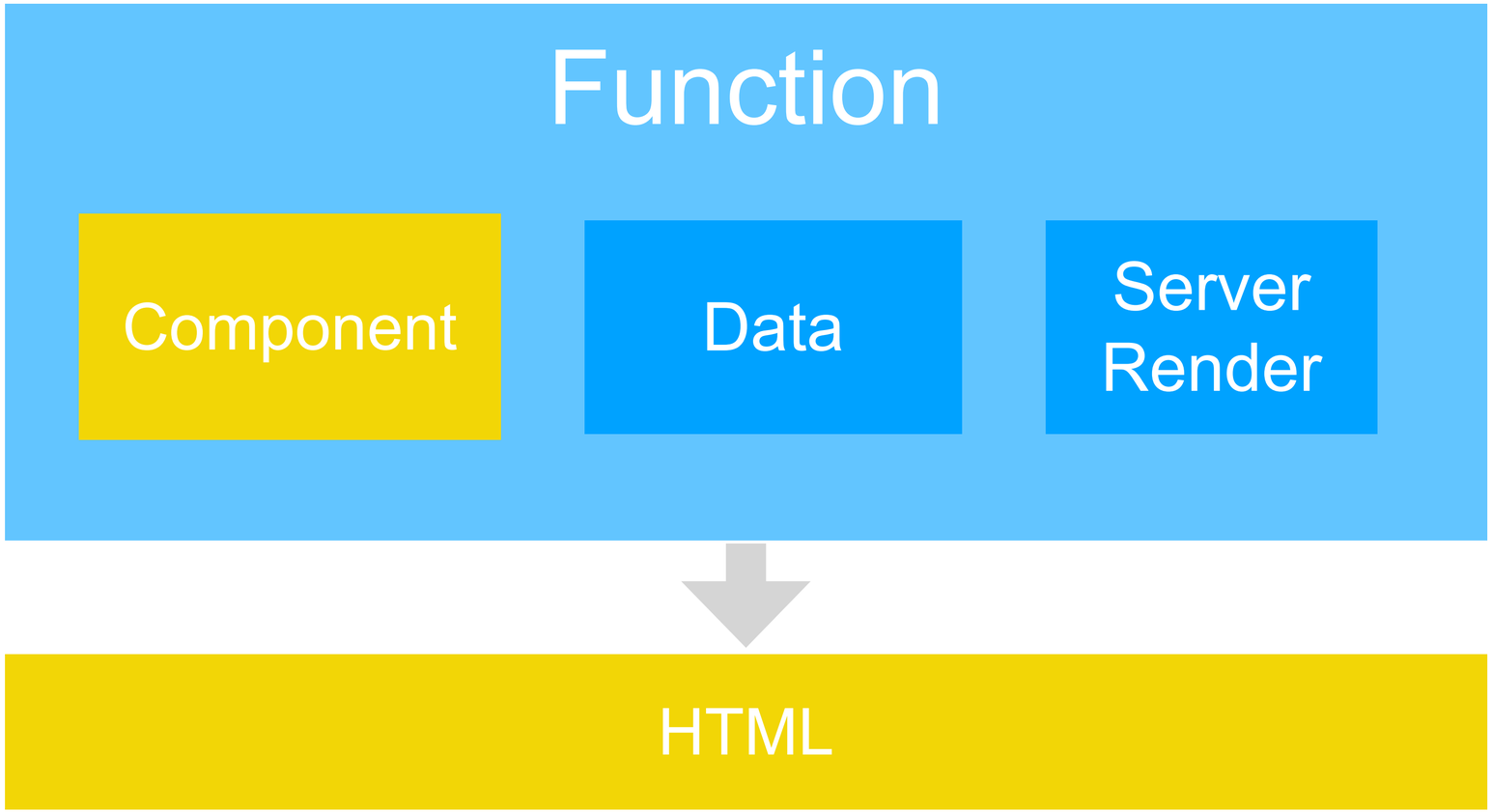
```
exports.handler = (req, res) => {
  res.send('a simple function project');
};
```

renderComponentToHTML

req： http.ClientRequest

res： http.ServerResponse

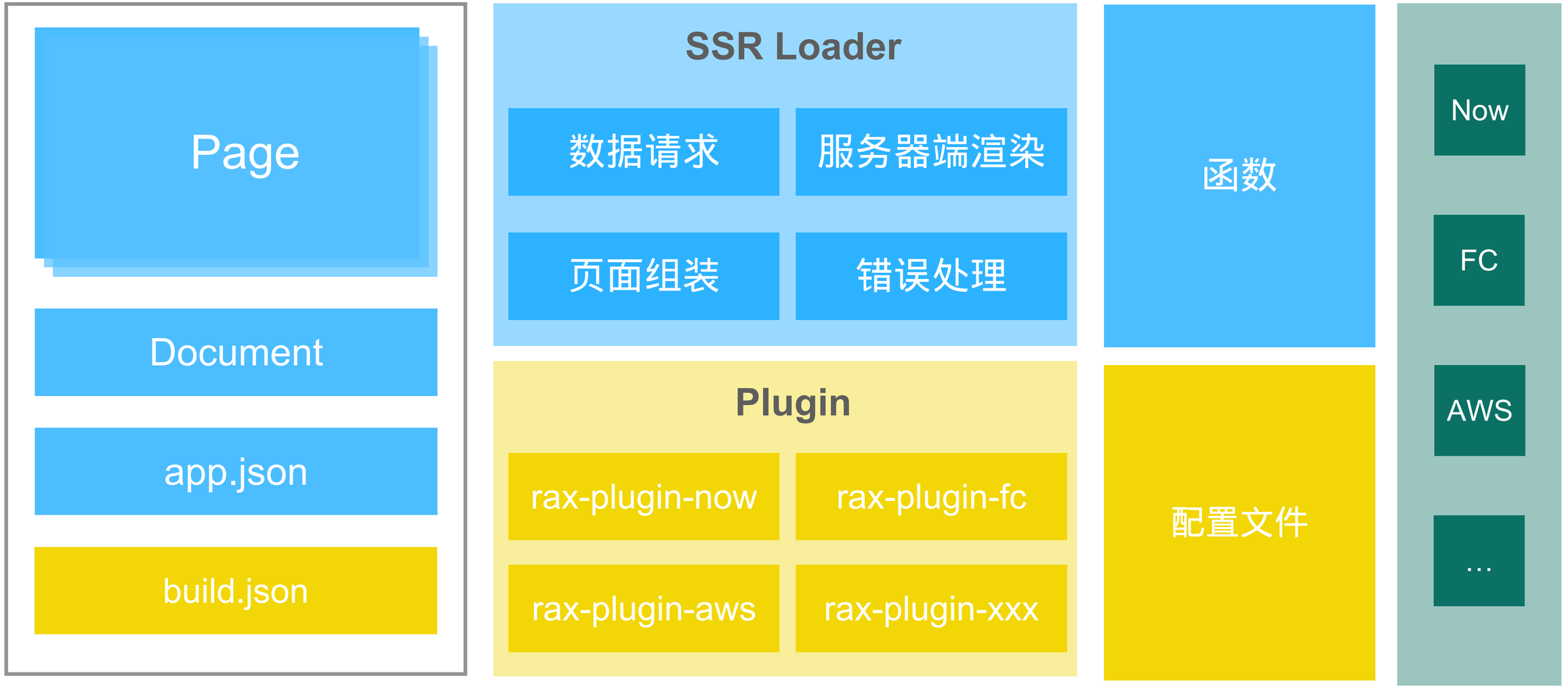
基本原理



目标

CSR 工程　SSR 工程　FaaS 工程

工程实现


Page
Document
app.json
build.json
SSR Loader
数据请求
服务器端渲染
页面组装
错误处理
Plugin
rax-plugin-now
rax-plugin-fc
rax-plugin-aws
rax-plugin-xxx
函数
配置文件
Now
FC
AWS
...

数据请求

```
import { createElement } from 'rax';

function Page({ stars }) {
  return <div>Rax stars: {stars}</div>
}

export default Page;
```

Page.getInitialProps

聚合业务逻辑，两端共用

```
Page.getInitialProps = async ({ req }) => {
  const res = await fetch('https://api.github.com/repos/alibaba/rax')
  const json = await res.json()
  return { stars: json.stargazers_count }
}
```

母板管理

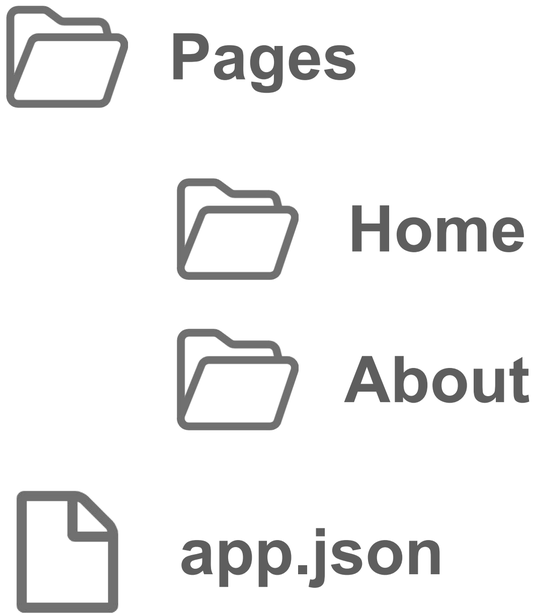
Document.jsx

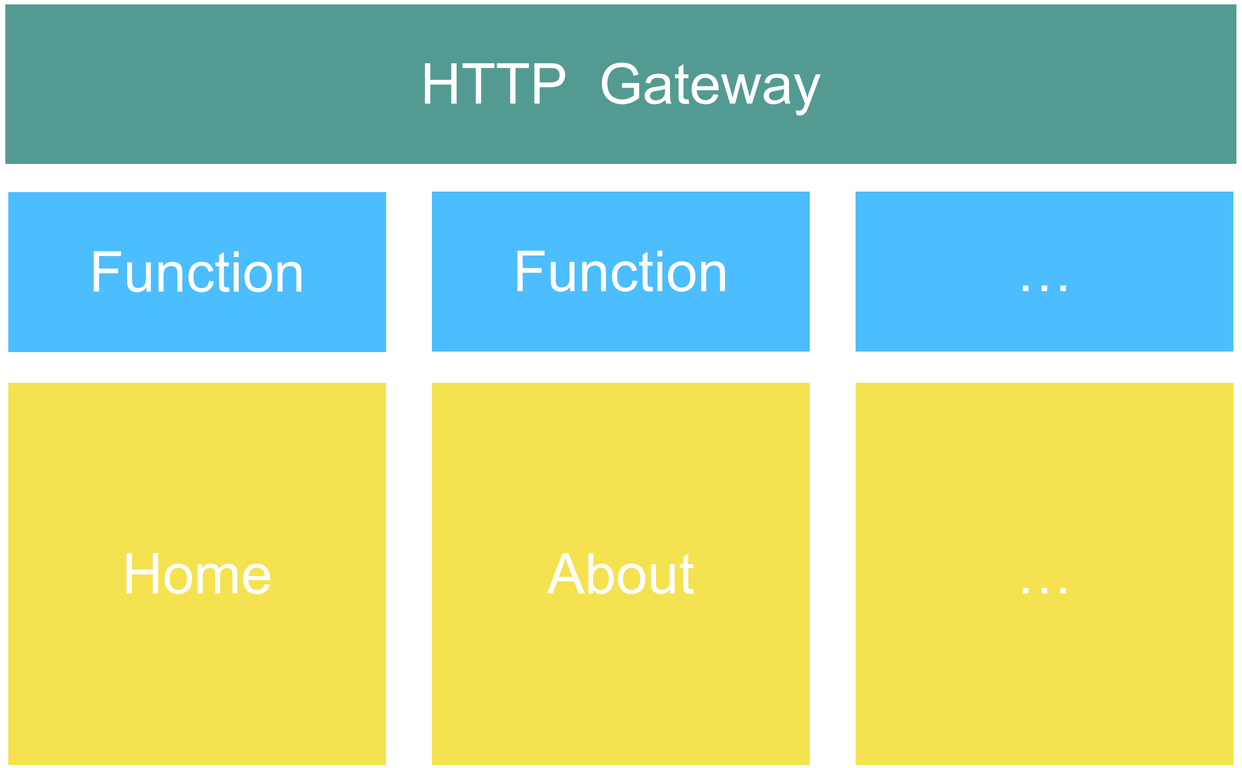
更适合 Server 端渲染，统一模板语言

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device
    <title>Rax App</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

```
import { createElement } from 'rax';

function Document(props) {
  const {
    initialHtml,
    initialData,
    styles,
    scripts,
  } = props;

  return (
    <html>
      <head>
        <meta charset="utf-8" />
        <meta name="viewport" content="width
        <title>Rax app</title>
        {styles.map((src) => <link rel="styl
      </head>
      <body>
        <div id="root" dangerouslySetInnerHT
        <script data-from="server" dangerous
        {scripts.map((src) => <script src={`
      </body>
    </html>
  );
}

export default Document;
```

路由


Pages
Home
About
app.json


HTTP Gateway
Function
Function
…
Home
About
…


ZEIT


阿里云

SSR 的持续投入

- 极致的渲染性能 6x React
- 自适应的 Hydration 模式
- SPA 下的混合渲染
- …

Thanks