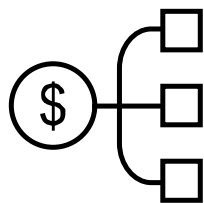




## 目录



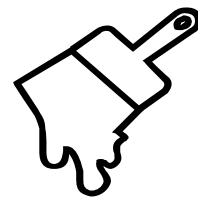
Web App生态现状



PWA价值与收益



打磨Web App体验

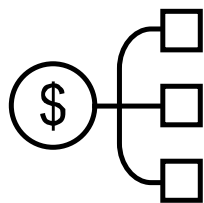


最佳实践经验

## 目录



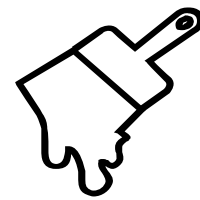
Web App生态现状



PWA价值与收益

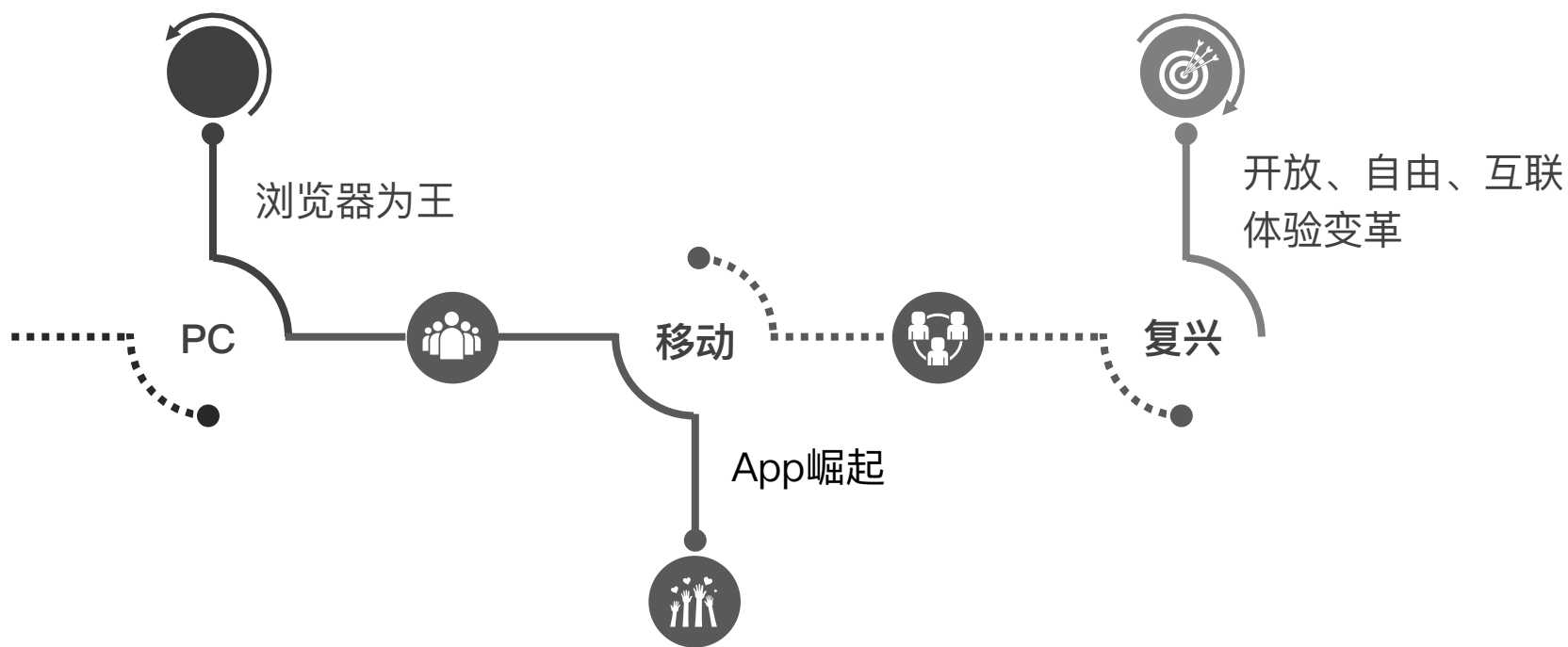


打磨Web App体验



最佳实践经验

# Web生态环境变化



## 移动Web的困境

### Web的困境

入口不再单一，桌面直达成为主流

与App有体验差距

### App的困境

90%用户打开不超过10个App

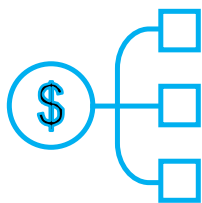
封闭，且获取路径长

80%时间消耗在3个App中

## 目录



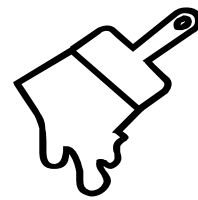
Web App生态现状



PWA价值与收益



打磨Web App体验



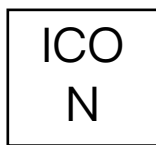
最佳实践经验

## PWA概念

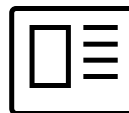
Progressive Web Apps (渐进式Web Apps) 是一种**渐进式**改善Web Apps体验的**技术理念**。



离线



桌面图标



启动动画



消息推送

*相比传统Web多出的技术特性*

## PWA关键技术

### Serviceworkers

- 通过浏览器网络代理，提供Web离线缓存能力
- 规范地址：  
<https://www.w3.org/TR/service-workers/>
- 规范状态：Working Draft

### AddToHomeScreen

- 让页面可以被添加到手机主屏，通过icon调起
- 规范地址：  
<https://www.w3.org/TR/app-manifest/>
- 规范状态：Working Draft

### Push&Notifacation

- 提供通知推送功能，增加Web站点与用户互动能力
- 规范地址：  
<https://www.w3.org/TR/push-api/>  
<https://www.w3.org/TR/notifications/>
- 规范状态：Working Draft、Recommendation



## PWA VS NA

|            | 安装方式    | 访问入口    | 通知    | 开发迭代          |
|------------|---------|---------|-------|---------------|
| PWA        | 在线、自动安装 | 图标、URL  | 可通知服务 | 无需安装<br>多平台可用 |
| Native App | 应用商店    | 图标、相关调起 | 可通知服务 | 分平台<br>需发版    |

PWA与Native App特性对比

## PWA对站点的价值

### **konga.com**

页面首次加载体积减少63%

首次交易流程过程中网络传输体积减少84%

### **flipcart.com**

Web平台成单率提升80%

### **extrastores.com**

订单量提升4倍

From Google PWA Case Study

# Why PWA&Web

体验突破，向App靠拢

- ★ 离线能力、添加到主屏、调起交互

开发模式的变革

- ★ Web开发同时考虑离线模式和体验

友好的运营方式

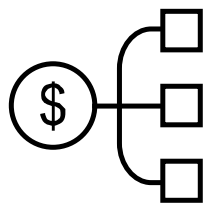
- ★ 与Web互联、url直达、不依赖应用分发



## 目录



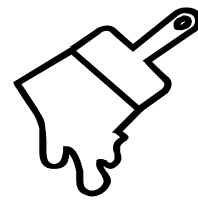
Web App生态现状



PWA价值与收益



打磨Web App体验



最佳实践经验

## 架构设计新思路



离线化


视图框架

异步渲染

交互感

核心思想：赋予Web App类App的体验

## 架构设计新思路



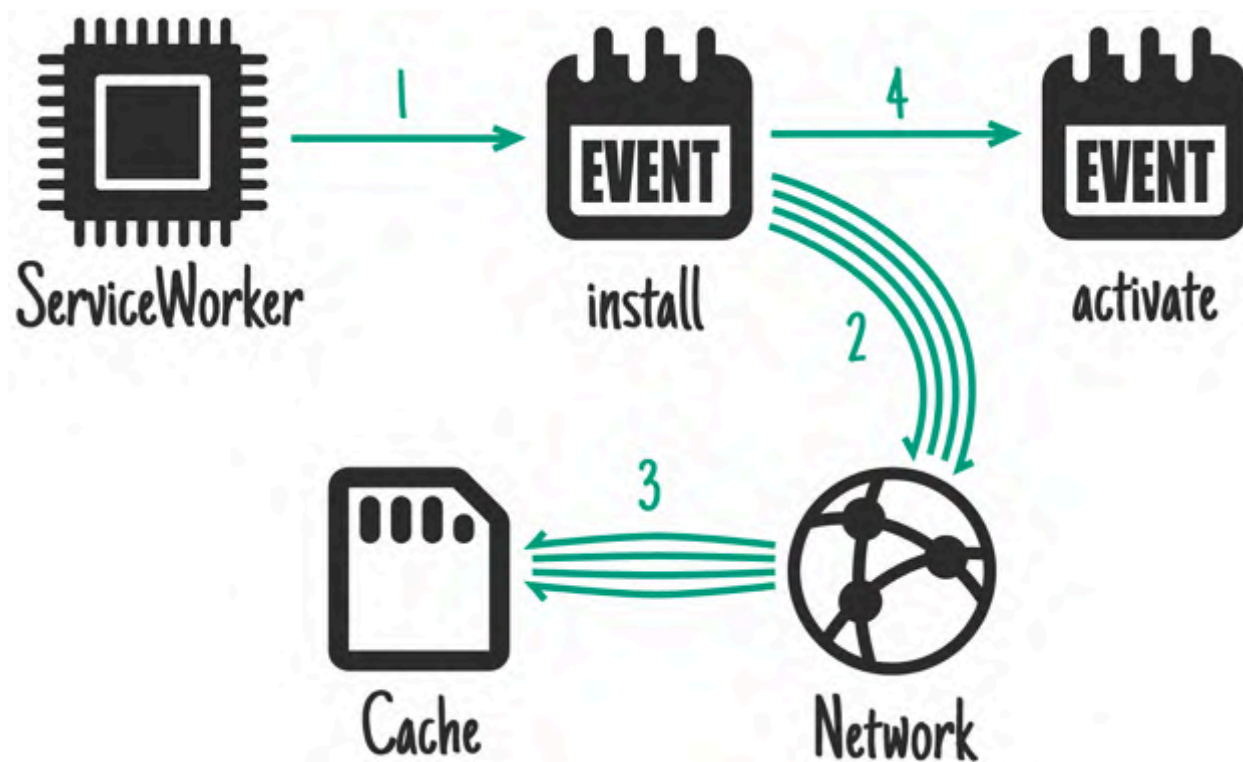
离线化

视图框架

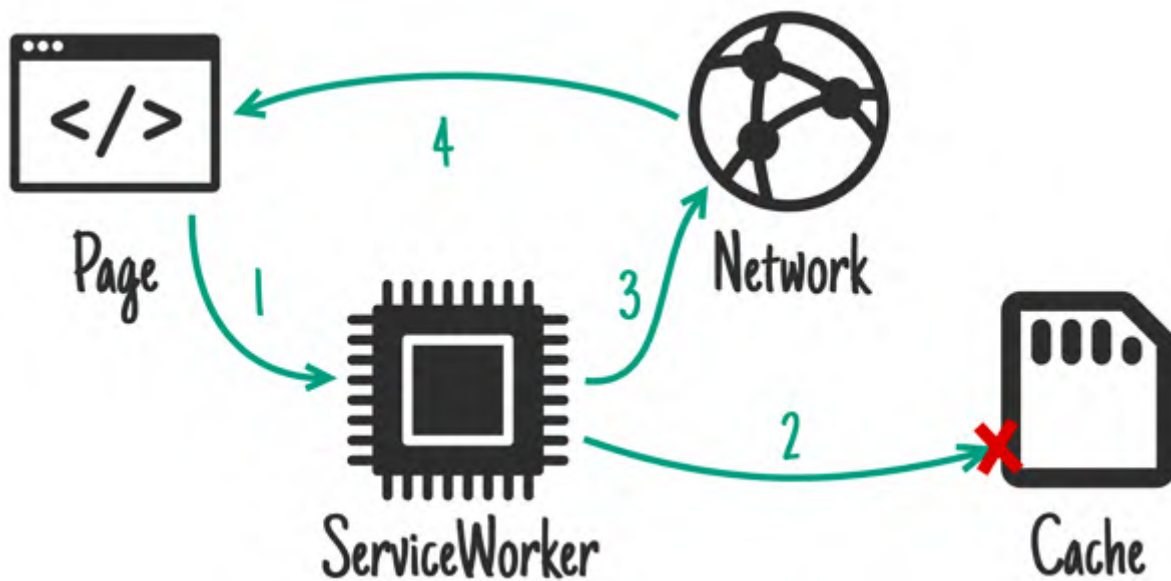
异步渲染

交互感

## 离线基础技术-Serviceworker初始化

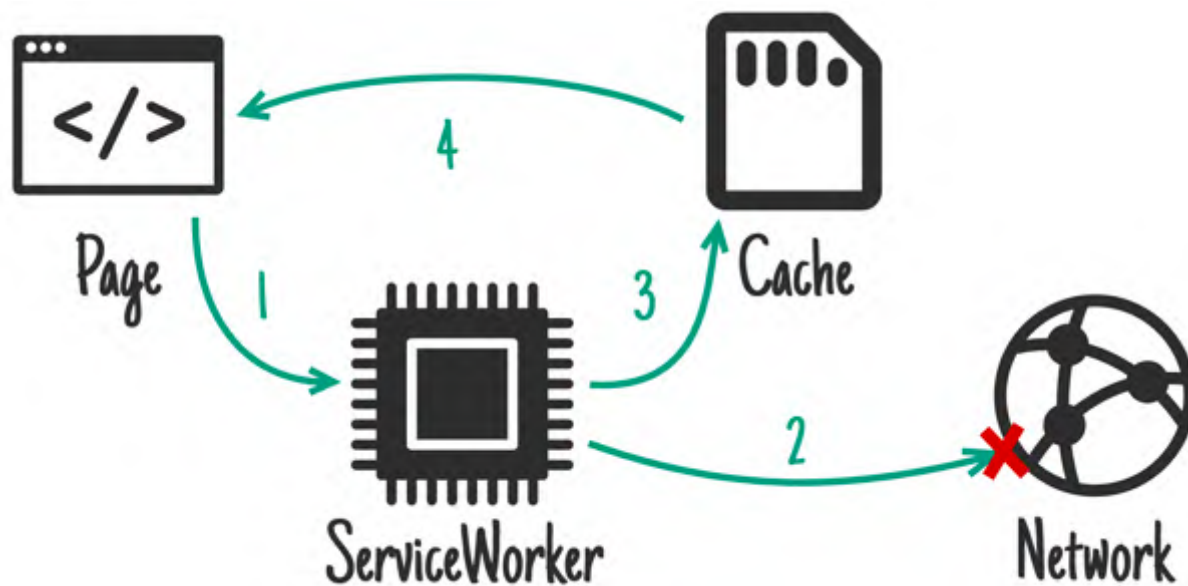


## 离线基础技术-缓存优先





## 离线基础技术-网络优先



## 离线缓存策略方案

- ✓ 缓存 key 对应于版本号
- ✓ 获取资源（样式、脚本、背景图片）时离线优先（Cache First）
- ✓ 获取数据（天气信息、地理位置）时网络优先（Network First）
- ✓ 所有网络请求都会被更新到缓存

## 架构设计新思路



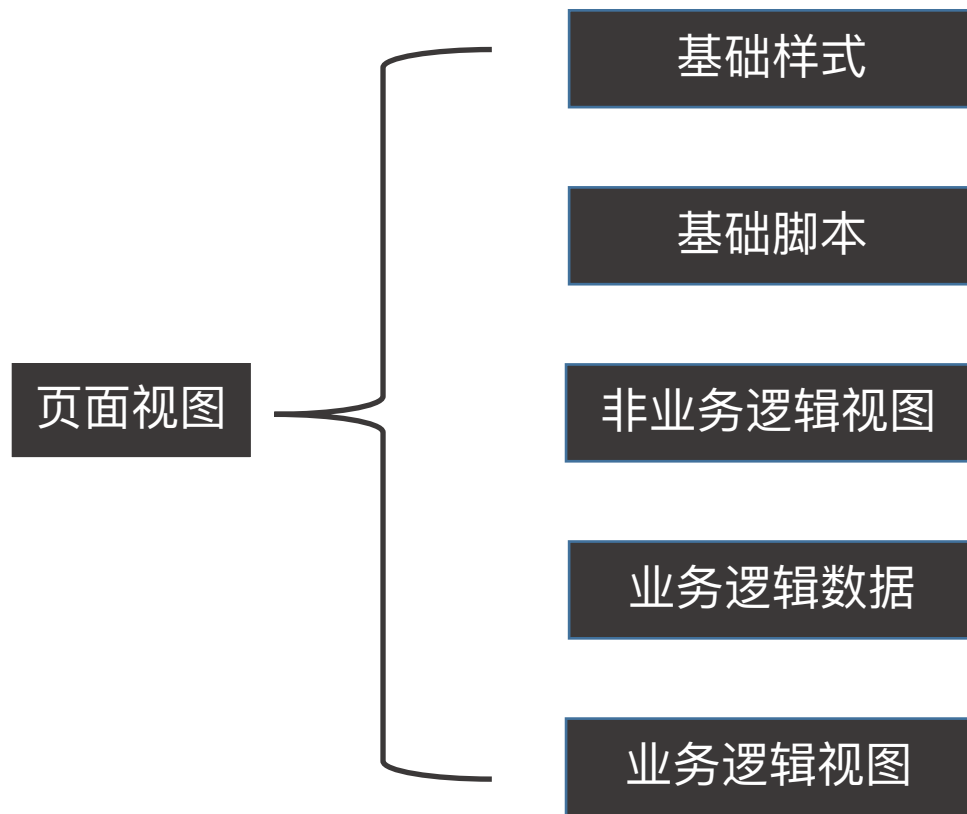
离线化

视图框架

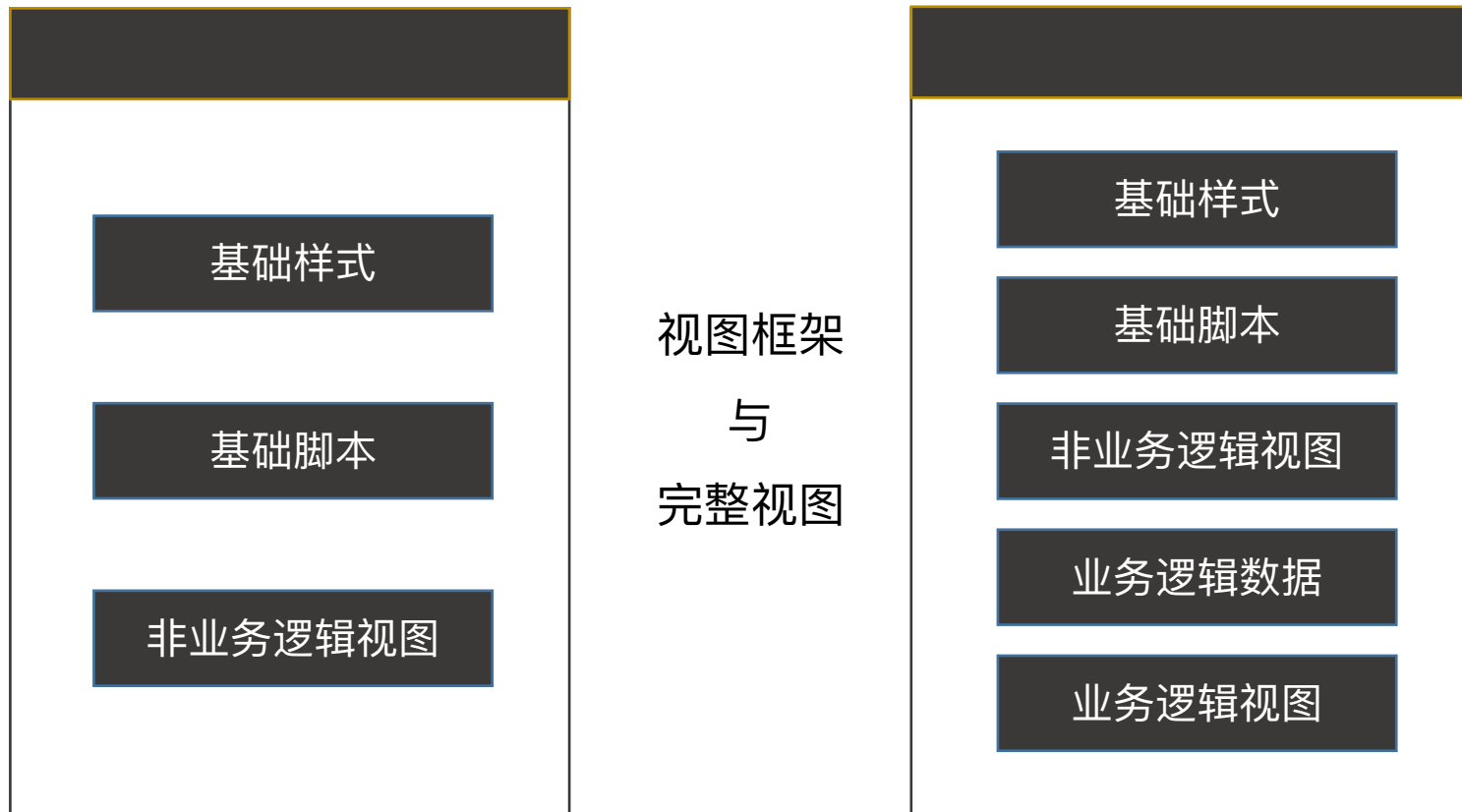
异步渲染

交互感

## 页面视图组成



## 页面视图分类



## 构建视图框架方案

- ✓ 分离出视图框架
  - ✓ 基础样式、基础脚本、基础视图
- ✓ 视图框架采用离线优先策略
  - ✓ 保证每次访问（无论是否离线）都能给用户良好反馈

## 架构设计新思路



离线化

视图框架

异步渲染

交互感

## 异步渲染的概念和目的

### ✓ 概念

异步渲染是分离视图框架与业务逻辑的一种有效手段

### ✓ 目的

使业务逻辑视图和数据能独立于框架运行加载

使页面能良好交互



## 异步渲染方案选型

- ✓ 方案一：字符串拼接+异步结构化数据
  - ✓ 缺点：无法承载复杂视图业务
- ✓ 方案二：前端模板+异步结构化数据
  - ✓ 优点：管理轻量级视图
  - ✓ 缺点：无法承载数据和视图逻辑复杂业务
- ✓ 方案三：基于MVVM的组件方案（数据绑定与组件化）
  - ✓ 优点：能承载复杂业务，且易于管理
  - ✓ 缺点：量级较重，没必要应用于简单业务

## 架构设计新思路



离线化

视图框架

异步渲染

交互感

让页面有交互感是体验提升的重要一步

## 效果对比



普通跳转

高下立判!

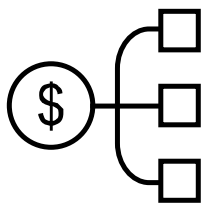


交互跳转

## 目录



Web App生态现状



PWA价值与收益

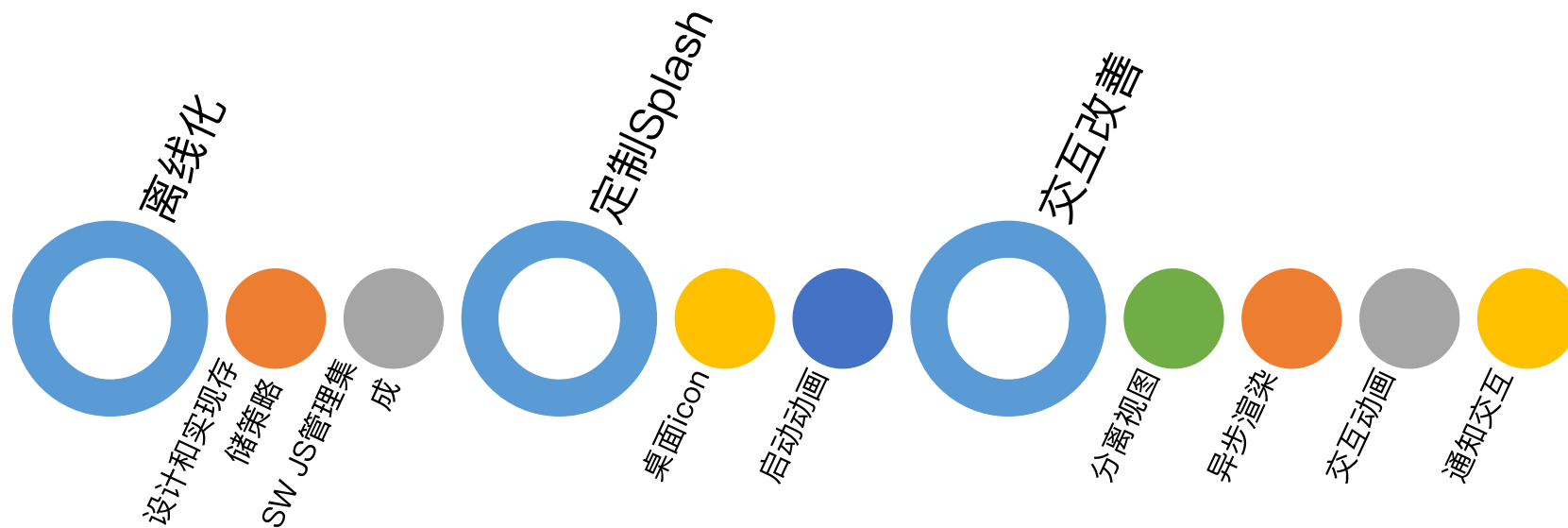


打磨Web App体验



最佳实践经验

# 渐进式Web Apps实现步骤



## 注册&部署ServiceWorkers

```
if ('serviceWorker' in navigator) {  
  navigator.serviceWorker  
    .register('./sw.bundle.js', { scope: '/' })  
    .then(reg => console.log(`Service worker registered to ${reg.scope}`))  
    .catch(err => console.log(`Registration failed with ${err}`));  
}
```

### 注册SW

- ✓ 要求：根目录访问
- ✓ 缓存机制：变更时更新or天级更新（浏览器行为）
- ✓ 建议：文件名不带hash版本，eg: [pwa.example.com/sw.js](http://pwa.example.com/sw.js)

### SW文件部署

## 添加主屏

Chrome 提示的条件:

[increasing engagement in chrome](#)

- 注册有 service worker
- 添加 manifest 文件
- 5分中内浏览两次以上





## 配置Splash

Manifest 配置启动屏幕：

```
<link rel="manifest" href="manifest.json">
```

- short\_name 桌面图标文字。
- name 启动屏幕文字。
- start\_url 从桌面启动的页面。
- icons 桌面和启动图标（建议包含 144x144 大小的 PNG）。

下午3:21



百度天气

## 视图框架与异步渲染

```
<!DOCTYPE html>
  <head>
    <meta charset=UTF-8>
    <title>天气</title>
    <meta name=viewport content="width=device-width,minimum-scale=1,maximum-scale=1,user-scalable=no">
    <meta name=msapplication-TileImage content=/images/icons/icon-144x144-3e2a23dc.png>
    <meta name=msapplication-TileColor content=#2F3BA2> <link rel=apple-touch-icon href=/images/icons/icon-152x152-158c03ca.png>
    <link rel=manifest href=manifest.json>
  </head>
  <body>
    <div id=app></div>
    <script src=./index.bundle.js></script>
  </body>
</html>
```

### 百度天气视图框架代码

- ✓ 异步渲染：推荐直接使用类似vue的组件库
- ✓ 开源库：Vue for PWA、React for PWA、Angular for PWA

## Web核心价值与机遇

去中心化

避免信息孤岛

万物互联

App很好，Web更好

App里面是长不出亚马孙森林的，但是互联网上可以长出亚马孙森林，Web上也可以。

—from 《在线》

# Thanks!

## Q&A