

# 腾讯文档 PWA 实践

Tencent Docs PWA Practice

彭伟宏

[harrypeng@tencent.com](mailto:harrypeng@tencent.com)

## 分享大纲

当我们讨论 PWA 时，我们在讨论些什么

PWA 是如何在腾讯文档业务中落地的

腾讯文档做了哪些 PWA 实践，收效如何

分享大纲

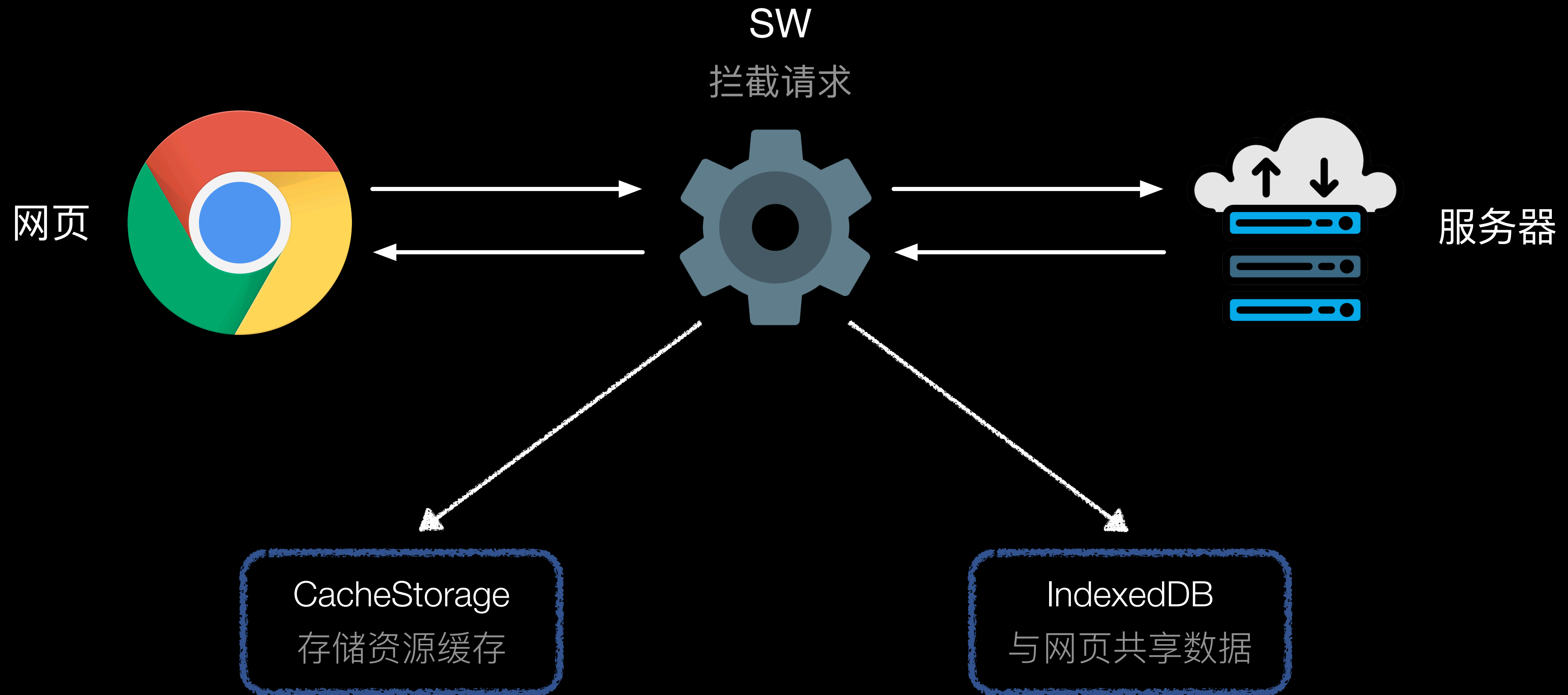
当我们讨论 PWA 时，我们在讨论些什么

PWA 是如何在腾讯文档业务中落地的

腾讯文档做了哪些 PWA 实践，收效如何

## 当我们讨论 PWA 时，我们在讨论些什么

Service Worker 是独立于网页运行的后台线程，**核心功能是拦截和处理网络请求**，通过对缓存的管理来处理请求的响应。



当我们讨论 PWA 时，我们在讨论些什么

一切源于请求

拦截作用域下一切请求，接管并处理请求

# 当我们讨论 PWA 时，我们在讨论些什么

## 一切源于请求

拦截作用域下一切请求，接管并处理请求

## 缓存控制能力

拥有完整的缓存控制能力，读写不受限

## 当我们讨论 PWA 时，我们在讨论些什么

### 一切源于请求

拦截作用域下一切请求，接管并处理请求

### 缓存控制能力

拥有完整的缓存控制能力，读写不受限

### 一套代码，处处执行

天然的浏览器运行环境，抹除各平台差异

## 当我们讨论 PWA 时，我们在讨论些什么

### 一切源于请求

拦截作用域下一切请求，接管并处理请求

### 缓存控制能力

拥有完整的缓存控制能力，读写不受限

### 一套代码，处处执行

天然的浏览器运行环境，抹除各平台差异

### 安全有保障

基于 https 协议，用得放心



## 当我们讨论 PWA 时，我们在讨论些什么

### 一切源于请求

拦截作用域下一切请求，接管并处理请求

### 缓存控制能力

拥有完整的缓存控制能力，读写不受限

### 一套代码，处处执行

天然的浏览器运行环境，抹除各平台差异

### 安全有保障

基于 https 协议，用得放心

### 用户体验增强

无网/弱网体验增强，网页向应用程序化发展

## 当我们讨论 PWA 时，我们在讨论些什么

### 一切源于请求

拦截作用域下一切请求，接管并处理请求

### 缓存控制能力

拥有完整的缓存控制能力，读写不受限

### 一套代码，处处执行

天然的浏览器运行环境，抹除各平台差异

### 安全有保障

基于 https 协议，用得放心

### 用户体验增强

无网/弱网体验增强，网页向应用程序化发展

### 对前端开发者友好

生态完善，无语言成本，使用 JS 即可编写

## 分享大纲

当我们讨论 PWA 时，我们在讨论些什么

PWA 是如何在腾讯文档业务中落地的

腾讯文档做了哪些 PWA 实践，收效如何

分享大纲

当我们讨论 PWA 时，我们在讨论些什么

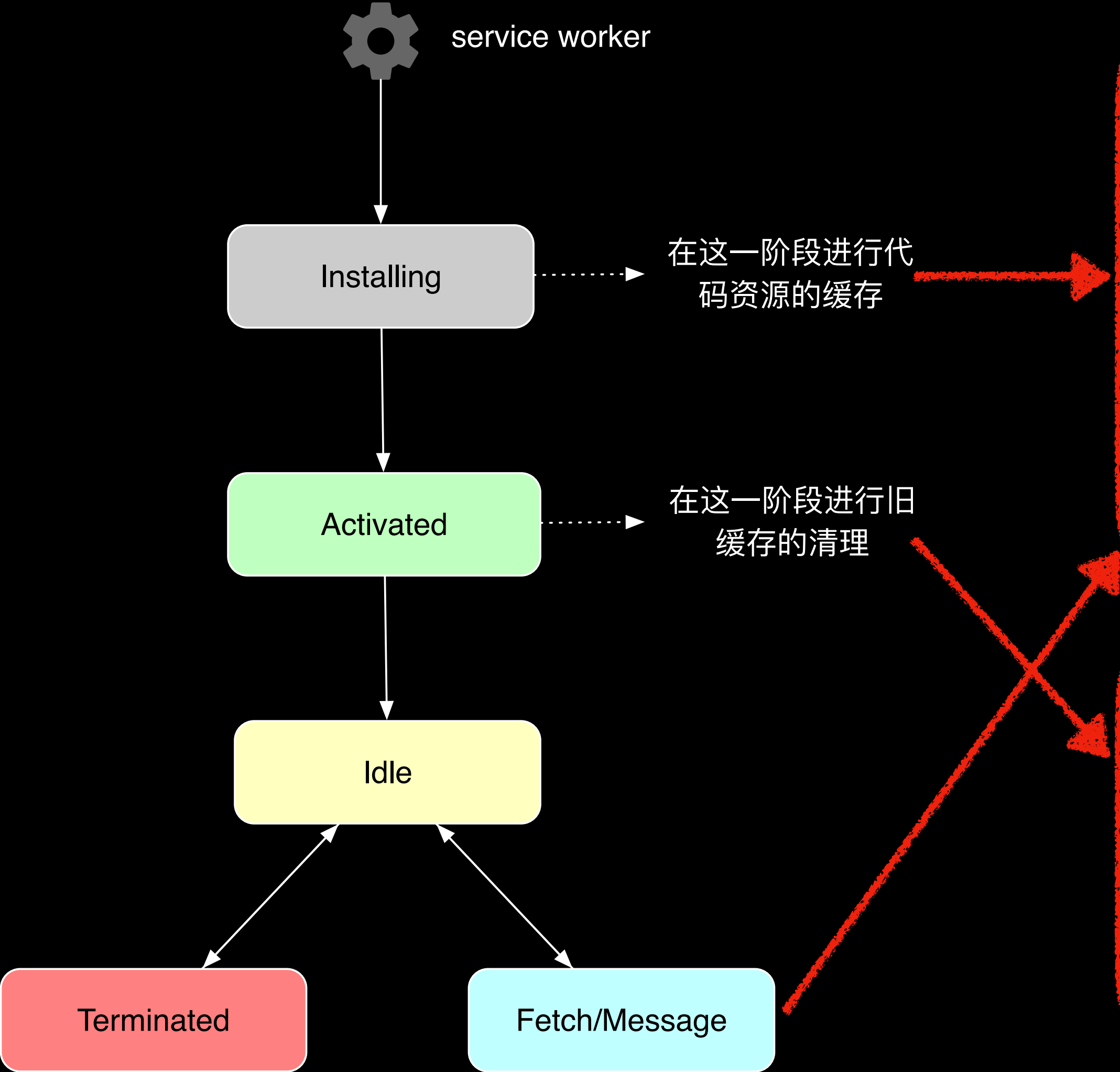
PWA 是如何在腾讯文档业务中落地的

腾讯文档做了哪些 PWA 实践，收效如何

PWA 是如何在腾讯文档业务中落地的

💡 一个简单网站的 PWA 缓存接入流程

# PWA 是如何在腾讯文档业务中落地的



```
const PRECACHE = 'precache-v1';
const RUNTIME = 'runtime';

// A list of local resources we always want to be cached.
const PRECACHE_URLS = [
  'index.html',
  './', // Alias for index.html
  'styles.css',
  '../../styles/main.css',
  'demo.js'
];

// The install handler takes care of precaching the resources we always need.
self.addEventListener('install', event => {
  event.waitUntil(
    caches.open(PRECACHE)
      .then(cache => cache.addAll(PRECACHE_URLS))
      .then(self.skipWaiting())
  );
});

// The activate handler takes care of cleaning up old caches.
self.addEventListener('activate', event => {
  const currentCaches = [PRECACHE, RUNTIME];
  event.waitUntil(
    caches.keys().then(cacheNames => {
      return cacheNames.filter(cacheName => !currentCaches.includes(cacheName));
    }).then(cachesToDelete => {
      return Promise.all(cachesToDelete.map(cacheToDelete => {
        return caches.delete(cacheToDelete);
      }));
    }).then(() => self.clients.claim())
  );
});
```

PWA 是如何在腾讯文档业务中落地的

🤔 腾讯文档也可以这么容易就接入吗?

## 业务特性

在线办公套件，多品类共存

分品类开发、构建、部署

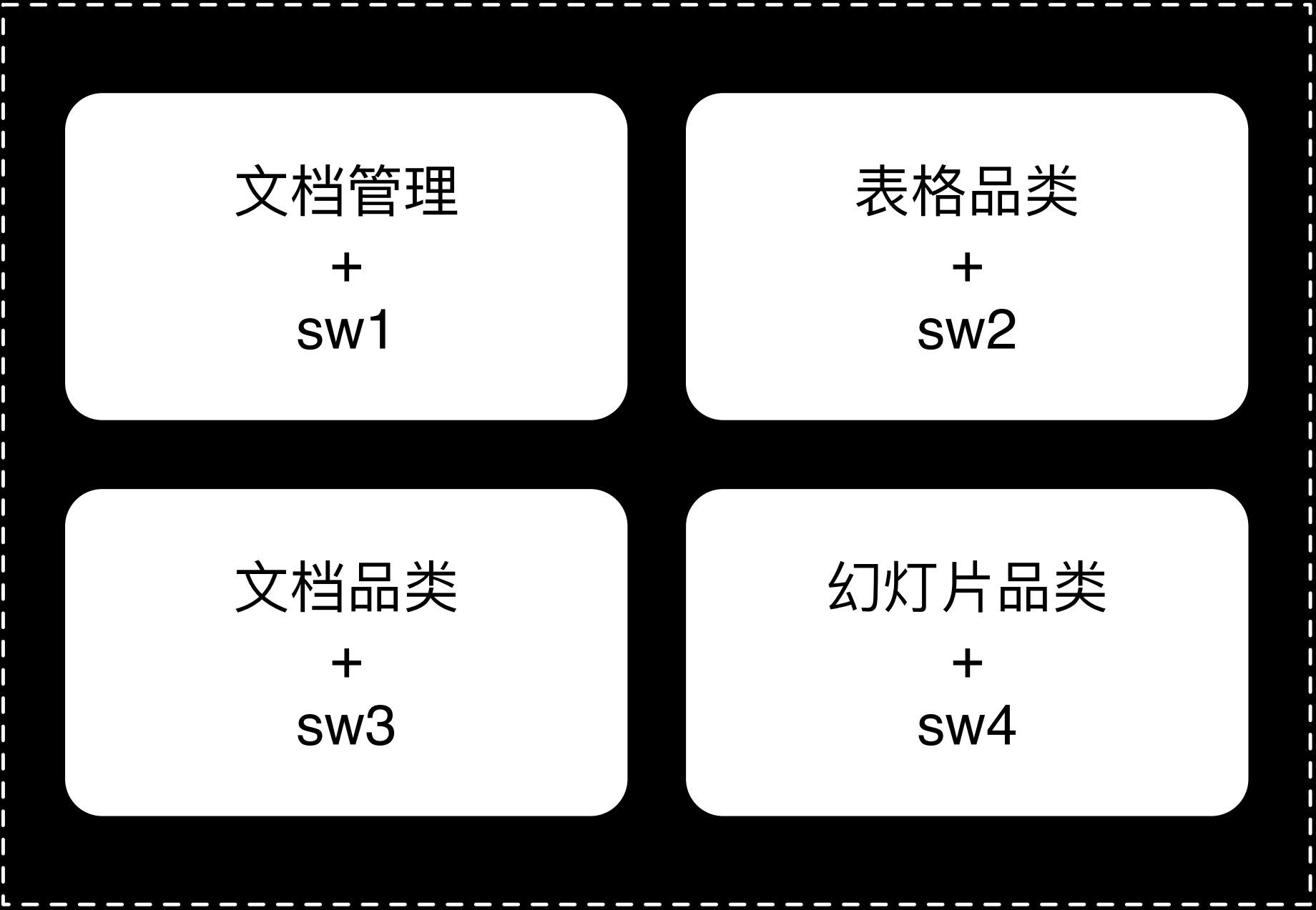
多页应用，按路径划分品类入口



# SW 实例管理

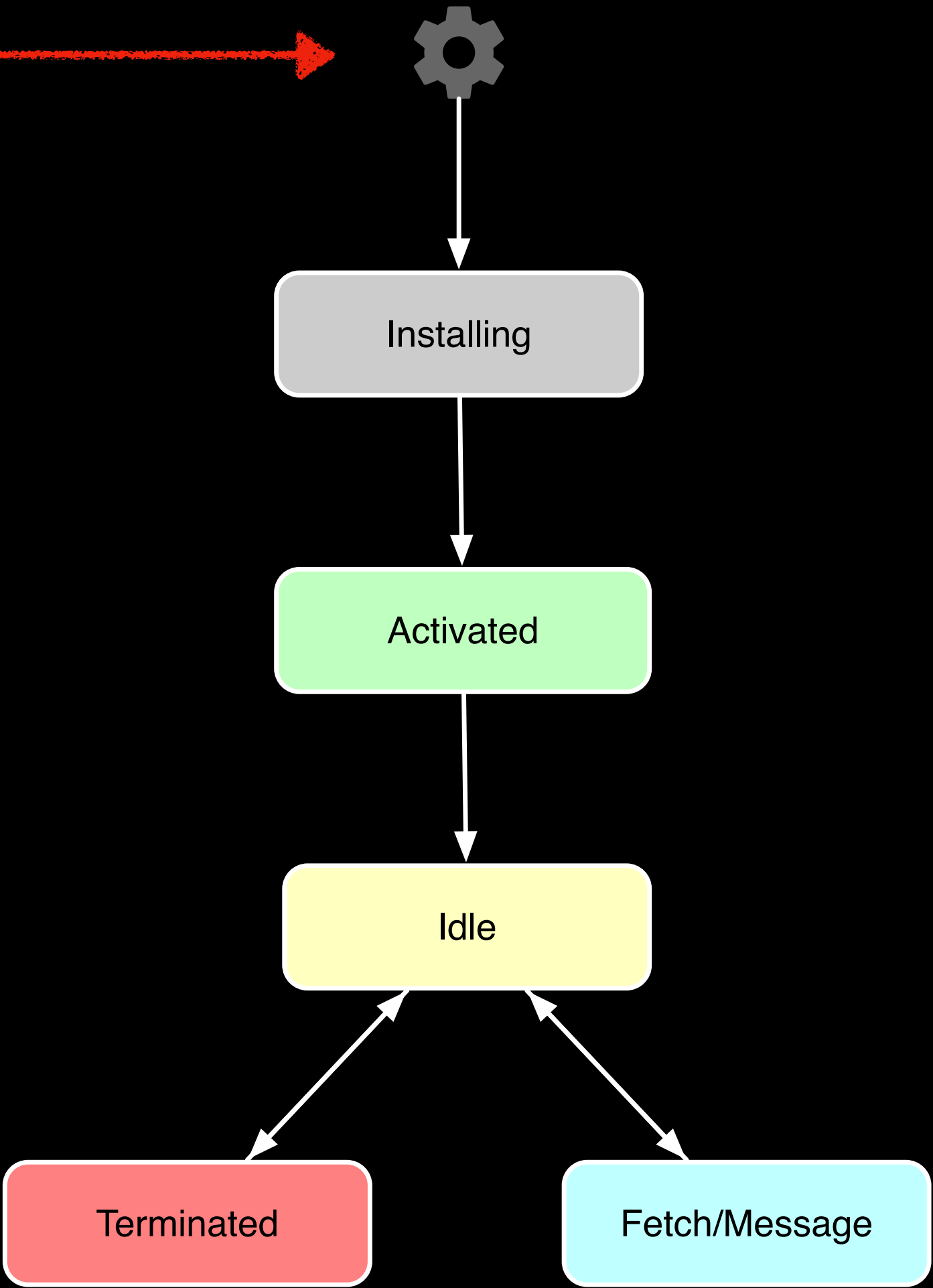
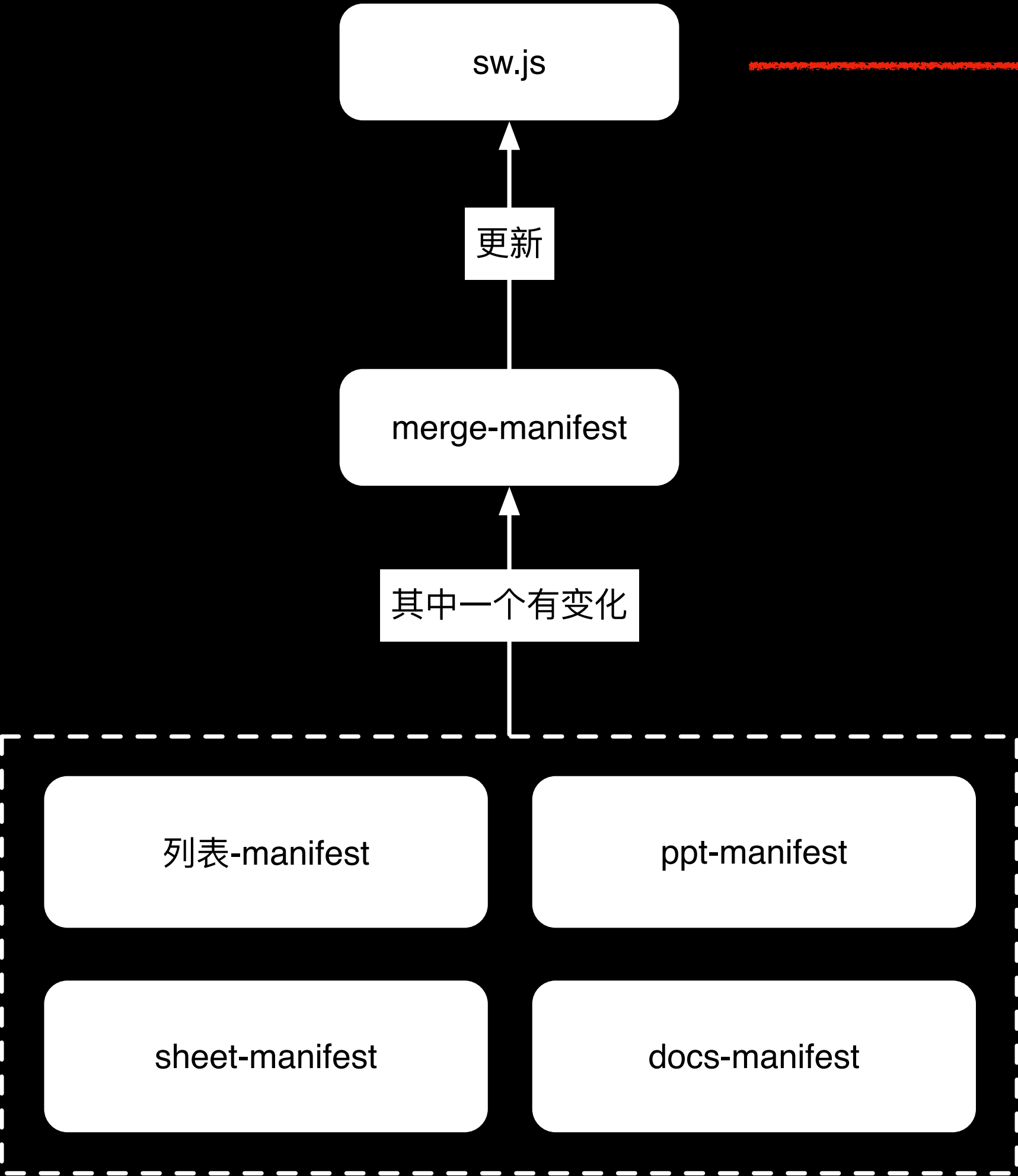


VS



# PWA 是如何在腾讯文档业务中落地的

## 为什么不采用常规缓存方案



### 缺点

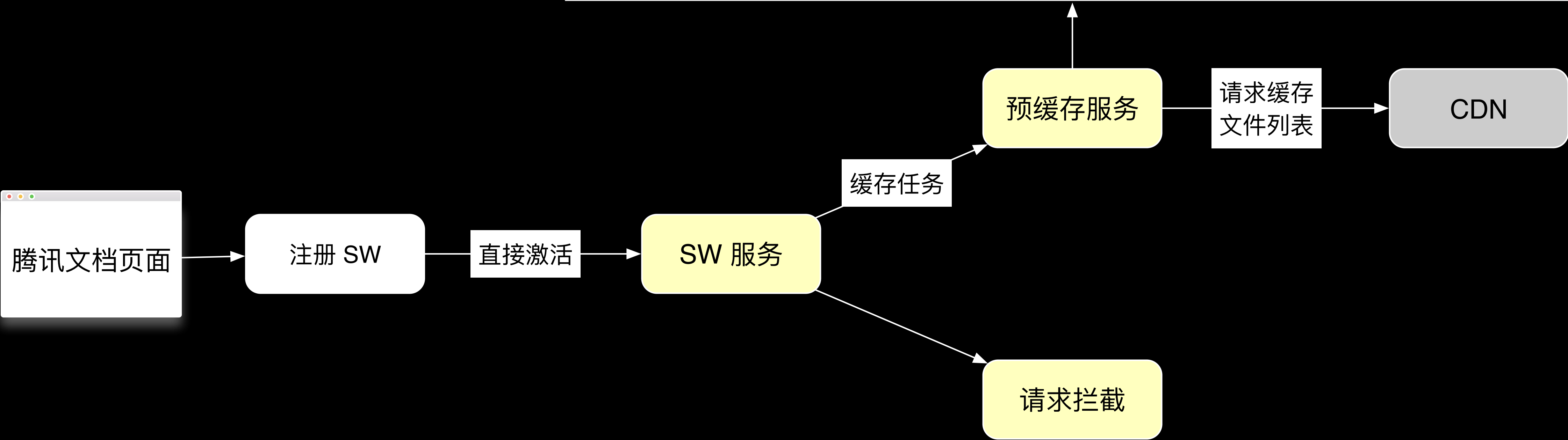
- 频繁更新 sw
- 不够灵活
- 不便管理
- 耦合 manifest

### 思路：

- 不耦合 manifest
- 运行时缓存

# 运行时缓存方案

```
[log] [sw] addPrecacheTask ▶ {url: "", dver: "2.8.0", docTypeDesc: "sheet"}
[log] [sw] addPrecacheTask ▶ {url: "", dver: "2.1.0", docTypeDesc: "doc"}
[log] [sw] addPrecacheTask ▶ {url: "", dver: "1.0.0", docTypeDesc: "slide"}
[log] [sw] addPrecacheTask ▶ {url: "", dver: "1.0.0", docTypeDesc: "desktop"}
```



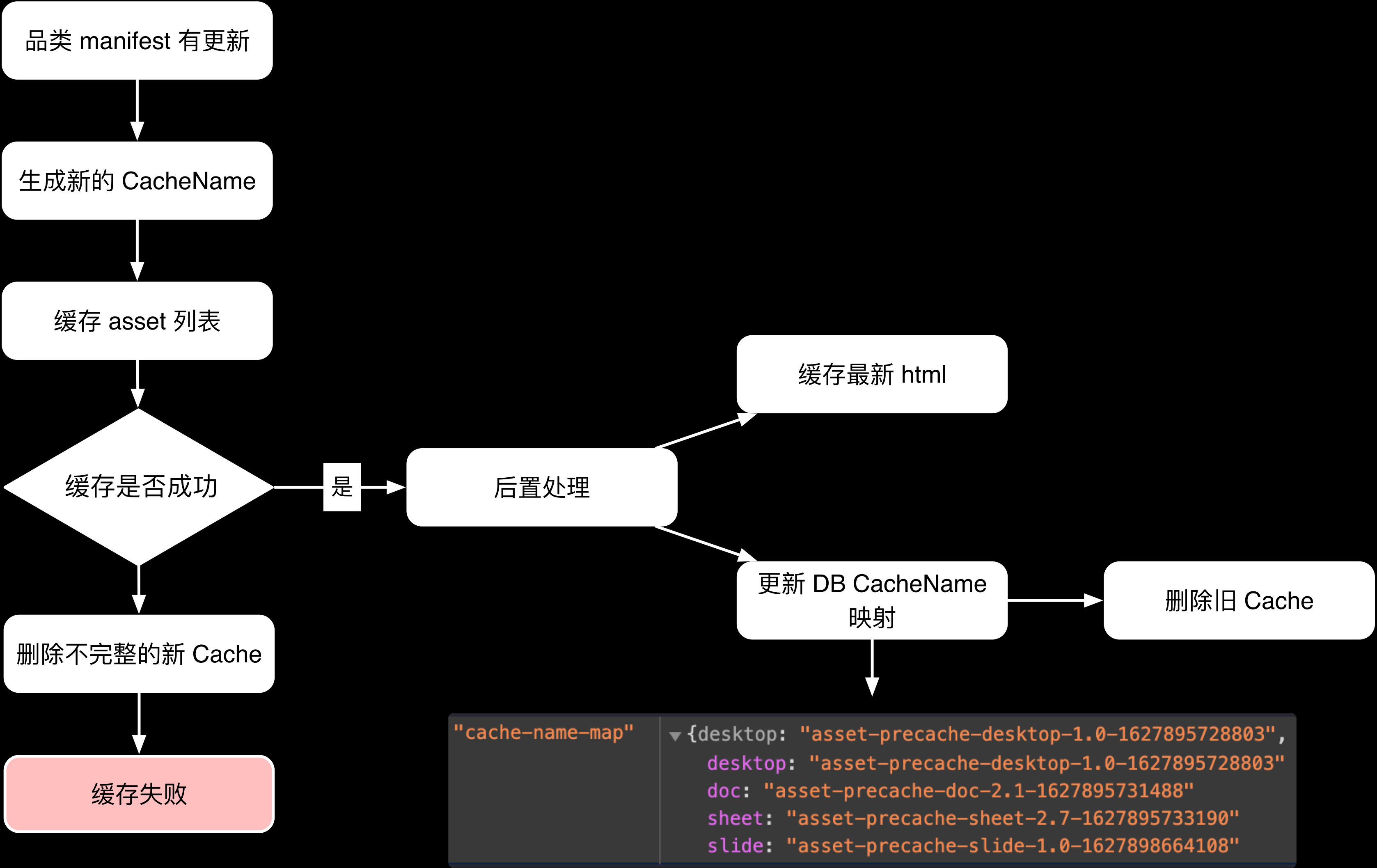
# 品类接入

```
{  
  // html cdn 路径  
  index,  
  // 静态资源 cdn 路径列表  
  flieList,  
}
```

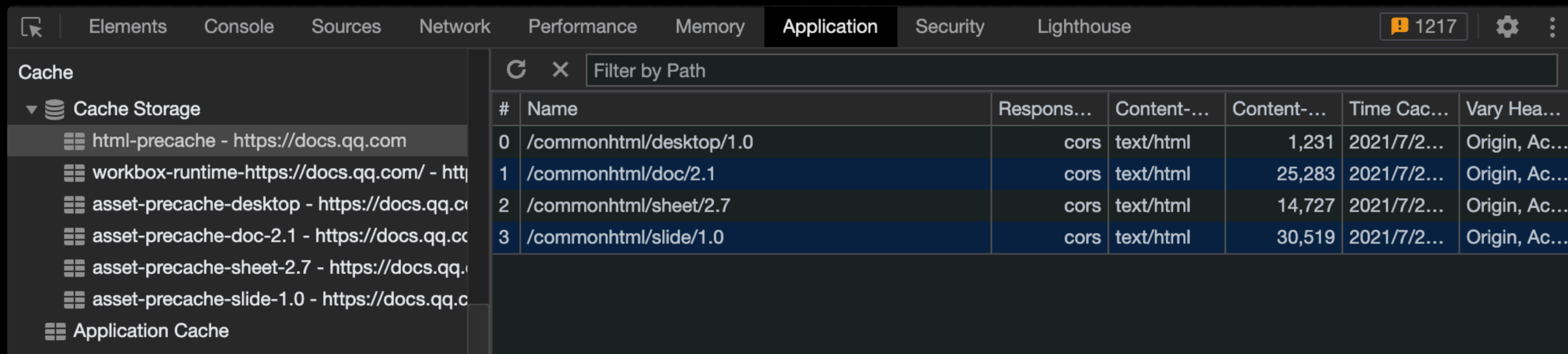
构建 -> 版本发布 -> 资源配置文件发布

# PWA 是如何在腾讯文档业务中落地的

## 缓存升级过程如何保障



# 资源如何存储

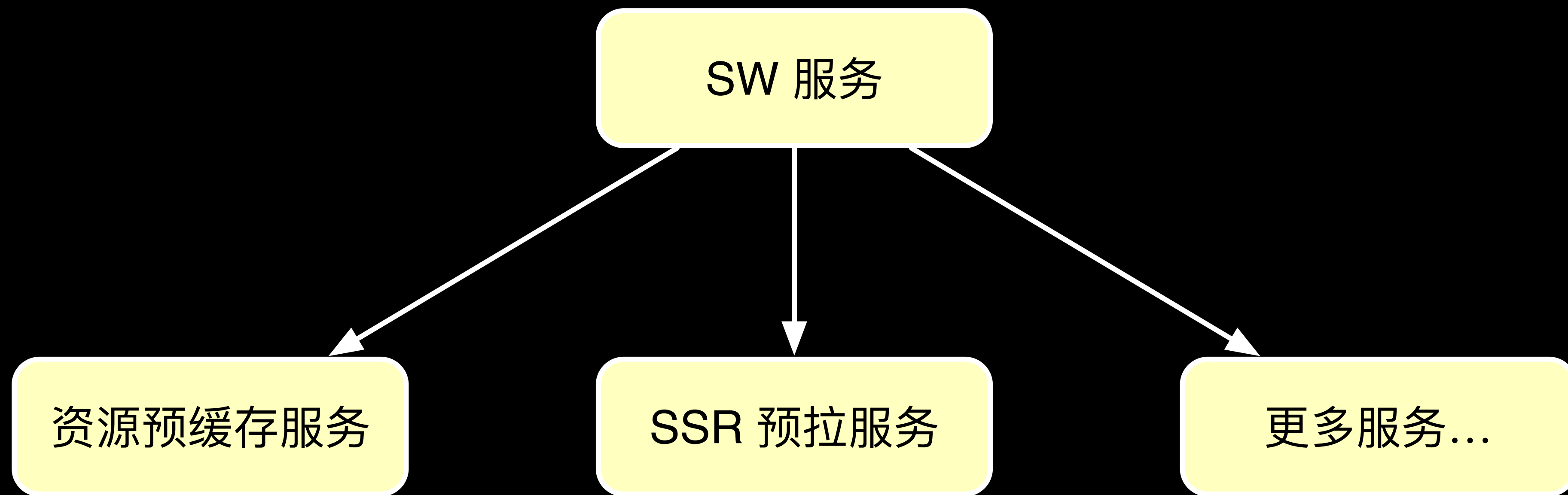


#	Name	Respons...	Content-...	Content-...	Time Cac...	Vary Hea...
0	/commonhtml/desktop/1.0	cors	text/html	1,231	2021/7/2...	Origin, Ac...
1	/commonhtml/doc/2.1	cors	text/html	25,283	2021/7/2...	Origin, Ac...
2	/commonhtml/sheet/2.7	cors	text/html	14,727	2021/7/2...	Origin, Ac...
3	/commonhtml/slide/1.0	cors	text/html	30,519	2021/7/2...	Origin, Ac...

HTML 存储：同品类只存储一份 html

Asset 存储：存储 html 对应资源列表

# SW 实践小结





## 分享大纲

当我们讨论 PWA 时，我们在讨论些什么

PWA 是如何在腾讯文档业务中落地的

腾讯文档做了哪些 PWA 实践，效果如何



分享大纲

当我们讨论 PWA 时，我们在讨论些什么

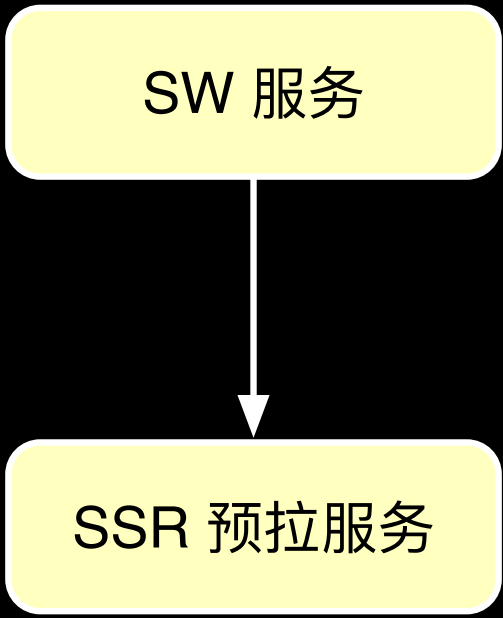
PWA 是如何在腾讯文档业务中落地的

腾讯文档做了哪些 PWA 实践，效果如何

腾讯文档做了哪些 PWA 实践，效果如何

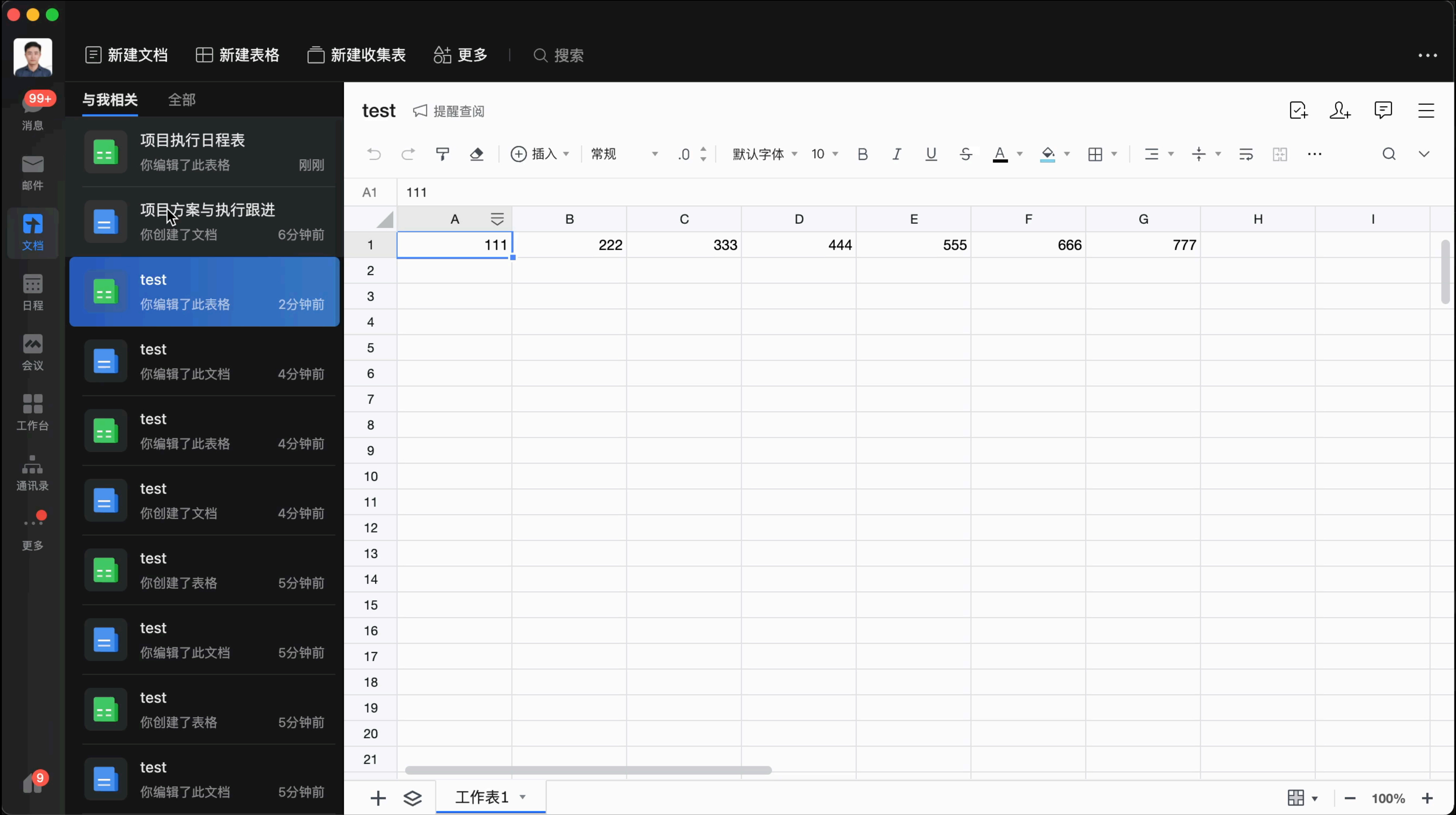
秒开体验

秒开体验



# 腾讯文档做了哪些 PWA 实践，效果如何

秒开体验

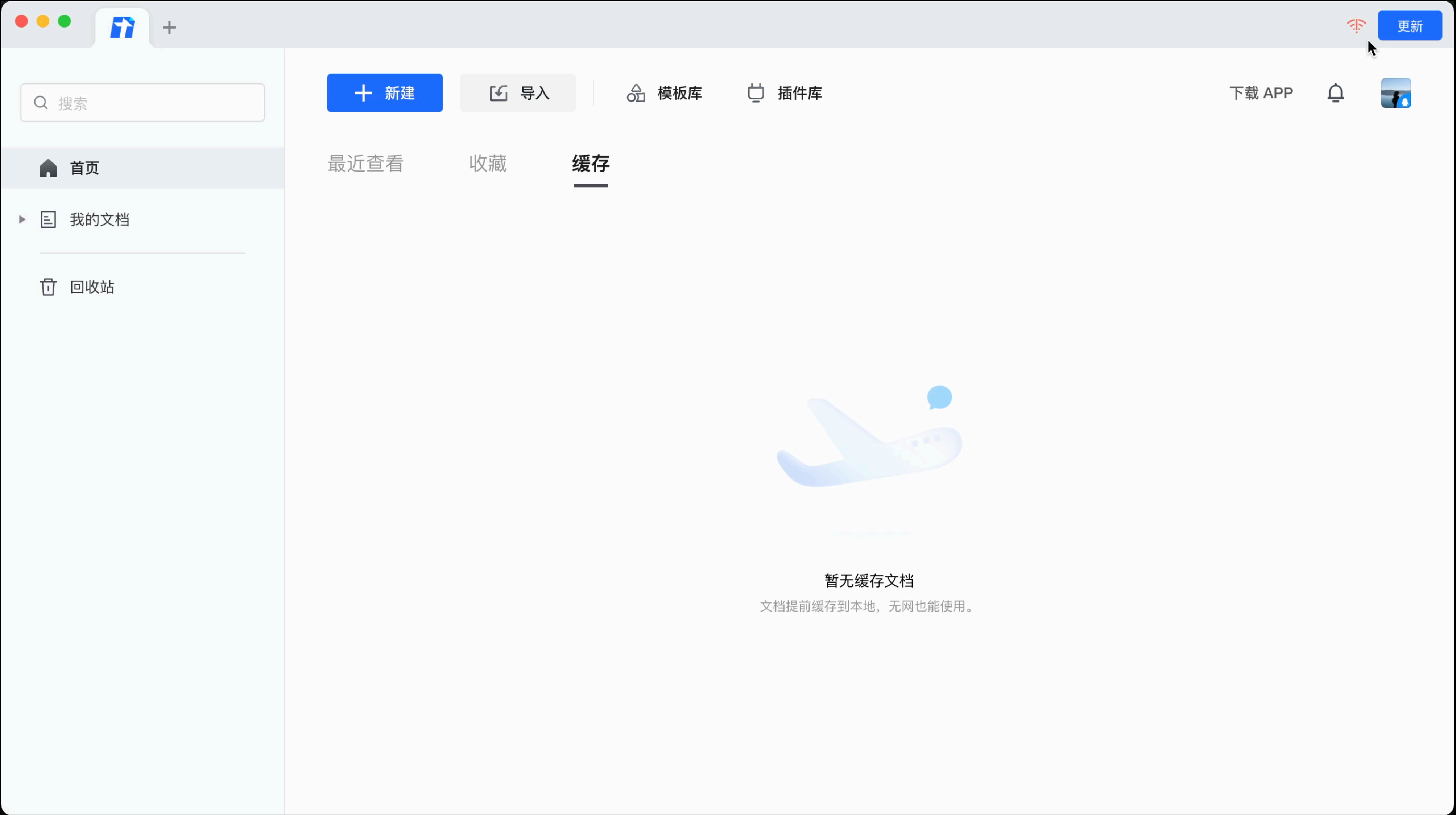


腾讯文档做了哪些 PWA 实践，效果如何

离线使用

# 腾讯文档做了哪些 PWA 实践，效果如何

## 离线使用



THANK YOU

Q & A