

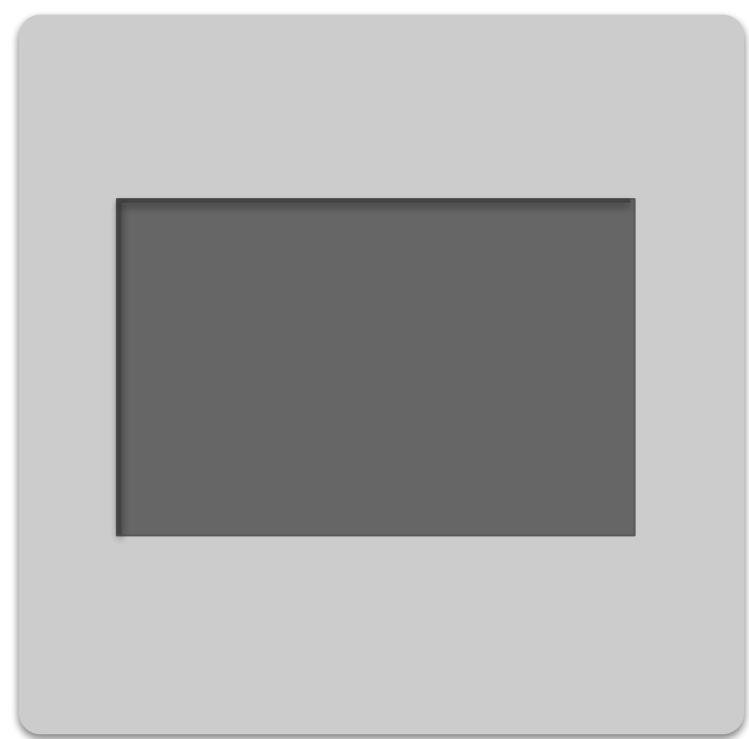
Web Packaging



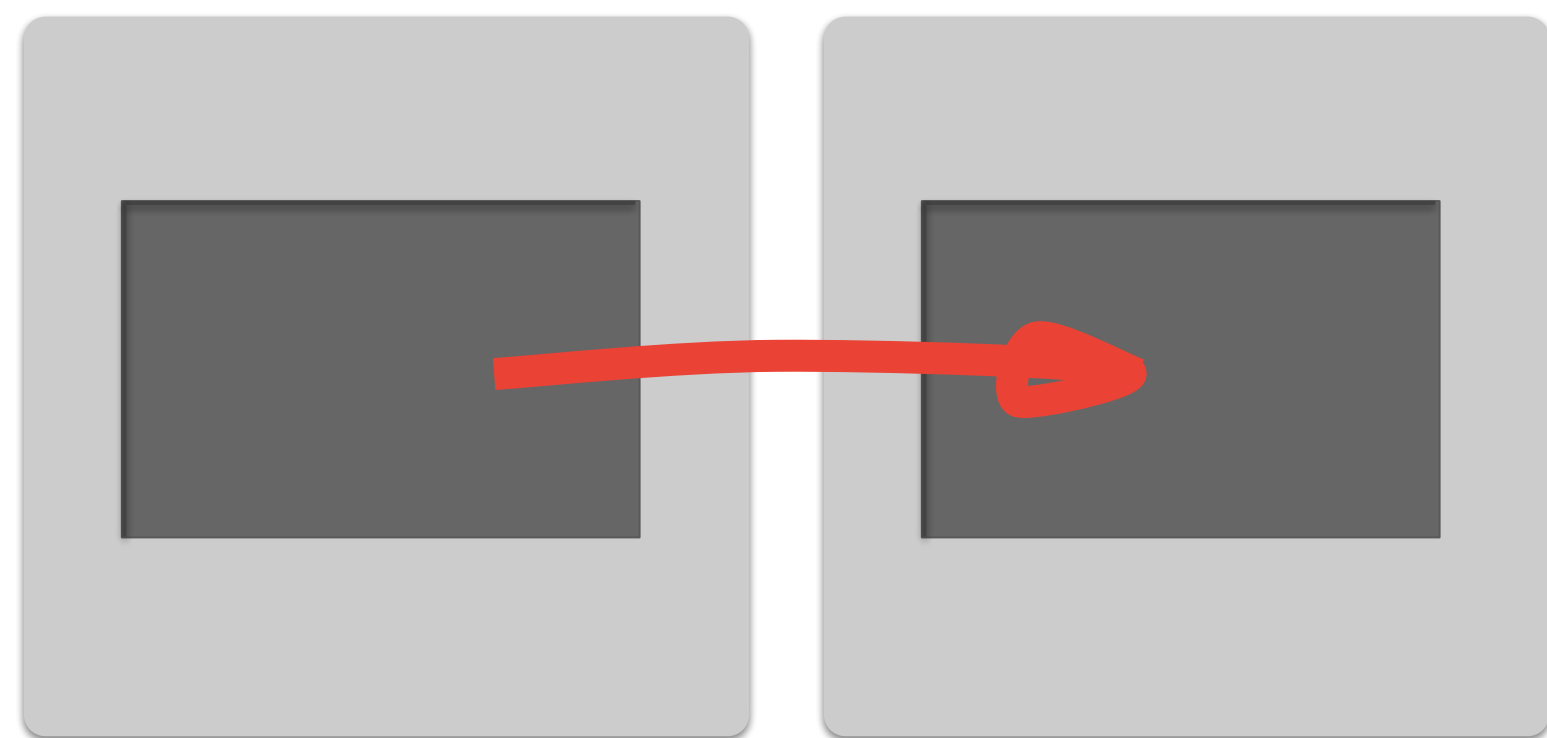
Michael Yeung

杨智行

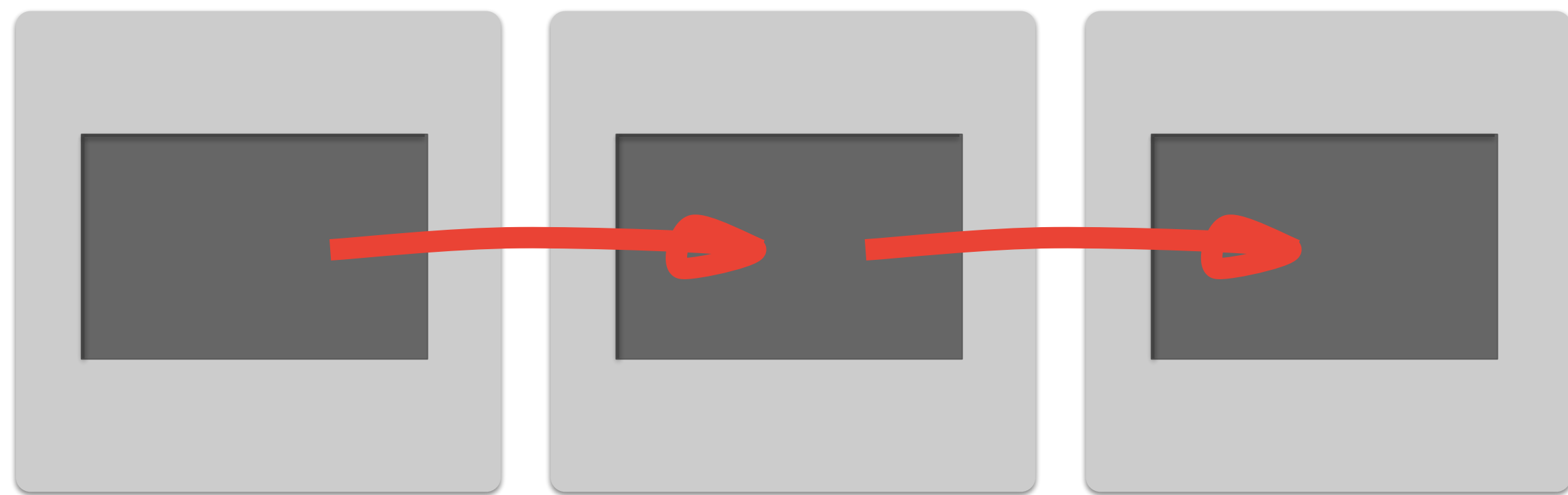
今天的 Web - 不禁让人感受到 navigation 的慢



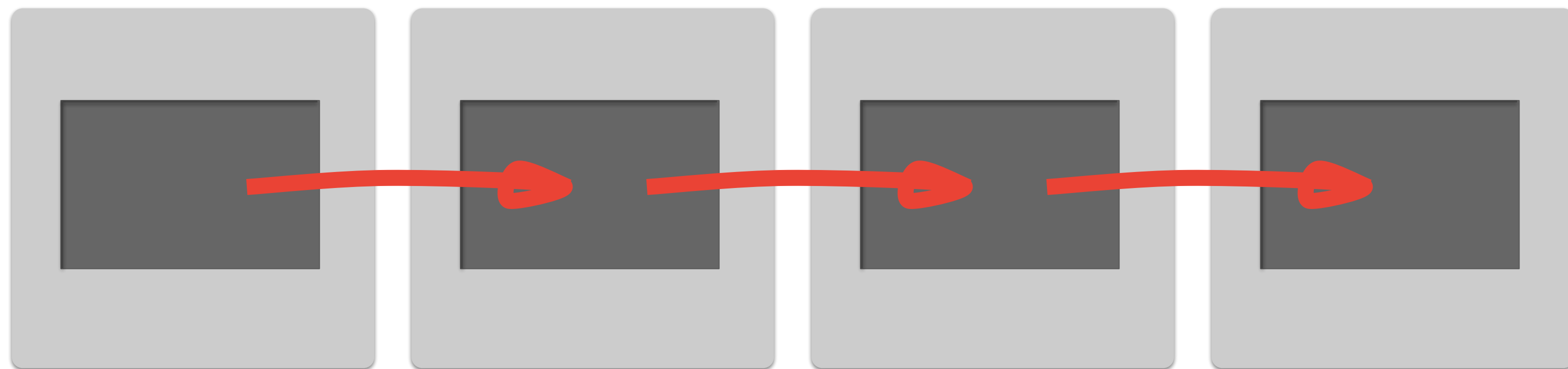
今天的 Web - 不禁让人感受到 navigation 的慢



今天的 Web - 不禁让人感受到 navigation 的慢



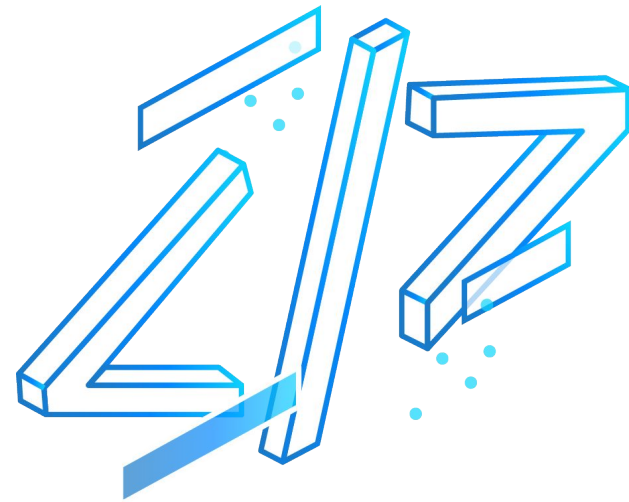
今天的 Web - 不禁让人感受到 navigation 的慢





AMP

三层不同的加速方式



**AMP HTML &
JAVASCRIPT**

默认快速，强制验证



AMP CACHE

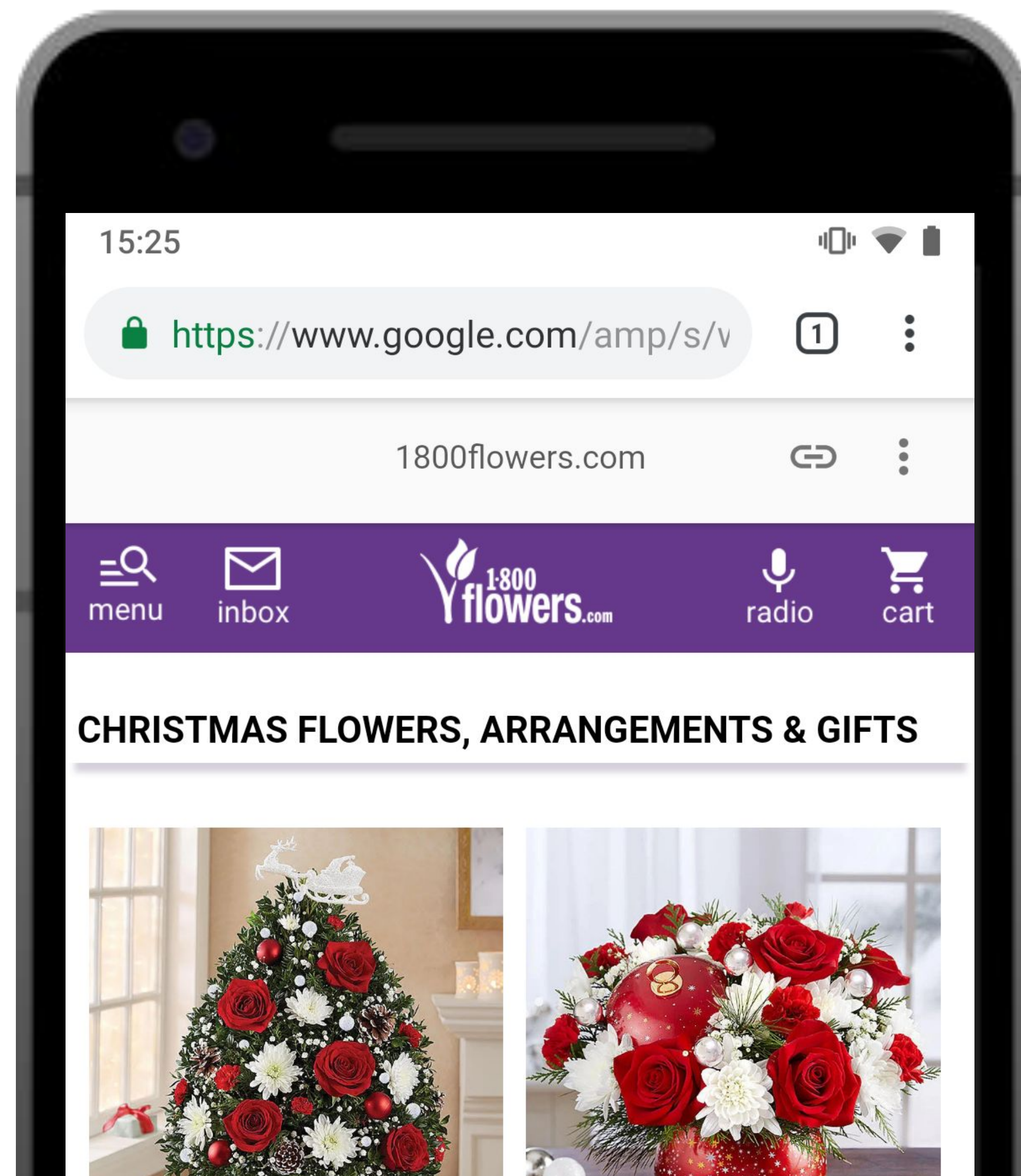
近距离缓存及优化



平台 Pre-render

特异功能！

... 产品里——实现 (lòu xiàn)



浏览器 address bar

AMP Viewer header

从 Cache 加载的 AMP 页面
(AMP Viewer 里渲染)



Improving URLs for AMP pages

January 9, 2018

[amphtml](#)

TL;DR: We are making changes to how AMP works in platforms such as Google Search that will enable linked pages to appear under publishers' URLs instead of the google.com/amp URL space while maintaining the performance and privacy benefits of AMP Cache serving.

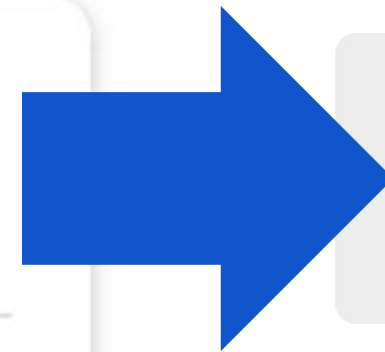
When we first launched AMP in Google Search we made a big trade-off: to achieve the user experience that users were telling us that they wanted, instant loading, we needed to start loading the page before the user clicked. As we detailed in a deep-dive [blog post](#) last year, privacy reasons make it basically impossible to load the page from the publisher's server. Publishers shouldn't know what people are interested in until they actively go to their pages. Instead, AMP pages are loaded from the Google AMP Cache but with that behavior the URLs changed to include the google.com/amp/ URL prefix.

We are huge fans of meaningful URLs ourselves and recognize that this isn't ideal. Many of y'all agree. It is

This is an example AMP page

⚡ example.com

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore
magna aliqua. ...



google.com/amp/www.example.com



把 AMP 的学习经验带到 The Whole Web



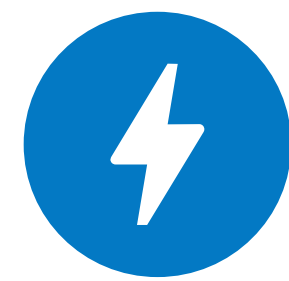
AMP 的特别之处
(AMP viewer, 格式, 等)

非AMP 页面

难！

Web Platform

把 AMP 的学习经验带到 The Whole Web



AMP

非 AMP 页面



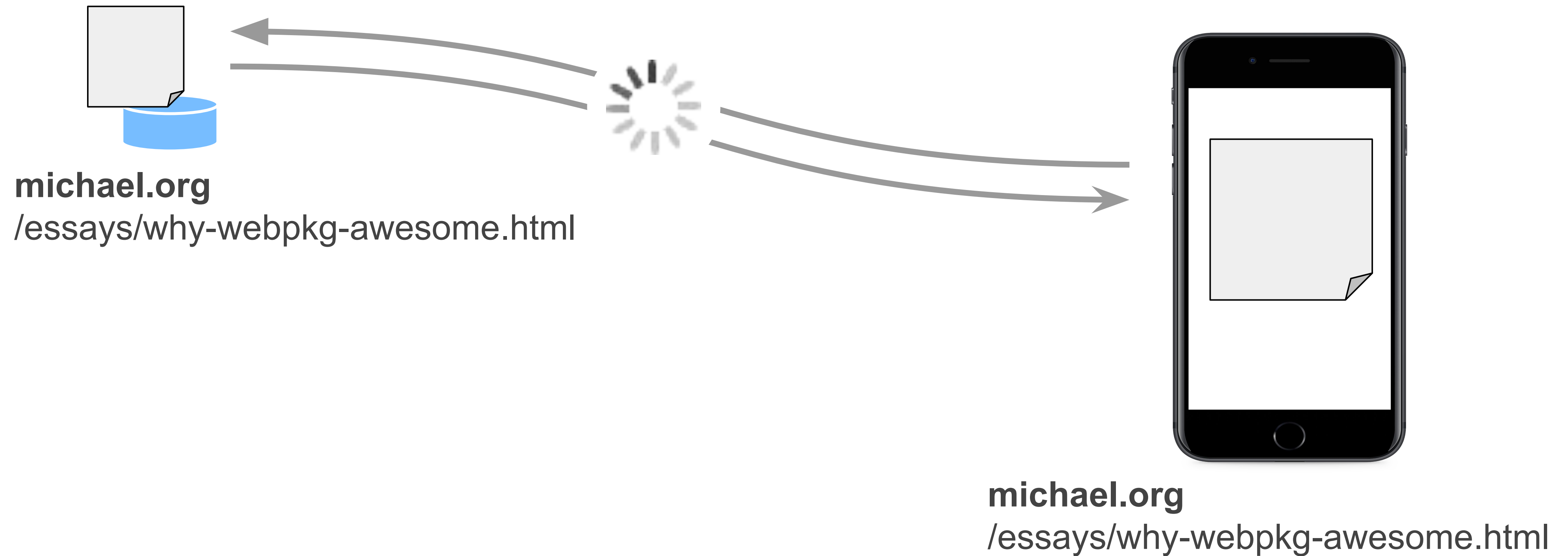
提升用户体验的新 Web 标准

Web Packaging

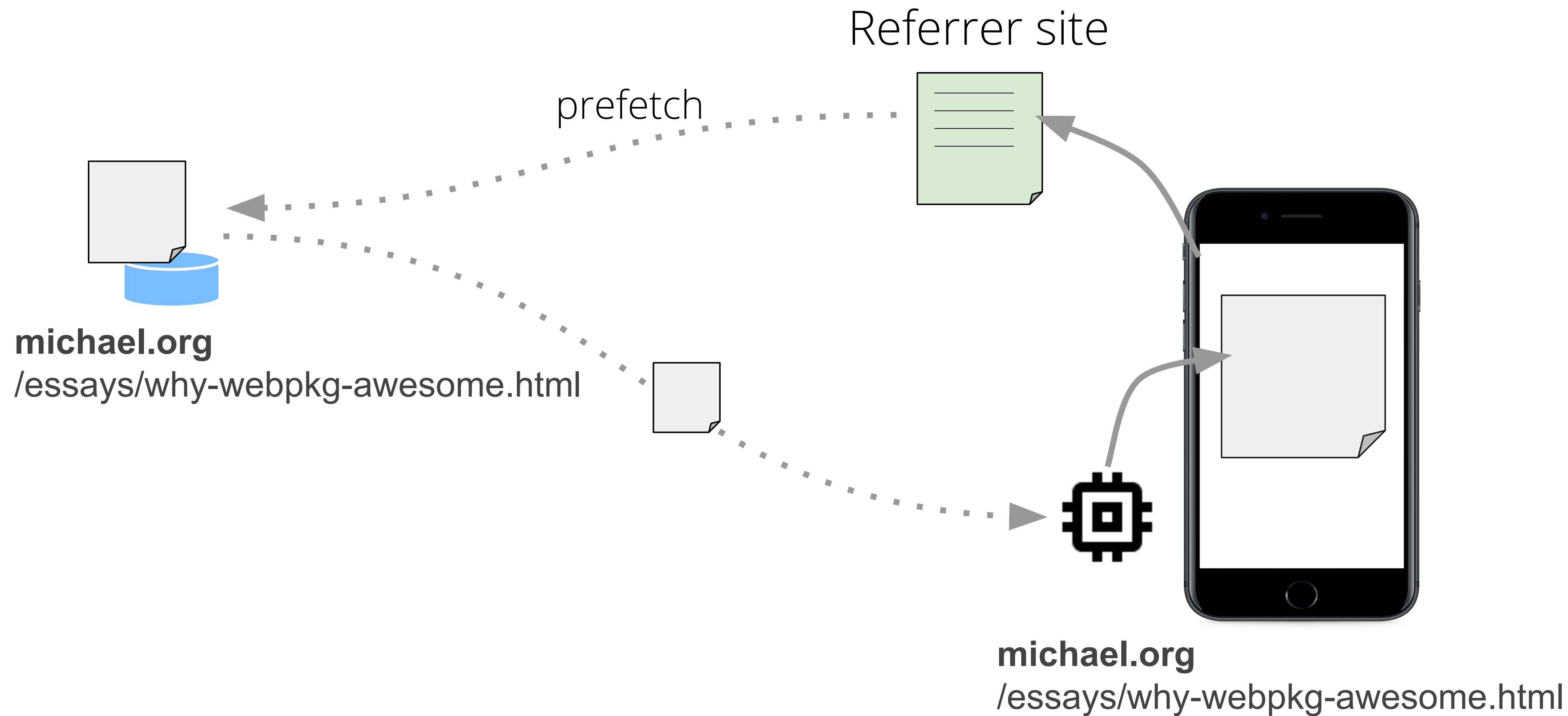
Web Packaging

保障用户隐私的 Instant Navigation

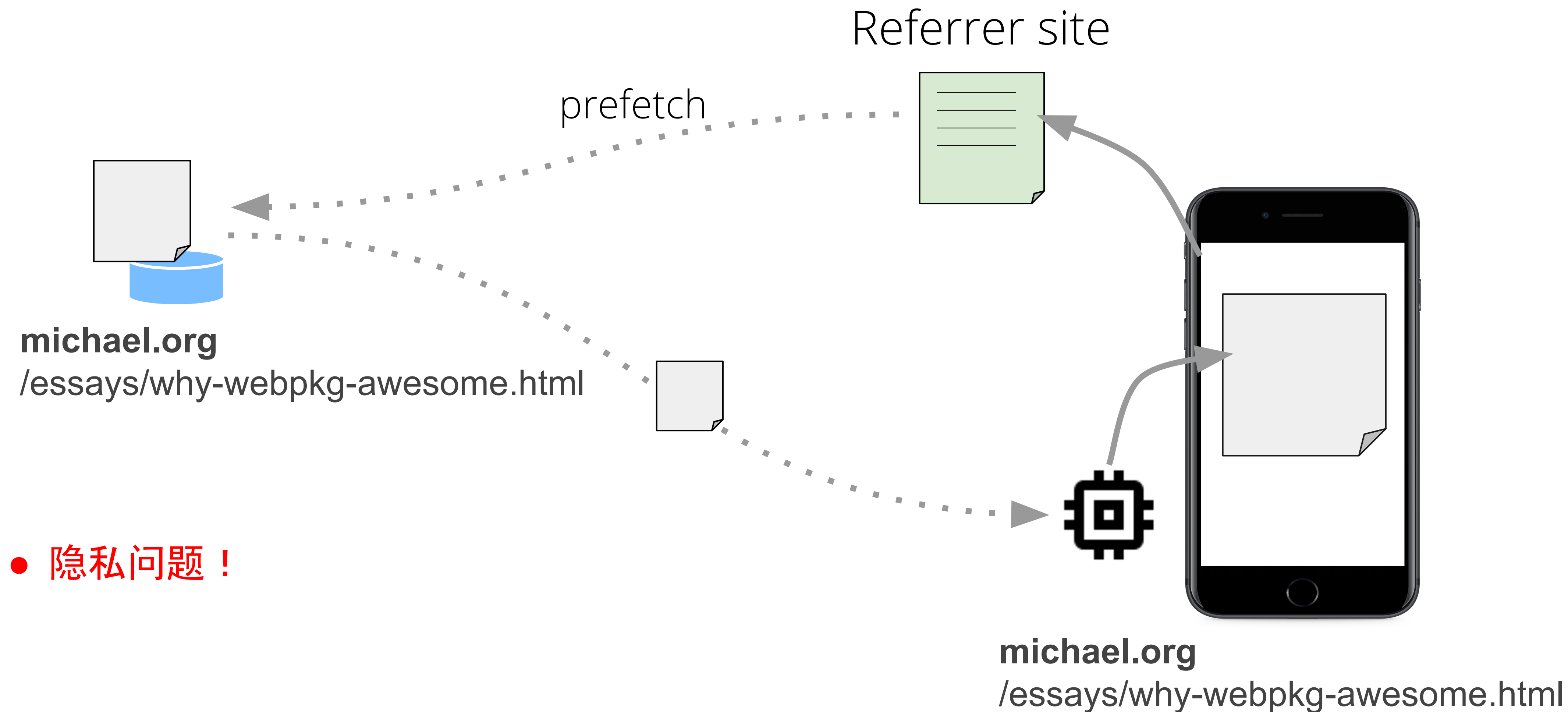
目标: 既 Instant 又 reliable 的加载



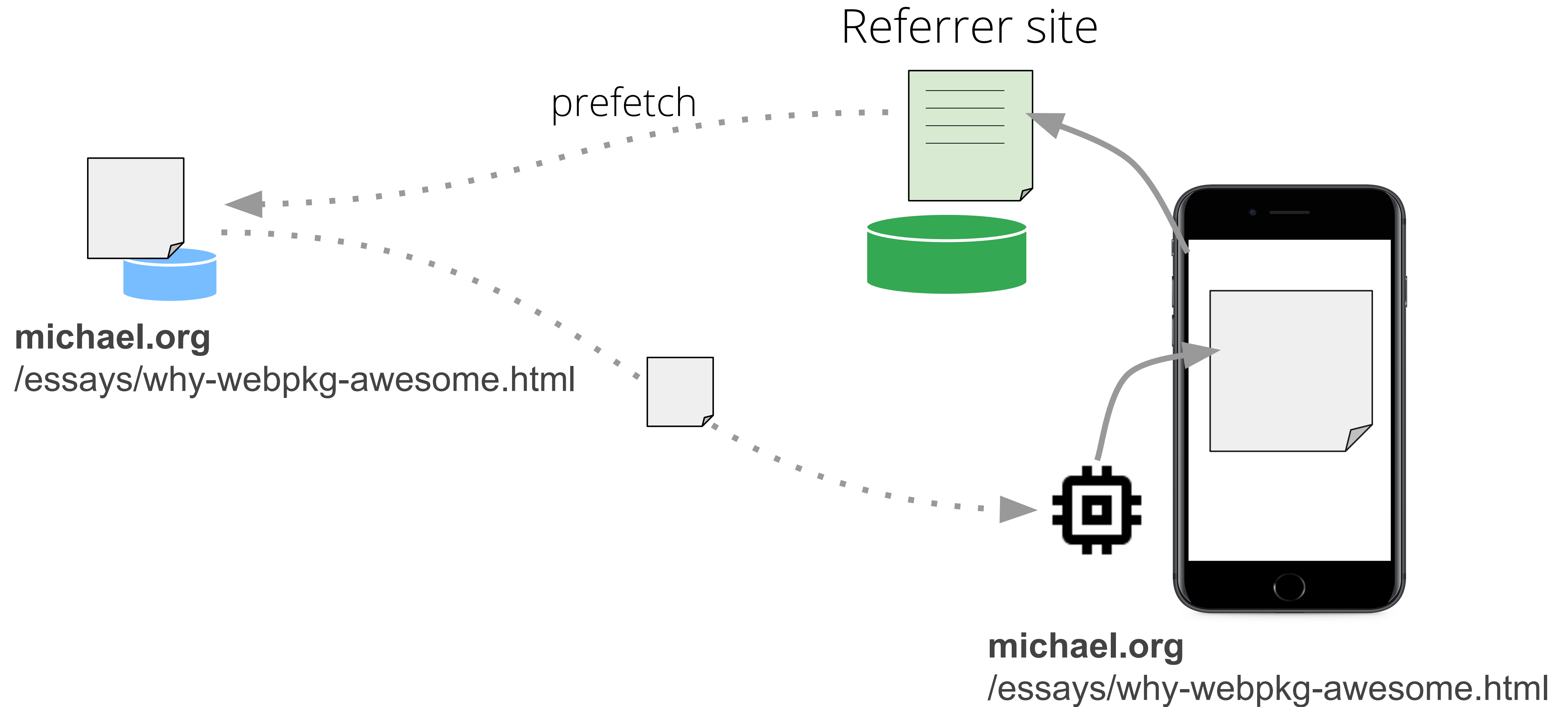
方案 #1: Prefetch



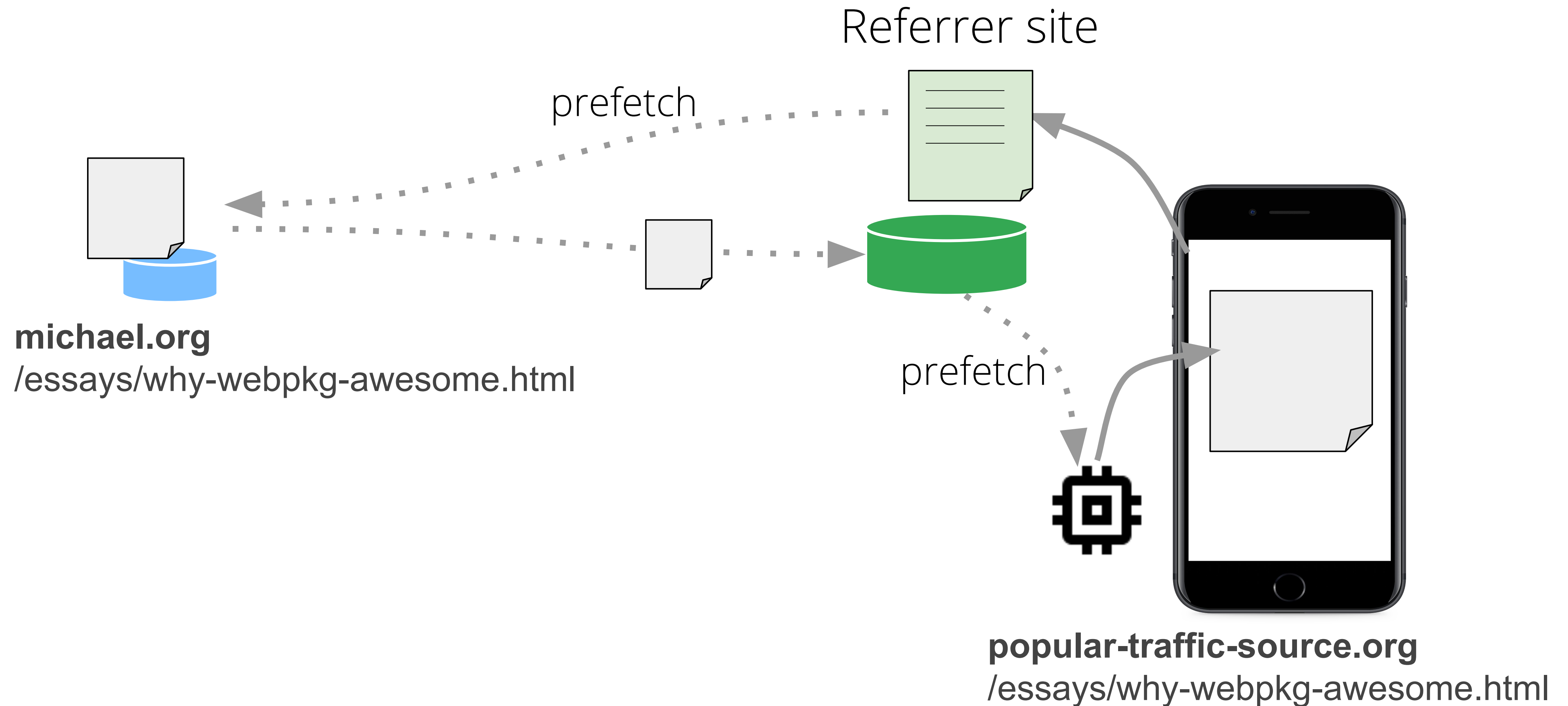
方案 #1: Prefetch



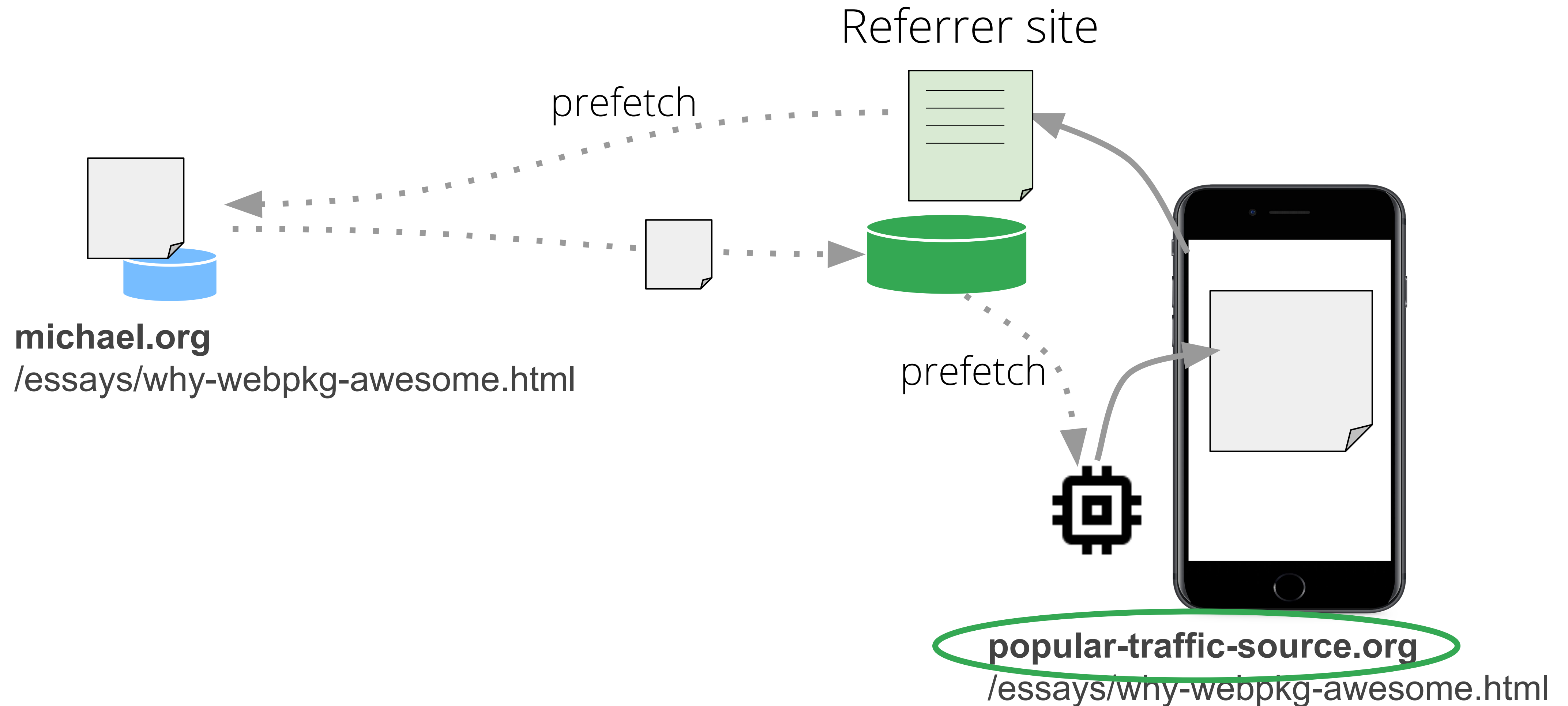
方案 #2: Prefetch + Cache



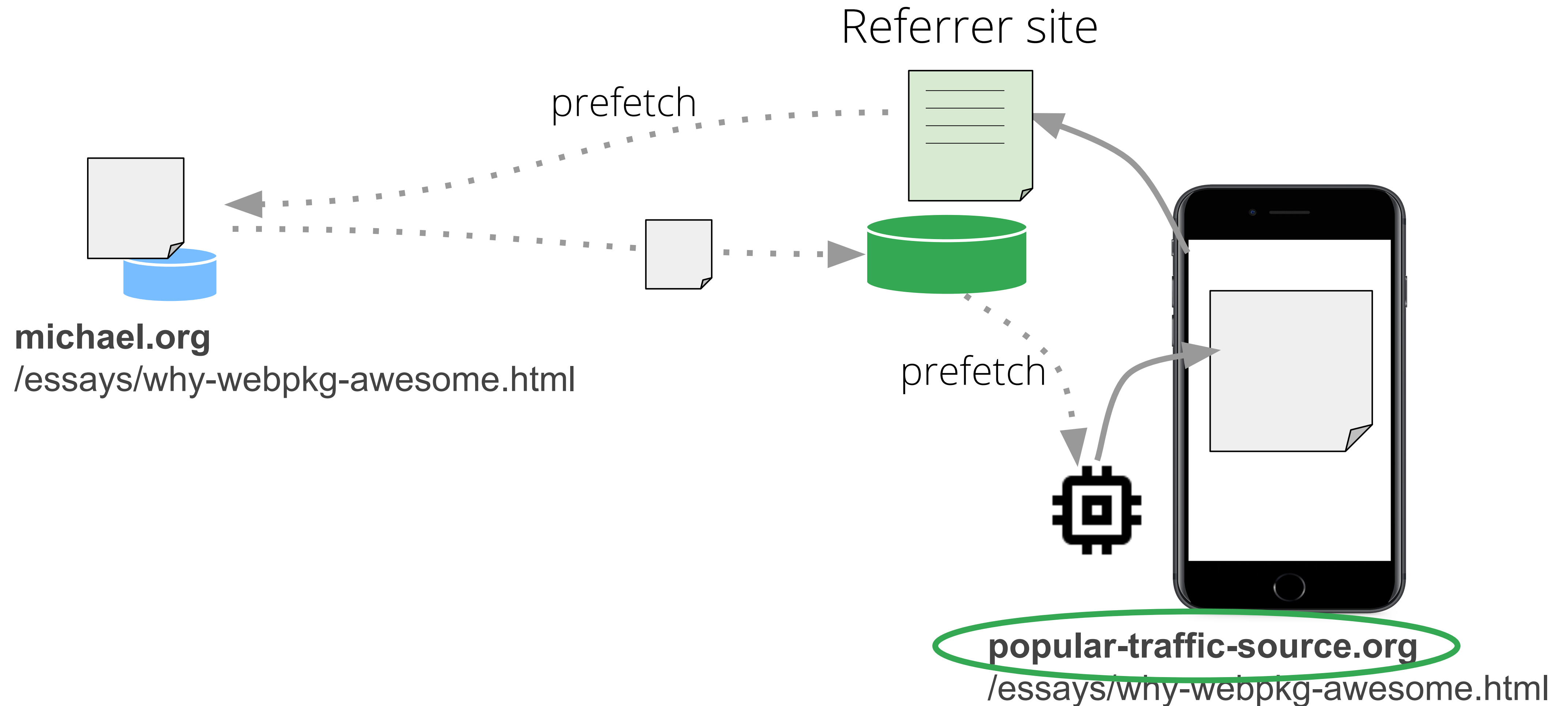
方案 #2: Prefetch + Cache



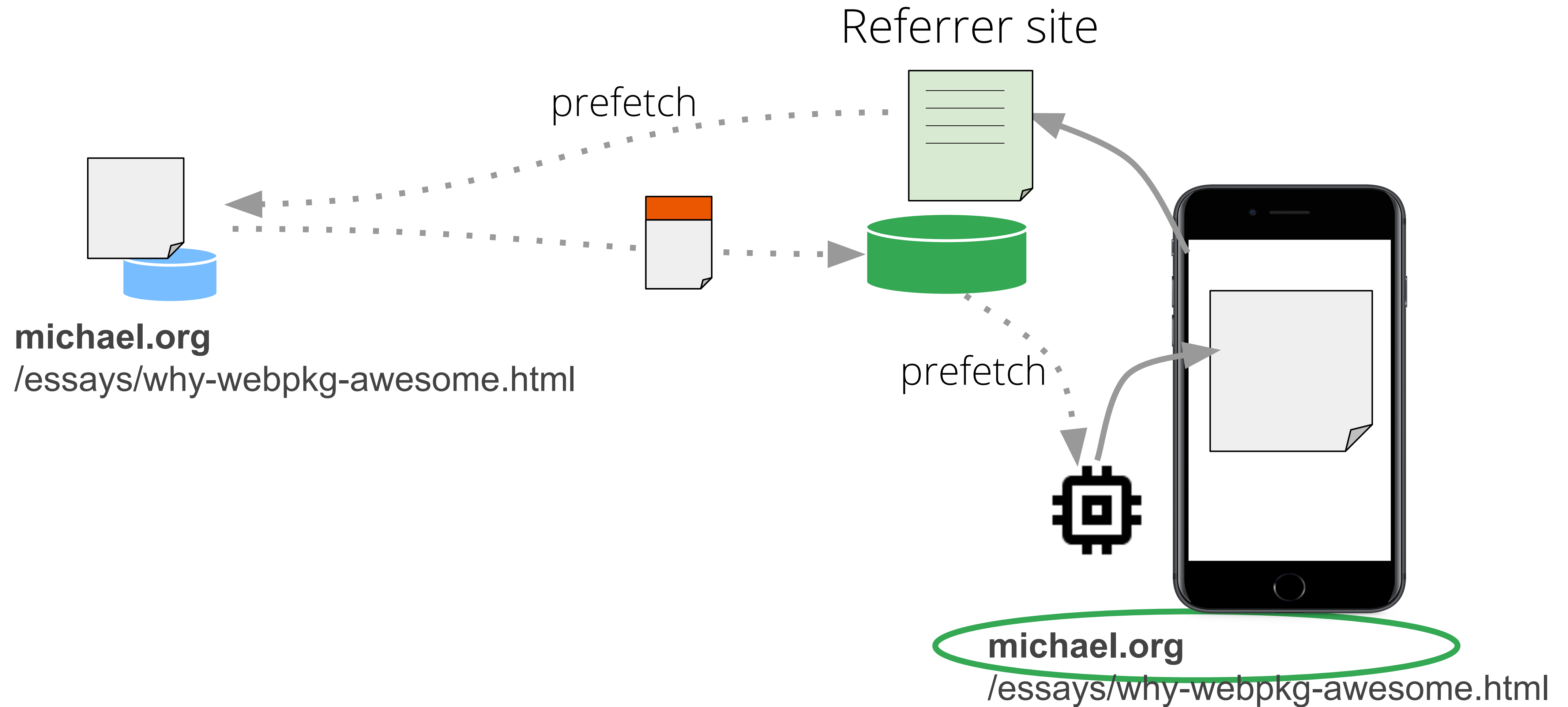
方案 #2: Prefetch + Cache



方案 #2: Prefetch + Cache



方案 #3: Prefetch + Cache + 授权加载

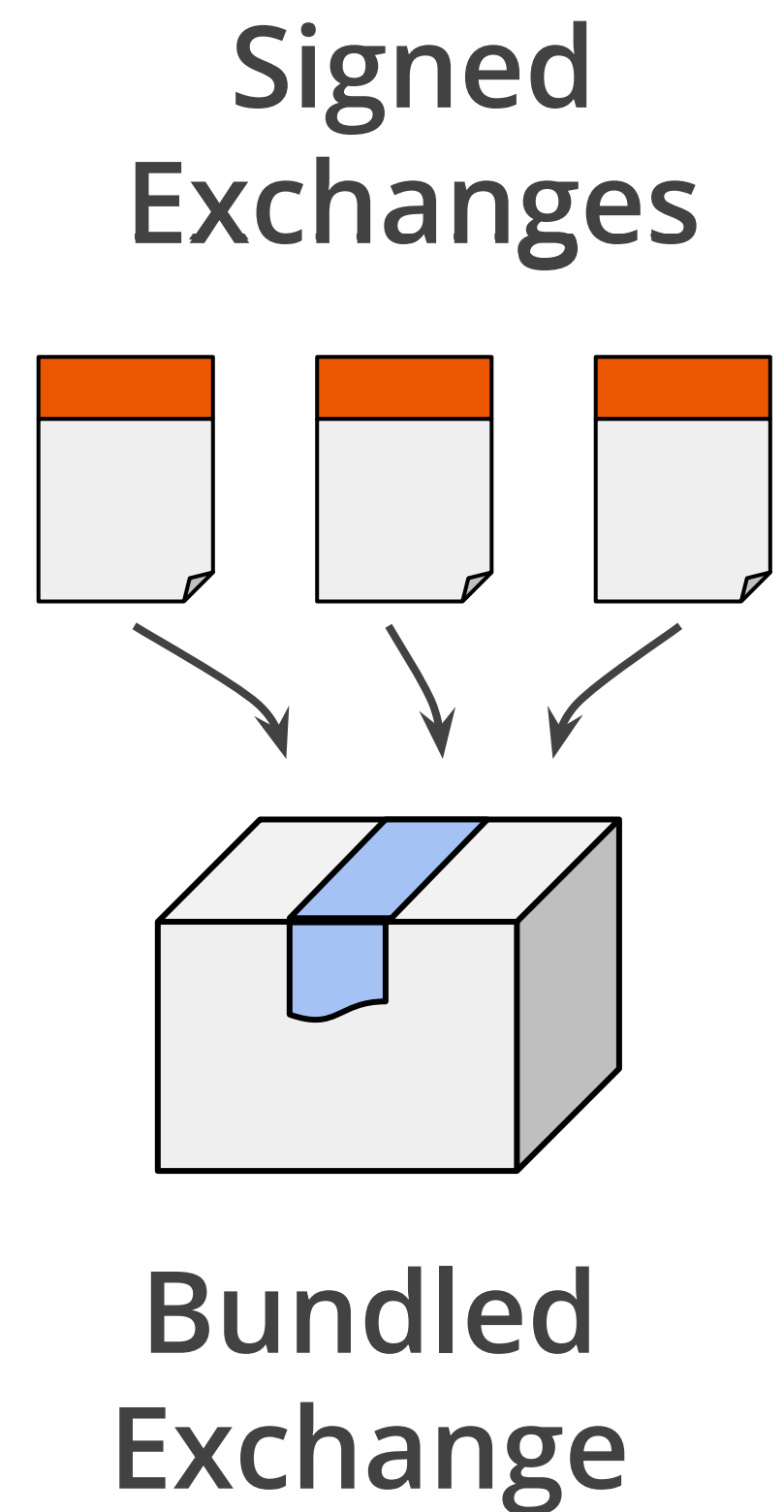


你的 resource 应该有个 proof-of-origin
来证明这个 resource 确实是属于你的。

这就是 Web Packaging 的功效。

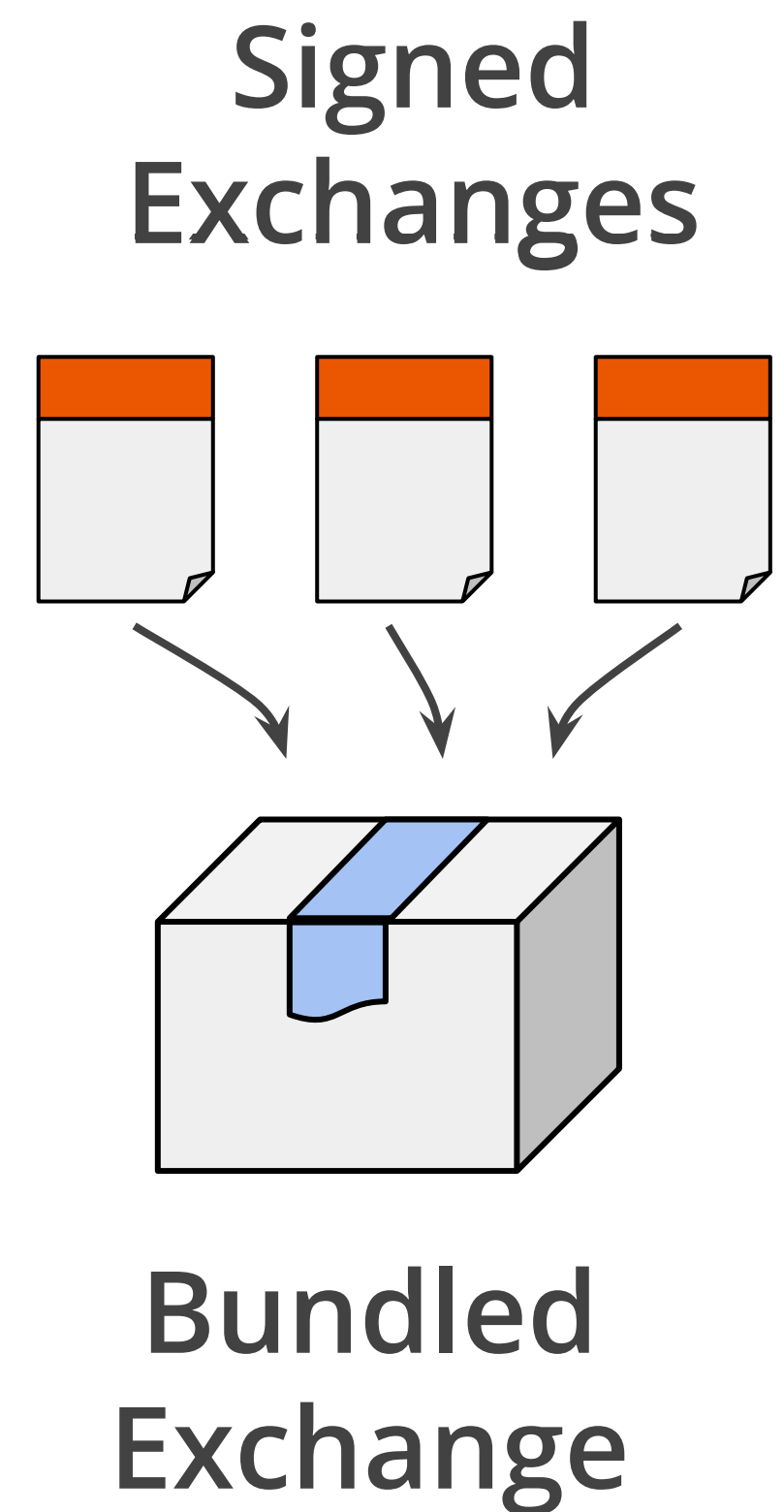
Web Packaging

- Exchanges = HTTP request/response pairs
- Signed Exchanges = 已加密签名的 Exchanges, 好让浏览器验证 resource 的来源
- Bundled Exchanges = Exchanges 的“包” (未必需要是Signed)



Web Packaging

- Exchanges = HTTP request/response pairs
- Signed Exchanges = 已加密签名的 Exchanges, 好让浏览器验证 resource 的来源
- Bundled Exchanges = Exchanges 的“包” (未必需要是Signed)



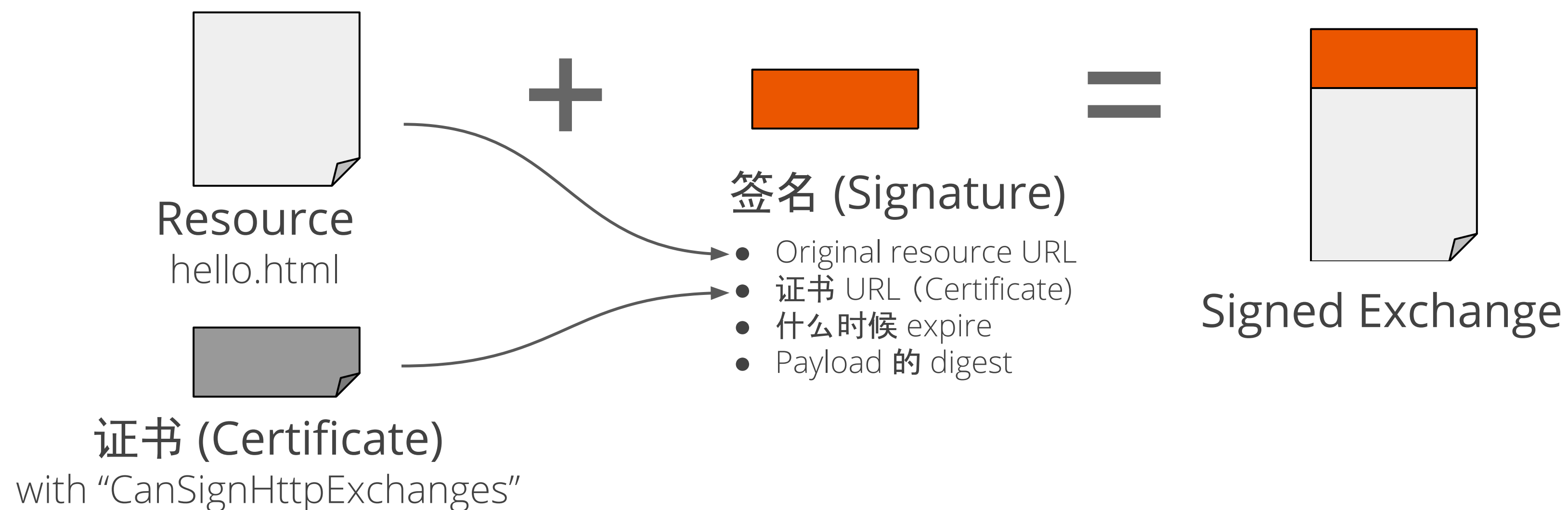
试用 Signed Exchange!

Chrome 71+

```
<!-- prefetch sample.sxg, 就是  
      https://your-site.com/hello.html 的 SXG 版本-->  
<link rel="prefetch" href="https://cache.example.com/sample.sxg">  
  
<!-- 点击这链接就会带到原先的网址:https://your-site.com/hello.html -->  
<a href="https://your-site.com/sample.sxg">Hello World, SXG!</a>
```

- 把 flag 打开: **chrome://flags/#enable-signed-http-exchange**
- Open experiment: <https://bit.ly/try-sxg>

在你的 site 使用 Signed Exchange



生成 Signed Exchange 的开源工具: bit.ly/webpackaging

更多: bit.ly/try-sxg

14:11



1



Search or type web address



Articles for you

Show

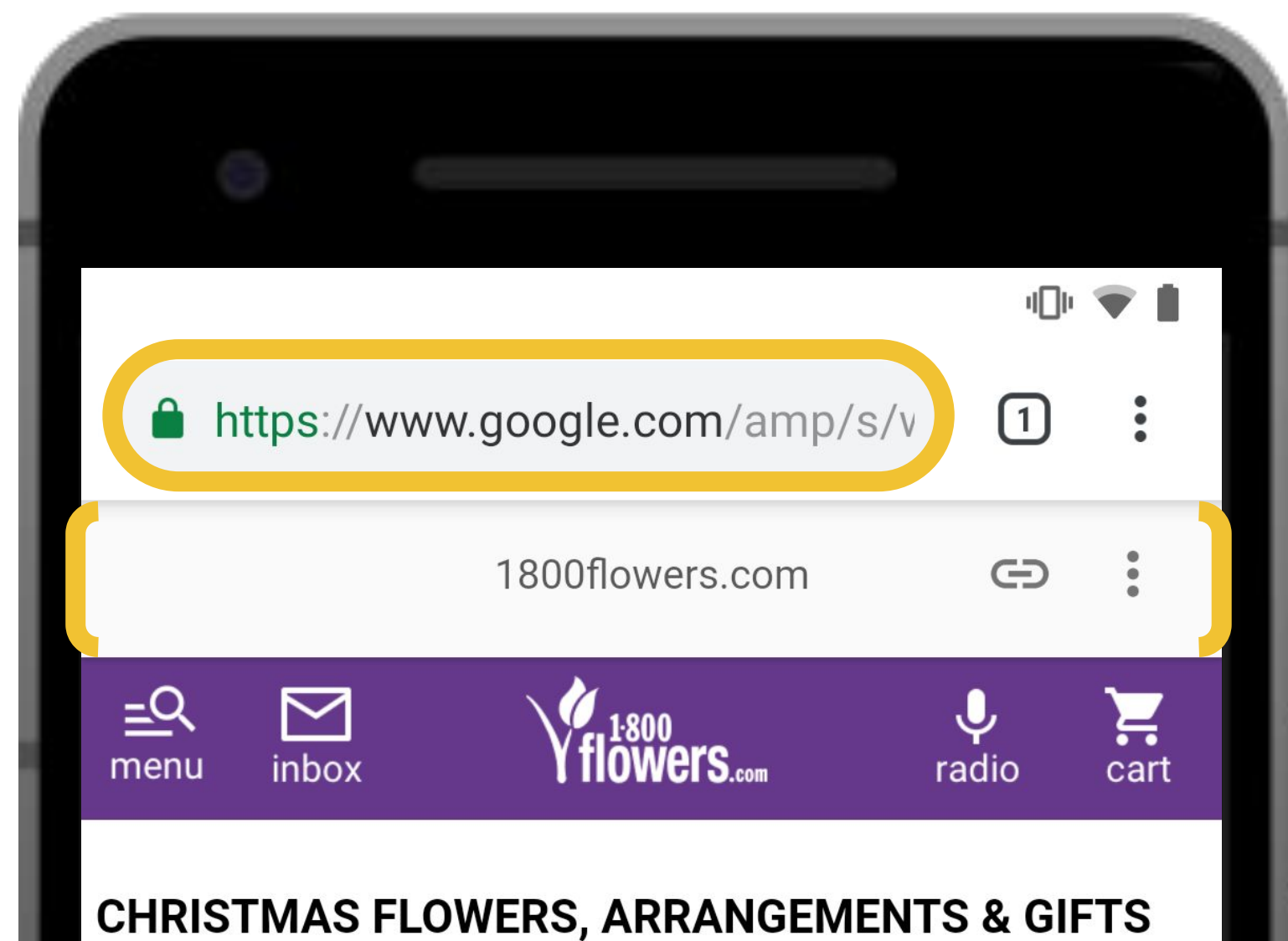
[Learn more](#) about suggested content



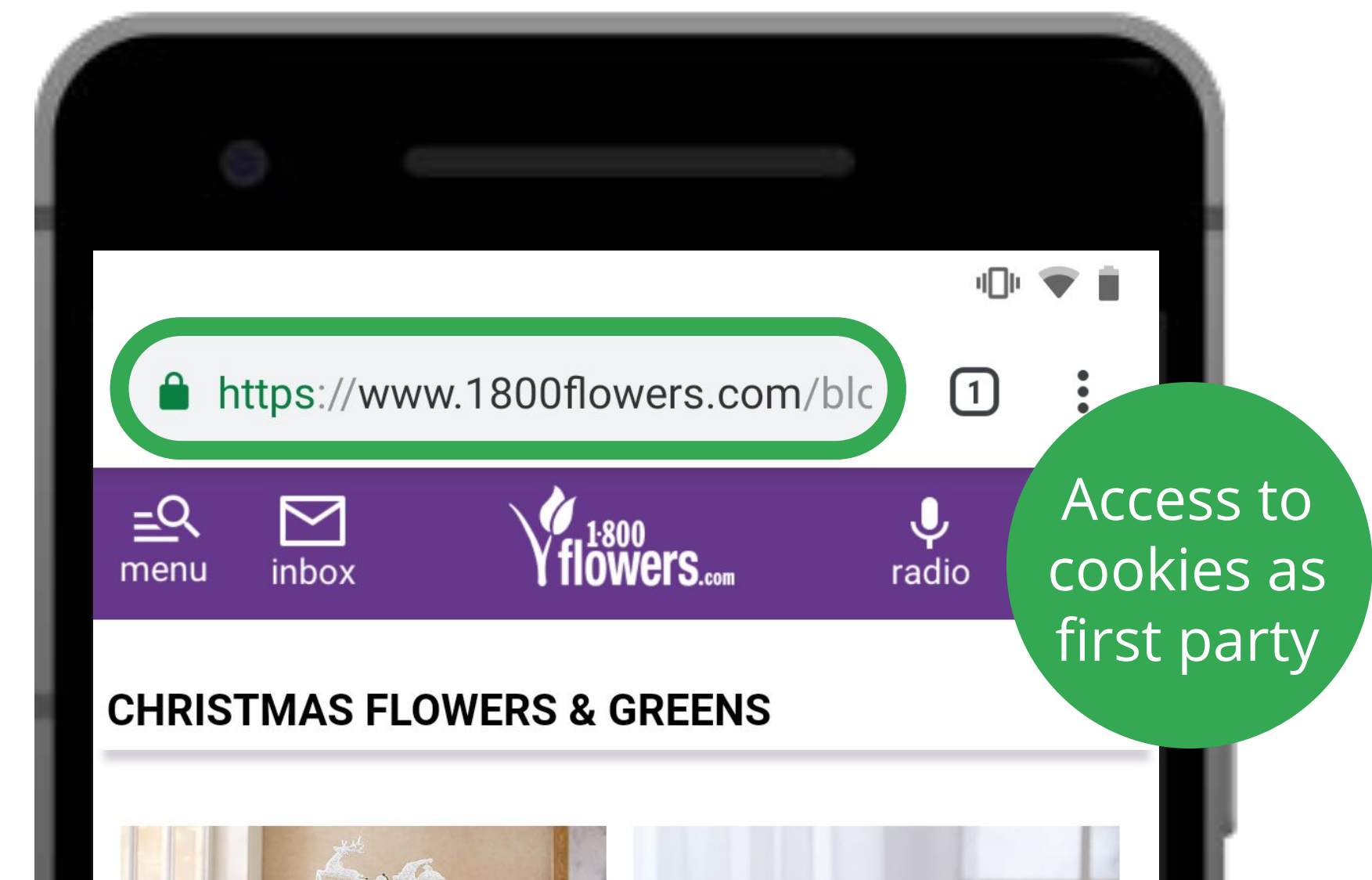
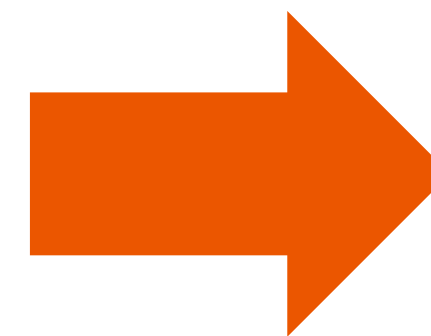
AMP Web Packaging - 谷歌搜索的开发者预览版

g.co/webpackagepreview

CURRENT without signed exchange

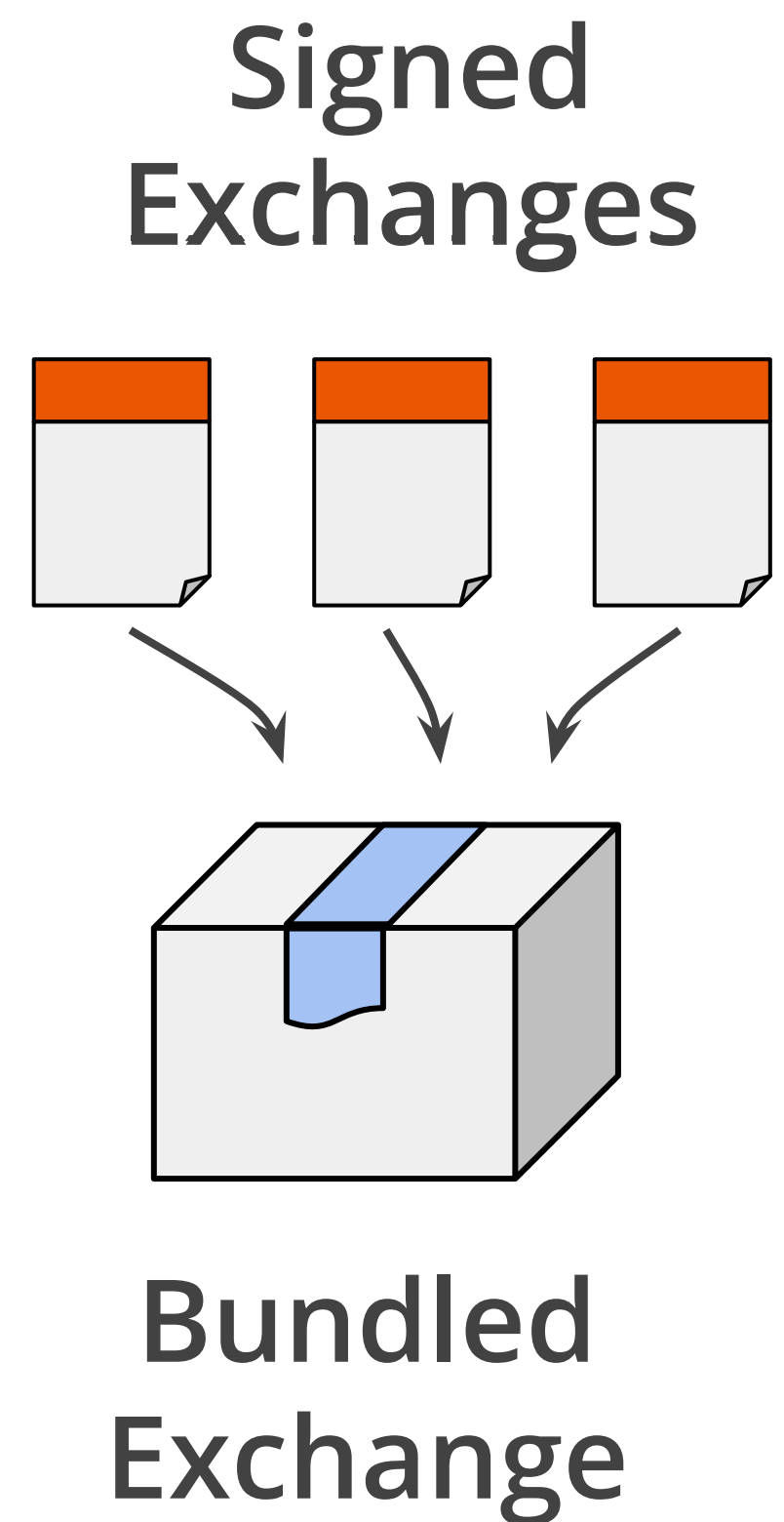


WITH SIGNED EXCHANGE



Web Packaging

- Exchanges = HTTP request/response pairs
- Signed Exchanges = 已加密签名的 Exchanges, 好让浏览器验证 resource 的来源
- Bundled Exchanges = Exchanges 的“包”（未必需要是Signed）

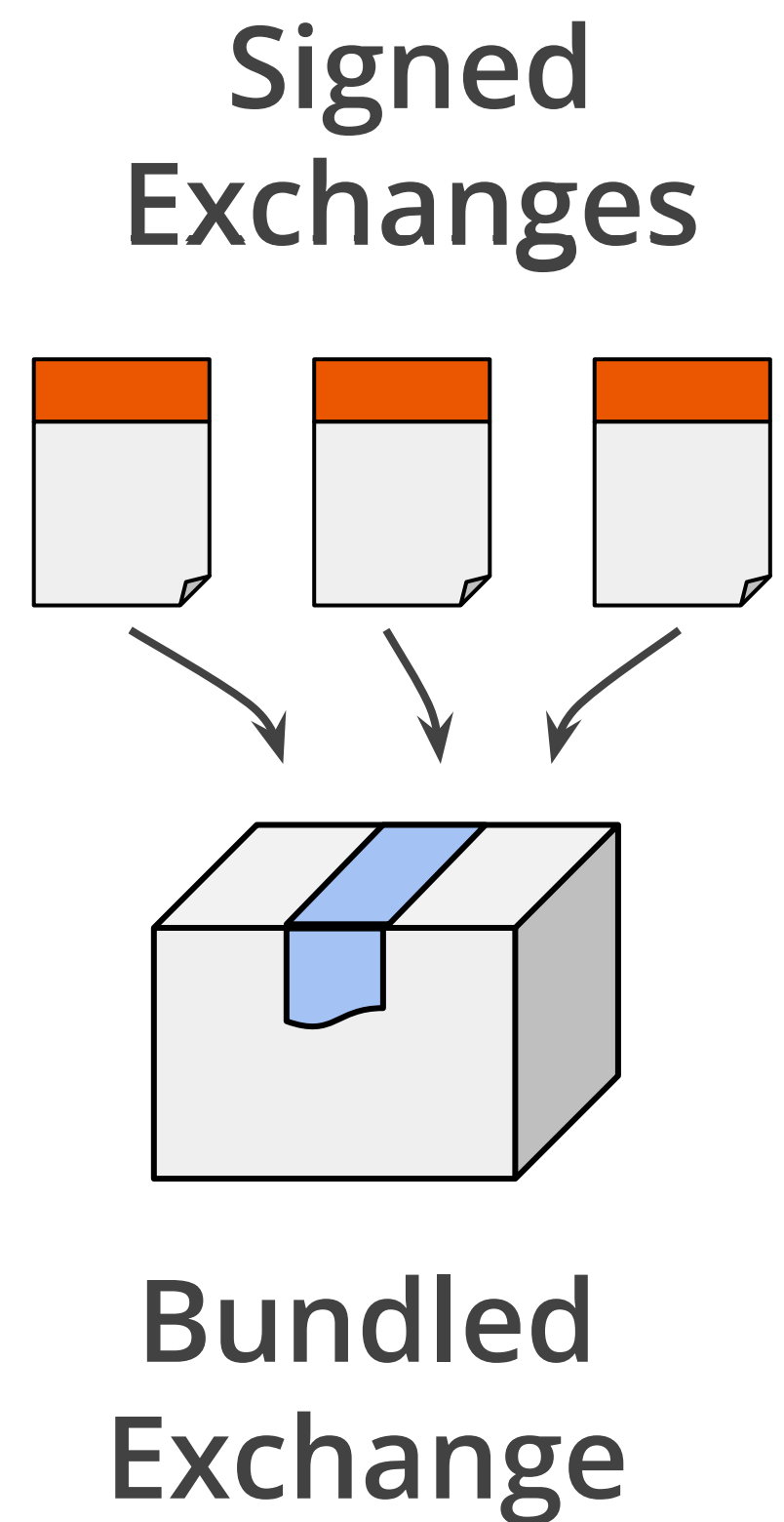


Bundled Exchanges

- Status - 早期研发中！

愿景：

- 离线加载 Web 内容
- 存储 + 分享页面给附近的人
- 更加快速加载

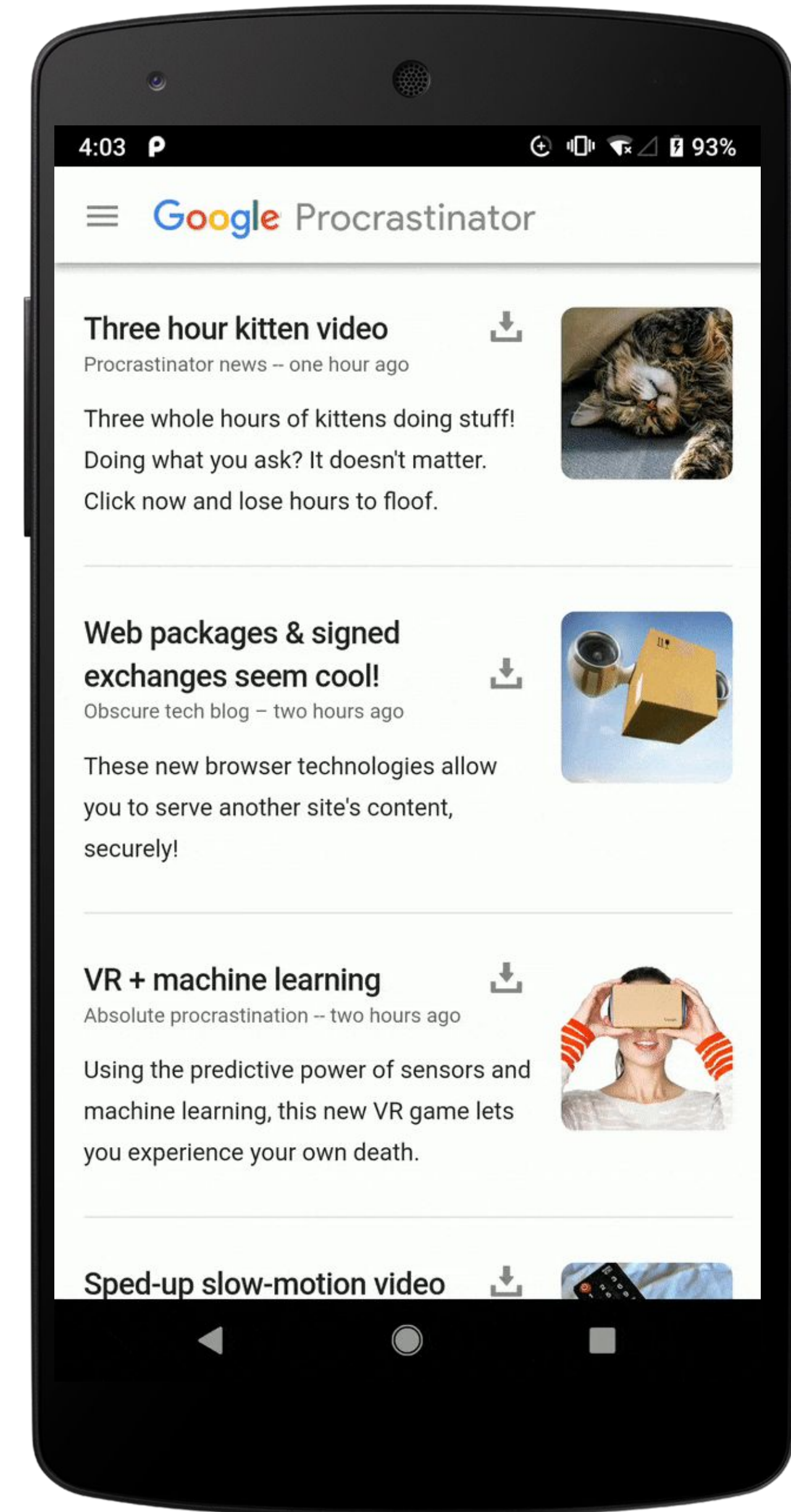


Bundled Exchanges

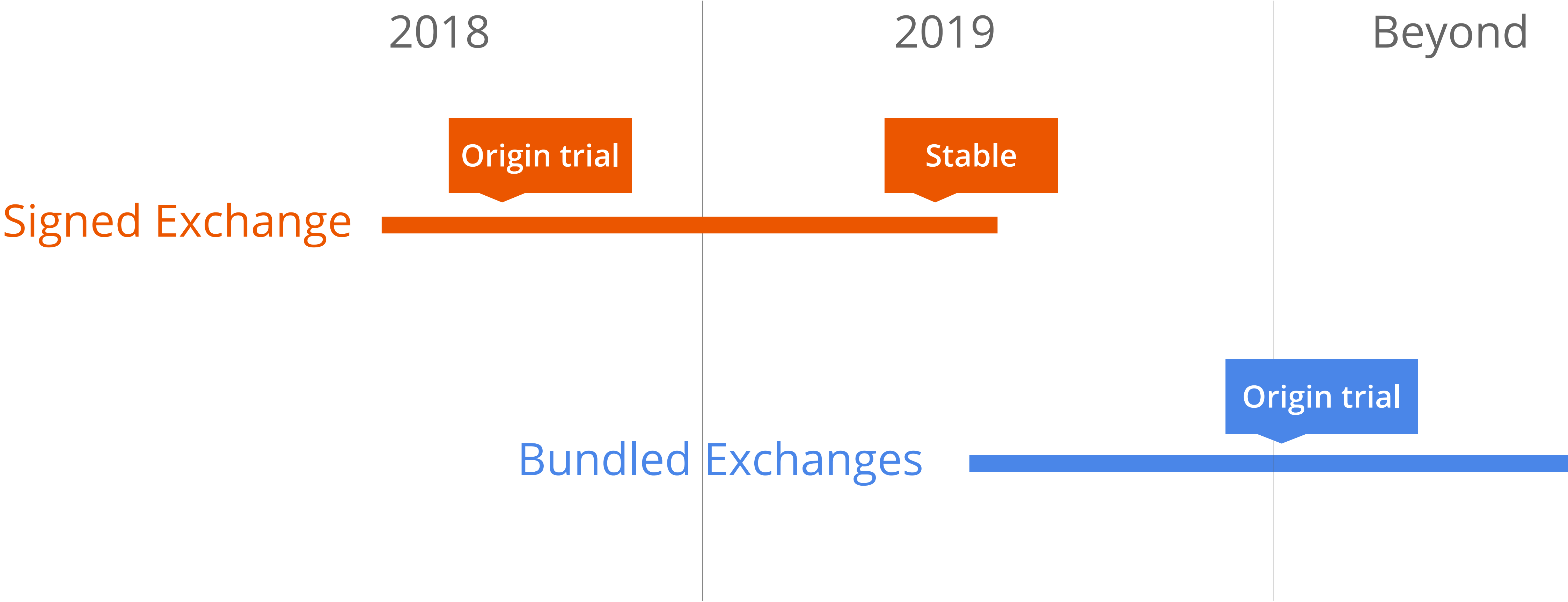
Demo: News-Reader PWA

离线加载多个站点的内容

Based on github.com/jakearchibald/site-sketches/reader



路线图



把 AMP 的学习经验带到 The Whole Web

