

Ecosystem Impacts of Web Content Syndication

Eric Rescorla
Mozilla
ekr@rtfm.com

June 6, 2019

1 Introduction

In the traditional Web setting, content that was logically associated with a given entity (e.g., a publisher) was served off of infrastructure that was under the control of or at least responsible to that entity. In many cases this meant that the entity would have their own servers, although increasingly it means that they use some kind of cloud hosting, whether that's in the form of the publisher's software running on cloud hardware or of outsourcing much or all of the publishing process as in the case of a CDN.

Regardless of the exact arrangements, the publisher ultimately was responsible for the behavior of the infrastructure serving the content. Indeed, from the perspective of the browser viewing the content, the servers providing the content *are* the publisher, which is why CDNs which serve HTTPS have certificates in the publisher's name — thus ensuring that the content is in the publisher's *origin* — rather than in the name of the CDN's domain.

Syndication systems such as AMP¹ fundamentally alter this relationship: instead of being served off infrastructure associated with the publisher, the syndicated content is served off of infrastructure associated with the syndicator. In some cases, this may be with the cooperation of the publisher but from a technical perspective, many of the mechanisms being proposed (e.g., Web Packaging²) would allow any syndicator to serve content under the rubric of the publisher without either them or the user having any interaction at all with the publisher (though of course the content might link back directly to the publisher's site).

The remainder of this document explores the ecosystem impacts of a widespread shift to this sort of content publication.

2 Ecosystem Impacts

2.1 Lowering Barriers

The major positive ecosystem impact of syndicated publishing is that it has the potential to lower the barriers to publishing content which is potentially highly popular. One common experience is that a given piece of content will “go viral” and then swamp the publisher's server (aka the “slashdot effect”). It is of course possible to manage this situation by using a CDN or some other third party service (e.g., hosting on Github pages), but that requires a certain amount of groundwork prior to experiencing the traffic spike.

To the extent to which the same entities which are the source of the traffic (e.g., Facebook and Twitter links or Google searches) are those which are doing syndication, then this potentially lowers ecosystem barriers to publishers who might experience unexpected popularity. This change seems like it is likely to be less important for existing publisher such as newspapers or magazines which already have clearer traffic patterns and mechanisms for handling flash crowds, but it might potentially help lower barriers for new entrants in the publishing ecosystem.

¹<https://developers.google.com/amp/>

²<https://tools.ietf.org/html/draft-yasskin-dispatch-web-packaging-00>

2.2 Web Platform Gatekeeping

One major risk of any syndication format is gatekeeping of the Web platform by the syndicators. This is not simply hypothetical: Google AMP has an extensive series of requirements³ that sites must conform to in order to be syndicated. For instance:

JavaScript is powerful, it can modify just about every aspect of the page, but it can also block DOM construction and delay page rendering (see also Adding interactivity with JavaScript). To keep JavaScript from delaying page rendering, AMP allows only asynchronous JavaScript.

AMP pages can't include any author-written JavaScript. Instead of using JavaScript, interactive page features are handled in custom AMP elements. The custom AMP elements may have JavaScript under the hood, but they're carefully designed to make sure they don't cause performance degradation.

Whether these restrictions are good or bad is beside the point: the end result is to radically shift the balance of power on the Web from one in which the publishers and the browsers determine which Web features are valuable to one in which the syndicators make that determination: publishers which don't conform to those standards are minimally at a performance disadvantage— because they are served off of the publisher's infrastructure rather than the syndicator's⁴— and potentially risk being disfavored by whatever ranking algorithms the syndicator uses. The end result is that a relatively small group of powerful syndicators is in a position to determine the direction of the Web.

2.3 WebCompat Lockin

Another risk of syndication is that it tends to favor the majority browser implementations and create negative experiences for users of minority browsers. Despite the existence of an impressive array of Web standards, not all browsers behave identically. There are a number of reasons for this, including:

- Some browsers implement features that others do not.
- In some cases the specification is imprecise in a way that leads to differing implementations.
- Implementations have defects that render them nonconformant.

The end result is that it takes effort to make content that has acceptable results on every browser, and in many cases sites deal with this by detecting which browser they are talking to— or which features it implements— and adjusting the content accordingly. A prime example of this is sites which have both mobile and desktop versions, but it is quite common to serve different content to (for instance) Chrome and Firefox.

Syndication pushes against this in two major ways. First, because the syndicator rather than the site decides what content to serve, the user is reliant on the syndicator to do whatever version or feature adaptation is required. This is made even more difficult by features such as Web Packaging, which make it hard for the syndicator to modify the site's content even if it wanted to.⁵ Second, the emphasis by syndicators on packaging size (for instance, AMP has a maximum size of 50KB for inline CSS) makes it hard for sites to package multiple content variants in the package for each browser.⁶

The end result of this is that syndication makes it even more likely that sites will develop content that works well in majority browsers but bad in minority browsers, further increasing lockin to those browsers. This increases the risk of browser monoculture.

³<https://amp.dev/about/how-amp-works>

⁴ This is a matter of both infrastructure cost and of the syndicator being able to preload syndicated content when it cannot preload non-syndicated content

⁵ It seems likely that Web Packaging will get even more rigid than it is not in response to mix-and-match attacks like those described by Martin Thomson (<https://docs.google.com/document/d/1ha00dSGKmjoeH2mRiG8FIA5sJ1KihTuZe-AXX1r8P-8/edit>)

⁶ Note that client-side adaptation is a move in the opposite direction of technologies such as client hints. Ironically, Chrome is a proponent of both client hints and Web packaging.

3 Summary

The apparent goal of syndication: fast Web pages at low cost to the publishers, is a worthy one, the current designs seem to come with a number of undesirable consequences. Most importantly, it tends to further centralize power and decision making in a few large publishers and client software developers, moving it away from smaller entities such as publishers, independent software developers, and users.

DRAFT