ESCAPE position paper

Rich Salz <rsalz@akamai.com> and Utkarsh Goel <ugoel@akamai.com>, Akamai Technologies.[1]

It is important to think of the system as a whole -- not only the individual, enabling technology (SXG) in this case, but also how we know or expect it will be used (in search engine result pages). While this is generally important -- should we use atomic bombs to excavate for the Interstate?[2] -- it is particularly important here, where the proposed technology strains, perhaps to the breaking point, the Web architecture that has to-date served so well. Specifically, it brings third-parties into the user/origin content delivery relationship. This architecture, ensconced in the terms *user agent* and *origin server*, dates back to RFC 1945, published in 1996.

Some will no doubt raise the issue of HTTP proxies and CDN's being third-parties from the user/origin perspective. These issues are not relevant for two separate reasons. First, the behavior of a proxy is constrained -- there are headers it must not modify, and specific error codes to be used when it fails, for example. By contract, the entire purpose of an SXG cache is to enable the unnamed third-party to act *as if* it is the origin server presenting content, and user-agents are to display the origin's identification. For the CDN issue, the origin has an out-of-band legal agreement with the CDN, and has contracted with them to act as its surrogate, even to the point of having it present a TLS identity representing the origin itself.

This position paper will therefore consider only SXG and some of its intended uses, and disregard other putative third-party content distributions. In light of this, we have several concerns.

The End-to-End Principle[3]

The End-to-end argument states that it is most efficient to avoid replication of endpoint functions (e.g., confirmation of transmission, confirmation of receipt) by intermediaries in the network. In the case of SXG, this inefficiency is not just about the timeliness of the transmission or utilization of network resources but also the conveyance of value: The origin gets value knowing the client has obtained the content.

In an origin-client world, that is conveyed automatically. In a CDN world, that information is conveyed as part of the CDN-origin business agreement. In an origin-SXG-client world, the SXG intermediary inherits that role or reduces the benefit to the origin to make the content available in the first place. One stated rationale for this is that the intermediary should not disclose to the

[1] Affiliations given for identification only; this is not an official position of Akamai Technologies.

[2] http://www.world-nuclear.org/information-library/non-power-nuclear-applications/industry/peaceful-nuclear-explosions.aspx

[3] http://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.pdf

origin that the client has “fetched” the content until it decides that the client has done so. The intermediary has now subsumed publication and read-disclosure to itself, in the interests of speed.

Purge before Expiration is important

There are countless reasons why it can be important to remove content before its intended expiration date. These include:

- DMCA takedowns
- “Right to be forgotten” rulings
- GDPR issues
- National issues such as firewalls or “unlicensed” content within their borders
- Inadvertent or malicious disclosure of pre-release data, such as mergers and acquisitions

These things happen all the time, especially at the major content aggregators. Browser source is readily available, and easily modified to ignore any SXG expiration check. It seems possible that an authority such as the Internet Archive or Wikipedia will offer such a browser in order to let users view historical (expired) information.

Related to this issue is the fact that revocation does not work on the Web. How are content producers expected to revoke SXG content? Even if there are readily available tools to do so, how will they know where to send the revocations? And can they be assured that the revocation will be honored? It seems quite possible that origins will only offer SXG content to the one or two parties that they trust, or have to trust. This will result in further separation of the Web into multiple classes.

CA’s as Content Approvers

In the WebPKI, CA’s function as identify verifiers. SXG requires a special certificate extension to be present. (Nit: it should be an extendedKeyUsage value.) The SXG certificate must also have subjectAltNames covering the origin’s domain name(s). At first glance, this seems to make the SXG CA be the same as the WebPKI CA; do we know of any commercial CA’s that will, for example, sign an SXG certificate for a site that uses LetsEncrypt?

We anticipate pressure on CA’s by entities that mistake use of a certificate with the making the issuer responsible for that content. It does not seem infeasible to imagine that a country bans a CA because an origin signed SXG content that was a news article about political unrest.

User privacy and data protection with SXG

Many origins and publishers make use of various security policies to maintain user-privacy. For example, some publishers prevent leaking of user data from cross-site scripting attacks while other publishers use content security policies to allow only certain third-party origins to interact with the first-party content. SXG-based content delivery currently does not come with well-established guidelines on how to impose the original publisher’s policies on the SXG content, leaving the user-privacy and data protection in the hands of republishers. We believe that this aspect should be discussed at the meeting.

User’s trust with re-publishers

It also remains unclear as to how a user should develop trust for a re-publisher’s content. Given Google’s dominance in the market for search engine and for hosting SXG-based content, many re-publishers could publish the same content under different origin names. As a result, its becomes difficult for a user to know which republisher of SXG content to trust. In fact, in some scenarios, a user may naturally be inclined to trust SXG-content that comes from Google’s caches, building an even stronger trust between an end-user and Google.

Perverse Economic Incentive

Similarly to AMP pages, the economic incentives to get SXG-based content in the search results carousel will provide unnatural pressure for publishers to agree to this kind of thing even though they might prefer more direct control of their content.

Getting us Stuck with This

Google is the world’s most popular search engine. It is also the provider of the world’s most popular desktop browser, and mobile. While all of the above issues nibble around the edges, we want to make it explicit that we are concerned with the dominant browser, when directed to the dominant search engine, to rapidly deploy and use, and act as a gravitational force, for SXG without enough time for the Internet community to consider it.

We[4] used to be very concerned about Microsoft and the so-called “embrace and extend” policy. We were worried about a single company controlling the client operating system, and the servers they connect to. For a number of reasons (including the rise of the Web and the rise of Linux), the reason for these concerns have evaporated. But as the statistics show, we seem to be again at the point where a single company is in a dominant position, and appears to be using this position to force the Internet to accept something it wants.

Connectivity and connection speed are both globally increasing. Facebook had TV ads about its long-flying gliders, SpaceX just launched 60 Internet-bearing satellites into orbit, and Amazon Web Services now offers satellite ground stations as a service (can true satellite capability be far behind?). Clearly many entities are thinking about how to increase connectivity and decrease latency. SXG, by contrast, is a throwback to the “store and forward” era.

[4] The IETF, and perhaps the W3C, community.