AMP SXG

Devin Mullins, Google
IETF 106

Past

We observed:

- folks feeling discontented with mobile web[1]
- content moving off the web onto native platforms
  - or never making it onto the web

We want the web to thrive because:

- (AMP hat) We’re children of the web.
- (Google hat) Search is only as useful as the web it searches.

[1]https://www.theverge.com/2015/7/20/9002721/the-mobile-web-sucks

Past

Many possible avenues for improvement, e.g.:

- Content creation
- Hosting
- Discovery
- Monetization
- Metrics

Approaches include AMP, Google Sites, Google My Business, Lighthouse, etc.

AMP: Intended properties

- Fast is easy to write; slow is hard to write.
- statically-guaranteed
  - UX
    - loading speed (mean & variance)
    - interaction speed and no jank
    - layout stability
  - security and privacy on cache
  - deeper integration
    - embedding
    - scroll to text

AMP Cache

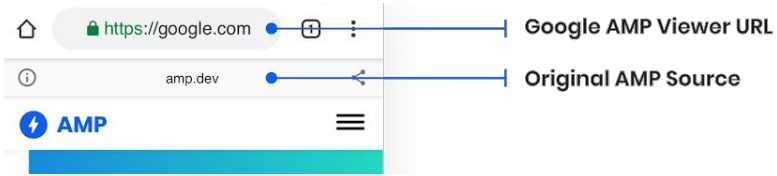
- static analysis
- predictably fast page loading
  - benefits from prefetching
    - requires privacy guarantee: prerender doesn’t leak user interest
      - requires first-party prefetching
        - requires security guarantee: sites on cache are isolated from each other

AMP URLs

Unfortunate consequence of the aforementioned constraints.

Problems include:

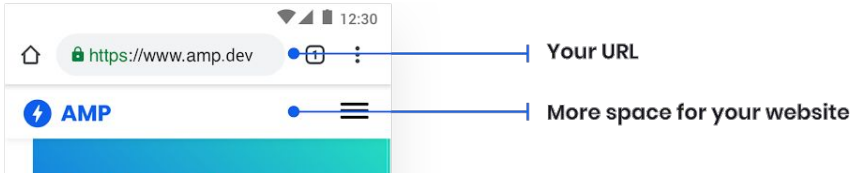
- confusing UX / **branding**
- wasted **screen real estate** due to AMP Viewer shell
- users/publishers have to **trust caches** not to make significant **modifications**
- problems with **cookies**, CORS, localstorage, indexeddb, websql…



Present: Signed exchanges

Fixes all that:

- intuitive UX / branding
- publisher has full screen real estate
- caches cannot make modifications
- publisher origin for cookies, CORS, localstorage, indexeddb, websql…



Present: Signed exchanges

… while still satisfying the static UX/security/privacy constraints:

- static assertion of the other UX guarantees (jank, layout stability, etc.)
  - AMP format
- predictable network speed
  - prefetch (incl. subresources) w/o leaking user interest to 3p
- isolated security contexts (protect publishers from the cache and each other)
  - signed origin

Present: AMP's minimal requirements of SXG

- content-based URL for display
- content-based origin for websec
- prenavigate[1] enables useful-subset render unblocked by network
  - "useful-subset render" ≈ render of a publisher-specified subset of the page, up to and including the entire page
  - "unblocked by" ≈ in parallel with
- graceful degradation if cross-origin trust[2] is not established
- ???

[1] https://github.com/w3c/resource-hints/issues/82

[2] https://tools.ietf.org/html/draft-yasskin-http-origin-signed-responses-08#section-4

Future: Continued improvement

- Caches can serve stale content.
- Publishers must sign per-cache variants.
- AMP is a requirement for various integrations.

Stale content

users/publishers have to trust caches not to serve significantly stale content

- improvement over status quo
  - though the potential cost of stale content is higher when origin is correct
- mitigated with signature `expires` (max 7d)
- mitigated with runtime checks

Per-cache variants

AMP SXGs link to per-cache URLs for subresources.
Publisher has to sign a package for each distributor.

- solved by:
  - signed subresource substitution[1]?
  - bundles?
  - sig-SRI[2]?

[1] https://github.com/WICG/webpackage/issues/347

[2] https://github.com/mikewest/signature-based-sri

AMP requirement

TODO: Generalize AMP requirement for various interaction surfaces.

- SXG prefetch whenever:
  - P(click) is high
  - E[cost of network bytes] is low
- Validate UX guarantees with metrics & feature policies[1]
  - e.g. First Contentful Paint, Largest Contentful Paint, Layout Stability, No oversized images, No sync XHR, etc.
- Deeper integration
  - 3p portals?
  - ScrollToTextFragment?

Thanks!

Devin Mullins (twifkak)