July 18, 2018

SERVER-SIDE COOKIE MANAGEMENT AND CONTROL

Matija Prekajski

Ruslan Kudubayev

Chandan Giri

Avi Mehta

Saurabh Mahajan

See next page for additional authors

Inventor(s)
Matija Prekajski, Ruslan Kudubayev, Chandan Giri, Avi Mehta, Saurabh Mahajan, Frederik Kemner, and Wilfred Yeung

SERVER-SIDE COOKIE MANAGEMENT AND CONTROL

In many instances, requests for additional content on a web page may be redirected through a content management system or content server, with a conversion cookie provided by the content management system or content server. The cookie contains an identifier for attribution to the request. A content publisher may add a script to corresponding pages on their website that sends a conversion ping or short request to the content management system (e.g. a request for a 1x1 pixel image). The conversion ping may include the cookie (e.g. in the request) and allows the conversion to be attributed to the request.

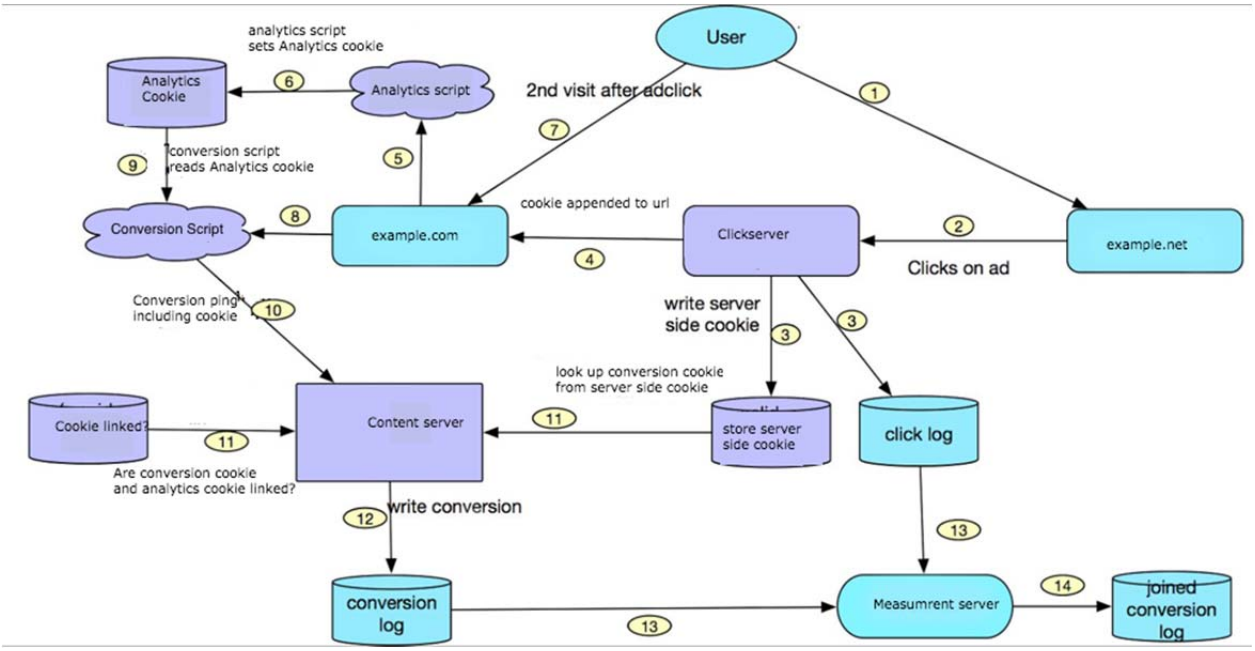
For example, a cookie may be set on a click of a hyperlink on the page, and read at a subsequent conversion time. However, depending on where these cookies are generated, they may not be useable by all browsers. In many implementations, the cookies are “first party cookies” and may be retained by the client for a period of time (e.g. 30 days). To allow identification of requests and data sharing across the system, various mechanisms include:

1. Auto-Tagging embedded content and identifiers within a parameter-value pair in a uniform resource locator (URL) request to publisher URLs. The request may include an identifier identifying the impression or click (e.g. user selection). The parameter is collected via an embedded script as discussed above, associating the cookie with the click.
2. Publishers can also link embedded content accounts.

In some implementations of web browsers, cookies may be prevented from being provided to third party content providers to provide additional user security. To support such implementations, some systems may use scripts such as those discussed above to insert identifiers into parameter-value pairs in URL requests, such as in short ping requests. This may be accomplished, in some implementations, via the following:

1. Storing conversion cookie server side in clickserver;
2. Storing identifier in new cookie on content provider domain;
3. Conversion javascript reading cookie; and
4. Conversion handing in server.

A flow chart illustrating one such implementation is provided and discussed below:



In some implementations, at step 1, a user or client device may visit a website, and at step 2, the user may click on an item of linked content (e.g. sending a request to the corresponding domain, via a content server). At step 3, a ClickServer or content redirector may generate a identifier to Conversion Cookie mapping for each eligible click. This storage may be online or distributed storage to provide speed and scalability, and the mapping may be stored for any period of time, such as up to 90 days.

At step 4, the identifier may be included as a parameter-value pair in a URL request to the destination corresponding to the selected content at step 2. Analytics scripts typically run on all pages on a domain. As such, when a user clicks on a link at step 2, it is very likely that the landing page retrieved at step 4 may include the analytics script. Once the landing page loads, at step 5, the analytics script may parse the parameter-value pair out of the URL and store it in a local cookie at step 5 for a conversion script to later use (e.g. step 9). In case a local cookie is already present, the analytics scripts may overwrite it in some implementations.

The analytics script may not store the cookie in the browser in some implementations. To make the cookie or identifier available a new cookie may be generated with the identifier, with an expiration date of a significant time (e.g. 90 days).

In some implementations, a user device and/or a publisher domain may be able to opt out of the analytics script or cookie management via an included code. The script may parse the

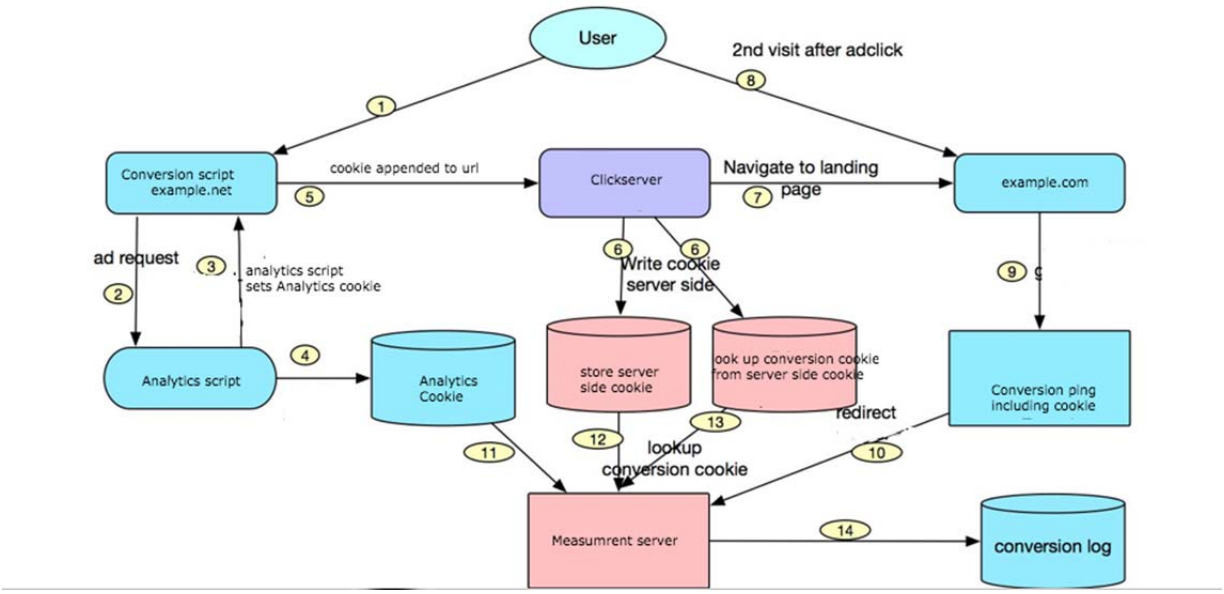
page code (e.g. HTML, Javascript, etc.) for the included code, and upon discovering the code, may perform no further functions of the flow chart above.

Multiple analytics scripts may be utilized on a single page, with each set to parse a different portion of a URL and/or generate an independent cookie. This may provide for multiple accounts or properties to be utilized. In some implementations, a checksum mechanism is used to ensure that the parameter or value is not reused. In some implementations, a linker may be utilized to provide for passing of cookies and identifiers between linked domains via identifiers included in URLs.

When a customer's site spans across multiple domains (eg.: foo.com , shopping.foo.com and bar.com), the analytics script may provide users with a way to share the some of the identifiers across various domains. For example, in case of foo.com and shopping.foo.com, a customer may choose to write the cookies to foo.com domain and thus make them available to both foo.com and its subdomain. For the case of foo.com and bar.com , the customer may utilize cross-domain cookie sharing. Once set up, any user visiting bar.com via a link from foo.com and vice-versa will pass the client identifiers and parameter-value pairs from one domain to another. As a result of this, the identifiers may be stored in the generated cookie on both the domains. Since the identifier is only transferred on a navigation from one domain to another, if the user does not navigate from a domain to another domain, the new identifier will not get stored on the other domains. This could lead to out-of-sync identifiers. Since the cookies are still shared between domain, subdomains, this issue only presents for domains which sync using the linker.

During the conversion ping, the conversion scripts may retrieve identifiers or cookies previously set, and include them in a request URL as parameter for a request to an analytics server.

In another implementation, conversions may be identified via server-side cookies. In one such implementation, on a click of a hyperlink, a request is sent to a clickserver, which generates a conversion cookie for server-side storage. On a subsequent conversion ping, if a conversion cookie is present on the client, it may be used; if a conversion cookie is absent, then the request may be redirected to the click server, which may perform a server-side search and retrieval for the stored cookie, and attribute the conversion based on domain associated with the cookie when generated. In the event no cookie may be found, then a new cookie may be generated. A flow chart illustrating this implementation is provided below:



Cookie generation may be performed differently depending on the location of the requesting device and a time since a most recent login to a trusted server. For example, if a user has logged in recently, then a server-side mapping may be accurate and a mapped cookie may be utilized.