TWO WAYS TO MEASURE DISPLAY AD VIEWABILITY

Douglas de Jager and Simon Overell, spider.io

Abstract

There are just two ways to measure whether display ad impressions are viewable. The two approaches are described here, with a survey both of the relevant intellectual property and of how the approaches are used by leading viewability measurement vendors. The authors agree with the 3MS committee that 2013 has seen the key hurdles removed for viewability measurement and 2014 will see media sales transacting on viewable impressions.

Introduction

Viewable impressions are coming. This has been announced by the 3MS (the consortium of trade bodies representing advertisers) and Google (the largest display ad network):

> *2013 is the year for overcoming all of the obstacles and bringing to life meaningful viewable impression measurement for display and video ads.*
> *(3MS, 2012)*

> *Advertisers are able to buy viewable impressions on the Google Display Network Reserve[...]. In the near future, Google will be making the Active View metric universal*
> *(Google, 2013)*

There is now no technical limitation to measuring the viewability of display ad impressions across all major desktop browsers, whether the ads are included inline with the webpage or confined to cross-domain iframes (MRC, 2013). The industry's willingness to move to a new billable unit has now also been made clear by Google.

There are two ways to think of any individual display ad impression being viewable: the active browser window encloses the ad impression; or the ad creative is being painted to the user's screen. Corresponding with these distinct ways of thinking about ad viewability, there are two distinct ways of measuring whether individual display ad impressions are viewable. In this paper we describe these two approaches. We describe the core limitations of each approach and the browser-specific workarounds to mitigate these limitations. We summarise the intellectual property issues that have surrounded ad viewability measurement and we highlight the approaches taken by the major vendors to measure viewability in cross-domain iframes.

It is clear that the industry has been mired by the technical challenges and intellectual property hurdles presented by viewable impressions. However, 2013 has seen clarity brought to both these issues. The MRC's view is that "by the end of December 2013, the MRC expects to lift its Advisory against transacting on viewable impression" (3MS, 2012). We conclude by noting that of the two methods to measure ad viewability, only the paint measurement method works in all major desktop browsers, both inline and in cross-domain iframes.

Method 1: The Geometric Method

Suppose that a webpage is larger than the dimensions of the browser window being used to view this webpage. Then some of this webpage's content will be enclosed by the browser window and some of this webpage's content will not. If any webpage content is enclosed by a browser window, then we may infer that this is currently fully viewable by the user of the computer—ignoring, for the moment, any questions relating to browser minimisation or background tabs. Any webpage content that is positioned outside of the browser window is not viewable by the user of the computer.

This forms the basis of the first method for measuring whether ad impressions are viewable. Measure the position and size of the ad impression; and measure the position and size of the browser window. With these four measurements one can determine how much of the ad impression is enclosed by the browser window.

If an ad impression is included inline within the host webpage, then these four measurements can be performed using JavaScript. For example, one can measure the scroll offset, which is the horizontal and vertical distance the user has scrolled his browser across the host webpage, using `window.pageXOffset` and `window.pageYOffset`. One can measure the size of the browser using `window.outerWidth` and `window.outerHeight`. One can measure the position of the top left corner of the ad impression using `element.offsetLeft`, `element.offsetTop` and `element.offsetParent`. One can also measure the size of the ad creative using `element.offsetWidth` and `element.offsetHeight`.

Cross-Domain Iframes

Typically the geometric position of an individual ad impression relative to the browser window can only be measured across ~65% of ad impressions.

This is because ~75% of display ad impressions are served within cross-domain iframes; and if an ad impression and the associated JavaScript is served within a cross-domain iframe, then the geometric position of the ad impression can typically only be measured in Firefox 3.6+ and Internet Explorer 6–10.

Let us consider. An iframe is an HTML element whereby one webpage can be embedded within another; and a cross-domain iframe is an iframe that is served from a different domain to the domain of the parent webpage. JavaScript served from within a cross-domain iframe cannot measure the position of the associated ad impression relative to the host webpage because of a browser security policy called *the same origin policy*. In much the same way that one would not expect any information leakage between the homepages of Google.com and Yahoo.com, browsers prevent information leakage between cross-domain iframes and the parent webpages in which they are embedded. *The same origin policy* is the browser security policy that prevents this information leakage (W3C, 2010). It requires browsers to ensure that “documents retrieved from distinct origins are isolated from each other.” Specifically, JavaScript served within a cross-domain iframe cannot interrogate the host webpage to determine where within the host webpage the cross-domain iframe has been embedded.

Firefox 3.6+

In Gecko 1.9.2 two non-standard API methods were added that provide an alternative way to measure the geometric position of an ad impression from within a cross-domain iframe in Firefox 3.6 and above. These two non-standard API methods change the positional frame of reference from the host webpage to the screen.

The two non-standard API methods are `window.mozInnerScreenX` and `window.mozInnerScreenY`. These provide the location of the iframe relative to the screen (MDN, 2013; MRC, 2013). These measurements can be compared to the position of the browser window relative to the screen, available through the standard JavaScript API calls, `window.screenX` and `window.screenY`.

This approach has a limitation over the standard geometric method insofar as the iframe does not have access to the size of the browser viewport. This must be approximated from the outer size of the browser window—approximating, for example, the size of the browser bar, status bar, bookmarks bar and plugin menus.

Internet Explorer 6–10

Internet Explorer 6-10’s event model makes the co-ordinates of the mouse pointer available with respect both to the screen and to the current document whenever an event is triggered—this includes when the current document is a cross-domain iframe. By subtracting one measurement from the other one can calculate the position of the current document relative to the screen. This in turn allows for the screen to be used as a positional frame of reference when measuring the viewability of an ad impression from within a cross-domain iframe.

Mouse positions can be polled at regular intervals across Internet Explorer 6-10. This is achieved by artificially triggering events using Internet Explorer’s `fireEvent` API, and reading the `offsetX`, `offsetY`, `screenX` and `screenY` values from the generated event object.

Note that by using this method one must access the co-ordinates of the mouse pointer wherever it is on the screen—not just over the browser window—regardless of whether the browser is in focus or even minimised (spider.io, 2012). This is a known security vulnerability, which has now been closed in Internet Explorer 11 (Hachamovitch, 2012). The closure of this security vulnerability means that it is not possible to use the geometric method in Internet Explorer 11 to measure ad viewability from within cross-domain iframes.

Method 2: The Paint measurement Method

Let us reconsider what it means for an ad impression to be viewable.

If we abstract away entirely from the geometric position of the ad creative, then there is an alternative, more literal, definition of what it means for an ad impression to be viewable: the ad creative is being painted to the user’s display. This definition forms the foundation for the second method for measuring whether individual ad impressions are viewable (de Jager, one right way, 2012).

When browsers paint content to the display they dedicate different resources (memory, processing, etc.) to the painted content than to content that is not being painted to the display. This is increasingly the case as web applications become more resource-hungry and as mobile devices are increasingly being used to access the web. These trends put growing demands on browser vendors to optimise browsers—increasingly biasing resource allocation to just those elements being painted to the display.

An example of such an optimisation may be seen where elements within the browser viewport are redrawn with a higher frequency than elements outside the viewport. This redraw rate can be measured by calculating the frame progression rate of an Adobe Flash application embedded in the webpage. By measuring the change in frame rate of a given Flash application one can determine whether the application is in view. Flash applications can then be constructed that cover only a single pixel, allowing one to measure whether individual pixels are in view. These pixels can be placed on opposite corners of an ad creative (see Figure 1)—so that when both pixels are in view, the whole ad is in view.

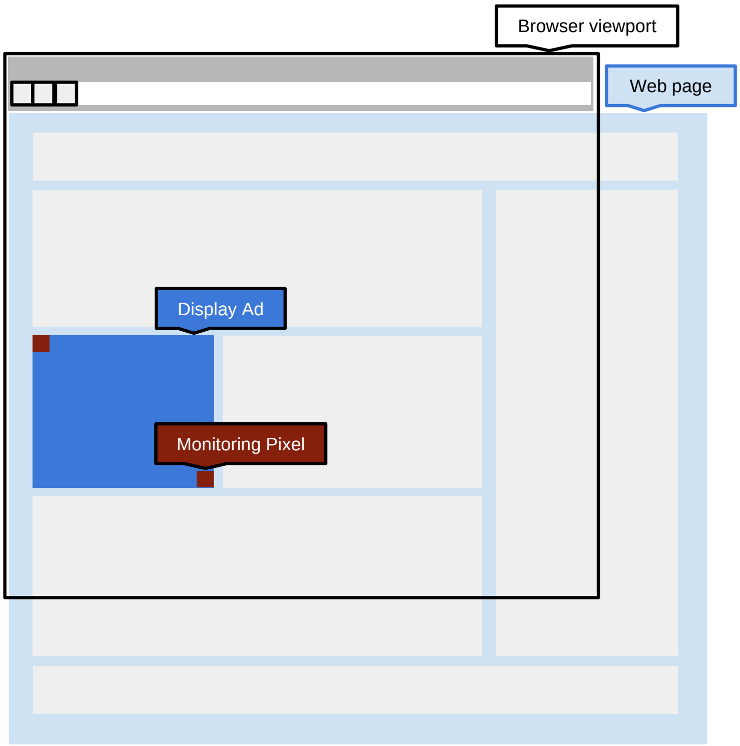


Figure 1: A display ad impression with two monitoring pixels

By increasing the number of pixels monitored and by moving the monitored pixels around an ad creative one can determine what percentage of the ad is viewable.

By monitoring APIs exposed by the browser, rather than by Flash, one can remove the requirement for Flash to be installed. For example, one may determine the redraw rate in Firefox using the `window.mozPaintCount` API. This provides a count of how many times the current document has been painted to the screen. Rather than using pixel-sized Flash applications, one can create pixel-sized test iframes. The paint count of the test iframe is read using the value of the `window.mozPaintCount` API. Periodically the appearance of the test iframe is changed (for example, through a change in colour or opacity) to induce a repaint. After a short period, if the new paint count is greater than the original paint count, then the test iframe is in view.

This method has some clear advantages over the geometric method. It reports whether ads are actually being painted to the display rather than whether they are simply positioned within the viewport but potentially obscured by other elements. Further, as no geometric information is required, the paint measurement method does not fall foul of *the same origin policy* when in cross-domain iframes. In fact, this is the only way to measure the viewability of individual ad impressions across all major desktop browsers from within cross-domain iframes.

Limitations

A key limitation of using the paint measurement method is that there are no standard browser API calls to determine which pixels are currently being painted and which are not. This means that independent implementations are required for Internet Explorer, Chrome, Firefox, Safari and Opera.

Furthermore, Chrome, Safari and Opera most readily expose this information via APIs in Adobe Flash. Interrogating this information does not require that the ad creative be Flash across Chrome, Safari and Opera. It requires only that the Flash Player be installed.

It is also not trivial to use the paint measurement method to measure whether 50% of the ad is in view—as required for the proposed 3MS viewability standard. The method allows one to interrogate whether any part of a region is being painted to the display, but the method does not immediately make available the dimensions of what is viewable. When the ad is only partially in view (Figure 2), one strategy is to move multiple pixel-sized regions around the ad in both dimensions to search for the points at which the viewport intersects with the ad creative.

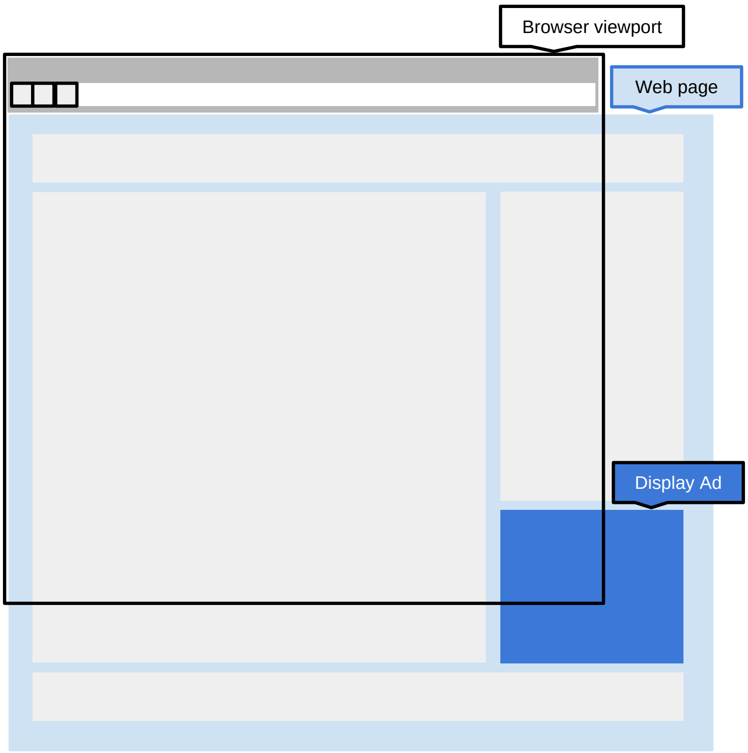


Figure 2: A partially viewable display ad impression

The Page Visibility API

Up to this point we have only considered ads contained in browsers where the browser window is neither minimised nor in a background tab. The Page Visibility API was developed to enable web developers to reduce CPU-intensive JavaScript functions in these cases (Mann & Jain, 2013). In addition, in Chrome, the Page Visibility API allows one to determine whether a webpage has been *prerendered*—when the browser pre-emptively fetches and renders an additional webpage in advance of the user actually navigating to this webpage.

The Page Visibility API is now an accepted standard and present in the latest version of all major desktop browsers: Internet Explorer 10+, Chrome 14+, Firefox 10+, Safari 7+ and Opera 12+. Determining whether ads are minimised or in background tabs via the Page Visibility API is independent of the geometric and paint measurement methods. Most viewability measurement vendors will use the Page Visibility API where it is available—first appealing to the Page Visibility API to determine whether any part of the whole webpage is viewable, and then applying either the geometric or paint measurement method to determine whether a specific part of the webpage is in view.

SafeFrame

A discussion of viewable impressions would not be complete without a mention of SafeFrame. The SafeFrame technology—proposed by the IAB—provides a standardised API-enabled iframe into which ads can be injected (iab, 2013). In addition to protecting publishers from malicious advertisers, it provides an API whereby various information, including viewability status, can be passed to the advertiser. Viewability information provided via the SafeFrame API is collected inline and passed into the iframe. If SafeFrame is used to pass viewability measurements, then either of the viewability measurement methods described in this paper will still need to be implemented.

A fundamental problem with the SafeFrame API is that the publisher controls it. This gives publishers full control over the viewability information provided to advertisers. If advertisers want to verify the information that they are passed via the API they will need to include their own independent viewability measurement technology capable of working in cross-domain iframes. This and other issues with SafeFrame are discussed in (de Jager & Overell, IAB's SafeFrames Review, 2013).

Key Intellectual Property

Measurement methodologies for determining the viewability of display ad impressions have been the centre of a series of intellectual property disputes. Both methods described in this paper are covered by intellectual property claims.

comScore owns a family of patents relating to the geometric method of measuring viewability. These were purchased from Nielsen following a patent dispute in 2011 (Nielsen, 2011). comScore have since pursued Integral Ad Science, DoubleVerify and Moat for patent infringement (Griffith, Moat fires back, 2012). Integral Ad Science and DoubleVerify have now settled, though the Moat case continues (Griffith, Moat continues, 2013) (Griffith, Yikes, 2013).

spider.io owns a family of pending patents, with a priority date of June, 2011, describing the paint measurement method (de Jager, Overell, & Hodgson, Advertisements in view, 2011).

Comparing Vendor Approaches to Viewability Measurement

The following vendor comparison is based on a review of accreditation disclosures, patent applications, press releases and other publicly available material (Figure 3). We are confident that the following is correct as of the time of writing—however, if anyone can demonstrate that we are in any way mistaken, we will correct immediately. It is our firm belief that our industry will benefit greatly from increased transparency.

The viewability products of Google, comScore, spider.io and DoubleVerify are accredited by the MRC (MRC, 2013). Google only provides viewability measurements in cross-domain iframes when ads are contained in the webpages of its partner publishers. comScore and DoubleVerify use the geometric method in Internet Explorer 6–10 and Firefox. spider.io use the paint measurement method in all browsers.

Similar to comScore, Alenty uses the geometric method in Internet Explorer 6–10 and Firefox. Additionally Alenty uses the paint measurement method in Chrome and Safari.

Moat and Integral Ad Science are currently under review by the MRC (MRC, 2013). They use the paint measurement method in Internet Explorer and Chrome, and the geometric method in Firefox. Integral Ad Science also uses the paint measurement method in Opera and Safari.

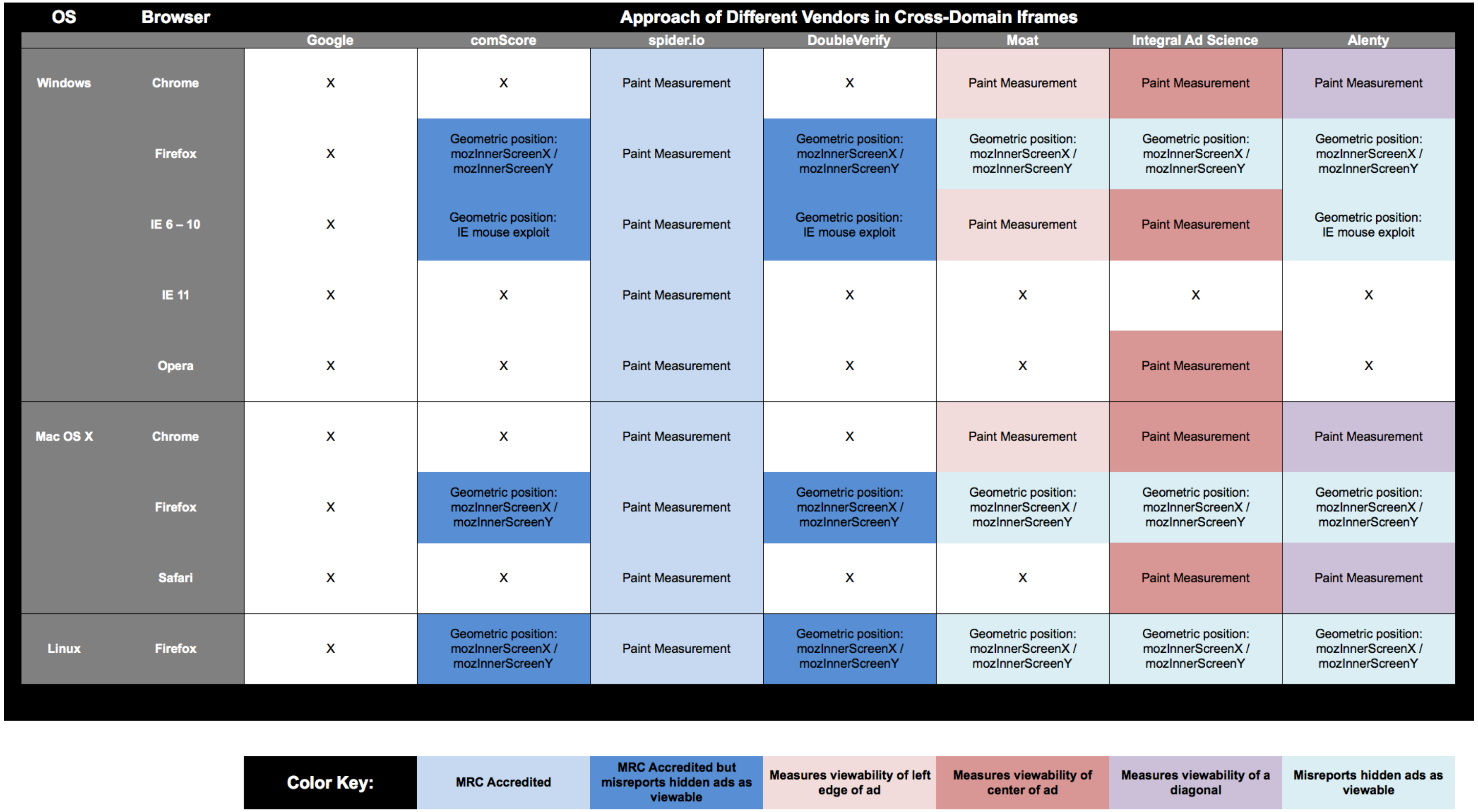
spider.io employs non-Flash implementations of the paint measurement method for Internet Explorer 8+ and Firefox. Other vendors using the paint measurement method always interrogate Flash APIs. All vendors measuring viewability in Chrome, Safari and Opera require the Adobe Flash Player to be installed.

There are three points worth noting across the surveyed methods: (1) measurement of the 3MS’s 50% metric; (2) Flash requirement; and (3) misreporting of hidden ads as being viewable.

Calculating the 3MS’s 50% metric when using the paint measurement approach is not trivial, and so most vendors measure an alternative metric. Integral Ad Science measures whether a static pixel in the middle of the creative is viewable. Moat measures whether a static pixel in the middle of the left edge of the creative is viewable. Alenty dynamically moves a pixel along the ad creative’s diagonal and measures whether the moving pixel is viewable. Figure 2 illustrates why these metrics are different from the 3MS’s 50% metric—showing a situation where only 26% of an ad is in view, but the central, central-left or moving pixel is reported as being viewable. spider.io directly measures the 3MS’s metrics by dynamically determining where the viewport bisects the ad in both dimensions.

As things stand currently, all vendors measuring viewability in Chrome, Safari and Opera require the Adobe Flash Player to be installed—though efforts are underway to remove this requirement. This is particularly important for measuring ad viewability from within cross-domain iframes across mobile devices. spider.io already employs non-Flash implementations of the paint measurement method for Internet Explorer 8+ and Firefox. Other vendors using the paint measurement method always interrogate Flash APIs.

A concern for vendors employing the geometric method is that it may not always be appropriate to infer from the position of the ad impression that the ad is actually viewable. This is of particular importance when ads are fraudulently hidden by publishers—in 0x0 iframes, or by setting opacity to zero, or by obscuring ads with other HTML elements. spider.io estimates that this affects 2% of ad impressions (spider.io, 2013).



OS	Browser	Google	comScore	spider.io	DoubleVerify	Moat	Integral Ad Science	Alenty
		Approach of Different Vendors in Cross-Domain Iframes						
Windows	Chrome	X	X	Paint Measurement	X	Paint Measurement	Paint Measurement	Paint Measurement
	Firefox	X	Geometric position: mozInnerScreenX / mozInnerScreenY	Paint Measurement	Geometric position: mozInnerScreenX / mozInnerScreenY	Geometric position: mozInnerScreenX / mozInnerScreenY	Geometric position: mozInnerScreenX / mozInnerScreenY	Geometric position: mozInnerScreenX / mozInnerScreenY
	IE 6 – 10	X	Geometric position: IE mouse exploit	Paint Measurement	Geometric position: IE mouse exploit	Paint Measurement	Paint Measurement	Geometric position: IE mouse exploit
	IE 11	X	X	Paint Measurement	X	X	X	X
	Opera	X	X	Paint Measurement	X	X	Paint Measurement	X
Mac OS X	Chrome	X	X	Paint Measurement	X	Paint Measurement	Paint Measurement	Paint Measurement
	Firefox	X	Geometric position: mozInnerScreenX / mozInnerScreenY	Paint Measurement	Geometric position: mozInnerScreenX / mozInnerScreenY	Geometric position: mozInnerScreenX / mozInnerScreenY	Geometric position: mozInnerScreenX / mozInnerScreenY	Geometric position: mozInnerScreenX / mozInnerScreenY
	Safari	X	X	Paint Measurement	X	X	Paint Measurement	Paint Measurement
Linux	Firefox	X	Geometric position: mozInnerScreenX / mozInnerScreenY	Paint Measurement	Geometric position: mozInnerScreenX / mozInnerScreenY	Geometric position: mozInnerScreenX / mozInnerScreenY	Geometric position: mozInnerScreenX / mozInnerScreenY	Geometric position: mozInnerScreenX / mozInnerScreenY

Color Key:	MRC Accredited	MRC Accredited but misreports hidden ads as viewable	Measures viewability of left edge of ad	Measures viewability of center of ad	Measures viewability of a diagonal	Misreports hidden ads as viewable

Figure 3: A summary of the different capabilities of the major viewability vendors when measuring ads in cross-domain iframes to the best of the authors' knowledge.

Conclusion

In this paper we have presented the only two methods for measuring the viewability of display ad impressions. We have identified which of these two methods are used by the leading viewability measurement vendors in cross-domain iframes. Of these two methods, only the paint measurement method provides coverage across all major desktop browsers in cross-domain iframes.

Viewability measurement has matured over 2013, as documented by the MRC's accreditation process. Issues surrounding intellectual property are being clarified and market leaders are pushing for the 3MS standard. Looking to the future, we believe that the display advertising ecosystem can now realise the 3MS prediction: *On January 1, 2014, media sales transactions are expected to be conducted on a viewable impression measurement standard* (3MS, 2012).

Bibliography

3MS. (2012). *Making Measurement Make Sense and the First of Five Solutions: The Shift to Viewable Impression*. (Making Media Make Sense) Retrieved September 26, 2013, from Making Measurement Make Sense: http://www.measurementnow.net/faqs.html

de Jager, D. (2012). *There are two ways to measure ad viewability. There is only one right way.* Retrieved September 26, 2013, from spider.io: http://www.spider.io/blog/2012/12/there-are-two-ways-to-measure-ad-viewability-there-is-only-one-right-way/

de Jager, D., & Overell, S. (2013). *IAB's SafeFrames Review*. Retrieved September 26, 2013, from Exchange Wire: http://www.exchangewire.com/blog/2012/12/05/iabs-safeframe-review-by-dr-douglas-de-jager-founder-ceo-dr-simon-overell-chief-scientist-of-spider-io/

de Jager, D., Overell, S., & Hodgson, B. (2011). *Patent No. US20120324098 A1*. US.

Google. (2013). *Active View*. Retrieved September 26, 2013, from Google Ads | Ad Innovations: http://www.google.com/ads/innovations/activeview.html

Griffith, E. (2012). *Let the ad viewability games begin: Moat fires back at comScore*. Retrieved September 26, 2013, from PandoDaily: http://pandodaily.com/2012/10/12/let-the-ad-viewability-games-begin-moat-fires-back-at-comscore/

Griffith, E. (2013). *Two startups settle patent case with ComScore as Moat continues to fight*. Retrieved September 26, 2013, from PandoDaily: http://pandodaily.com/2013/07/02/two-startups-settle-patent-case-with-comscore-as-moat-continues-to-fight/

Griffith, E. (2013). *Yikes! comScore CEO assaults a Moat employee during patent deposition*. Retrieved September 26, 2013, from PandoDaily: http://pandodaily.com/2013/09/11/yikes-comscore-ceo-assaults-a-moat-employee-during-patent-deposition/

Hachamovitch, D. (2012). *Update to Alleged Information and Security Issue with Mouse Position Behavior*. (Microsoft) Retrieved September 26, 2013, from IEBlog: http://blogs.msdn.com/b/ie/archive/2012/12/13/update-to-alleged-information-and-security-issue-with-mouse-position-behavior.aspx

iab. (2013). *SafeFrame*. (The Interactive Advertising Bureau) Retrieved September 26, 2013, from iab.

Mann, J., & Jain, A. (2013). *Page Visibility*. (World Wide Web Consortium) Retrieved September 26, 2013, from W3C: http://www.w3.org/TR/page-visibility/

MDN. (2013). *Window.mozInnerScreenX*. (Mozilla) Retrieved September 26, 2013, from Mozilla Developer Network: https://developer.mozilla.org/en-US/docs/Web/API/window.mozInnerScreenX

MRC. (2013). *Accredited Services and Services Under Review*. Retrieved September 26, 2013, from Media Rating Council: http://mediaratingcouncil.org/Accredited%20Services.htm

MRC. (2013). *MRC Accredited Viewable Impression and/or Ad Verification Vendors: Enhanced Description*. (Media Ratings Council) Retrieved September 26, 2013, from Media Rating Council: http://mediaratingcouncil.org/Viewability%20Verification%20Disclosure%20August%201,%202013%20Final.pdf

Nielsen. (2011). *NIELSEN AND COMSCORE SETTLE PATENT DISPUTES*. (Nielsen) Retrieved September 26, 2013, from Nielsen Press Room: http://www.nielsen.com/us/en/press-room/2011/nielsen-and-comscore-settle-patent-disputes.html

spider.io. (2013). *At Least Two Percent of Monitored Display Ad Inventory is Hidden*. Retrieved September 26, 2013, from spider.io: http://www.spider.io/blog/2013/04/at-least-two-percent-of-monitored-display-ad-exchange-inventory-is-hidden/

spider.io. (2012). *Internet Explorer Data Leakage*. Retrieved September 26, 2013, from spider.io: http://www.spider.io/blog/2012/12/internet-explorer-data-leakage/

W3C. (2010). *Same Origin Policy*. Retrieved September 26, 2013, from World Wide Web Consortium: http://www.w3.org/Security/wiki/Same_Origin_Policy