KEEPING THE INTERNET
OPEN • INNOVATIVE • FREE

www.cdt.org

CENTER FOR DEMOCRACY
& TECHNOLOGY

1634 I Street, NW
Suite 1100
Washington, DC 20006

P +1-202-637-9800
F +1-202-637-0968
E info@cdt.org

POSITION PAPER: TRACKING PREFERENCE BEYOND THE UA

22 October 2012

W3C Workshop on "Do Not Track and Beyond", 26-27 November 2012
Joseph Lorenzo Hall (joe@cdt.org)

While aimed squarely at defining the technical and compliance requirements around a protocol for user agents (UA) that speak HTTP, the Do Not Track (DNT) header from w3c's Tracking Protection Working Group (TPWG)[1] will need to be set by users interacting with a user interface (UI) and it will need to work in the complex environment of applications built using HTML5. This raises a number of privacy-relevant questions across w3c Working Groups that the TPWG — or the new Privacy Interest Group (PING)[2] — may be able to advance, while not straying from the scope of charters that specify flexible UI design.

I. One DNT Preference, or One for Every UA?

Platform-level implementations for DNT-like tracking preference expression (TPE) settings are already in production. Apple's iOS 6 includes a "Limit Ad Tracking" (LAT) setting that, when set, signals to applications that only a limited set of restricted uses are permitted for a unique ubiquitous identifier on that platform, the "Identifier for Advertisers" (IFA).[3] As LAT is a more general type of tracking preference expression, it is not set at the level of individual user agents (browsers, apps, etc.) but at the OS level. It is not hard to imagine a more granular analog where every application on the iOS platform has a tracking preference setting. This is similar to what iOS does for location sharing preferences (*See:* Fig. 1), and iOS has extended this to other types of data sharing preferences including Contacts, Calendars, Reminders, Photos, etc.

Are there considerations at the platform-level within the purview of TPWG when TPE becomes pervasive? I have two broad classes of concerns with intra-platform TPE. First, they may work, like IFA, as a cross-party "super-cookie"; e.g., IFA is the same value for every application on a device until the device settings are reset. This would be equivalent to, for example, desktop browsers returning a unique identifier to each site visited upon request (until browser uninstall). This drastically increases the potential for server-side sharing of browsing-related activities per device without any notice to the user.

---

[1] *See:* http://www.w3.org/2011/tracking-protection/

[2] *See:* http://www.w3.org/Privacy/

[3] G.S. Hans and Joseph Lorenzo Hall, "Apple iOS 6 and Privacy", Center for Democracy & Technology Blog, 1 October 2012, available at: https://www.cdt.org/blogs/0110apple-ios-6-and-privacy-0

Second, with intra-platform TPE, an IFA-like value is allowed for a narrow range of permitted uses but it is unclear how those restrictions can be ensured or, equivalently, how users can convince themselves parties only engage in permitted uses. Should compliance be mechanistic? That is, the OS might refuse to provide the IFA value unless the requesting party attests that it is engaged in a specific permitted use? Or should compliance be controlled by policy? That is, the platform controller (Apple, in the case of iOS) might identify calls to the IFA query function during app certification and require developers to attest that the uses they engage in are from the set of permitted uses.

II. Tracking Preference Expression in HTML5

In a similar vein, we at CDT have been thinking increasingly about privacy issues in HTML5 and to what extent tracking preference expression and other privacy concerns may suffer if HTML5 WGs operate without a common privacy model or some level of coordination. HTML5 has a dizzying array of APIs,[4] many of which implicate core privacy interests of users. For example, clearly HTML5 elements such as the Calendar API and Contacts API will mediate potentially privacy-sensitive user interactions as they deal with sensitive data (time-location data and social network data/personal contact details). However, more generic and more powerful HTML5 elements exist, such as WebRTC — for peer-to-peer in-browser audiovisual interaction — and Web Intents — which allows seamless remote computation on local resources. These can create what appear to be local user interactions but that are fulfilled by remote or "cloud" infrastructure. How does a user specify that they do not want "local" (or user-contributed) resources to be used or tracked for certain activities? The familiar debate of "collect vs. use" raised in TPWG discussions becomes somewhat more serious when very rich user content and activities applied to that content is at issue (e.g., tracking the exact steps a user employs to edit a photograph using Web Intents functionality that provides a remote image editing application).

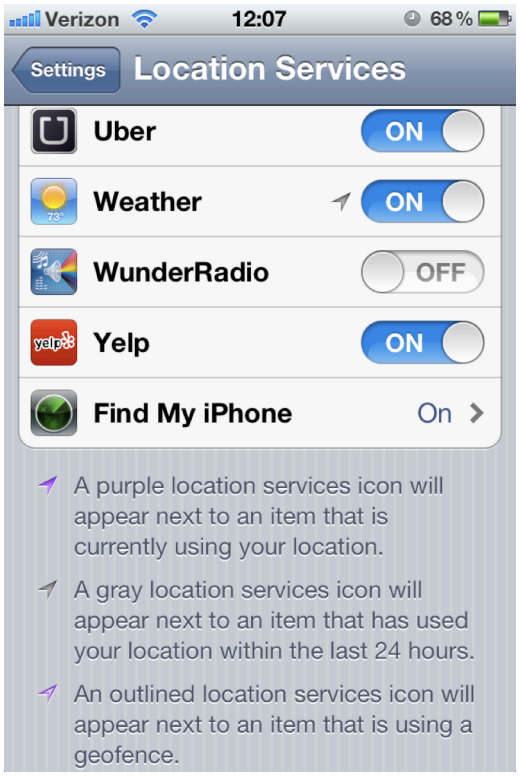

Fig. 1: iOS' "Location Services" preference pane showing granular control and notice iconography.

The need here is for a coherent and sensible model for how privacy protection might work in web applications built with HTML5 APIs. Unfortunately, the independent development of HTML5 APIs means that understanding the privacy implications across the suite of APIs is exceedingly difficult. We hope to discuss these more compounded privacy tensions in rich web applications at November's workshop.

[4] Erik Wilde, "HTML5 Landscape Overview", dretblog, 18 October 2012, available at: http://dret.typepad.com/dretblog/html5-api-overview.html