

October 19, 2015

Subscription-Less Web Push Framework
[draft-chiussi-webpush-subscription-less-framework-01](#)

Abstract

Subscription is an integral part of the current Web Push service. However, the current explicit subscription requirement makes it difficult to use web push in a number of use cases where there may be a need to reach User Agents that are not subscribed to the Web Push service. In addition, the current Web Push subscription model does not provide a mechanism to achieve coordination and control among subscriptions associated to different applications. This document describes a framework for making subscription more flexible and solve these issues.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/1id-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Terminology	4
2.	Subscription-Less Web Push Use Cases	4
3.	Subscription Management Across Applications	6
4.	Web Push Subscription Authority	7
5.	Security Considerations	7
6.	IANA Considerations	7
7.	References	8
7.1	Normative References	8
7.2	Informative References	8
	Authors' Addresses	8

1. Introduction

The notion of subscription is one of the pillars of the current Web Push service [I-D. [draft-thomson-webpush-protocol](#)][[W3C API](#)]. There are good reasons to require explicit subscription from the User Agent (UA) [[RFC6973](#)]. In fact, by its very nature, Web Push is an invasive service, which requires some form of explicit acceptance by the UA and some form of regulation by the Push Service to make sure that the push capabilities of different applications do not proliferate in such a way that the volume and invasiveness of push traffic becomes uncontrollable.

The Web Push explicit subscription model as it is currently defined, however, has two limitations.

First, the requirement for explicit subscription prevents, or at least makes it awkward and potentially ineffective, the use of Web Push in an increasing number of relevant use cases in which there is a need for a mechanism to reach User Agents in a timely fashion, but it is unknown whether all the User Agents that need to be reached in a certain location have subscribed to any Web Push service. For similar reasons, the explicit subscription requirement limits the usefulness of Web Push in broadcast use cases.

Second, the Web Push explicit subscription model as it is currently defined, does not provide a mechanism to achieve or impose coordination among subscription traffic generated by different applications. As a consequence, the Web Push mechanism is not scalable to a large number of subscriptions to different applications. Although this is intentional in the design at this time, it is recognized as rather crude constraint that does not quite address the control of the total volume of Web Push traffic across applications directed to the same UA, which is ultimately what matters from a UA perspective, and as such it is the most important factor that defines a satisfactory User Experience with Web Push. Some provisions to ameliorate this problem are included in [I-D. [draft-thomson-webpush-protocol](#)], but more comprehensive mechanisms for a global management of Web Push traffic across applications still need to be defined to make Web Push as useful and usable as possible.

Both these current limitations point towards the need to define a trusted Web Push Subscription Authority (WPSA), in charge of managing the Web Push subscription traffic "globally," i.e., across all applications, directed to each UA at any given location.

If such a WPSA is trusted, it can also be used to provide a "subscription-less" Web Push to accommodate the additional use cases

mentioned above. It should be noted that such a subscription-less Web Push does not mean uncontrolled Web Push. Quite the contrary, the WPSA is in charge of controlling the global Web Push traffic to each UE, and thus can block or throttle offending subscription-less Web Push traffic.

In both cases, the WPSA can apply policies that can be rather elaborate and globally defined, either across applications, or depending on the urgency and criticality of reaching the UA with a Web Push with a certain service in a given situation.

This document defines a Web Push subscription framework using a trusted WPSA to provide both a subscription-less form of Web Push and a mechanism for subscription management that achieves coordination and control among applications.

The two main challenges in the design of WPSA are the definition of trusted authority and the architecture of the WPSA to make it scalable and applicable on a local basis.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

This document uses the terminology defined in [I-D. [draft-thomson-webpush-protocol](#)].

In addition, this document uses the following terms.

- o Web Push Subscription Authority (WPSA): A trusted entity in charge of managing subscriptions across applications and enabling subscription-less Web Push, based on globally defined policies.

2. Subscription-Less Web Push Use Cases

Several emerging popular use cases need a mechanism to push information to a set of UAs. Because of the characteristics of these use cases, it is in general unknown or irrelevant whether the UA has subscribed to any Web Push services.

These use cases require a form of Web Push that does not depend on explicit subscription, which we refer to as "subscription less" Web Push.

At least four such uses cases come to mind.

1. Local Emergency, Alert, and Urgent Notification Services. These may include hyper local services in smart building or smart venues such as amber alert, local emergencies, crowd management, missing person, and other similar services.

A common characteristic of these service is that the information to be pushed has a very clear "value" to the user and needs to be delivered in a timely fashion. In other words, the information is so valuable and timely for the user that it is clear that the user is willing to receive unsolicited traffic and trade some privacy for the benefit of receiving the information in real time, regardless of subscriptions. These services are rapidly proliferating and user are starting to expect them.

Clearly, requiring subscription for such services would not be practical. An unsubscribed user may even be unaware that the services exist in a certain area. On the other hand, the very nature of these services demands that when the emergency, alert, or urgent notification arises, all UAs in the area are notified, not just those that have explicitly subscribed.

2. Dormant Mobile Device Presence Determination. With the proliferation of hyper-local location based services that need to be triggered when a user enters or exits a certain geographical area (especially exiting constitutes a challenge), there is a strong need for a lightweight mechanism capable of pinging and waking up a dormant mobile device.

Ideally, such a mechanism would only involve the mobile browser, and thus be usable without requiring the installation of native clients on the device, be air-interface agnostic and capable to operate over technologies used for indoor positioning, and thus be usable to sense presence with small cells, WiFi, Bluetooth, etc., and operate without requiring paging.

Web Push potentially fits all these requirements. However, also in this case, in order for Web Push to serve the purpose, a subscription-less flavor of Web Push would be required, since the goal is to sense presence of all UAs, not just those that have subscribed to a Web Push service.

3. Venues with Relaxed User Privacy Expectations. The use of Web Push would be beneficial in Smart Buildings and other environments where the user has a manifested expectation of tapping into available services in an unsolicited fashion. Examples include enterprise smart applications and venues where a provider clearly "owns" the connectivity (e.g., in a store associated to a captive portal) or there is an established expectation by the user that a

provider is making available certain services to all the UAs in that space.

In such scenarios, the user may have an established tolerance or even an expectation for a controlled volume of information being pushed to the device. Also in this case, the notion of subscription is implicit because it is clearly specific to time and location, since the subscription is associated with the duration of the user presence in a certain venue.

These use cases, which are rapidly growing in popularity are an indication that some notion of "subscription-less Web Push," or more precisely Web Push with some form of relaxed subscription requirements, is actually desirable and useful to increase the applicability of Web Push.

4. Broadcast Local Web Push. In general, for services that need to broadcast information using Web Push to an audience in a location, the requirement for explicit subscription often reduces the effectiveness of the service. A controlled Web Push service that does not rely on explicit subscription would be very useful to enable more effective services of this kind.

3. Subscription Management Across Applications

The second limitation of the current Web Push subscription model is the fact that it does not directly provide a mechanism to control the global volume of traffic destined to a given UA. Instead, the current control relies on judiciously distributing subscriptions to applications and potentially placing a limit on the number of applications that can be allowed to use Web Push.

This approach, in addition to curtailing the scalability of Web Push across different applications, may not provide effective control on the quality of the User Experience when Web Push is used.

The quality of experience is largely determined by the global volume and timing of Web Push traffic that a UA receives, rather than what is specifically generated by an individual application. In addition, limiting the number of applications that can use Web Push is recognized as a rather crude way to control the global traffic, since it is not the behavior of a single application that defines the quality of experience, but rather the combined behavior of all the relevant applications in a certain location.

The metric of interest is not simply the volume of traffic, but the timing of the traffic. Since applications are scarcely aware of one another, it is certainly conceivable or even likely that the

combination of traffic may be perceived by the user as significantly more intrusive than the traffic from each individual applications.

All these reasons point to the need for an entity in charge of managing Web Push across applications.

4. Web Push Subscription Authority

Both the subscription-less Web Push service and the capability of managing Web Push subscriptions globally across applications are achieved by introducing a trusted WPSA in charge of controlling the traffic generated by Web Push services and destined to each UA.

The WPSA imposes policies that control the combined volume and profile of Web Push traffic destined to each UA and provide coordination among traffic generated by different applications.

By its nature, the WPSA is a "policer" of Web Push traffic across applications. As such, it must provide mechanisms to throttle traffic generated by each application, and when necessary, prioritize traffic from one application versus another.

A first challenge in defining the WPSA is the definition of trust underlying this entity. Trust must be established in two ways. In order to enable subscription-less Web Push, the UAs must trust the WPSA to control the nature and volume of allowed subscription-less traffic. In order to enable global Web Push traffic management, the applications must recognize the WPSA with the authority of prioritizing different applications based on globally-defined policies.

A second challenge in defining the WPSA is the distributed nature of the WPSA function. For example, most the use cases advocating subscription-less Web Push are highly-local in nature. Accordingly, the WPSA needs to be capable to apply policies that are global across applications, but may be location specific. Since local instances of the WPSA may be required for scalability of such a model, a simple mechanism to discover them and associate them with each location is also required.

5. Security Considerations

TBD.

6. IANA Considerations

TBD.

7. References

7.1 Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [I-D. [draft-thomson-webpush-protocol](#)] M. Thomson et al., "Generic Event Delivery Using HTTP Push", [draft-thomson-webpush-protocol-00](#) (work in progress), April 2015.

7.2 Informative References

- [W3CAPI] Sullivan, B., Fullea, E., and M. van Ouwerkerk, "Web Push API", ED push-api, February 2015, <<https://w3c.github.io/push-api/>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", [RFC 6973](#), July 2013.

Authors' Addresses

Fabio Chiussi
Seattle, WA 98116
Email: fabiochiussi@gmail.com

