

# Using public websub hubs for PushMDQ

## Introduction

The distribution of metadata is the key link that binds together identity federation both at the national and international (inter-federation) level with eduGAIN. As the scale of federations and inter-federations (eduGAIN membership) increases the dominant mechanism for sharing metadata, based on the exchange of a flat file structure becomes problematic. Firstly from the size of the file itself, and secondly from the associated processing delays and distribution of any updates notifying entities of changes. This can have a potentially serious effect in the case of compromise, where a considerable propagation delay can potentially leave a whole federation vulnerable until it is detected.

These issues can be ameliorated by the adoption of the more recent MDQ protocol which allows for the query of metadata relating to one or more entities in a federation, thus resulting in smaller amount of data transfer and potentially smaller propagation delays. However, this approach suffers from both being a single point of failure (for the querying entity) and from adopting a 'pull-based' model where the query is subject to deterministic scheduling and thus can not discover changes in real-time.

To address these issues a new mechanism -PushMDQ- has been proposed which is based on a publisher - subscriber model known as websub (formerly PubSubHubbub) which was originally intended for data feeds such as RSS or ATOM, but can be applied more generally. Briefly, a publisher of information (in this case metadata) publishes information to a hub to which interested parties can subscribe and receive notifications from when new content is available. Hence a subscriber (in this case an entity within a federation) can receive instant notification when an update to metadata has occurred and this can (potentially) be propagated across the federation (or federations in the case of eduGAIN) with minimal delay.

There are a number of benefits to such a system:

- Changing from deterministic pull to a push mechanism reduces pulls by up to 95%;
- Updates can be propagated to all entities instantly (for practical purposes);
- Use of MDQ within PushMDQ reduces the processing overhead;
- Backwards compatibility with existing flat-file distribution is possible;
- There is no longer a single point of failure;
- Websub hubs do not need to be intelligent and may exist outside the federation.

In particular the ability to 'piggy-back' onto an existing maintained and publically available hub may be advantageous as it removes the necessity to own and operate this part of the

infra-structure. A number of publicly available hubs are considered against the requirements in the context of PushMDQ.

## Websub hub requirements

A websub hub must be capable of establishing a trusted relationship with more than one publishing (generally the aggregating publisher) or subscribing entity (where in general they may be in different federations)

A websub may re-publish metadata where a trust relationship does not exist between it and the root publisher. (i.e. a websub must support publishing and subscription)

A websub hub may point to the self-link of the root publisher where it can trust such a source, for example as a result of a signed certificate from such publisher.

A websub hub must comply with w3C websub specification 23rd January 2018

A websub hub must use https for all requests

A websub hub should accept a subscription request with the optional hub.secret parameter as a means to authenticate and trust the hub.

A websub hub should not need to perform any additional processing beyond (re)publishing and subscription. (i.e. no intelligence required)

A websub hub must provide availability equivalent to that targeted by eduGAIN i.e at least 99%

A websub hub must be consistent with the eduGAIN privacy policy

A websub hub must not require financial consideration for the use of its services

## Public websub hub survey

There are a number of publicly available websub hubs. Some are free to use and others have free and paid options or are associated with various publishing/subscription platforms. Websub is a less popular mechanism for subscriber notification than Webhooks (which has been around for longer) and consequently there are less publicly available and supported services. Some of the principle ones are:

# Google

Google provides two options:

1)

<https://pubsubhubbub.appspot.com/>

This conforms to PubSubHubbub Core 0.4 (2014) not the later W3C spec. There is some degree of backwards compatibility, but also differences.

Use requires acceptance of Google's terms of service and privacy policy. There seem to be no additional terms or privacy policy specific to the service, so only Google's general policies apply.

Typically a licence to allow Google to::

- host, reproduce, distribute, communicate and use your content – for example, to save your content on our systems and make it accessible from anywhere that you go
- publish, publicly perform or publicly display your content, if you've made it visible to others
- modify your content, such as reformatting or translating it
- sublicense these rights to:
  - other users to allow the services to work as designed, such as enabling you to share photos with people that you choose
  - our contractors who've signed agreements with us that are consistent with these terms, only for the limited purposes described in the [Purpose](#) section below

SLA - not available.

2) A Pub/Sub provided as a part of the Google Cloud platform <https://cloud.google.com/pubsub>.. This is a paid service (after the free trial) but does have a defined SLA with 99.95% availability and a good level of service. Obviously, this service requires a sign-up with implications for data gathering and privacy.

# Superfeedr

<http://pubsubhubbub.superfeedr.com/>

This conforms to PubSubHubbub Core 0.4 (2014) not the later W3C spec. There is some degree of backwards compatibility, but also differences.

Recommends the use of the optional hub.secret for authentication and increased security.

A free hosted hub is available, customised version including landing page and subscription control is \$200/month for publishers and from \$1/20 subscriptions/month for subscribers.

Privacy policy - <https://superfeedr.com/privacy>

T&C - <https://superfeedr.com/terms>

Terms of use are wide ranging.

SLA - 99.999% uptime

## Wordpress

<https://wordpress.com/support/pubsubhubbub/>

This is supported through Wordpress plugins.

PuSHPress conforms to the latest W3C websub specification and creates its own.

PubSubHubbub uses as a default the Google and Superfeeder hubs described above.

WP PubSubHubbub is as PubSubHubbub but allows the user to specify one or more of their own hubs.

Requires WordPress account ranging from free to £20/month

## Twitter

Twitter uses the Superfeeder PubSubHubbub implementation located at

<https://twitter.superfeedr.com/>

## Discussion

Whilst there are many implementations of WebSub or the prior PubSubHubbub versions available on github and other repositories there seem to be only a few community accessible websub hubs. The two main services (used by many other organisations) are those from Google and Superfeedr. Both provide free services, but both only conform to the 0.4 version of PubSubHubbub and not the later W3C recommendation.

Superfeedr requires sign-up to get access to the service hub and sign up implies acceptance of the terms of service and privacy policies. There are various terms but a key part is the right for Superfeedr to 'use, modify, reproduce, distribute, prepare derivative works of, display, perform, and otherwise fully exploit the User Submissions' whilst this seems unlikely with the metadata submissions, these rights do exist, as do others modify and reformat. In addition Superfeedr may terminate access without cause and provide the service 'as is' without express warranties. With respect to privacy Superfeedr collects personal information as a part of the registration process and may pass this on to affiliated businesses they do not control and agents. The privacy policy has some protections, but is very outdated (2010) and makes no reference to more recent legal changes with respect to GDPR and the State of California.

Google does not seem to require any form of sign-up to gain access to the free hub, but usage implies acceptance of Google's terms of service and privacy policy. There seem to be no specific terms applicable to this service so Google's general terms apply. Key

requirements as mentioned above the right of Google to use the content in a variety of ways. It is a requirement that an authorised organisational representative agrees to the terms. The warranty for the Google service references 'reasonable skill and care' and thus gives some greater assurance than Superfeedr. Google does not reserve the right to terminate access without cause but only for specific events such as breach, legal necessity, conduct. The privacy policy requires certain information such as unique identifiers, IP address etc. to be collected for services such as this where there is no user login and this could be used for the 'normal' Google purposes. However, since personal information is more limited the impact is reduced. Google meets the legal requirements for GDPR with the necessary protections. The Google Cloud service has a higher SLA, but has fees <https://cloud.google.com/pubsub/pricing> and additional privacy issues.

## Recommendation

There are many websub implementations but few publically accessible services. The principal candidates for community hubs are those from Google and Superfeedr. Of the two the Google free service is to be preferred in terms of its more open terms of use and more limited collection of information. The express warranty for the service is also superior to Superfeedr's. That said Google's implementation does not support the latest W3C implementation (if that turns out to be significant) and it does not publish any availability or SLA for the service that I could find. This is not the case for the Google Cloud version, but this is paid for and requires registration.

There are a number of considerations relevant to the requirements. Firstly, the privacy policies and terms of use, which although in the case of Google's free service may be acceptable, do not accord with the respective terms from eduGAIN - which the hub will be a part of.

Secondly, the requirement that there is no financial consideration for the use of the hub would preclude all but the free services. It might be that the costs could be funded by the project, but the sustainability, post project, would need to be clear before going down this route.

Next the availability and SLA seems to be rather limited from Google unless a paid service is taken, although Superfeedr do give some availability guarantee. It is not clear how robust the hubs are and whether there is provision for fail-over in the event of hub failure. This also raises issues of how back-off would be handled in these services and the effect this might have.

Another issue is compatibility of the websub versions. Both Google and Superfeedr use an early version of websub - PubSubHubbub 0.4 and this could give compatibility issues between different deployments of publishers, subscribers and hubs depending on which implementation and version was used and which feature of the specs. are required.

Finally, there is the issue of 'processing' as it applies to establishing a trusted relationship between the hub and its root publishers. Although re-publishing is possible where not trust exists in general some form of authentication of publishers is required to prevent

subscriptions to non-approved content. This could imply a degree of intelligent processing in the hub which might be difficult for a community hub to support.

Taking account of these issues, unknowns and limitations I would recommend we host our own implementation rather than use a community hub. This would give us more control and allow us to have deterministic outcomes to these issues and meet the requirements (hopefully!). This would of course require consideration of how this could be managed and sustained, but this seems an easier problem to solve than the others above