



XMPP

XEP-0277: Microblogging over XMPP

Peter Saint-Andre
<mailto:xsf@stpeter.im>
<xmpp:peter@jabber.org>
<http://stpeter.im/>

Joe Hildebrand
<mailto:jhildebr@cisco.com>
<xmpp:hildjj@jabber.org>

Sergey Dobrov
<mailto:binary@jrudevels.org>
<xmpp:binary@jrudevels.org>
<http://jrudevels.org/>

Valerian Saliou
<mailto:valerian@valeriansaliou.name>
<xmpp:valerian@jappix.com>
<https://valeriansaliou.name/>

2017-11-28
Version 0.6.2

Status	Type	Short Name
Deferred	Standards Track	microblog

This specification defines a method for microblogging over XMPP.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2017 by the [XMPP Standards Foundation](#) (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy>) or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1	Introduction	1
2	Protocol	1
2.1	Location	1
2.2	Subscribing to a Microblog	2
2.3	Publishing a Post	2
2.3.1	Publishing a Post with Rich Content	3
2.4	Receiving a Post	4
2.5	Replying to a Post	5
2.6	Repeating a Post	6
2.7	Post Categories	7
3	Comments	8
3.1	Post Comments	8
3.2	Adding a Comment	9
4	Pubsub Node Configuration	9
4.1	Microblog node configuration	9
4.2	Comments node configuration	10
5	Microblog Metadata	10
6	Geotagging	11
7	Aggregators	12
7.1	Pubsub Item ID vs. Atom Entry id	13
7.2	Aggregator Usecases	13
7.2.1	Representing Posts in the Web	13
8	Message Body	13
9	Security Considerations	14
9.1	Comment Author	14
10	IANA Considerations	14
11	XMPP Registrar Considerations	14
12	XML Schema	14
13	Acknowledgements	14

1 Introduction

Microblogging is an increasingly popular technology for lightweight interaction over the Internet. It differs from traditional blogging in that:

- Posts are short (typically less than 140 characters, which is the limit in SMS).
- Posts are in plain text.
- People can reply to your posts, but not directly comment on them.
- People learn about your posts only if they have permission to view them.
- Your microblogging feed is discovered based on your identity at a domain or with a service.

These characteristics map well to instant messaging systems such as those built using Jabber/XMPP technologies (e.g., permissions can be based on existing presence subscriptions as reflected in the XMPP roster or "buddy list"). Furthermore, the push nature of XMPP (especially as formalized in the [Personal Eventing Protocol \(XEP-0163\)](https://xmpp.org/extensions/xep-0163.html)¹ profile of [Publish-Subscribe \(XEP-0060\)](https://xmpp.org/extensions/xep-0060.html)²) overcomes the problems of polling for updates via HTTP, which has caused scaling issues in existing microblogging services. Therefore this specification defines a method for microblogging over XMPP, building on the existing method for transporting Atom syndication data [RFC 4287](http://tools.ietf.org/html/rfc4287)³ over XMPP as described in [AtomSub](https://tools.ietf.org/html/draft-saintandre-atompub-notify-07)⁴. These XMPP-based methods are complementary to HTTP-based methods, and can provide an XMPP interface to existing microblogging services (which may also be accessible via HTTP, Short Message Service (SMS), and other messaging transports).

2 Protocol

2.1 Location

A person's microblog SHOULD be located at a personal eventing (PEP) node whose name is "urn:xmpp:microblog:0" but MAY be located at a generic publish-subscribe node that is not attached to a user's IM account. For instance, if the Shakespearean character Romeo has a JabberID of <romeo@montague.lit> then his microblog would be located at that JID with a node of "urn:xmpp:microblog:0". Outside of native XMPP systems, this node can be referred to as the following XMPP URI (see XEP-0060 § 12.21).

¹XEP-0163: Personal Eventing Protocol <<https://xmpp.org/extensions/xep-0163.html>>.

²XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

³RFC 4287: The Atom Syndication Format <<http://tools.ietf.org/html/rfc4287>>.

⁴Atomsub: Transporting Atom Notifications over the Publish-Subscribe Extension to the Extensible Messaging and Presence Protocol (XMPP) <<https://tools.ietf.org/html/draft-saintandre-atompub-notify-07>>. Work in progress.

```
xmpp:romeo@montague.lit?;node=urn%3Axmpp%3Amicroblog%3A0
```

Note that the ":" character from the namespace URN is percent-encoded in the query component (see [RFC 5122](#)⁵ and [RFC 3986](#)⁶).

Naturally, this node can be discovered by contacting romeo@montague.lit directly using [Service Discovery \(XEP-0030\)](#)⁷.

2.2 Subscribing to a Microblog

Let us imagine that Juliet wishes to receive the posts that Romeo publishes to his microblog. She has two options:

1. Implicitly subscribe by advertising support for "urn:xmpp:microblog:0+notify" in her [Entity Capabilities \(XEP-0115\)](#)⁸ data. Romeo's PEP service then automatically sends posts to her when it receives presence from her, until and unless she sends presence of type unavailable or stops advertising an interest in microblog updates.
2. Explicitly subscribe by sending a formal subscription request to the "urn:xmpp:microblog:0" node at Romeo's JabberID. Romeo's PEP service may send her all posts even if she is offline at the time (depending on service policies regarding presence integration).

2.3 Publishing a Post

Romeo can publish a post via any interface provided by his service, such as a website, the Atom Publishing Protocol (see [RFC 5023](#)⁹), SMS, an IM bot, or XMPP pubsub. Here we assume that the post is provided via XMPP pubsub.

The post content itself can be either text (content element without "type" attribute or with "type" attribute with "text" value) or XHTML ("content" element "type" attribute with "xhtml" value). If Romeo publishes XHTML content, his client MUST publish two "content" elements: a text one, and a XHTML one. For XHTML publishing, see [Publish-Subscribe \(XEP-0060\)](#)¹⁰.

Note: Publishing via HTTP, AtomPub, SMS, or IM bot is simpler for the client (e.g., because the client does not need to generate an Item ID).

⁵RFC 5122: Internationalized Resource Identifiers (IRIs) and Uniform Resource Identifiers (URIs) for the Extensible Messaging and Presence Protocol (XMPP) <<http://tools.ietf.org/html/rfc5122>>.

⁶RFC 3986: Uniform Resource Identifiers (URI): Generic Syntax <<http://tools.ietf.org/html/rfc3986>>.

⁷XEP-0030: Service Discovery <<https://xmpp.org/extensions/xep-0030.html>>.

⁸XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

⁹RFC 5023: The Atom Publishing Protocol <<http://tools.ietf.org/html/rfc5023>>.

¹⁰XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

Listing 1: Publishing a post

```

<iq from='romeo@montague.lit/pda'
  id='pub1'
  to='romeo@montague.lit'
  type='set'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:microblog:0'>
      <item id='1cb57d9c-1c46-11dd-838c-001143d5d5db'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title type='text'>hanging out at the Caf&#233;
            Napolitano</title>
          <id>tag:montague.lit,2008-05-08:posts-1cb57d9c-1c46-11dd-838
            c-001143d5d5db</id>
          <published>2008-05-08T18:30:02Z</published>
          <updated>2008-05-08T18:30:02Z</updated>
        </entry>
      </item>
    </publish>
  </pubsub>
</iq>

```

Note: The "title" element is required to be included in an "atom:entry" element according to RFC 4287¹¹. An implementation MAY provide also "atom:summary" and/or "atom:content" elements too if it needs.

2.3.1 Publishing a Post with Rich Content

It's possible to insert some rich content in the post or comment. It can be some text formatting, images, etc. Only "xhtml" content type is supported for the moment by this document but possibly it will be extended later. Also, it is RECOMMENDED for the client to restrict XHTML content to the XHTML-IM subset (XHTML-IM (XEP-0071)¹²).

Listing 2: Publishing a post with rich content

```

<iq from='romeo@montague.lit/pda'
  id='pub1'
  to='romeo@montague.lit'
  type='set'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:microblog:0'>
      <item id='1cb57d9c-1c46-11dd-838c-001143d5d5db'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title type='xhtml'>
            <div xmlns="http://www.w3.org/1999/xhtml">

```

¹¹RFC 4287: The Atom Syndication Format <<http://tools.ietf.org/html/rfc4287>>.

¹²XEP-0071: XHTML-IM <<https://xmpp.org/extensions/xep-0071.html>>.

```

        <p>hanging out at the <strong>Caf&#233; Napolitano</
          strong></p>
      </div>
    </title>
    <id>tag:montague.lit,2008-05-08:posts-1cb57d9c-1c46-11dd-838
      c-001143d5d5db</id>
    <published>2008-05-08T18:30:02Z</published>
    <updated>2008-05-08T18:30:02Z</updated>
  </entry>
</item>
</publish>
</pubsub>
</iq>

```

2.4 Receiving a Post

Because Juliet has sent presence to Romeo including Entity Capabilities data that encapsulates the "urn:xmpp:microblog:0+notify" feature, Romeo's XMPP server will send a PEP notification to Juliet. The notification can include an XMPP message body for backwards-compatibility with Jabber clients that are not pubsub-capable (see [Message Body](#)).

Listing 3: Receiving a post

```

<message from='romeo@montague.lit'
  to='juliet@capulet.lit'
  type='headline'>
  <body>hanging out at the Caf&#233; Napolitano</body>
  <event xmlns='http://jabber.org/protocol/pubsub#event'>
    <items node='urn:xmpp:microblog:0'>
      <item id='1cb57d9c-1c46-11dd-838c-001143d5d5db' publisher='
        romeo@montague.lit'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title type='text'>hanging out at the Caf&#233;
            Napolitano</title>
          <link rel='alternate'
            type='text/html'
            href='http://montague.lit/romeo/posts/1cb57d9c-1c46-11
              dd-838c-001143d5d5db' />
          <link rel='alternate'
            href='xmpp:romeo@montague.lit?;node=urn%3Axmpp%3
              Amicroblog%3A0;item=1cb57d9c-1c46-11dd-838c-001143
              d5d5db' />
          <id>tag:montague.lit,2008-05-08:posts-1cb57d9c-1c46-11dd-838
            c-001143d5d5db</id>
          <published>2008-05-08T18:30:02Z</published>
          <updated>2008-05-08T18:30:02Z</updated>
        </entry>
      </item>
    </items>
  </event>
</message>

```

```

    </items>
  </event>
</message>

```

Note: these alternate links were not posted by client because client can't compute them itself. These things SHOULD be inserted at server side though.

2.5 Replying to a Post

Anyone can publish a post in reply to Romeo's post. Here we assume that a reply comes from Benvolio.

Note: Inclusion of the "thr:in-reply-to" element defined in RFC 4685¹³ indicates the post to which the user is replying. This reply includes two such elements (one pointing to the HTTP URL for the post and the other pointing to the XMPP URI for the post).

Note: The post can be a reply to more than the only one another.

Listing 4: Publishing a reply

```

<iq from='benvolio@montague.lit/mobile'
  id='uv2x37s5'
  to='benvolio@montague.lit'
  type='set'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:microblog:0'>
      <item id='c4145006-1c53-11dd-b2d5-000bcd82471e'>
        <entry xmlns='http://www.w3.org/2005/Atom'
          xmlns:thr='http://purl.org/syndication/thread/1.0'>
          <author>
            <name>Benvolio Montague</name>
            <uri>xmpp:romeo@montague.lit</uri>
          </author>
          <title type='text'>@romeo cappuccino this late in the day</title>
          <link rel='alternate'
            type='text/html'
            href='http://montague.lit/benvolio/posts/c4145006-1c53-11dd-b2d5-000bcd82471e' />
          <link rel='alternate'
            href='xmpp:benvolio@montague.lit?;
            .....node=urn%3Axmpp%3Amicroblog%3A0;
            .....item=c4145006-1c53-11dd-b2d5-000bcd82471e' />
          <id>tag:montague.lit,2008-05-08:posts-c4145006-1c53-11dd-
            b2d5-000bcd82471e</id>
          <published>2008-05-08T18:31:21Z</published>
          <updated>2008-05-08T18:31:21Z</updated>

```

¹³RFC 4685: Atom Threading Extensions <<http://tools.ietf.org/html/rfc4685>>.


```

    <thr:in-reply-to
      ref='tag:montague.lit,2008-05-08:posts-1cb57d9c-1c46-11
        dd-838c-001143d5d5db'
      type='application/xhtml+xml'
      href='http://montague.lit/romeo/posts/1cb57d9c-1c46-11
        dd-838c-001143d5d5db' />
    <thr:in-reply-to
      ref='tag:montague.lit,2008-05-08:posts-1cb57d9c-1c46-11
        dd-838c-001143d5d5db'
      href='xmpp:romeo@montague.lit?;node=urn%3Axmpp%3
        Amicroblog%3A0;item=1cb57d9c-1c46-11dd-838c-001143
        d5d5db' />
  </entry>
</item>
</publish>
</pubsub>
</iq>

```

Assuming that Romeo has also shared presence with Benvolio and has advertised support for "urn:xmpp:microblog:0+notify", he will receive the reply that Benvolio sent.

2.6 Repeating a Post

When Benvolio wants to repeat a Romeo's post, his client publishes the same post under a different name. But to be able to track the repeated post original author, Benvolio's client MAY use specific "atom:author" child node, "atom:name" and "atom:uri", containing, respectively, the name of the original post author, and his XMPP URI (JID).

If a comments link is present (see the [Post Comments](#) section of this document), the client SHOULD repeat it to keep the same discussion about the post. The client also MAY create a separate node to discuss and specify it or specify both.

The client SHOULD also put an "atom:link" element with "rel" attribute set to "via" and point it to the original post.

Listing 5: Repeating a Post

```

<iq from='benvolio@montague.lit/mobile'
  id='pub2'
  to='benvolio@montague.lit'
  type='set'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:microblog:0'>
      <item id='1re57d3c-1q46-11dd-748r-024943d2d5rt'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <author>
            <name>Romeo Montague</name>
            <uri>xmpp:romeo@montague.lit</uri>
          </author>

```

```

<title type='text'>hanging out at the Caf&#233;
  Napolitano</title>
<link rel='alternate'
  type='text/html'
  href='http://montague.lit/benvolio/posts/1re57d3c-1q46
    -11dd-748r-024943d2d5rt' />
<link rel='alternate'
  href='xmpp:benvolio@montague.lit?;node=urn%3Axmpp%3
    Amicroblog%3A0;item=1re57d3c-1q46-11dd-748r-024943
    d2d5rt' />
<link rel='via'
  href='xmpp:romeo@montague.lit?;node=urn%3Axmpp%3
    Amicroblog%3A0;item=1cb57d9c-1c46-11dd-838c-001143
    d5d5db'
  ref='tag:montague.lit,2008-05-08:posts-1cb57d9c-1c46
    -11dd-838c-001143d5d5db' />
<id>tag:montague.lit,2008-05-08:posts-1re57d3c-1q46-11dd-748
  r-024943d2d5rt</id>
<published>2008-05-08T18:30:02Z</published>
<updated>2008-05-08T18:32:02Z</updated>
</entry>
</item>
</publish>
</pubsub>
</iq>

```

Thus, a different author JID value lets the client know the microblog item has been repeated from another one.

It's also possible for Benvolio to add his own thoughts to the repost. To do this he SHOULD wrap the original content in the "xhtml:blockquote" element and after it add his own content. Also, the client MAY post reply without quotation to the original thread to inform users about the repost.

2.7 Post Categories

It's often handy to specify categories (or tags) to a post to make it easier to find it or to structure a blog. It's possible by adding "atom:category" element to the entry (it can be blog or replies entry).

Listing 6: Specify post's categories

```

<entry xmlns='http://www.w3.org/2005/Atom'>
  ...
  <category term='humour' />
  <category term='xmpp' />
  ...
</entry>

```

3 Comments

Juliet and Benvolio may want to discuss about latest Romeo's post. To enable this, Romeo's client MAY add a "atom:link" element to the PubSub item. The element MUST have "rel", "title" and "href" attributes, where "rel" MUST have the "replies" value; "title" MUST have the "comments" value; "href" MUST be an XMPP URI (see [RFC 5122](#)¹⁴ and [RFC 3986](#)¹⁵).

3.1 Post Comments

We assume Romeo's client first created a comments node (named "urn:xmpp:microblog:0:comments/ID", where "ID" is the microblog item ID, or the SHA-1 encoded attachment URI, as defined in [RFC 3174](#)¹⁶).

Listing 7: Adding a comments link to a Post

```
<iq from='romeo@montague.lit/pda'
  id='pub4'
  to='romeo@montague.lit'
  type='set'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:microblog:0'>
      <item id='2ze57d9c-1c46-21df-830c-002143d3d2qgf'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title type='text'>hanging out at the Caf&#233;
            Napolitano</title>
          <link rel='replies'
            title='comments'
            href='xmpp:pubsub.montague.lit?;node=urn%3Axmpp%3
              Amicroblog%3A0%3Acomments%2
              Fdd88c9bc58886fce0049ed050df0c5f2' />
          <id>tag:montague.lit,2008-05-08:posts-2ze57d9c-1c46-21df-830
            c-002143d3d2qgf</id>
          <published>2008-05-08T18:38:02Z</published>
          <updated>2008-05-08T18:38:02Z</updated>
        </entry>
      </item>
    </publish>
  </pubsub>
</iq>
```

¹⁴RFC 5122: Internationalized Resource Identifiers (IRIs) and Uniform Resource Identifiers (URIs) for the Extensible Messaging and Presence Protocol (XMPP) <<http://tools.ietf.org/html/rfc5122>>.

¹⁵RFC 3986: Uniform Resource Identifiers (URI): Generic Syntax <<http://tools.ietf.org/html/rfc3986>>.

¹⁶RFC 3174: US Secure Hash Algorithm 1 (SHA1) <<http://tools.ietf.org/html/rfc3174>>.

3.2 Adding a Comment

If Juliet wants to comment Romeo's latest post, her client sends a new Atom entry to the defined PubSub node.

Note: A comments node SHOULD be located at a generic publish-subscribe node that is not attached to a user's IM account, but MAY be located at a personal eventing (PEP) node.

Listing 8: Adding a comment to a comments node

```
<iq from='juliet@capulet.lit/pc'
  id='comment1'
  to='pubsub.capulet.lit'
  type='set'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:microblog:0:comments/
      dd88c9bc58886fce0049ed050df0c5f2'>
      <item id='b2106a80de39ef5ec6b8f7b69cb610c2'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <author>
            <name>Juliet Capulet</name>
            <uri>xmpp:juliet@capulet.lit</uri>
          </author>
          <title type='text'>She is so pretty!</title>
          <published>2008-05-08T18:39:02Z</published>
        </entry>
      </item>
    </publish>
  </pubsub>
</iq>
```

If Benvolio wants to retrieve the comments node, his client will send a standard PubSub stanza to request all items (see [Publish-Subscribe \(XEP-0060\)](#)¹⁷ for all items retrieving).

4 Pubsub Node Configuration

We have described two pubsub nodes types here: the one is for the microblog itself and the other one is for comments to posts.

Usage specific requires the special parameters to these nodes to be specified. This section describes recommendation for them.

4.1 Microblog node configuration

Here are recommendations for a microblogging node (usually located at PEP and named by the "urn:xmpp:microblog:0" namespace) configuration:

¹⁷XEP-0060: Publish-Subscribe <<https://xmpp.org/extensions/xep-0060.html>>.

1. The "pubsub#notify_retract" MUST be set to true to provide clients the ability to track if some items were retracted and reflect such changes in the UI correctly.
2. The "pubsub#max_items" SHOULD be increased from the default value to some reasonable value.
3. The "pubsub#send_last_published_item" SHOULD be changed to "never".

4.2 Comments node configuration

Here are recommendations for a comments node configuration:

1. The "pubsub#notify_retract" MUST be set to true to provide clients the ability to track if some items were retracted and reflect such changes in the UI correctly.
2. The "pubsub#max_items" SHOULD be increased from the default value to some reasonable value.
3. The "pubsub#access_model" SHOULD be set to "open" to allow any user to comment to a post. Other values are suitable too according to the user's settings.

5 Microblog Metadata

In order to provide users with some metadata (i.e. blog title or author information) about the microblog, the client MUST to add an item with such information. The client MUST set the ID of the item to "0".

Listing 9: Publishing microblog metadata

```
<iq from='romeo@montague.lit/pc'
  id='pub8'
  to='romeo@montague.lit'
  type='set'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:microblog:0'>
      <item id='0'>
        <feed xmlns='http://www.w3.org/2005/Atom'>
          <title>Romeo&apos;s Microblog</title>
          <id>tag:montague.lit,2008:home</id>
          <updated>2008-05-08T18:30:02Z</updated>
          <author>
            <name>Romeo Montague</name>
            <uri>xmpp:romeo@montague.lit</uri>
          </author>
        </feed>
      </item>
    </publish>
  </pubsub>
</iq>
```

```

    </item>
  </publish>
</pubsub>
</iq>

```

It also necessary to link a comments node to the post which discussed in the node. We will do it by adding the "atom:link" element with "rel=start" attribute:

Listing 10: Publishing comments node metadata

```

<iq from='romeo@montague.lit/pc'
  id='pub8'
  to='pubsub.montague.lit'
  type='set'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:microblog:0'>
      <item id='0'>
        <feed xmlns='http://www.w3.org/2005/Atom'>
          <title>Comments to a post</title>
          <id>tag:pubsub.montague.lit,2008:comments-2ze57d9c-1c46-21df-830c-002143d3d2qgf</id>
          <updated>2008-05-08T18:30:02Z</updated>
          <link rel='start'
            href='xmpp:romeo@montague.lit?;node=urn%3Axmpp%3Amicroblog%3A0;item=1cb57d9c-1c46-11dd-838c-001143d5d5db'
            ref='tag:montague.lit,2008-05-08:posts-1cb57d9c-1c46-11dd-838c-001143d5d5db' />
        </feed>
      </item>
    </publish>
  </pubsub>
</iq>

```

6 Geotagging

Juliet may want to know which places are Romeo's notices related to. That's why Romeo's client MAY geotag microblog entries, using the [User Geolocation \(XEP-0080\)](https://xmpp.org/extensions/xep-0080.html)¹⁸ protocol for storing geolocation information.

Romeo's client MUST create a "geoloc" element, with the [User Geolocation \(XEP-0080\)](https://xmpp.org/extensions/xep-0080.html)¹⁹ reference namespace: "http://jabber.org/protocol/geoloc".

Listing 11: Geotagging a Post

¹⁸XEP-0080: User Geolocation <<https://xmpp.org/extensions/xep-0080.html>>.

¹⁹XEP-0080: User Geolocation <<https://xmpp.org/extensions/xep-0080.html>>.

```

<iq from='romeo@montague.lit/mobile'
  id='pub7'
  to='romeo@montague.lit'
  type='set'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='urn:xmpp:microblog:0'>
      <item id='1zr23z8a-3g12-34fh-750b-120867gjc1sqh'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <author>
            <name>Romeo Montague</name>
            <uri>xmpp:romeo@montague.lit</uri>
          </author>
          <title type='text'>Is lost in the forest. Need help!</title>
          <link rel='replies'
            title='comments'
            href='xmpp:pubsub.montague.lit?;node=urn%3Axmpp%3A
              Amicroblog%3A0%3Acomments%2
              F36ec6dfe61e52b1e2cdb248823520233' />
          <id>tag:montague.lit,2008-05-08:posts-1zr23z8a-3g12-34fh-750
            b-120867gjc1sqh</id>
          <published>2008-05-08T18:43:01Z</published>
          <updated>2008-05-08T18:43:01Z</updated>
          <geoloc xmlns="http://jabber.org/protocol/geoloc">
            <lat>48.171761</lat>
            <lon>-3.3667986</lon>
            <country>France</country>
            <countrycode>FR</countrycode>
            <region>Brittany</region>
          </geoloc>
        </entry>
      </item>
    </publish>
  </pubsub>
</iq>

```

7 Aggregators

In order to provide some statistical information, to represent the blogs other ways than XMPP (i.e. in web or NNTP), we will need other entities called "Aggregators".

You can think of aggregators like about search engines in the web: they gather information from the whole web and then represent it suitable ways. The same is true here: Aggregators just subscribe to many entities and then they can build a database to make queries and show appropriate information according to these queries. Also they can be used to represent information in web or other networks.

Unlike web search engines, an XMPP aggregator does not need to gather information by downloading it frequently to check if something was changed. Instead, it can listen to pubsub

events and make its database up-to-date based on this information.

Aggregators can be started by different people with different aims. It can be aggregator which devoted to the concrete blog service provider which will be subscribed only to its own users, or it can be a global search engine which tries to subscribe to most of users or to aggregate the feeds of other aggregators.

This section will describe most used aggregator usecases but the list is not exhaustive.

7.1 Pubsub Item ID vs. Atom Entry id

There are two different things that carry a similar sense: the XMPP pubsub Item ID and the "atom:id" element. This section is devoted to make a separation between them.

The pubsub Item ID MUST be used when linking to an entry with an XMPP channel (i.e. by including it in the URI with the "xmpp" schema). the Atom entry ID MUST be built according to [RFC 4287](#)²⁰ and used in aggregators with the aim of revealing of post duplicates, reposts, mentions, syndications, etc.

Note that the rules of comparing, building and security notes for "atom:id" are listed in the [RFC 4287](#)²¹.

7.2 Aggregator Usecases

7.2.1 Representing Posts in the Web

One of the possible aims of aggregator services is to provide web representation of blogs.

TBD: insert alternate link to the post with the http address of the post.

8 Message Body

Depending on service policies and the value of the "pubsub#include_body" node configuration option, microblogging notifications SHOULD include a message "body" element for backwards-compatibility with Jabber clients that are not pubsub-capable. It is RECOMMENDED for the XML character value of the "body" element to be the same as that of the "atom:title" child of the "atom:entry".

²⁰RFC 4287: The Atom Syndication Format <<http://tools.ietf.org/html/rfc4287>>.

²¹RFC 4287: The Atom Syndication Format <<http://tools.ietf.org/html/rfc4287>>.

9 Security Considerations

9.1 Comment Author

The client SHOULD check that the comment author information (provided in the "author" element) is valid, by checking that the "publisher" item attribute value matches the "uri" element value. If there is a difference or the check can not be performed because there was not a "publisher" attribute included, the comment can be displayed, but it is RECOMMENDED to specify there is a security problem.

10 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)²².

11 XMPP Registrar Considerations

The [XMPP Registrar](#)²³ is requested to issue an initial namespace of "urn:xmpp:microblog:0".

12 XML Schema

This specification re-uses the schema for the Atom content format, i.e., the 'http://www.w3.org/2005/Atom' namespace (see RFC 4287).

13 Acknowledgements

Thanks to Ralph Meijer and Paul Scott for their suggestions.

²²The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

²³The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.