XEP-0395: Atomically Compare-And-Publish PubSub Items

Florian Schmaus
mailto:flo@geekplace.eu
xmpp:flo@geekplace.eu

2018-12-06
Version 0.2.0

Status	Type	Short Name
Deferred	Standards Track	cap

This specification provides a mechanism to atomically Compare-And-Publish items to a PubSub node.

Legal

Copyright

Permissions

Warranty

Liability

Conformance

Contents

1 Introduction

This specification provides a mechanism to atomically publish items to a PubSub node depending on the item ID of the node's latest item. This allows to prevent race conditions and avoids data loss in certain situations.

2 Discoverying Support

If an entity supports the Compared-And-Publish feature it MUST advertise the fact by returning a <feature/> with the 'var' attribute set to 'urn:xmpp:pubsub:cap:0' in response to a Service Discovery (XEP-0030) [1] query for information.

3 Compare-And-Publish PubSub Items

3.1 PubSub Item Compare-And-Publish Value (CAP-V)

PubSub services supporting the Compare-And-Publish PubSub extension MUST include a Comapre-and-Publish value (CAP-V) for every item in every response. The CAP-V value MUST change if the content of the item changed and different item content under the same node MUST NOT yield the same CAP-V. A simple computation of the CAP-ID would be to hash the String representation of the item's content.
CAP-Vs are assoicated with PubSub node's items via the item ID. The maping information is placed by the PubSub service in a <cap-v-map/> extension element, qualified by the 'urn:xmpp:pubsub:cap:0' namespace, as child element of the <items/> element. The <cap-v-map/> element contains one or more <cap-v-map-entry/> elements, of which each MUST have a 'item-id' and a 'cap-value' attribute. The former contains the PubSub item ID value and the later contains the according CAP-V of the item.

Listing 1: Service returns some items and their according CAP-Vs

```
<iq type='result'
    from='pubsub.shakespeare.lit'
    to='francisco@denmark.lit/barracks'
    id='items1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <items node='princely_musings'>
      <item id='368866411b877c30064a5f62b917cffe'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title>The Uses of This World</title>
          <summary>
O, that this too too solid flesh would melt
```

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

```
Thaw and resolve itself into a dew!
          </summary>
        </entry>
      </item>
      <item id='3300659945416e274474e469a1f0154c'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title>Ghostly Encounters</title>
          <summary>
O all you host of heaven! O earth! what else?
And shall I couple hell? O, fie! Hold, hold, my heart;
And you, my sinews, grow not instant old,
But bear me stiffly up. Remember thee!
          </summary>
        </entry>
      </item>
      <item id='4e30f35051b7b8b42abe083742187228'>
        <entry xmlns='http://www.w3.org/2005/Atom'>
          <title>Alone</title>
          <summary>
Now I am alone.
O, what a rogue and peasant slave am I!
          </summary>
        </entry>
      </item>
      <cap-v-map xmlns='urn:xmpp:pubsub:cap:0'>
        <cap-value-map-entry
          item-id='368866411b877c30064a5f62b917cffe'
          cap-value='35a204c2-5d6c-43a2-8e0d-a235a627b04a'/>
        <cap-value-map-entry
          item-id='3300659945416e274474e469a1f0154c'
          cap-value='166b7c04-ed4d-4872-aa56-a58268da84e2'/>
        <cap-value-map-entry
          item-id='4e30f35051b7b8b42abe083742187228'
          cap-value='67f7f792-f2ee-4918-8894-36a3c4a6dd5f'/>
      </cap-v-map>
    </items>
  </pubsub>
</iq>
```

3.2 PubSub publishing using Compare-And-Publish

In order to atomically compare-and-publish an item, a client sends a Publish-Subscribe (XEP-0060) [2] <publish/> IQ with a 'pubsub#prev_item_cap_value' precondition publishing option, set to the value of the currently assumed CAP-V of the latest item of the node.
The PubSub service MUST only publish the item if the node's latest item CAP-V is equal to the CAP-V found in the 'pubsub#prev_item_cap_value' field.

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

Listing 2: Atomically publishing with Compare-And-Publish

```
<iq type='set'
    from='hamlet@denmark.lit/blogbot'
    to='pubsub.shakespeare.lit'
    id='pub1'>
  <pubsub xmlns='http://jabber.org/protocol/pubsub'>
    <publish node='princely_musings'>
      <item id='2'>
        <entry xmlns='https://example.org'>
          <title>Soliloquy</title>
          <summary>
To be, or not to be: that is the question:
Whether 'tis␣nobler␣in␣the␣mind␣to␣suffer
The␣slings␣and␣arrows␣of␣outrageous␣fortune,
Or␣to␣take␣arms␣against␣a␣sea␣of␣troubles,
And␣by␣opposing␣end␣them?
␣␣␣␣␣␣␣␣␣␣</summary>
␣␣␣␣␣␣␣␣</entry>
␣␣␣␣␣␣</item>
␣␣␣␣</publish>
␣␣␣␣<publish-options>
␣␣␣␣␣␣<x␣xmlns='jabber:x:data'␣type='submit'>
␣␣␣␣␣␣␣␣<field␣var='FORM_TYPE'␣type='hidden'>
␣␣␣␣␣␣␣␣␣␣<value>http://jabber.org/protocol/pubsub#publish-options</
    value>
␣␣␣␣␣␣␣␣</field>
␣␣␣␣␣␣␣␣<field␣var='pubsub#prev_item_cap_value'>
␣␣␣␣␣␣␣␣␣␣<value>1</value>
␣␣␣␣␣␣␣␣</field>
␣␣␣␣␣␣</x>
␣␣␣␣</publish-options>
␣␣</pubsub>
</iq>
```

3.3 Could not publish because newest item ID did not match

In case the Compare-And-Publish operation failed because the latest node id is not the same as given in the 'previd' attribute in the request, the server returns an <conflict/> error of type 'modify' which a pubsub-specific condition of <precondition-not-met/> and a <compare-and-publish-failed/> element qualifed by the 'urn:xmpp:pubsub:cap:0' namespace. The element MUST have a 'cap-id' attribute with the CAP-V of the lastest item.

Listing 3: Service returns IQ response notifying of failed Compare-And-Publish operation

```
<iq type='error'
```

```
    from='pubsub.shakespeare.lit'
    to='hamlet@denmark.lit/elsinore'
    id='retract1'>
  <error type='modify'>
    <conflict xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
    <precondition-not-met xmlns='http://jabber.org/protocol/pubsub#
        errors'/>
    <compare-and-publish-failed xmlns='urn:xmpp:pubsub:cap:0' cap-id='
        2'/>
  </error>
</iq>
```

4 Rationale

Unfortunately it was not possible to re-use the PubSub item ID for the ”Atomically Compare-And-Publish” purpose. This is mostly due XEP-0060 § 12.8 stating that:
”If a publisher publishes an item and the ItemID matches that of an existing item, the pubsub service MUST overwrite the existing item and generate a new event notification.”
Which means that the content of an item could change without its ID, rendering the item ID unusable for CAP.
Injecting a ”cap”-namespaced attribute carrying the item’s CAP-V into PubSub’s <item/> would be a very elegant approach to assign CAP-Vs to PubSub items (and the favored one of the XEP’s author). But the usage of namespaces attributes within XMPP is controversial. Therefore this XEP resorts to using the <cap-v-map/> approach for now.

5 Security Considerations

This extension protocol does not add any further security considerations to the ones mentioned in XEP-0060 § 14..

6 IANA Considerations

This document requires no interaction with the Internet Assigned Numbers Authority (IANA).

7 XMPP Registrar Considerations

This specification defines the following XML namespaces:

- urn:xmpp:pubsub:cap:0

```
<var>
    <name>urn:xmpp:pubsub:cap:0</name>
    <desc>Indicates support for Compare-And-Publish</desc>
    <doc>XEP-XXXX</doc>
</var>
```

This specification defines the following <publish-options/> fields:

- pubsub#prev_item_cap_value

```
<field var='pubsub#prev_item_cap_value'
         type='text-single'
         label='Precondition: The assumed value of the latest item&
            apos; CAP-V of the node'/>
```

8 XML Schema

TODO: Add after the XEP leaves the 'experimental' state.

9 Acknowledgements

Thanks to Kim Alvefur and Dave Cridland for their feedback.