XEP-0321: Remote Roster Management

Sergey Dobrov
mailto:binary@jrudevels.org
xmpp:binary@jrudevels.org

Jan Kaluža
mailto:hanzz.k@gmail.com
xmpp:hanzz@njs.netlab.cz

2018-11-03
Version 0.1.1

Status	Type	Short Name
Deferred	Standards Track	NOT_YET_ASSIGNED

This document defines a way remote entities may manage user's roster to provide a simple way to keep rosters in sync.

Legal

Copyright

Contents

1 Introduction

It's often desirable for some XMPP based services to make it possible to manage user's roster to provide some contacts there. The most obvious example of such kind of service is a gateway to some legacy IM system (see Gateway Interaction (XEP-0100) [1]). The current way that's recommended by the Gateway Interaction (XEP-0100) [2] is to use the Roster Item Exchange (XEP-0144) [3] to synchronize items that's not sutiable in certain situations.

2 Requirements

This document addresses the following requirements:

1. Make it possible for remote services or entities to manage user's roster by the same mechanisms that descibed in the RFC 6121 [4].

2. Provide a way for users to control which services have permission to manage their roster.

3 Glossary

Remote entity   the entity that wants to modify user's roster.

User   the entity which roster the remote entity wants to have access to.

User's server   the XMPP server User connected to.

Roster   the list of User's contacts as defined in the RFC 6121 RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>..

4 Use Cases

4.1 Remote entity asks for permission to manage user's roster

In order to be able to make any changes to the user's roster the remote entity MUST ask permission to do so first.
NOTE: in order to be able to perform the query, the remote entity MUST have a presence subscription to the User

---

[1] XEP-0100: Gateway Interaction <https://xmpp.org/extensions/xep-0100.html>.
[2] XEP-0100: Gateway Interaction <https://xmpp.org/extensions/xep-0100.html>.
[3] XEP-0144: Roster Item Exchange <https://xmpp.org/extensions/xep-0144.html>.
[4] RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.

Listing 1: Remote entity asks for permission

```
<iq from='icq.example.com' to='juliet@example.com' type='set' id='
    roster_1'>
  <query xmlns='urn:xmpp:tmp:roster-management:0' reason='Manage␣
      contacts␣in␣the␣ICQ␣contact␣list' type='request'/>
</iq>
```

If the presence subscription is absent, the server MUST NOT proceed with the request but MUST respond with the "forbidden" error:

Listing 2: The remote entity has no presence subscription

```
<iq from='juliet@example.com' to='icq.example.com' type='error' id='
    roster_1'>
  <error type='modify'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
    <text xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'>You must have a
        presence subscription to be able to request remote roster
        management service.</text>
  </error>
</iq>
```

The user's server SHOULD then generate a form request using Data Forms (XEP-0004) [5] to client in order to ask user if they are OK with granting the permission to the remote entity. The "challenge" form field is generated by the server and is used to identify the client's response. The server also MUST immediatly answer to the request IQ.
NOTE: if the entity is already granted with the permission, the server SHOULD immediatly answer with a success response and skip querying the user.

Listing 3: Server asks user for the permission

```
<message from='example.com'
         to='juliet@example.com'>
  <body>icq.example.com wants to be able to manage your roster with
      following reason:
    Manage contacts in the ICQ contact list
    Do you want to allow it? Send "yes␣5439123" or "no␣5439123" back,
        pleas.</body>
  <x xmlns='jabber:x:data' type='form'>
    <title>Roster management permission request</title>
    <instructions>icq.example.com wants to be able to manage your
        roster with following reason:
      Manage contacts in the ICQ contact list.
      Do you allow it?</instructions>
    <field type='hidden' var='challenge'><value>5439123</value></field
        >
```

---

[5]XEP-0004: Data Forms <https://xmpp.org/extensions/xep-0004.html>.

```
    <field type='hidden' var='FORM_TYPE'>
      <value>urn:xmpp:tmp:roster-management:0</value>
    </field>
    <field type='boolean'
           label='Allow␣icq.example.com␣to␣edit␣roster?'
           var='answer'>
      <value>1</value>
    </field>
  </x>
</message>
```

The client can answer by submit the form or with a text message response:

Listing 4: Client responds with to the data form

```
<message from='juliet@example.com/home'
         to='example.com'
         xml:lang='en'>
  <x xmlns='jabber:x:data' type='submit'>
    <field var='FORM_TYPE'>
      <value>urn:xmpp:tmp:roster-management:0</value>
    </field>
    <field var='challenge'><value>5439123</value></field>
    <field var='answer'><value>1</value></field>
  </x>
</message>
```

Listing 5: Client responds with a text message response

```
<message from='juliet@example.com/home'
         to='example.com'
         xml:lang='en'>
  <body>yes 5439123</body>
</message>
```

4.1.1 The remote entity is allowed to manage roster

If the user allowed the roster management then the server MUST inform the remote entity that it has been granted the permission:

Listing 6: The server informs the remote entity that it's allowed to manage the User's roster

```
<iq from='juliet@example.com' to='icq.example.com' type='set' id='
    roster_2'>
  <query xmlns='urn:xmpp:tmp:roster-management:0' type="allowed"/>
</iq>
```

3

4.1.2 The remote entity is allowed to manage roster

If the user disallowed the roster management then the server MUST inform the remote entity that it hasn't been granted the permission:

Listing 7: The server informs the remote entity that it's allowed to manage the User's roster

```
<iq from='juliet@example.com' to='icq.example.com' type='set' id='
    roster_2'>
  <query xmlns='urn:xmpp:tmp:roster-management:0' type="rejected"/>
</iq>
```

4.1.3 Reject the permission to manage roster

In order to reject the permission to manage roster it's enough to reject entity's presence subscription:

Listing 8: The user rejects entity's presence subscription

```
<presence to='icq.ecample.com' type='unsubscribed'/>
```

If the presence subscription is restored then the permission is needed to be rerequested as defined above.

4.2 The remote entity requests current user's roster

The remote entity being granted the permission to manage roster can request it from the User's server using usual jabber:iq:roster protocol to be able to edit it:

Listing 9: The remote entity requests current roster

```
<iq from='icq.example.com' to='juliet@example.com' type='get' id='
    roster_5'>
  <query xmlns='jabber:iq:roster'/>
</iq>
```

The server MUST then answer with User's roster including there only the items that belongs to the entity's hostname:

Listing 10: The server responds with the roster

```
<iq to='icq.exampel.com' from='juliet@example.com' type='result' id='
    roster_5'>
  <query xmlns='jabber:iq:roster'>
    <item jid='123456789@icq.example.com'
          name='Romeo'
```

```
                  subscription='both'>
        <group>Friends</group>
      </item>
      <item jid='554323654@icq.example.com'
            name='Mercutio'
            subscription='from'>
        <group>Friends</group>
      </item>
      <item jid='997665667@icq.example.com'
            name='Benvolio'
            subscription='both'>
        <group>Friends</group>
      </item>
    </query>
</iq>
```

4.3 The user updates roster

If client updates roster and this update affects the remote entity's contacts (i.e. belongs to its hostname) then the server MUST forward these items to the remote entity:

Listing 11: The user updates roster

```
<iq from='juliet@example.com/chamber' type='set' id='roster_3'>
  <query xmlns='jabber:iq:roster'>
    <item jid='123456789@icq.example.net'
          name='Romeo'
          subscription='both'>
      <group>Friends</group>
      <group>Lovers</group>
    </item>
  </query>
</iq>
```

Listing 12: The server sends push roster request to the remote entity

```
<iq from='juliet@example.com' to='icq.example.com' type='set' id='
    roster_3'>
  <query xmlns='jabber:iq:roster'>
    <item jid='123456789@icq.example.net'
          name='Romeo'
          subscription='both'>
      <group>Friends</group>
      <group>Lovers</group>
    </item>
  </query>
</iq>
```

4.4 The remote entity updates the user's roster

The remote entity can also send the push query to the roster with the same semantics as specified for the jabber:iq:roster protocol described in the RFC 6121 [6]:

Listing 13: The remote entity sends push roster request

```
<iq to='juliet@example.com' type='set' id='roster_3'>
  <query xmlns='jabber:iq:roster'>
    <item jid='123456789@icq.example.net'
          name='Romeo'
          subscription='both'>
      <group>Friends</group>
      <group>Lovers</group>
    </item>
  </query>
</iq>
```

The server MUST then inform the remote entity of success or an error and in the case of success also forward the push request to all connected User's resources.
If the entity tries to make changes into the items it's not allowed to, the server MUST NOT do any changes in the User's roster but respond to the entity with an error:

Listing 14: The server responds with an error

```
<iq from='juliet@example.com' type='set' id='roster_3'>
  <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
  <text xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'>You have tried to
    modify the item you don't allowed to.</text>
</iq>
```

4.5 Client requests list of components with permissions to edit their roster

User can ask the server to provide a list of components or servers which have permissions to edit their roster.

Listing 15: User asks the server to get list of components which can edit their roster

```
<iq from='juliet@example.com/home' to='icq.example.com' type='get' id=
    'roster_5'>
  <query xmlns='urn:xmpp:tmp:roster-management:0'/>
</iq>
```

In this case, server responds with list of components or servers which can edit user's roster.

---

[6]RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <http://tools.ietf.org/html/rfc6121>.

Listing 16: Server responds with list of componentss which can edit user's roster

```
<iq from='juliet@example.com/home' to='icq.example.com' type='result'
   id='roster_5'>
  <query xmlns='urn:xmpp:tmp:roster-management:0'>
      <item jid='icq.example.com' reason='Manage␣ICQ␣contacts.'/>
      <item jid='j2j.example.com' reason='Manage␣Jabber␣gateway␣
         contacts.'/>
  </query>
</iq>
```

Eventually, user can reject permission granted to component to edit their roster.

Listing 17: User rejects permissions to edit his roster

```
<iq from='juliet@example.com/home' to='icq.example.com' type='set' id=
   'roster_6'>
  <query xmlns='urn:xmpp:tmp:roster-management:0' type="reject"/>
</iq>
```

5 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='urn:xmpp:tmp:roster-management:0'
    xmlns='urn:xmpp:tmp:roster-management:0'
    elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-XXXX: http://www.xmpp.org/extensions/xep-xxxx.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='query'>
    <xs:complexType>
      <xs:attribute name='type' use='required'>
        <xs:simpleType base='xs:NMTOKEN'>
          <xs:enumeration value='request'/>
          <xs:enumeration value='allowed'/>
          <xs:enumeration value='rejected'/>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name='reason' use='optional' type='xs:string'/>
      <xs:sequence minOccurs='0'>
```

```
        <xs:element ref='item' minOccurs='0' maxOccurs='unbounded'/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name='item'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='empty'>
          <xs:attribute name='jid' type='fullJIDType' use='required'/>
          <xs:attribute name='reason' type='xs:string' use='optional'/
            >
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

</xs:schema>
```