



XMPP

XEP-0447: Stateless file sharing

Marvin Wißfeld
<mailto:xmpp@larma.de>
<xmpp:jabber@larma.de>

2022-02-10
Version 0.1.2

Status	Type	Short Name
Experimental	Standards Track	sfs

This specification describes a protocol for stateless asynchronous file sharing with integrity and transport flexibility. It allows clients to provide a good interoperable user experience in combination with Carbons and MAM.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2020 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	Requirements	1
3	Use cases	2
3.1	Sharing a file	2
3.2	Receiving a file	3
3.3	Attaching a source	4
4	Security Considerations	5
5	IANA Considerations	5
6	XMPP Registrar Considerations	6
6.1	Protocol Namespaces	6
7	Acknowledgements	6

1 Introduction

This is a reiteration on [Stateless Inline Media Sharing \(XEP-0385\)](#)¹ with some significant changes:

- No focus on media, generic for every file type.
- No mixed content, body is used for fallback only and not to transmit additional information.
- Using [File metadata element \(XEP-0446\)](#)².
- Using XML for structured data instead of URIs when possible, adding further extensibility (like providing proper means of sharing encrypted files on http servers).
- Not relying on underspecified usage of [References \(XEP-0372\)](#)³.

2 Requirements

- Do not require any server components for easier deployment
- Should work and enable a good UX in multi-user chats like [Multi-User Chat \(XEP-0045\)](#)⁴ and [Mediated Information eXchange \(MIX\) \(XEP-0369\)](#)⁵
- Should work great together with conversation synchronization protocols like [Message Carbons \(XEP-0280\)](#)⁶ and [Message Archive Management \(XEP-0313\)](#)⁷
- Reuse existing protocols for the actual transport of the data, i.e. [Jingle File Transfer \(XEP-0234\)](#)⁸ or [HTTP File Upload \(XEP-0363\)](#)⁹
- Guarantee file integrity
- Enable aggressive caching
- Provide users with metadata, e.g. file size, file type or thumbnail, to help them decide whether or not they want to load the file
- Support referring to third party hosting services
- Backwards compatibility with existing, widely-deployed protocols

¹XEP-0385: Stateless Inline Media Sharing (SIMS) <<https://xmpp.org/extensions/xep-0385.html>>.

²XEP-0446: File metadata element <<https://xmpp.org/extensions/xep-0446.html>>.

³XEP-0372: References <<https://xmpp.org/extensions/xep-0372.html>>.

⁴XEP-0045: Multi-User Chat <<https://xmpp.org/extensions/xep-0045.html>>.

⁵XEP-0369: Mediated Information eXchange (MIX) <<https://xmpp.org/extensions/xep-0369.html>>.

⁶XEP-0280: Message Carbons <<https://xmpp.org/extensions/xep-0280.html>>.

⁷XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

⁸XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

⁹XEP-0363: HTTP File Upload <<https://xmpp.org/extensions/xep-0363.html>>.

3 Use cases

3.1 Sharing a file

To share a file, a user sends a message stanza including `<file-sharing/>` to the intended recipient contact or group. The `<file-sharing/>` element includes a `<file/>` metadata element as described in [File metadata element \(XEP-0446\)](#)¹⁰ as well as a `<sources/>` element. The `<sources/>` element provides one or multiple sources that the receiving client may use to obtain the file.

Listing 1: Sharing summit.jpg with juliet@shakespeare.lit

```
<message to='juliet@shakespeare.lit' from='romeo@montague.lit/resource
  ' id='sharing-a-file'>
  <file-sharing xmlns='urn:xmpp:sfs:0'>
    <file xmlns='urn:xmpp:file:metadata:0'>
      <media-type>image/jpeg</media-type>
      <name>summit.jpg</name>
      <size>3032449</size>
      <dimensions>4096x2160</dimensions>
      <hash xmlns='urn:xmpp:hashes:2' algo='sha3-256'>2
        XarmwTlNxDAMkvymloX3S5+VbylNrJt/l5QyPa+YoU=</hash>
      <hash xmlns='urn:xmpp:hashes:2' algo='id-blake2b256'>2
        AfMGH807UNPTvUVAM9aK13mpCY=</hash>
      <desc>Photo from the summit.</desc>
      <thumbnail xmlns='urn:xmpp:thumbs:1' uri='cid:sha1+
        ffd7c8d28e9c5e82afea41f97108c6b4@bob.xmpp.org' media-type='
        image/png' width='128' height='96' />
    </file>
    <sources>
      <url-data xmlns='http://jabber.org/protocol/url-data' target='
        https://download.montague.lit/4a771ac1-f0b2-4a4a-9700-
        f2a26fa2bb67/summit.jpg' />
      <jinglepub xmlns='urn:xmpp:jinglepub:1' from='romeo@montague.lit
        /resource' id='9559976B-3FBF-4E7E-B457-2DAA225972BB'>
        <description xmlns='urn:xmpp:jingle:apps:file-transfer:5' />
      </jinglepub>
    </sources>
  </file-sharing>
</message>
```

It is RECOMMENDED that the file metadata specifies name, media-type, size and one or multiple hash elements as described in [Use of Cryptographic Hash Functions in XMPP \(XEP-0300\)](#)¹¹. The hash elements provide end-to-end file integrity and allow efficient caching and flexible retrieval methods.

The message MAY include a suitable fallback body. The fallback body MUST NOT include any

¹⁰XEP-0446: File metadata element <<https://xmpp.org/extensions/xep-0446.html>>.

¹¹XEP-0300: Use of Cryptographic Hash Functions in XMPP <<https://xmpp.org/extensions/xep-0300.html>>.

information that is not also represented in `<file-sharing/>`. If the `<sources/>` element includes an `<url-data/>` element that can be represented as a single URL, adding a [Out-of-Band Data \(XEP-0066\)](#)¹² `x-oob` reference is RECOMMENDED for compatibility.

Listing 2: Sharing `summit.jpg` with `juliet@shakespeare.lit` with fallback

```
<message to='juliet@shakespeare.lit' from='romeo@montague.lit/resource
  ' id='sharing-a-file'>
  <file-sharing xmlns='urn:xmpp:sfs:0'>
    <!--{}- ... -{}-->
  </file-sharing>
  <body>Photo from the summit: https://download.montague.lit/4a771ac1-
    f0b2-4a4a-9700-f2a26fa2bb67/summit.jpg</body>
  <x xmlns='jabber:x:oob'><url>https://download.montague.lit/4a771ac1-
    f0b2-4a4a-9700-f2a26fa2bb67/summit.jpg</url></x>
</message>
```

If the message has an empty body, it is RECOMMENDED to add a message processing hint, see [Message Processing Hints \(XEP-0334\)](#)¹³, to indicate the message to be stored in message stores like [Message Archive Management \(XEP-0313\)](#)¹⁴.

Listing 3: Sharing `summit.jpg` with `juliet@shakespeare.lit` without fallback

```
<message to='juliet@shakespeare.lit' from='romeo@montague.lit/resource
  ' id='sharing-a-file'>
  <file-sharing xmlns='urn:xmpp:sfs:0'>
    <!--{}- ... -{}-->
  </file-sharing>
  <store xmlns='urn:xmpp:hints' />
</message>
```

3.2 Receiving a file

On receive of a message including a `<file-sharing/>` element, the receiving entity SHOULD lookup in a local storage, whether the file with any of the provided hashes has already been retrieved and is available. In that case no transfer needs to be initiated and the cached file can be used instead.

If the file is not available locally, the file can be obtained by one of the sources listed in the `<sources/>` element. If further sources have been attached (as described in [Attaching a source](#)), the receiving entity may also try to obtain the file from any of those.

When the source is an `<url-data/>` element as described in [URL Address Information \(XEP-0103\)](#)¹⁵, the receiving entity MAY obtain the file by downloading it from the specified URL.

¹²XEP-0066: Out of Band Data <<https://xmpp.org/extensions/xep-0066.html>>.

¹³XEP-0334: Message Processing Hints <<https://xmpp.org/extensions/xep-0334.html>>.

¹⁴XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

¹⁵XEP-0103: URL Address Information <<https://xmpp.org/extensions/xep-0103.html>>.

If the URL uses HTTP or HTTPS and additional HTTP request information as specified in [HTTP Scheme for URL Data \(XEP-0104\)](#)¹⁶ is provided, the receiving entity SHOULD use such information when obtaining the file. When sending and receiving files using `<url-data/>`, it is RECOMMENDED to prefer secure protocols (e.g. HTTPS, FTPS). Please read [security considerations](#) when implementing support for insecure URLs.

When the source is a `<jingle-pub/>` element as described in [Publishing Available Jingle Sessions \(XEP-0358\)](#)¹⁷, the receiving entity MAY obtain the file using the protocol described in [Publishing Available Jingle Sessions \(XEP-0358\)](#)¹⁸. If a `<hash/>` is provided, the receiving entity MAY obtain the file by requesting it as described in [Jingle File Transfer \(XEP-0234\)](#)¹⁹. If sources of any other type are provided, clients MAY attempt to obtain the files from such sources. The details of obtaining such file are out of scope of this document.

If the `<media-type/>` of the shared file is such that it can be displayed inline, the receiving entity MAY display the file inline. If no `<media-type/>` is provided or the `<media-type/>` indicates that the file can not be displayed inline, i.e. when the media type is `application/octet-stream`, the receiving entity SHOULD NOT display the file inline and instead offer to download it or save it on the users file system.

3.3 Attaching a source

TODO: The following section relies on [Message Attaching \(XEP-0367\)](#)²⁰, however other methods to attach information to another message like the recently proposed [Message Fastening \(XEP-0422\)](#)²¹ might be suitable here as well. This is to be clarified before advancing to Draft. After a user shared a file using one entity and another entity in the conversation obtained it or found it in its local storage, that entity MAY announce that the file is now available with an additional source. This increases availability of the file in case the sender goes offline before all the intended recipients were able to fetch the file. It also allows for peer-to-peer file distribution in group chats.

The entity MUST NOT announce itself as an additional source before verifying that *all* hashes provided match the hash of the file. If no hashes are provided, the entity SHOULD NOT announce itself as an additional source.

The attaching itself is performed by sending a message including a `<sources>` element with further sources using the protocol described in [Message Attaching \(XEP-0367\)](#)²².

Depending on the lifetime of the newly attached source, it may be useful to add a message processing hint, see [Message Processing Hints \(XEP-0334\)](#)²³, to indicate the message to be stored in message stores like [Message Archive Management \(XEP-0313\)](#)²⁴.

¹⁶XEP-0104: HTTP Scheme for URL Data <https://xmpp.org/extensions/xep-0104.html>.

¹⁷XEP-0358: Publishing Available Jingle Sessions <https://xmpp.org/extensions/xep-0358.html>.

¹⁸XEP-0358: Publishing Available Jingle Sessions <https://xmpp.org/extensions/xep-0358.html>.

¹⁹XEP-0234: Jingle File Transfer <https://xmpp.org/extensions/xep-0234.html>.

²⁰XEP-0367: Message Attaching <https://xmpp.org/extensions/xep-0367.html>.

²¹XEP-0422: Message Fastening <https://xmpp.org/extensions/xep-0422.html>.

²²XEP-0367: Message Attaching <https://xmpp.org/extensions/xep-0367.html>.

²³XEP-0334: Message Processing Hints <https://xmpp.org/extensions/xep-0334.html>.

²⁴XEP-0313: Message Archive Management <https://xmpp.org/extensions/xep-0313.html>.

Listing 4: romeo@montague.lit/resource2 attaches itself as an additional source for the file

```
<message to='juliet@shakespeare.lit' from='romeo@montague.lit/  
resource2'>  
  <attach-to id='sharing-a-file' xmlns='urn:xmpp:message-attaching:1' /  
  >  
  <sources xmlns='urn:xmpp:sfs:0'>  
    <jinglepub xmlns='urn:xmpp:jinglepub:1' from='romeo@montague.lit/  
resource2' id='9559976B-3FBF-4E7E-B457-2DAA225972BB'>  
      <description xmlns='urn:xmpp:jingle:apps:file-transfer:5' />  
    </jinglepub>  
  </sources>  
  <store xmlns='urn:xmpp:hints' />  
</message>
```

4 Security Considerations

If a <hash/> using any supported algorithm is provided, the receiving client SHOULD verify that the <hash/> of the announced file matches the obtained file before presenting it to the user. If no <hash/> is provided or the <hash/> elements provided use unsupported algorithms, receiving clients MUST ignore any attached sources from other senders and only obtain the file from the sources announced by the original sender. If no <hash/> is provided or the <hash/> elements provided use unsupported algorithms, receiving clients MUST ignore any sources that use unsecure protocols (e.g. HTTP without TLS).

For most methods of transferring a file proposed through the <sources/> element, obtaining files requires revealing private information like IP addresses to the sending user or third-parties. Sources that do not require revealing private information to untrusted entities SHOULD be preferred by receiving entities. Receiving entities SHOULD ask users for confirmation before obtaining a file, if doing so would require revealing private information to untrusted entities. If the protocol that is used when obtaining the file is not secure (e.g. HTTP without TLS), this SHOULD be considered as if the protocol reveals private information. The security considerations of [File metadata element \(XEP-0446\)](#)²⁵ apply.

5 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)²⁶.

²⁵XEP-0446: File metadata element <<https://xmpp.org/extensions/xep-0446.html>>.

²⁶The 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/>>.

6 XMPP Registrar Considerations

6.1 Protocol Namespaces

The [XMPP Registrar](#) ²⁷ includes 'urn:xmpp:sfs:0' in its registry of protocol namespaces (see <https://xmpp.org/registrar/namespaces.html>).

- urn:xmpp:sfs:0

7 Acknowledgements

Thanks to the authors of [Stateless Inline Media Sharing \(XEP-0385\)](#) ²⁸ which heavily inspired this XEP.

²⁷The 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/>.

²⁸XEP-0385: Stateless Inline Media Sharing (SIMS) <https://xmpp.org/extensions/xep-0385.html>.