# XEP-0447: Stateless file sharing

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

2020-12-30
Version 0.1.1

| 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



## Permissions



## Warranty



## Liability



## 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

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-xxxx) [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 aggresive 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

[1] XEP-0385: Stateless Inline Media Sharing (SIMS) <https://xmpp.org/extensions/xep-0385.html>.
[2] XEP-xxxx: File metadata element <https://xmpp.org/extensions/inbox/file-metadata.html>.
[3] XEP-0372: References <https://xmpp.org/extensions/xep-0372.html>.
[4] XEP-0045: Multi-User Chat <https://xmpp.org/extensions/xep-0045.html>.
[5] XEP-0369: Mediated Information eXchange (MIX) <https://xmpp.org/extensions/xep-0369.html>.
[6] XEP-0280: Message Carbons <https://xmpp.org/extensions/xep-0280.html>.
[7] XEP-0313: Message Archive Management <https://xmpp.org/extensions/xep-0313.html>.
[8] XEP-0234: Jingle File Transfer <https://xmpp.org/extensions/xep-0234.html>.
[9] XEP-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 inteded recipient contact or group. The <file-sharing/> element includes a <file/> metadata element as described in File metadata element (XEP-xxxx) [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
          AfMGH8O7UNPTvUVAM9aK13mpCY=</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

[10] XEP-xxxx: File metadata element <https://xmpp.org/extensions/inbox/file-metadata.html>.
[11] XEP-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 proivded hashes has already been retrieved and is available. In that case no transfer needs to be initated 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.

[12] XEP-0066: Out of Band Data <https://xmpp.org/extensions/xep-0066.html>.
[13] XEP-0334: Message Processing Hints <https://xmpp.org/extensions/xep-0334.html>.
[14] XEP-0313: Message Archive Management <https://xmpp.org/extensions/xep-0313.html>.
[15] XEP-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].

[16] XEP-0104: HTTP Scheme for URL Data <https://xmpp.org/extensions/xep-0104.html>.
[17] XEP-0358: Publishing Available Jingle Sessions <https://xmpp.org/extensions/xep-0358.html>.
[18] XEP-0358: Publishing Available Jingle Sessions <https://xmpp.org/extensions/xep-0358.html>.
[19] XEP-0234: Jingle File Transfer <https://xmpp.org/extensions/xep-0234.html>.
[20] XEP-0367: Message Attaching <https://xmpp.org/extensions/xep-0367.html>.
[21] XEP-0422: Message Fastening <https://xmpp.org/extensions/xep-0422.html>.
[22] XEP-0367: Message Attaching <https://xmpp.org/extensions/xep-0367.html>.
[23] XEP-0334: Message Processing Hints <https://xmpp.org/extensions/xep-0334.html>.
[24] XEP-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 obained 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 entitites. 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-xxxx) [25] apply.

## 5 IANA Considerations

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

[25] XEP-xxxx: File metadata element <https://xmpp.org/extensions/inbox/file-metadata.html>.
[26] The 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.

[27] The 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/>.

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