



XMPP

XEP-0385: Stateless Inline Media Sharing (SIMS)

Tobias Markmann

<mailto:tobias.markmann@isode.com>

<xmpp:tm@ayena.de>

2018-01-25

Version 0.2.1

Status	Type	Short Name
Deferred	Standards Track	sims

This specification describes a protocol for stateless asynchronous media 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 – 2024 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	Related XEPs	1
2.1	Bits of Binary	1
2.2	Jingle File Transfer	1
2.3	HTTP File Upload	2
2.4	Comparison of File Transfer Protocols	2
3	Requirements	2
4	Use Cases	3
4.1	Sharing a photo	3
4.2	Receiving a shared photo	4
5	Business Rules	5
5.1	Transport Method Preference	5
5.2	Media Support	5
5.3	Automatic retrieval of shared media	6
5.4	MUC and MIX related rules	6
5.5	MAM and Carbons related rules	7
5.6	XHTML-IM related rules	7
6	Implementation Notes	8
7	Accessibility Considerations	8
8	Internationalization Considerations	8
9	Security Considerations	9
9.1	Clearing of privacy sensitive metadata	9
9.2	The value and cost of end-to-end media integrity	9
10	Acknowledgements	9
11	IANA Considerations	9
12	XMPP Registrar Considerations	9
12.1	Protocol Namespaces	9
13	XML Schema	10

1 Introduction

File sharing in XMPP has mainly been addressed by synchronous solutions like [SI File Transfer \(XEP-0096\)](#)¹ and [Jingle File Transfer \(XEP-0234\)](#)². However, these extensions only address the transfer of files and there is more to file sharing than the simple transfer of the data.

Extensions that go beyond the simple transfer of data are [File Information Sharing \(XEP-0329\)](#)³ and [HTTP File Upload \(XEP-0363\)](#)⁴. XEP-0329 allows sharing folder structures to other users, allowing them to browse the remote folder and fetch interesting files using existing file-transfer protocols. XEP-0363 describes a protocol to ask a server component for a HTTP storage URL where a client can use HTTP PUT to save a file to and afterwards share the public URL with other users to share the file. While this provides some form of asynchronous file sharing it does not provide integrity protection and requires a server component.

This proposal aims to provide a protocol that will enable XMPP clients to implement a great user experience (UX) around the process of sharing media in conversations. Shared media can take any form of static media like photos, videos, documents, compressed archives, etc. This is directly reflected in the requirements of this extension lined out in the following sections.

2 Related XEPS

The state of sharing media with chat partners in the XMPP community is a protocol zoo in 2016. There are three major protocols for sharing media in XMPP.

2.1 Bits of Binary

[Bits of Binary \(XEP-0231\)](#)⁵ is designed for small media, i.e. less than 8 KB in size, that is hosted server-side and transferred Base64 encoded in-band of an existing XMPP stream. Example use-cases are custom emoticons that are referenced in [XHTML-IM \(XEP-0071\)](#)⁶ img-tags, or thumbnails for [Jingle File Transfer \(XEP-0234\)](#)⁷.

2.2 Jingle File Transfer

[Jingle File Transfer \(XEP-0234\)](#)⁸ describes a peer-to-peer protocol for synchronous file-transfer between two XMPP entities. It attempts a direct transmission, followed by a proxied

¹XEP-0096: SI File Transfer <<https://xmpp.org/extensions/xep-0096.html>>.

²XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

³XEP-0329: File Information Sharing <<https://xmpp.org/extensions/xep-0329.html>>.

⁴XEP-0363: HTTP File Upload <<https://xmpp.org/extensions/xep-0363.html>>.

⁵XEP-0231: Bits of Binary <<https://xmpp.org/extensions/xep-0231.html>>.

⁶XEP-0071: XHTML-IM <<https://xmpp.org/extensions/xep-0071.html>>.

⁷XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

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

transmission, via [Jingle SOCKS5 Bytestreams Transport Method \(XEP-0260\)](#)⁹. If neither works it will fallback to [Jingle In-Band Bytestreams Transport Method \(XEP-0261\)](#)¹⁰ which will transfer the data inband of the existing XMPP stream.

2.3 HTTP File Upload

[HTTP File Upload \(XEP-0363\)](#)¹¹ was designed as a simpler to implement alternative to [Bits of Binary \(XEP-0231\)](#)¹². This is achieved by reusing the HTTP APIs in today's mobile and language SDKs. It requires a server component where clients can request HTTP URLs to upload data to and share the corresponding download URL as part of plain text in a conversation.

2.4 Comparison of File Transfer Protocols

Protocol

[Bits of Binary \(XEP-0231\)](#) XEP-0231: Bits of Binary <<https://xmpp.org/extensions/xep-0231.html>>.

[Jingle File Transfer \(XEP-0234\)](#) XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

[HTTP File Upload \(XEP-0363\)](#) XEP-0363: HTTP File Upload <<https://xmpp.org/extensions/xep-0363.html>>.

3 Requirements

- Not require server components to work to ease deployment
- Can be improved by server components for taking load of clients
- Media sharing should work and enable a good UX in multi-user chats like [Multi-User Chat \(XEP-0045\)](#)¹³ and [Mediated Information eXchange \(MIX\) \(XEP-0369\)](#)¹⁴

⁹XEP-0260: Jingle SOCKS5 Bytestreams Transport Method <<https://xmpp.org/extensions/xep-0260.html>>.

¹⁰XEP-0261: Jingle In-Band Bytestreams Transport Method <<https://xmpp.org/extensions/xep-0261.html>>.

¹¹XEP-0363: HTTP File Upload <<https://xmpp.org/extensions/xep-0363.html>>.

¹²XEP-0231: Bits of Binary <<https://xmpp.org/extensions/xep-0231.html>>.

¹³XEP-0045: Multi-User Chat <<https://xmpp.org/extensions/xep-0045.html>>.

¹⁴XEP-0369: Mediated Information eXchange (MIX) <<https://xmpp.org/extensions/xep-0369.html>>.

- Media sharing should work great together with conversation synchronization protocols like [Message Carbons \(XEP-0280\)](#)¹⁵ and [Message Archive Management \(XEP-0313\)](#)¹⁶
- Reuse existing protocols for the actual transport of the data, i.e. [SI File Transfer \(XEP-0096\)](#)¹⁷, [Jingle File Transfer \(XEP-0234\)](#)¹⁸ or [HTTP File Upload \(XEP-0363\)](#)¹⁹
- 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 media file
- Support referring to third party hosting services

4 Use Cases

4.1 Sharing a photo

To share a photo, or any kind of media, a user sends a message stanza to the contact. If the message has an empty body, it is recommended to add a message processing hint, see [Message Processing Hints \(XEP-0334\)](#)²⁰, to indicate the message to be stored in message stores like [Message Archive Management \(XEP-0313\)](#)²¹.

Clients supporting [HTTP File Upload \(XEP-0363\)](#)²² can upload the media file to a HTTP service and add the corresponding HTTP URL to the sources.

Listing 1: Sharing an image with a contact

```
<message to='julient@shakespeare.lit' from='romeo@montague.lit'>
  <body>Look at the nice view from the summit.</body>
  <reference xmlns='urn:xmpp:reference:0' begin='17' end='20' type='
    data'>
    <media-sharing xmlns='urn:xmpp:sims:1'>
      <file xmlns='urn:xmpp:jingle:apps:file-transfer:5'>
        <media-type>image/jpeg</media-type>
        <name>summit.jpg</name>
        <size>3032449</size>
        <hash xmlns='urn:xmpp:hashes:2' algo='sha3-256'>2
          XarmwTlNxDAMkvyml0X3S5+VbylNrJt/15QyPa+YoU=</hash>

```

¹⁵XEP-0280: Message Carbons <<https://xmpp.org/extensions/xep-0280.html>>.

¹⁶XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

¹⁷XEP-0096: SI File Transfer <<https://xmpp.org/extensions/xep-0096.html>>.

¹⁸XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

¹⁹XEP-0363: HTTP File Upload <<https://xmpp.org/extensions/xep-0363.html>>.

²⁰XEP-0334: Message Processing Hints <<https://xmpp.org/extensions/xep-0334.html>>.

²¹XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

²²XEP-0363: HTTP File Upload <<https://xmpp.org/extensions/xep-0363.html>>.

```

    <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>
    <reference xmlns='urn:xmpp:reference:0' type='data' uri='
      https://download.montague.lit/4a771ac1-f0b2-4a4a-9700-
      f2a26fa2bb67/summit.jpg' />
    <reference xmlns='urn:xmpp:reference:0' type='data' uri='
      xmpp:romeo@montague.lit/resource?jingle;id=9559976B-3FBF-4
      E7E-B457-2DAA225972BB' />
  </sources>
</media-sharing>
</reference>
</message>

```

The file element is the same as from [Jingle File Transfer \(XEP-0234\)](#)²³. It MUST specify media-type, size, description, and one or multiple hash elements as described in [Use of Cryptographic Hash Functions in XMPP \(XEP-0300\)](#)²⁴. The hash elements are essential as they provide end-to-end file integrity and allow efficient caching and flexible retrieval methods.

4.2 Receiving a shared photo

On receive of a reference to a <media-sharing> element inside a message, a client SHOULD lookup in a local storage, whether the media with any of the provided hashes has already been retrieved and is available. In that case no transfer needs to be initiated and the image can be displayed in-line of the chat.

If the media file is not available locally, the media file can be obtained by one of the references in the <sources> element. If a client support HTTP downloads, it can simply download HTTP references.

If not, it can fetch the media file via a [Publishing Available Jingle Sessions \(XEP-0358\)](#)²⁵ URI reference in the sources and initiate a Jingle File-Transfer. If the client does not support [Publishing Available Jingle Sessions \(XEP-0358\)](#)²⁶, it can attempt fetching the media file via [Jingle File Transfer \(XEP-0234\)](#)²⁷ by using the hash elements in the file element as described in [Jingle File-Transfer](#).

A client MAY retrieve the file from other sources than these mentioned in the sources element. This may be via [Jingle File Transfer \(XEP-0234\)](#)²⁸ from the senders' other resources

²³XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

²⁴XEP-0300: Use of Cryptographic Hash Functions in XMPP <<https://xmpp.org/extensions/xep-0300.html>>.

²⁵XEP-0358: Publishing Available Jingle Sessions <<https://xmpp.org/extensions/xep-0358.html>>.

²⁶XEP-0358: Publishing Available Jingle Sessions <<https://xmpp.org/extensions/xep-0358.html>>.

²⁷XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

²⁸XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

or from a media caching service located at the local service. The standardization of such cache is out of scope for this document.

Regardless of the transport method used to obtain the file, the received content MUST be verified against one of the hashes. If the verification fails, the retrieved content MUST be discarded and retrieval using a different source can be attempted.

5 Business Rules

5.1 Transport Method Preference

This XEP delegates actual transport of the media data to one of the existing file-transfer XEPs. Thus a client supporting this XEP MUST implement [Jingle File Transfer \(XEP-0234\)](#)²⁹ and [HTTP File Upload \(XEP-0363\)](#)³⁰.

If a users server supports [HTTP File Upload \(XEP-0363\)](#)³¹, it SHOULD upload the file to the service and add the retrieval URL to the <sources> tag, unless the user specifically asked to not store media in the cloud.

Using [HTTP File Upload \(XEP-0363\)](#)³² for media file transfer highly increases the UX, since the HTTP server has a higher availability than XMPP end-user clients and can easily handle the load of lots of requests that result from sharing media in [Multi-User Chat \(XEP-0045\)](#)³³ and [Mediated Information eXchange \(MIX\) \(XEP-0369\)](#)³⁴ rooms.

5.2 Media Support

Sharing the raw data of media does not provide a complete user experience. Clients ideally need to be able to display the media inline of the chat. For this we set baseline requirements for audio, video and picture formats, that a client supports to display. These requirements are shown in the following table.

A client usually will always send in one format per media type, if it creates that media itself.

Media Type	Mime Type	Format/Container	Codec	Requirement	Comment
Audio	audio/m4a	MPEG4	AAC	SHOULD	Can be encoded/decoded by stock Android and iOS systems.

²⁹XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>.

³⁰XEP-0363: HTTP File Upload <<https://xmpp.org/extensions/xep-0363.html>>.

³¹XEP-0363: HTTP File Upload <<https://xmpp.org/extensions/xep-0363.html>>.

³²XEP-0363: HTTP File Upload <<https://xmpp.org/extensions/xep-0363.html>>.

³³XEP-0045: Multi-User Chat <<https://xmpp.org/extensions/xep-0045.html>>.

³⁴XEP-0369: Mediated Information eXchange (MIX) <<https://xmpp.org/extensions/xep-0369.html>>.

Media Type	Mime Type	Format/Container	Codec	Requirement	Comment
Image	image/jpeg	-	JPEG	SHOULD	Supported on common desktop and mobile systems. Use for photos.
Image	image/png	-	PNG	SHOULD	Supported on common desktop and mobile systems. Use for non-photos.
Video	image/gif	-	GIF	SHOULD	Widespread history animation format supported everywhere.
Video	video/mp4	MPEG4	H.264 AVC	SHOULD	Can be encoded/decoded by stock Android and iOS systems.

5.3 Automatic retrieval of shared media

Depending on the size of the shared media, a client MAY want to automatically download and display the media instead of fetching and displaying the thumbnail. The size threshold depends on the network environment the client currently runs in.

If a client supports automatic retrieval it MUST disclose this feature to the end user and provide a way to disable it, as it may result in high network traffic.

5.4 MUC and MIX related rules

In cases where media is shared in a [Multi-User Chat \(XEP-0045\)](#)³⁵ or [Mediated Information eXchange \(MIX\) \(XEP-0369\)](#)³⁶ room the sender has to expect that a large number of clients

³⁵XEP-0045: Multi-User Chat <<https://xmpp.org/extensions/xep-0045.html>>.

³⁶XEP-0369: Mediated Information eXchange (MIX) <<https://xmpp.org/extensions/xep-0369.html>>.

may retrieve the shared media automatically. Ideally multiple sources, including HTTP or other high availability sources, are provided in the <sources> tag of the <media-sharing> tag in case the media is shared in a MUC/MIX room.

TODO: Describe protocol for MIX members to advertise media availability to peers in a dedicated MIX channel PubSub node. Maybe as a dedicated XEP.

5.5 MAM and Carbons related rules

For the media sharing described in this XEP to work, it is REQUIRED for MAM to store the whole stanza instead of only the body content. If the MAM component of the user's server strips away the <media-sharing> tag, any shared media will be missing in archived messages. If sensitive media is shared a client MAY add relevant hints for the server via [Message Processing Hints \(XEP-0334\)](#)³⁷.

5.6 XHTML-IM related rules

To refer to shared media in a XHTML-IM message, this XEP takes advantage of the requirement for hash elements in the file metadata and [RFC 6920](#)³⁸ and its ni URI format. Using the URI format, XHTML-IM can easily refer to media that is attached to a message via a <media-sharing> element, as shown in the following example.

Listing 2: Sharing an image with a contact

```
<message to='julient@shakespeare.lit' from='romeo@montague.lit'>
  <body>Look at the nice view from the summit.</body>
  <html xmlns='http://jabber.org/protocol/xhtml-im'>
    <body xmlns='http://www.w3.org/1999/xhtml'>
      <p>Look at the nice <p style='font-weight:bold;_display:inline
        ">view</p>_from_the_summit.</p>
      <img_src="ni:///sha3-256;
        wqfDv80Gw7jCvx7Dl2ZRw4FHVskgYcOWYs014oKsw79Nw6Q7Sc064oCaw5gKS -
        KCrM04Q0o"_/>
    </body>
  </html>
  <reference xmlns='urn:xmpp:reference:0' _begin='17' _end='20' _type='
    data'>
    <media-sharing xmlns='urn:xmpp:sims:1'>
      <file xmlns='urn:xmpp:jingle:apps:file-transfer:5'>
        <media-type>image/jpeg</media-type>
        <name>summit.jpg</name>
        <size>3032449</size>
```

³⁷XEP-0334: Message Processing Hints <<https://xmpp.org/extensions/xep-0334.html>>.

³⁸RFC 6920: Naming Things with Hashes <<http://tools.ietf.org/html/rfc6920>>.

```

.....<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>
.....<reference_xmlns='urn:xmpp:reference:0'_type='data'_uri='
      https://download.montague.lit/4a771ac1-f0b2-4a4a-9700-f2a26fa2bb67
      /summit.jpg'/'>
.....<reference_xmlns='urn:xmpp:reference:0'_type='data'_uri='
      xmpp:romeo@montague.lit/resource?jingle-ft'/'>
.....</sources>
.....</media-sharing>
.....</reference>
.....</message>

```

This way the client can acquire the content addressable resource mentioned in the img-tag in the XHTML-IM message, and when finished show in in the rendered XHTML-IM message.

6 Implementation Notes

7 Accessibility Considerations

The size element in the file element provides clients to automatically load small files and if not provide the users with a hint on how long a transfer might take.

The OPTIONAL thumbnail element in the file element improves the user experience as it provides further hints for users on whether the file could be of interest to them.

The desc element in the file element is critical for clients to enable them to provide accessibility to users who use screen readers.

8 Internationalization Considerations

OPTIONAL.

9 Security Considerations

9.1 Clearing of privacy sensitive metadata

Mobile devices are able to attach the geographic location of where a photo was taken to the photo. It is RECOMMENDED that a client implementing this XEP attempts to detect privacy exposing metadata in media shared and if found provides the user with an option to clear the media of such metadata.

9.2 The value and cost of end-to-end media integrity

Requiring end-to-end media integrity prevents trivial server side optimizations or other processing on shared media as it will change the cryptographic hash of the media file. On the other hand, requiring a matching cryptographic hash guarantees that everybody sees the exact same media a user has shared in a group conversation.

10 Acknowledgements

Thanks to Kim Alvefur, Emmanuel Gil Peyrot, Kevin Smith, Nicolas Vérité, and Florian Schmaus for their helpful comments.

11 IANA Considerations

REQUIRED.

12 XMPP Registrar Considerations

The [XMPP Registrar](#)³⁹ includes the following information in its registries.

12.1 Protocol Namespaces

The XMPP Registrar will include the following namespace in its registry of protocol namespaces at <https://xmpp.org/registrar/namespaces.html>:

- urn:xmpp:sims:1

³⁹The 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/>.

13 XML Schema

REQUIRED for protocol specifications.