XEP-0504: Data Policy

Jérôme Poisson
mailto:goffi@goffi.org
xmpp:goffi@jabber.fr

2025-07-09
Version 0.1.0

Status	Type	Short Name
Experimental	Standards Track	data-policy

This document specifies metadata on how an entity handles its data (encryption, data retention, etc).

Legal

Copyright

Permissions

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

It is important for a service user to know how its data are handled: where data are stored, whether encryption is used and how, which jurisdiction applies, etc.
This document specifies fields to use with Service Discovery Extensions (XEP-0128) [1] to expose that information and is usable with any kind of XMPP entity (XMPP server, gateways, pubsub services, etc.). It is expected that those data are properly filled and exposed to end-users in an easy-to-understand way by XMPP clients.

2 Requirements

The goals of this specification are:

- Expose enough data policy information so end-users can obtain detailed information on how their data are handled.
- Be usable with any kind of XMPP entity.
- Use as much as possible existing specifications to expose those data.
- Avoid duplicating information: if information is already easily available with another XMPP specification/disco feature, and no other important information is needed, do not duplicate it.

3 Exposing Data Policy

Data policy is exposed via Service Discovery Extensions (XEP-0128) [2]: in response to a disco#info query sent to the bare JID of the entity, the implementation MUST return a data form using the 'FORM_TYPE' of "urn:xmpp:data-policy:0" as specified in Service Discovery Extensions (XEP-0128) [3].
This form can be used with any kind of XMPP entity. It is specially expected to be used with XMPP entities having one of the following identity categories: "collaboration", "conference", "pubsub", "server", and "store", but it can be used with any kind of entity.
If the service is proxying data to another (e.g., a gateway, or a storage service using another provider), the data form with a 'FORM_TYPE' of 'urn:xmpp:data-policy:0' applies to the policy of the service/gateway itself. The policy of the legacy network is specified by using the scope "urn:xmpp:data-policy:identity:<category>:<type>:0" where *<category>* and *<type>* are the category and type of the corresponding disco identity, as found in <https://xmpp.org/registrar/disco-categories.html>. That means that a service exposing data policy MUST have at least one main form with the "urn:xmpp:data-policy:0"

[1] XEP-0128: Service Discovery Extensions <https://xmpp.org/extensions/xep-0128.html>.
[2] XEP-0128: Service Discovery Extensions <https://xmpp.org/extensions/xep-0128.html>.
[3] XEP-0128: Service Discovery Extensions <https://xmpp.org/extensions/xep-0128.html>.

scope for the service itself, and SHOULD have one identity-tied form per legacy service. This way, a user can see exactly how its data are used when using a gateway with a specific legacy network.
The rest of this section describes the fields that MAY be used in this form.

3.1 Auth Mechanism

A field indicates how login data is used (login/password or equivalent data used for authentication). This field MUST have a 'var' attribute set to "auth_data", must be of type "list-single", and MUST have one of the following values:

- **no_auth**: the service does not require authentication.
- **plain**: the service receives the authentication data in plain text, meaning that it can access them and potentially copy them.
- **hidden**: the auth data are used but not seen by the service, because a technique to hide it (such as using a challenge to verify it) is used.
- **restricted**: no full authentication is used, instead a temporary access is given via a mechanism such as a token or OAuth, and the access is restricted in scope (i.e., what it is possible to do with the account) and time.

3.2 Encryption

While clients can obviously already determine if they can use end-to-end encryption with a service, they have no way to know if a service acting as a proxy/gateway does send its data encrypted or not to third-party services. For instance, a client can send data end-to-end encrypted to a gateway using OMEMO Encryption (XEP-0384) [4], and then the gateway decrypts it and may send it in a totally different way, even unencrypted! This could give a false sense of security to the end-user, as it would appear as end-to-end encrypted in most clients (while in reality it's only e2ee between the client and the gateway/proxy).
To make the situation more clear, a data transmission field SHOULD be used, indicating how the data is transmitted from and to legacy services (this field doesn't specify how data is transmitted between the XMPP client and the service itself). The field MUST have a 'var' attribute set to "data_transmission", a type of "list-single". The value MUST be one of the following:

- **plain**: the data is sent without any encryption. It may be viewed by anybody with access to the network between the XMPP service and the destination.

[4] XEP-0384: OMEMO Encryption <https://xmpp.org/extensions/xep-0384.html>.

- **encrypted**: the data is encrypted, but not end-to-end (e.g., with something such as TLS). It may be viewed by the XMPP service, the legacy service administrators, and the destination.
- **e2e**: the data is encrypted using an end-to-end algorithm, which SHOULD be specified with the encryption algorithm field (see below). It may be seen by the XMPP service and the destination.
- **gre**: Gateway Relayed Encryption is used: the data is encrypted by the XMPP client, passed by the XMPP service, and decrypted by the destination. Only the destination can see the data. When used, XMPP clients MUST ensure that GRE is actually used by the gateway.

When ”encrypted” or ”e2e” values are used in the previous field, the algorithm used SHOULD be specified with a field with a ”var” attribute of ”encryption_algorithm”, with a type of ”text-single”, and with the human-readable name of the algorithm used (e.g., ”TLSv1.3”). This is not necessary when ”gre” is used, as this data is already known by the client in this case.

3.3 Data Retention

The data retention field indicates for how long data such as messages or files are stored on the service. The field MUST have a ’var’ attribute set to ”data_retention”, be of type ”text-single”, and MUST indicate how long data is stored (at most) in hours. The following values have special meaning:

- **0**: data is not stored and is used in transit in this service. Note that the data can still be stored by another service if this one is a proxy or a gateway (this can be checked with the service-tied form).
- **infinite**: data is stored without automatic purge. Depending on the service, the user may or may not handle data deletion themselves (this is specified with the ”data_deletion” field, see below).
- **unknown**: data retention of this service cannot be determined. This can happen when the service is a gateway to another service, and data retention policy is not known.

3.4 Data Deletion

The data deletion field indicates if a user can explicitly delete data on this service (via ad-hoc commands, pubsub semantics, or any easy-to-use method). The field MUST have a ’var’ attribute set to ”data_deletion”, and be of type ”boolean”. It defaults to ”false”.

3.5 Encryption at Rest

When data retention has a value other than ”0”, it is important to know how the data is encrypted at rest. This is exposed with a field which MUST have a ’var’ attribute set to ”encryption_at_rest”, a type of ”boolean”, set to ”true” when data is encrypted at rest. It defaults to ”false”.

3.6 Terms of Service

A link to Terms of Service (ToS) can be specified with a field which MUST have a ’var’ attribute set to ”tos” and a type of ”text-single”. The value MUST be an URI to the ToS. The link can be an xmpp: URI (e.g., to a Pubsub item), an http(s) one, or using any kind of relevant scheme. As for other fields, if a proxy or a gateway is used, the ToS of the main scope applies to the service itself, while the ToS of legacy services use the identity-scoped forms.

3.7 Location

Physical location of the data is very important information for the user: it determines the jurisdiction that applies, and it also indicates risks (e.g., natural disasters, geopolitical considerations, war zones, risks of spying, etc.).
There is already a specification to indicate the location of an XMPP entity: User Geolocation (XEP-0080) [5], which should be used to indicate the location of the server via the pep node ”http://jabber.org/protocol/geoloc” as explained in the specification. However, data may be in different locations at once (with a cluster of servers), in which case multiple items should be used in the pep node, one per data cluster. Item IDs may give a hint on the related data (e.g., cluster name), and XEP-0080’s ”description” field should be used to give details on this cluster and when/how data is stored there.
The ”region” field of XEP-0080 SHOULD be duly specified, as it is information of major importance in the context of data policy: the law may differ from one administrative region to another within a given country.

3.8 Data Export

A field indicating whether users can export all their data from the service. This field MUST have a ’var’ attribute set to ”data_export” and a type of ”boolean”. The default value is ”false”.

3.9 Access Policy

A field indicating who has access to user accounts or data. This field MUST have a ’var’ attribute set to ”access_policy”, a type of ”list-multi”, and MUST include one or more of the

[5] XEP-0080: User Geolocation <https://xmpp.org/extensions/xep-0080.html>.

following values:

- **admins**: Administrators of the service can access user data for operational or security purposes (e.g., account management, system maintenance).
- **moderators**: Moderators (e.g., for group chats or blog comments) can access user data within their moderation scope (e.g., content review, enforcement of community guidelines).
- **organization_member**: Any member of the organization owning the service can access user data (e.g., employees, contractors under the organization's control).
- **government**: Government or legal authorities can access user data under legal requirements (e.g., court orders, national security requests).
- **advertisers**: Third-party advertisers or ad networks can access user data for targeted advertising or analytics.
- **partners**: Business partners or affiliated services can access user data under contractual or collaborative agreements.
- **none**: No entity (other than the user) can access user data. If used, this value MUST be the only one selected.

3.10 Full Erasure

A field indicating whether users can fully erase their account and associated data. This field MUST have a 'var' attribute set to "full_erasure" and a type of "boolean". The default value is "false".

3.11 Backup Frequency

A field indicating how often backups of user data are performed. This field MUST have a 'var' attribute set to "backup_frequency" and a type of "text-single". The value MUST specify the maximum frequency in hours between backups. A value of "0" indicates that no backups are performed. For example, a value of "24" indicates backups occur at most every 24 hours (i.e., daily). This field is RECOMMENDED for services storing user data.

3.12 Backup Retention

A field indicating how long backups are retained. This field MUST have a 'var' attribute set to "backup_retention" and a type of "text-single". The value MUST specify the maximum duration in hours. The following special values can be used:

- **0**: No backup is done.
- **infinite**: Backups are retained indefinitely.
- **unknown**: Backup retention policies are not publicly disclosed.

3.13 Extra Infos

Human-readable information can be added with a field which MUST have a 'var' attribute set to "extra_info" and a type of "text-multi". This is useful to add extra details, precision, or any kind of information that end-users may need to know. The "xml:lang" attribute of the disco#info request can be used to determine the language of the description returned by the service.

4 Summary

Below is a table summarizing all fields defined in this XEP for data policy discovery. All fields are optional. Fields with special constraints are noted in the "Comment" column.

Name	Field (var)	Type	Meaning	Comment
Auth Mechanism	auth_data	list-single	How authentication data is handled by the service	Values: no_auth, plain, hidden, restricted
Data Transmission	data_transmission	list-single	How data is transmitted to/from legacy services	Values: plain, encrypted, e2e, gre
Encryption Algorithm	encryption_algorithm	text-single	Name of the encryption algorithm used	Required if data_transmission is "encrypted" or "e2e"
Data Retention	data_retention	text-single	Maximum time (in hours) data is stored	Special values: 0, infinite, unknown
Data Deletion	data_deletion	boolean	Whether users can explicitly delete data	Defaults to false
Encryption at Rest	encryption_at_rest	boolean	Whether data is encrypted when stored	Defaults to false

Name	Field (var)	Type	Meaning	Comment
Terms of Service	tos	text-single	URI to the service's Terms of Service	May be xmpp:, http(s), or other URI schemes
Data Export	data_export	boolean	Whether users can export their data	Defaults to false
Access Policy	access_policy	list-multi	Entities that can access user data	Values: admins, moderators, organization_member, government, advertisers, partners, none
Full Erasure	full_erasure	boolean	Whether users can fully erase their account and data	Defaults to false
Backup Frequency	backup_frequency	text-single	Maximum interval (in hours) between backups	Value in hours; ”0” indicates no backup
Backup Retention	backup_retention	text-single	Maximum time (in hours) backups are retained	Special values: 0, infinite, unknown
Extra Info	extra_info	text-multi	Human-readable additional information	

5 Example

This example shows a gateway providing SMTP/IMAP bridging with two forms in its disco#info response: one for the main gateway service (which does not store data) and one for the SMTP identity (which clarifies that policies depend on the user's chosen external server).

Listing 1: Entity queries gateway for data policy

```
<iq type=’get’
  from=’juliet@capulet.lit/balcony’
  to=’gateway@example.org’
  id=’disco_1’>
  <query xmlns=’http://jabber.org/protocol/disco#info’/>
</iq>
```

Listing 2: Gateway responds with two data-policy forms

```
<iq type=’result’
    from=’gateway@example.org’
    to=’juliet@capulet.lit/balcony’
    id=’disco_1’>
  <query xmlns=’http://jabber.org/protocol/disco#info’>
    <identity category=’gateway’ type=’smtp’/>
    <feature var=’http://jabber.org/protocol/disco#info’/>
    <x xmlns=’jabber:x:data’ type=’result’>
      <!-{}- Main data-policy form for the gateway itself -{}->
      <field var=’FORM_TYPE’ type=’hidden’>
        <value>urn:xmpp:data-policy:0</value>
      </field>
      <field var=’auth_data’ type=’list-single’>
        <value>plain</value>
      </field>
      <field var=’data_transmission’ type=’list-single’>
        <value>plain</value>
      </field>
      <field var=’data_retention’ type=’text-single’>
        <value>0</value>
      </field>
      <field var=’data_deletion’ type=’boolean’>
        <value>false</value>
      </field>
    </x>

    <x xmlns=’jabber:x:data’ type=’result’>
      <!-{}- Identity-scoped form for SMTP gateway -{}->
      <field var=’FORM_TYPE’ type=’hidden’>
        <value>urn:xmpp:data-policy:identity:gateway:smtp:0</value>
      </field>
      <field var=’extra_info’ type=’text-multi’>
        <value>This gateway acts as a relay to external IMAP/SMTP
           servers. Data policies depend entirely on the external
           server chosen by the user. This gateway does not store or
           process user data.</value>
      </field>
    </x>
  </query>
```

```
</iq>
```

6 Implementation Notes

Client developers are encouraged to present data policy information in ways that are intuitive and accessible to all users, including those without technical expertise. While this specification defines detailed metadata fields, clients should prioritize visual indicators (e.g., security badges, warning icons) to summarize key privacy and security aspects at a glance. For example:

- Use color-coded badges to indicate encryption status (e.g., green for end-to-end encryption, red for unencrypted transmission).
- Display warning symbols for services with unclear data retention policies or third-party access.
- Provide tooltips or expandable sections to show technical details on demand.
- Include a security rating (e.g., a percentage score, a mark out of 10, or letter grades) to give users a quick overview of a service's overall security posture.

This approach ensures users receive actionable insights without being overwhelmed by technical specifications, and helps them make informed decisions about which services to trust.

7 Security Considerations

The information exposed in this document is highly useful to end-users to understand what happens to their data. However, it can also provide information to potential attackers (e.g., server location, who can access data, etc.). Service administrators should keep this in mind and find the right balance between providing legitimate end-user information and avoiding disclosure of too many details usable by attackers.

8 IANA Considerations

TODO

9 XMPP Registrar Considerations

TODO

10 XML Schema

TODO