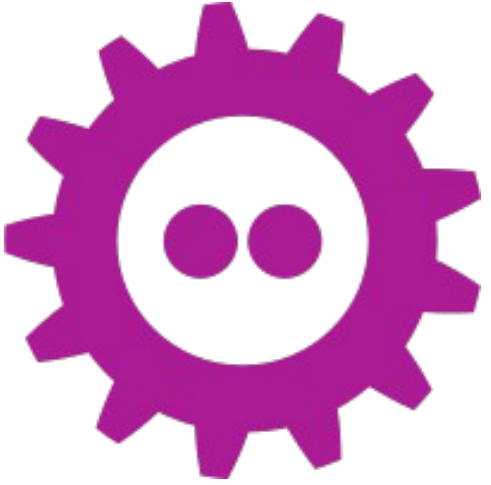

privacy
Winfried
Tilanus
e-health
information
identity
data
patient
analysis
system
practitioner
healthcare
treatment
relationship
relation
people
example
might
control
automated
patients
MEDDEV
WGBO
interventions
research
definition
approach
directive
two
new
third
relationships
become
care
systems
public
society
environment
medical
within
safe
surveillance
ethical
social
like
bubble
clear
legal
paper
one
based
use
organizations
autonomy
certain
results
forced
positioning
many
online
techniques
amount
second
need
intervention
quality
somebody
context
look
active
health
expectations
records
able
make
implications
always
makes
becomes
less
even
big
first
consent
used
digital
done
level
limits
created
expect
problems
evidence
informed
better
customers
want
help
possible
lot
kind
quite
panopticon
process
power
relations
time
patterns
understand
behaviour
question
normative
without
processes
behavior
influence
case
something
potential
much
way
edge
problem
party
individual
also
demands
limited
Also
digitalization
ability
user
tailored
inside
part
must
panopticons
using
take
concept
different
large
software
tailoring
scale
may
events
see
try

XMPP
get your shopping cart ready!

Winfried Tilanus
xmpp:winfried@tilanus.com
mailto:winfried@tilanus.com
Fosdem’20 2020-02-02

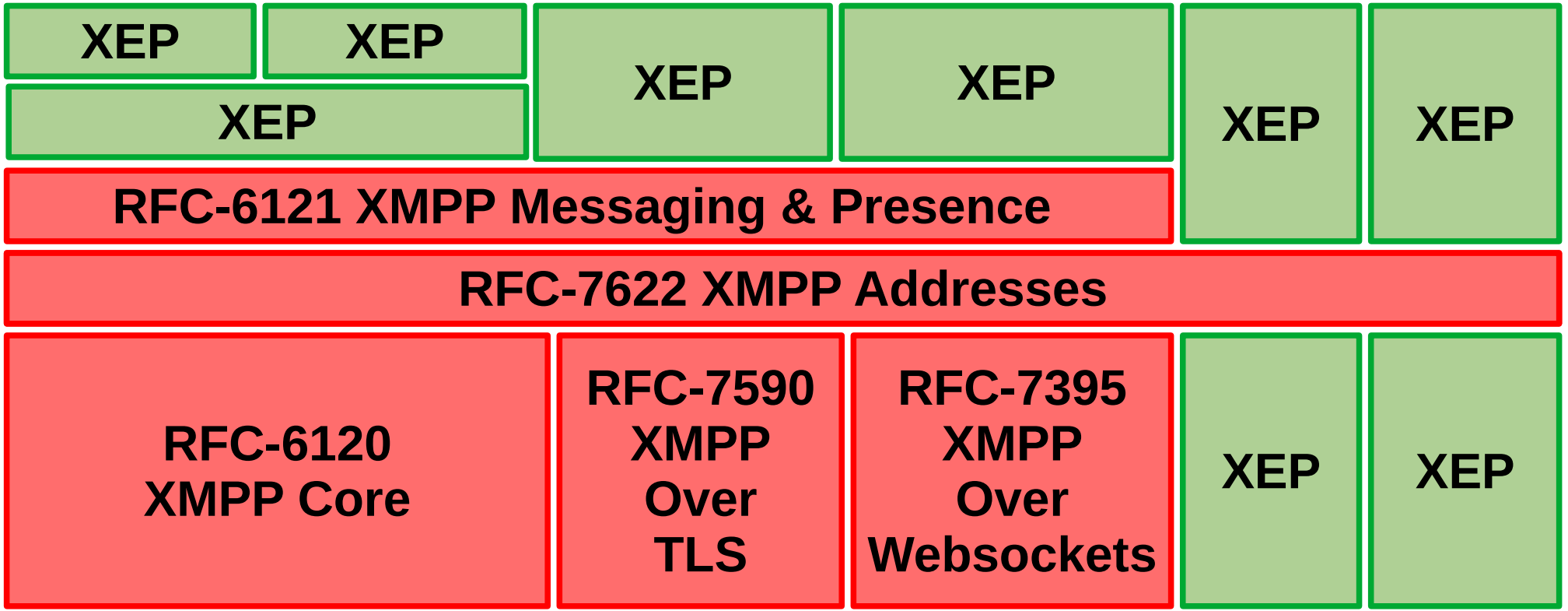


XMPP: a modular protocol

Core and extensions

- Core
  - Basic functionalities
  - Maintained by IETF & IANA
  - Published in RFC’s
- Extensions (XEPs)
  - Whatever you want to add to the core
  - Anybody can write them
  - Can be private, can be published

XMPP Stack

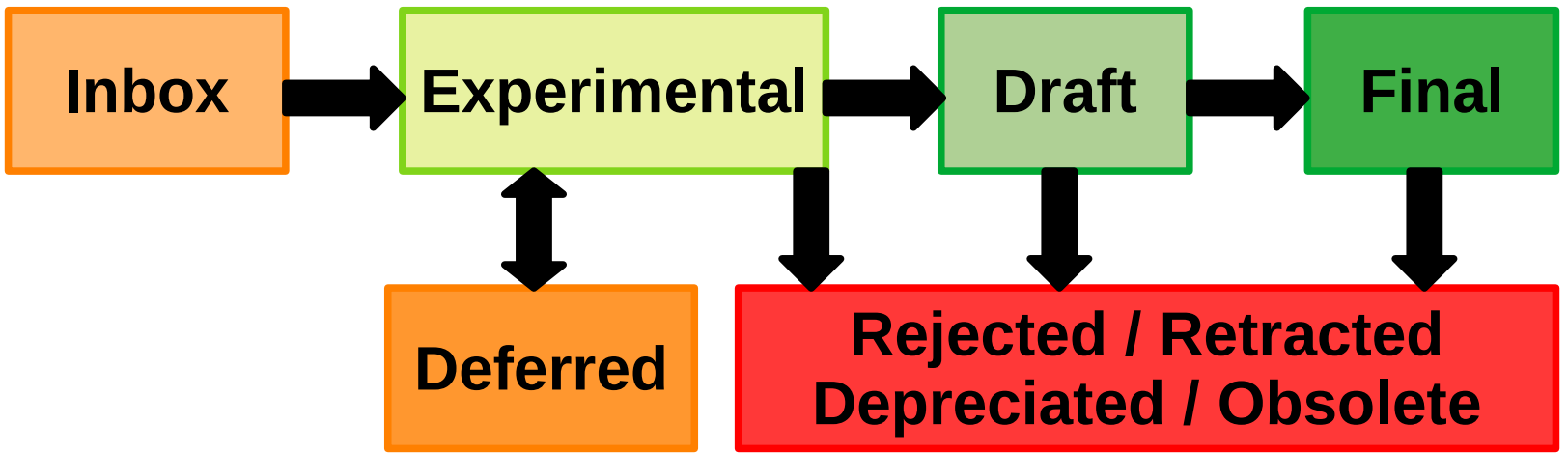

XEP
XEP
XEP
XEP
XEP
XEP
XEP
RFC-6121 XMPP Messaging & Presence
RFC-7622 XMPP Addresses
RFC-6120
XMPP Core
RFC-7590
XMPP
Over
TLS
RFC-7395
XMPP
Over
Websockets
XEP
XEP

XSF: XMPP Standards Foundation

- Elected members and elected teams
- Cooperates with IETF on core protocol
- Publishes and mandates a set of extensions
  - Free to implement by everyone

Others publish extensions too, like the IEEE on IOT

XSF standards process



Note: advancement to ‘Draft’ and ‘Final’ is slow

Special XEPs

- Historical
  - Obsolete but still in use
- Informational
  - Best practices
- Procedural
  - Procedures of the XSF
- Informational & procedural don’t have ‘Draft’ and ‘Final’ but ‘Proposed’ and ‘Active’

https://xmpp.org/extensions/

☑ Active ☐ Deferred ☐ Deprecated ☑ Draft ☑ Experimental ☑ Final ☐ Obsolete ☑ Proposed ☐ Rejected ☐ Retracted

Number	Name	Type	Status	Date
XEP-0001	XMPP Extension Protocols	Procedural	Active	2019-10-19
XEP-0002	Special Interest Groups (SIGs)	Procedural	Active	2002-01-11
XEP-0004	Data Forms	Standards Track	Final	2007-08-13
XEP-0009	Jabber-RPC	Standards Track	Final	2011-11-10
XEP-0012	Last Activity	Standards Track	Final	2008-11-26
XEP-0013	Flexible Offline Message Retrieval	Standards Track	Draft	2005-07-14
XEP-0019	Streamlining the SIGs	Procedural	Active	2002-03-20
XEP-0030	Service Discovery	Standards Track	Final	2017-10-03
XEP-0033	Extended Stanza Addressing	Standards Track	Draft	2017-01-11
XEP-0045	Multi-User Chat	Standards Track	Draft	2019-05-15
XEP-0047	In-Band Bytestreams	Standards Track	Final	2012-06-22
XEP-0048	Bookmarks	Standards Track	Draft	2007-11-07
XEP-0049	Private XML Storage	Historical	Active	2004-03-01
XEP-0050	Ad-Hoc Commands	Standards Track	Draft	2019-03-26

Example of a XEP

XEP-0410

Table of Contents

XEP-0410: MUC Self-Ping (Schrödinger's Chat)

Abstract This protocol extension for XEP-0045 Multi User Chat allows clients to check whether they are still joined to a chatroom.

Author Georg Lukas

Status Draft

NOTICE: The protocol defined herein is a **Draft Standard** of the XMPP Standards Foundation. Implementations are encouraged and the protocol is appropriate for deployment in production systems, but some changes to the protocol are possible before it becomes a Final Standard.

Type Standards Track

Version 1.1.0 (2019-09-25)

Document Lifecycle

Experimental
Proposed
Draft
Final

1. Introduction

The Multi-User Chat (XEP-0045) [1] protocol was not designed to handle s2s interruptions or message loss well. Rather often, the restart of a server or a component causes a client to believe that it is still joined to a given chatroom, while the chatroom service does not know of this occupant.

Getting lost...

- 430 XEPs
- All kind of stuff
- Some deferred XEPs are widely used
- So how to find a XEP that is
  - needed for interop
  - and future proof?

Compliance suite to the rescue!

- Yearly updated
- List of XEPs to use for:
  - Core functions
  - Web
  - Instant Messaging
  - Mobile connections
- All split up to core server, advanced server, core client and advanced client

XEP-0423: XMPP Compliance Suites 2020

2.1 Core Compliance Suite

Table 1: XMPP Core Compliance Levels ¶

Feature	Core Server	Core Client	Advanced Server	Advanced Client	Providers
Core features	✓	✓	✓	✓	RFC 6120 [16] [17]
TLS	✓	✓	✓	✓	RFC 7590 [19]
Direct TLS	×	×	✓ [20]	✓	SRV records for XMPP over TLS (XEP-0368) [21]
Feature discovery	✓	✓	✓	✓	Service Discovery (XEP-0030) [22]
Feature broadcasts	×	✓	✓	✓	Entity Capabilities (XEP-0115) [23]
Server Extensibility	✓	N/A	✓	N/A	Jabber Component Protocol (XEP-0114) [24]
Event publishing	×	×	✓ [25]	✓	Personal Eventing Protocol (XEP-0163) [26]

Software support

- Clients, servers and libraries document what XEPs they support
- Many of them have plug-ins to support extra XEPs
- Any payload that is send from client to client is always relayed by the XMPP servers, so you can easily create your own extension.

XEP-0114 Component Protocol

XEP-0114

Table of Contents

XEP-0114: Jabber Component Protocol

Abstract This specification documents the existing protocol used for communication between servers and "external" components over the Jabber network.

Author Peter Saint-Andre

Status Active

NOTICE: This Historical specification provides canonical documentation of a protocol that is in use within the Jabber/XMPP community. This document is not a standards-track specification within the XMPP Standards Foundation's standards process; however, it might be converted to standards-track in the future or might be obsoleted by a more modern protocol.

Type Historical

Version 1.6 (2012-01-25)

Document Lifecycle

Experimental
Proposed
Active

1. Introduction

The Jabber network has long included a wire protocol that enables trusted components to connect to Jabber servers. While this component protocol is minimal and will probably be superseded by a more comprehensive component protocol at some point, informational documentation of the existing protocol would be helpful for component and server developers. This specification provides such documentation.

XEP-0060 PubSub

XEP-0060

Table of Contents

XEP-0060: Publish-Subscribe

Abstract This specification defines an XMPP protocol extension for generic publish-subscribe functionality. The protocol enables XMPP entities to create nodes (topics) at a pubsub service and publish information at those nodes; an event notification (with or without payload) is then broadcasted to all entities that have subscribed to the node. Pubsub therefore adheres to the classic Observer design pattern and can serve as the foundation for a wide variety of applications, including news feeds, content syndication, rich presence, geolocation, workflow systems, network management systems, and any other application that requires event notifications.

Authors Peter Millard, Peter Saint-Andre, Ralph Meijer

Status Draft

NOTICE: The protocol defined herein is a **Draft Standard** of the XMPP Standards Foundation. Implementations are encouraged and the protocol is appropriate for deployment in production systems, but some changes to the protocol are possible before it becomes a Final Standard.

Type Standards Track

Version 1.17.0 (2019-10-06)

Document Lifecycle

Experimental
Proposed
Draft
Final

1. Introduction

XEP-0365 XMPP over HF Radio

XEP-0365

Table of Contents

XEP-0365: Server to Server communication over STANAG 5066 ARQ

Abstract This specification defines operation over XMPP over the NATO STANAG 5066 data link service for point to point links (ARQ). This enables optimized XMPP performance over HF Radio (which STANAG 5066 was designed for) and over other data links using STANAG 5066.

Author Steve Kille

Status Deferred

WARNING: This document has been automatically Deferred after 12 months of inactivity in its previous Experimental state. Implementation of the protocol described herein is not recommended for production systems. However, exploratory implementations are encouraged to resume the standards process.

Type Standards Track

Version 0.2.1 (2018-07-21)

Document Lifecycle

Experimental
Deferred
Proposed
Draft
Final

1. Introduction

This specification arose from requirements to operate over HF Radio, which has exceedingly high latency (sometimes minutes) low

XEP-0301 Real Time Text

XEP-0301

Table of Contents

https://xmpp.org/extensions/xep-0301.html#appendix-legal

XEP-0301: In-Band Real Time Text

Abstract This is a specification for real-time text transmitted in-band over an XMPP session. Real-time text is text transmitted instantly while it is being typed or created.

Authors Mark Rejhon, Gunnar Hellstrom

Status Draft

NOTICE: The protocol defined herein is a **Draft Standard** of the XMPP Standards Foundation. Implementations are encouraged and the protocol is appropriate for deployment in production systems, but some changes to the protocol are possible before it becomes a Final Standard.

Type Standards Track

Version 1.0 (2013-10-08)

Document Lifecycle

Experimental
Proposed
Draft
Final

1. Introduction

This document defines a specification for real-time text transmitted in-band over an XMPP network.

Real-time text is text transmitted instantly while it is being typed or created. The recipient can immediately read the sender's text as it is written, without waiting. It allows text to be used as conversationally as a telephone conversation, including in situations where speech is not practical (e.g., environments that must be quiet, environments too noisy to hear, restrictions on phone use, situations where speaking is a privacy or security concern, and/or when participant(s) are deaf or hard of hearing). It is also used for transmission of live

XEP-XXXX User-defined Data

XEP-XXXX

Table of Contents

XEP-XXXX: User-defined Data Transfer

Abstract This specification proposes a simple mechanism by which applications can transfer data safely, without needing additional protocol design work. It is intended to provide a protocol that is trivial to implement and can be driven with a simple API.

Author Dave Cridland

Status ProtoXEP

WARNING: This document has not yet been accepted for consideration or approved in any official manner by the XMPP Standards Foundation, and this document is not yet an XMPP Extension Protocol (XEP). If this document is accepted as a XEP by the XMPP Council, it will be published at <http://xmpp.org/extensions/> and announced on the <standards@xmpp.org> mailing list.

Type Standards Track

Version 0.0.1 (2019-12-30)

Document Lifecycle

Experimental
Proposed
Draft
Final

1. Introduction

Applications written on top of XMPP often need to exchange data that has no existing standard. Such applications are often written by

XEP-0419 Improving Security

XEP-0419

Table of Contents

signficant benefit that encryption is entirely transparent, providing excellent interoperability benefits with older implementations that may not have been upgraded.

We therefore recommend that all stanzas on the wire are fully encrypted with Double ROT-13. Given the following stanza:

Example 1. Original unencrypted stanza ¶

```
<message from='chris.davidland@crap-security.example' to='lucas.george@shiteam.example' type='chat' id='12
  <some-metadata xmlns='urn:xmpps:example:metadata'/>
  <body>Hey, Lucas!</body>
</message>
```

The following shows a correctly encrypted stanza:

Example 2. Stanza with encrypted meta-data and payload ¶

```
<message from='chris.davidland@crap-security.example' to='lucas.george@shiteam.example' type='chat' id='12
  <some-metadata xmlns='urn:xmpps:example:metadata'/>
  <body>Hey, Lucas!</body>
</message>
```

The following, however, has only encrypted the body text - this is NOT valid encryption, and an attacker can easily read the remaining

XEP-0419 Improving Security

XEP-0419

Table of Contents

signficant benefit that encryption is entirely transparent, providing excellent interoperability benefits with older implementations that may not have been upgraded.

We therefore recommend that all stanzas on the wire are fully encrypted with Double ROT-13. Given the following stanza:

Example 1. Original unencrypted stanza ¶

```
<message from='chris.davidland@crap-security.example' to='lucas.george@shiteam.example' type='chat' id='12
  <some-metadata xmlns='urn:xmpps:example:metadata'/>
  <body>Hey, Lucas!</body>
</message>
```

The following shows a correctly encrypted stanza:

Example 2. Stanza with encrypted meta-data and payload ¶

```
<message from='chris.davidland@crap-security.example' to='lucas.george@shiteam.example' type='chat' id='12
  <some-metadata xmlns='urn:xmpps:example:metadata'/>
  <body>Hey, Lucas!</body>
</message>
```

The following, however, has only encrypted the body text - this is NOT valid encryption, and an attacker can easily read the remaining

XEP-0239 Binary XMPP

XEP-0239

Table of Contents

Example 1. Traditional XMPP stanza ¶

```
<presence/>
```

That string can be represented in binary as follows:

Example 2. Binary representation ¶

```
0011110001110000011100100110010101110011011001010110111001100011011001010010111100111110
```

The bit sequence is therefore represented in Binary XMPP as follows (line breaks are provided only for the purpose of readability):

Example 3. Binary XMPP representation ¶

```
<zero/><zero/><one/><one/><one/><one/><zero/><zero/>
<zero/><one/><one/><one/><zero/><zero/><zero/><zero/>
<zero/><one/><one/><one/><zero/><zero/><one/><zero/>
<zero/><one/><one/><zero/><zero/><one/><zero/><one/>
<zero/><one/><one/><one/><zero/><zero/><one/><one/>
<zero/><one/><one/><zero/><zero/><one/><zero/><one/>
<zero/><one/><one/><zero/><one/><one/><one/><zero/>
<zero/><one/><one/><zero/><zero/><zero/><one/><one/>
<zero/><one/><one/><zero/><zero/><one/><zero/><one/>
<zero/><zero/><one/><zero/><one/><one/><one/><one/>
<zero/><zero/><one/><one/><one/><one/><one/><zero/>
```

Need to know more?

- https://xmpp.org/
- Don’t forget to look at this years compliance suite
- Feel free to ask hints, tips and assistance!
- Developers channels:
  - xmpp:jdev@muc.xmpp.org
  - mailto:jdev@jabber.org