Extensible Messaging and Presence Protocol

Logo of the XMPP Standards Foundation

Extensible Messaging and Presence Protocol (**XMPP**) is an open-standard communications protocol for message-oriented middleware based on Extensible Markup Language (XML).[1] The protocol was originally named **Jabber**,[2] and was developed by the Jabber open-source community in 1999 for, originally, near-real-time, extensible instant messaging (IM), presence information, and contact list maintenance. Designed to be extensible, the protocol today also finds application in Voice over Internet Protocol and file transfer signaling.

Unlike most instant messaging protocols, XMPP uses an open systems approach of development and application, by which anyone may implement an XMPP service and interoperate with other organizations' implementations. The software implementation and many client applications are distributed as free and open source software.

The Internet Engineering Task Force (IETF) formed an XMPP Working Group in 2002 to formalize the core protocols as an IETF instant messaging and presence technology. The XMPP WG produced four specifications (RFC 3920, RFC 3921, RFC 3922, RFC 3923) which were approved by the Internet Engineering Steering Group as Proposed Standards in 2004. The XMPP Standards Foundation (formerly the Jabber Software Foundation) is active in developing open XMPP extensions.

XMPP-based software is deployed widely across the Internet and by 2003 was used by over ten million people worldwide, according to the XMPP Standards Foundation.[3]

Google Wave's federation protocol is an extension to the XMPP protocol.[4]

History

Jeremie Miller began working on the Jabber technology in 1998 and released the first version of the `jabberd` server on January 4, 1999.[5] Although the early Jabber community focused on open-source software, mainly the jabberd server (e.g., version 1.0 in May 2000, version 1.2 in October 2000, and version 1.4 in February 2001), the primary outcome of the early community was creation of the XMPP protocol.

The early Jabber protocol as developed in 1999 and 2000 formed the basis for XMPP as published in RFC 3920 and RFC 3921 (the primary changes during formalization by the IETF's XMPP Working Group were the addition of TLS for channel encryption and SASL for authentication). XMPP has often been regarded as a competitor to SIMPLE, based on the Session Initiation Protocol (SIP) protocol, as the standard protocol for instant messaging and presence notification.[6] [7]

The first IM service based on XMPP was Jabber.org, which has operated continuously since 1999 and has offered free accounts to users of XMPP.[8] From 1999 until February 2006 the service used jabberd as its server software, at which time it migrated to ejabberd (both of which are free software application servers). In January 2010, the service plans to migrate to proprietary M-Link software produced by Isode Ltd.[9]

In August 2005, Google introduced Google Talk, a combination VoIP and IM system which uses XMPP for its instant messaging function and as a base for its voice and file transfer signalling protocol (although the initial launch did not include server-to-server communications, that feature was enabled on January 17, 2006).[10]

In September 2008, Cisco Systems acquired Jabber, Inc. the creators of the commercial product Jabber XCP.[11]

In February 2010, the social-networking site Facebook opened up its chat feature to third-party applications via XMPP.[12] The Facebook developers' site notes that Facebook Chat does not actually run an XMPP server internally, but merely presents an XMPP interface to clients; consequently, some server-side features like roster editing cannot be done via XMPP.[13]

In addition to Google Talk, many other public IM services use XMPP, including LiveJournal's "LJ Talk"[14] and Nokia's Ovi.[15] Furthermore, several enterprise IM software products that do not natively use XMPP nevertheless include gateways to XMPP, including IBM Lotus Sametime[16] [17] and Microsoft Office Communications Server (OCS).[18]

Strengths

Decentralization

The architecture of the XMPP network is similar to email; anyone can run their own XMPP server and there is no central master server.

Open standards

The Internet Engineering Task Force has formalized XMPP as an approved instant messaging and presence technology under the name of XMPP, and the XMPP specifications have been published as RFC 3920 and RFC 3921. No royalties are required to implement support of these specifications and their development is not tied to a single vendor.

History

XMPP technologies have been in use since 1998. Multiple implementations of the XMPP standards exist for clients, servers, components, and code libraries, with the backing of large companies such as Sun Microsystems and Google.

Security

XMPP servers may be isolated from the public XMPP network (e.g., on a company intranet), and robust security (via SASL and TLS) has been built into the core XMPP specifications. To encourage use of channel encryption, the XMPP Standards Foundation [19] currently runs an intermediate certification authority at StartSSL [20] (formerly at xmpp.net [21]) offering free digital certificates to XMPP server administrators under the auspices of the StartCom Certification Authority (which is the root CA for the intermediate CA).

Flexibility

Custom functionality can be built on top of XMPP; to maintain interoperability, common extensions are managed by the XMPP Software Foundation. XMPP applications beyond IM include network management, content syndication, collaboration tools, file sharing, gaming, and remote systems monitoring.

Weaknesses

In-band binary data transfer is inefficient

Because XMPP is encoded as a single long XML document, binary data must be first base64 encoded before it can be transmitted in-band. Therefore any significant amount of binary data (e.g., file transfers) is best transmitted out-of-band, using in-band messages to coordinate. The best example of this is the Jingle XMPP Extension Protocol, XEP-0166 [22].

Decentralization and addressing

The XMPP network uses a client–server architecture (clients do not talk directly to one another). However, it is decentralized—by design there is no central authoritative server, as there is with services such as AOL Instant Messenger or Windows Live Messenger. Some confusion often arises on this point as there is a public XMPP server being run at `jabber.org`, to which a large number of users subscribe. However, anyone may run their own XMPP server on their own domain. The standard TCP port for XMPP is 5222.

Every user on the network has a unique *Jabber ID* (usually abbreviated as *JID*). To avoid requiring a central server to maintain a list of IDs, the JID is structured like an e-mail address with a username and a domain name (or IP address[23]) for the server where that user resides, separated by an at sign (`@`), such as `username@example.com`.

Since a user may wish to log in from multiple locations, they may specify a **resource**. A resource identifies a particular client belonging to the user (for example home, work, or mobile). This may be included in the JID by appending a slash followed by the name of the resource. For example, the full JID of a user's mobile account would be `username@example.com/mobile`.

Each resource may have specified a numerical value called **priority**. Messages simply sent to `username@example.com` will go to the client with highest priority, but those sent to `username@example.com/mobile` will go *only* to the mobile client.

JIDs without a username part are also valid, and may be used for system messages and control of special features on the server. A resource remains optional for these JIDs as well.

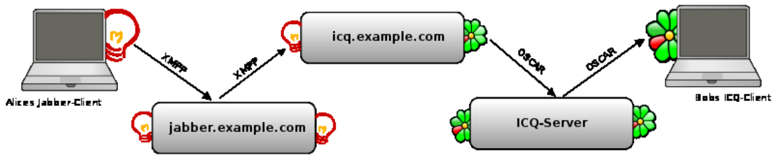
Message delivery scenario

Suppose *juliet@capulet.com* wants to chat with *romeo@montague.net*. Juliet and Romeo each respectively have accounts on the capulet.com and montague.net servers. When Juliet types in and sends her message, a sequence of events is set in motion:

1. Juliet's client sends her message to the capulet.com server
   - If montague.net is blocked on capulet.com, the message is dropped.
2. The capulet.com server opens a connection to the montague.net server.
   - If capulet.com is blocked on montague.net, the message is dropped.
3. Montague.net checks to see if Romeo is currently connected. If not, the message is stored for later delivery.
4. If Romeo is online, the montague.net server delivers the message to Romeo.

Juliet → capulet.com → montague.net → Romeo

Connecting to other protocols



Alice sends a message through the XMPP net to the ICQ transport. The message is next routed to Bob via the ICQ network.

Another useful feature of the XMPP system is that of *transports*, also known as *gateways*, which allow users to access networks using other protocols. This can be other instant messaging protocols, but also protocols such as SMS or e-mail. Unlike multi-protocol clients, XMPP provides this access at the server level by communicating via special gateway services running on a remote computer. Any user can "register" with one of these gateways by providing the information needed to log on to that network, and can then communicate with users of that network as though they were XMPP users. This means that any client which fully supports XMPP can be used to access any network for which a gateway exists, without the need for any extra code in the client and without the need for the client to have direct access to the Internet. This may violate terms of service on the protocol used; however, such terms of service are not legally enforceable in several countries. This also requires sending your IM username and password to the third-party site that operates the transport, which may raise privacy concerns.

XMPP via HTTP transport

Another aspect of XMPP is the HTTP binding for users behind restricted firewalls. In the original specification, XMPP could use HTTP in two ways: *polling*[24] and *binding*.[25] The polling method, now deprecated, essentially implies messages stored on a server-side database are being fetched (and posted) regularly by an XMPP client by way of HTTP 'GET' and 'POST' requests. With HTTP binding, the client uses longer-lived HTTP connections to receive messages as soon as they are sent. This push model of notification is more efficient than polling, where many of the polls return no new data.

Because the client uses HTTP, most firewalls allow clients to fetch and post messages without any hindrances. Thus, in scenarios where the TCP port used by XMPP is blocked, a server can listen on the normal HTTP port and the traffic should pass without problems. There also are various websites which allow people to sign in to XMPP via their browser. Furthermore, there are open public servers that listen on standard http (port 80) and https (port 443) ports and hence allow connections from behind most firewalls.

Implementations

XMPP is implemented by a large number of XMPP clients [26], servers [27], and code libraries [28].

Development

The IETF XMPP working group has produced a number of RFC protocol documents: RFC 3920, RFC 3921, RFC 3922, RFC 3923, RFC 4622, RFC 4854, RFC 4979

- RFC 3920, *Extensible Messaging and Presence Protocol (XMPP): Core* which describes client–server messaging using two open-ended XML streams. XML streams consist of <presence/>, <message/> and <iq/> (info/query). A connection is authenticated with Simple Authentication and Security Layer (SASL) and encrypted with Transport Layer Security (TLS).
- RFC 3921, *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence* describes instant messaging (IM), the most common application of XMPP.
- RFC 3922, *Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM)* relates XMPP and the Common Presence and Instant Messaging (CPIM) specifications.
- RFC 3923, *End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)* describes end to end encryption of XMPP messages using S/MIME. Conflicting this proposal, many clients currently use GPG for encrypting messages.[29]

The XMPP Standards Foundation (XSF) develops and publishes extensions to XMPP through a standards process centered on *XMPP Extension Protocols* (XEPs, previously known as Jabber Enhancement Proposals - JEPs). The following extensions are in especially wide use:

- Data Forms[30]
- Service Discovery[31]
- Multi-User Chat[32]
- XHTML-IM[33]
- File Transfer[34]
- Entity Capabilities[35]
- HTTP Binding[25]

- Personal Eventing Protocol[36]

XMPP is currently being extended to handle signaling / negotiation for Voice over Internet Protocol (VoIP) and other media sessions. This signaling protocol is called Jingle. Jingle is designed to be consistent with the Google Talk service and bridgeable with the Session Initiation Protocol.

See also

- Comparison of instant messaging clients
- Comparison of instant messaging protocols
- Comparison of XMPP server software
- Secure communication
- SIMPLE

References

[1] Johansson, Leif (18 April 2005). "XMPP as MOM" (http://www.gnomis.org/presentasjoner/oslo2005/xmpp.pdf). *Greater NOrdic MIddleware Symposium (GNOMIS)*. Oslo: University of Stockholm.

[2] Jabber Inc. (http://www.cisco.com/web/about/ac49/ac0/ac1/ac258/JabberInc.html)

[3] XMPP Standards Foundation (22 September 2003). "Jabber Instant Messaging User Base Surpasses ICQ" (http://xmpp.org/xsf/press/2003-09-22.shtml). Press release. . Retrieved 30 November 2007.

[4] "Google Wave Federation Protocol" (http://www.waveprotocol.org/draft-protocol-specs/draft-protocol-spec#intro-overview). Google. .

[5] Open Real Time Messaging System (http://tech.slashdot.org/article.pl?sid=99/01/04/1621211)

[6] "XMPP rises to face SIMPLE standard", Infoworld magazine, April 17, 2003 XMPP rises to face SIMPLE standard (http://www.infoworld.com/d/developer-world/xmpp-rises-face-simple-standard-018)

[7] "XMPP vs SIMPLE: The race for messaging standards", Infoworld magazine, May 23, 2003 Infoworld.com (http://www.infoworld.com/t/platforms/xmpp-vs-simple-race-messaging-standards-295)

[8] Chatting Up the Chef (http://www.linuxjournal.com/article/6489) Linux Journal March 1, 2003 by Marcel Gagné

[9] "Jabber.org - XMPP Server Migration" (http://www.jabber.org/2009/08/xmpp-server-migration/). 12 August 2009. . Retrieved 14 December 2009.

[10] Burd, Gary (17 January 2006). "XMPP Federation" (http://googletalk.blogspot.com/2006/01/xmpp-federation.html). . Retrieved 30 November 2007.

[11] "Cisco Announces Definitive Agreement to Acquire Jabber" (http://newsroom.cisco.com/dlls/2008/corp_091908.html). . Retrieved 2 January 2010.

[12] "Facebook Chat Now Available Everywhere" (http://blog.facebook.com/blog.php?post=297991732130). . Retrieved 11 February 2010.

[13] "Integrating with Facebook Chat" (http://wiki.developers.facebook.com/index.php/Integrating_with_Facebook_Chat). . Retrieved 21 February 2010.

[14] Question FAQ #270 (http://www.livejournal.com/support/faqbrowse.bml?faqid=270)

[15] Ovi Contacts (http://betalabs.nokia.com/betas/view/contacts-ovi)

[16] "Lotus Sametime 7.5 Interoperates with AIM, Google Talk", eWeek, December 6, 2006 Eweek.com (http://www.eweek.com/c/a/Messaging-and-Collaboration/Lotus-Sametime-75-Interoperates-with-AIM-Google-Talk/)

[17] "Lotus ships gateway to integrate IM with AOL, Yahoo, Google", Network World, December 6, 2006 Networkworld.com (http://www.networkworld.com/news/2006/120606-sametime-links-up-with-aim.html)

[18] "Unified Communications: Uniting Communication Across Different Networks", Microsoft Press Release, October 1, 2009 Microsoft.com (https://www.microsoft.com/Presspass/Features/2009/oct09/10-01UCInterop.mspx)

[19] http://xmpp.org/

[20] http://startssl.com

[21] http://xmpp.net

[22] http://xmpp.org/extensions/xep-0166.html

[23] RFC-Standard 3920 for XMPP (http://tools.ietf.org/html/rfc3920)

[24] XEP-0025: Jabber HTTP Polling (http://xmpp.org/extensions/xep-0025.html)

[25] XEP-0124: Bidirectional-streams Over Synchronous HTTP (BOSH) (http://xmpp.org/extensions/xep-0124.html)

[26] http://xmpp.org/xmpp-software/clients/

[27] http://xmpp.org/xmpp-software/servers/

[28] http://xmpp.org/xmpp-software/libraries/

[29] XEP-0027: Current Jabber OpenPGP Usage (http://xmpp.org/extensions/xep-0027.html)

[30] XEP-0004: Data Forms (http://xmpp.org/extensions/xep-0004.html)

[31] XEP-0030: Service Discovery (http://xmpp.org/extensions/xep-0030.html)

[32] XEP-0045: Multi-User Chat (http://xmpp.org/extensions/xep-0045.html)

[33] XEP-0071: XHTML-IM (http://xmpp.org/extensions/xep-0071.html)

[34] XEP-0096: File Transfer (http://xmpp.org/extensions/xep-0096.html)

[35] XEP-0115: Entity Capabilities (http://xmpp.org/extensions/xep-0115.html)

[36] XEP-0163: Personal Eventing Protocol (http://xmpp.org/extensions/xep-0163.html)

External links

- XMPP Standards Foundation (http://xmpp.org/)
- Jabber User Guide (http://archive.jabber.org/userguide/) - End user introduction to XMPP (archive)
- IETF Publishes XMPP RFCs (http://xmpp.org/xsf/press/2004-10-04.shtml)
- XMPP Case Studies (http://wiki.xmpp.org/web/XMPP_Case_Studies)
- Podcast: FLOSS 49: XMPP (http://twit.tv/floss49/) Peter Saint-Andre interviewed by Randal Schwartz and Leo Laporte
- Google Map of known XMPP servers (http://www.weavver.com/Company/Services/Chat/Server Map)

Article Sources and Contributors

Extensible Messaging and Presence Protocol *Source*: http://en.wikipedia.org/w/index.php?oldid=408831366 *Contributors*: 0x6adb015, Accountholder, Alanthehat, AlistairMcMillan, Alvestrand, Armando82, Arto B, Ashley Y, AxelBoldt, Beefyt, Bender235, Beno1000, Bjelli, Bryan Derksen, Callidior, Camw, Canaima, Chowbok, Closeapple, Cutter, Cybercobra, DGMurdockIII, Dabizi, Darco, Dasch, Diza, DoubleVibro, Dougher, Drangon, Droob, DustWolf, Ebukadneza, EdgeOfEpsilon, Ejabberd, Enjoi4586, Erik Zachte, Esrever, Evice, Feedmecereal, FlashSheridan, Fluffernutter, Folajimi, Foofy, Gabriel Acquistapace, Galoubet, Genericprodigy, Guyjohnston, Haakon, Hashar, Hm2k, Holizz, Iam, Ida Shaw, Ildmur, IlyaHaykinson, Imorgillo, Intgr, Isaac Dupree, Islam-gv, Iviney, J.delanoy, JTN, Jabber11, Jaho, Jarble, Jayjg, Jonabbey, Julesd, Kadin2048, Kathleen.wright5, Kbdank71, Kbolino, Kbrose, Keastes, Kinema, Kne1p, Lawrence Cohen, Lektu, Lelguea, Libertyernie2, Lzur, MBisanz, MMuzammils, Maggu, ManosFate, Markpeak, Maury Markowitz, MovGP0, Moxfyre, Mpcabd, MrBeauGiles, Mremond, Mrzaius, Mwtoews, NJA, Nad, NaturalBornKiller, Nealmcb, Neilc, Neustradamus, Nono64, Nopetro, Nubiatech, Oxymoron83, Pgan002, Phatom87, Phoenix720, Pne, Psz, Qu1j0t3, R'n'B, Ray Dassen, RedWolf, Rich Farmbrough, RickBeton, Riffic, Ringbang, Robby.is.on, S-1-5-7, SF007, SamJohnston, Sketchee, Slady, Slipstream, Spider-spax, Stephan Leeds, Stevietheman, Stpeter, Suruena, SusanLesch, Svick, SymlynX, Tero, TheDude1991, Thumperward, Tikiwont, Tobias Conradi, Tothwolf, Toussaint, TreveX, Txomin, Vi2, WakiMiko, Weschapinita, Wilinckx, Wolfcm, Wsheward, XZise, Yaronf, Yekrats, Youssef51, Zearin, Zigamorph, Zvar, 200 anonymous edits

Image Sources, Licenses and Contributors

File:XMPP Logo.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:XMPP_Logo.svg *License*: unknown *Contributors*: XMPP Standards Foundation

Image:Wie ein Jabber-Transport funktioniert.svg *Source*: http://en.wikipedia.org/w/index.php?title=File:Wie_ein_Jabber-Transport_funktioniert.svg *License*: GNU Free Documentation License *Contributors*: Peritus at de.wikipedia

License