

XMPP

Federico Scavone
federicoscavone@gmail.com

Universidad Católica Nuestra Señora de la Asunción. Facultad de Ciencias y
Tecnología

Abstract. Éste documento explica los que es XMPP, su historia, arquitectura, principales características, ventajas ;y los servidores, clientes y librerías que implementan éste protocolo. El objetivo no es entrar tanto en detalles técnicos sino más bien analizar a grandes rasgos que es y como funciona XMPP.

Key words: Xmpp, Protocolo, Mensajería instantánea, Miller

1 Introducción

En los últimos años el acceso a la información en tiempo real se ha vuelto de vital importancia, tanto en el ámbito laboral como en el personal. Las personas necesitan que que la distribución y/o el intercambio de la información sea inmediata y segura. Una de las herramientas más utilizadas para llevar a cabo ésta tarea es la mensajería instantánea, por ser la que mas ventajas ofrece. Para realizar el intercambio de información a través de la mensajerías instantánea, es necesario establecer un conjunto de reglas a seguir para que los equipos involucrados “hablen” el mismo idioma. A ésto se le denomina protocolo. A continuación veremos no de los protoclos más utilizados, el XMPP.

2 Historia

Todo comenzó en el año 1998, cuando a Jeremie Miller, cansado de tener que usar 4 clientes distintos de mensajería instanánea para poder usar los servicios de IM¹ de la época, tuvo la brillante idea de diseñar un protocolo abierto, al cual le puso el nombre de Jabber.

En Febrero de 2000, fueron publicados los RFCs² propuestos por Miller al IMPP WG³ del IETF⁴.

El RFC 2778 define un modelo abstracto para un sistema de mensajería instantánea y presencia. El RFC 2779 define los requisitos para la implementación de dichos sistemas.

¹ Instant Messenger

² Request for comment

³ Grupo de Trabajo en Protocolos de Mensajería Instantánea y Presencia

⁴ Internet Engineering Task Force

Como es común en este ámbito, existían diferencias entre los miembros del IMMP WG, por lo que nunca se lanzó ningún protocolo real. Pero como también suele pasar en el ambiente Open-source, la comunidad existente detrás del proyecto Jabber no se dió por vencida y siguió trabajando. En octubre del 2000 lanzaron la versión 1.2 del servidor abierto Jabberd (Jabber Daemon).

La comunidad seguía creciendo pero no estaba bien organizada y era muy difícil coordinar los trabajos de esa manera, por lo que en agosto de 2001 se constituye la Jabber Software Foundation (JSF), cuya misión era coordinar el desarrollo y documentar los protocolos XML utilizados por Jabber.

La JSF decide cambiar el nombre de Jabber a XMPP (eXtensible Messaging and Presence Protocol). Recién en febrero de 2002, la JSF vuelve a enviar a la IETF un borrador actualizado del protocolo Jabber. Todo iba sobre ruedas, gracias a la organización conseguida desde la fundación de la JSF. En junio del mismo año enviaron 3 nuevos borradores con excelentes resultados.

En octubre de 2002, la IESG⁵ designa un grupo de trabajo dedicado exclusivamente al XMPP. El XMPP WG se enfocó en realizar mejoras del protocolo, especialmente en la seguridad y la arquitectura. Mientras tanto, la JSF estaba centrada en el desarrollo de las extensiones de Jabber, que eran compatibles con XMPP.

Dos años más tarde, se publican los RFCs donde se define el protocolo XMPP completo.

- El RFC 3920 especifica las características principales del protocolo XMPP
- El RFC 3921 describe las aplicaciones y extensiones para proveer servicios de mensajería instantánea y presencia previstos en XMPP
- El RFC 3922 establece como realizar un mapeo entre XMPP y CPIM⁶
- RFC 3923 define un método de firma y encriptamiento de objetos punto a punto usando XMPP

3 Código abierto y Estándar abierto

Si bien XMPP fue desarrollado en sus inicios por la comunidad de código abierto Jabber, el protocolo en sí no es un proyecto de código abierto, más bien es un estándar abierto, lo que da como resultado una tecnología abierta que no está atada a ningún proyecto de software ni a ninguna compañía. Gracias a esto existen una infinidad de implementaciones de XMPP código abierto que pueden ser descargadas gratis por cualquier usuario, administrador de sistema o desarrollador. La mayoría de éstas implementaciones están enfocadas a la mensajería instantánea. Existen una infinidad de clientes código abierto para cualquier sistema operativo y/o dispositivo. También existen servidores código abierto que pueden implementarse en las empresas, colegios, universidades y proveedores de servicios. Existen librerías código abierto para casi todos los lenguajes de

⁵ Internet Engineering Steering Group

⁶ Common Profile for Instant Messaging

programación que pueden ser utilizadas para desarrollar clientes personalizados, servidores, aplicaciones en tiempo real, o bien extensiones para aplicaciones existentes.

4 Cómo funciona XMPP

4.1 Arquitectura

El protocolo XMPP usa una arquitectura cliente-servidor descentralizada, lo que permite a los desarrolladores de programas clientes enfocarse en la experiencia de los usuarios, y a los desarrolladores de servidores enfocarse en el rendimiento y escalabilidad. Es mucho más simple para una organización (recordemos que cualquiera puede montar su propio servidor y unirse a la red) mantener este tipo de redes, que las redes peer-to-peer. Además es bastante robusta, ya que no posee un punto de falla único.¹

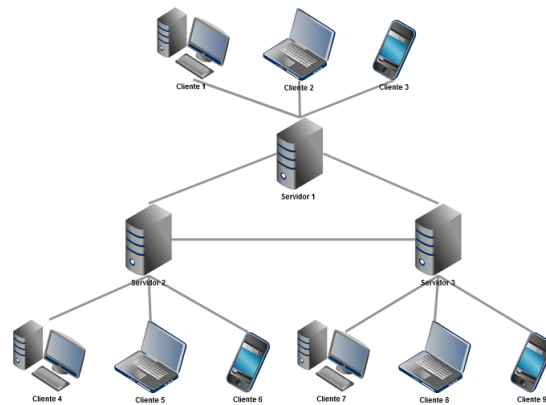


Fig. 1. Arquitectura cliente-servidor descentralizada

Cuando visitamos una página web, el navegador se conecta a un servidor web para obtener el contenido de la página, pero no siempre todo el contenido de la misma se encuentra alojado en el mismo servidor, cuando esto sucede, esos servidores no se conectan entre ellos, sino que el mismo navegador establece conexiones diferentes con cada servidor. Esto sucede porque la arquitectura Web no permite conexiones interdominios.²

Cuando enviamos un email a alguno de nuestros contactos ubicados en un dominio diferente al nuestro, el cliente de correo se conecta a nuestro servidor de correo, y éste se encarga de enrutar el mensaje hasta nuestro destino, pudiendo pasar por otros servidores de correo intermedios antes de llegar al destino final. Esto significa que la red de correos realiza múltiples saltos entre servidores.³

Como la red de correos, pero a diferencia de la Web, XMPP utiliza conexiones interdominios, pero de una forma diferente. Cuando enviamos un mensaje XMPP

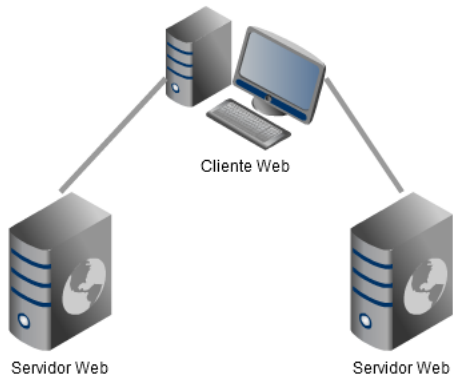


Fig. 2. Arquitectura Web

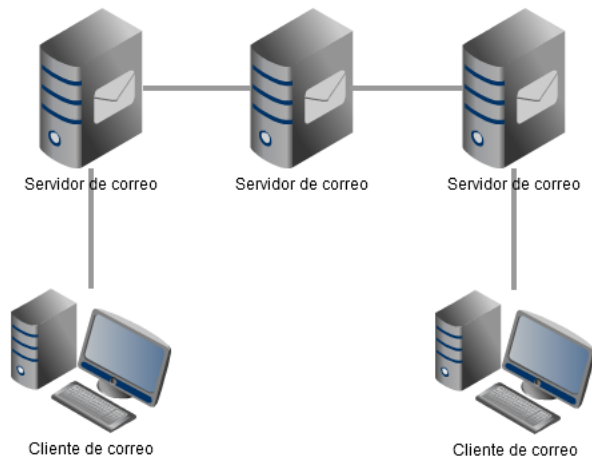


Fig. 3. Arquitectura Email

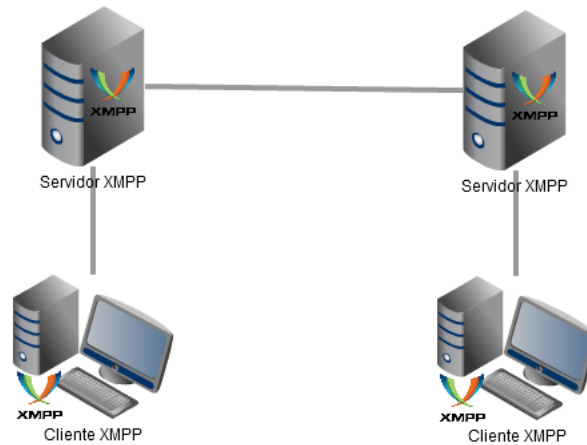


Fig. 4. Arquitectura XMPP

a algún contacto en otro dominio, el cliente XMPP se conecta a nuestro servidor XMPP, y éste se conecta directamente al servidor XMPP de nuestro contacto, sin realizar múltiples saltos. Ésta implementación de la arquitectura es mucho más segura porque previene la suplantación de identidad, y hasta cierta manera, el spam.⁴

Característica	Web	Email	XMPP
Conexiones interdominios	No	Si	Si
Saltos múltiples	N/A	Si	No

Table 1. Arquitecturas cliente-servidor

4.2 Direcciones

Cada entidad XMPP necesita tener su propia dirección, llamada JabberID (JID). La JID tiene el mismo formato que las direcciones de correo, usuario@dominio. La ventaja de ésto es que es mucho más facil recordar éste tipo de direcciones que si se utilizara directamente el protocolo IP. Pero éste formato depende completamente de la infraestructura del DNS.

4.3 Recursos

XMPP nos permite estar conectados al servidor desde varios lugares al mismo tiempo, para esto usa algo llamado Recurso. El Recurso no es mas que una cadena de referencia que identifica al cliente que está utilizando el usuario. Esto

se añade al JID precedido de un caracter quedando el identificador completo de la siguiente manera: usuario@dominio/recurso.

4.4 XML

XMPP es en esencia una tecnología para transmitir xml. Cuando iniciamos sesión en un servidor XMPP, se establece una conexión TCP persistente y luego se negocia los parámetros para que el servidor pueda establecer otra conexión de retorno. Una vez establecida la conexión podemos intercambiar mensajes asíncronamente con el servidor.

Existen 3 tipos de etiquetas XML para transmitir los diferentes tipos de mensaje:

- Message: Sirve para enviar mensajes entre los usuarios.
- Presence: Para informar la presencia de los usuario. Esto incluye cuando se conectan, desconectan, cambian de estado o de nombre.
- Iq: Se utiliza para la transmisión de comandos internos.a nueva conexión, informar de errores y cosas que no tienen nada que ver con los usuarios.

```
Cliente: <stream:stream>
Cliente: <presence/>
Cliente: <iq type="get">
Cliente: <query xmlns="jabber:iq:roster"/>
Cliente: </iq>

Servidor: <iq type="result">
Servidor: <query xmlns="jabber:iq:roster">
Servidor: <item jid="fscavone@uca.edu.py"/>
Servidor: <item jid="jeuzzarru@jeuzzarru.com"/>
Servidor: </query>
Servidor: </iq>

Cliente: <message from="fscavone@uca.edu.py"
Cliente: to="jeuzzarru@jeuzzarru.com">
Cliente: <body>Esto es un mensaje !</body>
Cliente: </message>
.
.
.
Cliente: <presence type="unavailable"/>
Cliente: </stream:stream>
```

Table 2. Ejemplo XMPP

4.5 Pasarelas

Una característica muy importante de XMPP es la posibilidad de interactuar con otras redes de mensajería como Windows Live Messenger, ICQ, Yahoo! Messenger, SMS, Email u otros. Lo que hace posible realizar esto son las pasarelas. La pasarela es un servicio transparente para el usuario que funciona como mediador entre los distintos protocolos. Ésto permite a cualquier cliente XMPP comunicarse con cualquier red para la cual exista una pasarela, sin necesidad de modificar el cliente.

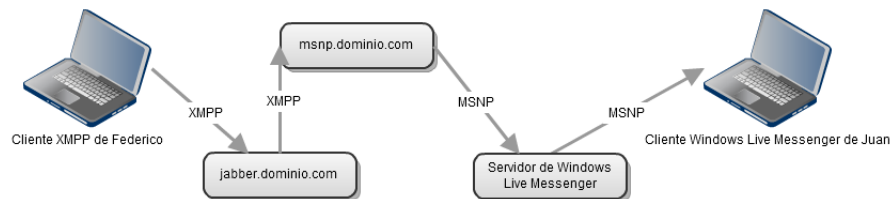


Fig. 5. Pasarela XMPP

4.6 Extensiones

Otra característica importante de XMPP es la extensibilidad. Si bien el núcleo del protocolo ofrece todas las funcionalidades básicas de la mensajería instantánea y presencia, el hecho de que sea extensible nos brinda la posibilidad de incluir prácticamente cualquier información estructurada y adecuar el uso a nuestro antojo. Cuando se utiliza XMPP para enviar a través del mismo esos datos estructurados, se les da el nombre de Extensiones XMPP. Al día de hoy se han desarrollado una buena cantidad de éstas extensiones, pero a continuación veremos las más importantes de esa lista que se han vuelto un estándar.

- XEP-0004 - Data Forms: Procesamiento de datos de formularios, representación de los campos
- XEP-0009 - Jabber-RPC: Intercambio de datos usando XML-RPC usando XMPP en vez de HTTP.
- XEP-0012 - Last Activity: Información sobre la última actividad registrada de una entidad XMPP.
- XEP-0030 - Service Discovery: Información sobre las entidades XMPP; identidad y capacidades, protocolos y características soportadas.
- XEP-0077 - In-Band Registration: Registro a servicios de mensajería basados en XMPP, como salas de chat o pasarelas XMPP.
- XEP-0085 - Chat State Notifications: Información sobre el estado de un usuario en una sesión de chat (escribiendo, inactivo, ausente, etc).

- XEP-0138 - Stream Compression: Negociación entre entidades para la compresión de los datos XML.
- XEP-0174 - Serverless Messaging: Negociación entre entidades para establecer una conexión directa en los mismos, sin servidor como intermediario.
- XEP-0199 - XMPP Ping: Envío de pings a través de mensajes XML.
- XEP-0202 - Entity Time: Comunicación de la hora local, así como el horario UTC de la entidad.
- XEP-0203 - Delayed Delivery: Entrega de mensajes retrasados, como cuando un mensaje se guarda en el servidor si el destinatario no estaba en línea, o el historial de una sala de chat.

5 Ventajas

- Abierto: Es un protocolo abierto y de fácil comprensión, motivo por el cual existen una buena cantidad de implementaciones de clientes, servidores y librerías para los desarrolladores.
- Estándar: Al ser un estándar, toda implementación de la misma es compatible con todas las demás.
- Experiencia: Fue lanzada en 1998, hace más de una década, lo cual es un periodo bastante prolongado en el campo de la tecnología.
- Descentralizado: Su arquitectura cliente-servidor descentralizada permite a las compañías montar su propio servidor, y así tener total control de sus comunicaciones.
- Seguro: Además de soportar seguridad en la capa de transporte, puede ser instalada en una red privada y utilizar otros protocolos de seguridad como SASL.
- Extensible: Además de la larga lista de extensiones existentes actualmente, existe la posibilidad de crear nuestras propias extensiones.
- Flexible: Si bien las primeras aplicaciones eran todas orientadas a la mensajería instantánea, hoy en día sus aplicaciones se han extendido al campo de compartimiento de archivos, monitoreo de sistemas remotos, servicios web, computación en la nube, juegos, etc. vicios web, computación en la nube, etc.

6 Desventajas

Sus dos principales desventajas son:

- La sobrecarga y redundancia de datos de presencia: Alrededor del 70% del tráfico entre servidores corresponde a datos de presencia y alrededor del 60% de los mismos son datos redundantes.
- La imposibilidad de entregar datos binarios sin modificar: Al ser codificado en un único documento xml, se hace imposible la transmisión de datos binarios en su estado original. Si surge la necesidad de transmitir datos binarios, es necesario codificar los datos de antemano en base64.

Protocolo	Licencia	Asíncrono	Seguridad en la Capa de transporte (TLS/Otro)	Audio/VoIP	Video/Webcam
Microsoft Messenger service (NetEUI-SMD- Net Send)	Propietario	Si	No	No	No
MSNP (Windows Live Messenger)	Propietario	Si	No	Si	Si
IRC	Abierto	Si	Si	No	No
OSCAR (AIM, ICQ)	Propietario	Si	Si	Si	Si
RVP (Windows Messenger)	Propietario	No	No	No	No
SIP/SIMPLE	Abierto	Si	Si	Si	Si
Skype	Propietario	No	Si	Si	Si
TOC (AOL)	Propietario	Si	No	No	No
TOC2 (AOL)	Propietario	Si	No	No	No
XMPP	Abierto	Si	Si	Si	Si
YMSG (Yahoo! Messenger)	Propietario	Si	No	No	No

Table 3. Comparación entre protocolos de mensajería instantánea

7 Servidores, Clientes, Librerías

7.1 XMPP vs. SIMPLE

SIMPLE⁷ es un protocolo de mensajería instantánea y presencia basado en SIP⁸ y manejado por la IETF. Al igual que XMPP es un estándar abierto, está basado en XML y utiliza una arquitectura descentralizada con clientes, servidores y puertas de enlace.

Ventajas de SIMPLE

- La mayoría de las aplicaciones VoIP utilizan SIP como protocolo de sesión, esto facilita la implementación de SIMPLE sin necesidad de desarrollar otro protocolo.
- Como está construido por encima de SIP, puede aprovechar todas las funcionalidades de éste, incluyendo autorización, autenticación y compresión.

Ventajas de XMPP

- XMPP es más liviano, son pocos los RFCs que se necesitan implementar para un sistema funcional. La lista de de RFCs a implementar en un sistema SIMPLE es mucho más larga, lo que resulta muy costoso a la hora de implementar SIMPLE en dispositivos móviles. Además, esto hace que los sistemas basados en XMPP escalan mucho mejor.

⁷ SIP Instant Messaging and Presence Leveraging Extensions

⁸ Session Initiation Protocol

- La penetración de XMPP en el mercado es mucho mayor. Gigantes como Google y Facebook utilizan ésta tecnología en su servicio de mensajería instantánea.
- Se pueden agregar funcionalidades adicionales a través de las extensiones.

Nombre	Licencia	Plataforma
Apache Vysper	Windows / Linux	Apache License Version 2.0
Citidel	Linux	GPL3
CommuniGate Pro	Linux / Mac OS X / Windows	Commercial
djabberd	Linux	GPL3
ejabberd	Linux / Mac OS X / Solaris / Windows	GPL2
IceWarp	Linux / Windows	Commercial
iChat Server	Mac OS X	Commercial
in.jabberd	Linux	GPL2
Isode M-Link	Linux / Solaris / Windows	Commercial
jabberd 1.x	Linux	GPL2
jabberd 2.x	Linux / Solaris / Windows	GPL2
Jabber XCP	Linux / Solaris / Windows	Commercial
Jerry Messenger	Linux / Windows	Commercial
Kwickserver	Windows	GPL
Openfire	Linux / Mac OS X / Solaris / Windows	GPL
Open IM	Linux	BSD
Prosody	Linux / Mac OS X / Windows	MIT/X11
psyced	Linux / Mac OS X / Windows	GPL2
SoapBox Server	Windows	Commercial
Oracle Communications Instant Messaging Server	Linux / Solaris / Windows	Commercial
Tigase	Linux / Solaris / Mac OS X / Windows	GPL3

Table 4. Lista de servidores populares

7.2 Ejabberd vs Openfire

Ejabberd implementa las siguientes funcionalidades:

- Panel de administración web y herramientas de línea de comandos que aseguran un configuración sencilla.
- Soporte para encriptar conexiones para asegurar una comunicación segura.
- Conferencias.
- Cluster con múltiples servidores
- Multiplataforma: Windows, Linux, MacOSX and Solaris.
- Recolección de estadísticas para reportes y análisis del sistema.
- Soporta Virtual hosting.
- Soporte para almacenamiento externo.
- Soporte para IPv6.

Openfire implementa las siguientes funcionalidades:

- Panel de administración web
- SSL/TLS
- Conferencias
- Interacción con MSN, Google Talk, Yahoo messenger, AIM, ICQ, Jingle
- Estadísticas del Servidor, mensajes, paquetes, etc.
- Cluster con múltiples servidores
- Transferencia de Archivos
- Compresión de datos
- Autenticación vía Certificados, Kerberos, LDAP, PAM y Radius
- Almacenamiento en Active Directory, LDAP, MS SQL, MySQL, Oracle y PostgreSQL
- SASL: ANONYMOUS, DIGEST-MD5 y Plain

Ventajas de Ejabberd

- Menor consumo de CPU.
- Es más estable.
- Más simple de configurar.
- Más escalable.

Esto lo convierte en el mejor servidor para sistemas con gran cantidad de usuarios.

Ventajas de Openfire

- Interacción con otros servicios como MSN, Google Talk, Yahoo messenger, AIM, ICQ, Jingle que no son soportados por ejabberd.

7.3 Google Talk

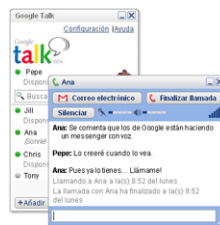


Fig. 6. Google Talk

Es un cliente bastante sencillo y popular. Son muy pocas las funcionalidades que implementa en su versión estándar:

Nombre	Plataforma
Agile Messenger	Móvil
Candy	Web
Chatopus	Móvil
Citron	Windows
Clartext EIM	Linux / Mac OS X / Windows
Coccinella	Linux / Mac OS X / Windows
Crosstalk Mobile	iOS
Digsby	Windows
Empathy	Linux
Exodus	Windows
Gajim	Linux / Windows
Google Talk	Windows / OS X / Linux / Móvil / Web
iChat	Mac OS X
Jabbear	Windows / Web
Jabber Mix Client	Móvil
jabber.el	Linux
Jabbim	Linux / Mac OS X / Windows
Jabbim for Android	Móvil
JBuddy Messenger	Linux / Mac OS X / Windows
Kopete	Linux
Miranda IM	Windows
Pidgin	Linux / Mac OS X / Windows
Swift	Linux / Mac OS X / Windows
Synapse	Linux
Trillian	Windows / Mac OS X / Móvil / Web

Table 5. Lista de clientes populares

- Mensajes instantáneos.
- Actualización de estado.
- Tránsferencia de archivos.
- Llamadas gratuitas: realiza llamadas desde tu equipo a cualquier equipo del mundo.
- Conferencia de audio: habla con varias personas a la vez.

7.4 Pidgin

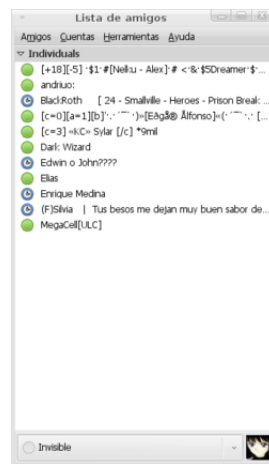


Fig. 7. Pidgin

Es uno de los clientes que implementan XMPP más utilizados. Si bien es un cliente multiprotocolo, nos enfocaremos sólo en las funcionalidades disponibles con XMPP.

Implementa algunas funcionalidades más que Google Talk:

- Permite el reemplazo de nombres en la lista de contactos.
- Videollamadas.
- Geolocalización.
- Consola XMPP para enviar y recibir XMPP en crudo.
- Descubrimiento de servicios XMPP.
- Capacidad para conectarse con múltiples cuentas en simultáneo.

8 Otras aplicaciones

Como dijimos anteriormente, una de las características más interesantes de XMPP es la extensibilidad. Existen numerosas extensiones muy interesantes que

permiten agregar funcionalidades más allá de la mensajería instantánea, aunque todavía se encuentran todavía en una fase experimental.

Una de la más interesantes es Jingle (XEP-0166), una extensión que permite manejar sesiones multimedia punto a punto, utilizando cualquier protocolo de transporte existente(TCP, UDP, inclusive protocolos específicos de la aplicación). Esto permitirá implementar funcionalidades de voz, video, transferencia de archivos entre otras cosas.

Otra extensión interesante es Publish-Subscribe(XEP-0060), una extensión que permite crear servicios de notificación de eventos. Esto permitirá implementar funcionalidades de geolocalización, sindicación de contenido, administración de redes, servicios de correo, redifusión de contenido.

Una lista completa de la extensiones en fase experimental se puede encontrar en [1].

Por más que se encuentren fase experimental, muchas de las extensiones ya están siendo utilizadas por algunos servicios, especialmente Jingle.

Sería muy interesante ver un servicio de correo corriendo sobre XMPP. Sería mucho mas eficiente un servicio de notificación de correo comparado a como funciona el correo hoy en día, donde el cliente está consultando al servidor cada cierto intervalo de tiempo.

Nombre	Lenguaje
agsXMPP SDK	C# / .NET / Mono
AnyEvent::XMPP	Perl
as3xmpp	Flash / ActionScript
AXMPP	Ada
Babylon	Ruby
Blather	Ruby
cl-xmpp	Lisp
dojox.xmpp	JavaScript
Echomine Feridian	Java
Eiffel	PHP
emite	Java
gloox	C++
headstock	Python
hsxmpp	Haskell
iksemel	C
IP*Works Internet Toolkit	ActiveX, C++, C#, .NET, Mono, Delphi, Java
Iris	C++
jabber-net	C# / .NET / Mono
jabber.py	Python
Jabber Stream Objects (JSO)	Java
JAXL	PHP
Loudmouth	C / Ruby
Net::XMPP	Perl
pyxmpp	Python
QXmpp	C++
seesmic-as3-xmpp	Flash / ActionScript
SleekXMPP	Python
Smack	Java
Swiften	C++
Tinder	Java
txmpp	C++
Twisted Words	Python
Verse	Lua
XIFF	Flash / ActionScript
xmpp-psn	Python
xmpp4gwt	JavaScript
xmpp4js	JavaScript
XMPP4R	Ruby
xmppphp	PHP
xmpppy	Python

Table 6. Lista de librerías de desarrollo

References

1. <http://www.xmpp.org>: (Xmpp standards foundation webpage)
2. Adams, D.: Jabber-rpc. Number 0009 in XEP, XMPP Standards Foundation (2006)
3. Garca, J.C.D.: Estudio del protocolos xmpp de mensajera instantnea, de sus antecedentes, y de sus aplicaciones civiles y militares. (2008)
4. Jeremie Miller, Thomas Muldowney, P.S.A.: Last activity. Number 0012 in XEP, XMPP Standards Foundation (2008)
5. Joe Hildebrand, P.S.A.: Stream compression. Number 0138 in XEP, XMPP Standards Foundation (2009)
6. Joe Hildebrand, Peter Millard, R.E.P.S.A.: Service discovery. Number 0030 in XEP, XMPP Standards Foundation (2008)
7. Peter Saint-Andre, D.S.: Chat state notifications. Number 0085 in XEP, XMPP Standards Foundation (2009)
8. Peter Saint-Andre, M.N.: Entity time. Number 0202 in XEP, XMPP Standards Foundation (2009)
9. Ryan Eatmon, Joe Hildebrand, J.M.T.M.P.S.A.: Data forms. Number 0004 in XEP, XMPP Standards Foundation (2007)
10. Saint-Andre, P.: In-band registration. Number 0077 in XEP, XMPP Standards Foundation (2009)
11. Saint-Andre, P.: Xmpp ping. Number 0199 in XEP, XMPP Standards Foundation (2009)
12. Saint-Andre, P.: Delayed delivery. Number 0203 in XEP, XMPP Standards Foundation (2009)
13. Saint-Andre, P.: Serverless messaging. Number 0174 in XEP, XMPP Standards Foundation (2008)
14. Saint-Andre, P., Smith, K., TronCon, R.: XMPP: The Definitive Guide: Building Real-Time Applications with Jabber Technologies. 1 edn. O'Reilly Media (2009)
15. Day, M., Rosenberg, J., Sugano, H.: A Model for Presence and Instant Messaging. RFC 2778 (Informational) (2000)
16. Day, M., Aggarwal, S., Mohr, G., Vincent, J.: Instant Messaging / Presence Protocol Requirements. RFC 2779 (Informational) (2000)
17. Saint-Andre, P.: Extensible Messaging and Presence Protocol (XMPP): Core. RFC 3920 (Proposed Standard) (2004) Obsoleted by RFC 6120, updated by RFC 6122.
18. Saint-Andre, P.: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. RFC 3921 (Proposed Standard) (2004) Obsoleted by RFC 6121.
19. Saint-Andre, P.: Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM). RFC 3922 (Proposed Standard) (2004)
20. Saint-Andre, P.: End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP). RFC 3923 (Proposed Standard) (2004)