

# **Proseminar Virtuelle Präsenz**

Universität Ulm

Sommersemester 2005

Jabber/XMPP

Florian Holzauer

mail: fh@fholzauer.de

jabber: fh@zwoop.de

# Inhaltsverzeichnis

1. Geschichte und Hintergründe zu Jabber.....	2
1.1 Anstoss für Jabber.....	2
1.2 Geschichte von Jabber.....	3
1.3 Neuerungen an Jabber.....	3
1.4 Problemstellen.....	3
1.5 Neuerungen.....	4
2. Technik und Semantik.....	5
2.1 Semantik: JID.....	5
2.2 Jid-Parsing.....	5
2.3 Serverübergreifende Nachrichten.....	5
2.4 Routing innerhalb eines Jabber-Servers.....	6
2.5 Transport - Gateways.....	6
2.6 Protokoll.....	7
2.7 Beispiel: jabber.ccc.de.....	7
3. Ausblicke.....	9
3.1 Client: Lluna.....	9
3.2 Protokoll: Pubsub.....	9
3.3 Anwendung: trakm8.....	10
4. Anhang.....	10
4.1 URL-Verzeichnis:.....	10

## 1. Geschichte und Hintergründe zu Jabber

Jabber wird seit 1998 entwickelt, und ist heute IETF-Standard für Instant Messaging. Dieser erste Teil soll einen groben Überblick über Intention, Geschichte sowie Vor- und Nachteile geben.

### 1.1 Anstoss für Jabber

Seit etwa 1996<sup>1</sup> existiert Instant Messaging in der Form, wie wir sie heute kennen. Die vier grossen Messaging-Systeme ICQ<sup>a</sup>, AIM<sup>b</sup>, Yahoo!<sup>c</sup> und MSN<sup>d</sup> basieren alle auf grösstenteils proprietären und nicht öffentlichen Protokollen, und sind fast durchgehend nicht miteinander kompatibel, lediglich die Kommunikation zwischen ICQ und AIM ist möglich.

Die offiziellen Clients dieser Messaging-Systeme werden meist nur für Windows-Systeme angeboten, sind relativ unübersichtlich und häufig auch durch Werbung innerhalb der Programme finanziert.

Unerlässlicher Bestandteil der Infrastruktur dieser Clients sind die Server der jeweiligen Systembetreiber, ein eigener Betrieb eines solchen Servers oder gar eines geschlossenen Netzes ist nicht möglich. Auch ist es so nicht möglich, einem Ausfall dieser Server vorzubeugen – fallen diese zentralen Server aus, ist die Nutzung des Dienstes nicht mehr möglich.

Client-Software für andere Betriebssysteme oder sogenannte Multi-Protocol-Clients existieren zwar, sind aber fast durchgehend durch nicht autorisiertes Reverse Engineering der Protokolle des jeweiligen Messengers entstanden. Eine Protokolländerung seitens der Messenger-Betreiber kann so durchaus dazu führen, dass diese inoffiziellen Clients erst durch ein Update wieder benutzbar sind – das natürlich erst dann möglich ist, wenn das neue Protokoll entschlüsselt wurde. Teilweise sind diese inoffiziellen Clients auch innerhalb der Servernetze verboten, so gibt es zum Beispiel eine

<sup>1</sup> ICQ entstand 1996, alle anderen Messenger zwischen 1997 und 1999 – siehe [http://de.wikipedia.org/wiki/Instant\\_Messaging](http://de.wikipedia.org/wiki/Instant_Messaging)

Liste<sup>e</sup> mit autorisierten Clients für MSN. Lediglich mit diesen Clients ist die Nutzung des MSN-Service gestattet. Eine Nutzung mit anderen Clients ist zwar technisch möglich, aber explizit verboten.

## **1.2 Geschichte von Jabber**

All diese Probleme, wie auch beispielsweise Datenschutzfragen<sup>2</sup>, die durch den Serverstandort in einem anderem Land mit anderer Gesetzgebung entstehen, führten dazu, dass 1998 Jeremie Miller mit der Entwicklung von Jabber begann, einem Protokollstandard auf XML-Basis, der vollständig offen sein sollte.

Im Januar 1999 kündigte Jeremie die Entwicklung dieses Protokolls sowie darauf basierender Applikationen auf der Nachrichten-Seite Slashdot an<sup>3</sup>. Dieses Posting wird heute als Geburtsstunde von Jabber gesehen, da sehr viele Entwickler so auf das Protokoll aufmerksam wurden, und die Entwicklung so deutlich beschleunigt wurde.

Anfang März 2000 wurde Jabber Inc<sup>f</sup> gegründet, eine US-Amerikanische Firma, die Infrastruktur und Dienste für das Jabber-Projekt bereitstellt, und sich über einen kommerziellen Jabber-Server sowie Beratungsdienstleistungen und kommerzielle Entwicklung finanziert.

Im Mai 2000 wurde dann der erste offizielle Jabberserver, jabberd 1.0, veröffentlicht.

Durch das grosse Interesse an Jabber sowie die Vielzahl an verschiedensten Entwicklungen und Erweiterungen des Protokolls wurde im August 2001 die Jabber Software Foundation (JSF)<sup>g</sup> gegründet. In der JSF wird die Entwicklung und Standardisierung des Jabber-Protokolls in Form von JEPs, Jabber Enhancement Proposals, koordiniert und vorangetrieben.

So wurde im Oktober 2004 ein Teil des Jabber-Protokolls, XMPP<sup>h</sup>, ein RFC<sup>i</sup>-Standard. Diese RFCs werden von der IETF<sup>j</sup> ratifiziert und legen Standards im Internet fest. Inzwischen ist XMPP auch als offizieller Standard für Instant Messaging im Internet definiert worden.

## **1.3 Neuerungen an Jabber**

Durch die Eingangs erwähnten Probleme der bestehenden Messaging-Protokolle gibt es verschiedene Kern-Ziele, die elementar für die Entwicklung von Jabber sind. Neben der offenen und standardisierten Protokollbasis ist sind das vor allem die Abwärtskompatibilität – soweit es sinnvoll ist, können Jabber-Clients die nicht jeden Protokollerweiterung unterstützen, trotzdem die Inhalte darstellen und über Jabber kommunizieren – ein Client muss also nicht das vollständige Spektrum der Möglichkeiten des Jabber-Protokolls beherrschen.

Durch verschiedenste Ideen wurde auch dem Sicherheits- und Datenschutzaspekt Rechnung getragen. So ist der Betrieb der Server dezentral, die Vernetzung findet dynamisch statt. Sowohl die Kommunikation zwischen Benutzer und Server als auch zwischen zwei Benutzern kann auf verschiedene Arten verschlüsselt werden. Der Betrieb eines Jabber-Servers ist für jeden möglich, und durch die Client-Vielfalt gibt es selbst für exotische Plattformen einen Jabber-Client<sup>k</sup>.

## **1.4 Problemstellen**

Es darf natürlich nicht verschwiegen werden, dass ein so junges Protokoll auch seine Schattenseiten hat. Die Entwicklung und der Betrieb des heutigen Jabber-Verbundes findet meist im Hobbybereich statt, was manchmal der Qualität und Verständlichkeit für den Benutzer nicht unbedingt zuträglich ist. Als sehr positives Beispiel für einen Jabber-Server sei hier der grösste deutsche Server, amessage<sup>l</sup> aus München, genannt.

Sehr problematisch ist hier vor allem das klassische Anwendungsproblem, wenn Techniker für Endanwender programmieren. Ein unbedarfter Benutzer, der bislang keinen Kontakt zur „Jabber-

2 Als Beispiel sei <http://yro.slashdot.org/yro/05/03/11/2359226.shtml> genannt

3 <http://slashdot.org/article.pl?sid=99/01/04/1621211&tid=126>

Welt“ hatte, wird mit einer schier unüberschaubaren Menge an Optionen und schwer verständlichen Fachbegriffen überschüttet.

## 1.5 Neuerungen

Diesen Problemen stehen aber auch interessante Vorteile gegenüber, die zum Teil vollständig neu sind, zum Teil auch bereits bei dem einen oder anderem herkömmlichen Messenger existieren. So ist die Contact List (bei Jabber Roster genannt), also die Liste der Kontakte, dessen Online-Status der Jabberbenutzer sehen möchte auf dem Server gespeichert, genauso wie auch Konfigurationsoptionen des Clients. Im Idealfall kann so ein Benutzer also an einer neuen Arbeitsstation einfach seinen Client starten, und sich in seiner altgewohnten Umgebung wiederfinden.

Nachrichtenfilterung ist – sofern der Jabberserver das unterstützt<sup>4</sup> - bereits auf dem Server möglich, Nachrichten die der Benutzern nicht sehen möchte, belasten so also gar nicht erst seine Bandbreite.

Selbstverständlich ist auch der parallele Betrieb mehrerer Clients problemlos möglich.

Gerade im Unternehmensbereich gern genutzt ist die Tatsache, dass eine Jabber-ID optisch einer Email-Adresse entspricht – es wird hier oft eine gleich lautende ID wie Mailadresse benutzt, so dass man weniger Informationen austauschen muss, was auch der Übersichtlichkeit dient. Ebenfalls bei Unternehmen sehr gerne gesehen ist das Feature der dynamic Rosters. Durch die Speicherung der Roster, der oben bereits erwähnten Kontaktliste, kann man gegebenenfalls diese Liste einfach und automatisiert manipulieren. Wenn also ein Mitarbeiter einer neuen Arbeitsgruppe zugeteilt wird, werden so automatisch seine neuen Kollegen in den Roster eingefügt – und nach Projektende verschwinden sie dort auch wieder. Genauso sind virtuelle Benutzer möglich – am Beispiel einer Universität würde das zum Beispiel bedeuten, dass ein Student, der an der Vorlesung Praktische Informatik 1 teilnimmt, automatisch auch den virtuellen Kontakt „[pi1-tutor@unidomain.tld](mailto:pi1-tutor@unidomain.tld)“ im Roster hat – der den Status online einnimmt, sobald einer der Tutoren dieses Fachs online ist.

Wichtig bei Jabber ist auch die Thematik der Verschlüsselung. Es gibt verschiedenste Techniken und Ansätze zur Nachrichtenverschlüsselung bei Jabber, angefangen von Techniken wie SSL oder TLS, die lediglich die Kommunikation zwischen Client und Server verschlüsseln, bis hin zu OTR<sup>m</sup> oder GPG, Verschlüsselungsmechanismen die Serverübergreifend den Nachrichtenverkehr sichern. So kann selbst ein neugieriger Server-Administrator nicht mehr mitlesen.

Durch den offenen Standard ist es natürlich auch möglich, mit eigenen Mitteln und relativ schnell Scripte und Programme für Jabber zu erstellen. Gerade hier liegt ein grosser Reiz dieses Protokolls.

4 `mod_filter` ist bei verschiedenen grösseren Jabber-Servern deaktiviert, da sich hier auch Nachrichtenloops bilden können, die zu sehr hoher Serverlast führen.

## 2. Technik und Semantik

Auch wenn durch den begrenzten Platz dieser Ausarbeitung nicht tief auf die Technik und Syntax hinter Jabber eingegangen werden kann, soll im folgendem doch zumindest ein Überblick gegeben werden. Weitergehende Informationen bieten sowohl die RFCs 3920-3923, als auch die Dokumentation auf [xmpp.org](http://xmpp.org) und [jabber.org](http://jabber.org)

### 2.1 Semantik: JID

Die Jabber-ID, kurz JID sieht optisch fast gleich wie eine Email-Adresse aus, auch die Semantik ist sehr ähnlich.

Eine vollständige Jabber-ID sieht zum Beispiel so aus:

**fh@zwoop.de/MyClient**

Diese ID setzt sich aus drei Teilen zusammen:

Der Teil vor dem @, in unserem Beispiel also fh, ist der Benutzername, der auf jedem Server einmalig ist. Hinter dem @ ist analog zur Email-Adresse die Domain des Jabber-Servers angegeben. Der dritte Teil ist meist optional – und wird auch selten angegeben, wenn man seine Jabber-ID angibt. Es handelt sich hier um die sogenannte Resource – also eine Identifikation des jeweiligen Clients, die frei wählbar ist. Da jeder Jabber-Benutzer mit mehreren Clients gleichzeitig online sein kann, dient diese Resource zur Unterscheidung der einzelnen Clients, es ist auch nicht möglich, mehreren Clients den gleichen Ressourcenamen zuzuordnen. Üblich sind hier vor allem Orts-Namen wie zum Beispiel „daheim“ oder „Laptop“.

Manchmal wird zusätzlich hinter der Jabber-ID, in diesem Fall grundsätzlich mit Resource, eine Zahl angegeben, häufig in Klammern – also beispielsweise **fh@zwoop.de/MyClient (4)**. Es handelt sich hier um die Priority, einer Zahl, die der Priorisierung der einzelnen Clients dient.

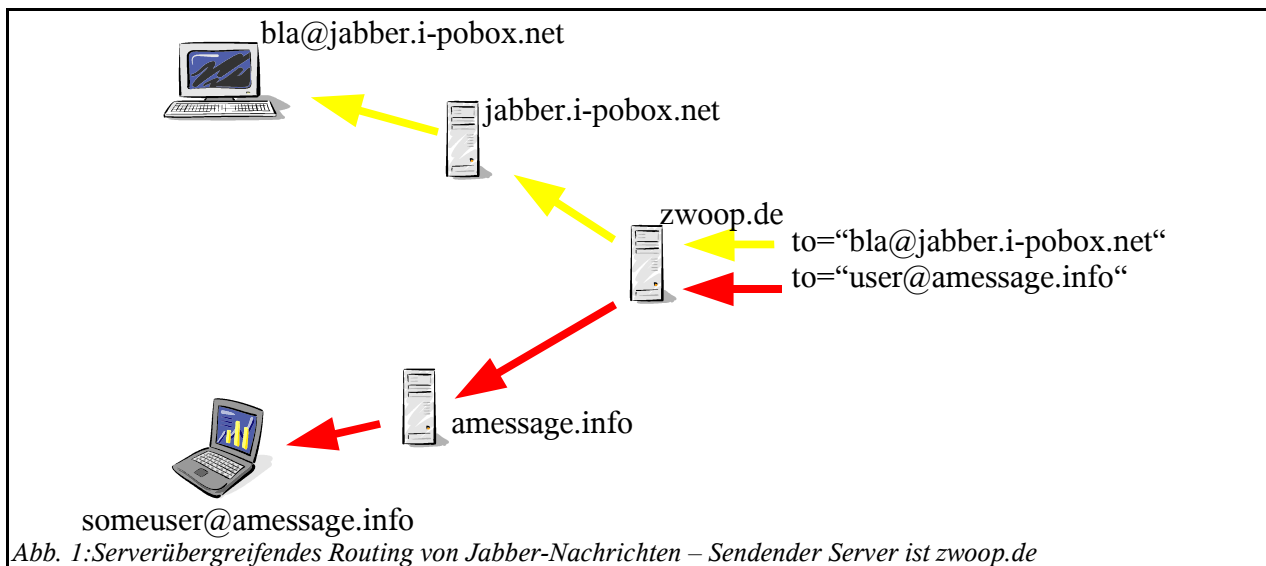
### 2.2 Jid-Parsing

Jabber-Ids, denen der Teil vor dem @ fehlt, sind meist administrativer Natur, und dienen der Kommunikation mit dem Server. So ist die Jabber-ID **servername.tld/echo** – mit explizit angegebener Resource „echo“ ein sehr hilfreiches Werkzeug – die an diese Jabber-ID geschickte Nachricht wird vom Server einfach zurückgeschickt. Wenn die Nachricht also erfolgreich wieder zurück kommt, ist klar, dass die Verbindung zum getestetem Server problemlos funktioniert.

Eine Nachricht an **benutzer@servername.tld** wird an den Client mit der höchsten Priorität geschickt, der gerade online ist. Will man an einen Client eine Nachricht schicken, der nicht die höchste Priorität hat, ist das natürlich ebenfalls möglich – man muss hierzu als Empfänger dann explizit die Resource mit angeben.

### 2.3 Serverübergreifende Nachrichten

Auch Serverübergreifendes Nachrichtenrouten funktioniert entsprechend. Wenn eine Nachricht an einen anderen Jabber-Server geschickt werden soll, baut der Jabber-Server des Absenders eine Verbindung zum Empfänger-Server auf, sofern diese nicht bereits besteht, und übermittelt dann die Nachricht. Dieser Mechanismus funktioniert analog zu Emails, es gibt also keine starre Hierarchie mit mehreren Hops, sondern ein dynamisch aufgebautes Netz.



Der Verbindungsaufbau findet hier mittels eines Drei-Wege Handshakes statt, um sicherzustellen, dass die Absender-Domain auch der Wahrheit entspricht. Modernere Serverversionen können diese Verbindung mittels TLS auch verschlüsseln, allerdings ist dieses Feature noch nicht in allen Servern implementiert.

## 2.4 Routing innerhalb eines Jabber-Servers

Die meistens Jabber-Daemons sind modular aufgebaut, und kommunizieren über TCP/IP miteinander – so ist es auch ohne grossen Aufwand möglich, bei starker Serverlast die einzelnen Teile eines Servers auf verschiedene Maschinen auszulagern, um die Belastung zu verteilen.

Kernstück jedes Servers ist der Router, der die einzelnen Protokollpakete verteilt – auch innerhalb des Jabberd wird mit Streaming XML kommuniziert. Üblicherweise besteht ein Jabber-Server zusätzlich noch aus einem c2s („Client to Server“) Modul, über das die Kommunikation mit den Clients stattfindet, einem s2s („Server to Server“) Modul für die Serverübergreifende Kommunikation sowie zwei internen Modulen, einem DNS-Resolver und einer Instanz für das Sessionmanagement. Je nach konkreter Konfiguration finden sich zusätzlich noch Transports zu anderen IM Netzen oder Gateways, die verschiedene Dienste übernehmen. Bei Servern mit mehr als 500 Benutzern parallel online wird c2s und s2s gerne auch auf mehrere Instanzen parallelisiert.

## 2.5 Transport - Gateways

Eines der heute am häufigsten genutzten Features von Jabber sind sogenannte Transports. Es handelt sich hier um Gateways in andere Instant Messenger Netze – über solche Gateways ist es also möglich, mit Benutzern von ICQ, AIM und ähnlichem zu kommunizieren. Das Gateway bildet hier das fremde Netzwerk nach Jabber ab, die Umsetzung ist für den Client also vollständig transparent. Eine ICQ-Uin spricht man so beispielsweise mit der Jabber-ID

**739201@icq.jabber.ccc.de**

an – die ICQ-Nummer steht hier als Benutzername vor dem @.

Es ist auch problemlos möglich, Gateways anderer Server über die s2s Kommunikation zu nutzen, allerdings sind die Gateways öffentlicher Server oft überlastet. Es empfiehlt sich also, sofern die technische Infrastruktur vorhanden ist, ein eigenes Gateway aufzubauen und zu nutzen.

In der Jabberwelt ist hier die Benennung dieses Dienstes nicht ganz klar – an manchen Stellen wird von Transports, an anderen von Gateways gesprochen. Üblich ist es, von einem Transport zu

sprechen, wenn es sich um eine „Übersetzung“ in ein anderes Netzwerk handelt, und allgemeiner von Gateways wenn es sich um irgendeine Form von Dienst handelt, der ein anderes Medium nach Jabber transportiert. Als Beispiel seien hier zum Beispiel RSS-Gateways oder Email-Gateways genannt.

Es gibt Dutzende verschiedener Gateways für verschiedenste Dienste. Neben den IM-Diensten sind die beiden bereits angesprochenen RSS-Gateway und Email-Gateway sehr populär, mit denen sich Nachrichten bzw Emails über Jabber signalisieren lassen. Gerne genutzt werden auch SMS-Gateways, hier sei beispielhaft `aspsms.swissjabber.ch` genannt, Gateways ins IRC<sup>1</sup>, oder Terminerinnerungsdienste.

Eine gute, wenn auch nicht vollständige, Übersicht über diese Gateways (sowie allgemein Jabber-basierte Software) bietet `jabberstudio.org`.

## 2.6 Protokoll

Jabber basiert auf XML – sowohl als Protokoll als auch in den Konfigurations- und Spooldateien. Die Kommunikation via Jabber findet in Form von „Streaming XML“ statt.

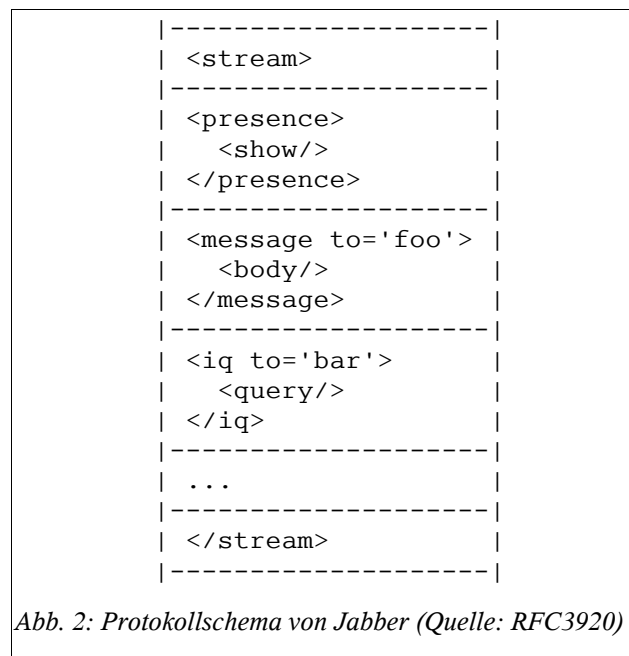


Abb. 2: Protokollschema von Jabber (Quelle: RFC3920)

Interessant ist an diesem XML-Ansatz vor allem, dass dieses Protokoll einfach und nahezu beliebig mittels eigener Namespaces erweiterbar ist, es können also prinzipiell beliebige Daten über Jabber transportiert werden. Diese Erweiterungen werden mittels sogenannter JEPs, Jabber Enhancement Proposals durch die JSF zu Protokollstandards, also formal definierte Erweiterungen des Jabber Protokolls, die Clients implementieren können.

## 2.7 Beispiel: `jabber.ccc.de`

Um ein grobes Beispiel für aktuelle Zahlen und Server-Layout eines mittelgrossen Jabber-Servers zu nennen, seien hier Zahlen des öffentlichen Jabber-Servers `jabber.ccc.de` genannt<sup>2</sup>:

Der Server bietet Gateways zu allen Instant Messenger Systemen, ein Benutzerverzeichnis und Conferences an. Täglich sind je nach Tageszeit zwischen 400 und 1200 Clients online, ca 50%

<sup>1</sup> Internet Relay Chat

<sup>2</sup> Stand: Mai 2005

davon über verschlüsselte Verbindungen. Innerhalb einer Woche wurden 1,5 Millionen Nachrichten und knapp 4 Millionen Statusnachrichten übermittelt, ein Teil davon an die rund 500 Server, mit denen jabber.ccc.de im Moment verbunden ist. Registriert sind knapp 10000 Benutzer.

Jabber.ccc.de ist seit Dezember 2003 im Betrieb, und läuft auf einem Linux-basiertem Server mit 1 Gigabyte Ram, 2.5 Ghz CPU und RAID1-Festplatten. Im Laufe des Betriebs hat sich gezeigt, dass vor allem Ram und CPU die limitierenden Faktoren im Betrieb eines solchen Servers sind – da die Statusinformationen jedes Benutzers, der gerade online ist im Speicher vorgehalten werden muss, ist der Ram-Verbrauch durchaus beachtlich. Die CPU-Last entsteht insbesondere durch das zeitaufwendige Parsen von grossen verschachtelten XML-Paketen, die beispielsweise beim Login eines Benutzers anfallen, der viele Offline-Nachrichten vorliegen hat.

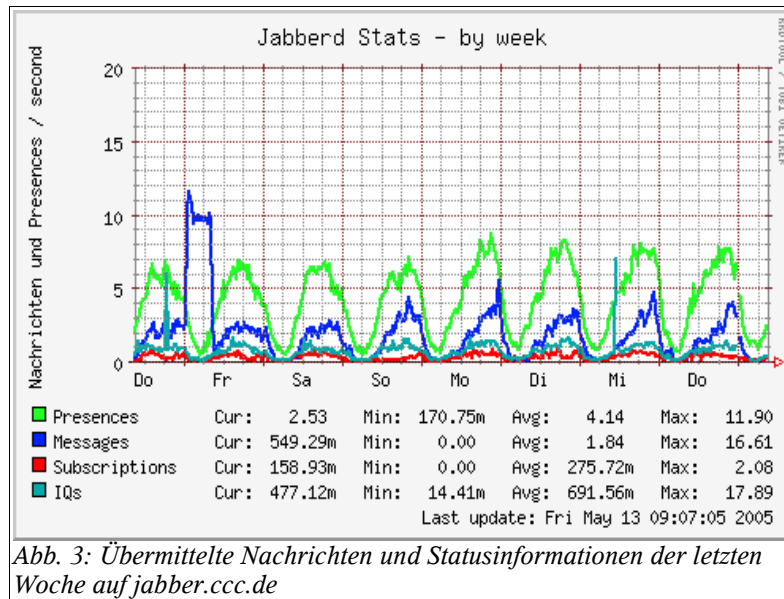


Abb. 3: Übermittelte Nachrichten und Statusinformationen der letzten Woche auf jabber.ccc.de

Der grösste öffentliche Jabber-Server, jabber.org, hat knapp 200 000 registrierte Benutzer, von denen meist etwa 5000-7000 Clients online sind.



### 3. Ausblicke

Sehr interessant ist Jabber-Bereich auch die Weiterentwicklung von Jabber, bedingt durch das noch sehr junge Medium entstehen einige vielversprechende Produkte. Im Folgendem seien einige exemplarisch herausgegriffen.

#### 3.1 Client: Lluna

Ein Client mit etwas anderem Ansatz ist der Jabber-Client Lluna<sup>9</sup>, der eher für das Umfeld des „Collaborative Browsers“ gedacht ist.

Lluna ist eine Erweiterung des Webbrowsers, die den Benutzer als Avatar an der Fusszeile des Browsers darstellt. Andere Besucher der selben Website, die ebenfalls Lluna installiert haben, befinden sich ebenfalls als Avatare auf der Seite, so dass eine Kommunikation mittels stilisierter Sprechblasen und Gesten zwischen Besuchern der Website, die gerade geöffnet ist, ermöglicht wird.

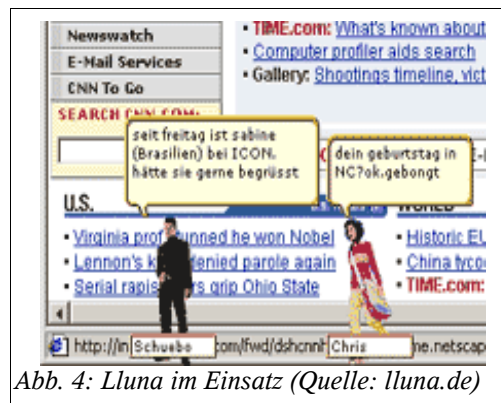


Abb. 4: Lluna im Einsatz (Quelle: lluna.de)

Gerade im Hinblick auf Dienstleistungen des Websitebetreibers ist diese Applikation sehr reizvoll, bietet sie doch zum Beispiel die Möglichkeit, Supportmitarbeiter ebenfalls an den Diskussionen der Besucher teilhaben zu lassen, und so gegebenenfalls bei Fragen und Problemen zu assistieren.

Realisiert wird diese Technik mit Jabber-Conferences, also Gruppenchats auf Jabber-Basis. Es ist durch das zugrundeliegende Protokoll für einen Jabber-Server-Betreiber übrigens nicht möglich herauszufinden, auf welchen Webseiten sich die Benutzer im Moment bewegen, da die URL als eine MD5-Summe verschlüsselt übermittelt wird.

Lluna wird von der Firma Bluehands<sup>0</sup> aus Karlsruhe entwickelt, und hat seinen Ursprung an der Universität Ulm.

#### 3.2 Protokoll: Pubsub

Eine relativ neue und noch selten verbreitete Protokollerweiterung von Jabber ist Pubsub („Publish Subscribe“) (JEP0060). In dieser Erweiterung wird ein Subscriptionsservice für Inhalte definiert – ein Anbieter eines Inhaltes wie zum Beispiel eine Nachrichtenseite, der Betreiber eines Blogs, also eines digitalen „Tagebuchs“ oder ähnlichem bietet einen Channel für seine Inhalte innerhalb des Pubsub-Dienstes an. Jeder Benutzer, den diese Inhalte interessieren, kann nun diesen Channel subscriben, und bekommt bei einer Änderung des Inhalts sofort eine entsprechende Status-Information zugestellt. Diese Informationen müssen nicht zwangsläufig Text sein, es sind prinzipiell beliebige Binärdaten denkbar, also auch Audiodaten oder Bilder.

Bislang wird dieses Prinzip meist mittels sogenannten RSS-Feeds bereitgestellt. Hier muss der Benutzer in regelmässigen Abständen eine XML-Datei auf einem Webserver downloaden, und kann dann überprüfen, ob sich etwas an den Inhalten geändert hat. Der Nachteil dieses RSS-Feeds ist klar: Es entsteht viel überflüssiger Datenverkehr – der Benutzer muss die Datei immer erst downloaden, auch wenn sich nichts geändert hat, und er bekommt neue Informationen erst mit dem

nächsten Aktualisierungsintervall, was unter Umständen durchaus mehrere Stunden Verzögerung bedeuten kann.

Es gibt erste Anbieter, die Pubsub bereits nutzen, wie zum Beispiel der Dienst Mimir<sup>p</sup>, der vom Jabber-Entwickler Ralph Meijer zur Verfügung gestellt wird. Dieser Dienst aggregiert verschiedene Nachrichtenseiten über HTTP, und verschickt neue Nachrichten wahlweise als Jabber-Nachricht oder Pubsub-Notification.

Leider ist der Support für Pubsub bislang auf der Client-Seite sehr schlecht, und dementsprechend selten genutzt. Mehrere grössere Jabber-Clients wie Psi oder jBother werden in den nächsten Versionen allerdings Support für Pubsub anbieten, daher wird sich dieser Engpass in der nächsten Zeit entschärfen.

### **3.3 Anwendung: trakm8**

Als Anwendung von Jabber in der Industrie sei hier beispielhaft der englische Dienst trakm8<sup>q</sup> erwähnt. Der Anbieter überwacht Fahrzeugpositionen mittels GPS, und übermittelt diese Daten über ein Jabber-Backend an seine Kunden, die so ihren Fuhrpark überwachen können. Ein Dienst, der vor allem für das Speditionswesen und die Mietwagenbranche sehr wichtig ist. Durch das Jabber-Backend war es sehr einfach möglich, die Standortdaten in verschiedensten Medien (Webbrowser, proprietäre Clients, ..) darzustellen.

## **4. Anhang**

### **4.1 Literaturverzeichnis:**

- a ICQ: <http://www.icq.com>
- b Aol Instant Messenger AIM: <http://www.aim.com>
- c Yahoo Messenger: <http://de.messenger.yahoo.com>
- d Microsoft Messenger MSN: <http://messenger.msn.com>
- e Autorisierte MSN-Clients: <http://messenger.msn.de/Help/Authorized.aspx>
- f Jabber Inc: <http://www.jabber.com>
- g Jabber Software Foundation (JSF): <http://www.jabber.org/jsf/>
- h eXtensible Messaging and Presence Protocol (XMPP): <http://xmpp.org>
- i Request for Comments (RFC): <http://www.ietf.org/rfc.html>
- j Internet Engineering Task Force (IETF): <http://www.ietf.org/>
- k Liste verschiedener Jabberclients: <http://www.jabber.org/software/clients.shtml>
- l Amessage, grösster deutscher Jabberserver: <http://web.ameessage.info>
- m Off the Record Messaging: <http://www.cypherpunks.ca/otr/>
- n Lluna: <http://lluna.de>
- o Bluehands: <http://www.bluehands.de>
- p Mimir: <http://mimir.ik.nu>
- q Trakm8: <http://mytrakm8.com>