

Jabber - mehr als Chat, Filetransfer und Co.

Chemnitzer Linux-Tage 2005

6. März 2005

info@jabber-fanatix.org

Agenda

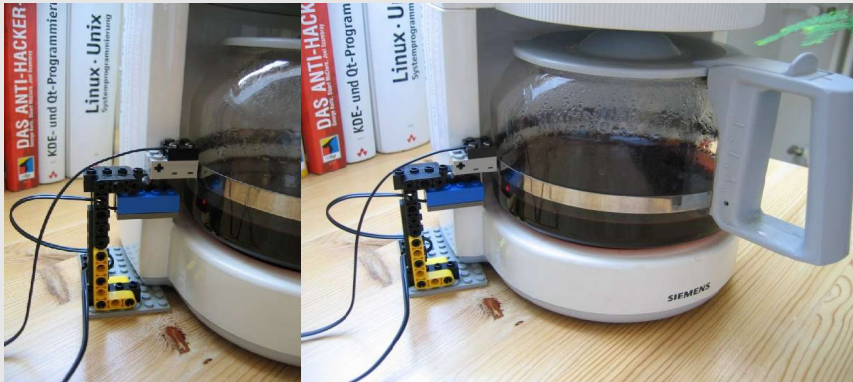
- Fallbeispiel
- Was ist Jabber?
 - Aufbau
 - Netzarchitektur
 - Grundbegriffe
- Programmieren mit Jabber
 - Voraussetzungen
 - Beispiele für eigene Implementierungen
- Features
- Jabber-basierte Projekte

Fallbeispiel – Das Problem

- Kaffee ist das Benzin der Entwickler
oder
Programmer: a biological machine to convert caffeine into code.
- Problem:
 - Weg zur Kaffeemaschine ist meist zu weit
 - Kaffee ist meistens alle
 - Niemand setzt eine neue Kanne auf
 - Kaffee ist nie frisch gebrüht
- Webcam?
 - man muß ständig einen Viewer pollen
 - informiert einen nicht wenn der Kaffee frisch zubereitet ist
 - benachrichtigt nicht den Verantwortlichen, neuen Kaffee anzusetzen
- sondern...

Fallbeispiel – Die Lösung

- Der CoffeeMachineObserver



```
coffee
[coffee@skippy.tom.blm.de]
Abonnement: to
perlsript (0)
Letzter Status @ Sam Mär 13 08:43:16 2004
Statusnachricht
coffeepot is absent
```



```
coffee
[coffee@skippy.tom.blm.de]
Abonnement: to
perlsript (0)
Letzter Status @ Sam Mär 13 08:51:35 2004
Statusnachricht
coffee is available!
```



```
coffee
[coffee@skippy.tom.blm.de]
Abonnement: to
perlsript (0)
Letzter Status @ Sam Mär 13 08:45:06 2004
Statusnachricht
coffeepot is empty
```

100 g LEGO-Bausteine
1 LEGO Mindstorms Brick
1 LEGO Mindstorms IR-Tower
20 Zeilen Perl-Code

Einführung

- XMPP: eXtensible Messaging and Presence Protocol
- XMPP/Jabber ist ein Internet-Standard
 - RFC 3920: XMPP Core
 - RFC 3921: XMPP Instant Messaging and Presence
 - RFC 3922: Mapping XMPP to Common Presence and Instant Messaging (CPIM)
 - RFC 3923: End-to-End Signing and Object Encryption for XMPP
- Implementierungen des Protokoll in vielen verschiedene Programmiersprachen verfügbar (Java, C/C++, Python, Perl...)

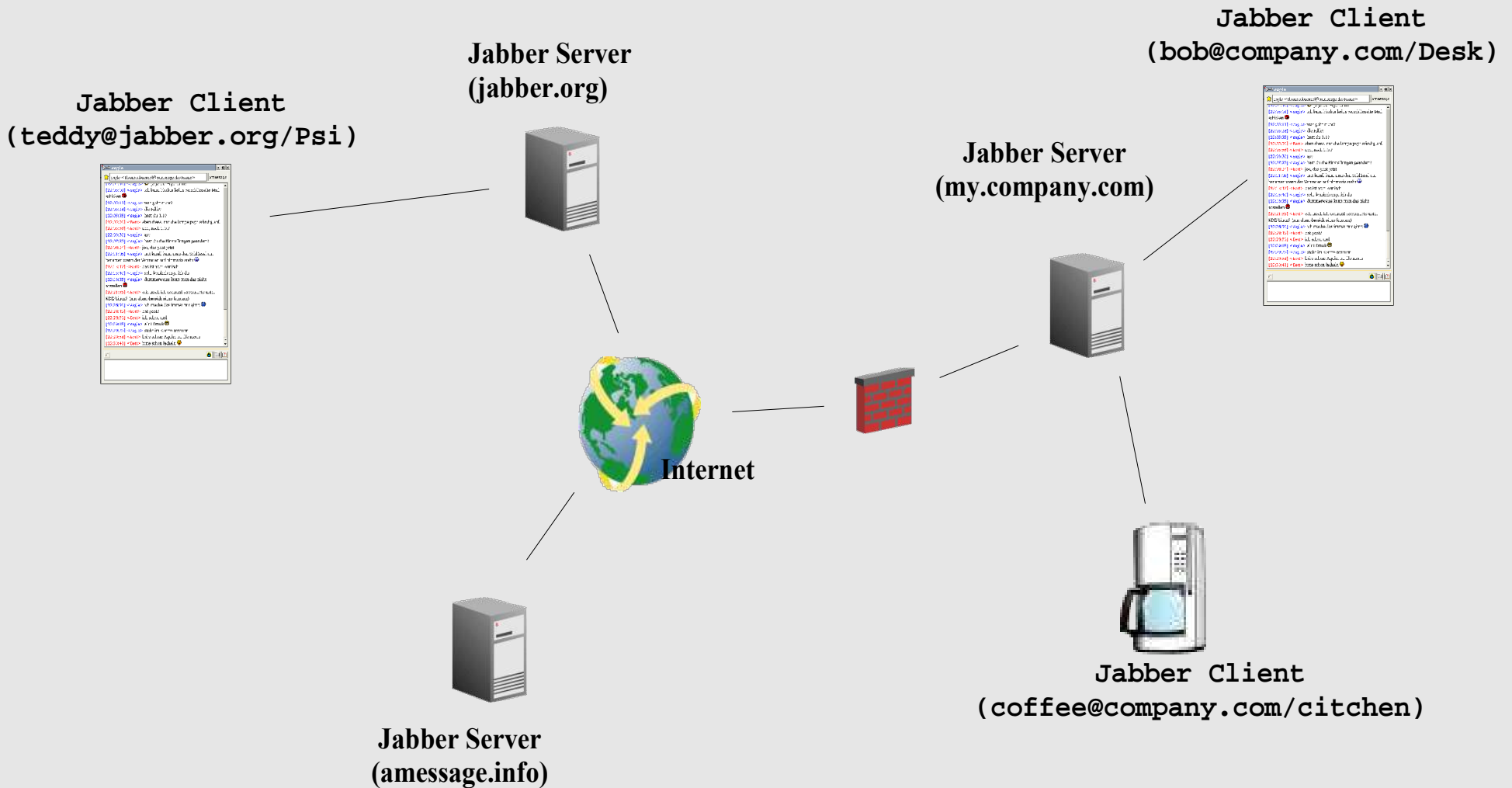
Missverständnisse über Jabber

- Jabber ist mehr als...
 - nur Instant Messaging
 - nur eine Brücke zu anderen Instant Messaging Systemen
 - nur ein Protokoll
- Jabber is best known as "the Linux of instant messaging" -- an open, secure, ad-free alternative to consumer IM services like AIM, ICQ, MSN, and Yahoo (see the IM quickstart). Under the hood, Jabber is a set of streaming XML protocols and technologies that enable any two entities on the Internet to exchange messages, presence, and other structured information in close to real time. [jabber]

Wie funktioniert es?

- Austausch von XML-Datenströmen zwischen den Endpunkten einer Verbindung
- direkte Kommunikation eines Clients mit einem einzigen Server
- Routing der XML-Nachrichten zwischen den Servern des Jabber-Netzwerkes
- XML-Datenströme transportieren
 - Presence (Statusinformationen)
 - Message (Nachrichten)
 - IQ (speziellen Payload, z.B. Authentifizierung)

Architektur des Jabber Netzwerks



Die Nutzerkennung (JID)

- username@hostname/resource
 - jabber.org
 - user@jabber.org
 - jabber.org/announce/online
 - user@jabber.org/Laptop

Die Verbindung zum Server

- TCP Verbindung
- bidirektionaler XML Stream
- TLS Handshake (optional)
- Authentisierung / Resource Bindung
- Message und Presence Austausch

Programmieren mit Jabber

- Jabber bietet erweiterbare Client/Server Architektur
- Jabber stellt Basisfunktionalitäten zur Verfügung
 - Authentifizierung
 - Sicherung der Verbindungen (SSL, End-to-End-Verschlüsselung)
 - Routing der Nachrichten im Netzwerk
- Community stellt Bibliotheken, IM-Clients und fertige Server-Implementierungen zur Verfügung
- Programmierer verwendet stabile und verbreitete Technologien und kümmert sich nur noch um die Umsetzung des fachlichen Problems.

Was braucht man?

- Jabber-Server
 - vorhandene Server [jabber.network] [server.available]
 - amessage.info,
 - jabber.org,
 - eigene Server [jabber.server]
 - jabberd1.x/2.x
 - ejabberd
 - jivemessenger

Was braucht man?

- Viel Zeit zur Implementierung von RFC 3920 und RFC 3921.
- ODER
- Bibliotheken, die die Jabber-Protokolle
 - auf unterschiedlichen Ebenen
 - in unterschiedlichen Abstraktionleveln
 - in den verschiedensten Sprachenimplementieren
- Zugriff auf externe Informationsquellen
 - Forums
 - Mailinglisten

Jabber-Bibliotheken

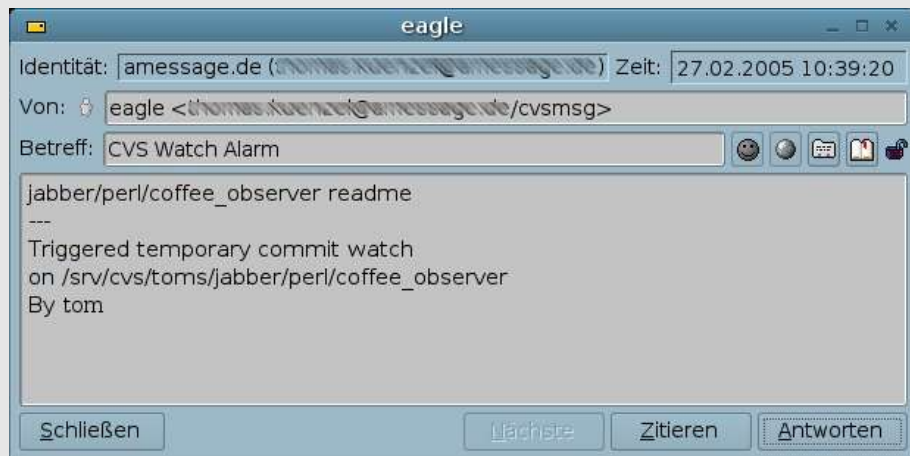
Auswahl von Bibliotheken, mit denen mehr oder weniger Erfahrungen vorliegen [jabber.lib]:

Bibliothek	Sprache	Client	Server	Comp.	Anmerkungen
smack	Java	x			High Level Bibliothek. Bietet grundlegende Funktionen zum Arbeiten mit Jabber. Leicht erlernbar.
Class.Jabber.PHP	PHP	x			High Level Bibliothek. Integration von Jabber Diensten in Web-Anwendungen.
JSO	Java	x	x	x	Low Level Bibliothek. Zugriff auf alle Details der Jabber-Protokolle. Ausbaufähige Dokumentation.
Net::Jabber	Perl	x		x	High Level Bibliothek. Eingeschränkte Doku.
jabber.py	Python	x		x	High Level Bibliothek. Gute Doku. Ausgereift, wird aber nicht mehr weiterentwickelt.
iksemel	C	x		x	XML-Parser. Small Footprint. Geeignet für Embedded Systeme.

Programmierbeispiel (1)

CVSNotificator

- CVS – Versionskontrollsystem für Teamarbeit
- Information über Status von Dateien im CVS (commit / edit / unedit)
- Mitteilung kann abhängen vom Jabber online Status



```
#!/usr/bin/python

import jabber, xmlstream
import sys

Server = 'jabber.org'
Username = 'cvsserver'
Password = 'secret'
Resource = 'cvsmsg'

cvsvuser = sys.argv[1]
message = ''

for line in sys.stdin.readlines(): message = message + line

con = jabber.Client(host=Server,port=5223,
                    connection=xmlstream.TCP_SSL)

try:
    con.connect()
except xmlstream.error, e:
    print "Couldn't connect: %s" % e
    sys.exit(0)

con.auth(Username,Password,Resource)
con.send(jabber.Message(cvsvuser, message,
                        subject="CVS Watch Alarm"))

con.disconnect()

def presenceCB(con, prs):
    type = prs.getType()
    who = prs.getFrom().getStripped()

    if type == 'subscribe':
        con.send(jabber.Presence(to=who, type='subscribed'))
        con.send(jabber.Presence(to=who, type='subscribe'))

    elif type == 'unsubscribe':
        con.send(jabber.Presence(to=who, type='unsubscribed'))
        con.send(jabber.Presence(to=who, type='unsubscribe'))

    elif type == 'available':
        prs.getStatus()
        if prs.getShow() != 'dnd' and who == cvsvuser:
            con.send(jabber.Message(cvsvuser, message, subject="CVS Watch Alarm"))
```

Programmierbeispiel (1)

Aufbau der Verbindung

```
con = jabber.Client(host=Server,port=5223,
                    connection=xmlstream.TCP_SSL)

try:
    con.connect()
except xmlstream.error, e:
    print "Couldn't connect: %s" % e
    sys.exit(0)
```

Authentifizierung beim Server

```
con.auth(Username>Password,Resource)
con.setDisconnectHandler(disconnectCB)
con.registerHandler('presence', presenceCB)
con.getRoster()
con.sendInitPresence()
```

Versenden der Nachricht(en)

```
def presenceCB(con, prs):
    type = prs.getType()
    who = prs.getFrom().getStripped()

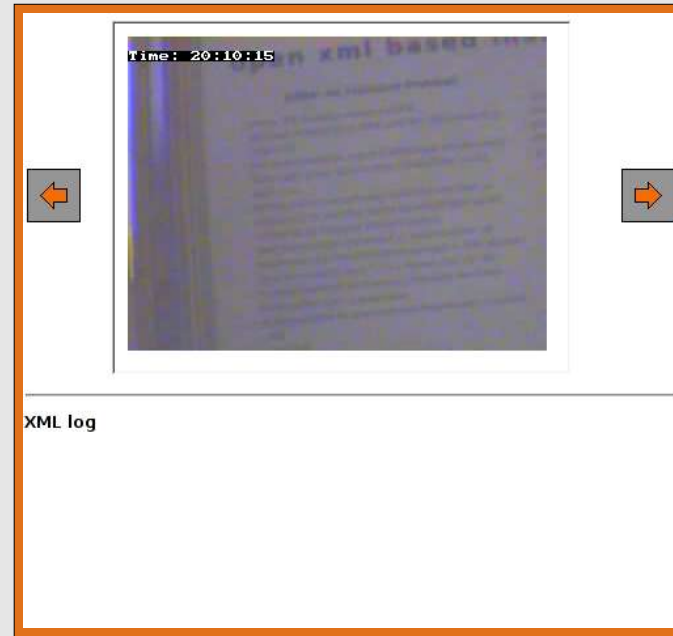
    if type == 'subscribe':
        con.send(jabber.Presence(to=who,
                                  type='subscribed'))
        con.send(jabber.Presence(to=who,
                                  type='subscribe'))
    elif type == 'available':
        prs.getStatus()
        if (prs.getShow() != 'dnd'
            and who == cvsuser):
            Con.send(jabber.Message(cvsuser,
                                     message,
                                     subject="CVS Watch Alarm"))
```

Einsatz von Python-Bibliotheken
jabber und xmlstream [jabberpy]

Pogrammierbeispiel (2)

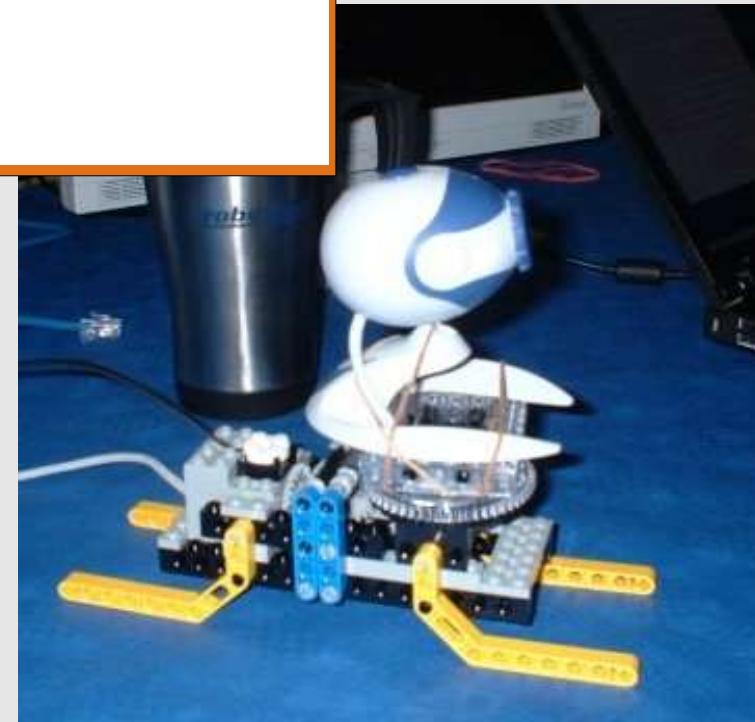
Camera

- WebCam streamt Daten via CAMSERV
- Kamera ist auf beweglichem Stativ montiert
- Stativmotor kann via Jabber-Nachrichten ferngesteuert werden



Remote

- Einfache Webanwendung
- Anzeige von Videostream, und von Buttons zur Steuerung der Kamera



Programmierbeispiel (2)

Camera-Steuerung

- Verbindungsaufbau
Listener registrieren

```
$con = new Net::Jabber::Client;
$status = $con->Connect( hostname => $server,
                        port      => $port );

# Callbacks
$con->SetCallbacks( presence => \&InPresence,
                   message => \&InMessage);

# Ident/Auth
$con->AuthSend( username => $user,
              password => $pass,
              resource => $resource );

# Roster
$con->RosterGet();

# Initial presence dependent upon initial status
set_presence($con, $initial_status);
```

- auf Nachrichten
reagieren

```
sub InMessage
{
    my $message = $_[1];
    my $from    = $message->GetFrom();
    my $type    = $message->GetType();
    my $body    = $message->GetBody();

    print "Got a wrong message from $from\n";
    my $response = new Net::Jabber::Message();
    my $response = $message->Reply();
    $response->SetSubject("Error");
    $response->SetBody($errmsg);
    $jabber->Send($response);
}
```

Einsatz des Perl-Moduls Net::Jabber [netjabber]

Programmierbeispiel (2)

Remote

- Verbindungsaufbau

```
...  
XMPPConnection connection = new XMPPConnection(  
    jabberServer, 5222);  
connection.login(remUser, remPassword);
```

- Nachrichten absetzen

```
Chat newChat = connection.createChat(  
    "camera@amessage.de/camera/door");  
Message newMessage = newChat.createMessage();  
newMessage.setBody(direction.toUpperCase());  
newChat.sendMessage(newMessage);  
...
```

Einsatz der Java-Bibliothek smack [smack]

Jabber-Features

- PubSub
 - „Abonnenten-Modell“
 - Publisher veröffentlicht Information zur PubSub Komponente
 - Subscriber abonnieren Informationen an der PubSub Komponente
 - PubSub Komponente verwaltet Rechte, verteilt bereitgestellte Informationen an alle Abonnenten
 - Entkopplung von Sender und Empfänger der Information
 - Verringerung des Aufwands im Sender
 - entwickelt sich auf Grund der Flexibilität zu einem Kernprotokoll innerhalb Jabber (Avatar, Moods, .)

Jabber-Features

- Erweiterbarkeit
 - des Servers (durch Komponenten):
 - durchsuchbare User-Datenbank gegen LDAP/SQL-Database/...
 - Gateways zu ICQ, Yahoo, MSN, IRC
 - RSS Feeds
 - Mail/Jabber Gateway
 - MultiUserChat
 - Message Logging
 - ...
 - des Protokoll (durch Verwendung eigener Namespaces)

Jabber-Features

- Verschlüsselung
 - Client - Server durch SSL
 - Server - Server (noch nicht weit verbreitet)
 - Client - Client durch Verschlüsselung/Authentisierung einzelner Nachrichten
 - Umschlag (Sender/Empfänger) bleiben lesbar
 - setzt GPG Infrastruktur voraus

Projektbeispiele

- J-EAI

- „J-EAI The first Enterprise Application Server built upon the XMPP standard“[jeai]

- Mimir

- Mimir is a Jabber enabled news service. It collects news from various sources and notifies its users when new news items have been found. [mimir]

- Volity

- Volity is a platform for creating and playing Internet-based multiplayer games, with a focus on board and card games. [volatile]

- Poncho

- Basically Poncho is a Library that provides a simple but high-level Network Interface implemented in Java and C++. It allows you to quickly network integrate devices like temperature sensors as well as server monitoring system. [poncho]

- ...

Zusammenfassung

- Jabber ist offen auf allen Ebenen
- Jabber ist eine Plattform für alle nachrichtenbasierte Anwendungsfälle
- Jabber bietet für jeden Entwickler freie Bibliotheken
- Jabber bietet große Community
- Jabber entlastet den Entwickler von „08/15“-Aufgaben (Authentifizierung, Nachrichtenrouting, etc.)
- Jabber bietet einen etablierten und stabilen Server-Verbund
- Jabber ist als IM-Plattform eingeführt und weit verbreitet
- Verstärkt werden Projekte entwickelt, die über reine IM-Funktionalität hinausgehen

- im Anschluss an Vortrag
- am Stand von jabber-fanatix
- info@jabber-fanatix.org

Macht! Mit!

Links

- [jabber] - <http://www.jabber.org/about/overview.shtml>
- [jabberpy] - <http://jabberpy.sourceforge.net/>
- [smack] - <http://www.jive-software.com/smack>
- [poncho] - <http://www.hta-bi.bfh.ch/Projects/poncho/doc.html>
- [volatile] - <http://www.volity.org/>
- [mimir] - <http://mimir.ik.nu/>
- [jeai] - http://www.erlang-projects.org/Public/projects/...eai__esb/j-eai_1.0_beta_4/view
- [jabber.lib] - <http://www.jabber.org/software/libraries.shtml>
- [jabber.server] - <http://www.jabber.org/software/servers.shtml>
- [jabber.network] - <http://www.jabber.org/network/>
- [jabber.avail] - <http://public.jabbernet.dk/unreliable.php>