

Jabber-Comets-Integration

Jabber-Comets-Integration

FU Berlin - Institut für Informatik

SoSe 2003

Internet Learning

Dozenten:

Klaus-Dieter Graf

Marco Rademacher

Referent:

Stefan Gerber

Jabber-Comets-Integration

1. Jabber-Protokoll

2. Comets-Protokoll

3. Jabber und Comets - Ein gutes Duo?

Jabber-Comets-Integration

1. Jabber Protokoll

Jabber ist ein Instant-Messaging-Dienst.

Diverse Marktforschungsunternehmen

prophezeien, dass IM mittelfristig E-mail den

Rang der beliebtesten Anwendung im Internet

ablaufen wird. Bereits heute ist die Verbreitung

rasanter als es bei E-mail der Fall war/ist.

Jabber-Comets-Integration

1. Jabber Protokoll

Instant-Messaging ist gekennzeichnet durch:

Presence (sehen wer online ist)

Messaging (Nachrichten senden/empfangen)

Conferencing (chatten)

Jabber-Comets-Integration

1. Jabber Protokoll

Das Jabber-Protokoll ist offen und standardisiert und wird daher von einer Vielzahl von Herstellern client- und serverseitig unterstützt. APIs existieren für zahlreiche Programmiersprachen und Plattformen.

Jabber-Comets-Integration

1. Jabber Protokoll

Im Gegensatz zu vielen anderen IM-Services setzt Jabber nicht auf eine P2P-Technik, sondern verwendet den klassischen Client-Server-Ansatz. Das gesamte Protokoll zwischen Client und Server besteht aus XML-Datenströmen.

Jabber-Comets-Integration

1. Jabber Protokoll

Alle Verbindungen (Sockets) zwischen Clients und dem Server bleiben bei Jabber dauerhaft geöffnet, um so PUSH-Funktionalität zu erhalten.

Jabber-Comets-Integration

1. Jabber Protokoll

Ein gutes Beispiel für den PUSH-Ansatz ist der Online-Status, den man automatisch empfängt:

```
<presence from='karl@10.0.0.1'  
to='peter@10.0.0.1'>  
<status>Online</status>  
<priority>2</priority>  
</presence>
```



Jabber-Comets-Integration

1. Jabber Protokoll

Wichtigste Funktionalität:

- Login (Authentifizierung)
 - Nachrichten senden
 - Nachrichten empfangen
 - Rostermanagement
-

Jabber-Comets-Integration

1. Jabber Protokoll

Login (Codefragment):

```
<iq type='set' id='1'>  
<query xmlns="jabber:iq:auth"  
<username>peter</username>  
<password>secret</password>  
</query>  
</iq>
```



Jabber-Comets-Integration

1. Jabber Protokoll

Nachricht senden (Codefragment):

```
<message from='peter@10.0.0.1'  
to='karl@10.0.0.1'>  
  
<body>  
Hello!  
  
</body>  
  
</message>
```



Jabber-Comets-Integration

1. Jabber Protokoll

Nachricht empfangen (Codefragment):

```
<message from='peter@10.0.0.1'>
```

```
<body>
```

```
Hello!
```

```
</body>
```

```
</message>
```

Jabber-Comets-Integration

1. Jabber Protokoll

Das Rostermanagement verwaltet potentielle Gesprächspartner auf dem Server. Eingeloggte Clients müssen ihre Präsenz selbst verwalten und allen gewünschten Kommunikationspartnern mitteilen bzw. abfragen.

Jabber-Comets-Integration

1. Jabber Protokoll

Kontaktliste anfordern (Codefragment):

```
<iq type='get' id='anfrage1'>  
<query xmlns='jabber:iq:roster' />  
</iq>
```



Jabber-Comets-Integration

1. Jabber Protokoll

Kontaktliste empfangen (Codefragment):

```
<iq type='result' id='anfrage1'>
<query xmlns='jabber:iq:roster'>
<item jid='peter@10.0.0.1' name='Peter'
subscription='both'>
<group>Uni</group>
</item>
</query>
</iq>
```

Jabber-Comets-Integration

1. Jabber Protokoll

Es geht auch einfacher.

Die Muse API von Echomine kapselt alle nötigen Protokoll- und Verbindungsrountinen in Java-Objekten.

Nachteil: Muse-Bibliothek ist recht gross.

Jabber-Comets-Integration

1. Jabber Protokoll

Mit Muse eine Verbindung aufbauen:

```
JabberContext context = new JabberContext("peter",  
    "secret", "10.0.0.1");
```

```
Jabber jabber = new Jabber();
```

```
JabberSession session =  
    jabber.createSession(context);
```



Jabber-Comets-Integration

1. Jabber Protokoll

Login (Codefragment):

```
session.connect("10.0.0.1", 5222);  
session.getUserService().login();
```



Jabber-Comets-Integration

1. Jabber Protokoll

Nachricht senden (Codefragment):

```
JabberChatService chat = session.getChatService();  
chat.sendPrivateMessage(  
    "peter@10.0.0.1", "Hello");
```



Jabber-Comets-Integration

1. Jabber Protokoll

Nachricht empfangen (Codefragment):

Da Nachrichten wie schon erwähnt, nicht aktiv abgeholt werden müssen, registriert man zum

Empfang einen JabberMessageListener:

```
session.addMessageListener(  
    new JabberMessageListener());
```



Jabber-Comets-Integration

1. Jabber Protokoll

Wie geht es nun weiter?

Um das Protokoll weiter entdecken und testen zu können, empfiehlt sich das Aufsetzen eines eigenen Jabber-Servers. Mit einem der zahlreichen Jabber-Clients und einem Proxy-Server können die XML-Streams visualisiert und analysiert werden.

Jabber-Comets-Integration

1. Jabber Protokoll

Tools:

jabberd (Open Source Server in C für Linux)

gabber (GNOME Jabber-Client)

SimpleProxyServer (Proxy aus Java in a Nutshell)

Muse API (High-Level API in Java)

Commandline Jabber-Client (Verwendet Muse)

Jabber-Comets-Integration

1. Jabber-Protokoll

2. Comets-Protokoll

3. Jabber und Comets - Ein gutes Duo?

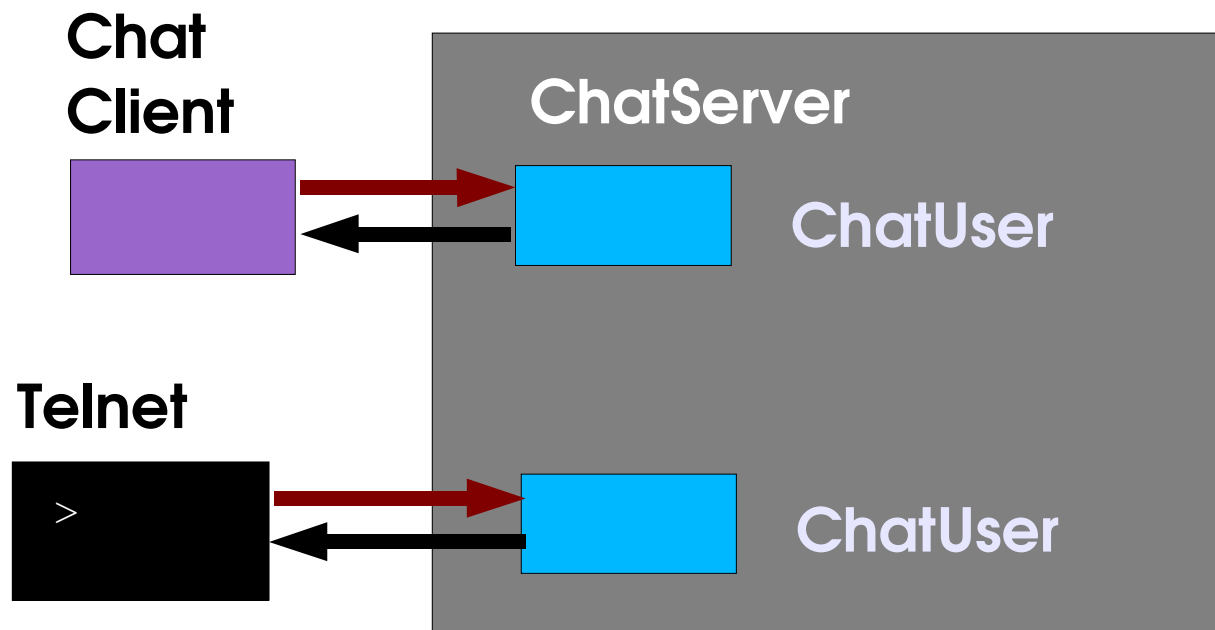
Jabber-Comets-Integration

2. Comets-Protokoll

Auch Comets verwendet Sockets, die während der gesamten Sitzung geöffnet bleiben. Momentan ist das Protokoll nur zur Hälfte (nur Antworten vom Server) in XML realisiert.

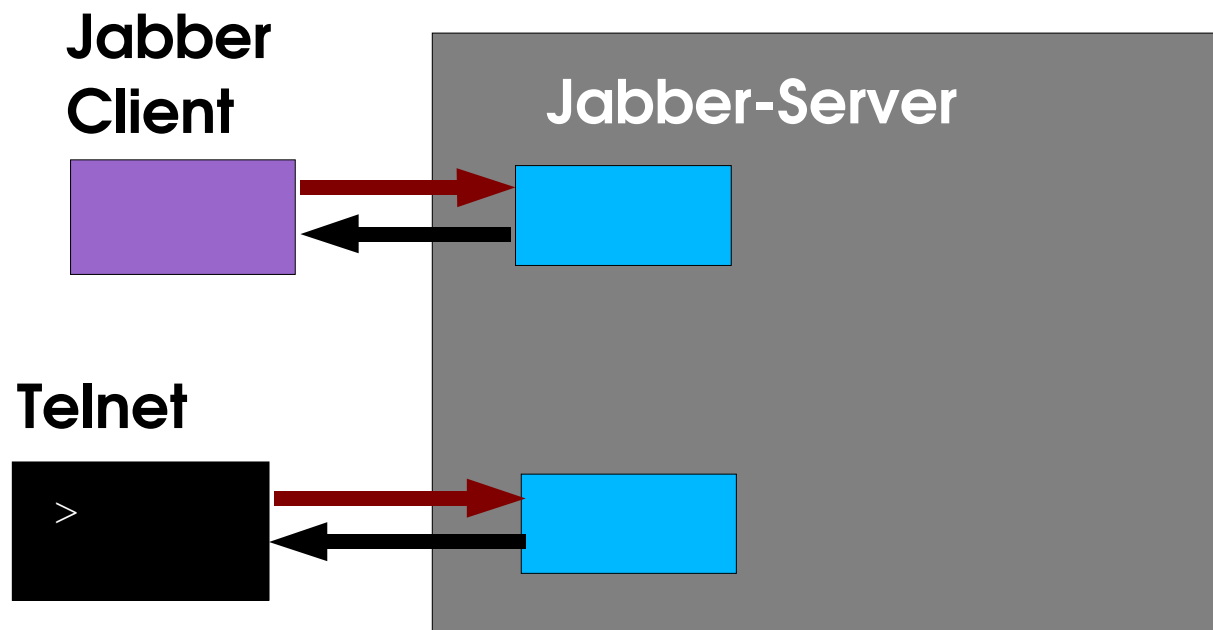
Jabber-Comets-Integration

2. Comets-Protokoll



Jabber-Comets-Integration

2. Comets-Protokoll



Jabber-Comets-Integration

2. Comets-Protokoll

Ein Login fehlt. Es werden Netzwerkverbindungen aufgebaut ohne eine Autorisierung.

Nachrichten sendet man, indem einfach nur die Nachricht an sich über die Verbindung gesendet wird. Der Server erkennt nur an der Verbindung, wer die Nachricht gesendet hat (Jabber macht das genauso).

Jabber-Comets-Integration

2. Comets-Protokoll

Nachrichten empfangen:

```
<message>
```

```
<sender>
```

```
flavorflav
```

```
</sender>
```

```
<body>
```

```
Don't believe the hype!
```

```
</body>
```

```
</message>
```

Jabber-Comets-Integration

2. Comets-Protokoll

Es existieren wichtige also Parallelitäten zwischen Comets und Jabber:

1. Sockets mit dauerhaften Verbindungen
 2. Client-Server-Architektur
 3. Protokoll als roher Text (UTF-8) verschickt im Vergleich zu Serialisierung oder DataStreams in Java.
-

Jabber-Comets-Integration

2. Comets-Protokoll

Jabber und Comets sind sich sowohl technisch als auch thematisch sehr ähnlich.

Eine Verknüpfung wäre also möglich und eventuell auch sinnvoll.

Jabber-Comets-Integration

1. Jabber-Protokoll

2. Comets-Protokoll

3. Jabber und Comets - Ein gutes Duo?

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

Ein erster Ansatzpunkt der Integration wäre auch eine Komplettierung und Anpassung des Comets-Protokolls an das Jabber-Format.

Vorteil: Anlehnung an einen Standard

Nachteil: Wenn man XMPP nicht kennt, wird das Protokoll etwas komplexer.

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

Was kann Jabber, das Comets nicht kann?

- private Nachrichten/Chats
 - Dateiübertragungen
 - Authentifizierung (Login)
 - Presence (selbst definierbar)
-

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

Was kann Comets, das Jabber nicht kann?

Comets hat ein innovatives Presence-Konzept, das Raum (Bewegungen auf Internet-Seiten) und Zeit (online/offline) verbindet.

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

Es sind unterschiedliche Einsatzszenarien vorstellbar.

Nach Server-Location:

1. Comets soll mit jabber.org kommunizieren können
 2. Comets soll mit einem eigenen jabberd kommunizieren
-

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

Nach Integrationsgrad:

1. Comets als vollwertiger Jabber-Client
 2. Comets integriert nur spezielle Features;
parallel wird ein zusätzlicher Jabber-Client eingesetzt.
-

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

Jabber als öffentlicher Dienst (jabber.org):

Problem:

Comets-Applet kann nicht direkt mit jabber.org kommunizieren. Sicherheitseinstellungen verhindern einen Verbindungsaufbau mit fremden Hosts.

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

Eine mögliche Lösung wären signierte Applets mit veränderten Sicherheitseinstellungen.

Dies erhöht jedoch die Anforderungen an den Nutzer und die Abhängigkeit von bestimmten Virtual Machines.

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

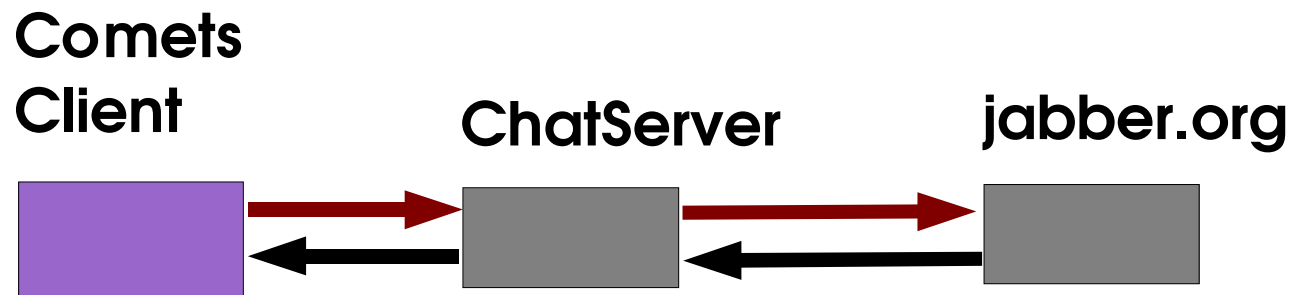
Besser geeignet ist vielleicht ein „Durchschleifen“ der Kommunikation mit jabber.org durch den Comets-Server.

Die Erweiterung wäre am Server sehr einfach zu realisieren und alle Jabber-spezifischen Features müssten nur vom Client (Applet) implementiert werden.

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

„Durchschleifen“



Jabber-Comets-Integration

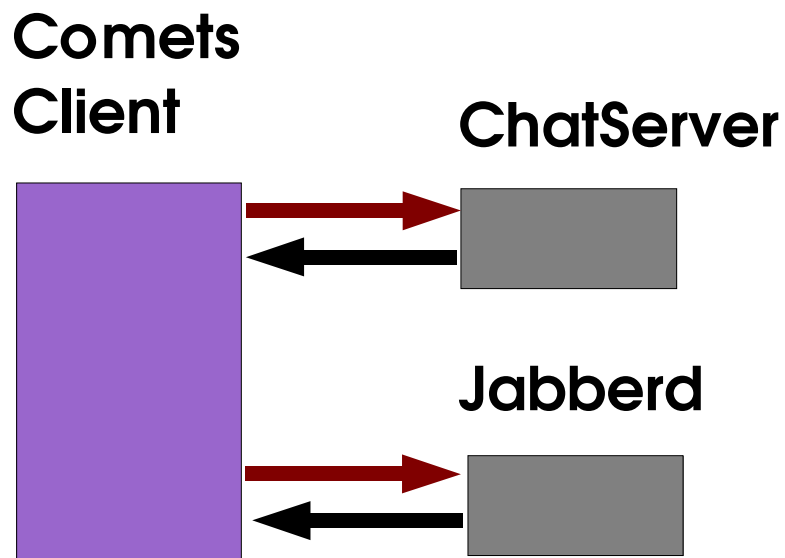
3. Jabber und Comets - Ein gutes Duo?

Einen Jabber-Server kann auch auf dem Rechner des Comets-Servers installiert werden. Projektgruppen haben so neben Comets auch eine eigene IM-Plattform. Der Comets-Client wird dann um einige Jabber-Funktionen erweitert.

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

Lokaler jabberd / CometsClient = JabberClient



Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

Comets könnte bei einer lokalen Installation die Passwort-Dateien von jabberd verwenden, um eine eigene Login-Prozedur zu implementieren.

Jabber-Comets-Integration

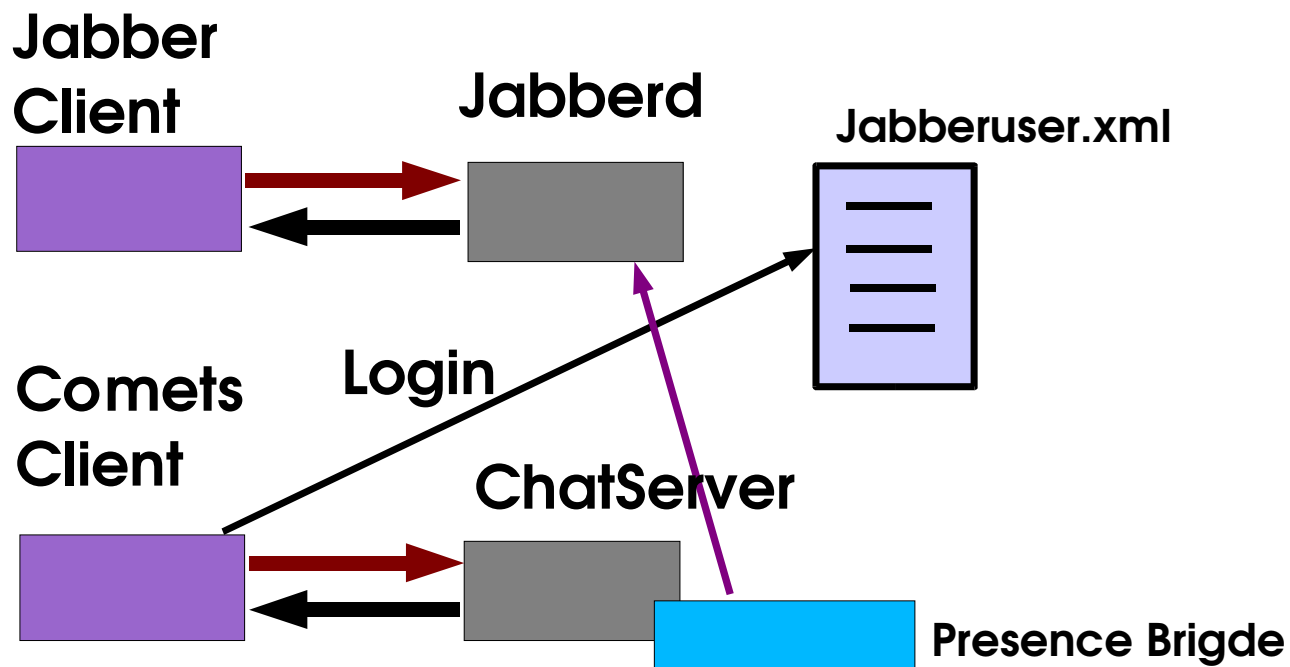
3. Jabber und Comets - Ein gutes Duo?

Ferner könnte von Comets nur die Jabber-Presence integriert werden. D.h. Comets würde nur anzeigen wie der aktuelle Jabber-Status der einzelnen Receiver aussieht.

Jabber-Comets-Integration

3. Jabber und Comets - Ein gutes Duo?

Lokaler jabberd/ CometsClient + JabberClient



Jabber-Comets-Integration

Vielen Dank

Fragen und Diskussion

Jabber-Comets-Integration

Vorschläge zur Diskussion

Soll Comets von Jabber unabhängig bleiben?

Wieviel Jabber braucht Comets?

Überfordern wir evtl. die Anwender?

Wie gross darf der Comets-Client werden?
