

# XMPP: Extensible Messaging and Presence Protocol (aka Jabber)

Lukas Kolbe  
<lukas@knup.de>

5. Dezember 2005

## Einleitung

Was ist XMPP?

## Architektur

Allgemeines

Kommunikation via XMPP: Streams, Stanzas

## Beispielanwendung

"Kommunikation eines Objekterkennungssystems"

## Implementierungen

Server & Clients

## Referenzen

## Zitat aus RFC 3920:

*XMPP is a protocol for streaming XML elements in order to exchange structured information in close to real time between any two network endpoints.*



## ... Protokoll?

### XMPP spezifiziert

- ▶ die Syntax
- ▶ die Semantik
- ▶ zu großen Teilen "best practices"
- ▶ wie Nachrichten von Client A nach Client B kommen
- ▶ Format der Nachrichten (XML)



” ... protocol for streaming XML ... ”

- ▶ XML als Basis der Kommunikation
- ▶ Fokus liegt auf aktuellem Element, nicht dem Dokument

” ... change structured information ... ”

- ▶ z.B. Chatnachrichten, Verfügbarkeitsinformationen
- ▶ Datenaustausch intelligenter Systeme
- ▶ oder SOAP
- ▶ oder RSS oder XML-RPC ...

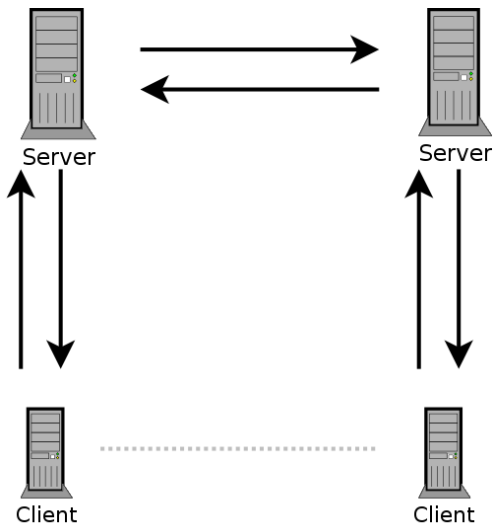


”... close to real time between any two network endpoints.”

- ▶ keine direkte Verbindung zwischen Clients
- ▶ Server vermitteln
- ▶ Austausch zwischen beliebigen Clients über Server
- ▶ daher ”fast-Echtzeit”



## Allgemeines



# Dokument-zentrisch – Element-zentrisch

## XML Dokument ...

- ▶ als Ganzes zu verarbeiten
- ▶ grÖÙe bekannt
- ▶ kann geöffnert, verarbeitet und gespeichert werden.

## ... und XML Stream

- ▶ einzelne Elemente sind Informationsträger
- ▶ XML Namespaces definieren Bedeutung der Elemente
- ▶ Dokument vollständig erst mit **Ende** des Streams



# XML Stanza

- ▶ deutsch: etwa "Absatz", "Strophe"
- ▶ **ein** XML-Element inkl. Kind-Nodes
- ▶ grundlegende Verarbeitungseinheit in XML Streams
- ▶ sind atomar



## "Kommunikation eines Objekterkennungssystems"

- ▶ 2 fiktive Agenten
- ▶ Agent 1, in Bielefeld
  - ▶ erkennt Objekte auf Bildern
  - ▶ schickt die Koordinaten zwecks Speicherung und Veröffentlichung zum Server
- ▶ Agent 2, in Münster
  - ▶ interessiert sich für Position von Objekten
  - ▶ wird vom Server benachrichtigt, sobald es neues gibt
  - ▶ bearbeitet die Daten lokal weiter ...
- ▶ Realisierung durch eine Erweiterung von XMPP ("Publish-Subscribe")



## "Kommunikation eines Objekterkennungssystems"

### Initialisierung des Streams: Client Sendet

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  version='1.0'
  to='techfak.uni-bielefeld.de'>
```

Server antwortet



## "Kommunikation eines Objekterkennungssystems"

### Initialisierung des Streams: Client Sendet

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  version='1.0'
  to='techfak.uni-bielefeld.de'>
```

Server antwortet



## "Kommunikation eines Objekterkennungssystems"

### Initialisierung des Streams: Client Sendet

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  version='1.0'
  to='techfak.uni-bielefeld.de'>
```

Server antwortet



## "Kommunikation eines Objekterkennungssystems"

### Initialisierung des Streams: Client Sendet

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  version='1.0'
  to='techfak.uni-bielefeld.de'>
```

### Server antwortet

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  version='1.0'
  id='stream1'
  from='techfak.uni-bielefeld.de'>
```



## Initialisierung des Streams: Client Sendet

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  version='1.0'
  to='techfak.uni-bielefeld.de'>
```

## Server antwortet

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  version='1.0'
  id='stream1'
  from='techfak.uni-bielefeld.de'>
```



## Initialisierung des Streams: Client Sendet

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  version='1.0'
  to='techfak.uni-bielefeld.de'>
```

## Server antwortet

```
<stream:stream
  xmlns='jabber:client'
  xmlns:stream='http://etherx.jabber.org/streams'
  version='1.0'
  id='stream1'
  from='techfak.uni-bielefeld.de'>
```

## Agent 1 in Bielefeld veröffentlicht eine Objektposition

```
<iq type="set" id="publish1"
  from="agent1@techfak.uni-bielefeld.de"
  to="pubsub.techfak.uni-bielefeld.de"
  <pubsub xmlns="http://jabber.org/protocol/pubsub">
    <publish node="generic/positions">
      <item>
        <position xmlns="http://techfak.../xml-isy/positions"
          source="http://techfak.../xml-isy/bild1.png">
          <shape type="rect"
            x="23" y="42" width="128" height="256" />
        </position>
      </item>
    </publish>
  </pubsub>
</iq>
```

## Agent 1 in Bielefeld veröffentlicht eine Objektposition

```
<iq type="set" id="publish1"
  from="agent1@techfak.uni-bielefeld.de"
  to="pubsub.techfak.uni-bielefeld.de"
  <pubsub xmlns="http://jabber.org/protocol/pubsub">
    <publish node="generic/positions">
      <item>
        <position xmlns="http://techfak.../xml-isy/positions"
          source="http://techfak.../xml-isy/bild1.png">
          <shape type="rect"
            x="23" y="42" width="128" height="256" />
        </position>
      </item>
    </publish>
  </pubsub>
</iq>
```

## Agent 1 in Bielefeld veröffentlicht eine Objektposition

```
<iq type="set" id="publish1"
  from="agent1@techfak.uni-bielefeld.de"
  to="pubsub.techfak.uni-bielefeld.de"
  <pubsub xmlns="http://jabber.org/protocol/pubsub">
    <publish node="generic/positions">
      <item>
        <position xmlns="http://techfak.../xml-isy/positions"
          source="http://techfak.../xml-isy/bild1.png">
          <shape type="rect"
            x="23" y="42" width="128" height="256" />
        </position>
      </item>
    </publish>
  </pubsub>
</iq>
```



## "Kommunikation eines Objekterkennungssystems"

### Antwort des Servers an Agent 1

```
<iq type="result"
  from="pubsub.techfak.uni-bielefeld.de"
  to="agent1@techfak.uni-bielefeld.de"
  id="publish1">
  <item id="item1" />
</iq>
```

## Agent 2 in Münster bekommt von *pubsub.techfak...*

```
<message to="agent2@ki.uni-muenster.de"
  from="pubsub.techfak.uni-bielefeld.de">
  <event xmlns="http://jabber.org/protocol/pubsub#event">
    <items node="generic/positions">
      <item id="item1">
        <position xmlns="http://techfak.../xml-isy/positions"
          source="http://techfak.../xml-isy/bild1.png">
          <shape type="rect"
            x="23" y="42" width="128" height="256" />
        </position>
      </item>
    </items>
  </event>
</message>
```

## Agent 2 in Münster bekommt von *pubsub.techfak...*

```
<message to="agent2@ki.uni-muenster.de"
  from="pubsub.techfak.uni-bielefeld.de">
  <event xmlns="http://jabber.org/protocol/pubsub#event">
    <items node="generic/positions">
      <item id="item1">
        <position xmlns="http://techfak.../xml-isy/positions"
          source="http://techfak.../xml-isy/bild1.png">
          <shape type="rect"
            x="23" y="42" width="128" height="256" />
        </position>
      </item>
    </items>
  </event>
</message>
```



## "Kommunikation eines Objekterkennungssystems"

# Offene Fragen? Verständnisprobleme?



## Server

- ▶ 15 Implementationen
- ▶ 9 frei, 6 kommerziell
- ▶ keiner 100% Standard-Konform

## Clients

- ▶ ca. 80
- ▶ vom Kommandozeilenprogramm über Handy-Programme bis zum Windows-GUI
- ▶ ... und das sind nur die Instant Messenger!



## Bibliotheken zur Server- und Client-Implementierung

- ▶ für fast jede Programmierumgebung
- ▶ in fast jedem Abstraktionsgrad
- ▶ von J2ME für mobile Geräte
- ▶ über C, C++
- ▶ und Java, .NET
- ▶ bis Python, Ruby, Perl, PHP

- ▶ Spezifikationen:  
<http://www.jabber.org/protocol/>
- ▶ Publish-Subscribe (aus dem Beispiel):  
<http://www.jabber.org/jeps/jep-0060.html>
- ▶ Server/Clients:  
<http://www.jabber.org/software/>

# Danke für Eure Aufmerksamkeit!

# Diskussion

- ▶ Unterschied XMPP und Webservices
- ▶ Yet Another Middleware?
- ▶ Es gibt 165 "Jabber Enhancement Proposals". Zuviel Bürokratie?