Jabber für Fortgeschrittene XMPP-Server für Anfänger*innen

Mathias Ertl (jabber.at, jabber.fsinf.at)
2014-04-26

Disclamer: Every technology sucks :-)

Jabber hat Probleme, auf die in dieser Präsentation auch eingangen wird. Aber das hat jede andere Technologie auch – kein Grund also, Jabber nicht zu verwenden!

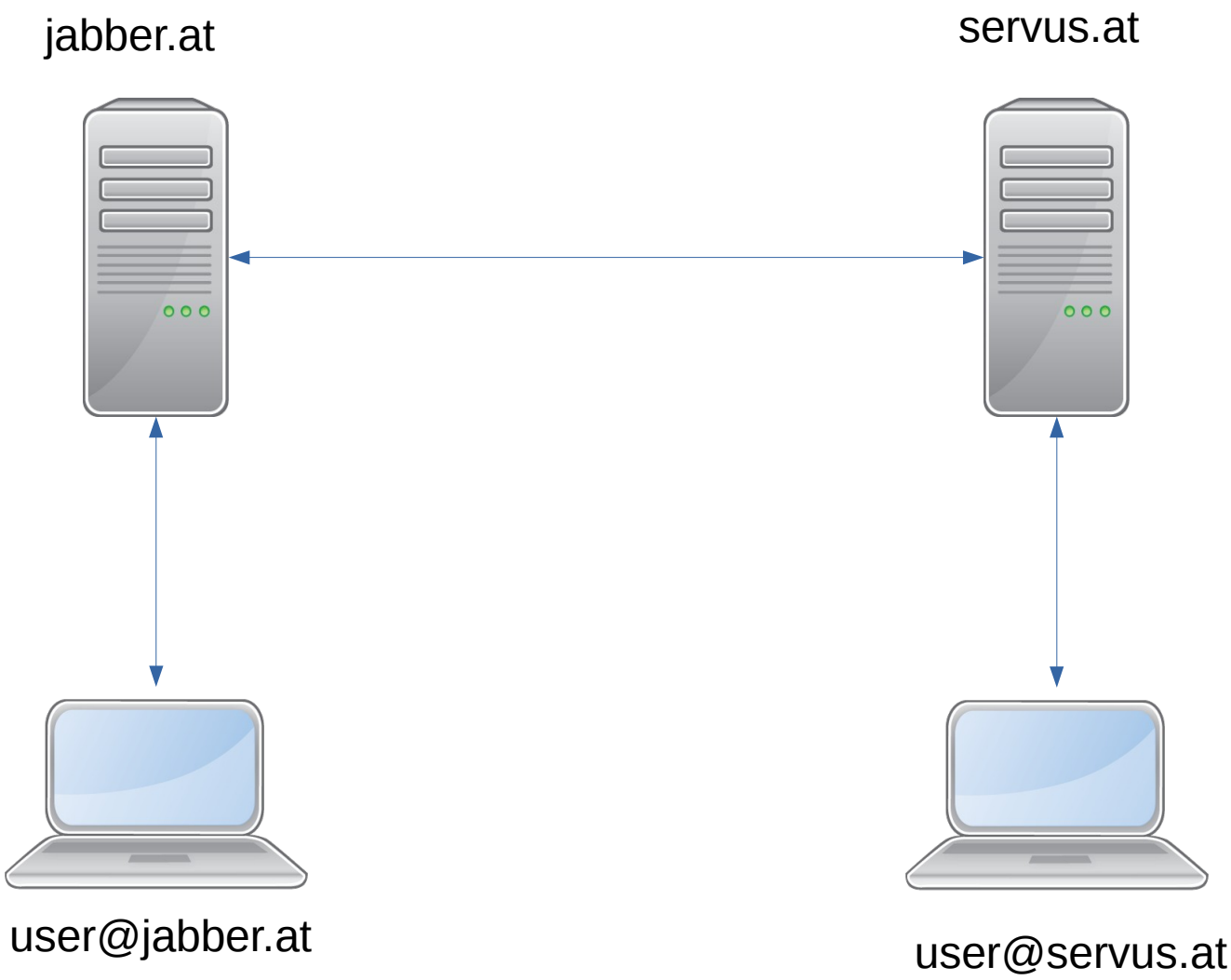
Eine kurze Geschichte

- 1999/2000: Erste Entwicklung [1]
- 2002: Umbenennung in „XMPP”
- 2004: Erster offizieller Standard (RFC 3920)
- 2005: Google Talk (inkl. Jingle)
- 2009: GMX Multimessenger
- 2010: Facebook Chat
- 2011: Aktualisierter Standard (RFC6120 bis RFC6122)

XMPP: Übersicht

- Offener IETF Standard
- Entwickelt von der XMPP Standards Foundation („XSF“)
- Baut auf etablierten Internet-Standards auf
- Traditionelles Client-to-Server Protokoll („c2s“)
- Optional: Server-to-Server „Federation“ („s2s“)
- Erweiterbar („XEPs”)

Verbindungsübersicht



XEPs

- ~340 „XMPP Enhancement Proposal“
- 13 Final, 35 Active, 67 Draft, 39 Experimental
  - File Transfer
  - Voice/Video Chat
  - Publish/Subscribe
  - Multi User Chat
  - Service Discovery
  - ...

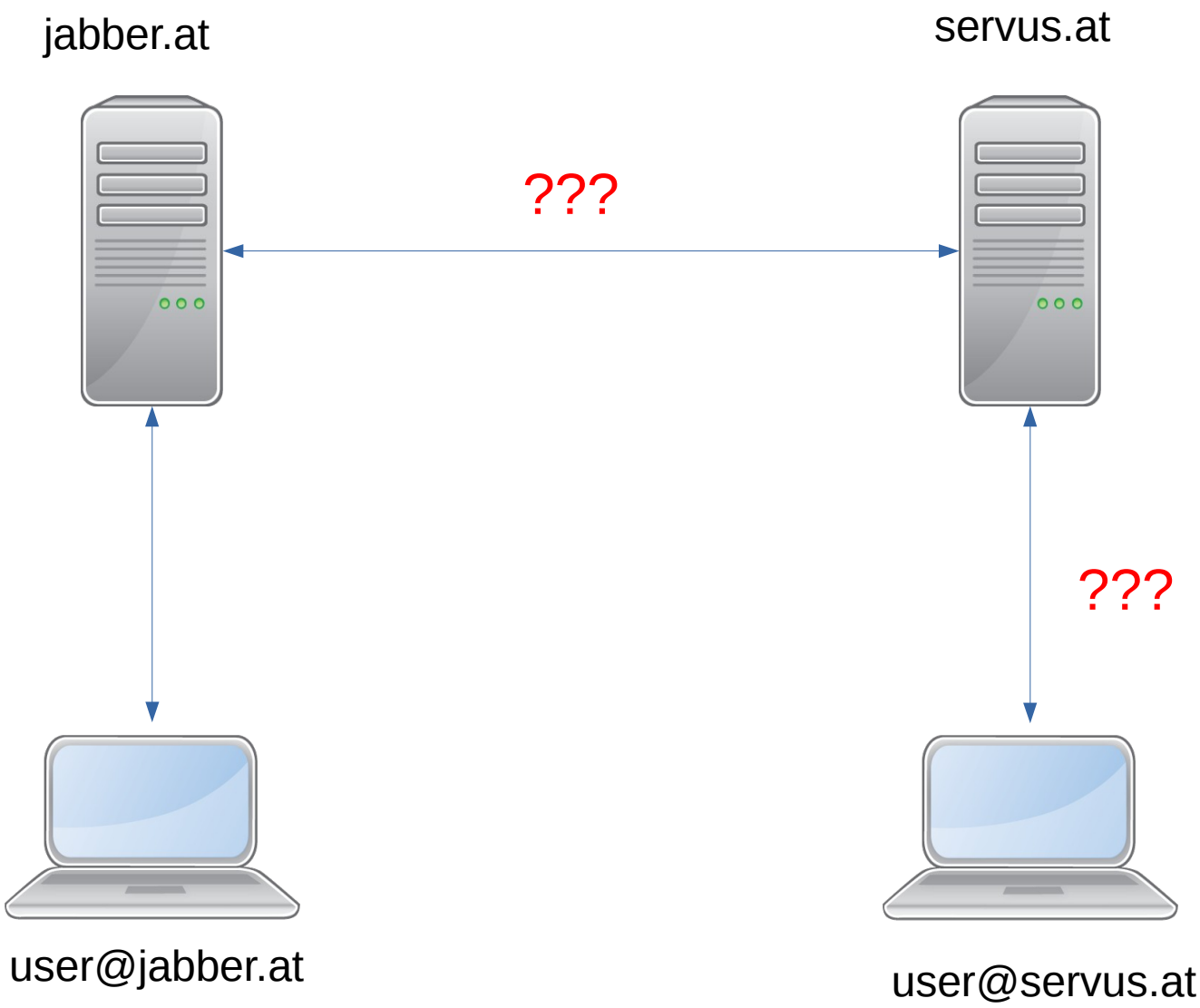
Standards

- XML als Datenformat
  - Hard for Humas and Machines to read ;-)
- DNS SRV-Records
  - Gibt Host und Ports für Verbindungen vor
  - Fallback: Port 5222/5269 auf A/AAAA-Record
  - Port 5222 (c2s) und 5269 (s2s) meistens verwendet

Standards: Verschlüsselung

- TLS für Verschlüsselung
  - Optional für Client-to-Server
    - Typischerweise strenge Zertifikatsprüfung
  - Verpflichtend für Server-to-Server
    - „DNS Dialback“ oder Prüfung des Zertifikats
  - Problem: Benutzer*in sieht immer nur die eigene Verbindung

Verbindungsübersicht



Standards: Authentifizierung

- DIGEST-MD5:
  - Passwort geht nicht über die Leitung
  - Dafür muss es im Klartext am Server gespeichert sein [2]!
  - Immer noch weit verbreitet :-(
- PLAIN
  - Passwort im Klartext über die Leitung.
  - Am Server: Hash oder Klartext
- SCRAM-SHA-1:
  - Sicherlich die beste Lösung
  - Passwort geht nicht über die Leitung
  - Passwort am Server als Hash gespeichert
  - Wenig Unterstützung (aber weiterhin Plain möglich)

XMPP: Grundsätzliche Probleme

- Langsame Protokoll-Entwicklung
- Sehr unterschiedlich weit entwickelte Clients
- Relativ alt (1999!):
  - Viele “Altlasten” (z.B. nur optionales TLS, oft im Klartext gespeicherte Passwörter!)
  - Konzeptionelle Schwächen (z.B. Mobile)

Server Software: “ferner liefen”

- jabberd
- jabberd2
- iChat Server (basiert auf jabberd2)
- Tigase
- Openfire
- Isode M-Link (proprietär, auf jabber.org)

Server Software: ejabberd

- Geschrieben in Erlang (“Ericsson Telecom Language”)
- Datenbank: Mnesia, MySQL, PostgreSQL
- Pros:
  - Sehr ausgereift, am weitesten verbreitet
  - Viele große Server (jabber.ccc.de, jabber.at, …)
  - Große Anzahl unterstützter XEPs[3]
- Cons:
  - Aufwendig zu Konfigurieren, Warten, Debuggen
  - Freier Software, aber kommerzielle “Business Edition”
  - Erlang: Sehr exotisch, aufwendige Entwicklung

Server Software: Prosody

- Geschrieben in Lua
- Datenbank: Files, SQLite3, MySQL, PostgreSQL
- Pros:
  - Dynamische Entwicklung und Community
  - Einfachere Wartung
  - Viele neue Server
  - Unterstützt fast so viele XEPs wie ejabberd, obwohl viel jünger[4]
- Cons:
  - Lua ist fast genauso exotisch wie Erlang

Transports

- Transports sind Gateways in andere Netzwerke, z.B. ICQ, MSN, …
- Spectrum2
  - Auf jabber.at in Verwendung
  - Nicht mehr aktiv entwickelt
- Py(ICQ|MSN|...)t
  - Nicht mehr weiterentwicklet
  - Letzter Änderung PyICQt: 2009-09-21
- ejabberd IRC transport

Server-Admin in spe: Bedenke...

- Kosten: Domain, Zertifikat, Server(-Betrieb)
- Dauer: Sollte länger betrieben werden
- Arbeit:
  - Server-Administration
  - End-Benutzer*innen-Support
- Verantwortung:
  - Datenschutz
  - Rechtliche Konsequenzen

Exkurs: TLS im Jabber-Netzwerk

- August 2013: 3-Teiliger Blogpost „The State of TLS on XMPP“ [5][6][7] testet 100 XMPP-Server.
  - 43x invalide Zertifikate (inkl. CAcert)
  - 14x SSLv2 Support (sehr unsicher)
  - Rangliste (von 100, max. Score = 100):

Rang	Score	Server:
1-3	93	jabber.at, lightwitch.org, macjabber.de
4-12	90	Jappix.com, jabber.no, … - Etwas ältere ejabberd-Version
	~70-75	30 Server mit altem Debian/Ubuntu UND altem ejabberd!

    → **Alte Server sind verantwortungslos!**
  - Siehe auch: https://jabber.at/how-good-tls-encryption

Exkurs: Rechtliche Konsequenzen

- Disclaimer: Ich bin kein Rechtsanwalt
- Vorratsdatenspeicherung:
  - Sie ist de facto Geschichte \o/
  - Gilt nur für kommerzielle Services
- Bei laufenden Ermittlungen:
  - Staatsanwaltschaft kann mit richterlichen Beschluss Auskunft über Daten einzelner Benutzer*innen anfordern.
  - Kooperationspflicht – im Extremfall Beugehaft!

Tägliche Arbeit

- ~25 Support-Mails seit 2014-01-01
- Homepage warten
- Updates einspielen (Heartbleed!)
- Auf Abstürze (sehr schnell!) reagieren
- SPAM-Bekämpfung
- DDOS-Attacken
- Operators-Mailingliste[8]

Registrierung

- In-Band Registration („IBR“) direkt im Client
  - Intuitiv – was Benutzer erwarten?
  - Durch CAPTCHA absicherbar
  - CAPTCHA bringt Usability-Probleme
- Via Web
  - Prosody und ejabberd bieten eigene Module dafür
  - 3rd-Party Software (z.B. https://account.jabber.at)
- Auf jabber.at in den letzten 31 Tagen:
  - 273x In-Band
  - 492x Web-Interface

SPAM und (D)DOS

- SPAM/DDOS wurden LANGE von der Community ignoriert:
  - „However, XMPP does not need to be perfect. You don't need to be the fastest antelope in the herd to avoid being eaten by the lion, you just need to be faster than the slow antelope who get caught.“
    - Peter Saint-Andre, 2008 [9]
  - Server haben noch immer in vielerlei Hinsicht zu wenige Spam-Schutz Mechanismen (z.B. in MUCs)
  - SPAM im gesamten Netzwerk ein Problem

Firewalls (Am Server)

- Port 5222 manchmal von Firewalls gefiltert
  - SRV-Record könnte auf Port 80 oder 443 zeigen
- Proxy via HTTP (auch für Webclients):
  - XMPP over BOSH (XEP-0206)
  - Websockets – In Entwicklung

Firewalls (am Client)

- Direkte Kommunikation zwischen Clients notwendig (File Transfer, Voice/Video, ...)
- Großes Problem: NATs
  - XEP-0279: Server IP Check
  - STUN („Session Traversal Utilities for NAT“)
  - TURN („Traversal Using Relays around NAT“)
  - Fragmentierter Support in Servern und Clients

    → Schlechte Kombination. Macht oft Probleme!

References

[1] http://xmpp.org/about-xmpp/history/

[2] http://en.wikipedia.org/wiki/Digest_access_authentication

[3] http://www.process-one.net/en/ejabberd/protocols/

[4] https://prosody.im/doc/xeplist

[5] https://blog.thijsalkema.de/blog/2013/08/26/the-state-of-tls-on-xmpp-1/

[6] https://blog.thijsalkema.de/blog/2013/08/28/the-state-of-tls-on-xmpp-2/

[7] https://blog.thijsalkema.de/blog/2013/09/02/the-state-of-tls-on-xmpp-3/

[8] http://mail.jabber.org/mailman/listinfo/operators

[9] http://mail.jabber.org/pipermail/juser/2008-August/006555.html