

Installatie en bediening van ejabberd: Kerstmiseditie

Alexey Shchepin
<mailto:alexey@sevcom.net>
<xmpp:aleksey@jabber.ru>

vertaling: Sander Devrieze
<mailto:s.devrieze@pandora.be>
<xmpp:sander@l4l.be>

23 december, 2004



Inhoudsopgave

1	Inleiding	3
2	Installatie vanaf de broncode	4
2.1	Installatievereisten	4
2.1.1	Unix-achtigen	4
2.1.2	Windows	4
2.2	Downloaden	5
2.3	Compilatie	5
2.3.1	Unix-achtigen	5
2.3.2	Windows	5
2.4	Opstarten	6
3	Configuratie	7
3.1	Initiële configuratie	7
3.1.1	Hostnaam	7
3.1.2	Standaardtaal	8
3.1.3	Toegangsregels	8
3.1.4	Configuratie van shapers	10
3.1.5	Listened sockets	10
3.1.6	Modules	13
3.2	Configuratie en toezicht houden op een draaiende ejabberd server	15
3.2.1	Webgebaseerde beheerdersinterface	15
3.2.2	Via de commandoregel met <code>ejabberdctl</code>	16
4	Clustering	17
4.1	Werking	17
4.1.1	Router	17
4.1.2	Local router	17
4.1.3	Session manager	17

4.1.4	S2S manager	18
4.2	Een cluster opzetten	18
A	Ingebouwde modules	19
A.1	Algemene opties	19
A.1.1	iqdisc	19
A.1.2	host	20
A.2	mod_announce	20
A.3	mod_configure	21
A.4	mod_disco	21
A.5	mod_echo	22
A.6	mod_irc	22
A.7	mod_last	23
A.8	mod_muc	23
A.9	mod_offline	23
A.10	mod_privacy	24
A.11	mod_private	24
A.12	mod_pubsub	24
A.13	mod_register	25
A.14	mod_roster	25
A.15	mod_service_log	25
A.16	mod_stats	26
A.17	mod_time	26
A.18	mod_vcard	26
A.19	mod_version	27
B	I18n/L10n	27

1 Inleiding

ejabberd is een vrije (GPL) gedistributeerde fault-tolerance Jabberserver en is hoofdzakelijk geschreven in Erlang.

De meest in het oog springende mogelijkheden van **ejabberd** zijn:

- **Multiplatform:** **ejabberd** draait onder zowat alle populaire platformen: van unix afgeleide systemen (getest onder Linux, FreeBSD en NetBSD) en Windows NT/2000/XP.
- **Gedistribueerd:** **ejabberd** kan op een reeks computers gedraaid worden die samen Jabber op één domein aanbieden. Goedkope computers kunnen gebruikt worden om de capaciteit van de cluster uit te breiden.
- **Fault-tolerance:** met **ejabberd** kan een cluster opgezet worden zodat alle gegevens die nodig zijn voor een goed werkende Jabberserver steeds beschikbaar zijn op meerdere nodes. Dit betekent dat indien één node defect raakt, dat de anderen dan zonder onderbrekingen verder blijven draaien. Daarnaast is het ook mogelijk om nodes “on the fly” te vervangen.
- **Modulair:** elke module kan geladen en gestopt worden zonder **ejabberd** te moeten herstarten. Er zijn modules beschikbaar voor:
 - Multi-User Chat¹.
 - Een IRC-transport.
 - Publish-Subscribe².
 - Een Jabbergebruikersdatabase gebaseerd op de vCards van gebruikers.
 - Een webgebaseerde beheerdersinterface.
 - HTTP Polling³
 - Service Discovery⁴.
 - Statistieken via JEP-0039⁵ (Statistics Gathering).
 - En nog veel meer.
- **Lokalisatie:**
 - Ondersteuning voor `xml:lang`.
 - Ondersteuning voor IDNA⁶.
 - **ejabberd** is vertaald in zeven talen.
- **Gemakkelijke installatie:** omdat **ejabberd** gebouwd is bovenop Erlang/OTP hoeft u geen externe database, geen externe webserver,... te installeren omdat alles reeds voor handen is en direct werkt.
- **Ondersteuning voor STARTTLS, SASL en SSL.**

¹<http://www.jabber.org/jeps/jep-0045.html>

²<http://www.jabber.org/jeps/jep-0060.html>

³<http://www.jabber.org/jeps/jep-0025.html>

⁴<http://www.jabber.org/jeps/jep-0030.html>

⁵<http://www.jabber.org/jeps/jep-0039.html>

⁶<http://www.ietf.org/rfc/rfc3490.txt>

- Ondersteuning voor authenticatie via LDAP.
- Mogelijkheid tot gebruik van externe componenten zoals Jabber ICQ Transport⁷, Jabber Yahoo! Transport⁸, PyMSNt⁹, pyAIM-t¹⁰, pyICQ-t¹¹,...
- Migratie van Jabberd14 is mogelijk.
- Bijna volledig XMPP-complaint.
- IPv6 mogelijk bij zowel C2S- als S2S-verbindingen.

De minpunten van ejabberd zijn:

- Geen ondersteuning voor virtuele domeinnamen.
- Geen ondersteuning voor authenticatie en STARTTLS in S2S-verbindingen.

2 Installatie vanaf de broncode

2.1 Installatievereisten

2.1.1 Unix-achtigen

Om ejabberd te compileren hebt u volgende pakketten nodig:

- GNU Make;
- GCC;
- libexpat 1.95 of recenter;
- Erlang/OTP R8B of recenter;
- OpenSSL 0.9.6 of recenter (optioneel).

2.1.2 Windows

Om ejabberd te compileren onder MS Windows hebt u volgende pakketten nodig:

- MS Visual C++ 6.0 Compiler
- Erlang/OTP R10B-1a¹²

⁷<http://jit.jabberstudio.org/>

⁸<http://yahoo-transport-2.jabberstudio.org/>

⁹<http://msn-transport.jabberstudio.org/>

¹⁰<http://pyaim-t.jabberstudio.org/>

¹¹<http://pyicq-t.jabberstudio.org/>

¹²http://erlang.org/download/otp_w_{in}32_R10B-1a.exe

- Expat 1.95.7¹³
- Iconv 1.9.1¹⁴ (optioneel)
- Shining Light OpenSSL¹⁵ (nodig voor TLS en STARTTLS)

2.2 Downloaden

Een stabiele versie van ejabberd kan gedownload worden vanaf <http://www.jabberstudio.org/projects/ejabberd/releases/>.

De recentste alpha-versie kunt u steeds via CVS bekomen met volgende commando's:

```
export CVSROOT=:pserver:anonymous@jabberstudio.org:/home/cvs
cvs login
<druk op Enter wanneer men naar een wachtwoord vraagt>
cvs -z3 co ejabberd
```

2.3 Compilatie

2.3.1 Unix-achtigen

```
./configure
make
su
make install
```

Door bovenstaande commando's uit te voeren zal ejabberd geïnstalleerd worden in de map /var/lib/ejabberd, ejabberd.cfg in de map /etc/ejabberd en zal de map /var/log/ejabberd aangemaakt worden voor de logbestanden.

2.3.2 Windows

- Installeer Erlang emulator (bijvoorbeeld in C:\Program Files\erl5.3).
- Installeer de Expat-bibliotheek in de map C:\Program Files\Expat-1.95.7.
Kopieer C:\Program Files\Expat-1.95.7\Libs\libexpat.dll naar de systeemmap van Windows (C:\WINNT of C:\WINNT\System32).

¹³http://prdownloads.sourceforge.net/expat/expat_win32bin_1.95.7.exe?download

¹⁴<http://ftp.gnu.org/pub/gnu/libiconv/libiconv-1.9.1.tar.gz>

¹⁵<http://www.slproweb.com/products/Win32OpenSSL.html>

- Compileer en installeer vervolgens de Iconv-bibliotheek in de map `C:\Program Files\iconv-1.9.1`.
Kopieer ook `C:\Program Files\iconv-1.9.1\bin\iconv.dll` naar de systeemmap.
Opmerking: in plaats van de bestanden `libexpat.dll` en `iconv.dll` te kopiëren naar de systeemmap kunt u ook de mappen `C:\Program Files\Expat-1.95.7\Libs` en `C:\Program Files\iconv-1.9.1` toevoegen aan de omgevingsvariabele `PATH`.
- Ga naar `ejabberd\src` en voer hiernavolgende opdrachten uit:

```
configure.bat
nmake -f Makefile.win32
```

- Bewerk `ejabberd\src\ejabberd.cfg` en voer volgend commando uit:

```
werl -s ejabberd -name ejabberd
```

2.4 Opstarten

Gebruik één van volgende commando's om `ejabberd` te starten:

```
erl -pa /var/lib/ejabberd/ebin -name ejabberd -s ejabberd
```

of

```
erl -pa /var/lib/ejabberd/ebin -sname ejabberd -s ejabberd
```

In het tweede geval zal de Erlang-node geïdentificeerd worden met alleen het eerste gedeelte van de hostnaam waardoor andere nodes buiten het domein er niet mee kunnen communiceren.

Merk op dat wanneer u één van bovenstaande commando's gebruikt, dat `ejabberd` dan in de huidige map zal zoeken naar het configuratiebestand en deze map ook zal gebruiken voor de opslag van de database en de logbestanden.

U kunt dit standaardgedrag omzeilen door de paden manueel op te geven:

```
erl -pa /var/lib/ejabberd/ebin \
    -sname ejabberd \
    -s ejabberd \
    -ejabberd config \"/etc/ejabberd/ejabberd.cfg\" \
    log_path \"/var/log/ejabberd/ejabberd.log\" \
    -sasl sasl_error_logger \{file,\"/var/log/ejabberd/sasl.log\"}\} \
    -mnesia dir \"/var/lib/ejabberd/spool\"
```

Nog andere handige opties kunt u terugvinden in de Erlang/OTP R10B documentatie¹⁶.

Om meer dan 1024 verbindingen te kunnen opzetten, moet u de omgevingsvariabele `ERL_MAX_PORTS` instellen:

¹⁶<http://www.erlang.se/doc/doc-5.4.3/doc/>

```
export ERL_MAX_PORTS=32000
```

Hou er rekening mee dat `ejabberd` hierdoor meer geheugen zal gebruiken (ongeveer 6MB meer).

Om het geheugengebruik te verminderen, kunt u de omgevingsvariabele `ERL_FULLSWEEP_AFTER` instellen:

```
export ERL_FULLSWEEP_AFTER=0
```

Merk op dat het hierdoor mogelijk kan zijn dat `ejabberd` vertraagt.

3 Configuratie

3.1 Initiële configuratie

De inhoud van het configuratiebestand wordt bij de eerste start van `ejabberd` naar een database overgezet. Nadien wordt de configuratie door `ejabberd` ingeladen vanaf die database en alle opties in het configuratiebestand worden toegevoegd aan de ingangen in de database. Het configuratiebestand is opgebouwd uit een reeks Erlang-uitdrukkingen. Elke regel die begint met het `'%'`-teken wordt genegeerd. Elke uitdrukking is een tuple, waar het eerste element de naam is van de optie en de andere de waarden zijn van de optie. Als het configuratiebestand bijvoorbeeld geen `"host"`-definitie bevat, dan zal de oude waarde uit de database worden gebruikt.

Om de oude waarden uit de database teniet te doen, kunnen volgende regels toegevoegd worden aan de configuratie:

```
override_global.  
override_local.  
override_acls.
```

Deze regels zullen respectievelijk de oude globale opties, de oude lokale opties of de oude ACL's verwijderen voordat er nieuwe worden toegevoegd.

3.1.1 Hostnaam

Met de optie `hostname` kunt u de domeinnaam die `ejabberd` zal moeten aanbieden, opgeven. Om bijvoorbeeld `jabber.org` te gebruiken moet u onderstaande regel opnemen in het configuratiebestand:

```
{host, "jabber.org"}.
```


3.1.2 Standaardtaal

De standaardtaal van berichten die `ejabberd` verzendt naar gebruikers kan gewijzigd worden met behulp van de optie `language`. De taal die standaard gebruikt wordt als de Jabber-client niet overweg kan met het `xml:lang`-attribuut en er daardoor geen opgeeft, is het Engels (taalcode: `'en'`). Het veranderen van de standaardtaal werkt enkel indien er een vertaling (`<taalcode>.msg`) aanwezig is in de map `msgs` van `ejabberd`. Als u bijvoorbeeld het Nederlands als standaardtaal wilt instellen, neem dan volgende regel op in het configuratiebestand:

```
{language, "nl"}.
```

3.1.3 Toegangsregels

Toegangscontrole doet `ejabberd` met Access Control Lists (ACL). Deze hebben volgende syntaxis:

```
{acl, <naamacl>, {<soortacl>, ...}}.
```

`<soortacl>` kan volgende waarden aannemen:

`all` Komt overeen met alle JID's. Voorbeeld:

```
{acl, all, all}.
```

`{user, <gebruikersnaam>}` Komt overeen met de lokale gebruiker `<gebruikersnaam>`. Voorbeeld:

```
{acl, admin, {user, "aleksey"}}.
```

`{user, <gebruikersnaam>, <server>}` Komt overeen met de gebruiker met als JID `<gebruikersnaam>@<server>` en al zijn bronnen. Voorbeeld:

```
{acl, admin, {user, "aleksey", "jabber.ru"}}.
```

`{server, <server>}` Komt overeen met elke JID van server `<server>`. Voorbeeld:

```
{acl, jabberorg, {server, "jabber.org"}}.
```

`{user_regex, <reguliere_expressie>}` Komt overeen met alle lokale gebruikers waarvan de naam voldoet aan de reguliere expressie `<reguliere_expressie>`. Voorbeeld:

```
{acl, tests, {user, "^test[0-9]*$"}}.
```

`{user_regex, <reguliere_expressie>, <server>}` Komt overeen met alle gebruikers waarvan de naam voldoet aan de reguliere expressie `<reguliere_expressie>` en die geregistreerd zijn op server `<server>`. Voorbeeld:

```
{acl, tests, {user, "^test", "localhost"}}.
```

{server_regexp, <reguliere_expressie>} Komt overeen met alle JID's op de servers waarvan de naam voldoet aan de reguliere expressie <reguliere_expressie>. Voorbeeld:

```
{acl, icq, {server, "^icq\\\\"}}.
```

{node_regexp, <reguliere_expressie_voor_gebruiker>, <reguliere_expressie_voor_server>} Komt overeen met alle gebruikers waarvan de naam voldoet aan de reguliere expressie <reguliere_expressie_voor_gebruiker> en die geregistreerd zijn op een servers waarvan de naam voldoet aan de reguliere expressie <reguliere_expressie_voor_server>. Voorbeeld:

```
{acl, aleksey, {node_regexp, "^aleksey$", "^jabber.(ru|org)$"}}.
```

```
{user_glob, <glob>}
```

```
{user_glob, <glob>, <server>}
```

```
{server_glob, <glob>}
```

{node_glob, <glob_voor_gebruiker>, <glob_voor_server>} Dit doet hetzelfde als hierboven maar dan met "shell glob patronen" in plaats van een reguliere expressie. Deze patronen kunnen volgende speciale tekens bevatten:

* Voldoet aan elke tekenreeks inclusief de lege tekenreeks (null string).

? Voldoet aan elk enkelvoudig teken.

[...] Voldoet aan één van de tekens tussen de haakjes. Reeksen tekens kunnen worden opgegeven door twee tekens te scheiden met een liggend streepje ('-'). Als het eerste teken na het eerste vierkante haakje ('[') een uitroepteken ('!') is, dan voldoet elk teken dat er niet tussen staat.

De volgende ACL's zijn van te voren gedefiniëerd:

all Voldoet aan alle JID's.

none Voldoet aan geen enkele JID.

Een ingang die de toegang tot een dienst regelt ziet er ongeveer zo uit:

```
{access, <dienst>, [{allow, <naamacl>},  
                    {deny, <naamacl>},  
                    ...  
                    ]}.
```

Tijdens de toegangscontrole van een JID tot de dienst <dienst>, zal **ejabberd** achtereenvolgens controleren of die JID voldoet aan (één van) de ACL('s) waarnaar verwezen wordt achter het eerste element van elke tuple van de lijst. Als dit het geval is, dan geeft dat het eerste element van de overeengekomen tuple als resultaat, in alle andere gevallen geeft dat "deny" (weigeren).

Voorbeeld:

```
{access, configure, [{allow, admin}]}.  
{access, iets, [{deny, badmans},  
                {allow, all}]}.
```

Volgende toegangsregels zijn van te voren gedefiniëerd:

`all` Geeft altijd “allow” (toelaten) als resultaat.

`none` Geeft altijd “deny” (weigeren) als resultaat.

3.1.4 Configuratie van shapers

Met shapers is het mogelijk om verbindingsverkeer te beperken. Shapers hebben in het configuratiebestand deze syntaxis:

```
{shaper, <naam_shaper>, <soort>}.
```

Momenteel is er slechts één soort shaper geïmplementeerd: `maxrate`. En die heeft volgende syntaxis:

```
{maxrate, <snelheid>}
```

Waar `<snelheid>` staat voor de maximale inkomende snelheid in bytes/seconde. Om bijvoorbeeld een shaper te definiëren met de naam “normaal” en als maximumsnelheid 1000 bytes/seconde, kunt u onderstaande regel toevoegen aan het configuratiebestand:

```
{shaper, normaal, {maxrate, 1000}}.
```

3.1.5 Listened sockets

De optie `listen` bevat een lijst met alle listened sockets en de diensten die erop worden aangeboden. Elk element van de lijst is een tuple die drie elementen bevat:

- Poortnummer;
- Module die instaat voor deze poort;
- Opties voor die module.

Momenteel zijn volgende modules aanwezig:

`ejabberd_c2s` Deze module beheert de C2S-verbindingen.

Hierbij zijn volgende opties mogelijk:

`{access, <toegangsregel>}` Met deze optie kunt u de toegang tot de opgegeven C2S-poort beperken. De standaardwaarde luidt “all” (iedereen).

`{shaper, <toegangsregel>}` Deze optie is gelijkaardig aan voorgaande maar gebruikt shapers in plaats van “allow” en “deny”. De standaardwaarde luidt: “none” (niets).

`{ip, IPNummer}` Met deze optie kunt u de te gebruiken netwerkinterface instellen. Bijvoorbeeld `{ip, {192, 168, 1, 1}}`.

`inet6` Zet ondersteuning voor IPv6 aan.

`starttls` Met deze optie maakt u STARTTLS beschikbaar voor verbindingen naar deze poort. Als u deze optie gebruikt, moet u ook de “certfile”-optie instellen.

`tls` Met deze optie wordt het verkeer op deze poort met SSL versleuteld vlak na het verbinden. Als u deze optie gebruikt moet u ook de “certfile”-optie instellen.

`ssl` Met deze optie wordt het verkeer op deze poort met SSL versleuteld. Als u deze optie gebruikt moet u ook de “certfile”-optie instellen. Het is aan te raden om in plaats hiervan de `tls`-optie te gebruiken.

`{certfile, Pad}` Pad naar het SSL-certificaat.

`ejabberd_s2s_in` Deze module beheert de inkomende S2S-verbindingen.

`ejabberd_service` Deze module beheert de verbindingen met Jabberdiensten (bv. diegenen die de `jabber:component:accept`-namespace gebruiken).

Volgende bijkomende opties kunnen ingesteld worden (naast de opties `access`, `shaper`, `ip` en `inet6` die ook hier bruikbaar zijn):

`{host, Hostnaam, [HostOpties]}` Met deze optie kunt u de hostnaam van de externe dienst opgeven en bijkomende opties daarvoor zoals `{password, Geheim}`.

`{hosts, [Hostnamen], [HostOpties]}` Idem als hierboven maar zo kunt u meerdere hostnamen opgeven.

`ejabberd_http` Deze module beheert de inkomende HTTP-verbindingen.

Volgende opties kunnen ingesteld worden:

`http_poll` Deze optie zet ondersteuning voor HTTP Polling¹⁷ aan die daarna beschikbaar is op `http://server:poort/http-poll/`.

`web_admin` Deze optie zet de webgebaseerde beheerdersinterface aan die daarna beschikbaar is op `http://server:poort/admin/`, om van deze interface gebruik te kunnen maken, moet u gebruikersnaam en wachtwoord opgeven van een geregistreerde gebruiker met beheerdersprivileges (deze privileges kunnen toegekend worden met de toegangsregel “configure”).

Onderstaande voorbeeldconfiguratie biedt volgende diensten aan:

- C2S-verbindingen op poorten 5222 en 5223 (SSL) en daarnaast wordt gebruiker “gargamel” geblokkeerd.
- S2S-verbindingen op poort 5269.

¹⁷<http://www.jabber.org/jeps/jep-0025.html>

- Webgebaseerde beheerdersinterface op poort 5280 en ondersteuning voor HTTP Polling.
- Alle gebruikers exclusief de beheerders hebben een limiet van 1000 Bytes/seconde op het netwerkverkeer.
- pyAIM-t (aim.voorbeeld.org) op poort 5233 met als wachtwoord “aimgeheim”.
- pyICQ-t (icq.voorbeeld.org en sms.voorbeeld.org) op poort 5234 met als wachtwoord “jitgeheim”.
- pyMSNt (msn.voorbeeld.org) op poort 5235 met als wachtwoord “msngeheim”.
- Jabber Yahoo! Transport (yahoo.voorbeeld.org) op poort 5236 met als wachtwoord “yahoogeheim”.
- jabber-gg-transport (gg.voorbeeld.org) voor Gadu-Gadu op poort 5237 met als wachtwoord “gggeheim”.
- ILE¹⁸ (ile.voorbeeld.org) op poort 5238 met als wachtwoord “ilegeheim”.

```
{acl, blocked, {user, "gargamel"}}.
{access, c2s, [{deny, blocked},
              {allow, all}]}.
{shaper, normaal, {maxrate, 1000}}.
{access, c2s_shaper, [{none, admin},
                    {normal, all}]}.
{listen,
  [{5222, ejabberd_c2s,      [{access, c2s}, {shaper, c2s_shaper}]},
   {5223, ejabberd_c2s,      [{access, c2s},
                             ssl, {certfile, "/pad/naar/ssl.pem"}]},
   {5269, ejabberd_s2s_in,   []},
   {5280, ejabberd_http,     [http_poll, web_admin]},
   {5233, ejabberd_service, [{host, "aim.voorbeeld.org",
                             {password, "aimgeheim"}}]},
   {5234, ejabberd_service, [{hosts, ["icq.voorbeeld.org", "sms.voorbeeld.org"],
                             {password, "jitgeheim"}}]},
   {5235, ejabberd_service, [{host, "msn.voorbeeld.org",
                             {password, "msngeheim"}}]},
   {5236, ejabberd_service, [{host, "yahoo.voorbeeld.org",
                             {password, "yahoogeheim"}}]},
   {5237, ejabberd_service, [{host, "gg.voorbeeld.org",
                             {password, "gggeheim"}}]},
   {5238, ejabberd_service, [{host, "ile.voorbeeld.org",
                             {password, "ilegeheim"}}]}
]
}.
```

Opmerking: voor transports die gebruik maken van Jabberd14 of WPJabberd moet u ervoor zorgen dat zij zelf logs bijhouden en XDB-aanvragen zelf behandelen:

¹⁸<http://ile.jabberstudio.org/>

```

<!--
  De elogger- en rlogger-secties zijn vereist bij gebruik van ejabberd.
  Ze zorgen ervoor dat het transports zelf de logs bijhoudt.
-->

<log id='logger'>
  <host/>
  <logtype/>
  <format>%d: [%t] (%h): %s</format>
  <file>/var/log/jabber/service.log</file>
</log>

<!--
  Omdat ejabberd zelf geen XDB aanbiedt, moet het transport dit zelf regelen.
  xdb_file.so moet daarom worden geladen om alle XDB-aanvragen te behandelen.
-->

<xdb id="xdb">
  <host/>
  <load>
    <!-- Dit is de XDB-bibliotheek van WPJabber of Jabberd (afhankelijk van het transport). -->
    <xdb_file>/usr/lib/jabber/xdb_file.so</xdb_file>
  </load>
  <xdb_file xmlns="jabber:config:xdb_file">
    <spool><jabberd:cmdline flag='s'>/var/spool/jabber</jabberd:cmdline></spool>
  </xdb_file>
</xdb>

```

3.1.6 Modules

De optie `modules` bevat een lijst met alle modules die geladen moeten worden bij het opstarten van `ejabberd`. Elk element van de lijst is een tuple waarin het eerste element de modulenaam is en het tweede een lijst met configuratieopties bevat voor die module. Zie bijlage A voor gedetailleerde informatie over elke module.

Voorbeeld:

```

{modules,
  [{mod_register, []},
   {mod_roster, []},
   {mod_privacy, []},
   {mod_configure, []},
   {mod_disco, []},
   {mod_stats, []},
   {mod_vcard, []},
   {mod_offline, []},
   {mod_echo, [{host, "echo.localhost"}]},
   {mod_private, []},

```

```
{mod_irc,      []},
{mod_muc,      []},
{mod_pubsub,   []},
{mod_time,     [{iqdisc, no_queue}]},
{mod_last,     []},
{mod_version,  []}
]}.

```

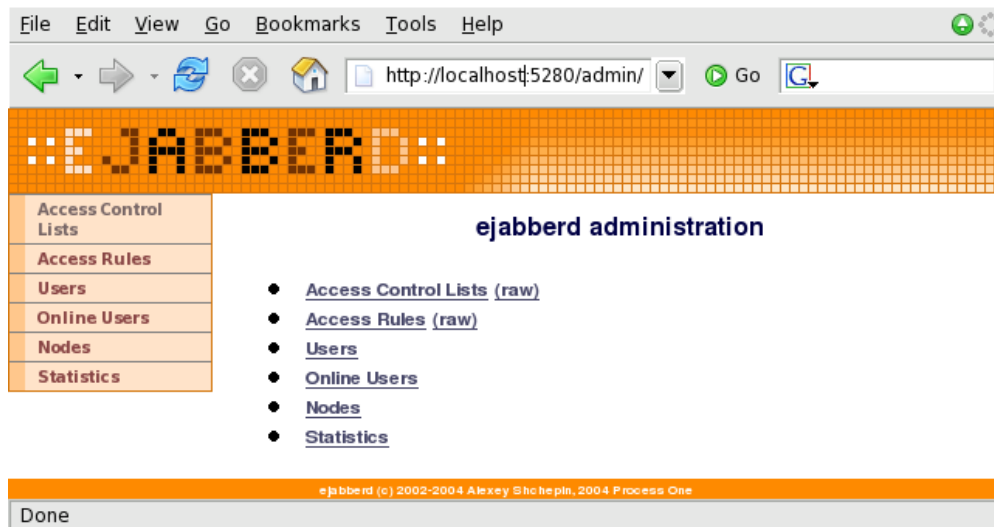
3.2 Configuratie en toezicht houden op een draaiende ejabberd server

3.2.1 Webgebaseerde beheerdersinterface

Om ejabberd te herconfigureren tijdens het draaien moet u de `ejabberd_http`-listener aanzetten en de optie “`web_admin`” opgeven (zie sectie 3.1.5). Daarna kunt u uw browser naar `http://server:poort/admin/` sturen en aanmelden met gebruikersnaam en wachtwoord van een geregistreerde gebruiker met beheerdersprivileges. Met volgende voorbeeldconfiguratie moet u uw browser naar `http://voorbeeld.org:5280/admin` sturen:

```
...
{host, "voorbeeld.org"}.
...
{listen,
 [ ...
  {5280, ejabberd_http, [web_admin]},
  ...
 ]
}.
```

Na authenticatie zou u net zoals in figuur 1 de beheerdersinterface moeten zien. Via deze inter-



Figuur 1: Hoofdpagina van de webgebaseerde beheerdersinterface

face kunt u de toegangsbeperkingen bewerken, gebruikers beheren, reservekopieën aanmaken, de database beheren, listened sockets aan- of uitzetten en statistieken bekijken.

3.2.2 Via de commandoregel met ejabberdctl

Het is ook mogelijk om enkele beheerderstaken uit te voeren met de commandoregel-toepassing ejabberdctl. U kunt de mogelijke argumenten zien door de toepassing uit te voeren zonder argumenten op te geven:

```
% ejabberdctl
```

```
Usage: ejabberdctl node command
```

```
Available commands:
```

stop	stop ejabberd
restart	restart ejabberd
reopen-log	reopen log file
register user password	register a user
unregister user	unregister a user
backup file	store a database backup in file
restore file	restore a database backup from file
install-fallback file	install a database fallback from file
dump file	dump a database in a text file
load file	restore a database from a text file
registered-users	list all registered users

```
Example:
```

```
ejabberdctl ejabberd@host restart
```

```
Vertaling:
```

```
% ejabberdctl
```

```
Gebruik: ejabberdctl node commando
```

```
Beschikbare commando's:
```

stop	ejabberd stoppen
restart	ejabberd herstarten
reopen-log	logbestand heropenen
register user password	gebruiker registreren
unregister user	registratie van een gebruiker ongedaan maken
backup file	reservekopie van de database in een bestand bewaren
restore file	reservekopie van de database vanaf een bestand herstellen
install-fallback file	reservekopie van de database vanaf een bestand installeren
dump file	database in een tekstbestand dumpen
load file	database herstellen vanaf een tekstbestand
registered-users	alle geregistreerde gebruikers tonen

```
Voorbeeld:
```

```
ejabberdctl ejabberd@host restart
```

4 Clustering

4.1 Werking

Een Jabber-domein kan worden aangeboden op één of meerdere `ejabberd` nodes. Deze nodes kunnen gedraaid worden op verschillende computers die met elkaar in verbinding staan via een netwerk. Ze moeten allemaal de mogelijkheid hebben om te verbinden met poort 4369 van alle andere nodes en ze moeten dezelfde “magic cookie” hebben (zie documentatie van Erlang/OTP, samengevat: het bestand `~ejabberd/.erlang.cookie` moet hetzelfde zijn op alle nodes). Dit is nodig omdat alle nodes informatie uitwisselen over verbonden gebruikers, S2S-verbindingen, geregistreerde diensten,...

Elke `ejabberd` node heeft volgende modules:

- router;
- local router.
- session manager;
- S2S manager.

4.1.1 Router

Op elke node is deze module de hoofdrouter van Jabber-pakketten. De module leidt ze naar de juiste locatie op basis van hun doeldomeinen en gebruikt daarvoor een globale routing-tabel. Het doeldomein wordt opgezocht in de routing-tabel en vervolgens wordt het pakket naar het aangewezen proces geleid. Als er niets wordt gevonden dan wordt het pakket doorgezonden naar de S2S manager.

4.1.2 Local router

Deze module verzorgt de aflevering van pakketten waarvan het doeldomein overeenkomt met het domein van de server. Als het gedeelte voor de `@` van de doel-JID niet leeg is, dan verzendt deze module het pakket door naar de session manager. Indien dit niet het geval is dan wordt het verwerkt op basis van de inhoud.

4.1.3 Session manager

Deze module leidt pakketten naar lokale gebruikers en zoekt uit via een statustabel naar welke gebruikersbron een pakket moet worden verzonden. Daarna wordt het pakket ofwel naar het geschikte C2S-proces doorgezonden, ofwel wordt het opgeslagen in het offline geheugen, ofwel wordt het teruggezonden.

4.1.4 S2S manager

Deze module leidt pakketten naar andere Jabberservers en controleert daarvoor eerst het of er al een geopende S2S-verbinding van het domein van de pakketbron naar het domein van het pakketdoel bestaat. Indien er een bruikbare open verbinding is dan leidt de S2S manager het pakket via die verbinding naar het proces en anders wordt er een nieuwe verbinding geopend.

4.2 Een cluster opzetten

Installeer ejabberd eerst op de eerste computer (`eerste_node`) op dezelfde wijze als in hoofdstuk 2. Om deze opstelling daarna uit te breiden naar een ejabberd-cluster moet je volgende stappen doorlopen:

1. Kopieer het bestand `~ejabberd/.erlang.cookie` vanaf `eerste_node` naar de tweede computer (`tweede_node`).
(alternatieve werkwijze) U kunt ook de optie `-cookie content_of_.erlang.cookie` toevoegen aan alle `erl`-commando's hieronder.
2. Ga op de `tweede_node` van de cluster naar de map waar de bestanden van ejabberd zullen worden opgeslagen en verander daar naar de systeemgebruiker voor ejabberd en voer er het volgende commando uit:

```
erl -sname ejabberd \  
    -mnesia extra_db_nodes ["'ejabberd@eerste_node'"] \  
    -s mnesia
```

Dit zorgt ervoor dat Mnesia dezelfde database zal gebruiken als `ejabberd@eerste_node`. Dit kunt u nazien met het commando `mnesia:info().`. U zou dan een heleboel tabellen op afstand en een regel zoals hieronder moeten zien:

```
running db nodes = [ejabberd@eerste_node, ejabberd@tweede_node]
```

3. Voer nu volgende commando uit in dezelfde Erlang-sessie:

```
mnesia:change_table_copy_type(schema, node(), disc_copies).
```

Hiermee zal een lokale opslagruimte aangemaakt worden voor de database.

(alternatieve werkwijze) Verander de opslagmethode van de tabel `'schema'` naar "RAM- en schijfkopie" op `tweede_node` via de webgebaseerde beheerdersinterface.

4. Nu kunt u kopieën van verschillende tabellen toevoegen aan deze node met het commando `mnesia:add_table_copy` of het `mnesia:change_table_copy_type`-commando van hierboven (vervang `schema` met een andere tabelnaam en `disc_copies` met `ram_copies` of `disc_only_copies`).

Welke tabellen gekopieerd moeten worden is erg afhankelijk van uw noden. Uit de uitvoer van het commando `mnesia:info().` kunt u, door te kijken op `eerste_node` naar de

grootte van de tabellen en de standaardopslagmethode voor elke tabel, opmaken welken gekopieerd dienen te worden.

Als u een tabel kopieert naar een node dan zullen opzoekingen erin sneller verlopen op die node maar dan zal het schrijven ernaar trager gebeuren. En het spreekt voor zichzelf dat als er een node uit de lucht gaat, dat dan andere nodes gebruikt kunnen worden als er kopieën zijn van de tabellen.

Raadpleeg sectie “5.3 Table Fragmentation”¹⁹ van de Mnesia User’s Guide als u er meer over wenst te weten.

(alternatieve werkwijze) Gelijkaardig als in vorige stap maar dan voor andere tabellen.

5. Typ “`init:stop().`” of “`q().`” en druk op Enter om de Erlang-shell te beëindigen. Dit kan mogelijk enige tijd in beslag nemen indien Mnesia nog niet klaar is met de overdracht en verwerking van alle vereiste gegevens van `eerste_node`.
6. Start nu ejabberd op `tweede_node` met bijna dezelfde configuratie als op `eerste_node` (waarschijnlijk moet u de opties “`acl`” en “`access`” niet verdubbelen — ze zullen automatisch overgenomen worden van `eerste_node`; daarnaast hoeven de modules `mod_muc` en `mod_irc` slechts op één computer in de cluster aangezet worden).

U kunt bovenstaande stappen herhalen om nog meer nodes toe te voegen aan uw cluster.

A Ingebouwde modules

A.1 Algemene opties

Omdat enkele opties gebruikt kunnen worden bij vele modules wijden we er deze aparte sectie aan.

A.1.1 `iqdisc`

Vele modules gebruiken verwerkers voor de behandeling van IQ-aanvragen van namespaces naar de server (bv. `voorbeeld.org`) of naar een gebruiker (bv. `gebruiker@voorbeeld.org`). De optie `iqdisc` definieert een verwerkingsdiscipline voor dergelijke aanvragen. Disciplines waartussen u kunt kiezen, zijn:

`no_queue` Met deze verwerkingsdiscipline worden alle aanvragen van een namespace onmiddellijk verwerkt. Dit houdt ook in dat er geen andere pakketten verwerkt kunnen worden voor de vorige pakketten zijn afgewerkt. Vandaar dat deze discipline niet aan te raden is wanneer de kans bestaat dat de verwerking van een aanvraag een relatief lange tijd in beslag kan nemen.

`one_queue` Met deze discipline wordt een afzonderlijke wachtrij voor de verwerking van IQ-aanvragen van een namespace aangemaakt en de eigenlijke verwerking ervan wordt parallel uitgevoerd met die van andere pakketten. Deze discipline is ten zeerste aangeraden.

¹⁹http://www.erlang.se/doc/doc-5.4/lib/mnesia-4.2/doc/html/Mnesia_chap5.html#5.3

parallel Met deze laatste discipline wordt een afzonderlijk Erlang-proces voortgebracht voor alle pakketten, zodat ze allemaal parallel verwerkt worden. Hoewel het voortbrengen van een Erlang-proces relatief weinig bronnen vereist, kan dit de normale werking van de server verstoren omdat de Erlang-emulator het maximum aantal processen limiteert (standaard maximum 32000).

Voorbeeld:

```
{modules,  
 [  
   ...  
   {mod_time, [{iqdisc, no_queue}]},  
   ...  
 ]}.
```

A.1.2 host

Met deze optie kunt u de hostnaam specificeren van een module die zich voordoeft als een dienst.

Voorbeeld:

```
{modules,  
 [  
   ...  
   {mod_echo, [{host, "echo.voorbeeld.org"}]},  
   ...  
 ]}.
```

A.2 mod_announce

Deze module maakt het mogelijk om broadcast announce-berichten en het bericht van de dag (MOTD) te verzenden. Als de module geladen is, dan worden berichten die verzonden worden naar de volgende JID's verwerkt (in de veronderstelling dat de domeinnaam van de server `voorbeeld.org` is):

`voorbeeld.org/announce/all` Een bericht dat hiernaar verzonden wordt, wordt verspreidt naar alle geregistreeerde gebruikers op de server. Als een gebruiker online is en verbonden op meerdere bronnen, dan zal alleen de bron met de hoogste prioriteit het bericht ontvangen. Als een geregistreeerde gebruiker niet verbonden is, dan wordt het bericht offline opgeslagen (als de module die instaat voor het bewaren van offline berichten geladen is).

`voorbeeld.org/announce/online` Een bericht naar deze locatie, wordt verzonden naar alle op de server aangemelde gebruikers. Als een gebruiker online is en verbonden is op meerdere bronnen, dan zullen die allemaal het bericht ontvangen.

`voorbeeld.org/announce/motd` Een bericht hiernaartoe, wordt ingesteld als MOTD en zal verzonden worden naar gebruikers op het ogenblik dat ze aanmelden. Daarnaast wordt het bericht ook opgestuurd naar alle verbonden gebruikers (zoals bij `announce/online`).

`voorbeeld.org/announce/motd/update` Als een bericht naar deze JID wordt verzonden, dan wordt het ingesteld als MOTD en zal het verzonden worden naar gebruikers op het ogenblik dat ze aanmelden. Het bericht wordt *niet verzonden* naar gebruikers die al online zijn.

`voorbeeld.org/announce/motd/delete` Elk bericht verzonden naar deze JID verwijdert het bestaande bericht van de dag.

Opties:

`access` Specificeert wie announce-berichten mag verzenden en het bericht van de dag (MOTD) mag instellen. De standaardwaarde is `none` (niemand).

Voorbeeld:

```
% Alleen beheerders mogen announce-berichten versturen:
{access, announce, [{allow, admin}]}.

{modules,
 [
  ...
  {mod_announce, [{access, announce}]},
  ...
 ]}.
```

A.3 `mod_configure`

Opties:

`iqdisc ejabberd:config-discipline` voor de verwerking van IQ-aanvragen (zie [A.1.1](#)).

A.4 `mod_disco`

Deze module voegt ondersteuning voor JEP-0030²⁰ (Service Discovery) toe.

Opties:

`iqdisc http://jabber.org/protocol/disco#items` en `http://jabber.org/protocol/disco#info-discipline` voor de verwerking van IQ-aanvragen (zie [A.1.1](#)).

`extra_domains` Met deze optie kunt u een lijst met domeinen opgeven die aan de antwoorden van de serveritems moeten worden toegevoegd.

²⁰<http://www.jabber.org/jeps/jep-0030.html>

Voorbeeld:

```
{modules,  
  [  
    ...  
    {mod_disco, [{extra_domains, ["jit.voorbeeld.org",  
                                  "etc.voorbeeld.org"]}]},  
    ...  
  ]}.
```

A.5 mod_echo

Deze module doet zich voor als een dienst en echoët elk bericht terug naar de zender. Dit kan handig zijn voor het opsporen van fouten.

Opties:

host Hiermee definieert u de hostnaam van de dienst (zie [A.1.2](#)). Indien niets wordt opgegeven dan zal het voorvoegsel “echo.” worden toegevoegd aan de basishostnaam.

A.6 mod_irc

Deze module is een IRC-transport.

Opties:

host Hiermee definieert u de hostnaam van de dienst (zie [A.1.2](#)). Indien niets wordt opgegeven dan zal het voorvoegsel “irc.” worden toegevoegd aan de basishostnaam.

access Specificeert wie van het IRC-transport gebruik mag maken. De standaardwaarde is **all** (iedereen).

Voorbeeld:

```
{modules,  
  [  
    ...  
    {mod_irc, [{access, all}]},  
    ...  
  ]}.
```

A.7 mod_last

Deze module voegt ondersteuning voor JEP-0012²¹ (Last Activity) toe.

Opties:

`iqdisc jabber:iq:last-discipline` voor de verwerking van IQ-aanvragen (zie A.1.1).

A.8 mod_muc

Deze module implementeert JEP-0045²² (Multi-User Chat).

Opties:

host Hiermee definieert u de hostnaam van de dienst (zie A.1.2). Indien niets wordt opgegeven dan zal het voorvoegsel “`conference.`” worden toegevoegd aan de basishostnaam.

access Specificeert wie de MUC-dienst mag gebruiken. De standaardwaarde is `all` (iedereen).

access_create Specificeert wie nieuwe chatruimtes mag aanmaken op de MUC-dienst. De standaardwaarde is `all` (iedereen).

access_admin Specificeert wie de MUC-dienst mag beheren (de standaardwaarde is `none` (niemand), wat betekent dat enkel de aanmaker zijn eigen chatruimte kan beheren).

Voorbeeld:

```
% Definieer een ACL voor de beheerder.
{acl, admin, {user, "admin"}}

% Definieer een toegangsregel voor de MUC-beheerder.
{access, muc_admin, [{allow, admin}]}

{modules,
 [
  ...
  {mod_muc, [{access, all},
             {access_create, all},
             {access_admin, muc_admin}]},
  ...
 ]}.

```

A.9 mod_offline

Deze module implementeert de mogelijkheid om offline berichten te bewaren.

²¹<http://www.jabber.org/jeps/jep-0012.html>

²²<http://www.jabber.org/jeps/jep-0045.html>

A.10 mod_privacy

Deze module implementeert “Privacy Rules” uit XMPP IM²³.

Opties:

`iqdisc jabber:iq:privacy-discipline` voor de verwerking van IQ-aanvragen (zie A.1.1).

A.11 mod_private

Deze module voegt ondersteuning voor JEP-0049²⁴ (Private XML Storage) toe.

Opties:

`iqdisc jabber:iq:private-discipline` voor de verwerking van IQ-aanvragen (zie A.1.1).

A.12 mod_pubsub

Deze module implementeert JEP-0060²⁵ (Publish-Subscribe).

Opties:

host Hiermee definieert u de hostnaam van de dienst (zie A.1.2). Indien niets wordt opgegeven dan zal het voorvoegsel “pubsub.” worden toegevoegd aan de basishostnaam.

served_hosts Specificeert welke hosts aangeboden worden door de dienst. Indien niets wordt opgegeven dan zal enkel de basishostnaam aangeboden worden.

Voorbeeld:

```
{modules,
 [
  ...
  {mod_pubsub, [{served_hosts, ["voorbeeld.org",
                                "voorbeeld.org"]}]},
  ...
 ]}.
```

²³<http://www.ietf.org/rfc/rfc3921.txt>

²⁴<http://www.jabber.org/jeps/jep-0049.html>

²⁵<http://www.jabber.org/jeps/jep-0060.html>

A.13 mod_register

Deze module voegt ondersteuning voor JEP-0077²⁶ (In-Band Registration) toe.

Opties:

access Specificeert een toegangsregel om registraties te beperken. Als deze regel “deny” (weigeren) oplevert bij een gewenste gebruikersnaam, dan kan die niet geregistreerd worden. De standaardwaarde is `all` (iedereen), wat inhoudt dat er geen beperkingen zijn.

iqdisc jabber:iq:register-discipline voor de verwerking van IQ-aanvragen (zie A.1.1).

Voorbeeld:

```
% Weiger registraties met een te korte naam.
{acl, korte_naam, {user_glob, "?"}}.
{acl, korte_naam, {user_glob, "??"}}.
% Een variant daarop:
{acl, korte_naam, {user_regex, "^..?$"}}.

{access, register, [{deny, korte_naam},
                   {allow, all}]}.

{modules,
 [
   ...
   {mod_register, [{access, register}]},
   ...
 ]}.
```

A.14 mod_roster

Deze module zorgt voor het beheer van rosters.

Opties:

iqdisc jabber:iq:roster-discipline voor de verwerking van IQ-aanvragen (zie A.1.1).

A.15 mod_service_log

Deze module voegt ondersteuning voor archivering van gebruikerspakketten toe via een extern component zoals Bandersnatch²⁷. Deze pakketten worden ingekapseld in het route/*—elementen*aanverzendenaarhet

Opties:

²⁶<http://www.jabber.org/jeps/jep-0077.html>

²⁷<http://jabberstudio.org/projects/bandersnatch/project/view.php>

loggers Hiermee kunt u een lijst van één of meerdere componenten specificeren waarnaar de gebruikerspakketten moeten worden verzonden.

Voorbeeld:

```
{modules,  
 [
  ...
  {mod_service_log, [{loggers, ["bandersnatch.voorbeeld.org"]}]},
  ...
 ]}.
```

A.16 mod_stats

Deze module voegt ondersteuning voor JEP-0039²⁸ (Statistics Gathering) toe.

Opties:

iqdisc <http://jabber.org/protocol/stats-discipline> voor de verwerking van IQ-aanvragen (zie A.1.1).

A.17 mod_time

Deze module verzendt de GMT-tijd na `jabber:iq:time`-aanvragen.

Opties:

iqdisc `jabber:iq:time-discipline` voor de verwerking van IQ-aanvragen (zie A.1.1).

A.18 mod_vcard

Deze module implementeert een eenvoudige Jabbergebruikersdatabase gebaseerd op de vCards van gebruikers en verzendt ook de vCard van de server na een “`vcard-temp`”-aanvraag.

Opties:

host Hiermee definieert u de hostnaam van de dienst (zie A.1.2). Indien niets wordt opgegeven dan zal het voorvoegsel “`vjud.`” worden toegevoegd aan de basishostnaam.

iqdisc `vcard-temp-discipline` voor de verwerking van IQ-aanvragen (zie A.1.1).

search Hiermee kunt u de zoekfunctionaliteit van de dienst aan- (`true`, de standaardwaarde) of uitschakelen (`false`). Als `search` is ingesteld op `false`, dan wordt de optie `host` genegeerd en zullen de diensten niet verschijnen tussen de Jabber Discovery items.

²⁸<http://www.jabber.org/jeps/jep-0039.html>

`matches` Deze optie limiteert het aantal zoekresultaten. Als de waarde is ingesteld op `infinity` (oneindig) dan zullen alle zoekresultaten weergegeven worden. De standaardwaarde is 30.

Voorbeeld:

```
{modules,
 [
  ...
  {mod_vcard, [{search, false}, {matches, 20}]}
  ...
 ]}.
```

A.19 mod_version

Deze module verzendt de versie van ejabberd na `jabber:iq:version`-aanvragen.

Opties:

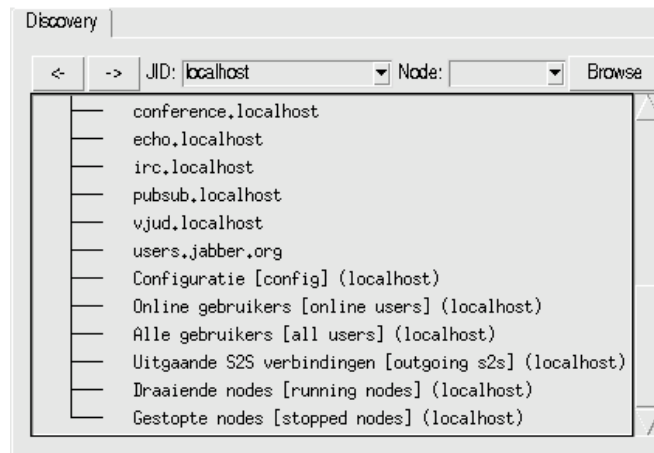
`iqdisc jabber:iq:version-discipline` voor de verwerking van IQ-aanvragen (zie [A.1.1](#)).

B I18n/L10n

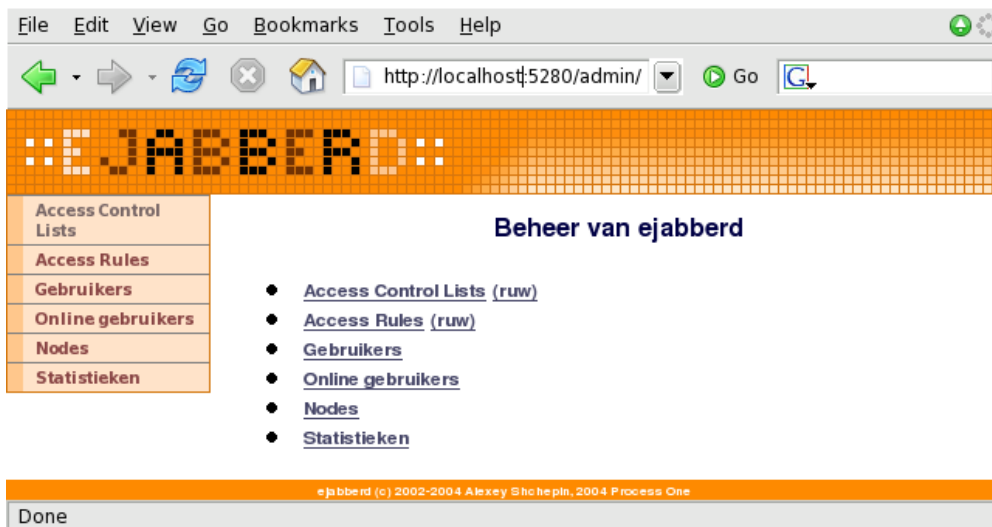
Alle ingebouwde modules ondersteunen het `xml:lang`-attribuut in IQ-aanvragen. In [figuur 2](#) is bijvoorbeeld het antwoord op volgende aanvraag te zien:

```
<iq id='5'
  to='e.localhost'
  type='get'
  xml:lang='nl'>
  <query xmlns='http://jabber.org/protocol/disco#items'/>
</iq>
```

De webgebaseerde beheerdersinterface ondersteunt ook de `Accept-Language` HTTP-header (zie [figuur 3](#) en vergelijk dit met [figuur 1](#))



Figuur 2: Discovery-resultaten bij `xml:lang='nl'`



Figuur 3: Startpagina van de webgebaseerde beheerdersinterface met als HTTP-header "Accept-Language: nl"