Opinnäytetyö (AMK)

Tietojenkäsittelyn koulutusohjelma

Yrityksen tietoliikenne ja tietoturva

2014

Joonas Hakamäki & Jani-Petteri Salonen

VIESTINTÄPALVELIMEN TOTEUTUS YRITYKSEN SISÄISTÄ VIESTINTÄÄ VARTEN

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma | Yrityksen tietoliikenne ja tietoturva

2014 | 80 sivua

Esko Vainikka

Joonas Hakamäki & Jani-Petteri Salonen

VIESTINTÄPALVELIMEN TOTEUTUS YRITYKSEN SISÄISTÄ VIESTINTÄÄ VARTEN

Tämän opinnäytetyön tavoitteena on toteuttaa yrityksen sisäiseen käyttöön tuleva palvelin pikaviestintää ja nettipuheluita varten. Palvelimen tarkoituksena on toimia tietoturvallisena ratkaisuna yrityksen sisäiseen viestintään, ja se toteutetaan avoimen lähdekoodin sovelluksilla Linux-ympäristöön.

Työn teoriaosuudessa kerrotaan käytettyjen pikaviestintä- ja nettipuheluprotokollien historiasta sekä toimintatavasta. Apuna on käytetty protokollien standardijärjestöjen tuottamaa materiaalia.

Työn empiirisessä osuudessa kerrotaan palvelimen rakentamisesta ja käyttöönotosta. Tämä perustuu työn aikana käytettyjen palvelinsovelluksien käyttöönotto- sekä käyttökokemuksiin.

Työn tuloksena on käyttövalmis palvelin, jonka avulla voidaan lähettää pikaviestejä ja soittaa nettipuheluja halutun asiakasohjelman avulla. Tämän lisäksi työstä tehtiin liitteeksi helppolukuinen käyttöohje, jonka avulla vastaavanlainen palvelin on helppo rakentaa uudestaan.

ASIASANAT:

pikaviestintä, nettipuhelut, open source, Linux

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology| Data Communications and Information Security

2014 | 80 pages

Esko Vainikka

Joonas Hakamäki & Jani-Petteri Salonen

THE IMPLEMENTATION OF A COMMUNICATIONS SERVER FOR INTERNAL CORPORATE COMMUNICATION

The objective of the present bachelor's thesis is to implement an internal communication server for internal corporate communication with instant messaging and internet calls. The purpose of the server is to be a secure solution for the company's internal communication and it is implemented with an open source server software in Linux environment.

The theoretical part of the thesis introduces the history and design of the used instant messaging and internet calls protocols.

The empirical part of the thesis discusses the user experiences of building and using the communications server.

The result of the thesis is a ready to use server that can be used to send instant messages and make internet calls with a chosen client program. In addition an easy to read manual is appended in order to facilitate the making of a similar server.

KEYWORDS:

instant messaging, internet calls, open source, Linux

SISÄLTÖ

LIITTEET

KUVIOT

LYHENTEET

IETF	Internet Engineering Task Force
IM	Instant Messaging
IMS	IP Multimedia Subsystem
ITU	International Telecommunication Union
LDAP	Lightweight Directory Access Protocol
NAT	Network Address Translation
RFC	Request for Comments
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
SDP	Session Description Protocol
SMTP	Simple Mail Transfer Protocol
URI	User resource Identifier
VoIP	Voice over Internet Protocol
XML	Extensible Markup Language

1 JOHDANTO

Tämä opinnäytetyö on tehty Trivore Oy:n toimeksiantona. Tarkoituksena on rakentaa Linux-pohjaisen CentOS-käyttöjärjestelmässä toimiva viestintäpalvelin XMPP- ja/tai SIP -protokollien avulla, mikä mahdollistaa käyttäjien kommunikaation pikaviestien ja nettipuheluiden avulla halutulla asiakasohjelmalla. Työssä vertaillaan kahta eri toteutusvaihtoehtoa edellä mainituilla protokollilla.

Ensimmäisessä luvussa selvitetään työn taustat, lähtökohdat ja tavoitteet, sekä tehdään työn rajaus. Toinen ja kolmas luku sisältävät XMPP- ja SIP-protokollia käsittelevän teoriaosuuden, jossa kerrotaan protokollien käyttötarkoituksista, historiasta ja ominaisuuksista protokollien standardijärjestöjen materiaalin avulla. Neljännessä luvussa kerrotaan työhön tarvittavien palvelinohjelmien kriteerit ja valitut ohjelmat. Viides ja kuudes luku sisältävät tutkimusosuuden, jossa käydään läpi tarkemmin palvelinohjelmien ominaisuuksia sekä tutkimustuloksia työn aikana saatujen käyttökokemuksien avulla. Seitsemännessä luvussa pohditaan, miten työssä onnistuttiin ja mitä siitä opittiin.

Liitteinä ovat palvelinsovellusten asennus- sekä käyttöönotto-ohjeet.

Toimeksiantaja

Työn toimeksiantaja Trivore Oy suunnittelee, tekee ja ylläpitää lukuisia ICT-palveluja, kuten virtuaalipalvelimia, www-sivuja ja ohjelmistoja. Trivore tarjoaa muun muuassa IT- ja mobiiliasiantuntijapalveluja, verkon laitteiden ja palveluiden valvontaa sekä tiedon varmistus- ja arkistointipalveluja. Yrityksen toimisto sijaitsee Turussa.

Tausta ja tavoitteet

Trivore Oy:n sisäiseksi kommunikaatiovälineeksi haluttiin sovellus, jolla työntekijät voisivat keskustella toisilleen tekstimuodossa ja nettipuheluilla. Jo saatavilla olevia sovelluksia, kuten Skypeä, ei pidetty käyttötarkoitukseen sopivina, koska niiden tietoturvallisuus koettiin puutteelliseksi. Tästä syystä päädyttiin rakentamaan oma palvelin, johon työntekijät voivat yhdistää valitsemallaan asiakasohjelmalla, eikä käyttäjien välinen liikenne kulje minkään kolmannen osapuolen kautta. Parhaiten tällaiseen tehtävään sopivat XMPP- ja SIP-protokollat, joilla palvelin päätettiin lopulta toteuttaa.

Palvelimen alustana tuli käyttää yrityksessä pääsääntöisesti käytössä olevaa Linux-pohjaista CentOS-käyttöjärjestelmää, ja palvelinsovelluksina tuli käyttää ilmaisia avoimeen lähdekoodiin perustuvia vaihtoehtoja. Koska käytetyt palvelinsovellukset ovat avoimeen lähdekoodiin perustuvia ja ilmaisia, käyttöohje aiotaan laittaa julkiseen jakoon, jotta kuka tahansa voi asentaa itselleen vastaavan palvelimen CentOS -käyttöjärjestelmään. Tästä syystä asennuksesta ja käyttöönotosta tuli tehdä hyvin selkeä ja runsaasti kuvitettu helppolukuinen käyttöohje, jotta palvelinympäristö olisi helppo asentaa ilman sovelluksien aikaisempaa tuntemista.

Työn rajaus

Liitteinä olevissa asennusohjeissa ei tulla käymään läpi CentOS-käyttöjärjestelmän asennusta ja käyttöönottoa, eikä tarkemmin Linux-ympäristön komentoja. Ohjeiden lukijalta odotetaan perustason tietämystä Linux-ympäristöjen toiminnasta. Jitsi-asiakasohjelman asennusta ei käydä ohjeissa läpi, koska sen tilalla voidaan käyttää mitä tahansa muuta XMPP/SIP-asiakasohjelmaa, joten Jitsi itsessään ei ole oleellinen osa työtä.

Opinnäytetyön työnjako

Opinnäytetyö jaettiin siten, että Joonas Hakamäki otti vastuun XMPP-protokollasta sekä Openfire-sovelluksesta, ja Jani-Petteri Salonen SIP-protokollasta sekä Siremis-sovelluksesta. Työnjaosta huolimatta molemmat osallistuivat myös toistensa osa-alueiden työstämiseen. Käyttöohjeet ovat molempien yhdessä kirjoittamat.

2 XMPP

XMPP (Extensible Messaging and Presence Protocol, aikaisemmin nimellä Jabber) on XML:ään pohjautuva avoimen lähdekoodin viestintä- ja läsnäoloprotokolla, jonka IETF on määritellyt RFC:ssä 6120, 6121 ja 6122. Toimintatavaltaan XMPP muistuttaa sähköpostia: yhtä keskitettyä palvelinta ei ole, ja kuka tahansa voi tehdä XMPP-palvelimen, jonka kautta XMPP-viestit kulkevat. (XMPP Standards Foundation 2013a; 2013b.)

XMPP-protokollaa käyttävät tai tukevat lukuisat suositut viestintäpalvelut, kuten esimerkiksi Facebookin pikaviestintä (Facebook 2013). Nämä on mahdollista liittää lähes mihin tahansa XMPP-asiakasohjelmaan, jolloin käyttäjä voi halutessaan keskustella omalla palvelimellaan olevan käyttäjätilinsä avulla esimerkiksi Facebookia käyttävän käyttäjän kanssa.

Käytännössä käyttäjä luo yhteyden ensin haluamallaan asiakasohjelmalla palvelimeen käyttäen XMPP-tunnustaan. Kuten sähköpostissa, XMPP:ssäkin käyttäjä tunnistetaan käyttäjänimen ja palvelimen domainin avulla, jolloin XMPP-tunnus on muodossa käyttäjänimi@domain, aivan kuten sähköpostiosoitteessa. Kun käyttäjä on tunnistautunut onnistuneesti, käyttäjät voivat vaihtaa reaaliajassa XML-muotoisia viestejä keskenään. (IETF 2011.)

XMPP-protokollaan on saatavilla joukko laajennuksia, joista käytetään nimitystä XEP (XMPP Extension Protocol). Laajennuksia voi tehdä kuka tahansa mihin tahansa käyttötarkoitukseen, ja mikäli täyttää XMPP:n standandijärjestö XSF:n (XMPP Standards Foundation) vaatimukset, laajennus voidaan ottaa viralliseen XEP-listaan. Laajennukset mahdollistavat XMPP:n käytön muun muassa ryhmäkeskusteluihin, ääni- ja videopuheluihin sekä tiedostojen jakoon. (XMPP Standards Foundation 2013c.)

Kehitys

XMPP:n juuret johtavat vuonna 1999 kehitettyyn Jabber-projektiin, josta kehiteltiin samannimistä pikaviestintä- ja läsnäoloprotokollaa. Protokollasta pyrittiin tekemään alusta saakka sellainen, että sitä olisi mahdollista laajentaa toimimaan muillakin tavoin. (XMPP Standards Foundation 2013d.)

Vuoden 2002 lopussa IETF perusti XMPP-työryhmän, jonka tehtävänä oli kehittää Jabberista IETF:n standardien mukainen pikaviestintäprotokolla. Syksyllä 2004 työryhmä julkaisi RFC3920:n ja RFC3921:n, jotka edustivat protokollan siihen mennessä kokonaisinta muotoa. (XMPP Standards Foundation 2013d.)

Julkaisunsa jälkeen XMPP otettiin laajaan käyttöön, ja käyttäjien sekä XSF:n testausten ja palautteiden pohjalta vuonna 2011 julkaistiin RFC 6120, 6121 ja 6122, jotka korvasivat XMPP:n aikaisemmat RFC:t. (XMPP Standards Foundation 2013d.)

Toimintaperiaate

XMPP:n tarkoitus on mahdollistaa XML-datapakettien lähetys palvelimen kautta käyttäjältä toiselle. Tällaisesta paketista käytetään nimitystä ”XML stanza”, ja se voi sisältää pikaviestejä, läsnäolotietoja ja kyselyitä. (IETF 2011.)

XMPP-käyttäjätunnus, josta käytetään nimitystä JID (Jabber ID), koostuu kolmesta osasta: paikallisesta osasta (verrattavissa käyttäjänimeen), domain-osasta ja resurssiosasta. Käytännössä JID on siis seuraavanlainen:

käyttäjänimi@domain/resurssi

Paikallisella osalla määritellään, millä käyttäjänimellä palvelimelle kirjaudutaan, ja domain-osalla määritetään, miltä palvelimelta käyttäjätunnus haetaan. Resurssiosalla erotetaan käyttäjätunnuksen samanaikaiset kirjautumiset toisistaan. Jotkut asiakasohjelmat käyttävät resurssiosaa, kun käyttäjä on kirjautunut palvelimelle useasta paikasta samaan aikaan. Käyttäjä voi olla

kirjautuneena esimerkiksi kotonaan ”/koti” resurssilla, ja työpaikallaan ”/toimisto” resurssilla. Resurssiosan käyttäminen ei ole pakollista, ja kaikki asiakasohjelmat eivät edes tue sitä. Ilman resurssiosaa käytettävää JID-tunnusta kutsutaan nimellä ”bare JID”, joka on yleisin tunnuksesta käytettävä muoto, jolla kirjaudutaan esimerkiksi XMPP-palvelimeen sisälle asiakasohjelmaa käyttäen tai etsitään muita käyttäjiä. (IETF 2011.)

XMPP:n käyttöprosessi on seuraavanlainen:

1. Määritetään IP-osoite/domain johon yhdistetään ja portti johon yhdistetään.
2. Aukaistaan TCP-yhteys.
3. Aukaistaan XML-lähetys TCP:n avulla.
4. Neuvotellaan TLS-salaus mikäli käytössä.
5. Autentikoidaan SASL:n (Simple Authentication and Security Layer) avulla.
6. Yhdistetään resurssit lähetykseen.
7. Vaihdetaan haluttu määrä XML stanzoja muiden käyttäjien kanssa.
8. Suljetaan XML-lähetys.
9. Suljetaan TCP-yhteys.

Palvelimelle yhdistäminen tapahtuu asiakasohjelmalla, johon määritetään JID-tunnuksen paikallinen osa sekä domain-osa. Paikallisella osalla määritellään millä käyttäjänimellä palvelimelle kirjaudutaan. Domain-osalla määritellään miltä palvelimelta käyttäjätunnus haetaan. Tämän jälkeen palvelimelle yhdistetään, minkä jälkeen käyttäjä kirjautuu sisään salasanallaan. Samalla tarkistetaan, minkä resurssin kanssa käyttäjä kirjautuu, jos sellainen on määritelty. Tämän jälkeen käyttäjä voi lähettää esimerkiksi pikaviestejä muille käyttäjille. Kirjautuessaan ulos palvelimelta yhteys katkaistaan. (IETF 2011.)

XEP

Yksi XMPP:n lähtökohdista oli tehdä protokollasta helposti laajennettava muihin käyttötarkoituksiin. Näistä laajennuksista käytetään nimitystä XEP. Kuka tahansa voi tehdä haluamaansa käyttötarkoitukseen laajennuksen, ja XMPP-yhteisö onkin kehittänyt aktiivisesti satoja XEP-protokollia. Nimestään

huolimatta XEP:n ei kuitenkaan tarvitse olla protokolla, vaan se voi olla myös esimerkiksi ohjeistus jo olemassa olevien protokollien käyttöön tai se voi tarkentaa jonkin olemassa olevan protokollan toimintatapoja. XSF jakaa XEP:t viiteen eri tyyppiluokkaan:

- **Standards Track XEP**, johon kuuluvat erilaiset XMPP:n rinnalla käytettävät protokollat. Näihin kuuluvat esimerkiksi ryhmäkeskustelun mahdollistava XEP-0045.
- **Informational XEP**, johon kuuluvat muiden protokollien yleiset käytäntöohjeet XMPP:n kanssa, kuten esimerkiksi SASL-autentikoinnin käytäntösuositukset sisältävä XEP-0175.
- **Historical XEP**, johon kuuluvat vanhat XEP:t, jotka on tehty ennen kuin XSF otti käyttöön XEP-standardoinnin, kuten esimerkiksi XML sanzojen tallennusprotokolla XEP-0049.
- **Humorous XEP**, johon kuuluvat huumorimielessä tehdyt XEP:t, jotka eivät ole oikeasti käyttökelvollisia. Näitä julkaistaan aprillipäivänä, ja ne hyväksytään automaattisesti listalle. Näihin kuuluvat esimerkiksi ääniviestien lähettämisen telepaattisesti mahdollistava XEP-0183.
- **Procedural XEP**, johon kuuluvat prosessit ja käytännöt, joita XSF:n tulee noudattaa. Näihin kuuluvat esimerkiksi XEP-standardivaatimukset sisältävä XEP-0001.

XSF on tehnyt virallisen XEP-listan, johon käyttäjien on mahdollista ehdottaa omia laajennuksiaan. Listaan päästäkseen XEP:n on täytettävä XSF:n asettamat standardit. Laajennuksen dokumentaation pitää sisältää tekijän tiedot sekä mahdolliset tiedot immateriaalioikeuksista, minkä lisäksi dokumentaation ulkoasun on vastattava XSF:n kriteerejä. Tämän jälkeen XSF:n valtuuttama sensori antaa XEP:lle numeron ja tyyppiluokan, jonka jälkeen tämä julkaisee laajennuksen virallisella standardilistalla ”Experiental” (kokeilullinen) -tilassa. Tämän jälkeen XSF:n hallinto suorittaa laajennuksen testauksen, jonka jälkeen äänestetään, liitetäänkö laajennus viralliselle listalle. (Saint-Andre 2010.)

3 SIP

SIP (Session Initiation Protocol) on IETF:n RFC:ssä 3261 määrittelemä signalointiprotokolla istuntojen luomiseen, muokkaamiseen ja päättämiseen IP-verkkojen yli. Istunnot voivat koostua yhdestä tai useammasta mediavirrasta. Eniten protokollaa käytetään VoIP-puheluissa, mutta sitä voidaan käyttää myös muihin sovelluksiin, kuten videopuheluihin, pikaviestintään, tiedostojen siirtoon, tilatietojen välitykseen ja peleihin. (IETF 2002.)

SIP on kevyt, yksinkertainen ja tekstipohjainen protokolla, joka ei välitä kuljetuskerroksen protokollasta. SIP-pakettien välittämiseen voidaan käyttää niin TCP- kuin UDP-protokollaakin. Monilta osin se on hyvin samankaltainen HTTP- ja SMTP-protokollien kanssa. SIP-osoite muistuttaakin hyvin paljon sähköpostiosoitetta. SIP URI on muotoa käyttäjä@palvelin(:porttinumero), esimerkiksi kayttaja@voippalvelu.fi. (VoIP.ms Wiki 2011.)

SIP-protokollaa itsessään käytetään vain päätelaitteen paikantamiseen IP-verkossa, varsinainen puhelun muodostus tapahtuu hyväksikäyttäen SDP-protokollaa, joka on määritelty RFC:ssä 4566. SDP-viestit välitetään SIP-pakettien sisällä. SDP-protokollan avulla puhelun osanottajat vaihtavat keskenään kaiken tarvittavan tiedon varsinaisen puheen välittävän RTP-virran muodostamiseen. Tarvittavia tietoja ovat vähintään IP-osoite, porttinumero ja käytettävä koodekki. (IETF 2006; Toncar 2013.)

Varsinaisen puheen tai videon välittämiseen käytetään edellä mainittua RTP-protokollaa, joka on määritelty RFC:ssä 3550. Protokolla on kehitetty täysin reaaliaikaisen multimedian eli käytännössä äänen ja kuvan välittämiseen IP-verkkojen yli. Yleisesti ottaen RTP-paketteja kuljetetaan UDP:n yli datan reaaliaikavaatimusten takia ja tästä syystä protokollaan on sisäänrakennettu latenssinhallinta ja virheenkorjaus. Paketit sisältävät myös tiedon siitä, missä formaatissa itse pakettien sisältämä video- tai puhedata on. RTP:n yhteydessä käytetään myös sisarprotokollaa RTCP:tä, jonka päätehtävänä on välittää QoS-tietoa RTP-streamin osanottajille, jotta päätelaitteet huomaavat mahdollisen

yhteyden huononemisen ja osaavat säätää RTP-streamin parametrejä muuttuneeseen tilanteeseen sopiviksi. (IETF 2003.)

Kehitys

Internetin yleistyessä 1990-luvulla haluttiin kehittää protokolla puheluiden välittämiseen IP-verkon yli. Perinteiset puhelinverkot ovat piirikytkentäisiä, eli puhelulle taataan tietty reitti päätepisteestä toiseen, ja puheluiden laatu pysyy tasaisena koko puhelun ajan. IP-verkot taas ovat pakettikytkentäisiä ja paketit saattavat kulkea eri reittejä, ja tämä luonnollisesti asettaa haasteita puheluiden laadun varmistamiseen. Tätä varten tarvittiin signalointiprotokolla, joka toimii IP:n yli ja yksinkertaisimmillaan mahdollistaa puheluiden aloittamisen, konfiguroinnin ja lopettamisen. (Shaham 2009.)

SIP-protokollan kehittäminen aloitettiin jo 1990-luvun alkupuolella, mutta ensimmäinen standardiluonnos oli valmis vasta vuonna 1997. Tämän jälkeen protokollaan tehtiin merkittäviä muutoksia ja protokollan version 2.0 standardiluonnos oli valmis vuonna 1998. SIP hyväksyttiin protokollaksi vuonna 1999 RFC:ssä 2543, mutta protokollaa selvennettiin ja bugeja korjattiin vielä tämän jälkeen, ja vuonna 2000 protokolla päivittyi nykymuotoiseksi, kun RFC 3261 korvasi 2543:n. (Johnston 2009, 20.)

Protokollan kehitys ei luonnollisestikaan pysähtynyt tähän, vaan sitä kehitetään yhä. Vuosien mittaan protokollaan on lisätty lukuisia ominaisuuksia, kuten tilatiedot, pikaviestit, tuki puhelujen siirtoon, ja niin edelleen. Näitä ominaisuuksia kehitettäessä on syntynyt monia standardeja. Esimerkiksi pikaviestit ja tilatiedot muodostuvat yhteensä yli kymmenestä internet-standardista. (Shaham 2009.)

Epäilemättä protokollaan tullaan lisäämään yhä enemmän ominaisuuksia, koska protokolla on osa IMS-viitekehystä, joka on käytössä kolmannen sukupolven matkapuhelinverkoissa ja käyttää SIP-protokollaa signalointiin. Vielä tärkeämmässä roolissa SIP on LTE-verkoissa, joissa puhelutkin kulkevat IP:n yli. (LTE University 2012.)

SIP ei ole ainoa protokolla puheluiden välittämiseen internetin yli. Kilpaileva protokolla on ITU:n kehittämä H.323-protokolla, joka tähtää käytännössä samaan toiminnallisuuteen. Nykyisin SIP kuitenkin käytännössä hallitsee VoIP-maailmaa ja H.323 on käytössä lähinnä isoilla teleoperaattoreilla. Syy tähän on yksinkertainen: SIP on ohjelmoijaystävällinen ja siihen on helppo lisätä toiminnallisuuksia. SIP siis kehittyy nopeammin ja toteutuksia on saatavilla niin kaupallisilta valmistajilta kuin avoimen lähdekoodin yhteisöiltäkin. Tämä taas helpottaa huomattavasti VoIP-palveluiden tarjoamista, koska sovellustarjonta on laajaa ja olemassa olevia sovelluksia on helppo muokata yrityksen tarpeita vastaaviksi. (Shaham 2009.)

Toimintaperiaate

Helpoimmillaan puhelun muodostaminen käyttäen SIP-protokollaa on hyvin yksinkertainen prosessi. SIP-protokolla tukee kuutta pyyntöä ja kuudenlaisia vastauksia. (3CX 2013; Matthews 2012.)

Pyynnöt

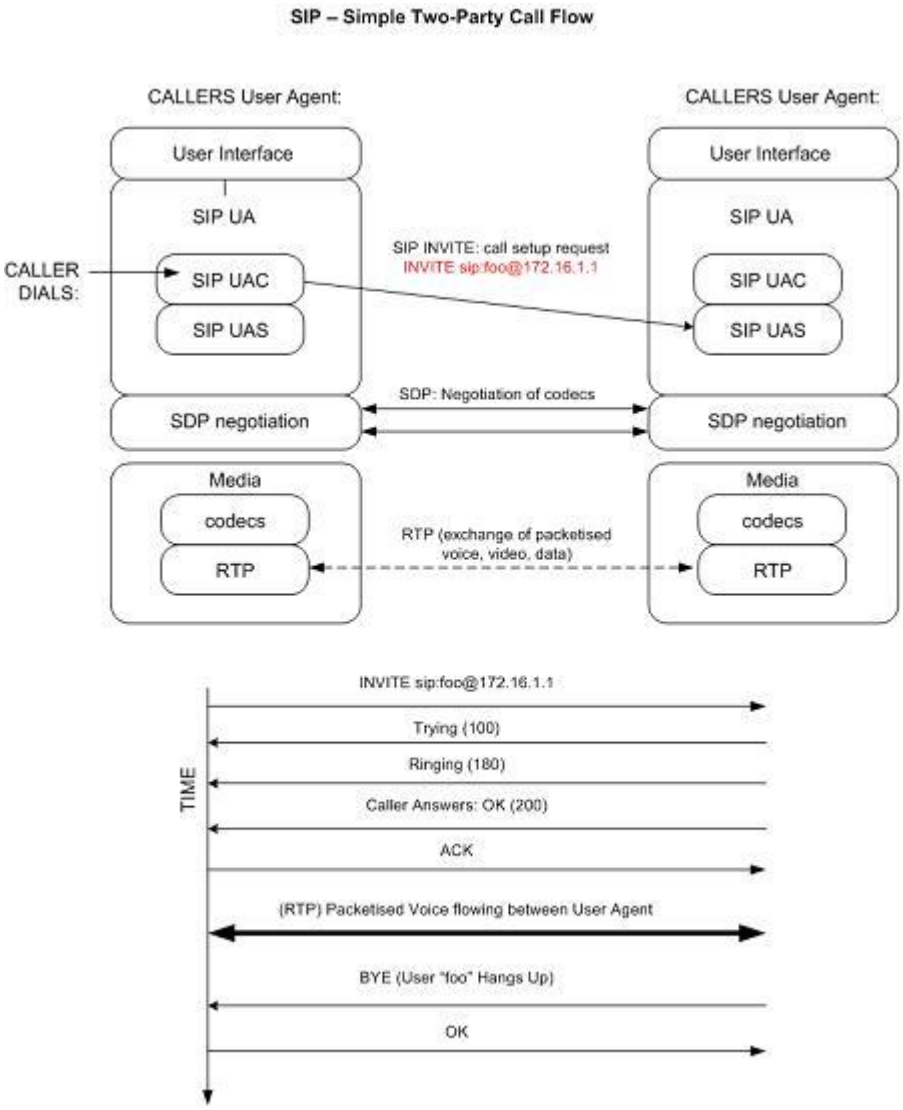
- INVITE – Aloittaa puhelun
- ACK - Vahvistaa INVITE-pyynnön
- BYE - Päättää puhelun
- CANCEL – Peruuttaa puhelun luomisen, mutta ei päätä olemassaolevia puheluita
- OPTIONS - Välittää tietoja soittavan ja vastaanottavan SIP-puhelimen ominaisuuksista
- REGISTER - Välittää käyttäjän sijainnin (verkkonimi, IP-osoite) rekisteröinti-palvelimelle.

Vastaukset

- SIP 1xx - Tietoa sisältävät vastaukset, kuten 180, joka tarkoittaa soittoa

- SIP 2xx - Pyyntöjen onnistumiset
- SIP 3xx - Uudelleenohjaukset
- SIP 4xx - Pyyntöjen epäonnistumiset
- SIP 5xx - Palvelinvirheet
- SIP 6xx – Globaalit virheet, pyyntöön ei osaa vastata mikään palvelin.

SIP-puhelun muodostaminen on kuvattu kuviossa 1.



Kuvio 1. SIP-puhelun muodostaminen. (Matthews 2012.)

Ensimmäiseksi päätelaite rekisteröityy palvelimelle ja ilmoittaa sille verkkosijaintinsa lähettämällä REGISTER-pyynnön, palvelin vastaa onnistuneeseen pyyntöön 200 OK.

Puhelu muodostetaan lähettämällä päätelaitteesta (UAC) INVITE-pyyntö palvelimelle, joka välittää pyynnön vastaanottajalle (UAS). Jos vastaanottaja vastaa puheluun, lähetetään puhelun soittajalle 200 OK -vastaus, muussa tapauksessa joku 4xx-vastauksista, tilanteesta riippuen. Tässä vaiheessa päätelaitteet myös neuvottelevat istunnon ominaisuudet, mm. käytettävän koodekin. INVITE-viestissä välitetään myös osoitetiedot, joiden avulla RTP-yhteys voidaan muodostaa. Käytännössä 200 OK -vastaus lähetetään vasta siinä vaiheessa, kun voidaan yrittää muodostaa RTP-yhteyttä. Soittaja kuittaa 200 OK -vastauksen lähettämällä ACK-pyynnön. (Matthews 2012.)

Puhelu voidaan päättää kumman tahansa osallistujan taholta lähettämällä BYE-pyyntö. Tähän osapuoli, jolle pyyntö lähetettiin, vastaa 200 OK -vastauksella. Samalla päätelaitteet myös terminoivat välillään olevan RTP-virran.

4 SOVELLUKSIEN VALINTA

Kriteerit

Sekä XMPP- että SIP-palvelinsovelluksien tuli toimia Linux-pohjaisessa CentOS-käyttöjärjestelmässä ja perustua avoimeen lähdekoodiin. Web-käyttöliittymä koettiin pakolliseksi ominaisuudeksi. Myös mahdollisuus LDAP-integraatioon otettiin huomioon, jotta palvelimelle pystyisi kirjautumaan samoilla käyttäjätunnuksilla kuin työasemille, mikäli käytössä on Active Directory-palvelin. Asiakasohjelman tuli toimia sekä XMPP- että SIP-protokollilla ja olla ilmainen.

Valitut ohjelmat

XMPP -sovellukseksi valittiin **Openfire**. Sen eduiksi nähtiin muun muassa

- helpolta vaikuttava asennus ja käyttöönotto
- web-käyttöliittymä
- käyttäjätilien hallinnointi MySQL-tietokannan avulla
- sisäänrakennettu VoIP
- LDAP-integraatio.

SIP -sovellukseksi valittiin **Kamailio** (entinen **OpenSER**). Sen eduiksi nähtiin muun muassa

- skaalautuvuus, Kamailio pystyy esimerkiksi käsittelemään tuhansia puheluita sekunnissa vanhemmillakin koneilla
- käyttäjätilien hallinnointi MySQL-tietokannan avulla
- sisäänrakennettu IM
- NAT-reititys
- mahdollisuus monipuolisiin lisäominaisuuksiin, muun muassa LDAP-integraatio, ns. moduuleiden avulla.

Lisäksi Kamailioon on saatavilla erikseen asennettava web-käyttöliittymä **Siremis**, joka otettiin myös käyttöön.

Asiakasohjelmaksi valittiin **Jitsi** (entinen **SIP Communicator**). Sen eduiksi nähtiin seuraavat:

- tuki sekä XMPP- että SIP-protokollille
- samanaikainen IM ja VoIP
- työpöydän jakaminen muille käyttäjille
- mahdollisuus käyttää useampaa käyttäjätiliä samaan aikaan.

Jitsin lisäksi testauksessa käytettiin kahta muuta asiakasohjelmaa, **Blinkiä** ja **Sparkia**, mutta näiden toiminnot koettiin rajallisiksi, joten Jitsi pidettiin pääasiallisena asiakasohjelmana.

5 OPENFIRE

Openfire on monipuolinen javalla kirjoitettu avoimen lähdekoodin XMPP-palvelinohjelmisto. Se tarjoaa runsaasti ominaisuuksia, joita ei välttämättä muutoin löydy XMPP- tai XEP-protokollista, kuten esimerkiksi palvelimelta palvelimelle yhdistäminen sekä tuki monille muille keskusteluprotokollille kuten IRC ja ICQ. Tärkeimmät ominaisuudet tässä tapauksessa olivat pikaviestintä, ryhmäkeskustelut, puhelut, videopuhelut, tiedoston siirto, tilatiedot ja LDAP-integraatio.

Tietoturva otettiin huomioon käyttämällä TLS-protokollaa asiakasohjelman ja palvelimen väliseen kommunikointiin, mutta myös UDP ja TCP ovat tuettuja.

MySQL:n tilalla käytettiin siihen pohjautuvaa MariaDB-nimistä relaatiotietokantajärjestelmää. Etuina MySQL:n nähden MariaDB:ssä on täysin avoin lähdekoodi sekä hieman tehokkaampi suorituskyky. Koska MariaDB tukee suoraan MySQL-tietokantoja, se voidaan asentaa vanhan MySQL:n tilalle muuttamatta tietokantoja. MariaDB käyttää myös täysin samoja komentoja kuin MySQL. Käytännössä eroa MySQL:ään ei siis ole, ja selkeyden vuoksi käytetään tekstissä nimeä MySQL.

Asennuskokemukset

CentOS-käyttöjärjestelmään Openfirea ei löytynyt suoraan pakettivarastosta, vaan se piti ladata Openfiren kotisivuilta. Muutoin asennus tehtiin kuten minkä tahansa muun Linux-ohjelman asentaminen. Ensimmäisellä käynnistyskerralla ohjelmisto ei kuitenkaan käynnistynyt. Ongelma ratkaistiin asentamalla puuttuvan kirjaston 32-bittinen versio, sillä kyseinen ongelma ilmenee vain asennettaessa Openfireä RPM-paketista 64-bittiseen RHEL-pohjaiseen käyttöjärjestelmään. Käynnistysskripti ei kirjaston asentamisenkaan jälkeen antanut mitään informaatiota Openfiren käynnistymisestä, vaikka ohjelma käynnistyi onnistuneesti.

Käyttöönotto ja konfiguroiminen MySQL-autentikaatiolla

Käyttöönotto MySQL-autentikaatiolla oli erittäin yksinkertaista web-käyttöliittymän avulla. Openfire ei kuitenkaan luo asennusvaiheessa tietokantaa, vaan se tulee luoda itse terminaalissa. Tästä syystä oli tiedettävä MySQL:n peruskomentoja uuden tietokannan luomista sekä Openfiren tietokantapohjan tuomista varten. Tietokannan luomista lukuun ottamatta koko käyttöönotto tehtiin web-käyttöliittymän kautta.

Web-käyttöliittymä oli selkeä ja yksinkertainen, joten asetusten konfigurointi sen kautta oli vaivatonta. Vaikka asetuksia ja ominaisuuksia oli paljon, kaikki tarvittavat löytyivät ja niiden konfiguroiminen oli helppoa. Myös käyttäjien ja ryhmien lisääminen onnistui vaivattomasti.

Käyttöönotto ja konfiguroiminen LDAP-autentikaatiolla

LDAP-autentikaation käyttöönotto ei juuri eronnut MySQL-autentikaation käyttöönotosta. Koska Openfire käyttää MySQL-tietokantaa myös muuhun kuin autentikaatioon, kuten esimerkiksi web-käyttöliittymän asetuksien tallentamiseen, MySQL-tietokanta täytyi luoda joka tapauksessa. LDAP-autentikaatiota käyttöönotettaessa määritellään LDAP-palvelin, polku/haara (Base DN), josta tiedot haetaan, sekä pääkäyttäjä, jolla on oikeudet lukea tietoja määritellystä polusta tai koko hakemistosta. Active Directorya LDAP-palvelimena käytettäessä oletusasetukset toimivat suoraan, mutta riippuen toteutuksesta voidaan joutua määrittelemään, mikä LDAP-palvelimelta löytyvä attribuutti vastaa mitäkin Openfiren kenttää.

Asetusten konfiguroiminen toimii kuten MySQL-autentikaatiota käytettäessä, mutta käyttäjät ja niiden tiedot haetaan LDAP-palvelimelta, eikä niitä pysty muokkaamaan web-käyttöliittymällä. Tämä johtuu siitä, että LDAP toimii vain luku-tilassa, joten käyttäjien- ja ryhmienhallinta täytyy tehdä AD-palvelimen puolelta. Tarvittaessa on mahdollista määritellä kuitenkin erilaisia suodattimia Openfireen, joilla LDAP hakee halutut käyttäjät ja ryhmät oikeasta paikasta AD-

palvelimelta. Suodattimia muokatessa on kuitenkin hyvä olla erittäin varovainen, koska niitä käsitellessä lukitsee helposti itsensä ulos koko Openfire-palvelimelta, mikäli Openfiren pääkäyttäjän sijaintia muutetaan huolimattomasti.

Käyttökokemukset

Openfirea testattiin Jitsillä ja Sparkilla, jotka molemmat ovat XMPP:tä tukevia asiakasohjelmia. Pikaviestintä, ryhmäkeskustelut, videopuhelut, tiedonsiirto, tilatiedot ja kaikki muut ominaisuudet toimivat ongelmitta. Testauksessa ei myöskään huomattu ongelmia eri asiakasohjelmien välisessä yhteentoimivuudessa.

Yhteenveto

Kokonaisuudessaan Openfire on erittäin hyvä XMPP-palvelin monipuolisilla ominaisuuksilla. Käytännössä konfigurointi on helppoa ja peruskäyttöön soveltuvan palvelimen saa toimintakuntoon hyvin nopeasti, noin puolessa tunnissa. Integrointi yrityksen olemassa oleviin järjestelmiin sujuu saumattomasti, tosin se vaatii aikaisempaa osaamista esimerkiksi LDAP:in ja MySQL:n osalta, mikä yritysmaailmassa tuskin on ongelma. Jos on tarvetta VoIP/IM-palvelimelle, Openfire tarjoaa kaiken samassa helposti asennettavassa ja ylläpidettävässä paketissa.

6 KAMAILIO

Kamailio on modulaarinen avoimen lähdekoodin SIP-palvelin, joka on kehitetty lähinnä palveluntarjoajien käyttöön. Tämä näkyy muun muassa siinä, että 4 GB:n muistilla varustettu palvelin pystyy palvelemaan 300 000 käyttäjää samanaikaisesti. Koska kyseessä on SIP-palvelin, Kamailion ominaisuuksiin kuuluu luonnollisesti myös perinteiseen puhelinverkkoon soittaminen, mutta sitä varten tarvitaan PSTN-gateway, jota ei ollut käytettävissä.

Suurin osa Kamailion ominaisuuksista on toteutettu moduulein, joita on tällä hetkellä jo yli 150, mutta vain murto-osaa niistä tarvitaan peruskäyttöön. Tärkeimmät ominaisuudet tässä tapauksessa olivat pikaviestintä, puhelut, videopuhelut, tilatiedot (SIMPLE-protokolla) sekä XCAP-serveri, joka tallentaa yhteystiedot palvelimelle. Myös integraatiota Active Directoryyn kaivattiin, mutta SIP-protokolla asettaa omat rajoitteensa käyttäjän tunnistamiselle, vaikka Kamailio itsessään tukee sekä LDAP- että RADIUS-protokollaa.

Myös Kamailiossa tietoturva otettiin huomioon käyttämällä TLS-protokollaa asiakasohjelman ja palvelimen väliseen kommunikointiin, mutta myös UDP ja TCP ovat tuettuja.

Kuten Openfirenkin kanssa, myös Kamailiota käytettäessä MySQL korvattiin MariaDB:llä. Käytännössä eroa ei kuitenkaan ole, ja selkeyden vuoksi tekstissä käytetään nimeä MySQL.

Asennuskokemukset

Kuten Openfirea, myöskään Kamailiota ei löytynyt CentOS-käyttöjärjestelmään suoraan pakettivarastosta, joten se ladattiin Kamailion kotisivuilta. Muutoin asennus tehtiin kuten minkä tahansa muun Linux-ohjelman asentaminen. Myös osa moduuleista oli pakattu erillisiin RPM-paketteihin, joten myös ne piti ladata erikseen. CentOS-käyttöjärjestelmään kaikkia moduuleja ei kuitenkaan ollut

sellaisinaan saatavilla. Kaikki peruskäyttöön tarvittavat moduulit olivat saatavilla erillisinä paketteina, mutta esimerkiksi LDAP-testauksessa käytetty RADIUS-moduuli piti kääntää lähdekoodista. Moduulia ei kuitenkaan käytetty muuhun kuin testaukseen, jossa yritettiin kiertää SIP-protokollan LDAP-autentikaatiolle aiheuttamat rajoitukset. Vaikka se teoriassa vaikutti helpolta, käytännössä autentikoinnin rajoituksien kiertäminen ei onnistunut.

NAT:n asettamien rajoitusten kiertämistä varten asennettiin RTPproxy -niminen ohjelma, joka tarvittaessa ottaa vastaan RTP-paketit ja lähettää ne eteenpäin, eli paketit kiertävät palvelimen kautta sen sijaan, että ne kulkisivat suoraan käyttäjältä toiselle kuten normaalisti. Tämän lisäksi Kamailioon asennettiin RTPproxy-moduuli.

Käyttöönotto ja konfiguroiminen MySQL-autentikaatiolla

Kamailio käyttää MySQL-tietokantaa autentikaation lisäksi myös käyttäjien tilatietojen tallentamiseen niin haluttaessa ja palvelimen osoitteen tallentamiseen. Käyttöönotto MySQL-autentikaatiolla oli erittäin helppoa, sillä Kamailion mukana tulee automaattinen ohjelma tietokannan luomiseen. Ohjelman asetustiedostoon oli vain muokattava haluttu tietokantatyyppi, jonka jälkeen ohjelma luo tietokannan automaattisesti.

Tavallisesti Kamailion käyttäjätilejä hallittaisiin terminaalin komennoilla, mutta Kamailiota varten on kehitetty myös Siremis-niminen web-hallintaympäristö, joka otettiin tässä tapauksessa käyttöön. Käyttäjätilien hallinnoimisen lisäksi Siremiksen kautta saa haettua dataa palvelimen käytöstä ja esimerkiksi tietoja laskutusta varten. Tässä tapauksessa Siremistä ei käytetty muuhun kuin käyttäjien lisäämiseen. Kuten Openfiressa, myös Siremiksessä käyttöliittymä on hyvin selkeä ja yksinkertainen.

Kamailion asetuksien konfiguraatio tehtiin pääsääntöisesti muokkaamalla yhtä asetustiedostoa. Asetustiedoston syntaksi on hyvin lähellä C-ohjelmointikieltä, joten jonkinlainen kokemus ohjelmoinnista on tarpeen, jos Kamailiosta haluaa saada kaikki mahdolliset ominaisuudet käyttöön. Peruskäyttöön tulevan

palvelimen konfiguroiminen oli varsin helppoa, sillä tiedostoon ei tarvitse lisätä kuin muutama rivi haluttujen moduulien aktivoimiseksi, mutta monimutkaisemmat moduulit vaatisivat esimerkiksi pakettien reititysasetuksien muokkaamista.

Käyttöönotto ja konfiguroiminen LDAP-autentikaatiolla

Käyttöönotto LDAP-autentikaatiolla vaikutti helpolta, mutta käytännössä se osoittautui mahdottomaksi. Ensin ladattiin LDAP-moduuli ja määriteltiin polku, mistä käyttäjiä haetaan, sekä käyttäjä, jolla on oikeus lukea tietoja kyseisestä polusta. Tämän jälkeen määriteltiin Kamailiolle, mitkä LDAP-attribuutit vastasivat käyttäjätunnusta ja salasanaa. Ongelmia ilmeni siinä vaiheessa, kun yritettiin käyttää Active Directorya LDAP-palvelimena, koska Active Directory tallentaa salasanat salattuina. Tämä on ongelma siksi, että Kamailiossa ja SIP-protokollassa autentikaatio tapahtuu vertaamalla esimerkiksi tietokantaan tallennettua salasanaa käyttäjän syöttämään salasanaan, mikä ei luonnollisestikaan onnistunut, kun tallennettu salasana oli salatussa muodossa. Ongelmaa yritettiin kiertää käyttämällä RADIUS-moduulia salasanan hakemiseen, mutta tässä ei lukuisten testausten jälkeen onnistuttu. Autentikaatio-ongelma ratkeaisi sillä, että lisättäisiin AD-palvelimelle attribuutti, joka sisältäisi SIP-salasanan selkokielisessä muodossa, mutta tässä on omat tietoturvariskinsä ja käyttäjähallinta vaikeutuu, kun käyttäjillä on kaksi salasanaa. Yksi ratkaisu olisi kirjoittaa oma moduuli, joka suorittaisi normaalin LDAP bind -operaation autentikoidakseen käyttäjän, mutta C-kielen opetteluun olisi mennyt aivan liian paljon aikaa, joten ongelma jäi ratkaisematta.

Asetusten konfigurointi LDAP-autentikaatiota käyttämällä ei eroa MySQL-autentikaation käytöstä muutoin kuin käyttäjienhallinnan osalta, mikä täytyy tehdä LDAP-palvelimen puolella.

Käyttökokemukset

Kamailiota testattiin Jitsillä ja Blinkillä, jotka molemmat ovat SIP-asiakasohjelmia. Keskustelut, puhelut ja tilatiedot toimivat ongelmitta Jitsillä, joskaan Blink ei tue pikaviestitoimintoa. Asiakasohjelmien välisessä yhteentoimivuudessa ei muutoin huomattu ongelmia.

Testauksen yhteydessä NAT aiheutti ongelmia puheluiden kanssa. Puhelut kyllä yhdistivät toisiinsa, mutta ääni ei liikkunut kumpaankaan suuntaan. Lopulta syyksi selvisi NAT-reitittimen SIP-helper -toiminto, joka muokkaa SIP-pakettien sisältöä vaihtamalla niiden sisältämän loppukäyttäjän IP-osoitteen NAT-reitittimen ulkoiseksi IP-osoitteeksi. Tämä johti siihen, että Kamailioon sisäänrakennettu NAT:n tunnistuslogiikka ei toiminut ollenkaan ja molemmat asiakasohjelmat yrittivät lähettää puheluiden RTP-paketit NAT-reitittimen ulkoiseen IP-osoitteeseen. Useimmat NAT-reitittimet eivät salli paketteja sisäverkosta ulkoiseen IP-osoitteeseensa, joten puhelut eivät toimineet.

Ongelma lähti ratkeamaan, kun huomattiin, että yksikään SIP-paketti ei sisältänyt sisäverkon IP-osoitetta, vaan se oli muuttunut NAT-reitittimen ulkoiseksi IP-osoitteksi. Lopulta TLS:n päällekytkeminen varmisti ongelman olemassaolon, koska TLS:ää käytettäessä puhelut toimivat. Ongelma ratkesi yksinkertaisesti kytkemällä SIP-helper palvelu pois päältä.

Yhteenveto

Kamailio on todella monipuolinen ohjelmisto, eikä testauksessa käytetty ominaisuuksista kuin murto-osaa. Se on myös valitettavan vaikeakäyttöinen. Osa rajoituksista johtuu suoraan SIP-protokollasta, kuten LDAP-autentikoinnin toimimattomuus, mutta myös siitä, että Kamailiota ei varsinaisesti ole tarkoitettu tällaiseen käyttöön, mihin sitä yritettiin ottaa. SIP-välityspalvelimena osana palveluntarjoajan runkoverkkoa Kamailio on varmasti paras ja tehokkain ratkaisu, mutta toimiston pikaviestintäympäristön pystyttämiseen on saatavilla parempiakin vaihtoehtoja.

7 POHDINTA

Opinnäytetyömme tavoitteena oli toteuttaa yrityksen sisäiseen käyttöön tuleva viestintäpalvelin, jonka avulla voitaisiin lähettää pikaviestejä ja soittaa nettipuheluja halutulla asiakasohjelmalla. Tämän lisäksi palvelinsovelluksien asennuksesta ja käyttöönotosta tuli tehdä helppolukuinen käyttöohje, jonka avulla palvelin olisi helppo rakentaa uudestaan. Tästä syystä työ painottuu suurimmaksi osaksi käytäntöön, ja teoriaosuus rajoittuu vain käytettyjen protokollien lyhyeen esittelyyn.

Jo työn alkuvaiheessa yllätyimme, kuinka helppoa yksinkertaisen viestintäpalvelimen rakentaminen on, sillä pelkillä perusominaisuuksilla varustettu palvelin voidaan ottaa käyttöön muutamassa kymmenessä minuutissa. Toisaalta monipuolisemman palvelimen käyttöönotossa voi mennä useita päiviä, jotta kaikki ominaisuudet saadaan toimimaan saumattomasti yhdessä.

Ainoiksi merkittäviksi ongelmakohdiksi työn aikana osoittautui Kamailiota vaivannut NAT-reititys sekä LDAP-autentikointi. Etsimme useita päiviä vikaa ja ratkaisua ongelmiin omasta palvelimestamme, vaikka syy olikin käytössä olleessa reitittimestä sekä itse SIP-protokollasta. Tästä opimme, että ongelmatilanteissa täytyy ottaa huomioon myös ulkopuoliset tekijät, ja että teoriapuoli on tärkeä osa ongelmanratkaisussa.

Kokonaisuutena työ oli erittäin opettava, sillä se opetti niin Linux-ympäristön käyttöä, ongelmanratkaisua kuin käyttöoppaan kirjoittamistakin. Lisäksi työ oli kaiken kaikkiaan erittäin mielenkiintoinen, ja sitä oli ilo tehdä.

Työ oli onnistunut ja toimeksiannon tavoitteet täytettiin. Ilmaisten avoimen lähdekoodin ratkaisujen sekä selkeän ja helppolukuisen käyttöohjeen vuoksi työstä on hyötyä toimeksiantajan lisäksi kenelle tahansa, joka haluaa rakentaa ilmaisen viestintäpalvelimen.

LÄHTEET

3CX 2013. Mitä ovat SIP-metodit / -pyynnöt ja –vastaukset? Viitattu 2.12.2013 http://www.3cx.fi/voip-sip/sip-methods.

Facebook 2013. Chat API. Viitattu 27.11.2013 https://developers.facebook.com/docs/chat.

IETF 2002. RFC 3261. Viitattu 2.12.2013 https://tools.ietf.org/html/rfc3261.

IETF 2003. RFC 3550. Viitattu 2.12.2013 http://www.ietf.org/rfc/rfc3550.txt.

IETF 2006. RFC 4566. Viitattu 2.12.2013 https://tools.ietf.org/html/rfc4566.

IETF 2011. RFC 6120. Viitattu 27.11.2013 http://tools.ietf.org/html/rfc6120.

Johnston, A. 2009. SIP: Understanding the Session Initation Protocol. 3. edition. Norwood, Massachusetts: Artech House.

LTE University 2012. The VOLTE “Conversation” Between IMS and LTE. Viitattu 2.12.2013 http://lteuniversity.com/get_trained/expert_opinion1/b/bbest/archive/2012/12/17/the-volte-conversation-between-ims-and-lte.aspx.

Matthews, D. 2012. SIP – Example Simple Two-Party Call Flow. Viitattu 2.12.2013 http://mccltd.net/blog/?p=906.

Saint-Andre, P. 2010. XEP-0001: XMPP Extension Protocols. Viitattu 27.11.2013 http://xmpp.org/extensions/xep-0001.html.

Shaham G., 2009. Wired for sound: how SIP won the VoIP protocol wars. Viitattu 2.12.2013 http://arstechnica.com/business/2009/12/wired-for-sound-how-sip-won-the-voip-protocol-wars.

SIP URI 2011. VoIP.ms Wiki. Viitattu 2.12.2013 http://wiki.voip.ms/article/SIP_URI.

Toncar, V. 2013. VoIP Protocols: SIP – Session Description Protocol. Viitattu 2.12.2013 http://toncar.cz/Tutorials/VoIP/VoIP_Protocols_SIP_Session_Description_Protocol.html.

XMPP Standards Foundation 2013a. About XMPP. Viitattu 27.11.2013 http://xmpp.org/about-xmpp.

XMPP Standards Foundation 2013b. XMPP Technologies Overview. Viitattu 27.11.2013 http://xmpp.org/about-xmpp/technology-overview.

XMPP Standards Foundation 2013c. XMPP Extensions. Viitattu 27.11.2013 http://xmpp.org/xmpp-protocols/xmpp-extensions.

XMPP Standards Foundation 2013d. History. Viitattu 27.11.2013. http://xmpp.org/about-xmpp/history.

MariaDB:n asennus CentOS-käyttöjärjestelmään

Luodaan MariaDB:n repo-tiedosto

```
file=/etc/yum.repos.d/mariadb.repo

cat >$file<<"EOF"

[mariadb]

name=MariaDB

baseurl=http://yum.mariadb.org/10.0/centos6-amd64

gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB

gpgcheck=1

enabled=1

EOF
```

Ladataan MariaDB:n GPG-avain

```
rpm --import https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
```

Asennetaan MariaDB

```
yum install MariaDB-server MariaDB-client
```

Käynnistetään MariaDB, varmistetaan että se on päällä ja asetetaan se käynnistymään automaattisesti koneen kanssa. Tämän jälkeen tehdään ensimmäinen pakollinen konfiguraatio ja käynnistetään MariaDB uudelleen:

```
service mysql start

chkconfig mysql on

mysql_secure_installation

service mysql restart
```

Openfiren asennus CentOS-käyttöjärjestelmään ja käyttöönotto

Siirrytään kansioon, johon rpm-tiedostot ladataan (tässä tapauksessa /var/local/rpm), jonka jälkeen Openfiren paketti ladataan ja asennetaan:

```
cd /var/local/rpm

wget http://download.igniterealtime.org/openfire/openfire-3.8.2-1.i386.rpm

rpm -ivh openfire-3.8.2-1.i386.rpm

cd
```

Ladataan Openfiren vaatimat sovellukset ja kirjastot. Openfire vaatii 32-bittisen libldb:n koneen bittisyydestä riippumatta:

```
yum install java libldb.i686
```

Käynnistetään Openfire, asetetaan se käynnistymään automaattisesti koneen kanssa ja varmistetaan, että se on päällä. Openfiren tämänhetkinen versio ei anna OK-signaalia, vaikka käynnistäminen onnistuisikin:

```
service openfire start

chkconfig openfire on

service openfire status
```

Aukaistaan palomuurien asetustiedosto jollain tekstieditorilla, esimerkiksi nanolla:

```
nano /etc/sysconfig/iptables
```

Tämän jälkeen aukaistaan tarvittavat palomuurin portit ja sallitaan kaikki yhteydet halutusta verkosta lisäämällä seuraavat rivit tiedoston ”:OUTPUT ACCEPT [0:0]” -osion loppuun. Ensimmäisen rivin x.x.x.x/yy tilalle määritellään sallitun verkon osoite ja aliverkkomaski:

```
-A INPUT -s x.x.x.x/yy -j ACCEPT
```

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT

-A INPUT -p tcp -i eth0 --dport 5222 -j ACCEPT

-A INPUT -p udp -i eth0 --dport 5222 -j ACCEPT

-A INPUT -p tcp -i eth0 --dport 7777 -j ACCEPT

-A INPUT -p udp -i eth0 --dport 7777 -j ACCEPT

-A INPUT -p tcp -i eth0 --dport 9090 -j ACCEPT

-A INPUT -p udp -i eth0 --dport 9090 -j ACCEPT

-A INPUT -p tcp -i eth0 --dport 9091 -j ACCEPT

-A INPUT -p udp -i eth0 --dport 9091 -j ACCEPT
```

Tämän jälkeen käynnistetään palomuuri edelleen:

```
service iptables restart
```

Openfire on nyt asennettu. Seuraavaksi käydään läpi käyttöönotto joko MySQL- tai LDAP-autentikoinnilla. Molempia ei ole mahdollista käyttää samaan aikaan. MySQL-autentikoinnissa käytetään Openfirelle luotavaa tietokantaa, johon tallennetaan käyttäjille luotavat tilit. LDAP-autentikoinnissa käytetään Active Directory -palvelinta, jolloin käyttäjille ei tarvitse erikseen luoda tiliä.

Esiasennus MySQL-autentikoinnilla.

Luodaan Openfirelle tietokanta ja annetaan käyttäjänimelle openfire käyttöoikeudet siihen salasanalla openfire, jonka jälkeen tuodaan Openfiren valmis tietokantapohja:

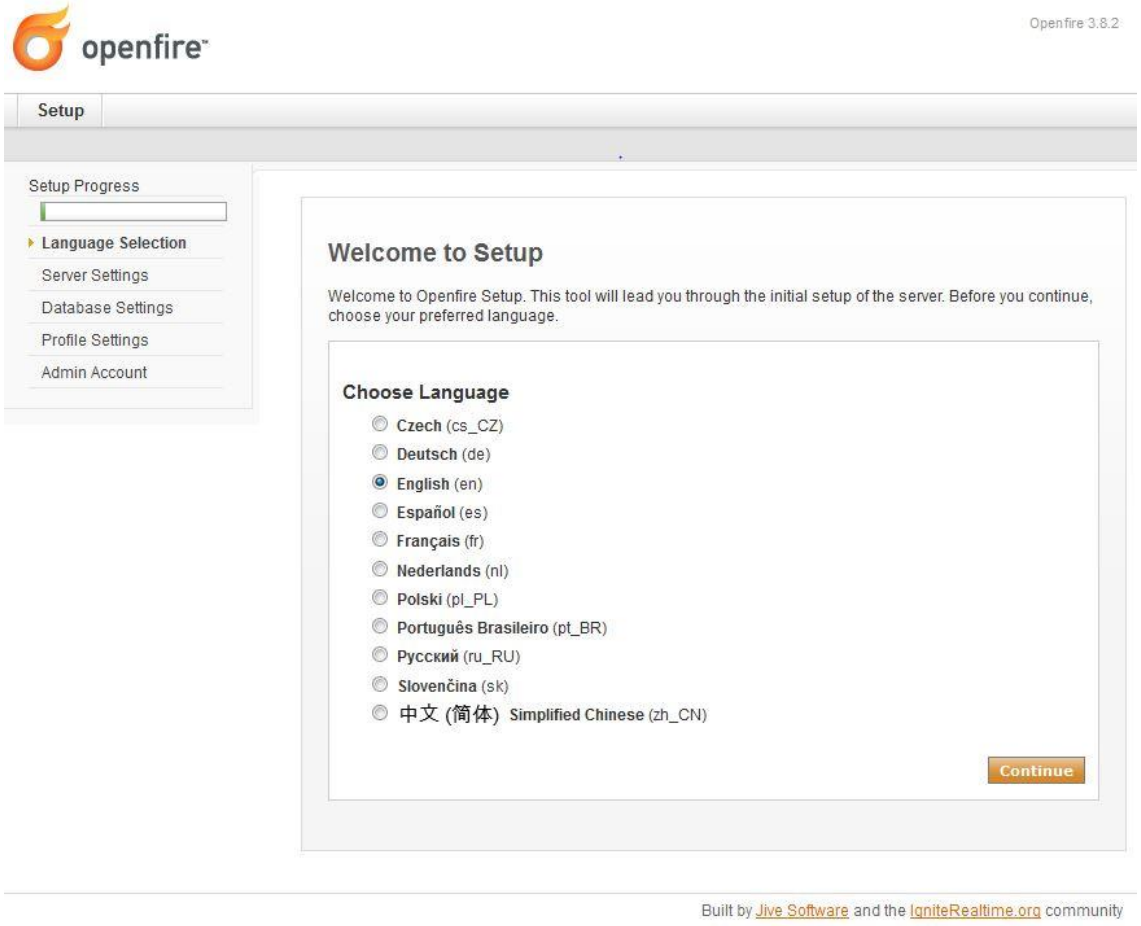
```
mysql -u root -e "CREATE DATABASE openfire;"

mysql -u root -e "GRANT ALL PRIVILEGES ON openfire.* TO openfire@localhost IDENTIFIED BY 'openfire';"

mysql -u root openfire < /opt/openfire/resources/database/openfire_mysql.sql
```

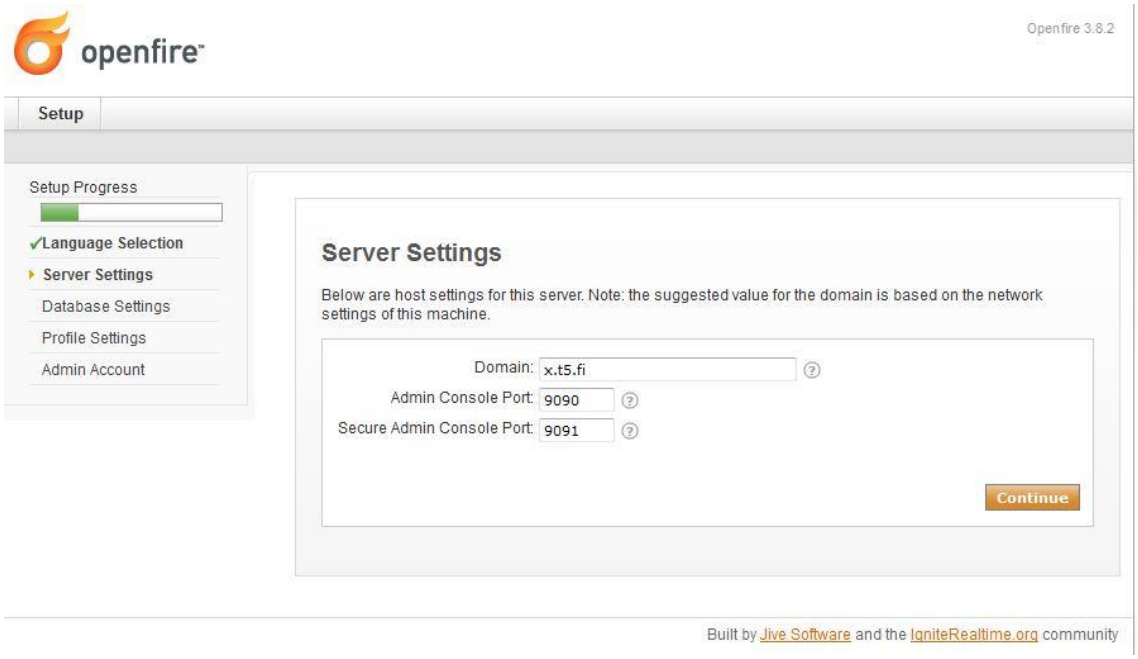
Aukaistaan selaimella web-käyttöliittymä osoitteesta https://x.x.x.x:9091, jossa x.x.x.x on palvelimen osoite tai domain. Selain saattaa ilmoittaa puuttuvasta sertifikaatista, tämä voidaan hyväksyä.

Ensimmäiseksi valitaan käyttöliittymän kieli (Kuva 1).



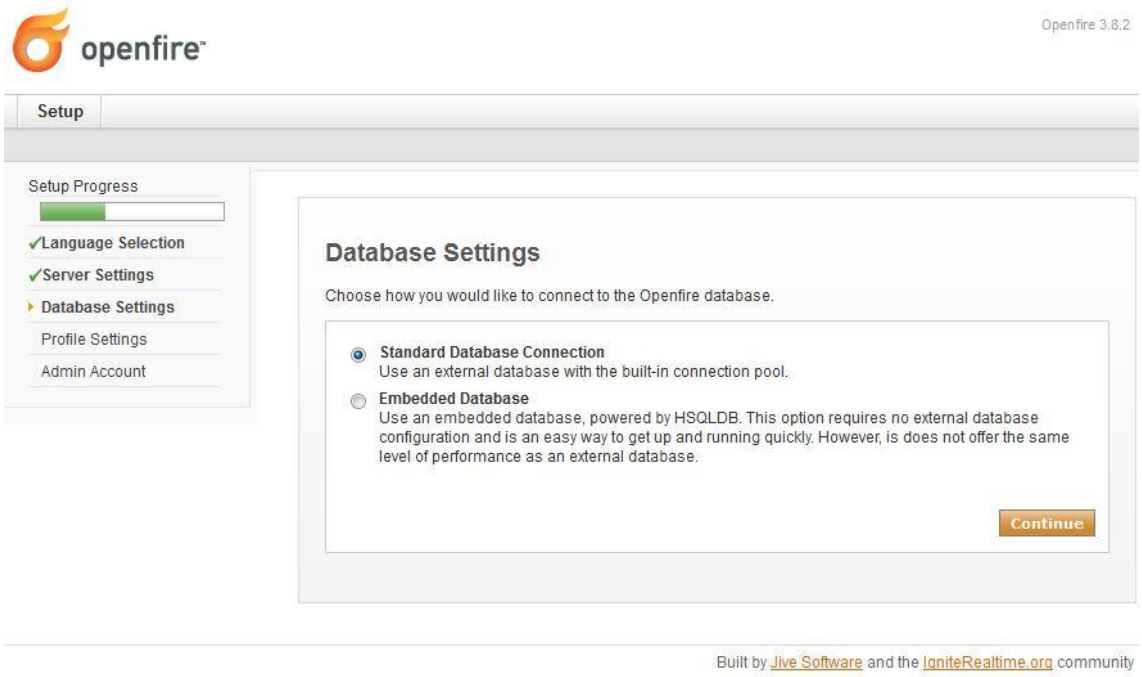
Kuva 1. Käyttöliittymän kielen valinta.

Seuraavaksi määritellään palvelimen asetukset. Domain-kenttään kirjoitetaan palvelimen käytössä oleva domain. Oletuksena Openfiren käyttöliittymään pääsee portista 9090 ja salatulla liikenteellä portista 9091. Nämä asetukset voi jättää oletuksiksi (Kuva 2).



Kuva 2. Palvelimen asetukset.

Tämän jälkeen määritellään tietokannan yhdistämistapa. Openfire tallentaa osan asetuksistaan tietokantaan riippumatta siitä, tallennetaanko käyttäjätunnukset sinne. Käytetään tässä tapauksessa Standard Database Connection -asetusta (Kuva 3).

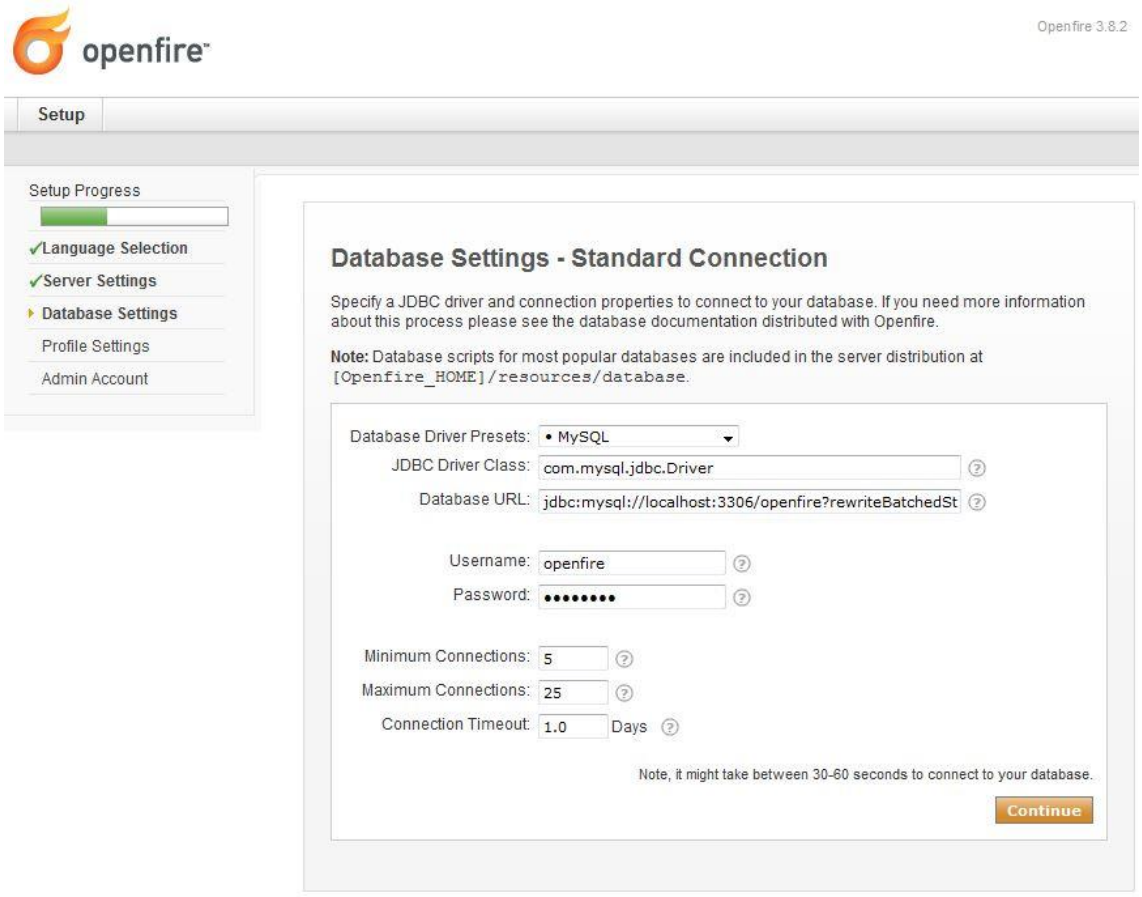


Kuva 3. Tietokannan yhdistämistavan valinta.

Tämän jälkeen valitaan Database Driver Presets -valikosta MySQL, jolloin Openfire hakee automaattisesti oikean ajurin. Database URL -kenttä määrittelee osoitteen, jolla Openfire yhdistää tietokantaan. Oletusarvoisesta osoitteesta kohdat [host-name] ja [database-name] korvataan tietokannan osoitteella ja nimellä. Tässä tapauksessa tietokanta sijaitsee samalla palvelimella kuin Openfire ja tietokannan nimi on openfire, jolloin Database URL -kenttä määritellään seuraavanlaiseksi:

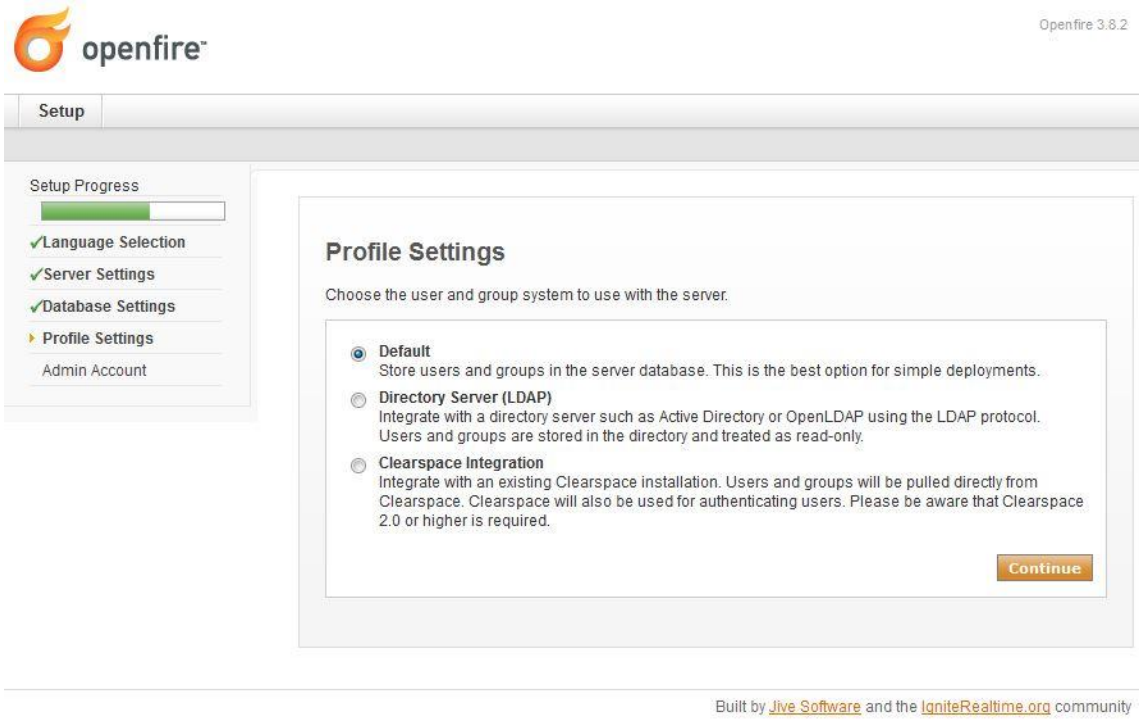
jdbc:mysql://localhost:3306/openfire?rewriteBatchedStatements=true

Username ja password -kenttiin määritellään Openfirelle annettu MySQL-käyttäjänimi. Tässä tapauksessa sekä käyttäjänimi että salasana ovat openfire. Muut asetukset voi jättää oletuksiksi. (Kuva 4).



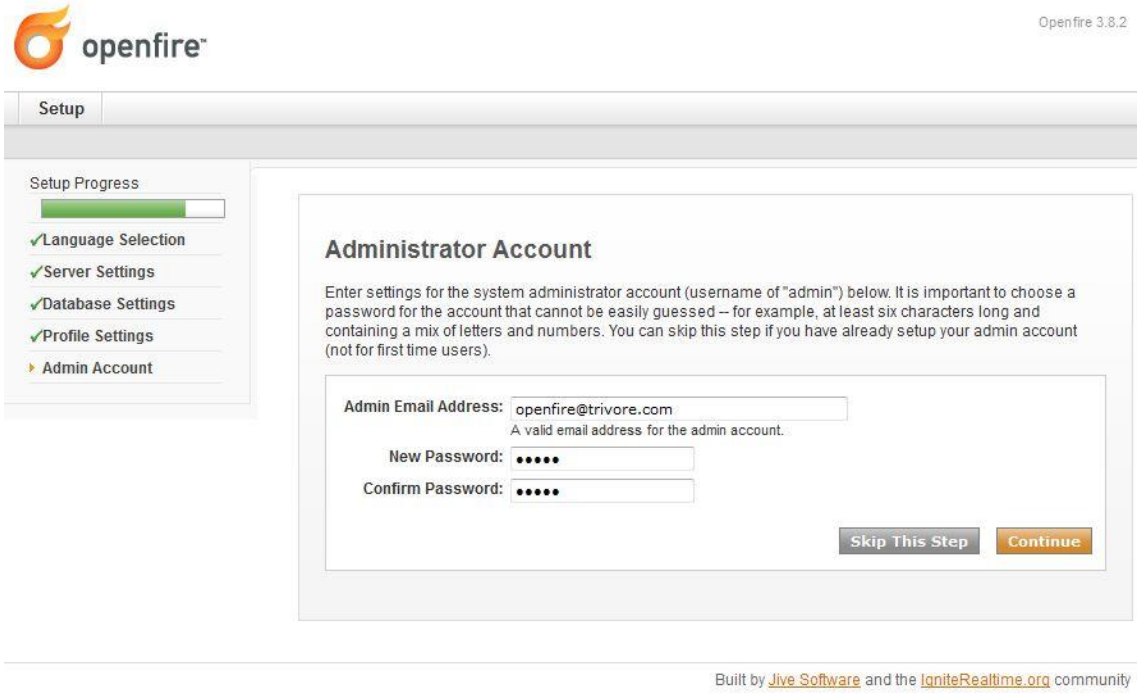
Kuva 4. Tietokannan yhdistämisasetukset.

Seuraavaksi määritellään käyttäjätilien tallennuspaikka. MySQL-tietokantaa varten valitaan Default (Kuva 5).



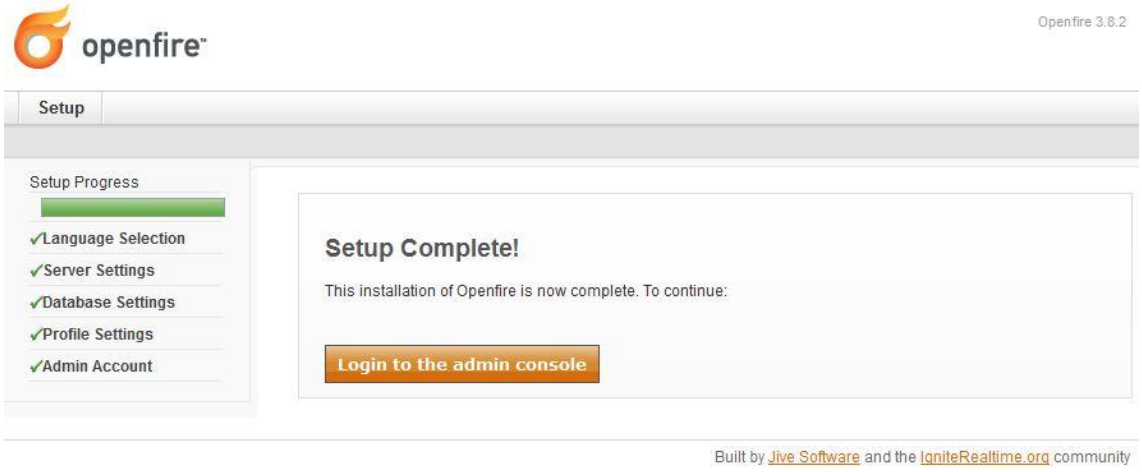
Kuva 5. Käyttäjätilien hallinta-asetukset.

Lopuksi määritellään järjestelmänvalvojan sähköpostiosoite ja salasana. Järjestelmänvalvojan käyttäjänimi on admin, eikä sitä voi muuttaa (Kuva 6).



Kuva 6. Järjestelmänvalvojan asetukset.

Tämän jälkeen Openfiren käyttöliittymä on käyttövalmis. Kirjautumisikkunaan siirrytään painamalla oranssia painiketta (Kuva 7).



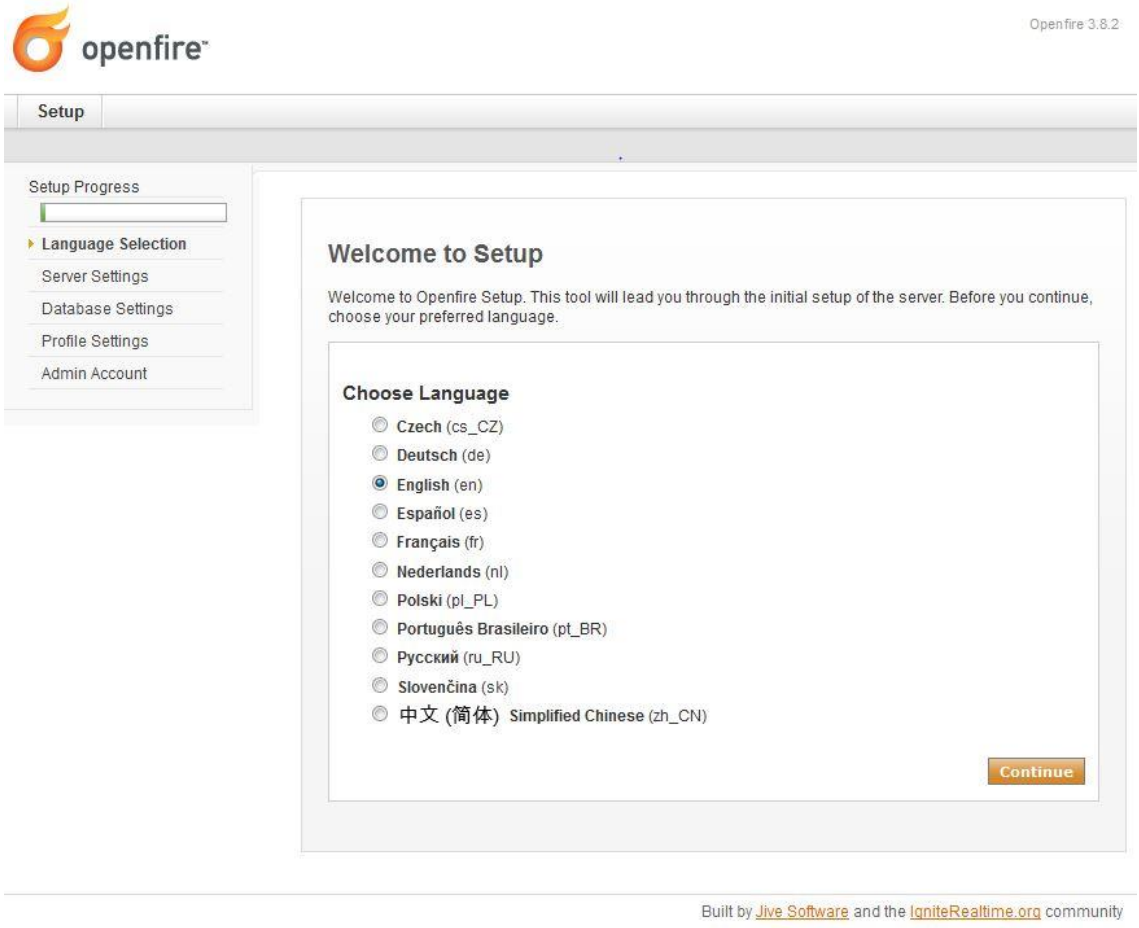
Kuva 7. Openfire käyttövalmiina.

Esiasennus MySQL-autentikoinnilla on nyt valmis.

Esiasennus LDAP-autentikoinnilla.

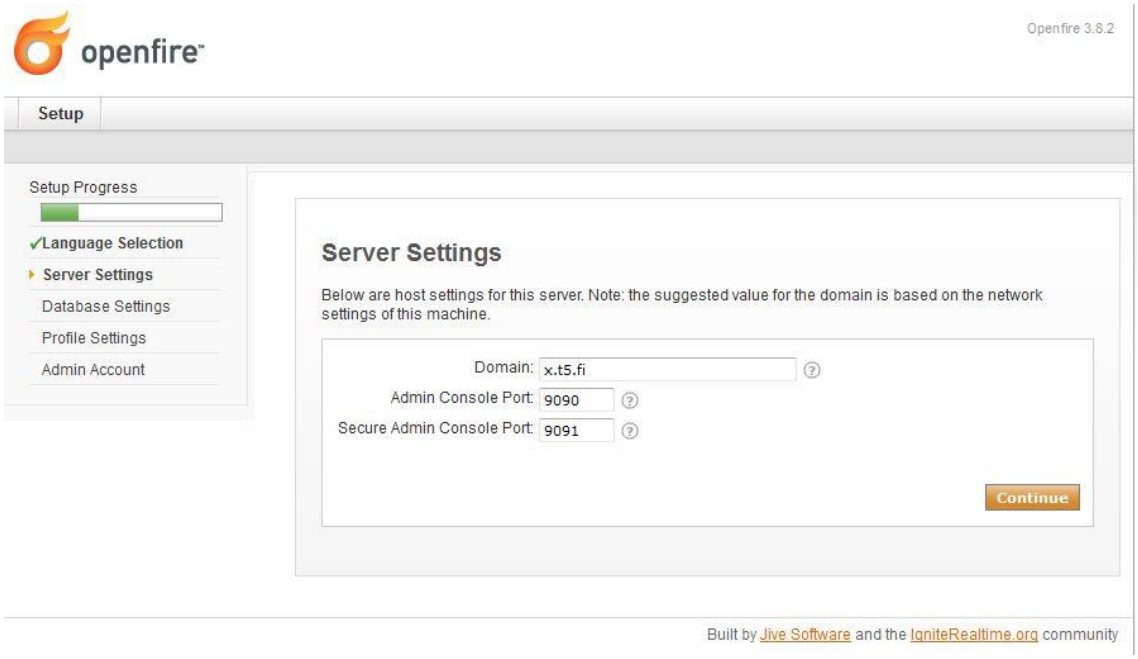
Aukaistaan selaimella web-käyttöliittymä osoitteesta https://x.x.x.x:9091, jossa x.x.x.x on palvelimen osoite tai domain. Selain saattaa ilmoittaa puuttuvasta sertifikaatista, tämä voidaan hyväksyä.

Ensimmäiseksi valitaan käyttöliittymän kieli (Kuva 8).



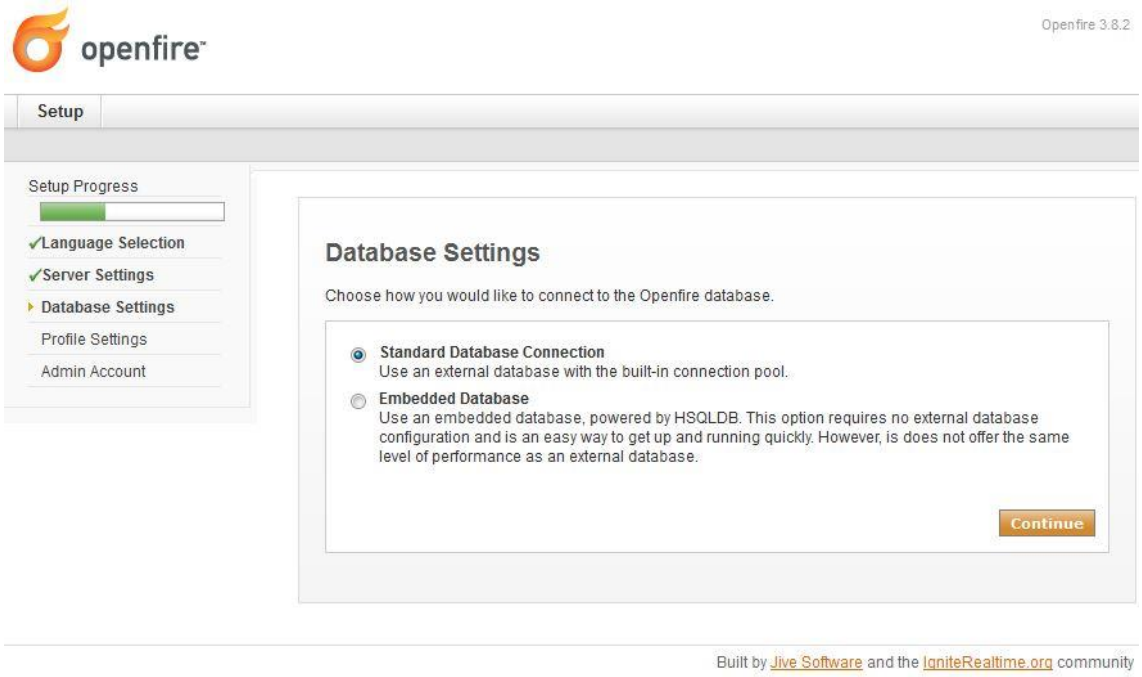
Kuva 8. Käyttöliittymän kielen valinta.

Seuraavaksi määritellään palvelimen asetukset. Domain-kenttään kirjoitetaan palvelimen käytössä oleva domain. Oletuksena Openfiren käyttöliittymään pääsee portista 9090 ja salatulla liikenteellä portista 9091 (Kuva 9).



Kuva 9. Palvelimen asetukset.

Tämän jälkeen määritellään tietokannan yhdistämistapa. Openfire tallentaa osan asetuksistaan tietokantaan riippumatta siitä, tallennetaanko käyttäjätunnukset sinne. Käytetään tässä tapauksessa Standard Database Connection -asetusta (Kuva 10).

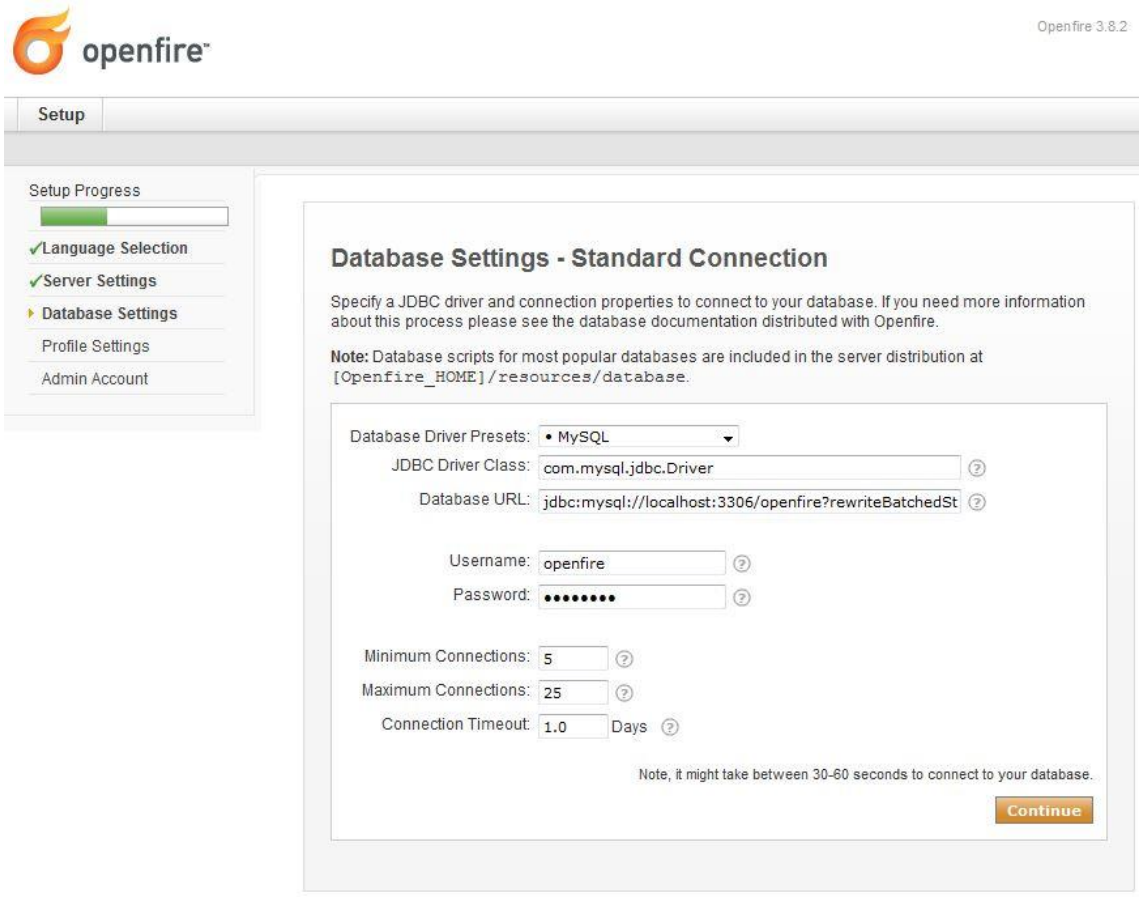


Kuva 10. Tietokannan yhdistämistavan valinta.

Tämän jälkeen valitaan Database Driver Presets -valikosta MySQL, jolloin Openfire hakee automaattisesti oikean ajurin. Database URL -kenttä määrittelee osoitteen, jolla Openfire yhdistää tietokantaan. Oletusarvoisesta osoitteesta kohdat [host-name] ja [database-name] korvataan tietokannan osoitteella ja nimellä. Tässä tapauksessa tietokanta sijaitsee samalla palvelimella kuin Openfire ja tietokannan nimi on openfire, jolloin Database URL -kenttä määritellään seuraavanlaiseksi:

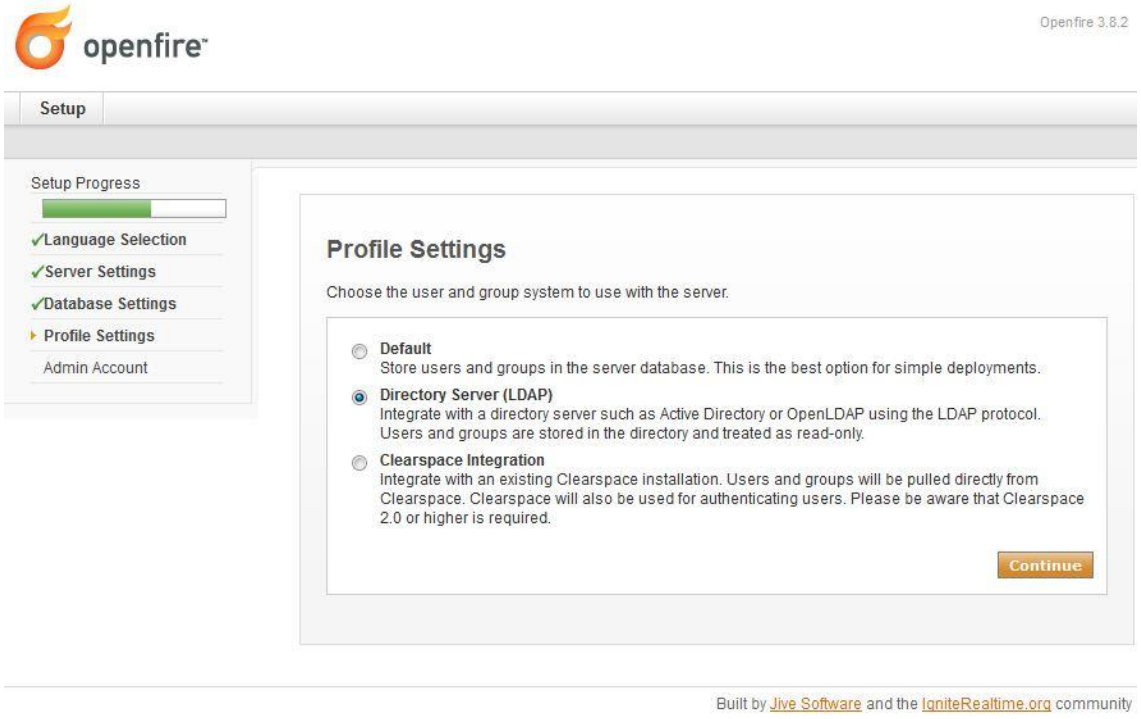
jdbc:mysql://localhost:3306/openfire?rewriteBatchedStatements=true

Username ja password -kenttiin määritellään Openfirelle annettu MySQL-käyttäjänimi. Tässä tapauksessa sekä käyttäjänimi että salasana ovat openfire. Muut asetukset voi jättää oletuksiksi (Kuva 11).



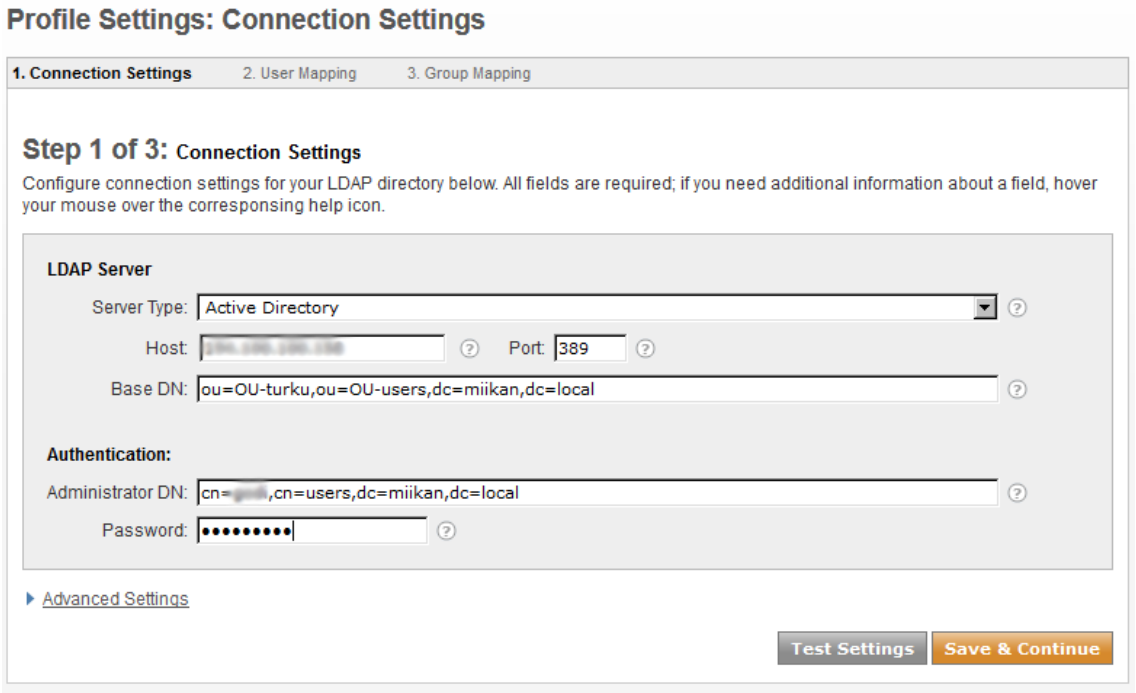
Kuva 11. Tietokannan yhdistämisasetukset.

Seuraavaksi määritellään käyttäjätilien tallennuspaikka. Koska käyttäjätilit sijaitsevat Active Directory -palvelimella, valitaan Directory Server (LDAP) (Kuva 12).



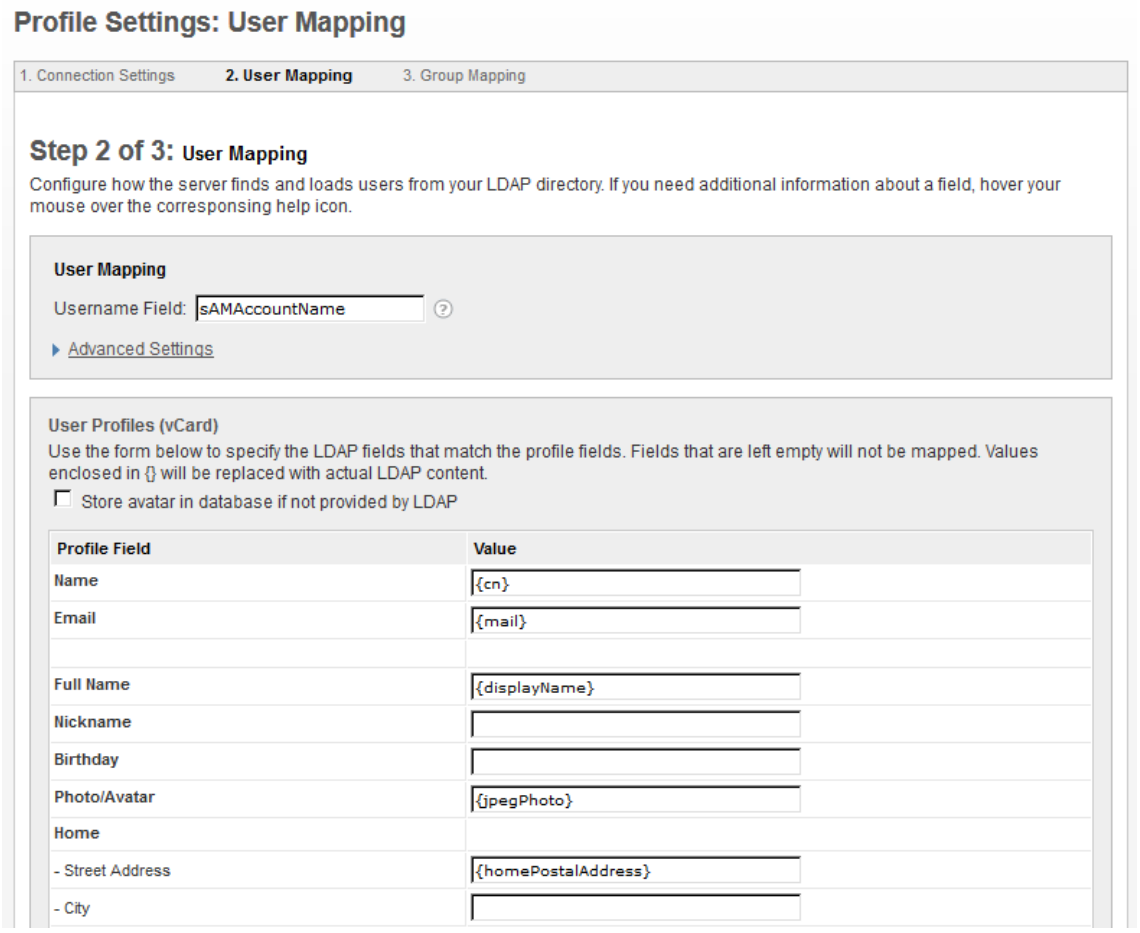
Kuva 12. Käyttäjätilien hallinta-asetukset.

Seuraavaksi määritellään LDAP-yhdistämisen asetukset. Server Type -valikosta valitaan Active Directory ja Host -kenttään määritellään AD-palvelimen osoite. Base DN -kenttään määritellään käyttäjien sijainti AD:ssa. Administrator DN-kenttään määritellään AD-palvelimen järjestelmänvalvojan käyttäjänimi ja alle käyttäjän salasana (Kuva 13).



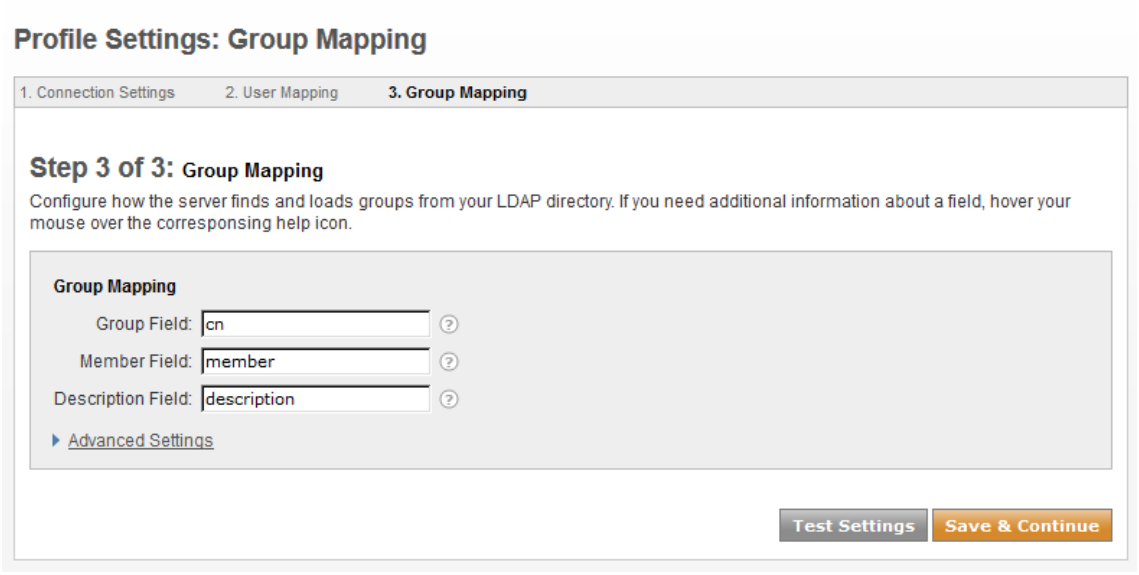
Kuva 13. AD-käyttäjätilien sijaintiasetukset.

Tämän jälkeen määritellään, mitä tietoja LDAP hakee AD:sta. Useimmissa tapauksissa asetukset voidaan jättää oletuksiksi (Kuva 14).



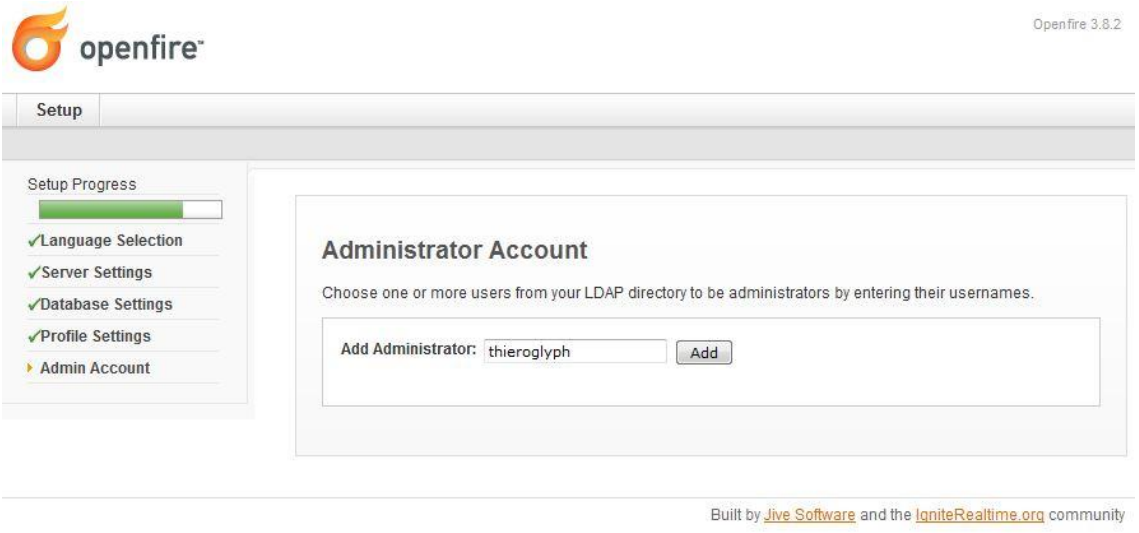
Kuva 14. Käyttäjätilien tietojenhaun asetukset.

Seuraavaksi määritellään ryhmien kartoitus. Tämäkin voidaan useimmissa tapauksissa jättää oletusasetuksiksi (Kuva 15).



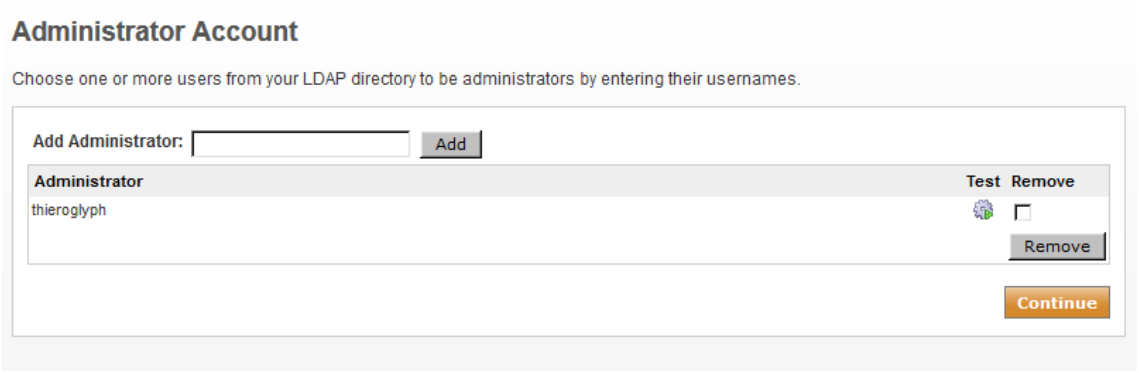
Kuva 15. Käyttäjätilien kartoitus.

Tämän jälkeen määritellään jokin AD-palvelimen käyttäjä Openfiren järjestelmänvalvojaksi. Kenttään kirjoitetaan halutun käyttäjän käyttäjänimi (Kuva 16).



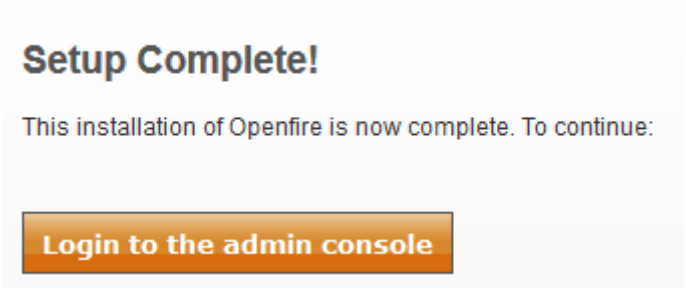
Kuva 16. Openfiren järjestelmänvalvojan lisäys.

Mikäli järjestelmänvalvojia haluaa lisätä useampia, tehdään se samalla tavalla kuin aikaisemmassakin vaiheessa. Mikäli useampia järjestelmänvalvojia ei tarvita, voidaan siirtyä seuraavaan vaiheeseen (Kuva 17).



Kuva 17. Lista Openfiren tämänhetkisistä järjestelmänvalvojista.

Kirjautumisikkunaan siirrytään painamalla oranssia painiketta (Kuva 18).

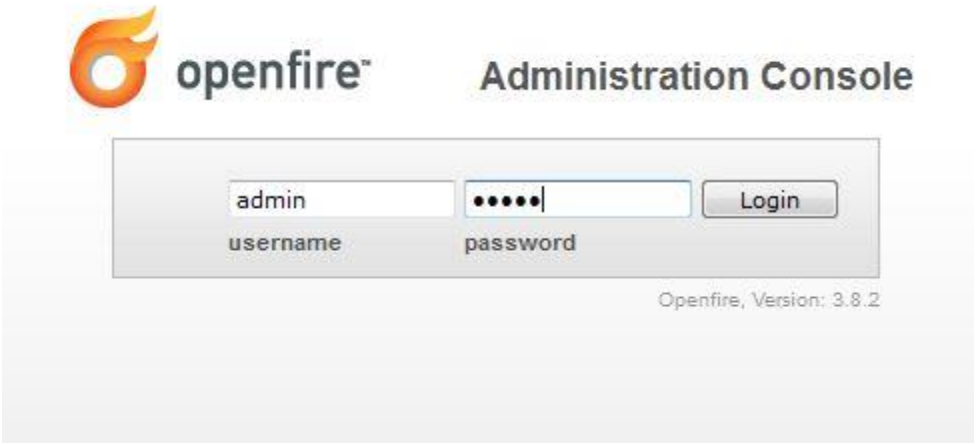


Kuva 18. Onnistunut esiasennus.

Esiasennus LDAP-autentikoinnilla on nyt valmis.

Asetuksien määrittäminen MySQL- ja LDAP-autentikoinnilla

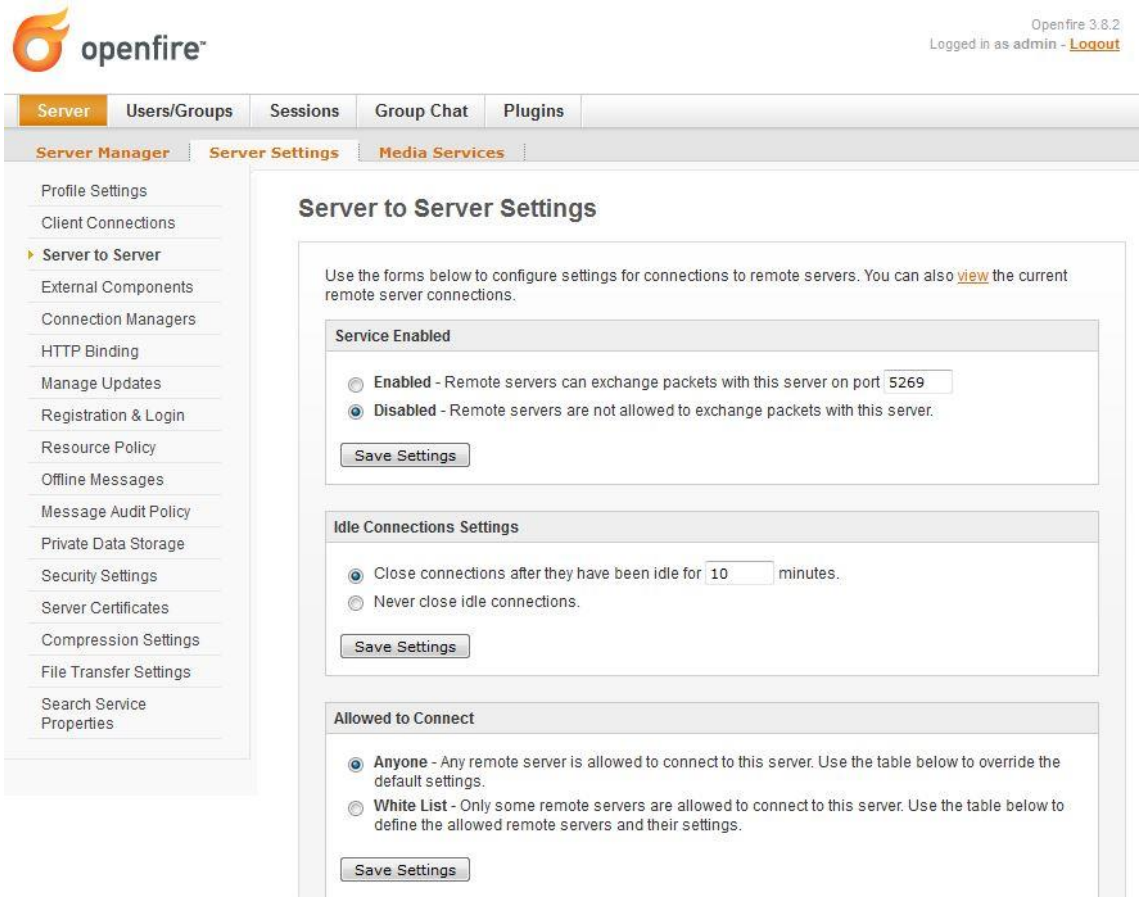
Sisäänkirjautuminen tapahtuu MySQL-autentikointia käytettäessä käyttäjänimellä admin, salasanan ollessa se, mikä esiasennuksen yhteydessä on määritelty. LDAP-autentikointia käytettäessä käyttäjänimi on esiasennuksen yhteydessä määritetty järjestelmänvalvoja, joka käyttää käyttäjätunnuksensa omaa salasanaa (Kuva 19).



Kuva 19. Openfiren käyttöliittymän kirjautumisikkuna.

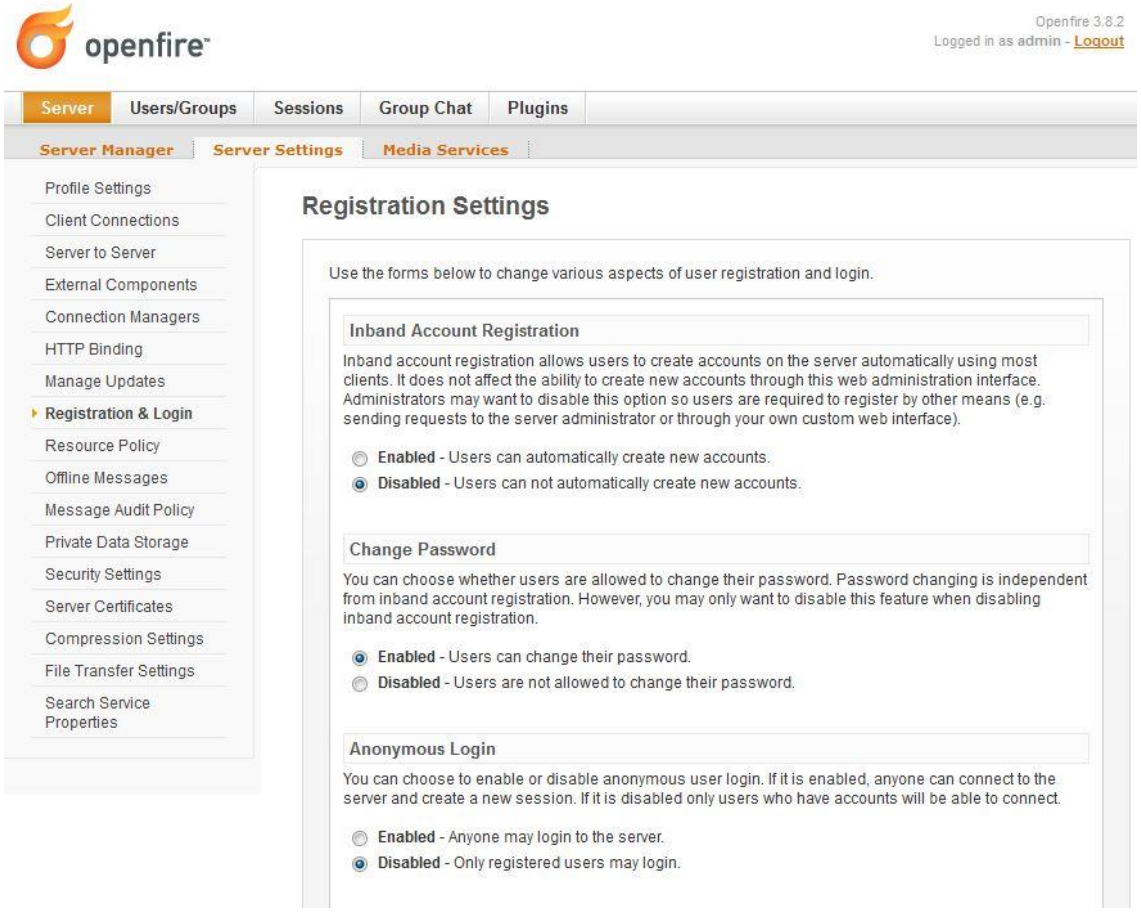
Kirjautumisen jälkeen ensimmäisenä avautuu yleisnäkymä käyttöliittymään, josta nähdään muun muuassa palvelimen päälläoloaika, Openfiren versio sekä käytetyn muistin määrä.

Turvallisuuden kannalta tärkeimmät asetukset säädetään heti ensimmäisellä kirjautumiskerralla. Ensiksi estetään muita Openfire-palvelimia yhdistämästä tälle palvelimelle. Painamalla käyttöliittymän yläosasta Server Settings päästään palvelimen asetuksiin. Sivupalkista valitaan Server to Server, jonka Service Enabled -vaihtoehto vaihdetaan muotoon Disabled. Tämän jälkeen painetaan Save Settings, jotta asetukset tallentuvat (Kuva 20).



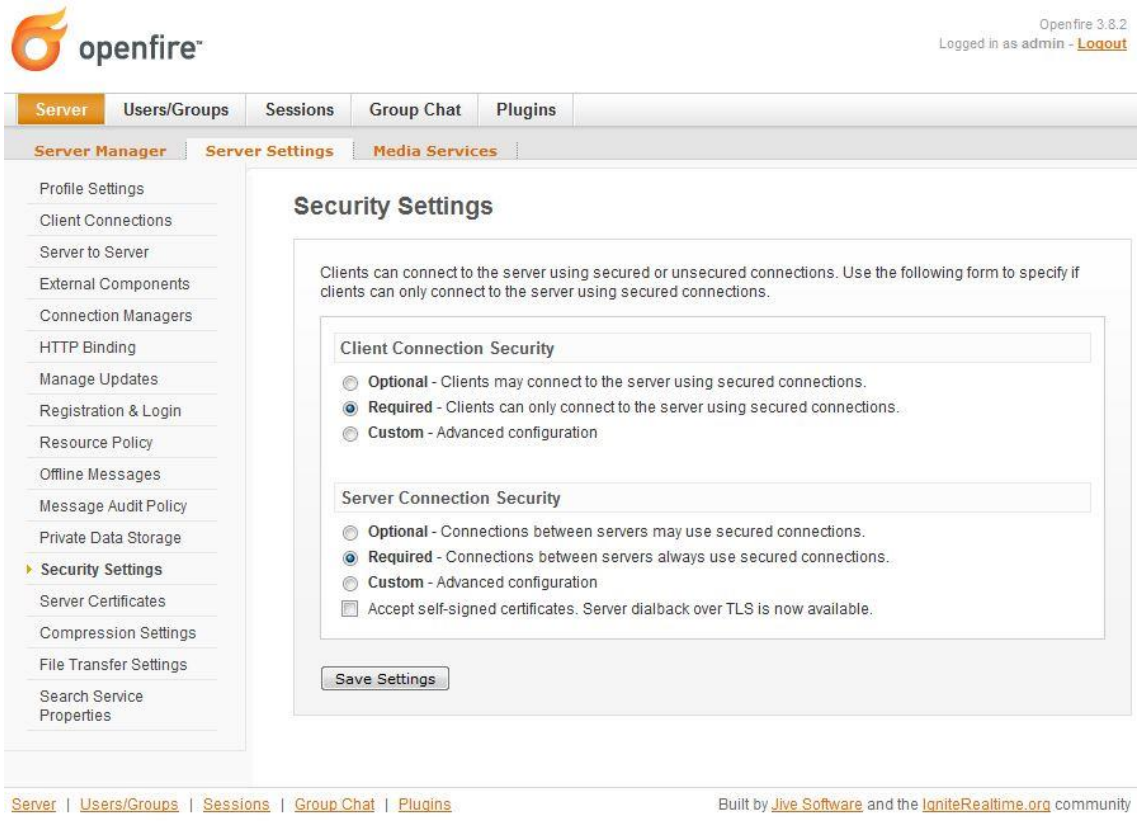
Kuva 20. Palvelinyhteyksien asetukset.

Seuraavaksi estetään tuntemattomia käyttäjiä luomasta käyttäjätilejä palvelimelle ja kirjautumasta sisään. Valitaan Registration & Login, jossa valitaan Disabled kohtiin Inband Account Registration sekä Anonymous Login. Muutosten jälkeen asetukset tallennetaan painamalla Save Settings -painiketta (Kuva 21).



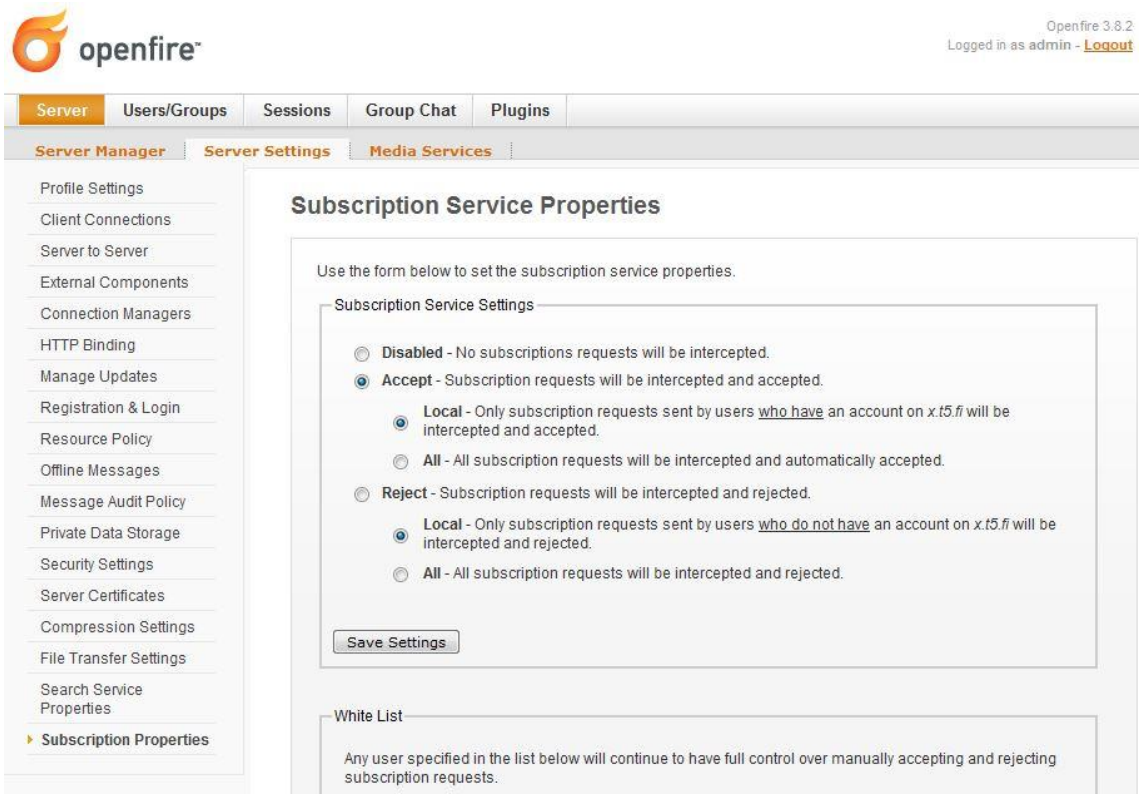
Kuva 21. Käyttäjien rekisteröintiasetukset.

Seuraavaksi sallitaan Openfireen yhdistäminen vain suojatuilla yhteyksillä. Valitaan sivupalkista Security Settings, ja vaihdetaan molempiin kohtiin Required. Muutosten jälkeen asetukset tallennetaan painamalla Save Settings -painiketta (Kuva 22).



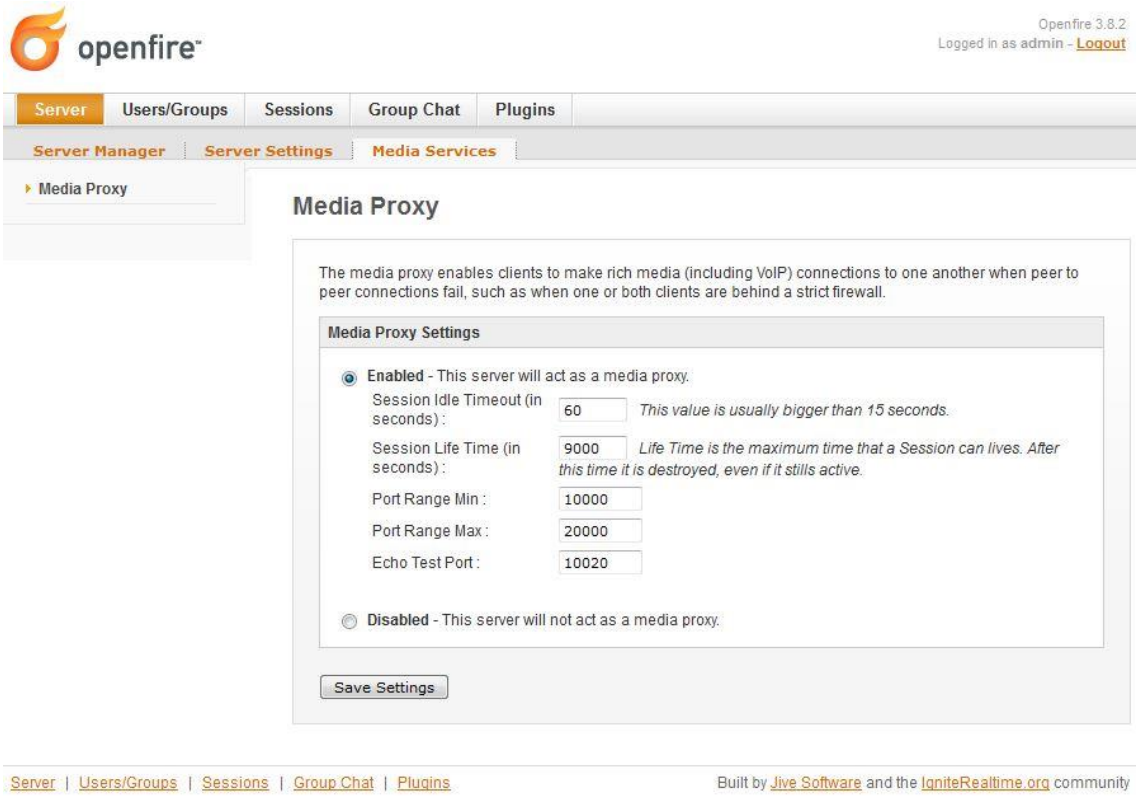
Kuva 22. Openfireen yhdistämisen suojausasetukset.

Lopuksi sallitaan käyttäjien lisätä toisiaan kontaktiluetteloihinsa. Valitaan sivupalkista Subscription Properties ja valitaan Accept -kenttä. Valitsemalla Local hyväksytään vain samalla palvelimella olevien käyttäjien kontaktikutsujen lähetys ja vastaanotto, kun taas All antaa käyttäjien lähettää ja vastaanottaa kutsuja keneltä tahansa. Muutosten jälkeen asetukset tallennetaan painamalla Save Settings -painiketta (Kuva 23).



Kuva 23. Käyttäjätilien kontaktipyyntöjen hallinta.

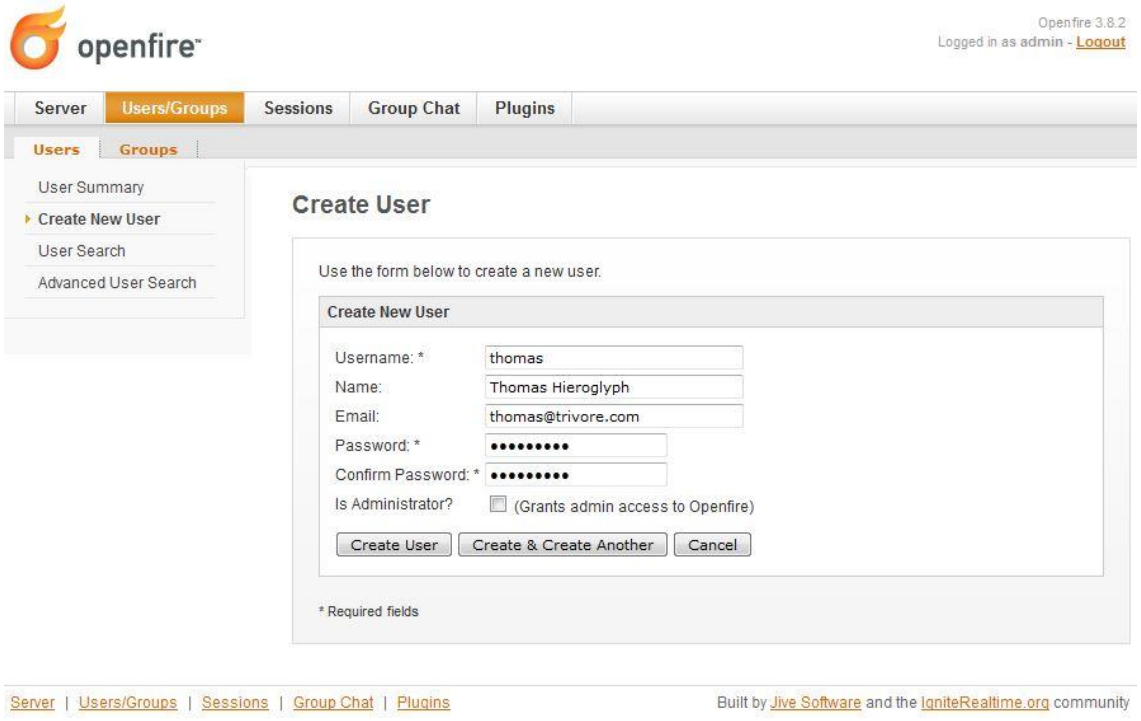
Openfirellä voi halutessaan soittaa VoIP-puheluita tai jakaa työpöytä. Tämän mahdollistamiseksi on otettava käyttöön Media Proxy, mikä tehdään valitsemalla käyttöliitymän yläosasta Media Services, jonka jälkeen valitaan Enabled. Muutosten jälkeen asetukset tallennetaan painamalla Save Settings -painiketta (Kuva 24).



Kuva 24. Media Proxyn asetukset.

LDAP-autentikointia käytettäessä käyttäjätilien hallinnointi tapahtuu AD-palvelimella, joten seuraavaa vaihetta ei tehdä.

MySQL-autentikointia käytettäessä luodaan käyttäjätilit. Valitaan käyttöliittymän yläosasta User/Groups, jonka jälkeen sivupalkista valitaan Create New User. Pakollisia kenttiä ovat Username ja Password, joihin määritellään käyttäjätilin nimi sekä salasana. Käyttäjälle voi halutessaan antaa järjestelmänvalvojan oikeudet Openfiren käyttöliittymään valitsemalla kohdan ”Is Administrator?”. Mikäli käyttäjiä haluaa luoda kerralla useampia, valitaan haluttujen tietojen syöttämisen jälkeen Create & Create Another, muutoin valitaan Create User (Kuva 25).

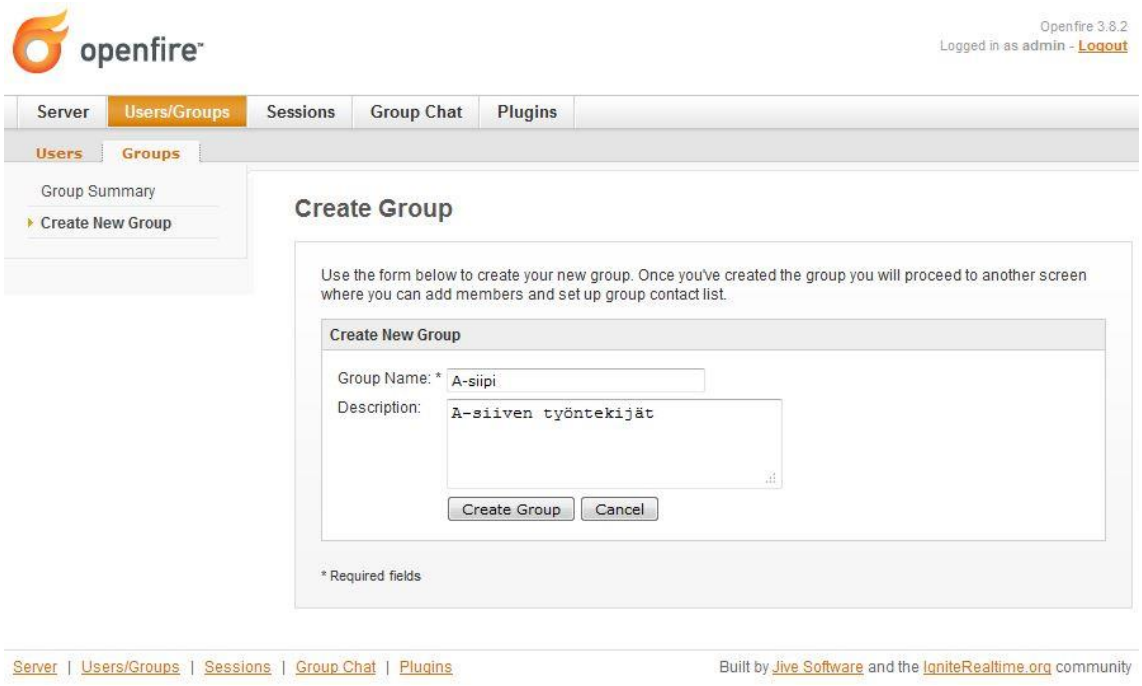


Kuva 25. Käyttäjätilien luonti.

LDAP-autentikointia käytettäessä käyttäjäryhmät haetaan automaattisesti AD -palvelimelta, joten seuraavaa vaihetta ei tehdä.

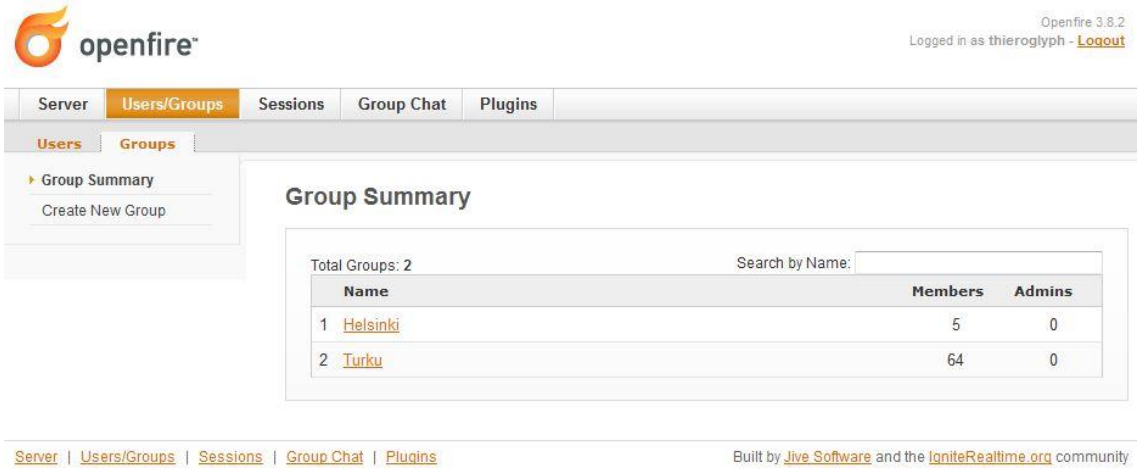
Myös MySQL-autentikointia käytettäessä käyttäjiä on mahdollista lisätä ryhmiin, jolloin ryhmä ja siinä olevat käyttäjät lisätään automaattisesti ryhmässä olevien käyttäjien kontaktilistoihin. Näin käyttäjien ei tarvitse manuaalisesti lisätä toisiaan kontakteihinsa.

Ryhmän luonti tapahtuu valitsemalla käyttöliittymän yläosasta Groups, jonka jälkeen sivupaneelista valitaan Create New Group. Group Name -kenttään määritetään ryhmän nimi, Description -kenttään voidaan määritellä selostus ryhmästä. Tämän jälkeen painetaan Create Group (Kuva 26).



Kuva 26. Käyttäjäryhmän luominen.

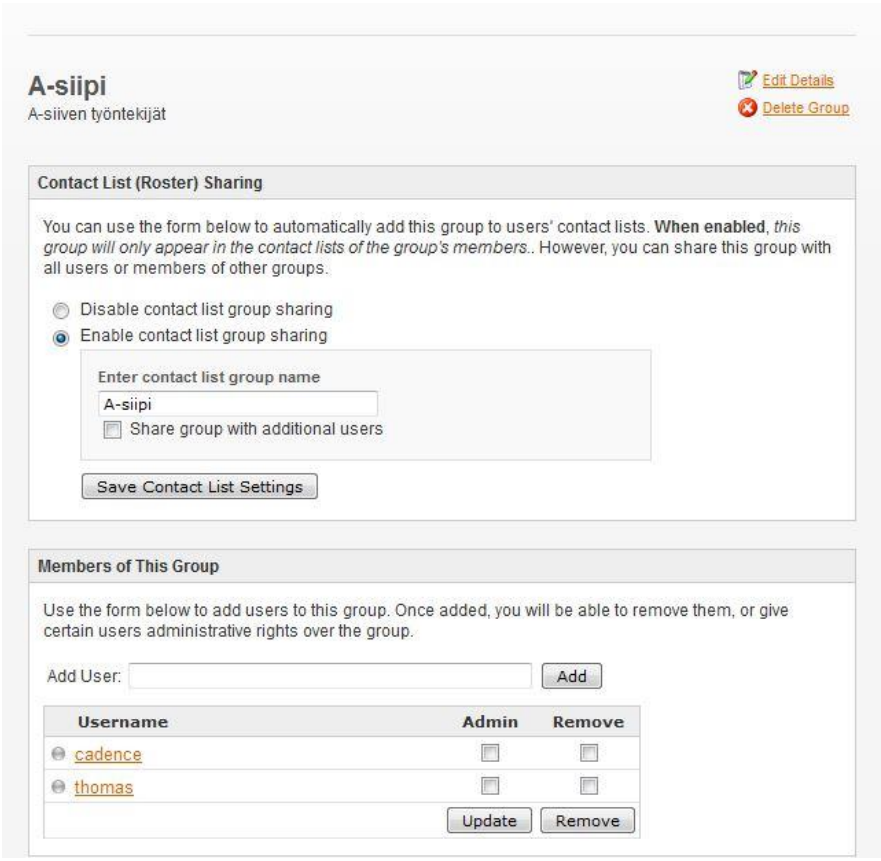
Sekä LDAP- että MySQL-autentikointia käytettäessä ryhmän jäsenten kontaktitiedot on mahdollista jakaa ryhmän jäsenten kesken, jolloin kontaktitietoja ei tarvitse lisätä manuaalisesti. MySQL-ryhmän luomisen jälkeen ryhmän asetusikkuna avautuu automaattisesti. LDAP-autentikaatiota käytettäessä valitaan käyttöliittymän yläosasta Groups, jonka jälkeen sivupaneelista valitaan Group Summary ja painetaan muokattavan ryhmän nimeä (Kuva 27).



Kuva 27. Lista käyttäjäryhmistä.

Tämän jälkeen muokataan ryhmän asetuksia ja lisätään ryhmään käyttäjiä. Contact List (Roster) Sharing -kohdasta valitaan Enable contact list group sharing, jotta ryhmän jäsentiedot jaetaan ryhmän jäsenten kesken. Enter contact list group name -kenttään määritellään nimi, jolla ryhmä näkyy käyttäjien kontaktilistoissa. Muutosten jälkeen asetukset tallennetaan painamalla Save Contact List Settings -painiketta (Kuva 28).

MySQL-autentikointia käytettäessä ryhmään on vielä lisättävä käyttäjät. Halutut käyttäjät lisätään syöttämällä käyttäjätilin nimi Add User -kenttään, jonka jälkeen painetaan Add-painiketta. Käyttäjälle voidaan antaa järjestelmänvalvojan oikeudet ryhmän sisällä valitsemalla käyttäjätilin nimen vieressä olevan Admin-kentän, jonka jälkeen painetaan Update -painiketta. Käyttäjä voidaan poistaa ryhmästä valitsemalla käyttäjätilin nimen vieressä olevan Remove-kentän, jonka jälkeen painetaan Remove-painiketta (Kuva 28).

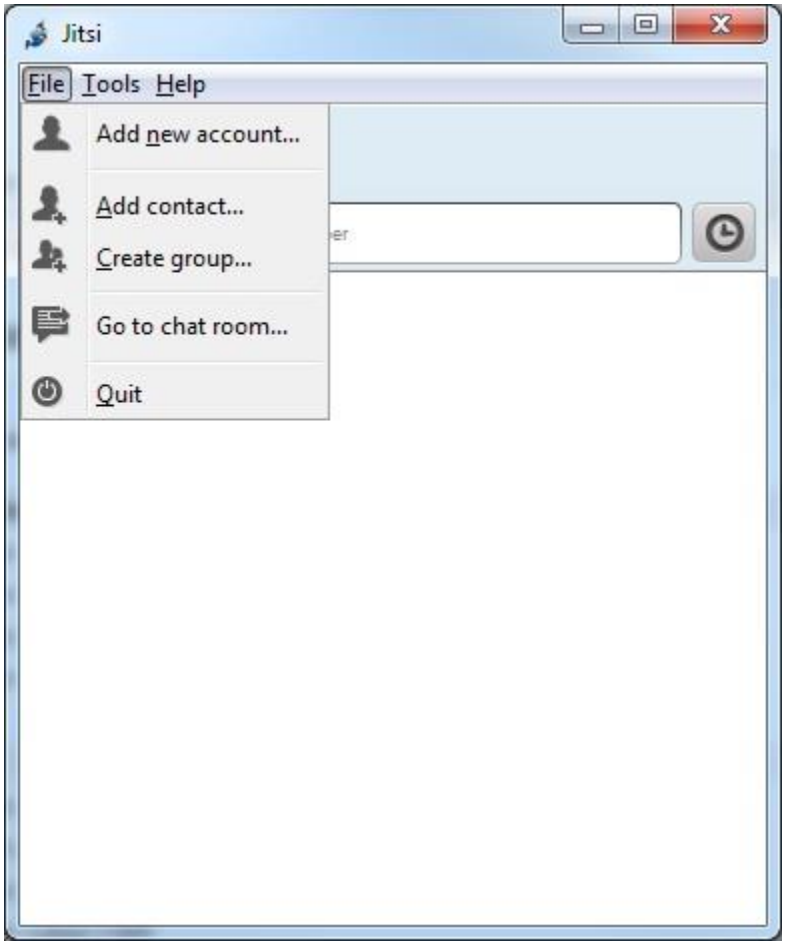


Kuva 28. Käyttäjäryhmän asetukset.

Asetusten, käyttäjätilien sekä ryhmien määrittäminen on nyt valmis. Käyttöliittymästä kirjaudutaan ulos painamalla oikeassa yläkulmassa olevaa Logout-painiketta. Takaisin käyttöliittymään pääsee selaimella osoitteella https://x.x.x.x:9091, jossa x.x.x.x on palvelimen osoite tai domain.

Käyttöönotto Jitsi-asiakasohjelmalla

Käynnistetään Jitsi-asiakasohjelma, jonka jälkeen lisätään käyttäjätili valitsemalla vasemmassa yläkulmassa olevasta File-valikosta Add new account (Kuva 29).

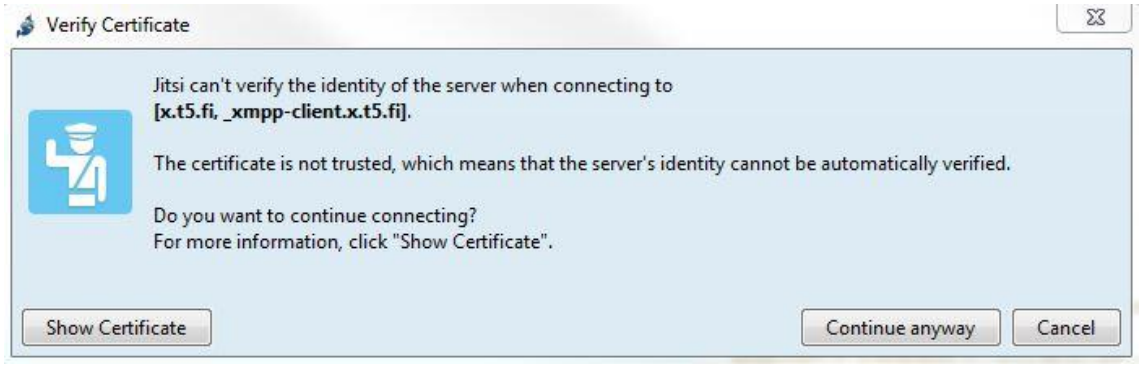


Kuva 29. Uuden käyttäjätilin lisääminen Jitsiin.

Seuraavaksi valitaan Network-valikosta XMPP, jonka jälkeen valitaan Existing XMPP account -kenttä. XMPP username -kenttään kirjoitetaan XMPP-tilin nimi muodossa [käyttäjänimi]@[domain], jossa [käyttäjänimi] on joko tietokannasta (MySQL-autentikointia käytettäessä) tai AD-palvelimelta (LDAP-autentikointia käytettäessä) löytyvä käyttäjänimi ja [domain] Openfiren käytössä oleva domain. Password-kenttään syötetään tilin salasana. Halutessaan Jitsin voi määrittää muistamaan tilin salasanan valitsemalla Remember password -kentän, jolloin sisäänkirjautuminen tapahtuu automaattisesti. Lopuksi painetaan Add-painiketta (Kuva 30).

Kuva 30. Käyttäjätilin tietojen määrittäminen.

Jitsi ilmoittaa, ettei tunnista palvelimen käyttämää sertifikaattia. Tämä johtuu siitä, että oletuksena Openfire käyttää omia sertifikaattejaan. Tämä hyväksytään painamalla Continue anyway -painiketta (Kuva 31).

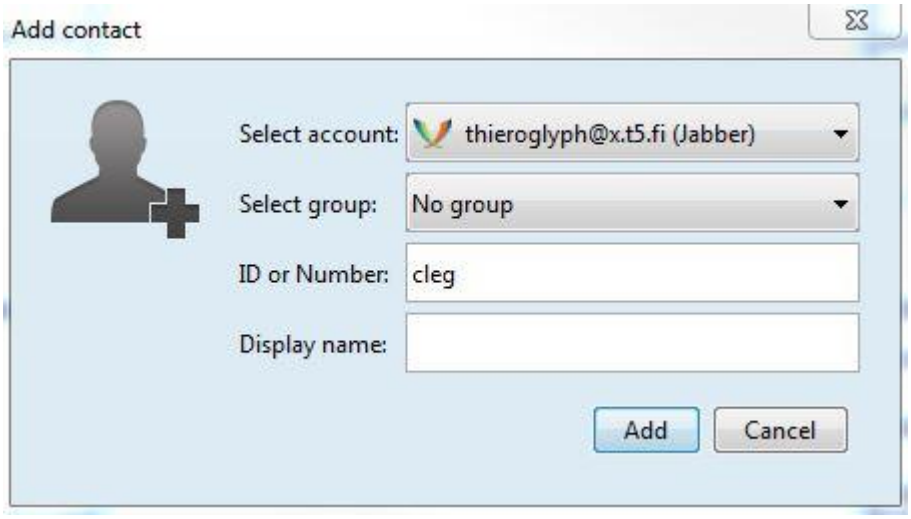


Kuva 31. Jitsin sertifikaatti-ilmoitus.

Käyttäjätili on nyt käyttövalmis. Mikäli käyttäjille on määritetty ryhmä, näkyvät ryhmän jäsenet automaattisesti pääikkunassa heti sisäänkirjautumisen yhteydessä. Muutoin kontaktilista on tyhjä (Kuva 32).

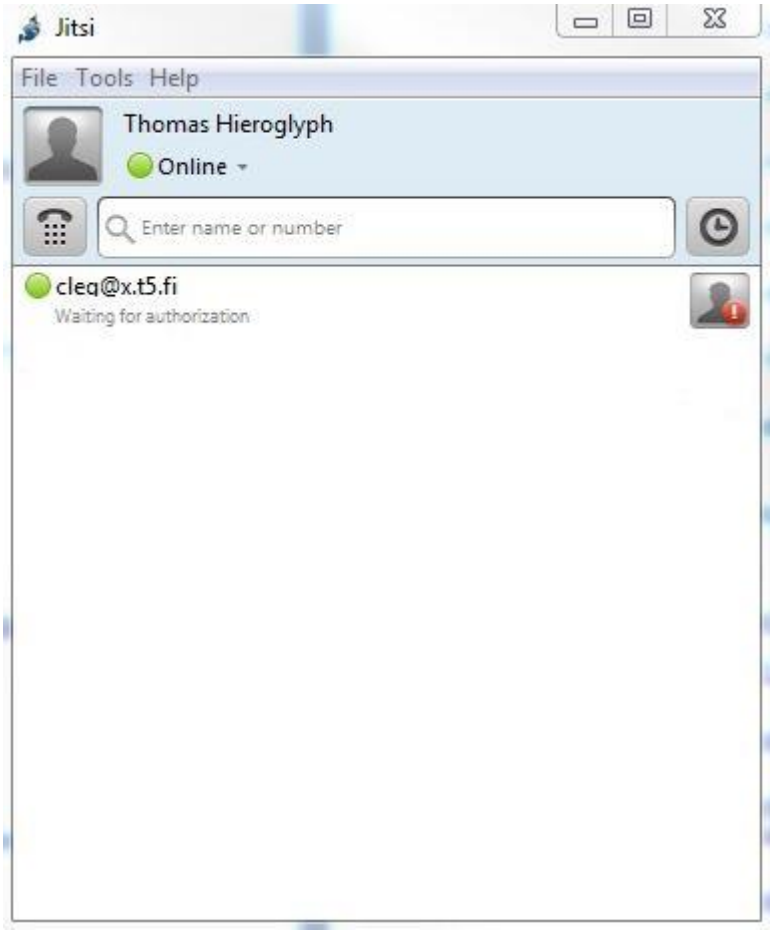
Kuva 32. Jitsin pääikkuna jossa valmiina ryhmän toinen käyttäjä.

Kontaktikutsujen lähettäminen tehdään valitsemalla vasemmassa yläkulmassa olevasta File-valikosta Add contact. Tällöin aukeavasta ikkunasta olevasta Select account -kentästä valitaan käyttäjätili, johon kontakti halutaan lisätä. Select group valikosta on mahdollista valita ryhmä, johon kontakti lisätään. ID or Number -kenttään määritetään halutun kontaktin käyttäjänimi. Display name -kenttään voi halutessaan määrittää kontaktille nimen, jolla kontakti näkyy kontaktiluettelossa (Kuva 33).



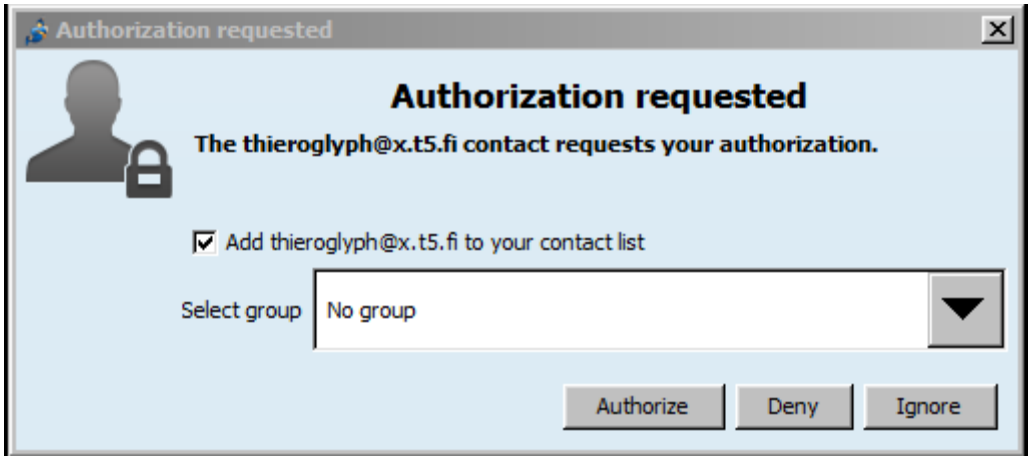
Kuva 33. Kontaktikutsun lähettäminen.

Kun kontaktikutsu on lähetetty, kontakti tulee näkyviin kontaktilistaan Waiting for authorization -tilassa. Tämä tarkoittaa sitä, että kontaktikutsun vastaanottaja ei ole vielä hyväksynyt kontaktikutsua. Kontaktiin ei pysty ottamaan yhteyttä, ennen kuin tämä on hyväksynyt kutsun. (Kuva 34.)



Kuva 34. Uusi kontakti odottaa hyväksyntää.

Kun kontaktikutsu on lähetetty, vastaanottajalle tulee ilmoitus kontaktipyynnöstä. Tämä hyväksytään painamalla Authorize -painiketta (Kuva 35). Kontaktikutsun hyväksynnästä tulee ilmoitus myös kutsun lähettäjälle (Kuva 36).



Kuva 35. Kontaktikutsun vastaanotto.

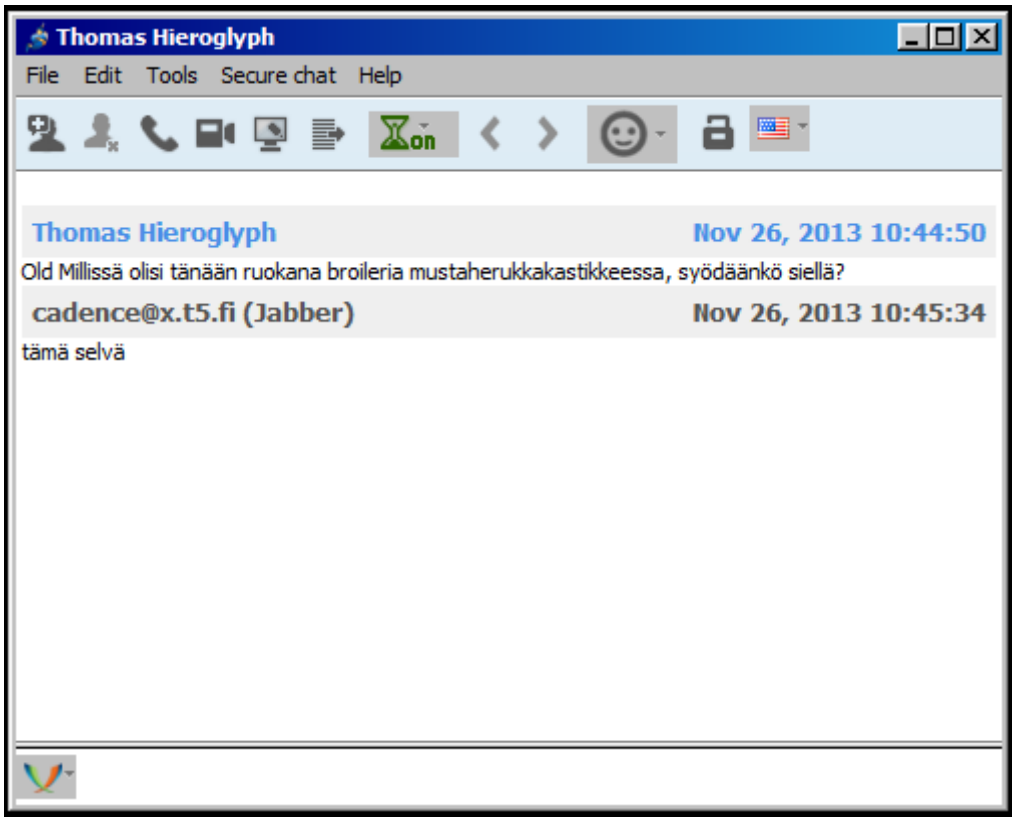
Kuva 36. Kontaktikutsun hyväksynnän vastausviesti.

Tämän jälkeen molemmat käyttäjät näkevät toisensa kontaktilistoissaan. Napsauttamalla kontaktin nimeä kahdesti saadaan auki IM-keskusteluikkuna, jonka alalaitaan kirjoittamalla ja enteriä painamalla voidaan lähettää toiselle käyttäjälle viesti (Kuva 37).

Ikkunan yläreunassa on rivi kuvakkeita muihin toimintoihin. Painamalla ensimmäistä kuvaketta vasemmalta voidaan keskusteluun lisätä toinen henkilö, kun taas tämän vieressä olevalla painikkeella keskustelusta voidaan poistaa henkilö. Kolmannella kuvakkeella voidaan soittaa toiselle henkilölle VoIP-puhelu (Kuva 37).

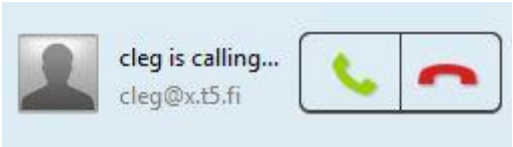
Neljännellä kuvakkeella voidaan soittaa videopuhelu ja viidennellä voidaan jakaa työpöytä. Kuudennella kuvakkeella voidaan lähettää tiedosto ja seitsemännellä lukea viestihistoriaa (Kuva 37).

Ensimmäinen kuvake oikealta avaa kieliopin tarkistuksen. Toinen kuvake oikealta mahdollistaa yhteyden salauksen, jolloin molempien osapuolien tulee syöttää yhteisesti sovittu koodi ennen keskustelua (Kuva 37).



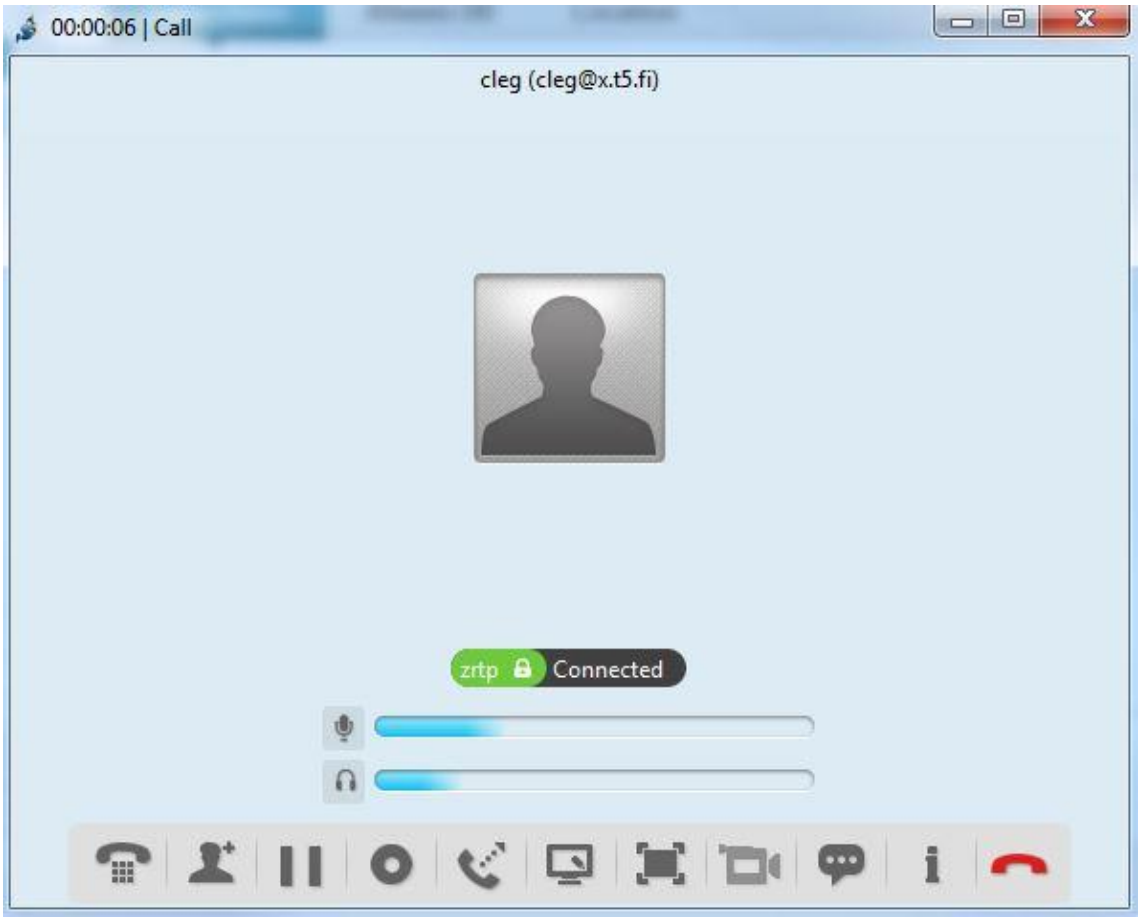
Kuva 37. Keskusteluikkuna.

VoIP-puhelua soitettaessa vastaanottajalle tulee ilmoitus saapuvasta puhelusta. Tämä hyväksytään painamalla vihreää painiketta, tai hylätään painamalla punaista painiketta (Kuva 38).



Kuva 38. Saapuva puhelu.

Puhelun hyväksymisen jälkeen aukeaa puheluikkuna. Kolmannella painikkeella oikealta puhelu voidaan asettaa pitoon, ja punaisella painikkeella puhelu lopetetaan (Kuva 39).



Kuva 39. Puheluikkuna.

Jitsistä ulos kirjautuminen tapahtuu painamalla Jitsin pääikkunan oikeassa yläreunassa olevasta File-valikosta Quit.

Kamailion asennus CentOS-käyttöjärjestelmään ja käyttöönotto

Siirrytään kansioon, johon rpm-tiedostot ladataan (tässä tapauksessa /var/local/rpm), jonka jälkeen sekä Kamailion että Kamailion MySQL-moduulin paketit ladataan ja asennetaan. Huomioidaan, että paketin sijaintiosoite tulee kirjoittaa kokonaan yhteen:

```
cd /var/local/rpm

wget
http://download.opensuse.org/repositories/home:/kamailio:/telephony/CentOS_C
entOS-6/x86_64/kamailio-4.0.4-2.1.x86_64.rpm

wget
http://download.opensuse.org/repositories/home:/kamailio:/telephony/CentOS_C
entOS-6/x86_64/kamailio-mysql-4.0.4-2.1.x86_64.rpm

rpm -ivh kamailio-4.0.4-2.1.x86_64.rpm

rpm -ivh kamailio-mysql-4.0.4-2.1.x86_64.rpm

cd
```

Kamailio on nyt asennettu. Seuraavaksi määritellään Kamailion asetukset.

Luodaan Kamailion paikallinen asetustiedosto, jossa määritellään Kamailion käyttämät asetukset. Tiedosto on helpointa luoda jollain tekstieditorilla, esimerkiksi nanolla:

```
nano etc/kamailio/kamailio-local.cfg
```

Tiedostoon määritetään rivi kerrallaan halutut parametrit. Pakollisina parametreina määritetään Kamailio käyttämään MySQL-tietokantaa, vaatimaan autentikoinnin ja ottamaan domain käyttöön:

```
#!define WITH_MYSQL

#!define WITH_AUTH
```

```
#!define WITH_MULTIDOMAIN
```

Oletuksena Kamailio hakee domainin automaattisesti. Manuaalinen määrittäminen tehdään lisäämällä paikalliseen asetustiedostoon myös seuraavat rivit, joista ensimmäinen estää domainin automaattisen hakemisen, ja jälkimmäinen määrittää domainin manuaalisesti. Domainin (esimerkkinä x.t5.fi) perään laitetaan myös käytettävä portti (Kamailio käyttää oletuksena porttia 5060):

```
auto_aliases=no

alias=x.t5.fi:5060
```

Mikäli palvelimen käyttäjät ovat NAT-reitittimen takana, täytyy Kamailio määrittää tarkistamaan mahdollinen NAT-liikenne. Tämä tehdään lisäämällä tiedostoon seuraavat rivit:

```
#!define WITH_USRLOCDB

#!define WITH_NAT
```

Tämän jälkeen tiedostosto tallennetaan. Mikäli NAT-liikenteen reititys määritettiin, täytyy palvelimelle asentaa RTPproxy, joka ohjaa liikenteen oikeaan osoitteeseen. Asentamisen jälkeen RTPproxy määritetään ohjaamaan NAT-reitittimen julkiseen IP-osoitteeseen tuleva liikenne. IP-osoite laitetaan komennossa x.x.x.x:n tilalle:

```
yum install rtpproxy

rtpproxy –F –l x.x.x.x –s udp:localhost:7722
```

Tämän jälkeen muokataan Kamailion sisäänrakennetun tietokannanluomisen asetuksia:

```
nano /etc/kamailio/kamctlrc
```

Etsitään tiedostosta seuraavanlainen rivi

```
# DBENGINE=MYSQL
```

Poistetaan rivistä kommentti, jolloin tietokanta luodaan MySQL:lle:

```
DBENGINE=MYSQL
```

Tämän jälkeen tiedosto tallennetaan, jonka jälkeen luodaan Kamailiolle tietokanta käyttäen Kamailion sisäänrakennettua tietokannan luontia. Tietokannan luonnin yhteydessä Kamailio kysyy, halutaanko tietokantaan asentaa lisätaulukoita. Näihin kaikkiin vastataan y. Oletuksena tietokanta luodaan nimellä kamailio, jonka read-write -käyttäjätunnuksen nimi on kamailio ja salasana kamailiorw:

```
kamdbctl create
```

Tämän jälkeen Kamailio käynnistetään, jonka jälkeen se määritetään käynnistymään aina koneen käynnistyksen yhteydessä:

```
service kamailio start

chkconfig kamailio on
```

Asetusten määrittäminen on nyt valmis.

Asennetaan Siremiksen vaatimat sovellukset

```
yum install make php php-mysql httpd
```

Siirrytään palvelimen www-kansioon, jonne ladataan Siremiksen paketti. Tämän jälkeen paketti puretaan ja poistetaan, jonka jälkeen annetaan apachelle muokkausoikeudet Siremiksen kansioon:

```
cd /var/www/

wget http://siremis.asipto.com/pub/downloads/siremis/siremis-4.0.0.tgz

tar xvfz siremis-4.0.0.tgz

rm siremis-4.0.0.tgz

chown -R apache siremis-4.0.0/
```

Siirrytään Siremiksen kansioon, jonka jälkeen valmistellaan Siremis ensimmäistä käynnistyskertaa varten. Tämän jälkeen luodaan apachen asetustiedosto:

```
cd siremis-4.0.0

make prepare

make apache-conf > /etc/httpd/conf.d/siremis.conf
```

Tämän jälkeen käynnistetään apache ja asetetaan se käynnistymään aina koneen uudelleenkäynnistyksen yhteydessä:

```
service httpd start

chkconfig httpd on
```

Seuraavaksi luodaan Siremikselle valmiiksi MySQL-käyttäjätili salasanalla siremis, jolle annetaan oikeus siremis nimiseen tietokantaan:

```
mysql -u root -e "GRANT ALL PRIVILEGES ON siremis.* TO siremis@localhost IDENTIFIED BY 'siremis';"
```

Asennus on nyt valmis.

Aukaistaan selaimella web-käyttöliittymä osoitteesta http://x.x.x.x/siremis, jossa x.x.x.x on palvelimen osoite tai domain.

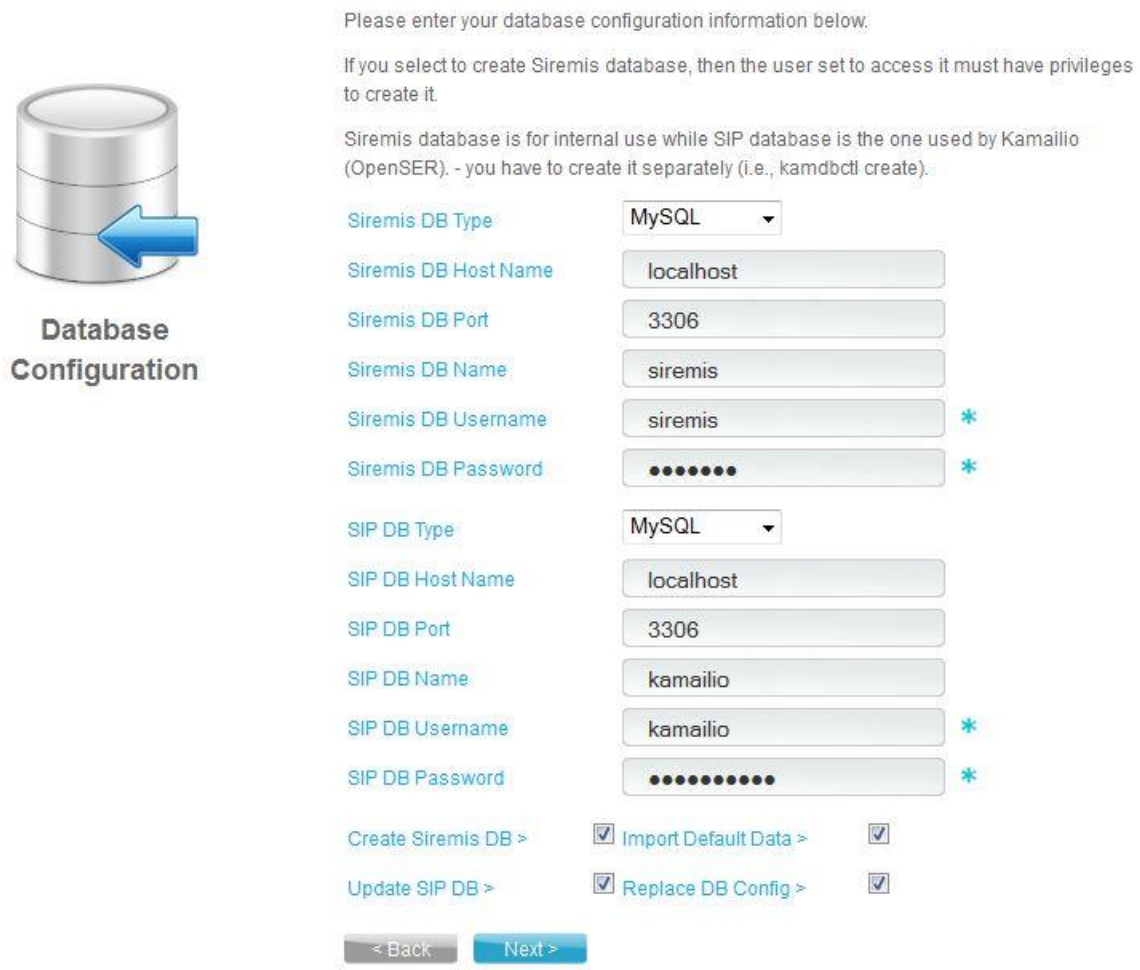
Aloitetaan esiasetusten määrittäminen Setup Now! -painikkeella (Kuva 40).

Kuva 40. Siremiksen aloitusruutu.

Tämän jälkeen Siremis tarkistaa tarvittavien ohjelmien ja lisäosien tilan (Kuva 41).

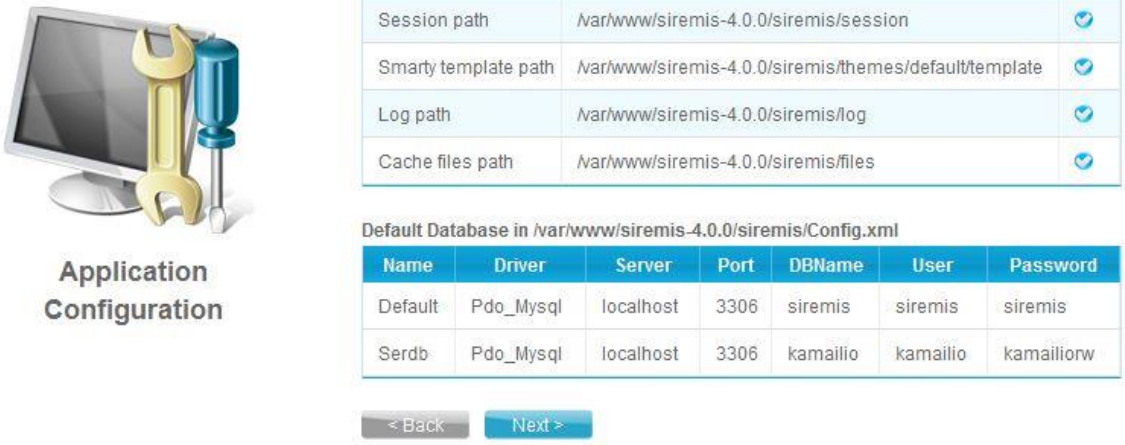
Kuva 41. Siremiksen tarvitsemien ohjelmien ja lisäosien tarkistus.

Seuraavaksi luodaan Siremiksen tietokanta ja määritellään Kamailion (tässä ikkunassa ”SIP”) tietokannan asetukset. Asetukset voidaan useimmissa tapauksissa jättää oletuksiksi, paitsi salasanat, jotka määritellään manuaalisesti. Siremiksen tietokannan salasana määritellään kentässä Siremis DB Password (tässä tapauksessa siremis), ja Kamailion tietokannan salasana määritellään kentässä SIP DB Password (tässä tapauksessa kamailiorw). Tämän lisäksi valitaan kaikki lopussa olevat kentät, jotta Siremis luo oman tietokantansa, tuo sinne oletustiedot sekä korvaa ja päivittää Kamailion tietokannan (Kuva 42).



Kuva 42. Siremiksen ja Kamailion tietokanta-asetukset.

Tämän jälkeen Siremis näyttää määriteltyjen asetuksien yhteenvedon (Kuva 43).



Kuva 43. Siremiksen asetusten yhteenveto.

Esiasetusten määrittäminen on nyt tehty. Lopuksi Siremis kehottaa poistamaan asennuskansion sekä vaihtamaan järjestelmänvalvojan salasanan. Poistetaan Siremiksen asennuskansio CentOS:stä:

```
cd /var/www/siremis-4.0.0/siremis/

rm –rf install
```

Tämän jälkeen lopetetaan esiasennus ja siirrytään sisäänkirjautumiseen painamalla Launch Siremis -painiketta (Kuva 44).

Kuva 44. Esiasetusten määrittäminen on valmis.

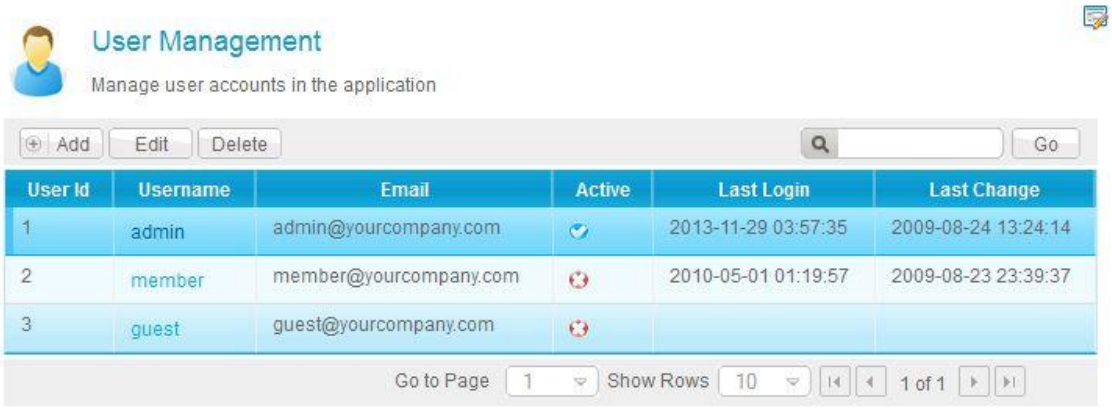
Kirjaudutaan Siremikseen käyttäjänimellä admin, jonka salasana on admin (Kuva 45).

Kuva 45. Siremiksen sisäänkirjautumisikkuna.

Ensimmäisen kirjautumisen yhteydessä vaihdetaan järjestelmänvalvojan salasana. Tämä tehdään valitsemalla System Administration -alueelta User management (Kuva 46).

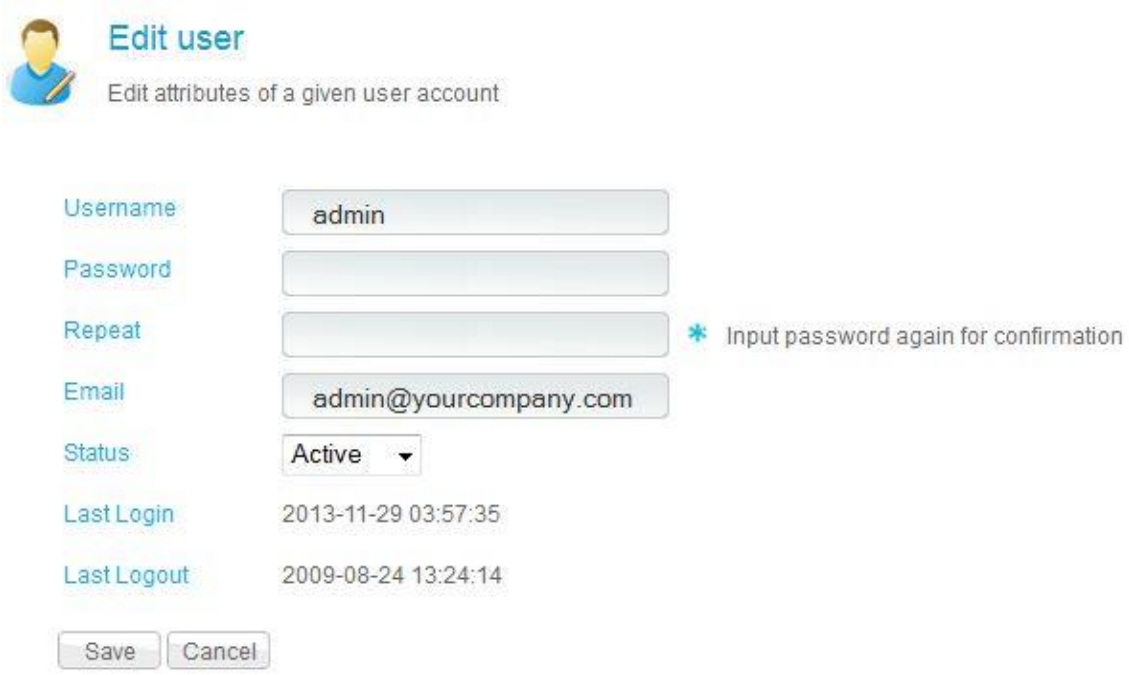
Kuva 46. Ote Siremiksen pääikkunasta.

Tämän jälkeen painetaan admin-käyttäjän vieressä olevaa User Id -kenttää, jonka jälkeen painetaan yläpuolelta Edit-painiketta (Kuva 47).



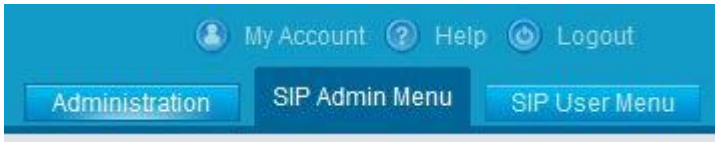
Kuva 47. Siremiksen käyttäjätilien hallintapaneeli.

Seuraavaksi Password-kenttään syötetään haluttu salasana, joka kirjoitetaan uudestaan Repeat-kenttään. Email-kenttään kirjoitetaan järjestelmänvalvojan sähköpostiosoite. Tämän jälkeen painetaan Save-painiketta (Kuva 48).



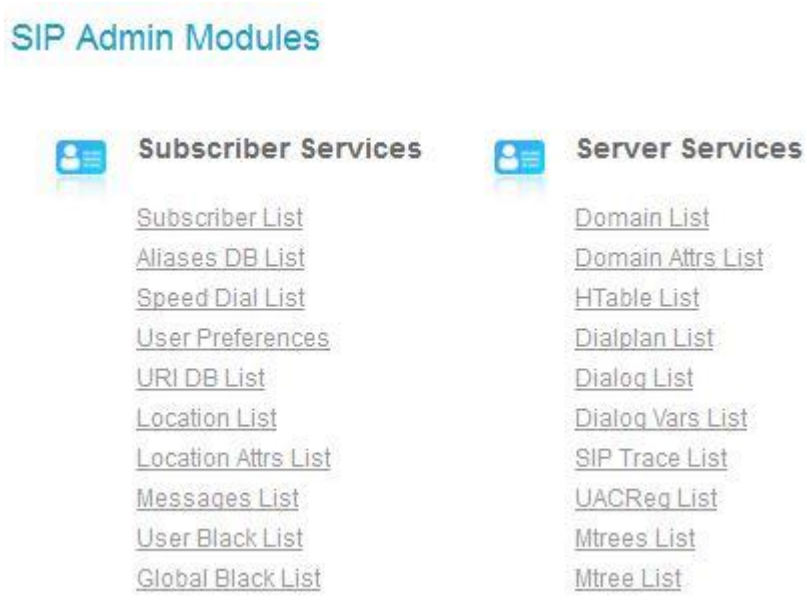
Kuva 48.Siremiksen käyttäjätilin muokkaus.

Tämän jälkeen lisätään Kamailion domain sekä käyttäjätilit. Tämä tehdään painamalla käyttöliittymän oikeassa yläkulmassa olevaa SIP Admin Menu -painiketta (Kuva 49).



Kuva 49. Käyttöliittymän vaihtaminen.

Ensimmäiseksi lisätään Kamailion tietokantaan domain painamalla SIP Admin Modules -hallintapaneelissa Domain List (Kuva 50).



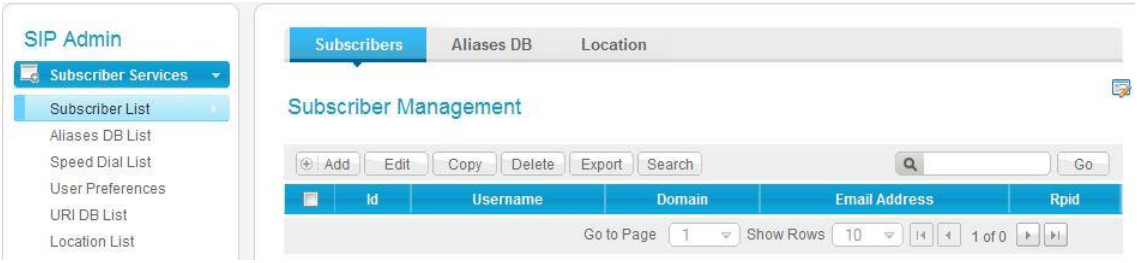
Kuva 50. Ote Kamailion tietokannan hallintapaneelista.

Tämän jälkeen aukeavasta hallintapaneelista painetaan Add-painiketta (Kuva 51), jonka jälkeen syötetään palvelimen domain Domain-kenttään, ja lopuksi painetaan Save-painiketta (Kuva 52).

Kuva 51. Kamailion tietokannan domain-taulukko.

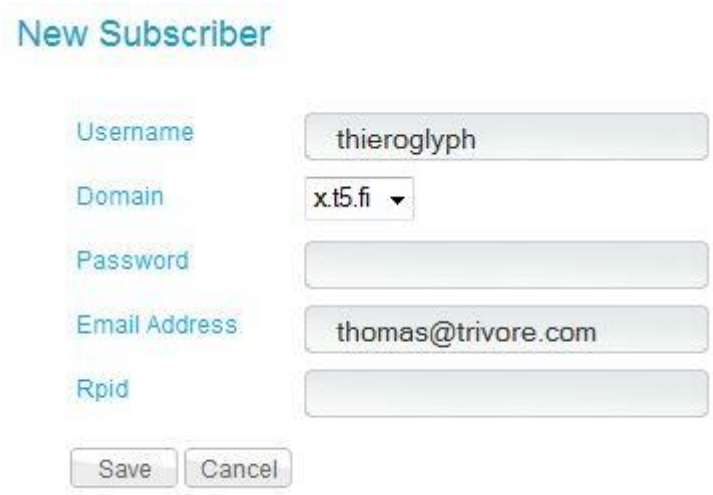
Kuva 52. Domainin lisääminen Kamailion tietokantaan.

Seuraavaksi luodaan käyttäjätilit Kamailion tietokantaan. Vasemmalla olevasta sivupalkista aukaistaan Subscriber Services -valikko, josta valitaan Subscriber List. Tämän jälkeen aukeavasta hallintapaneelista painetaan Add-painiketta (Kuva 53).



Kuva 53. Kamailion tietokannan käyttäjätili-taulukko.

Username-kenttään kirjoitetaan käyttäjätilin nimi ja domain-kenttään valitaan palvelimen käyttämä domain. Password-kenttään kirjoitetaan käyttäjätilin salasana. Huomioidaan, että salasana näkyy kentässä sensuroimattomana. Email Address -kenttään kirjoitetaan käyttäjän sähköpostiosoite. Lopuksi painetaan Save-painiketta (Kuva 54).



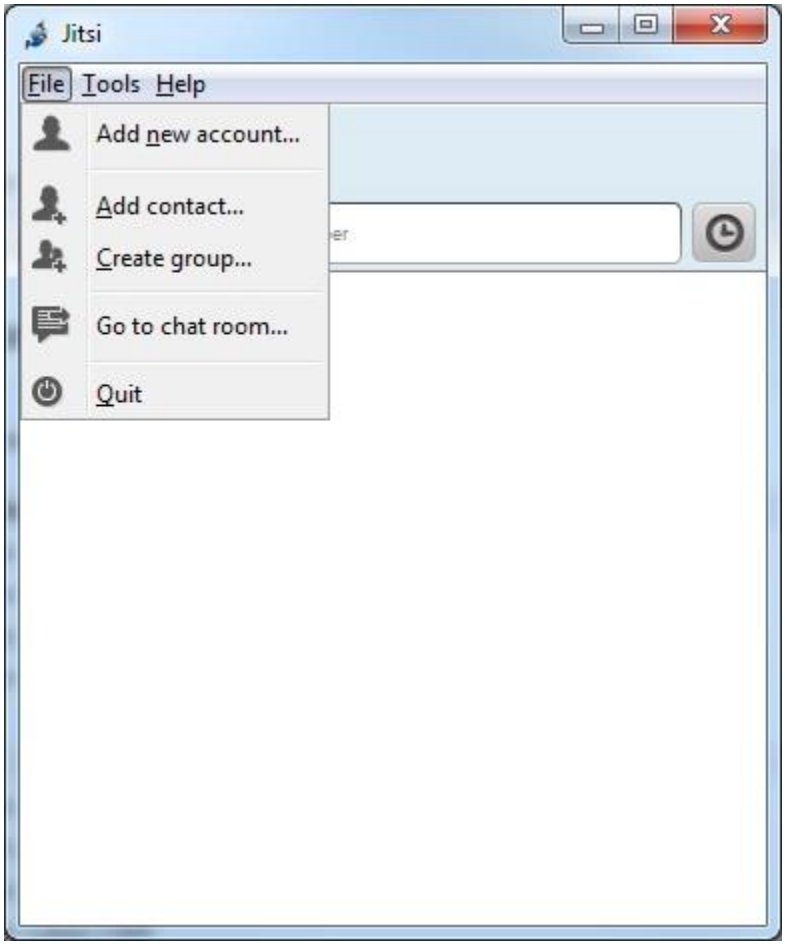
Kuva 54. Käyttäjätilin lisääminen.

Lisää käyttäjiä voidaan lisätä painamalla jälleen Add-painiketta.

Kun tarvittavat käyttäjät on luotu, Kamailio on käyttövalmis. Siremiksestä kirjaudutaan ulos painamalla käyttöliittymän oikeassa yläkulmassa olevaa Logout-painiketta.

Käyttöönotto Jitsi-asiakasohjelmalla

Käynnistetään Jitsi-asiakasohjelma, jonka jälkeen lisätään käyttäjätili valitsemalla vasemmassa yläkulmassa olevasta File-valikosta Add new account (Kuva 55.)

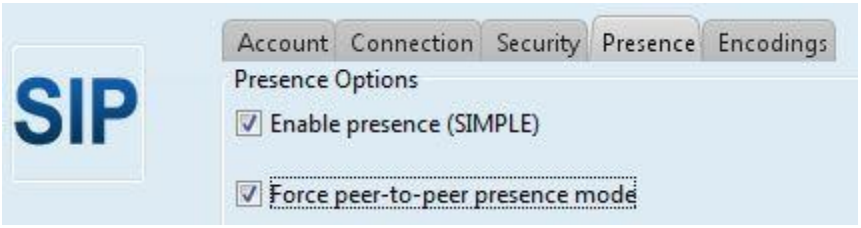


Kuva 55. Uuden käyttäjätilin lisääminen Jitsiin.

Seuraavaksi valitaan Network-valikosta SIP. SIP id -kenttään kirjoitetaan SIP-tilin nimi muodossa [käyttäjänimi]@[domain], jossa [käyttäjänimi] on tietokannasta löytyvä käyttäjänimi ja [domain] Kamailion käytössä oleva domain. Password-kenttään syötetään tilin salasana. Halutessaan Jitsin voi määrittää muistamaan tilin salasanan valitsemalla Remember password -kentän, jolloin sisäänkirjautuminen tapahtuu automaattisesti. Tämän jälkeen painetaan Advanced-painiketta lisäasetusten säätämiseksi. (Kuva 56.)

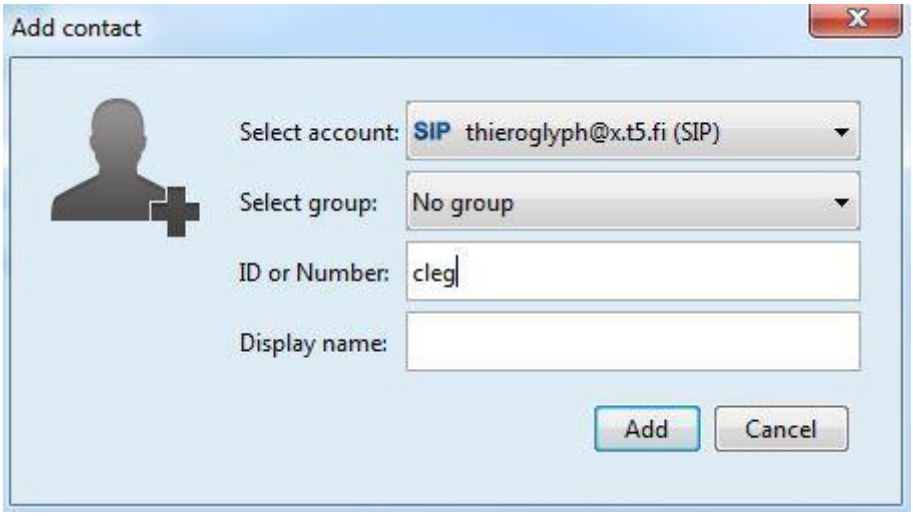
Kuva 56. Käyttäjätilin tietojen määritäminen.

Tämän jälkeen aukaistaan Presence-välilehti, josta valitaan Force peer-to-peer presence mode -kenttä (Kuva 57). Näin Jitsi lähettää käyttäjien tilatiedot käyttäjältä toiselle. Tämän jälkeen kirjaudutaan sisään painamalla Next ja sen jälkeen Sign in -painikkeita.



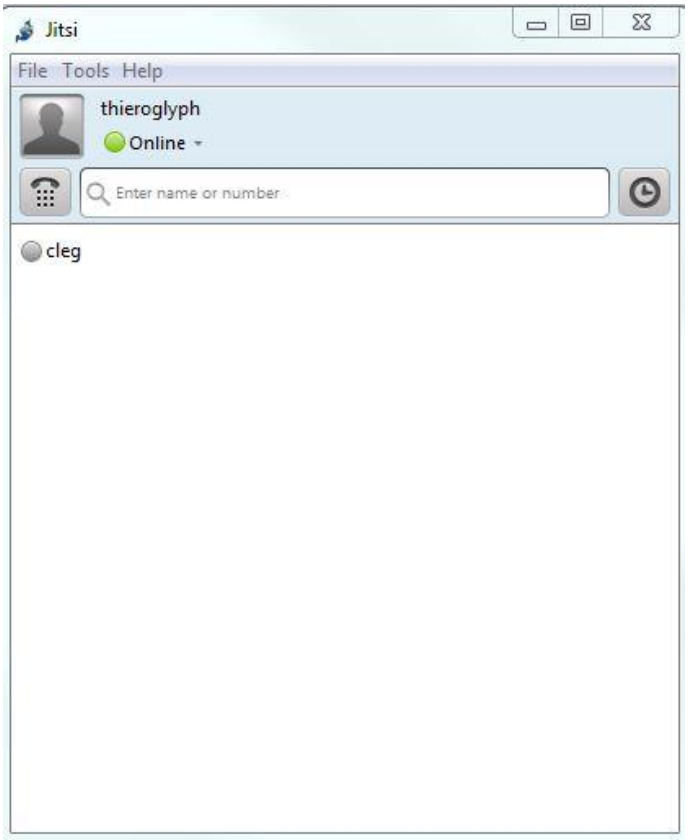
Kuva 57. Tilatietojen asetus.

Käyttäjätili on nyt käyttövalmis. Kontaktikutsujen lähettäminen tehdään valitsemalla vasemmassa yläkulmassa olevasta File-valikosta Add contact. Tällöin aukeavasta ikkunasta olevasta Select account -kentästä valitaan käyttäjätili, johon kontakti halutaan lisätä. Select group valikosta on mahdollista valita ryhmä, johon kontakti lisätään. ID or Number -kenttään määritetään halutun kontaktin käyttäjänimi. Display name -kenttään voi halutessaan määrittää kontaktille nimen, jolla kontakti näkyy kontaktiluettelossa (Kuva 58).



Kuva 58. Kontaktikutsun lähettäminen.

Kun kontaktikutsu on lähetetty, kontakti tulee näkyviin kontaktilistaan (Kuva 59). Kontaktitiedot eivät vaihdu automaattisesti keskenään, joten molempien käyttäjien on lähetettävä toisilleen kontaktikutsu.

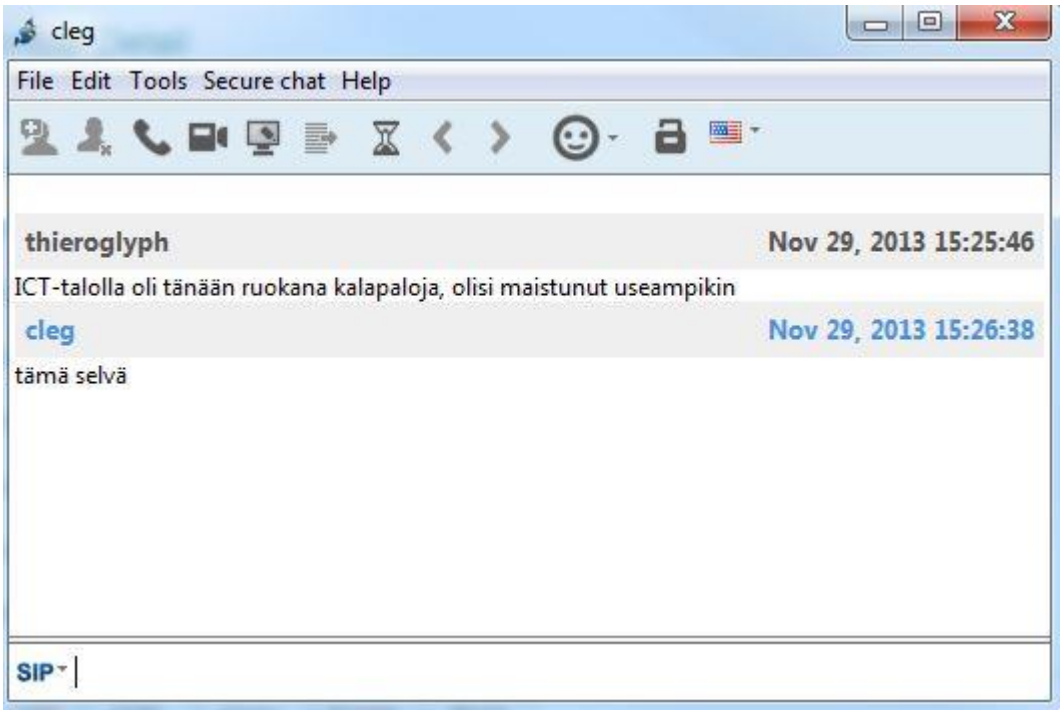


Kuva 59. Jitsin pääikkuna jossa yksi kontakti.

Napsauttamalla kontaktin nimeä kahdesti saadaan auki IM-keskusteluikkuna, jonka alalaitaan kirjoittamalla ja enteriä painamalla voidaan lähettää toiselle käyttäjälle viesti (Kuva 60).

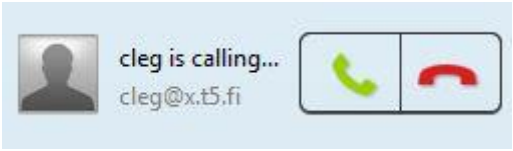
Ikkunan yläreunassa on rivi kuvakkeita muihin toimintoihin. Kolmannella kuvakkeella vasemmalta voidaan soittaa toiselle henkilölle VoIP-puhelu. Neljännellä kuvakkeella voidaan soittaa videopuhelu ja viidennellä voidaan jakaa työpöytä (Kuva 60).

Ensimmäinen kuvake oikealta avaa kieliopin tarkistuksen. Toinen kuvake oikealta mahdollistaa yhteyden salauksen, jolloin molempien osapuolien tulee syöttää yhteisesti sovittu koodi ennen keskustelua (Kuva 60).



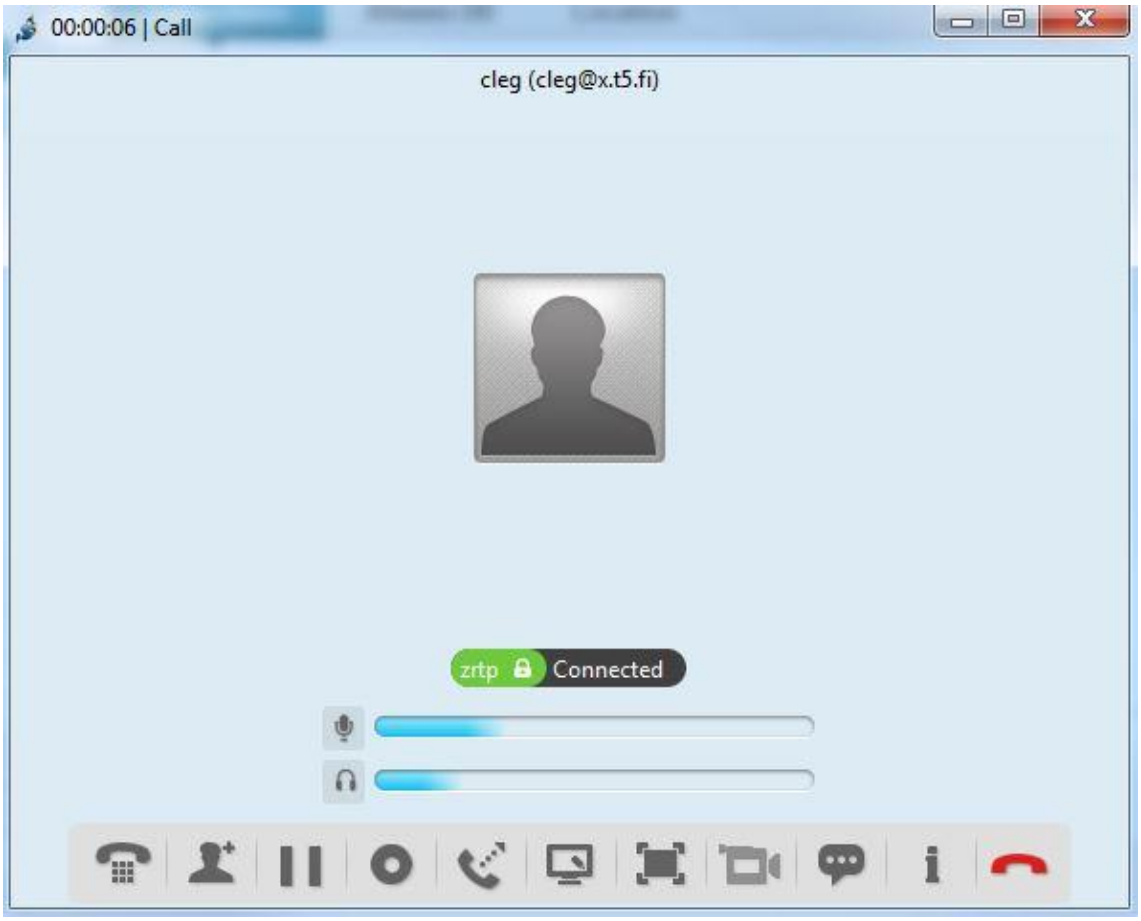
Kuva 60. Keskusteluikkuna.

VoIP-puhelua soitettaessa vastaanottajalle tulee ilmoitus saapuvasta puhelusta. Tämä hyväksytään painamalla vihreää painiketta, tai hylätään painamalla punaista painiketta (Kuva 61).



Kuva 61. Saapuva puhelu.

Puhelun hyväksymisen jälkeen aukeaa puheluikkuna. Kolmannella painikkeella oikealta puhelu voidaan asettaa pitoon, ja punaisella painikkeella puhelu lopetetaan (Kuva 62).



Kuva 62. Puheluikkuna.

Jitsistä ulos kirjautuminen tapahtuu painamalla Jitsin pääikkunan oikeassa yläreunassa olevasta File-valikosta Quit.