hyväksymispäivä arvosana

arvostelija

# Keskitetty tunnistautuminen web-sovelluksissa

Olli Jokinen

Helsinki 30.11.2011

Pro gradu -tutkielma

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

| Tiedekunta — Fakultet — Faculty | Laitos — Institution — Department |
|---|---|
| Matemaattis-luonnontieteellinen | Tietojenkäsittelytieteen laitos |

Tekijä — Författare — Author
Olli Jokinen

Työn nimi — Arbetets titel — Title

Keskitetty tunnistautuminen web-sovelluksissa

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

| Työn laji — Arbetets art — Level | Aika — Datum — Month and year | Sivumäärä — Sidoantal — Number of pages |
|---|---|---|
| Pro gradu -tutkielma | 30.11.2011 | 19 sivua + 119 liitesivua |

Tiivistelmä — Referat — Abstract

lol

ACM Computing Classification System (CCS): A.1 [Introductory and Survey]

I.7.m [Document and Text Processing]: Miscellaneous

Avainsanat — Nyckelord — Keywords
avainsana

Säilytyspaikka — Förvaringsställe — Where deposited
Tietojenkäsittelytieteen laitoksen kirjasto

Muita tietoja — övriga uppgifter — Additional information

# Sisältö

# 1 Johdanto

Ohjelmistoyritys Enemy & Sons toteuttaa ja ylläpitää web-sovelluksia. Tyypillisessä projektissa sovellus saadaan valmiina kokonaisuutena ja se on tarkoitus muuttaa skaalautuvaksi palveluperustaiseksi arkkitehtuuriksi. Järjestelmää pilkottaessa erillisiin komponentteihin, törmätään usein käyttäjän tunnistamisen ongelmaan. Yritys haluaa ratkaisun, jossa yksittäisten komponenttien tunnistautumisongelma on selvitetty. Ratkaisun täytyy olla monistettavissa tyypillisiin web-sovellusprojekteihin.

Enemy & Sons käyttää ohjelmistoissaan moderneja web-ohjelmointikieliä ja toteutettu ratkaisu ei saa olla kieli- tai ympäristöriippuvainen. Järjestelmä rakennetaan avoimien rajapintojen ja protokollien päälle. Järjestelmän tulee olla laajennettavissa myös käyttäjän pääsynhallintaan, jolloin pelkkä tunnistautuminen ei riitä, vaan käyttäjällä täytyy olla oikeus käyttää palvelua. Pääsynhallintaa ei toteuteta tutkimuksessa syntyvään prototyyppiin, mutta käytettyjen ratkaisujen täytyy tukea myös sitä.

Tutkielma jakautuu kahteen osaan. Luvuissa 2, 3 ja 4 käsitellään ongelmakenttää yleisellä tasolla. Luvussa 2 esitellään web-sovellukset, erityisesti palveluperustaiset web-sovellukset. Luvussa 3 käydään läpi käyttäjän tunnistautumisen tekniikoita ja ongelmakenttää. Luvussa 4 käsitellään keskitettyä tunnistautumista palveluperustaisissa arkkitehtuureissa. Tutkielman toisessa osassa esitellään ratkaisu yrityksen kohtaamaan ongelmaan. Luku 5 esittelee toteutuksen, joka evaluoidaan luvussa 6. Luvussa 7 pohditaan toteutuksen laajennettavuutta. Luku 8 on yhteenvetokappale.

# 2 Web-sovellukset

Mitä on web-palvelut? Arkkitehtuureja? n-tier? Komponentit?

Luvussa esitellään web-sovelluksien yleisiä periaatteita. Käydään läpi sovellusten

historia perinteisistä asiakas-palvelin malleista kohti nykyaikaisia palveluperustaisia web-sovelluksia. Palveluperustaisissa arkkitehtuureissa on olennaista, että palikat ovat autonomisia, mutta niissä tarvitaan käyttäjähallintaa. Luvun tarkoitus on joka tapauksessa kertoa sovelluksista yleisellä tasolla, varsinaiseen tunnistautumisen ongelmakenttään paneudutaan luvussa 4.

Pituus n. 5 sivua, lukujen 2-4 yhteispituus n. 20sivua.

Lähteitä:
Modeling Web application architectures with UML [Con99]
Web application characterization through directed requests [ECFR06] (yleistä pohdintaa mikä on web-applikaatio)
Solution Architecture for N-Tier Applications [SH06] (ehkä?)

---

Web-sovelluksella tarkoitetaan ohjelmaa, jota suoritetaan palvelimella ja jota käytetään WWW-selaimen avulla [Con99]. Käyttäjän selaimen kautta tekemät pyynnöt vaikuttavat palvelimen tilaan. Esimerkiksi lomakkeella voidaan lähettää palvelimelle tallennettavaa tietoa tai muokata olemassaolevaa. Web-sovellukset toteuttavat business logic (TODO: käännös), joka erottaa ne tavallisista web-sivuista [Con99]. Web-sivu voi toimia dynaamisesti, mutta ilman mahdollisuutta vaikuttaa business logiikan (WTF) tilaan, ei voida puhuta web-sovelluksesta.

Web-sovelluksissa on kolme perus komponenttia: selain, verkko ja palvelin. Käyttäjä pyytää selaimella verkon yli palvelimelta selaimen ymmärtämällä kielellä ohjelmoitua tiedostoa, jonka selain visualisoi käyttäjän ymmärtämään muotoon [Con99]. Käytetyin kieli on HTML, mutta yleisesti käytettyjä kieliä ovat myös JavaScript ja Flash. Pyyntö tehdään HTTP- tai HTTPS-protokollilla, riippuen käytetäänkö salaamatonta (HTTP) vai salattua (HTTPS) yhteyttä.

TODO: joko kuva tähän, joka kertoo mistä on kyse

TODO: luvun esittely, mitä seuraavissa aliluvuissa?

## 2.1 Historia

tässä kai voisi kertoa kehityksen perus server-client moskasta kohti javascriptillä toteutettavia DOM-virityksiä

Erityisesti tärkeää kertoa, että käyttöliittymä ja data erotetaan toisistaan: käyttöliittymä rakennetaan javascriptillä (tai flashilla, silverlightilla whatever) ja erilliset palvelut tarjoavat sitten json/xml-dataa, jota tuo käyttöliittymä visualisoi. Gradun ongelma onkin juuri tässä: miten toteuttaa käyttöliittymä, jossa javascript hakee dataa erilaisista datalähteistä ja käyttäjä pystytään tunnistamaan näissä eri datalähteissä. Tuskin tämä kuuluu web-palveluiden historia-kappaleeseen, mutta tämän kappaleen tarkoitus on kertoa lukijalle, että tällaisia ne web-palvelut nykyään on. Eli n-tier on tässä nyt jotenkin läsnä.

## 2.2 Palvelinarkkitehtuurit

Palvelinten yleinen arkkitehtuuri. Tietokantasovellus, sovellus yms. Ehkä vähän laajemmin kuin pelkästään palvelinten arkkitehtuurit, eli käyttäjä selaimineen mukaan. Miten kontrolli menee järjestelmässä, kun käyttäjä pyytää sivua x.

## 2.3 Palveluperustaiset web-sovellukset

Perus arkkitehtuurissa oleva sovellus pilkotaan autonomisiin kokonaisuuksiin, jotka juttelee keskenään web service -rajapintojen kautta.

### 2.4 Yhteenveto

Vedetään yhteen mitä edellä kuvattiin. Käyttöliittymät juttelee ziljoonan erilaisen web-palvelun kanssa.

## 3 Tunnistautuminen

Miten tunnistetaan perinteisesti?

Käyttäjädatan tallennus - passwd (ts tiedosto levyllä tms) - tietokanta - AD-kannat (LDAP)

—> käyttäjän datan abstraktoinnin tarve, ehkä mainita, että käyttäjädatan backendejä voi käytännössä olla monta erilaista. Esim. LDAP:n rinnalla tietokannat.

Pituus n. 5 sivua.

Lähteitä:
A Guide to Computer Network Security [Kiz09]
Authentication in distributed systems: theory and practice [LABW92]

LDAP:

Howes, T. A., The Lightweight Directory Access Protocol: X.500 Lite. CITI Technical Report 95–8, University of Michigan, 1995. [How95]
rfc:t 4510-4513 (ainakin 4513 "Authentication Methods and Security Mechanisms"kiinnostaa)

---

Johdatusta tunnistautumisen maailmaan. Käydään läpi perus käsitteitä tunnistautumisesta ja siitä miten se nykyään hoidetaan. Paikallinen tunnistautuminen, tunnistautuminen verkossa jne.

## 3.1 Ympäristön kuvaus

Tyypillisesti yrityksissä on LDAP tms tietokanta, johon itsenäiset palvelut autentikoivat. Master-kanta ja siitä useita kopioita.

## 3.2 Järjestelmien tietoturva

Palveluiden turvallisuus koostuu kolmesta tekijästä: tunnistautumisesta (authentication), pääsynvalvonnasta (access control) ja auditoinnista (audit) [SS96]. Tunnistautumisessa käyttäjän identiteetti varmistetaan, esimerkiksi käyttäjätunnuksen ja salasanan avulla. Tämän jälkeen pääsynvalvonta tarkistaa onko kyseisellä käyttäjällä oikeutta tehdä pyytämäänsä toimintoa. Auditoinnissa analysoidaan järjestelmän tuottamaa dataa, esimerkiksi lokitiedostoja, aktiivisesti ja käyttäjän pääsy järjestelmään voidaan estää, jos luvatonta käyttöä esiintyy [SS96].

Tämän tutkimuksen painopiste on käyttäjän tunnistautumisessa ja osittain myös pääsynvalvonnassa, auditointi ei kuulu tutkimuksen aihepiiriin. Tutkimuksessa käytetään termiä pääsynvalvonta sen suppeassa merkityksessä, toisinaan kirjallisuudessa pääsynvalvontaa pidetään kattoterminä, joka pitää sisällään tunnistautumisen, valtuutuksen (authorization) ja auditoinnin [TODO: joku lähde, jossa näin on].

## 3.3 Käyttäjädata

Miten käyttäjädataa onko käsitelty ja käsitellään. Kehitys paikallisesti käytetyistä tiedostopohjaisista systeemeistä kohti tietokantoja ja asiaan räätälöihin palveluihin (LDAP). LDAP oleelisin, mutta tutkimuksen kannalta abstraktointi on tärkeä juttu.

Johdanto puoli sivua, alaluvut 0.5-1 sivu.

### 3.3.1 passwd

Vanha kunnon /etc/passwd, tästä tunnistautuminen on varmaan lähtenyt käyntiin. Ikävää, kun webiin tunnistautuessa täytyy olla tunnus kyseisellä koneella ja muutenkin ei ole hyvä kun tunnukset siirtyy verkkoa pitkin.

### 3.3.2 Relaatiotietokannat

Käyttäjätietokannat, relaatiokannat lähinnä, ehkä NoSQL.

### 3.3.3 LDAP

Lightweight Directory Access Protocol (LDAP) on X.500 OSI-standardiin perustuva hakemistopalvelu, jota käytetään yleisesti käyttäjätiedon tallennukseen [TODO: lähde]. 1990-luvulla TCP/IP-mallin syrjäytettyä OSI-mallin, myös DAP kävi vanhanaikaiseksi [How95]. Korvaajaksi on noussut LDAP, josta käytetään myös nimeä X.500 Lite [How95].

LDAP:ssa asiakassovellukset (directory user agent, DUA) keskustelevat puumalliin perustuvan hakemistopalvelimen (directory system agent, DSA) kanssa käyttäen määriteltyä protokollaa (directory access protocol, DAP) [How95]. Asiakassovellukset voivat hakea hakemistopalvelimesta tietoa suodattimiin (filter) perustuvalla lukuoperaatiolla. Suodattimessa voidaan määritellä raja-arvot attribuutin arvolle tai hakea avainsanoilla attribuuteista.

LDAP-tietuille voidaan määritellä pakollisten attribuuttien (esim. etu- ja sukunimi) lisäksi valinnaisia attribuutteja. Tietueet on järjestetty puuhun niiden yksilöivän nimen (distinguished name, DN) mukaan ja ne voi olla hajautettu usealle palvelimelle. Suhteellinen nimi (relative distinguished name, RDN) identifioi tietueen omalla hierarkiatasollaan.

LDAP-tietueella voi olla tunnus sekä salasana ja LDAP-palvelinta voidaan käyttää käyttäjän tunnistautumiseen [TODO: lähde, ehkä rfc4513].

TODO: lisää tekstiä

### 3.3.4 Käyttäjädatan abstraktointi

Tutkimuksen kannalta abstraktointi on oleellista, oikeastaan sillä ei ole ison kuvan kannalta merkitystä, että onko siellä taustalla tietokanta, tiedosto, ldap vai mikä.

## 3.4 Yhteenveto

Vedetään yhteen tunnistautuminen. Ymmärretään miksi on ongelmallista, kun SOA-arkkitehtuurilla tehdyt ohjelmistot tunnistavat itsenäisesti master-tietokantaan.

# 4 Keskitetty tunnistautuminen

Esitellään palveluperustaisten web-sovellusten tunnistautumisen ongelmakenttää ja esitellään niihin suunniteltuja protokollia, aikajärjestyksessä kerberos -> saml -> oauth ja päädytään oauthiin, tarkemmin versioon 2.

Kappaleessa vedetään yhteen kappaleissa 2 ja 3 esiteltyjä perusjuttuja ja avataan nykyaikaisiin hajautettuihin web-sovelluksiin liittyviä tunnistautumisongelmia. Ratkaisuksi tarjotaan erilaisia protokollia ja tehdään niistä vertailua.

Pituus n. 10 sivua. Tämä on alkupään kappaleista oleellisin, koska tässä oikeasti pureudutaan ongelmaan, joka yrityksellä on.

Lähteitä:
A billion keys, but few locks: the crisis of web single sign-on [SBHB10]
A large-scale study of web password habits [FH07] Inside the identity management

game [Lyn11]

Decentralization: The Future of Online Social Networking [mAYLL+09]

Lampson & kumppanit: Authentication in distributed systems: theory and practice [LABW92].

Kerberos:

Enhancing Distributed Web Security Based on Kerberos Authentication Service [LC10]

Secure Secret-Key Management of Kerberos Service [Cao11]

RFC4120 [NYHR05]

RFC3244 [STB02]

SAML:

Research of Dynamic Authentication Mechanism Crossing Domains for Web Services Based on SAML [YsJ10]

Next steps for security assertion markup language (saml) [SS07]

OAuth:

2.0 draft: http://tools.ietf.org/html/draft-ietf-oauth-v2-22 [RHHL11]

---

Web-palvelujen määrä on kasvanut reilusti viime vuosina ja tyypillisesti kaikissa järjestelmissä on oma käyttäjähallintansa [SBHB10]. Tästä seuraa käyttäjän tietojen monistaminen moneen eri järjestelmään ja tyypillisesti käyttäjä kirjautuu näihin järjestelmiin eri identiteetillä. Vuonna 2007 julkaistun tutkimuksen mukaan web-käyttäjällä on n. 25 erillistä verkkoidentiteettiä, joista hän käyttää päivittäin kahdeksaa [FH07].

Keskitetyn tunnistautumisen tarkoituksena on tarjota palvelu, jota vasten käyttäjä voidaan tunnistaa erillisestä web-palvelusta ilman uuden identiteetin luontia. Tunnistautumispalvelun ja sitä käyttävien web-palveluiden välillä on luottamussuhde,

jolloin erilliseen web-palveluun ei tarvitse luoda omaa käyttäjätietokantaa, vaan se voi luottaa tunnistautumispalvelun tunnistamiin käyttäjiin.

Tunnistautumispalvelulla voidaan parantaa järjestelmien tietoturvaa ja parantaa tunnistautumisen luotettavuutta. Käyttäjät valitsevat vahvempia salasanoja palveluihin, jotka he kokeavat tärkeiksi, kuten sähköpostiin, verrattuna vähemmän tärkeisiin web-palveluihin [FH07]. Keskitetyssä palvelussa käyttäjän tunnistautumisen luotettavuutta voidaan parantaa esimerkiksi vaatimalla normaalia web-palvelua vahvempia salasanoja. Käyttäjien todennusta voidaan vahvistaa myös lisävarmistuksilla, kuten erilaisilla tunnuslukulistoilla tai puhelimen kautta tehtävällä todennuksella.

Toisaalta keskitetyn tunnistautumiseen käytetyn salasanan kalastelu käy houkuttelevammaksi, koska sen avulla pääsee käyttäjän tietoihin käsiksi useaan palveluun tai jopa luomaan käyttäjän identiteetillä tunnuksia uusin palveluihin. Palveluun kohdistuu myös normaalia web-palvelua suuremmat odotukset tietoturvalle, joten käytettyjen protokollien täytyy olla luotettavia.

## 4.1 Ongelmakenttä

Palvelusuuntautuneissa web-ohjelmistoissa käyttäjien tunnistautuminen on toteutettu monin tavoin. Tyypillisesti kysytään käyttäjältä tunnus ja salasana, joita verrataan ohjelmiston paikalliseen käyttäjätietokantaan. Paikallinen tietokanta on kopioitu palvelun varsinaisesta käyttäjätietokannasta ja paikallisiin tietokantoihin on käyttäjille luotu erillinen käyttäjätunnus.

Tästä seuraa monenlaisia synkronointiongelmia. Esimerkiksi työntekijän irtisanoutuessa joudutaan tunnus poistamaan kaikista tietokannoista erikseen. Myös osoitteen yms. tietojen muutokset täytyy päivittää kaikkiin tietokantoihin. Lisäksi käyttäjälle syntyy saman järjestelmän sisällä monia tunnuksia, joihin saattaa liittyä erilliset

salasanat. Käyttäjän kannalta on myöskin ikävää kirjautua jokaiseen osapalveluun erikseen.

## 4.2 Ympäristö

Tyypillisesti web-palvelun toimintakenttä on Internet, jossa palvelut toimivat itsenäisesti. Näiden palveluiden välinen integraatio on kasvussa ja palveluiden kesken halutaan jakaa tietoa, jolloin niiden täytyy pystyä identifioimaan käyttäjä keskenään. Yleisen identeettitarjoajan rakentaminen Internettiin on tutkimuksen alla ja OpenID ja mitä näitä nyt on.

Usein ei ole tarpeen tehdä palveluista julkisia, vaan käyttöoikeus niihin voidaan rajata tietylle osajoukolle kaikista Internetin käyttäjistä. Tällaisia osajoukkoja voi olla esimerkiksi yrityksen työntekijät, joilla on pääsy intranet-palveluihin tai tietyn sivuston käyttäjät, joilla on pääsy sivuston palveluihin. Tällöin voi olla järkevää eriyttää käyttäjähallinta omaksi palveluksi ja keskittää osapalveluiden tunnistautuminen siihen. Tämän tutkielman pääpaino on tunnetulle osajoukolle, esimerkiksi yrityksen työntekijöille, suunnatuissa palveluissa.

Tutkielmassa pyritään selvittämään kuinka yritys voi rakentaa keskitetyn tunnistautumispalvelun valmiin käyttäjädatan päälle. Lähtökohtaisesti tunnistautumista vaativat palvelut ovat web-pohjaisia, mutta myös työasemalla tai puhelimella käytettävät asiakasohjelmat pyritään ottamaan huomioon.

## 4.3 Käyttötapauksia

Millaisia käyttötapauksia kyseisessä järjestelmässä on. Eli avataan käyttötapauskaavioilla ongelmakenttää.

## 4.4 Keskitetyn tunnistaumisen periaatteet

Tunnistautumispalvelimella ja yksittäisellä sovelluksella on joku yhteinen salaisuus (yksityinen ja julkinen avain), jonka avulla tunnistautumispalvelin allekirjoittaa valtuutuksia (token), jolla käyttäjä voi todistaa identiteettinsä. Tähän periaatteeseen perustuvia protokollia on ainakin OAuth ja Kerberos, varmaan SAML:ssa on joku vastaava meininki.

Joku kiva kuva (sekvenssikaavio?), joka visualisoi asian.

Ehkä termikappalekin, johon voi siirtää yleisiä termejä kerberos ja oauth-kappaleista.

Lähteenä voi käyttää vaikka luottettavaakin luotettavampaa Lampsonia & kumppaneita, joka ehkä on kuitenkin jo vähän legacya [LABW92].

## 4.5 Keskitetyn tunnistautumisen rajapintaprotokollat

Yleisesti rajapintaprotokollista, tuodaan periaatteita konkreettiselle tasolle, ehkä vähän historiakatsausta. Alaluvuissa kerrotaan tarkemmin jokaisesta (tässä vaiheessa kerberos, saml ja oauth) protokollasta. Kappaleen oleellisin osa on OAuth, koska sitä tullaan käyttämään toteutusvaiheessa. Kerberos ja SAML esitellään, jotta OAuthin periaatteet avautuu. 3-4 sivua yhteensä.

### 4.5.1 Kerberos

Pituus max sivu, luultavasti tästäkin jotain asioita periaate-kappaleen puolelle.

Kerberos-protokolla on alunperin MIT:ssa kehitetty tunnistautumisprotokolla, jonka nykyisin käytössä oleva versio 5 julkaistiin alunperin syyskuussa 1993 ja päivitettynä heinäkuussa 2005 [NYHR05]. Se on yleisesti käytössä erilaisissa UNIX-pohjaisissa käyttöjärjestelmissä ja myös Microsoft on käyttänyt sitä oletus-tunnistautumismekanismina Windows 2000:sta lähtien [STB02].

Protokollan osapuolia ovat käyttäjä, luotettava kolmas osapuoli ja palvelu, joka vaatii tunnistautumisen. Luotettava kolmas osapuoli on tyypillisesti avaintenjakopalvelin (KDC, Key Distribution Center), joka tunnistaa käyttäjän ja myöntää lipun tunnistautuneelle käyttäjälle. Myönnettyyn lippuun on merkattu palvelu, johon sitä voidaan käyttää ja aikaleima, jonka ajan se on voimassa. Käyttäjä antaa lipun tunnistautumista vaativalle palvelimelle, joka tarkistaa omalla avaimellaan käyttäjän tunnisteen ja aikaleiman, joiden perusteella se myöntää pääsyn palveluun.

Keskitetty tunnistautuminen hajautettuihin järjestelmiin voidaan toteuttaa Kerberos-protokollalla [LC10]. Kerberos on luonteeltaan sopiva hajautettuihin järjestelmiin, koska avaintenjakopalvelin voi jakaa lippuja kaikkiin järjestelmiin, joiden kanssa se on vaihtanut salausavaimet. Tunnistautumispalvelin on tilaton, jolloin sen suorituskykyä voidaan parantaa tarvittaessa skaalaamalla, joten tunnistautumispalvelin voi palvella suurta määrää käyttäjiä [LC10].

Tunnistautumisessa käytetyt yksityiset avaimet tallennetaan tietokantaan, jolloin on riskinä, että kolmas osapuoli saattaa päästä käsiksi näihin avaimiin ja pystyä allekirjoittamaan lippuja. Jakamalla salaiset avaimet osiin ja hajauttaa se avaintenjakopalvelimeen, tunnistautumista vaativalle palvelimelle ja näiden välillä käytetylle reitittimelle [Cao11], voidaan parantaa protokollan luotettavuutta. Tämä tekee siitä mahdollisen vaihtoehdon käytetyksi menetelmäksi keskitettyyn tunnistautumiseen hajautetuissa järjestelmistä.

### 4.5.2 SAML

Pituus max sivu

Security Assertion Markup Language (SAML) on OASIS-komitean kehittämä XML-pohjainen avoin standardi tunnistautumiseen ja autorisointiin. Alunperin standardin versio 1.0 julkaistiin marraskuussa 2002, versio 2.0 maaliskuussa 2005 ja viimeksi

päivitetty versio lokakuussa 2009.

### 4.5.3 OAuth

Tämä kappale tulee luultavasti muuttumaan, yleisiä juttuja johdantoon ja tähän vain Oauth-spesifistä asiaa, miten eroaa kerberoksesta jne. Pituus 2-3 sivua, kuvia yms. Oleellisin näistä protokollista, koska tullaan käyttämään toteutuksessa.

OAuth on avoin tunnistautumisrajapinta hajautetuille web-sovelluksille. Se mahdollistaa käyttäjien resurssien jakamisen palveluiden välillä ilman käyttäjätunnuksen tai salasanan luovuttamista kolmannelle osapuolelle. Se perustuu erilaisten valtuutusavainten (token) välittämiseen palveluiden välillä. OAuth on yleisesti käytössä web-sovelluksissa, joissa halutaan näyttää käyttäjälle kuuluvia resursseja (esimerkiksi valokuvia), jotka sijaitsevat toisessa sovelluksessa [TODO: lähde].

OAuth on määritelty RFC-dokumentissa numero 5849. Sen ensimmäinen versio (1.0) julkaistiin lokakuussa 2007 ja päivitetty versio (1.0a) kesäkuussa 2009 [RHHL11]. OAuthin versio 2.0 on myös kehitteillä ja se on tarkoitus julkaista marraskuussa 2012 [RHHL11].

Alunperin OAuthin kehitystyö alkoi marraskuussa 2006, kun Blaine Cook kehitty Twitter-palveluun OpenID-tukea.

... tarvitaanko tätä?

**Asiakas** - HTTP-asiakas, joka tekee OAuth-tunnistaumiskutsuja.

**Palvelin** - HTTP-palvelin, joka ottaa vastaan OAuth-tunnistautumiskutsuja.

**Suojattu resurssi** - Resurssi, joka voidaan saada palvelimelta, jos pyyntö on OAuth-tunnistettu.

**Resurssin omistaja** - Omistaa suojatun resurssin ja valvoo siihen pääsyä.

**Valtuutustieto (credentials)** - Valtuutustieto koostuu yksilöivästä tunnisteesta ja siihen liittyvästä salaisesta avaimesta. OAuthissa käytetään kolmen tasoisia valtuutusavaimia: asiakasavaimia, jotka yksilöi ja varmistaa asiakasohjelmiston, väliaikaisia avaimia, joilla autorisoidaan pyyntö ja valtuutusavaimia, joilla pyyntö hyväksytään.

**Valtuutusavain (token)** - Palvelimelta saatava yksilöivä tunniste, jonka avulla asiakas voi pyytää resurssin omistajalta suojattua resurssia. TODO: yeah?

## 4.6 Yhteenveto

OAuth 2.0:n on mahdollista tarjota Kerberos-yhteensopivuus. Kerberos taas ei tarjoa OAuth-yhteensopivuutta, joten OAuth > Kerberos.

http://tools.ietf.org/html/draft-hardjono-oauth-kerberos-01

# 5 Toteutus

Toteutuksen kuvaus. Arkkitehtuuri yms, koodi liitteenä jos tarvii?

Käytännössä kyseessä on palveluna toimiva tunnistautumisjärjestemä, jolla on joku tietokanta ja rajapinta, jonka kautta tunnistautumispyyntöjä tehdään. Todennäköisesti käyttäjätietokanta on relaatiokanta ja rajapinta Oauth.

Tässä luvussa kuvataan ko. toteutus tarpeellisella tarkkuudella. Järjestelmän arkkitehtuuria, viestinkulkua yms kuvataan UML-tekniikalla.

Tutkielman tarkoitus on toteuttaa prototyyppi sovelluksesta, jonka avulla erillisten web-sovellusten käyttäjähallinta keskitetään yhteen komponenttiin. Sovellus hoitaa käyttäjän tunnistautumisen ja käyttäjätietojen hallinnan, joka synkronoidaan asiakasyrityksen käyttäjätietokannan kanssa.

Pituus n. 10sivua.

# 6 Toteutuksen evaluaatio 

Toteutuksen kriittinen arviointi, täyttääkö toteutus sille asetetut odotukset. Vielä hieman auki, mutta pituus noin 5 sivua.

# 7 Toteutuksen laajennettavuus 

Kuinka voisi laajentaa? Kertakirjautuminen ja autorisointi hyviä suuntia. SSO aika peruskamaa OAuthin kanssa, autorisoinnista sen sijaan voi saada ihan mielenkiintoista pohdintaa aikaan.

LDAP:n hyödyntäminen autorisoinnissa, voisiko esimerkiksi käyttäjäryhmät olla tallennettu jotenkin kätevästi LDAP:in ja niiden perusteella voidaan autorisoida pääsy resurssiin.

Sinällään pääsynhallinta tiettyihin palveluihin on jonkin tason autorisointia ja kuuluu ehdottomasti gradun aihepiiriin (toteutus tunnistaa x:n palvelun käyttäjiä samaa käyttäjädataa vasten, kaikille käyttäjille ei saa antaa oikeutta kaikkiin palveluihin).

Pituus yhteensä pari-kolme sivua.

## 7.1 Kertakirjautuminen

SSO on aika triviaali toteuttaa oauthin kanssa, varmaan voi mainita, että se on mahdollista tai jos se tulee toteutuksessa tehtyä, niin kertoa että btw, tämäkin toimii. Käytännössä kai jotain rajoitteita kuitenkin SSO:ssa on. Onko koko kappaleen otsikko sitten laajennettavuus&ongelmat vai mikä, riippuu vähän toteutuksesta.

### 7.2 Pääsynvalvonta

Pääsynvalvonta ei ole kovin triviaali juttu. Kuitenkin jotain vakavamielistä pohdintaa, miten ratkaisua voisi laajentaa siihen suuntaan.

## 8 Yhteenveto

Yhteenveto kaikesta edellämainitusta. Pituus varmaan n. 1-2 sivua.

## Lähteet


Cao11 Cao, L.-C., Secure secret-key management of kerberos service. Teoksessa *Emerging Research in Artificial Intelligence and Computational Intelligence*, Deng, H., Miao, D., Wang, F. L. ja Lei, J., toimittajat, osa 237 sarjasta *Communications in Computer and Information Science*, Springer Berlin Heidelberg, 2011, sivut 76–83, URL `http://dx.doi.org/10.1007/978-3-642-24282-3_11`.

Con99 Conallen, J., Modeling web application architectures with uml. *Commun. ACM*, 42, sivut 63–70. URL `http://doi.acm.org/10.1145/317665.317677`.

ECFR06 Elbaum, S., Chilakamarri, K.-R., Fisher, II, M. ja Rothermel, G., Web application characterization through directed requests. *Proceedings of the 2006 international workshop on Dynamic systems analysis*, WODA '06, New York, NY, USA, 2006, ACM, sivut 49–56, URL `http://doi.acm.org/10.1145/1138912.1138923`.

FH07 Florencio, D. ja Herley, C., A large-scale study of web password habits.

*Proceedings of the 16th international conference on World Wide Web*, WWW '07, New York, NY, USA, 2007, ACM, sivut 657–666, URL http://doi.acm.org/10.1145/1242572.1242661.

How95 Howes, T., The lightweight directory access protocol: X.500 lite. Tekninen raportti CITI Technical Report 95-8, University of Michigan, July 1995.

Kiz09 Kizza, J. M., Authentication. Teoksessa *A Guide to Computer Network Security*, Kizza, J. M., toimittaja, Computer Communications and Networks, Springer London, 2009, sivut 207–225, URL http://dx.doi.org/10.1007/978-1-84800-917-2_10.

LABW92 Lampson, B., Abadi, M., Burrows, M. ja Wobber, E., Authentication in distributed systems: theory and practice. *ACM Trans. Comput. Syst.*, 10, sivut 265–310. URL http://doi.acm.org/10.1145/138873.138874.

LC10 Lai-Cheng, C., Enhancing distributed web security based on kerberos authentication service. *Proceedings of the 2010 international conference on Web information systems and mining*, WISM'10, Berlin, Heidelberg, 2010, Springer-Verlag, sivut 171–178, URL http://dl.acm.org/citation.cfm?id=1927661.1927689.

Lyn11 Lynch, L., Inside the identity management game. *Internet Computing, IEEE*, 15,5(2011), sivut 78 –82.

mAYLL+09 man Au Yeung, C., Liccardi, I., Lu, K., Seneviratne, O. ja Berners-Lee, T., Decentralization: The future of online social networking. *W3C Workshop on the Future of Social Networking Posi-*

*tion Papers*, 2009, URL http://www.w3.org/2008/09/msnws/papers/decentralization.pdf.

NYHR05 Neuman, C., Yu, T., Hartman, S. ja Raeburn, K., The Kerberos Network Authentication Service (V5), RFC 4120 (Proposed Standard), July 2005. URL http://www.ietf.org/rfc/rfc4120.txt. Updated by RFCs 4537, 5021.

RHHL11 Recordon, D., Hardt, D. ja Hammer-Lahav, E., The oauth 2.0 authorization protocol. *Network Working Group*, 5849, sivut 1–47. URL http://tools.ietf.org/html/draft-ietf-oauth-v2-16.

SBHB10 Sun, S.-T., Boshmaf, Y., Hawkey, K. ja Beznosov, K., A billion keys, but few locks: the crisis of web single sign-on. *Proceedings of the 2010 workshop on New security paradigms*, NSPW '10, New York, NY, USA, 2010, ACM, sivut 61–72, URL http://doi.acm.org/10.1145/1900546.1900556.

SH06 Shan, T. ja Hua, W., Solution architecture for n-tier applications. *Services Computing, 2006. SCC '06. IEEE International Conference on*, sept. 2006, sivut 349 –356.

SS96 Sandhu, R. ja Samarati, P., Authentication, access control, and audit. *ACM Comput. Surv.*, 28, sivut 241–243. URL http://doi.acm.org/10.1145/234313.234412.

SS07 Saklikar, S. ja Saha, S., Next steps for security assertion markup language (saml). *Proceedings of the 2007 ACM workshop on Secure web services*, SWS '07, New York, NY, USA, 2007, ACM, sivut 52–65, URL http://doi.acm.org/10.1145/1314418.1314427.

STB02 Swift, M., Trostle, J. ja Brezak, J., Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols, RFC 3244 (Informational), February 2002. URL `http://www.ietf.org/rfc/rfc3244.txt`.

YsJ10 Yong-sheng, Z. ja Jing, Y., Research of dynamic authentication mechanism crossing domains for web services based on saml. *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, osa 2, may 2010, sivut V2–395 –V2–398.