Ville Liukkonen

OPISKELIJOIDEN PERUSREKISTERIN INTEGROINTI IDENTITEETINHALLINTA-JÄRJESTELMÄÄN

Opinnäytetyö
Sähköisen asioinnin ja arkistoinnin koulutusohjelma

Marraskuu 2010

MIKKELIN AMMATTIKORKEAKOULU
Mikkeli University of Applied Sciences

KUVAILULEHTI

MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences	**Opinnäytetyön päivämäärä** 27.11.2010
Tekijä(t) Ville Liukkonen	**Koulutusohjelma ja suuntautuminen** Sähköisen asioinnin ja arkistoinnin koulutusohjelma

Nimeke

Opiskelijoiden perusrekisterin integrointi identiteetinhallintajärjestelmään

Tiivistelmä

Opinnäytetyöni on osa Mikkelin ammattikorkeakoulun identiteetinhallintajärjestelmän integrointiprojektia, jonka tavoitteena oli liittää henkilökunnan ja opiskelijoiden perusrekisterit identiteetinhallinnan identiteettien lähdejärjestelmiksi.

Kehityshankkeenani oli valmistella integrointi opiskelijahallinnanjärjestelmän tietokantaan ja toteuttaa tekniset ratkaisut, joita integrointiin tietokannassa tarvittiin. Opinnäytetyössäni käyn läpi projektia vaihe vaiheelta aina esiselvityksestä käyttöönottoon ja toteutuksen arviointiin. Tutkimusongelmana oli, kuinka järjestelmäintegraatio opiskelijahallintajärjestelmän osalta toteutetaan.

Viitekehyksessäni luon katsauksen siihen, mitä identiteetinhallinta on ja mitä tietoturvallisuuden osa-alueita minun tuli huomioida käsitellessäni opiskelijahallintajärjestelmän tietokantaa, joka muodostaa henkilörekisterin. Lisäksi selvitän, mitä järjestelmäintegraatiolla tarkoitetaan, sen historiaa ja mitkä ovat sen edut ja haitat.

Työni lopuksi arvioin lopputuloksen lisäksi, mitkä ovat identiteetinhallinnan tulevaisuudennäkymät. Onko identiteetinhallinta vain tämän hetken muoti vai onko se yksi tulevaisuuden perusjärjestelmistä.

Asiasanat (avainsanat)

identiteetti, integraatio, järjestelmäarkkitehtuuri, järjestelmänhallinta, käyttäjätunnukset, tiedonhallinta, tietoturva, tunnistaminen

Sivumäärä	**Kieli**	**URN**
56 + 4 liitteet	suomi	

Huomautus (huomautukset liitteistä)

Ohjaavan opettajan nimi Jukka Selin	**Opinnäytetyön toimeksiantaja** Mikkelin ammattikorkeakoulu

DESCRIPTION

MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences	**Date of the master's thesis** 27 November 2010
Author(s) Ville Liukkonen	**Degree programme and option** eServices and Digital Archiving

Name of the master's thesis

The Integration of Student Management database to the Identity Management System

Abstract

My master's thesis was a part of a bigger project in Mikkeli University of Applied Sciences. The main goal for this project was to integrate staff and student databases to the identity management system that members of both groups could use their usernames and passwords when they access any of the network services in use.

My task was to compile the student management database for the integration. I had to design and implement the tools required in the database in terms of the integration. In general the main target of my master thesis and its research problem was to solve how to integrate the database to the identity management system and how it should be done.

The master's thesis could be described as a report of the whole project step by step. I also explained what the terms identity management and system integration meant and what they were used for. The student management database is an index of individuals and because of that a part of my thesis examined what kind of information security rules and laws affected the identity project.

In the end I analyzed the project, its outcome and how could I improve the system. I also discussed the future of identity management.

Subject headings, (keywords)

data management, identification, identity, information security, integration, system architecture, system management, user identification

Pages	**Language**	**URN**
56 + 4 appendixes	Finnish	

Remarks, notes on appendices

Tutor	**Master's thesis assigned by**
Jukka Selin	Mikkeli University of Applied Sciences

SISÄLTÖ

LIITE/LIITTEET

1 Käsitematriisi
2 Herätteiden PL/SQL koodi

1 JOHDANTO

Mikkelin ammattikorkeakoulussa oli tavoitteena siirtyä identiteetinhallintaan (IDM), joka tässä tapauksessa tarkoittaa sitä, että kaikkia tietojärjestelmiä käytetään yhden identiteetin eli tunnuksen kautta. Projekti alkoi opiskelijoista ja opiskelijahallintajärjestelmän (ASIO) integroinnista identiteetinhallinnan osaksi ja tunnusten lähteeksi. Suunnitteluvaihe käynnistyi vuoden 2009 keväällä ja toteutus sekä testaus loppukesällä 2009. Käyttöönotto tapahtui syksyllä 2009 ja se jatkui vuoden 2010 alussa, jolloin seuraava saapumiserä opiskelijoita aloitti.

Tämä opinnäytetyö käsittelee koko sitä projektin osaa, jonka tarkoituksena oli selvittää, mitä kaikkea opiskelijahallintajärjestelmän liittäminen osaksi identiteetinhallintajärjestelmää vaati, mitä siinä oli huomioitava ja miten se tehtiin. Teknisesti opinnäytetyöni keskittyy niihin ratkaisuihin ja ongelmiin, joita esiintyi opiskelijahallintajärjestelmässä, tietokannassa ja tietokannan toiminnoissa. Ulkopuolelle olen jättänyt varsinaisen identiteetinhallinnan toiminnot ja niistä on huomioitu vain ne rajoitukset ja vaatimukset, jotka vaikuttivat opiskelijanhallintajärjestelmään. Näin opinnäytetyössäni käsitellään tietovirtaa, joka alkaa tiedon synnystä opiskelijahallintajärjestelmään ja päättyy rajapintaan, jonka kautta tieto siirretään identiteetinhallintajärjestelmään. Käyn kuitenkin läpi myös identiteetinhallinnan perusteet sekä tietoturvallisuutta ja lainsäädäntöä henkilötietojen käsittelyn osalta.

Identiteetinhallintaprojektin tarkoituksena oli luoda järjestelmä, jossa opiskelijoilla on yksi ainoa identiteetti eli tunnus sähköisissä järjestelmissä. Opiskelijahallintajärjestelmä ASIOn merkitys oli olla se paikka, missä opiskelijan tunnus ensisijaisesti syntyy ja missä se niin sanotusti kuolee eli poistetaan käytöstä. Tämä tieto välitetään erilliseen identiteetinhallintajärjestelmään, joka yhdistää opiskelijoiden tunnuksen tai tunnukset henkilötunnukseen perustuvaan identiteettiin, joka puolestaan mahdollistaa muiden sähköisten järjestelmien käyttämisen ASIOssa olevilla opiskelijatunnuksilla.

Tavoitteena tällaisessa identiteettiin perustuvassa järjestelmässä on lisätä niin ylläpidon kuin käytönkin helppoutta. Opiskelijan ei tarvitse muistaa erikseen useita tunnuksia eri verkkopalveluihin, kuten sähköpostiin, Moodleen, verkon levyihin ja niin edelleen. Hän voi kirjautua jatkossa kaikkiin omalla opiskelijatunnuksellaan ja vaihtaa

salasanaansa yhdessä paikassa, joka on hänen tärkein verkkopalvelunsa opiskeluaikana.

Ylläpidolle tämä yhden tunnuksen tekniikka helpottaa hallintaa, kun tunnukset hallinnoidaan yhden järjestelmän kautta, joka välittää sen sitten automaattisesti muihin järjestelmiin. Kun tähän lisätään luonti- ja poistoautomatiikkaa, joka oli myös tämän projektin tarkoitus, ei ihmisen tarvitse olla mukana järjestelmän näkökulmasta katsottuna koko tunnuksen elinkaaren aikana. Opiskelijan aloittaessa opintonsa hän saa käyttöönsä tarvittavat järjestelmät ja hänen lopettaessaan opiskelut tunnukset poistuvat – automaattisesti.

Opinnäytetyöni keskittyy siihen, mikä opiskelijahallintajärjestelmässä on olennaista identiteetinhallintajärjestelmän osalta, mistä se tieto löytyy ja miten se välitetään rajanpintaan, josta IDM-järjestelmä saa sen käyttöönsä. Projektissa oli myös tarkoitus selvittää, mitä mahdollisia ongelmia ja erityisiä vaatimuksia voi syntyä, joita ei ole osattu ennakoida. Näitä oppeja hyödynnetään sitten identiteetinhallintajärjestelmän toisessa osaprojektissa henkilökunnan tunnusten osalta.

2 IDENTITEETINHALLINTA ELI IDENTITY MANAGEMENT (IDM)

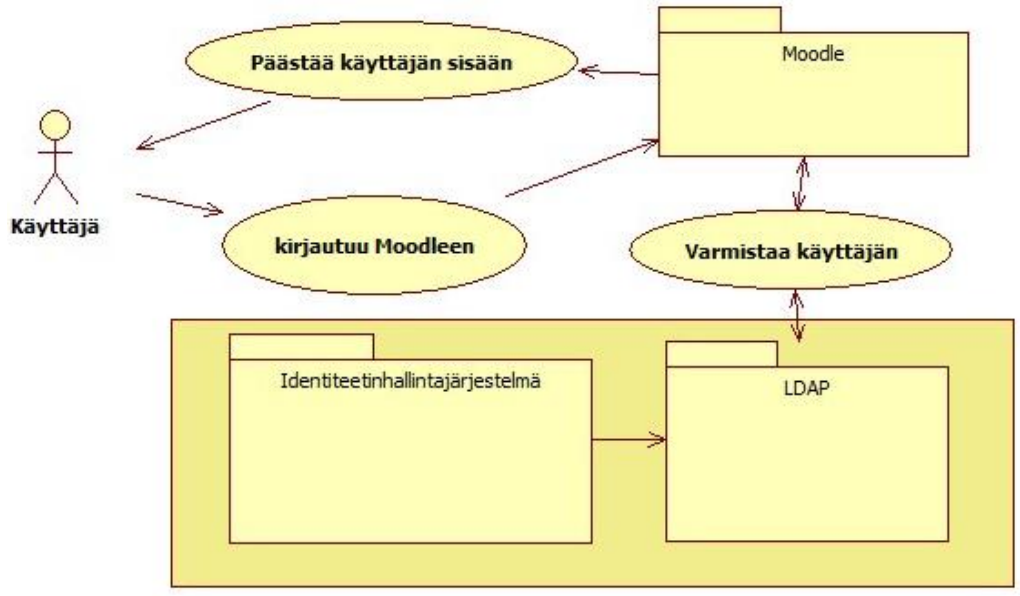
Identiteetinhallinta ei ole käsitteenä niin yksiselitteinen kuin moni muu tietotekninen termi. Osittain syynä on sanan pitkä historia, sillä identiteetinhallintaa on ollut olemassa jo tietotekniikan alkuajoilta lähtien. Ensimmäisten tietokoneiden yksinkertainen käyttäjänhallinta on kaukana nykyajan keskitetyistä verkkokirjautumispalveluista. (Linden 2009, 1.) Identiteetinhallinta ei nykyisin tarkoita vain käyttäjän kirjautumista järjestelmään tunnuksellaan ja salasanallaan, jolloin hallintajärjestelmä antaa hänelle käyttäjän tilin (account) mukaiset oikeudet. IDM voi vahvistaa ulkopuolisille järjestelmille käyttäjän olevan toisen järjestelmän hyväksymä käyttäjä tai identiteetti voi olla vain passiivinen tietokanta, jonne käyttäjä ei itse koskaan voi kirjautua, vaan identiteettiä käytetään tietovarastona. (Linden 2009, 9 – 23.) Identiteetinhallinnan tehtävänä on usein valvoa pääsyä muihin järjestelmiin, minkä vuoksi siitä käytetään myös termiä Identity and Access Management (IAM) (Engelbert 2009, 2 – 3).

Yleisesti identiteetinhallintajärjestelmällä tarkoitetaan nykyisin verkossa toimivaa hakemistopalvelua. Tällöin puhutaan keskitetystä käyttäjänhallinnasta ja tunnettuja hakemistopalveluja ovat esimerkiksi Novellin eDirectory ja Microsoftin Active Directory. Hakemistopalvelut ovat yleensä aina lähiverkkoon integroituja, mikä tarkoittaa sitä, että jokaisella lähiverkolla on oma hakemistopalvelunsa, joka hallinnoin siinä verkossa olevia käyttäjien identiteettejä. Hakemistopalvelu voi kuitenkin olla myös lähiverkon ulkopuolinen palvelu. Tällaisia hakemistopalveluja on yleensä internetissä, missä hakemistopalveluun voi kirjautua mistä verkosta tahansa ja käyttää kirjautumisen jälkeen internetin kautta tiettyjä palveluja (Linden 2009, 1.) Hakemistopalvelussa on määritetty tarkasti, mitä käyttöoikeuksia milläkin tunnuksella on mihinkin resurssiin.

2.1 Hakemistopalvelut

Lähiverkoissa identiteetinhallinnan perusidea on, että käyttäjä kirjautuu verkossa olevilta koneilta verkkoon käyttäen tunnuksia, jotka ovat hakemistopalvelun identiteetinhallinnassa. Hyväksytyn tunnistautumisen jälkeen käyttäjä voi käyttää lähiverkossa olevia palveluita, esimerkiksi verkkolevyjä tai verkkotulostimia, määritettyjen käyttöoikeuksien puitteissa. Yleensä käyttäjä voi myös tarkastella identiteettinsä tietoja ja tarvittaessa muuttaa niitä. Nykyisin lähiverkkoon on usein kytketty erillisiä järjestel-

miä tai ohjelmistoja, kuten esimerkiksi oppimisympäristöjä, jotka vaativat kirjautumisen. Niin helppokäyttöisyyden kuin ylläpidonkin kannalta on järkevää, että nämä ympäristöt käyttäisivät samoja tunnuksia kuin verkon omat palvelut. Tämä on ratkaistu LDAP (Lightweight Directory Access Protocol) standardilla. Identiteetinhallinta tarjoaa tietojaan LDAP standardin mukaisessa muodossa verkkoon. Monet ohjelmistot ja järjestelmät ymmärtävät LDAP standardia ja voivat käydä vahvistamassa käyttäjätunnukset sieltä. Näin käyttäjä voi syöttää normaalin verkkotunnuksensa näihinkin järjestelmiin, jolloin ne vahvistavat verkon identiteetinhallinnasta, että käyttäjä on kelvollinen verkon käyttäjä, ja päästää hänet järjestelmään (kuva 1). LDAP antaa näin mahdollisuuden identiteetin vahvistamiseen, mutta se ei varsinaisesti määritä käyttöoikeuksia, vaan kirjautumista pyytävä ohjelma tekee sen itse. (Howes ym. 1999, 45 – 47, 67 – 76.)



KUVA 1. Kirjautuminen verkkopalveluun IDM-tunnuksella

Identiteetinhallinta voi olla myös verkon ulkopuolinen hakemistopalvelu, jolloin se on käytännössä internetissä. Mikael Linden sanoo julkaisussaan Organisational and Cross-Organisational Identity Management (2009, 1), että yritykset luoda yhtä maailmanlaajuista identiteetinhallintajärjestelmää internetiin ovat epäonnistuneet. Esimerkiksi hän on nostanut Microsoftin Passport palvelun. Internetissä kuitenkin on toimivia identiteetinhallintajärjestelmiä, kuten juuri mainittu Passport (www.passport.net), mutta sen käyttö vain ei ole niin laajaa kuin alun perin on haaveiltu. Microsoft käyttää Passportia monissa omissa palveluissaan ja sinne tunnistautuneet käyttäjät voivat

käyttää muun muassa yrityskäyttöön tarkoitettua median jakelupalvelua, mistä voidaan ladata yrityksen käyttöön Microsoftin ohjelmistoja, joihin lisenssioikeudet löytyvät. Nykyisin Passport tunnetaan myös nimellä Live ID. (Microsoft 2009, Microsoft 2010.) Toinen jo varsin vanha internetin kautta maailmanlaajuinen identiteetinhallintajärjestelmä on nimipalvelimien muodostama verkko. Vaikka nimipalvelimet eivät hallitsekaan käyttäjätunnuksia, niin niiden muodostama verkko on internetissä olevien laitteiden identiteetinhallintajärjestelmä. Tämä identiteetti sisältää tiedon laitteen osoitteesta ja sen nimestä, joiden avulla käyttäjät ohjataan oikeaan paikkaan. (Howes ym. 1999, 45 – 47.)

Jos ajatellaan identiteetinhallintaa internetissä hieman laajemmin, niin hyvin monella käyttäjällä on nykyisin identiteetti jossain järjestelmässä. Internetissä identiteetit vain usein eivät ole tarkistettuja. Hyväksi esimerkiksi voidaan ottaa Facebook. Facebook on omanlaisensa identiteetinhallintajärjestelmä, joka on tarkoitettu sosiaalista mediaa varten. Facebookin kautta on käytettävissä satoja erilaisia internetpohjaisia ohjelmia, jotka tunnistavat käyttäjän Facebookin kautta. Facebookin kaltaisia IDM ratkaisuja ei pidetä oikeina, koska sinne kuka tahansa voi luoda millaisen identiteetin tahansa, eikä tietoja tarkasteta mistään. Juuri tämä on ollut ongelmana nettipohjaisen maailmanlaajuisen IDM:n rakentamisessa. On kuitenkin selvää, että lähitulevaisuudessa organisaatioiden tiedonhallinnan piiriin halutaan liittää myös oman kotiverkon ulkopuolisia sähköisiä palveluita ja sosiaalisia verkostoja, joita käytetään esimerkiksi ryhmätyövälineinä (Kaario & Peltola 2008, 148 – 150.) Suomessa käytetään vahvaan internetkirjautumiseen lähinnä suomalaisten pankkien verkkopankkitunnuksia, mutta pankkien verkkopankit eivät varsinaisesti ole erillisiä identiteetinhallintajärjestelmiä, vaan ne ovat pankkien verkoissa toimivia hakemistopalveluita, joihin vain on tunnistautumispalvelu internetistä. Pankkien tunnistautumispalvelut ovatkin käytännössä yritysten välistä identiteetinhallintaa.

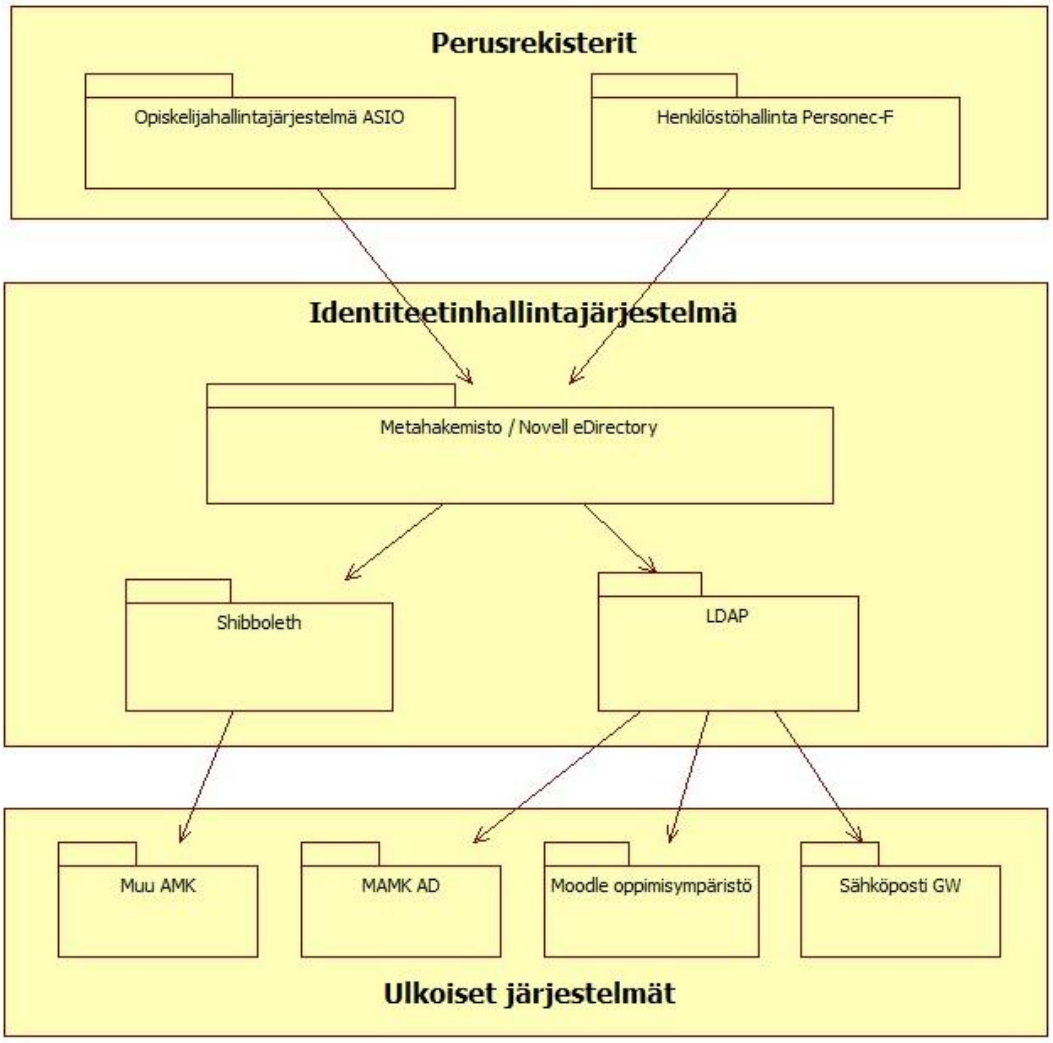
Suomessa korkeakouluilla on käynnissä HAKA-hanke, jonka tarkoituksena on luoda yksi yhteinen identiteetinhallintajärjestelmä tai paremminkin verkko. Perusajatuksena HAKAssa on, että korkeakoulujen IDM -järjestelmät ovat yhteydessä toisiinsa ja luottavat toistensa käyttäjiin. Näin esimerkiksi Tampereen ammattikorkeakoulun opiskelija voi käyttää Mikkelin ammattikorkeakoulun verkkopalveluita. Mikkelin identiteetinhallinta varmistaa Tampereen identiteetinhallinnasta Shibboleth ohjelmiston-avulla, että kirjautumista yrittävä todella on luotetun korkeakoulun opiskelija ja myönteisen

vastauksen saadessaan päästää hänet palveluihin, vaikka opiskelija ei Mikkelin oman IDM:n piirissä olisikaan. (CSC 2010.)

Identiteetinhallintajärjestelmiä, joihin käyttäjä kirjautuu ja voi jopa muuttaa tietojaan, kutsutaan usein online-hakemistoiksi. Yleensä IDM on online-tyyppinen, koska hakemistopalveluita käytetään käyttäjäoikeuksien myöntämiseen eri resursseille, mutta on olemassa myös offline-tyyppinen eli passiivinen hakemistopalvelu. Tällaisia identiteetinhallintajärjestelmiä ovat sellaiset, joiden tietoa käytetään muuhun kuin käyttöoikeuksiin. Identiteettitietoja voidaan käyttää henkilörekisterin tavoin tietovarastona. Hyvä esimerkiksi offline IDM:stä on veroviraston järjestelmä. Verohallinnon järjestelmässä on identiteetti jokaisesta suomalaisesta. Identiteetti sisältää tiedot ihmisestä, hänen varallisuudestaan, pankistaan, työnantajastaan ja niin edelleen. Viime vuosiin saakka tietoja on voinut vain katsella, eikä veroviraston tietoja ole voinut käyttää kirjautumiseen. Viime vuosina tosin verovirastonkin järjestelmä on muuttumassa online-tyyppisemmäksi, koska veroilmoitustaan voi jo täydentää sähköisesti, mutta se ei suoraan muuta identiteettiä. Verotustietojaan voi vain selailla. (Linden 2009, 23.)

2.2 Perusrekisterit ja synkronointi

Identiteetinhallintajärjestelmän perustehtäviin kuuluu usein käyttäjänhallinta, mutta esimerkiksi Mikkelin ammattikorkeakoulun kaltaisessa ympäristössä niin opiskelijoiden kuin henkilöstönkin perustiedot ovat omissa järjestelmissään ja niistä oleelliset tiedot on siirrettävä IDM:ään. Tämän vuoksi on tärkeää, että identiteetinhallintajärjestelmä on integroitavissa ulkoisiin järjestelmiin – perusrekistereihin – jotka toimivat identiteettitietojen lähteenä. Perusrekistereitä voi olla useampia ja identiteetinhallinta koostaa niistä yhden hakemiston, jota kutsutaan metahakemistoksi. Tästä hakemistosta identiteetinhallinta välittää tietoja muihin järjestelmiin, mikäli tietoja niissä tarvitaan. Kokonaisuutta voi tarkastella kuvasta 2, joka mallintaa integroitua identiteetinhallintaa Mikkelin ammattikorkeakoulussa. Metahakemisto voi olla myös virtuaalinen, jolloin identiteetinhallinta ei tee perusrekistereiden tiedoista kopiota omaan järjestelmäänsä, vaan tarjoaa yhtä näkymää ulko-puolisille järjestelmille ja hakee tiedot taustalla reaaliaikaisesti perusrekistereistä, kun niitä tarvitaan. (Howard ym. 1999, 649 – 672; Linden 2009, 25 – 27.)



KUVA 2. Kokonaiskuva integroidusta IDM-järjestelmästä

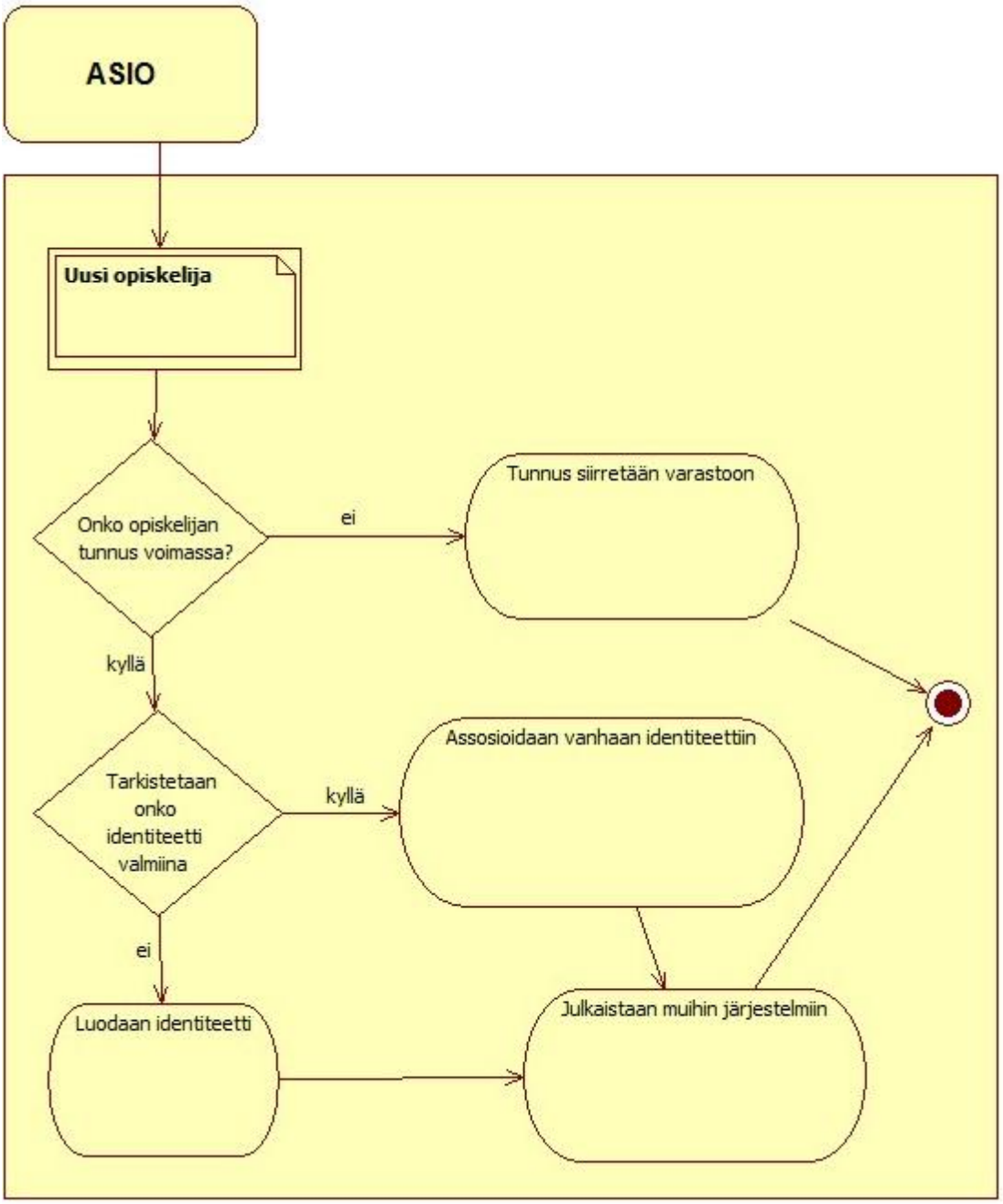
Metahakemiston ollessa kopio on tietojen ajantasaisuus tärkeää huomioida. Metahakemisto päivittää tietonsa tietyin väliajoin perusrekistereistä eli synkronoi tiedot. Synkronointi voi tapahtua molempiin suuntiin. Yhdensuuntaisessa synkronoinnissa muuttuvat tiedot välittyvät perusrekisteristä metahakemistoon, mutta metahakemistossa tapahtunut muutos ei välity perusrekisteriin, vaan synkronoinnin yhteydessä se korvautuu perusrekisterin arvolla. Kahdensuuntaisessa synkronoinnissa myös metahakemistossa tapahtuneet muutokset välittyvät perusrekisteriin. Usein perusrekisterin ja metahakemiston välillä on vain yhdensuuntainen synkronointi perusrekisteristä metahakemistoon päin, koska näin alkuperäisen tiedon tietoturvallisuus on parempi. Esimerkiksi palkanmaksujärjestelmässä oleviin henkilöstötietoihin ei usein haluta muutoksia metahakemistosta, vaikka sen kautta henkilöstö voisi päivittää yhteystietojaan. Metahakemiston ollessa virtuaalinen, ei synkronointia tarvita, koska päivitettävää me-

tahakemistoa ei ole. Tieto haetaan ja tallennetaan suoraan perusrekisteriä käyttäen. (Howard ym. 1999, 649 – 672; Linden 2009, 25 – 27.)

Automatisointi ja single sign-on

Ylläpidollisesti perusrekisterien ja metahakemiston välinen synkronointi on hyvä automatisoida. Automatisoidussa järjestelmässä metahakemisto ja/tai perusrekisterit päivitetään joko eräajona tietyin väliajoin tai reaaliaikaisesti tapahtumiin reagoiden. Tapahtumiin reagoinnilla tarkoitetaan sitä, että aina tiedon syntyessä, muuttuessa tai poistuessa järjestelmä välittää tapahtumatiedon eteenpäin. Kuvan 3 esimerkissä opiskelijatietojen perusrekisteriin lisätään uusi opiskelija. Tällöin järjestelmä ilmoittaa identiteetin-hallintaan uudesta opiskelijalta ja tiedot siirretään metahakemistoon ja opiskelijalle muodostetaan identiteetti. (Howard ym. 1999, 649 – 672; Linden 2009, 25 – 27.)

Automatisointi voidaan toteuttaa myös muualle kuin perusrekisterien ja metahakemiston välille. Uuden identiteetin syntyessä voidaan myös välittää tieto sähköpostijärjestelmälle, joka omien sääntöjensä pohjalta luo opiskelijalle sähköpostinosoitteet ja -tilin sekä välittää tiedot tunnuksista ja osoitteista metahakemiston kautta opiskelijahallintajärjestelmään yhteystietoihin – täysin automaattisesti. (Howard ym. 1999, 649 – 672; Linden 2009, 25 – 27.)



KUVA 3. Automatisoitu käyttäjätunnuksen käsittely

Identiteetinhallintaan yhdistetään usein termi single sign-on eli kertakirjautuminen. Kertakirjautuminen mahdollistaa sen, että käyttäjä kirjautuu kerran identiteettitunnuksellaan verkkoon, minkä jälkeen hänellä on käytössään kaikki järjestelmät, jotka ovat yhteydessä identiteetinhallintaan, ilman erillistä kirjautumista. Käyttäjä voi käyttää järjestelmiä niin kauan kunnes kirjautuu ulos verkosta. On myös mahdollista, että palvelut ovat selainpohjaisia, jolloin käyttäjä voi kirjautua sisään missä tahansa palvelussa ja sen jälkeen häneltä ei kysellä enää tunnuksia muihin palveluihin. Selainpohjaisis-

sa ratkaisuissa on kuitenkin ongelmia uloskirjautumisessa. Usein ainoa tapa kirjautua varmasti ulos on sulkea koko selain. (Linden 2009, 35 – 37.)

2.3 Hyödyt, haasteet ja uhat

Keskitetty identiteetinhallintajärjestelmä on nykyisin lähes pakollinen ratkaisu vähänkin isommassa yrityksessä. Jokainen työntekijä joutuu käyttämään lukuisia eri järjestelmiä ja ilman IDM:ää jokaiseen täytyisi kirjautua eri tunnuksilla. Ilman IDM:ää ylläpidon täytyisi huolehtia kaikista järjestelmistä ja niiden käyttäjähallinnasta järjestelmäkohtaisesti. Identiteetinhallintajärjestelmä on kuin yksi suuri kaavio organisaatiosta, jossa jokaiselle käyttäjälle on jaettu omat oikeutensa tiettyihin resursseihin. Jokainen uusi järjestelmä on helppo kytkeä IDM:ään LDAP-tunnistautumisen kautta, jolloin se saadaan välittömästi ainakin perustietoturvan piiriin ja käyttäjällä on oltava käyttäjä-tunnus ja salasana järjestelmään päästäkseen. (ComputerWeekly.com 2007; NextGov 1997.)

Haasteena identiteetinhallintajärjestelmien käyttöönotossa ovat usein vanhat järjestelmät, jotka eivät ole helposti liitettävissä IDM:ään. Vanhat ohjelmistot saattavat vaatia runsaasti ohjelmointia, jopa vanhoilla ohjelmointikielillä, jotta ne saadaan viestimään identiteetinhallinnan kanssa. Vanhat järjestelmät ovat kuitenkin monessa tapauksessa yhä käytössä, eikä niitä voida vaihtaa uusiin. Lisäksi tietohallinto-osastot ja -palvelut ovat ennestään ylikuormitettuja, eikä aikaa vanhojen järjestelmien liittämiseen ole tai niitä ei ehditä tekemään kunnolla. Tästä voi seurata vakavia tietoturvallisuuden puutteita. (NextGov 1997.)

Identiteetinhallinnalla on myös huonot puolensa. Identiteettivarkaudet ovat lisääntyneet maailmassa ja jos joku onnistuu saamaan käsiinsä toisen ihmisen tunnukset, joilla hän kirjautuu identiteetinhallintajärjestelmään, voi hän käyttää kaikkia IDM:ään kytkettyjä palveluita ja järjestelmiä varastamallaan identiteetillä aiheuttaen mahdollisesti paljonkin tuhoa. Tunnuksia ei välttämättä tarvitse edes varastaa, vaan esimerkiksi internetselainpohjaisissa ratkaisuissa käyttäjä voi luulla olevansa uloskirjautuneena, mutta ilman selaimen sulkemista seuraava selaimen käyttäjä pääsee kertakirjautumisen avustamana alkuperäisen käyttäjän palveluihin ja niiden tietoihin. Yhä kasvavassa määrin riskejä käyttäjätunnuksen ja salasanan käytössä on alettu pienentämään ottamalla käyttöön myös muita tunnistautumismenetelmiä kuten sormenjälkeen, verkko-

kalvon ja kämmenen kuvioon perustuvia tekniikoita. (ComputerWeekly.com 2007; NextGov 1997.) Identiteetinhallintajärjestelmän vaatii niin käyttäjältä kuin ylläpidoltakin huolellisuutta ja hyviä teknisiä ratkaisuja, jotta tietoturvallisuus ja helppokäyttöisyys kulkevat käsi kädessä.

2.4 identiteetinhallinnan onnistunut integrointi

Identiteetinhallinnan käyttöönotossa on hyvä huomioida viisi asiaa, jotta projekti onnistuu mahdollisimman hyvin:

Käyttötarve ja toimintaympäristö

Ennen identiteetinhallintaa on hyvä selvittää, mihin tarkoitukseen järjestelmää halutaan käyttää ja onko se oikeasti sitä, mitä yritys tarvitsee. Ei ole olemassa yhtä oikeaa tapaa pystyttää identiteetinhallintaa, vaan se on jokainen kerta suunniteltava juuri yrityksen oman liiketoiminnan tukemiseen. Jotta identiteetinhallinta voidaan toteuttaa, on tekijöiden ymmärrettävä, mitä yritys tekee ja mitä mahdollisia tietoturvariskejä IDM:n käyttöönotto voi tuota tullessaan. (Engelbert 2009, 3 – 5.)

Oikeiden ihmisten valinta projektiin

Identiteetinhallinnan käyttöönotossa ei voi olla vain tietoteknisen alan osaajia, vaan suunnittelun alusta saakka mukana on oltava niitä ihmisiä, jotka ymmärtävä yrityksen perustoimintaa. He tuovat projektiin tietoa siitä, miten asiat toimivat ja heidän tulee nähdä hyvissä ajoin, mitä hyötyä identiteetinhallinta voi heille antaa ja mitä he siltä toivovat. Myös loppukäyttäjän ajatuksia on hyvä ottaa mukaan jo suunnitteluvaiheessa. Luonnollisesti myös tietoteknisiä ihmisiä tarvitaan, koska heidän vastuulleen jää usein järjestelmän ylläpito, vaikka identiteetinhallinnan toteutus ja käyttöönotto tapahtuisikin ulkoisen toimittajan toimesta. (Engelbert 2009, 3, 5 – 7.)

Asteittainen käyttöönotto

Organisaationlaajuisen identiteetinhallinnan ottaminen käyttöön ja useiden järjestelmien liittäminen siihen vie paljon aikaa, mutta on viisasta ottaa IDM käyttöön asteittain. Projekti voi olla vielä monelta osin kesken, kun ennakkoon valittu ryhmä ottaa

valmistuneen osan käyttöön. Näin saadaan arvokasta tietoa järjestelmien toiminnasta ilman, että sen vaikutukset ovat suuria koko organisaation tasolla. Virheitä voidaan korjata ja koko projektin suuntaa korjata, mikäli ilmenee toimimattomuutta toiminnoissa tai toimintojen logiikassa. identiteetinhallinnan osuutta yrityksen toiminnoissa lisätään pala palalta testaten huolellisesti, kunnes se on täysin käytössä. Kerralla koko organisaation tuominen identiteetinhallinnan piiriin on asia, jota tulisi välttää vahinkojen ja riskien minimoimiseksi. (Engelbert 2009, 3, 7 – 9.)

Käyttäjien koulutus

Kaikki käyttäjät ylläpitohenkilöstöstä loppukäyttäjiin tulisi kouluttaa identiteetinhallinnan toimintoihin. Koulutuksen tulisi tapahtua jo ennen identiteetinhallintaprojektin alkamista, eikä se saisi olla vain kertaluonteinen tilaisuus. Kouluttamalla ihmisiä säännöllisin välein saadaan uudet ihmiset myös osaajiksi. Samoin myös ne, jotka vielä tarvitsevat lisäoppia tai ovat syystä tai toisesta olleet estyneitä aikaisempiin koulutuksiin. (Engelbert 2009, 3, 9.)

Valmista ei tule koskaan

Tietotekninen ala on aina muutoksessa ja niin laitteistot kuin ohjelmistotkin päivittyvät nopealla syklillä. Identiteetinhallinnan kanssa ei koskaan voida tulla tilanteeseen, jolloin se olisi täysin valmis, koska siihen liitetyt järjestelmät ja IDM itse päivittyvät jatkuvasti. Ylläpitohenkilöstön on selvitettävä laitteistojen ja ohjelmistojen vaikutukset koko identiteetinhallinnan toimintaan. Yrityksen muun henkilöstön on myös vaihdettava ajatuksia teknisen henkilöstön kanssa, jotta pystytään ennakoimaan muutokset järjestelmissä yrityksen muuhun toimintaan. Samoin yrityksen muut toiminnot heijastuvat identiteetinhallintaan ja muutokset esimerkiksi markkinoinnissa voi vaatia suuriakin muutoksia IDM:ssä, puhumattakaan organisaatiomuutoksen kokoisista tapahtumista. (Engelbert 2009, 3, 10.)

Identiteetinhallinnan tarjoama hyöty voi olla yritykselle suuri, mutta käyttöönottoprojektiin kannattaa lähteä maltillisesti ja huolellisesti suunnitellen. Rauhallisella toteutuksella ja vaiheittain käyttöönotettuna saavutetaan paras lopputulos. Yleensä Identiteetinhallinnan integrointi järjestelmiin epäonnistuu, kun tavoitteet ovat liian korkealla ja tarpeet epäselvät. (Engelbert 2009, 10 – 11.)

3 TIETOTURVALLISUUS HENKILÖTIETOJA KÄSITELTÄESSÄ

Tietoturvallisuus on nyky-yhteiskunnassa paljon käytetty sana ja mediassa usein esillä. Usein tietoturvallisuus mielletään kuitenkin liian suppeaksi käsitteeksi. Joskus jopa kulunvalvontaa pidetään tietoturvallisuuden synonyyminä. Tietoturvallisuus ei kuitenkaan ole vain pääsyn rajoittamista tietoihin tai tiloihin, vaan se koostuu klassisen mallin mukaan luottamuksellisuudesta, käytettävyydestä ja eheydestä. IDM-projektissa itse käytin ohjenuorana kuitenkin modernimpaa tietoturvajaottelua, joka auttaa hahmottamaan tietoturvaa hieman paremmin jakamalla sen useampaan osatekijään: luottamuksellisuuteen, käytettävyyteen, eheyteen, kiistämättömyyteen ja pääsynvalvontaan. (Hakala ym. 2006, 4 – 5.)

3.1 Tietoturvallisuuden osatekijät

Luottamuksellisuus

Luottamuksellisuus tarkoittaa käytännössä sitä, että varmistetaan, ettei mikään asiaton taho pääse järjestelmässä oleviin tietoihin käsiksi (Hakala ym. 2006, 4). Hyvään tiedonhallintatapaan kuuluu, että järjestelmän ylläpitäjä jo suunnitellessa varautuu vikatilanteisiin ja ettei rekisteritietoa joudu vääriin käsiin. Samaa edellyttää myös henkilötietolaki, jota käsitellään tarkemmin luvussa 3.2.

Käytettävyys

Käytettävyydellä tarkoitetaan sitä, että tieto on käytettävissä ja se on oikeassa muodossa. Käytettävyyden esteeksi katsotaan se, että tietoa ei saada järjestelmästä tarpeeksi nopeasti, joka voi syntyä esimerkiksi suuresta verkkoviiveestä. (Hakala ym. 2006, 4.)

Eheys

Jotta tieto olisi eheää, se ei saa olla väärää. Väärää tietoa voi syntyä tahallisesti esimerkiksi tietomurron yhteydessä, jolloin joku väärentää arvoja tai tahattomasti esimerkiksi automaattisen tiedonkäsittelyn aikana. Tällöin puhutaan tiedon korruptoitumisesta. (Hakala ym. 2006, 4.)

Kiistämättömyys

Laajennetussa tietotuvallisuuden määritelmässä kiistämättömyydellä tarkoitetaan tietojen käytön valvontaa. Järjestelmän tulee varmistaa käyttäjän oikeudet tietoon sekä pitää lokitietoa, josta voidaan tarkastaa, kuka tietoa on käyttänyt. Näin voidaan olla varma tiedon alkuperästä. (Hakala ym. 2006, 5.)

Pääsynvalvonta

Pääsynvalvonnalla tarkoitetaan tietoturvallisuuden ensimmäistä tasoa. Kun luottamuksellisuus ja kiistämättömyys keskittyvät käyttäjän valvontaan ja käyttöoikeuksien varmista tiedon suhteen, pääsynvalvonta käsittää pääsyn fyysisille laitteille, verkkoyhteyksiin ja tiloihin. (Hakala ym. 2006, 5 – 6.)

3.2 Lainsäädäntö

Lainsäädännössä ei ole olemassa erikseen lakia, joka käsittelisi tietoturvallisuutta, vaan tietoturvallisuus muodostuu useammasta eri laista, määräyksestä ja säännöksestä. Näistä tärkeimpiä ovat

- Suomen perustuslaki (11.6.1999/731) 10§ (Yksityiselämän suoja ja luottamuksellisen viestin salaisuus),
- Laki viranomaisten toiminnan julkisuudesta (21.5.1999/621) 18§ (Hyvä tiedonhallintatapa) ja erityisesti henkilötietoja käsiteltäessä
- Henkilötietolaki (22.4.1999/523) (Henkilötietojen käsittelyä koskevat periaatteet). (VAHTI 2004)

Henkilötietolain kolmannen pykälän kohdassa neljä määritellään rekisterinpitäjä ja se on ammattikorkeakoulun tapauksessa aina korkeakoulu, ei yksittäinen henkilö. Näin ollen vastuussa rekisteritietojen käsittelystä on aina ammattikorkeakoulu, ei yksittäinen tietohallinnon työntekijä tai muun osaston työntekijä. (Jäppinen 2007, 23.) Ammattikorkeakoulun toimiessa julkisilla varoilla voidaan katsoa, että sen toiminnan on myös noudatettava viranomaisille julkisuuslaissa määrättyä hyvää tiedonhallintatapaa, joka edellyttää tietotuvallisuuden osatekijöiden huomioimista laadukkaasti (Mäenpää 2009, 251 – 253).

Suunnitteluvelvollisuus

Laadukkaan rekisteritietojen varmistamiseksi henkilötietolaki ohjaa rekisterinpitäjiä suunnitelmallisuuteen. Kun henkilörekisterejä perustetaan tai niiden tietoja käsitteleviä järjestelmiä rakennetaan, on jo suunnitteluvaiheessa huomioitava henkilötietolain vaatimukset. Suunnitteluvaiheen oleellisia vaatimuksia on selvittää, mistä henkilötiedot saadaan, mihin niitä tarvitaan ja miten ne ovat oleellisia ylläpitäjän toiminnon kannalta. Myös rekisteriseloste on tehtävä, mielellään ennen tietojen keräämistä. (Jäppinen 2007, 37.)

Käyttötarkoitussidonnaisuus

Henkilötietolain edellyttämä käyttötarkoitussidonnaisuus tarkoittaa sitä, että kerättyjä tietoja ei saa käyttää muuhun kuin siihen, mihin ne on tarkoitettu ja kerätty. Tästä poiketen tietoja voidaan käyttää historiallisen ja tieteelliseen tutkimukseen sekä tilastojen laatimiseen. Henkilörekisterien yhdistämistä esimerkiksi osoitetietojen päivittämiseksi, kun tarkoituksena ei ole luoda uutta rekisteriä muuhun tarkoitukseen, joka on sopimaton kerättyjen tietojen käyttötarkoituksen kanssa, pidetään hyväksyttävänä. (Hallituksen esitys 96/1998.)

Tarpeellisuus ja virheettömyys

Opiskelijoiden henkilötietoja käsitellessä henkilötietolaki määrää, että vain tarvittavia tietoja saa käsitellä. Kun ollaan rakentamassa identiteetinhallintajärjestelmää, johon tuodaan opiskelijoiden tietoja opiskelijanhallintajärjestelmästä, ei opiskelijarekisteristä saada ottaa toiseen rekisteriin kuin välttämättömät tiedot, joita identiteetinhallinnasta tarvitaan. Identiteetinhallintaan ei voida varmuuden varalta siirtää tietoja, joita ehkä tulevaisuudessa tarvitaan. (Hallituksen esitys 96/1998, Konstari 1992, 145.)

Kun tietoja käsitellään ja niitä siirretään järjestelmästä, tulee rekisterin ylläpitäjän varmistaa, että tiedot ovat virheettömiä. Erityisen tärkeää tietojen virheettömyys on, jos tietojen perusteella tehdään päätöksiä, jotka vaikuttavat opiskelija oikeuksiin. Mikäli rekisterissä on vanhentunutta, puutteellista tai tarpeetonta tietoa, on rekisterinpitäjä velvoitettu poistamaan tai korjaamaan tiedot oma-aloitteisesti tai rekisteröidyn pyynnöstä ilman viivytystä (Henkilötietolaki 523/1999, 9§.)

Huolellisuus

Henkilötietolain viides pykälä edellyttää henkilörekisterien ylläpitäjiltä henkilötietojen laillista käsittelyä, huolellisuutta ja yleistä hyvää tietojenkäsittelytapaa. Erityisen tärkeää huolellisuuden osalta on yksityisyyden suojan toteutuminen. Yksityisyyden suojan korostus näkyy selvimmin siitä, että rekisterinpitäjä on korvausvelvollinen taloudellisten ja muiden vahinkojen osalta, joita rekisteröidylle aiheutuu, vaikka vahinkoa ei olisi aiheutettu tahallisesti. Henkilötietolaki pyrkii ohjaamaan rekisterien ylläpitoa siihen, että ylläpitäjä toimii oma-aloitteisesti huolehtimaan yksityisyydensuojasta mahdollisimman hyvin. (Hallituksen esitys 96/1998; Saarenpää, 82 – 83.)

4 JÄRJESTELMÄINTEGRAATIO

Järjestelmäintegraatiolla tarkoitetaan yhteensopimattomien järjestelmien liittämistä toisiinsa siten, että ne voivat kommunikoida keskenään ja lähettää tietoja toisilleen. Järjestelmäintegraatio ei ole mikään tietty tekniikka tai ratkaisu, vaan se on enemmänkin toimintamalli, jonka avulla järjestelmät liitetään toisiinsa. Tekniikoita ja tapoja on useita, joista integraatiota suunniteltaessa on valittava omaan tarpeeseen juuri se sopiva. (Lahti 2003, 3 – 4; Tähtinen 2005, 13 – 14, 48.)

4.1 Järjestelmäintegraation historia ja tausta

Järjestelmäintegraatio on käännös englanninkielisestä termistä ”system integration” ja se on tunnettu jo 1950-luvulta, vaikka aluksi järjestelmäintegraatiot olivatkin harvinaisia. 50-luvulla yritysten järjestelmät olivat yksittäisiin toimintoihin suunniteltuja, eivätkä ne juurikaan kommunikoineet keskenään. Seuraavalle vuosikymmenelle saavuttaessa alkoi kuitenkin tulla enemmän tarvetta yhdistää eri tietojärjestelmien tietoja. Koska yritysten ohjelmistot olivat usein yrityskohtaisia ja tiettyihin toimintoihin räätälöityjä, eikä yhteisiä standardeja juurikaan ollut, niiden välinen kommunikointi oli vaikeaa toteuttaa. Järjestelmäintegraation haasteet erilaisten ohjelmistojen ja laitteistojen vuoksi eivät ole kadonneet vielä tänäkään päivänä, vaikka niistä oltiin jo tietoisia 60 vuotta sitten. (Kaario & Peltola 2008, 63 – 64; Tähtinen 2005, 17 – 19.)

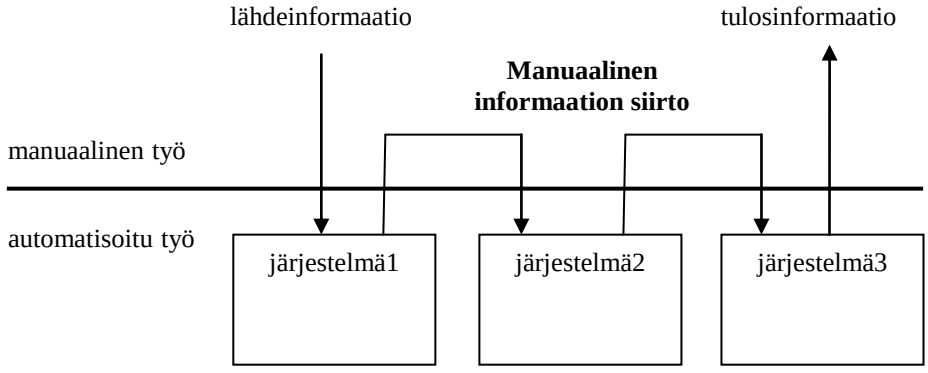
Ensimmäisiä järjestelmäintegraatioprojekteja oli toteutettu jo 40-luvun loppupuolella, mutta kaikkein tunnetuin ja tavallaan myös suurin integraatioprojekti aloitettiin 1960-luvun lopulla. Tämän projektin tarkoituksena oli erityyppisten ja hyvin eri tekniikoilla ja käyttöjärjestelmäalustoilla toimineet tietoliikenneverkot toisiinsa. Lisäksi tähän yhdistettyyn tietoliikenneverkkoon mahdollistettiin yhteyden muodostaminen hyvin erilaisilla päätelaitteilla ja siitä syntyi vähitellen maailmanlaajuinen verkko. Suurella yleisölle tämä maailmanlaajuinen verkko tuli tunnetusti 90-luvulla, jolloin verkon käyttö tavoitti tavalliset käyttäjät World Wide Webin keksimisen myötä. Syntyi internet sellaisena kuin me sen tänä päivänä tunnemme. (Tähtinen 2005, 19 – 20.)

Internet on kuitenkin vain yksi iso esimerkki järjestelmäintegraatiosta, mutta sen merkitys on ollut erityisen suuri koko tietekniselle aikakaudelle. Internetin käyttö ja käyttötavat ovat kehittyneet näihin päiviin saakka ja nykyisin on jo mahdollista toteuttaa

muita järjestelmäintegraatioita internetiä hyväksi käyttäen. Internet ei ole kuitenkaan ainoa suuri askel järjestelmäintegraation historiassa. 1960 – 1970-lukujen vaihteessa yleistyi ohjelmistojen paketoiminen, jolla tarkoitetaan ohjelmistojen eriyttäminen laitteistosta. Aikaisemmin ohjelmistot olivat laitteistokohtaisia ja hyvin erilaisia keskenään, mikä vaikeutti integraatioiden toteuttamista. IBM aloitti uuden aikakauden erottamalla ohjelmistot laitteistosta ja myymällä niitä erillisenä pakettina. Tämä mahdollisti sen, että ohjelmistoja sellaisenaan voitiin asentaa erilaisille alustoille, jolloin eri ympäristöissä olevat samat ohjelmistot kykenivät helpommin kommunikoimaan keskenään ja muodostivat näin helpomman ympäristön järjestelmäintegraatioille. (Tähtinen 2005, 20 – 21.)

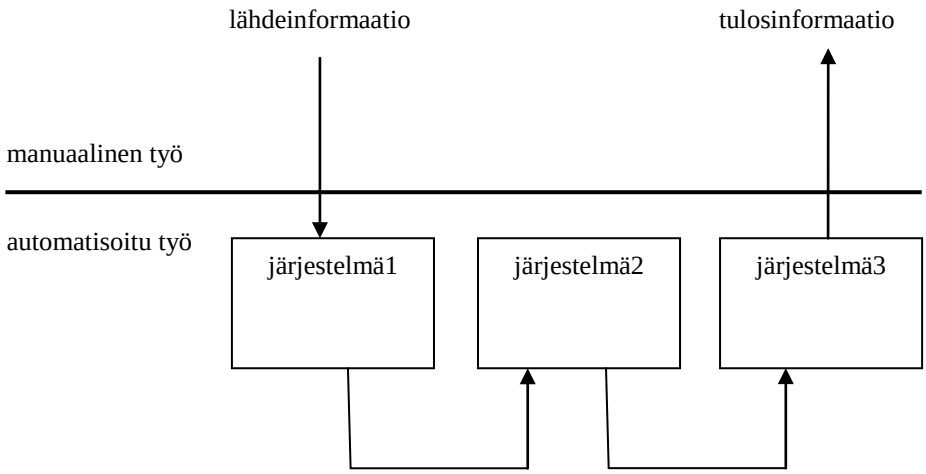
4.2 Järjestelmäintegraation tavoitteet ja hyödyt

Yritysten ohjelmistot ovat hyvin usein hankittu jotain tiettyä tarvetta varten ja ohjelmistojen käyttö noudattaa yrityksen organisaatiorakennetta. Esimerkiksi palkkahallinnolla on ohjelmistoja, joilla käsitellään työntekijöiden palkkoja ja muuta maksuliikennettä. Muilla organisaation yksiköillä tai toiminnoilla ei yleensä ole näitä ohjelmistoja käytössään, eikä siten myöskään niillä saatavia tietoja. Tällainen toimintamalli on kuitenkin hieman vanhanaikainen, koska yritysmaailmassa toimintaa ohjaavat prosessit ja nämä toimintaprosessit leikkaavat usein organisaation vanhojen yksikkörajojen yli ja tietoa on saatava yksiköstä toiseen. Järjestelmäintegraatioiden yksi keskeisimmistä tavoitteista onkin luoda automatisoitu tiedonvälitys yrityksen sisäisten järjestelmien välille, minkä tarkoituksena on tukea toimintaprosessien toimintaa. (Haavisto 2004, Lahti 2003, 4; Tähtinen 2005, 22.)



KUVA 4. Tiedonsiirto manuaalisessa järjestelmässä

Toinen tavoite järjestelmäintegraatioita suunniteltaessa on raha eli säästöjen saaminen. Järjestelmiä yhdistämällä ja automatisoimalla tiedonsiirtoa (kuva 5) saavutetaan selkää hyötyä verrattuna siihen, että tietoa käsiteltäisiin jokaisessa erillisessä järjestelmässä erikseen manuaalisesti (kuva 4). Integroidut järjestelmät tuottavat liiketoimintaprosesseille tarvittavaa tietoa nopeasti ja koostetusti, jolloin aikaa ei kulu tietojen hakemiseen eri järjestelmistä, vaan voidaan keskittyä yrityksen perusliiketoimintaan. Automatisointi vähentää manuaalista järjestelmien käyttöä ja vähentää näin myös inhimillisten virheiden määrää, mikä myös lisää kustannussäästöjä. (Haavisto 2004, 22 – 31.)



KUVA 5. Tiedonsiirto integroidussa ja automatisoidussa järjestelmässä

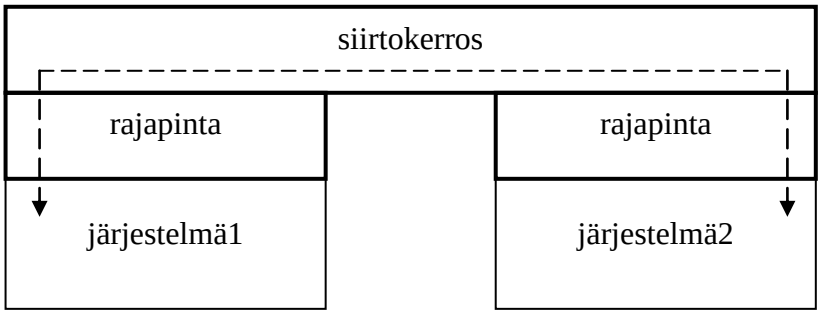
Järjestelmien integrointi luo yritykselle myös mahdollisuuden selkeyttää käsitteittään ja keskittää tiedonhallintaa. Tietojärjestelmäintegroinnissa on tiedettävä, mitä mikäkin käsite eri järjestelmissä tarkoittaa ja näin yrityksen eri yksiköissä saadaan virtaviivaistettua terminologiaa ja selkeytettyä toimintaa reaalimaailman toimintoja mallinnettaessa järjestelmään. Kun järjestelmät integroidaan keskitettyyn ratkaisuun, voidaan eri järjestelmiä hallita helpommin. Ohjattaessa tietoa keskitetyn ratkaisun kautta ohjelmien suorien kahdenvälisten yhteyksien sijaan, saadaan monimutkaisia rajapintaratkaisuja yksinkertaistettua ja esimerkiksi uuden järjestelmän liittäminen vaatii vain yhden rajapinnan ohjelmointia keskitettyyn järjestelmään, jolloin se on automaattisesti muiden järjestelmien käytössä. Järjestelmäintegraatio erityisesti keskitetyn ratkaisun kautta lisää ohjelmistojen käyttöikää, koska ohjelmistot voivat tukea toistensa toiminnallisuutta. Yhden ominaisuuden puuttuminen ei pakota koko ohjelmiston vaihtamiseen, koska jokin toinen ohjelma voi toteuttaa sen. (Haavisto 2004, Tähtinen 2005, 22 – 31.)

Yhteenvetona voidaan todeta, että järjestelmäintegroinnin tarkoituksena on tehostaa yrityksen toimintaa tukemalla sen projekteja, lisäämällä joustavuutta, parantamalla raportointia ja auttamalla ohjelmien hallinnoinnissa. (Kaario & Peltola 2008, 63 – 64; Tähtinen 2005, 48.)

4.3 Järjestelmäintegraation toimintaperiaate

Käytännössä järjestelmäintegraatio (kuva 6) koostuu kolmesta osasta: informaation siirrosta järjestelmästä toiseen, siirretyn tiedon muuntamisesta ja sekä siirron että muunnon valvonnasta ja raportoinnista. Nämä kolme osaa toteuttaa väliohjelmisto (middleware), joka toimii järjestelmien välissä kiinnittyen niiden tiedonsiirtorajapintoihin. (Lahti 2003, 5; Tähtinen 2005, 48.)

Tietojärjestelmien rajapinnat ovat informaation siirrossa välttämättömiä. Mikäli järjestelmissä ei ole olemassa jonkinlaista rajapintaa, johon väliohjelmisto kiinnittyy, ei kyseistä järjestelmää voida integroida muihin järjestelmiin helposti. Rajapinnan tehtävä on mahdollistaa ulkoisen järjestelmän luku- tai kirjoitusoperaatiot integroitavaan järjestelmään. Nykyisin lähes kaikista ohjelmistoista löytyy jokin rajapinta, jonka kautta järjestelmä saadaan integroitua. Rajapinta voi yksinkertaisimmillaan olla siirtotiedosto, jota muut järjestelmät voivat käyttää, sovelluksen ohjelmointirajapinta tai jopa sen käyttöliittymä. Usein käytetään tietokantarajapintaa ODBC:n tai JDBC:n kaltaisen väliohjelman avulla, jolloin tietokantojen välillä siirretään tietoa SQL:ää käyttäen. Myös ohjelmointirajapintaa käyttävät API-sovellukset ovat yleisiä väliohjelmistoja. (Lahti 2003, 5, 14 – 17, 21 – 27; Tähtinen 2005, 49 – 50.)



KUVA 6. Kahden järjestelmän yksinkertainen integrointi

Rajapinnan lisäksi tiedon siirtämiseen järjestelmien välillä tarvitaan jokin fyysinen siirtokerros, mitä pitkin tieto kulkee. Siirtotie voi olla esimerkiksi lähiverkko, internet

tai vaikka tekstiviesti. Tietoa voi siirtää myös perinteisesti magneettisilla tai optisilla tallennusvälineillä, jos siirron nopeudella ei ole merkitystä. Tärkeintä on kuitenkin huolehtia siirtokerroksen tietoturvallisuudesta, oli siirtomedia mikä tahansa. (Tähtinen 2005, 49 – 51.)

Tietoa siirrettäessä on myös usein muunnettava sen muotoa. Koska integroitavilla järjestelmillä on omat tapansa käsitellä tietoa, niiden lähettämä data on todennäköisesti yhteensopimatonta tai ainakin erilaista. Jotta tietoa saadaan liikkumaan järjestelmien välillä, väliohjelmisto muokkaa tietoa siten, että se kelpaa vastaanottavalle järjestelmälle. Yksi tapa toteuttaa tiedon muunto on käyttää yleisesti hyväksyttyä standardia kuvaamaan siirrettävää tietoa. Tällöin puhutaan metadatasta, joka on käytännössä tietoa tiedosta. Metadatassa kerrotaan, että tieto lähetään esimerkiksi tekstitiedostona niin, että se on muodossa sukunimi, etunimi, puhelinnumero. Tällöin lähtevä tieto muunnetaan metadatassa määritettyyn muotoon, tallennetaan tiedostoksi ja vastaanottava järjestelmä lukee metadatan mukaisesti tiedot tiedostosta ja muuntaa sen omaan muotoonsa. Nykyisin paljon käytetty väline tiedonsiirrossa on XML (eXtensible Markup Language)-standardiin perustuvat kuvauskielet, joissa siirtotiedostoon itseensä upotetaan niin siirrettävä tieto kuin sen muotokin. Tällainen siirtotiedosto voisi näyttää esimerkiksi tällaiselta:

```
<?xml version=”1.0” encoding=”ISO-8859-1”>
<!-- henkilön tiedot -->
<yhteystiedot>
        <nimi>
                <!-- nimitiedot -->
                <etunimi>Ville</etunimi>
                <sukunimi>Liukkonen</sukunimi>
        </nimi>
        <puhelin>
                <!-- puhelintiedot -->
                <lanka>015 336688></lanka>
                <gsm>045 12345678</gsm>
        </puhelin>
</yhteystiedot>
```

Tietomuunnoksella ei pyritä kuitenkaan vain järjestelmien väliseen ymmärrykseen datasta, vaan muunnoksella pyritään joissain tapauksissa helpottamaan käyttäjän toimintaa. Järjestelmien tavat esittää omia tietojaan ovat usein erilaisia. Käyttäjä ei välttämättä ymmärrä integroidusta järjestelmästä tuotua informaatiota, mikäli se esitetään hänelle muodossa, joka ei ole tuttu omasta järjestelmästä. Muunnoksella pyritään esittämään haettu tieto niin, että se näyttää tutulle ja on ymmärrettävää. Parhaiten käyttäjää palvellaan, kun integroitujen järjestelmien tieto yhdistetään ja muunnetaan entistä paremmin asiaa kuvaavaksi raportiksi, jossa tietoa on enemmän saatavilla. Tämä voi tapahtua esimerkiksi niin, että haetaan tieto integroidusta kohdejärjestelmästä, haetaan tieto omasta lähdejärjestelmästä, yhdistetään tiedot, muokataan niille uusi esitysmuoto ja tallennetaan omaan lähdejärjestelmään uuteen paikkaan. (Lahti 2003, 29 - 30, 33 – 34; Tähtinen 2005, 54 – 58, 81 – 83.)

4.4 Integroinnin kääntöpuoli

Järjestelmiä integroitaessa niiden arkkitehtuurisuunnittelussa vaaditaan paljon aikaa, vaivaa ja osaamista. Huonosti suunniteltu järjestelmä voi hyvin äkkiä kuormittaa yrityksen tietohallintoa liikaa ja loppujen lopuksi haluttuun toiminnan tehostukseen ja säästöihin ei päästä. Ennen integroinnin toteuttamista on selvitettävä, mitä järjestelmiltä vaaditaan ja mitä se tuo tullessaan. (Tähtinen 2005, 22 – 27, 103.)

Automatisointi on integroiduissa järjestelmissä toimintojen selkäranka. Ilman automatisointia ei yleensä vapaudu haluttua ihmisresurssiakaan järjestelmien ylläpidosta. On siis varmistettava, että automatisoinnille on oltava laitteet ja sen tekemiseen tietotaito, mikä voi nostaa projektin hintaa. Hyvällä suunnittelulla automatisointi vapauttaa henkilöstöä muuhun työhön ja lisää järjestelmien huoltovapautta. Huonosti toteutettuna väärin toimiva automatisointi voi johtaa katastrofiin ja vaikuttaa koko yrityksen liiketoimintaan negatiivisesti. (Tähtinen 2005, 104 – 105.)

Järjestelmiä integroitaessa kokonaisuus myös monimutkaistuu. Monimutkaisuutta voidaan helpottaa keskittämällä integroinnin tiettyyn keskusjärjestelmään ja vähentää näin rajapintojen määrää, mutta vikasietoisuus on silti varmistettava hyvin. Kun järjestelmät eivät ole enää erillisiä, vaan yhteydessä toisiinsa, voi virhe yhdessä järjestelmässä sotkea myös muita järjestelmiä. Keskitetyssä järjestelmässä keskusjärjestelmä on avainasemassa. Sen toiminta ei saa vaarantua missään olosuhteessa, koska se vai-

kuttaa kaikkiin järjestelmiin, joihin se on integroitu. Vikasietoisuuden varmistaminen voi myös vaikuttaa projektin hintaan korottavasti. (Tähtinen 2005, 105 – 107.)

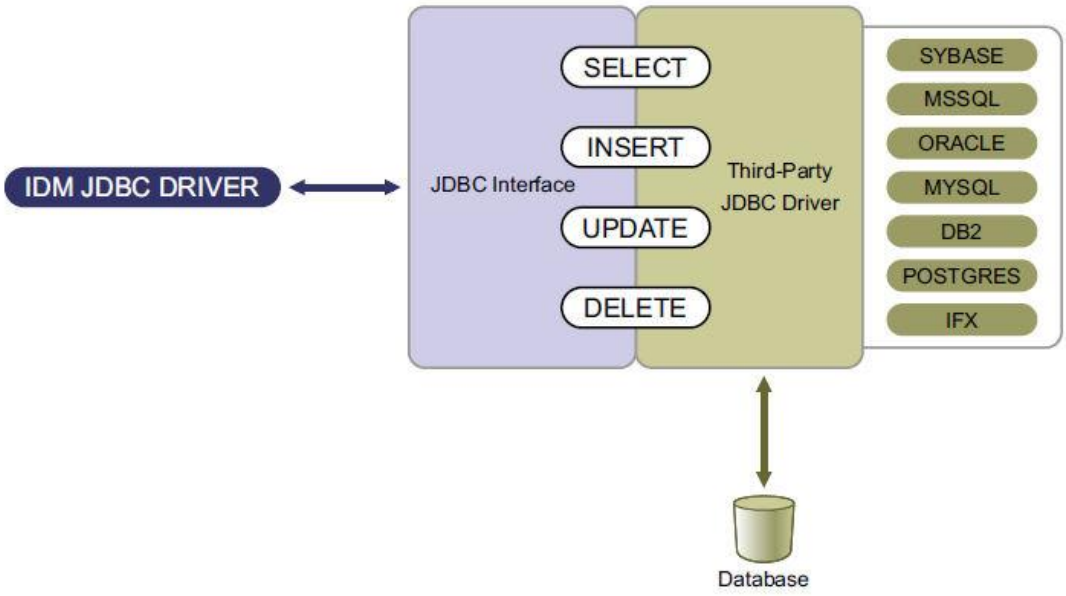
Valvonta ja tietoturvallisuus ovat integroiduissa järjestelmissä normaalia enemmän asioita, joista on pidettävä huolta. Järjestelmien liikuttaessa tietoja ohjelmistojen välillä jopa toiselle puolelle maailmaa esimerkiksi internetin välityksellä, on pystyttävä huolehtimaan, että tietoturvallisuuden osa-alueet eivät vaarannu. Koko integroitua kokonaisuutta on voitava valvoa, jotta mitään yllättävää ei pääse tapahtumaan ilman, että kukaan huomaa. Mikäli integroinnin yhteydessä on toteutettu myös yhteinen käyttäjänhallinta, jolloin samoilla tunnuksilla voidaan kirjautua useisiin järjestelmiin, tietoturvallisuus ja valvonta on toteutettava erityisen hyvällä suunnittelulla. (Tähtinen 2005, 108, 110 – 112.)

Integrointi tuo mukanaan enemmän vastuuta monella osa-alueella, joista ei välttämättä ole tarvinnut paljoa huolehtia järjestelmien ollessa itsenäisiä ja toisistaan riippumattomia. Tämä on hyvä huolehtia integrointiprojektia mietittäessä.

4.5 Novellin Identity Managerin integrointi Oracleen

Opinnäytetyöni on osa kokonaisuutta, jossa Novellin Identity Manager integroidaan Oracle-tietokantaan, jotta opiskelijahallintajärjestelmän tietoja voidaan käyttää käyttäjähallinnassa. Tämän vuoksi kuvaan lyhyesti ja esimerkintyylisesti yhden Novellin käyttämän tavan järjestelmien integrointiin.

Novellin integrointiratkaisu perustuu JDBC:en (Java DataBase Connectivity), joka on java-ohjelmointikielellä toteuttu rajapintatyökalu, jolla voidaan käyttää haluttuja tietokantoja. Novellin ratkaisussa kaksi JDBC-ajuria kommunikoivat keskenään muodostaen integroinnin (kuva 7). Novellin oma IDM JDBC-ajuri, joka sijaitsee Identity Manager-järjestelmässä, ottaa yhteyttä kohdejärjestelmän JDBC-ajuriin. Kohdejärjestelmän JDBC-ajurin on tehnyt tietokannan valmistaja eli tässä tapauksessa Oracle ja liittänyt sen osaksi Oracle-tietokannan rajapintaan, jotta se voidaan integroida muihin järjestelmiin. (Novell 2008, 13 – 14.)



KUVA 7. Identity Managerin ja tietokannan JDBC–ajurin yhteistoiminta (Novell 2008, 14)

Novellin JDBC -ajurin tiedonvälityksen kielenä toimii XML:ään perustuva XDS, joka on Identity Managerin rajapinnan tukema tiedonsiirtomuoto. Tästä XDS-muodosta IDM:n JDBC-ajuri osaa muodostaa SQL-standardin muotoisia lauseita, joilla voidaan käskyttää kohdetietokantaa. Identity Managerista tiedon siirtyessä integroituun tietokantaan se muunnetaan XDS:n kautta SQL-lauseiksi, jotka välitetään tietokantavalmistajan JDBC-ajurille, joka puolestaan käskyttää ne tietokantaan. Integroidusta tietokannasta taas havaitut muutokset välitetään käänteisessä järjestyksessä identiteetinhallintaan. (Novell 2008, 15 – 26.)

5 ESISELVITYS

Tietoteknisissä projekteissa on hyvä seurata yleistä projektimallia, jossa projekti jaetaan eri vaiheisiin. Projektit ovat monesti hankalia hahmottaa, koska teknisten asioiden lisäksi on saatava selvyys taustalla olevista tiedosta ja toiminnoista. Vaiheistamalla projekti saadaan selkeyttä projektin etenemiseen ja tehdään selkeitä päätöksiä jokaisessa vaiheessa, mikä auttaa seuraavan vaiheen toteuttamisessa. Projekteille ja erityisesti IT-projekteille tehtyjä vaihemalleja on useita. (Litke 2004, 37 – 41; Murch 2002, 57 – 60.) Itse kuitenkin päädyin jakamaan projektin esiselvitykseen, vaatimusmäärittelyyn, suunnitteluun, toteutukseen, testaukseen, käyttöönottoon ja lopputuloksen arviointiin.

Usein projektien vaihemalleissa käytetään termiä analyysivaihe, mutta itse pidän sanaa esiselvitys kuvaavampana. Esiselvityksessä on tarkoitus kartoittaa projektiympäristö ja selvittää perusteet käsiteltävistä asioista, järjestelmistä ja termeistä – ei vain analysoida. Kun on saatu selvitettyä kokonaisuus, päätetään mitkä ovat projektin välitavoitteet ja kuinka niihin päästään. Esiselvityksessä on myös huomioitava millaiseen laatuun halutaan ja miten. (Litke 2004, 80 – 82.)

5.1 Käsitteiden määrittely ja sisällön selvittäminen

Aloittaessani ASIO-opiskelijahallintajärjestelmän liittämistä identiteetin hallintaan ensimmäiseksi oli siis tehtävä esiselvitys. Vaikka työssäni toiminkin ASIOn teknisenä pääkäyttäjänä, ei minulla ole käytännön tietoa opiskelijahallintajärjestelmän varsinaisesta tietosisällöstä ja sen merkityksestä. Onkin tärkeää ymmärtää, että vaikka olisikin käytössään varsinainen data järjestelmästä, ei sitä voi käyttää järkevästi, jos ei ymmärrä sen merkitystä omassa ympäristössään (Kaario & Peltola 2008, 82 – 88.) Esimerkiksi tietokannassa oleva luokkatunnus T240KN sisältää opiskelijaryhmän tunnistetiedon lisäksi myös paljon muuta informaatiota, joita käytetään järjestelmässä eri paikoissa. Tunnistetiedon lisäksi tunnuksesta on nähtävissä koulutuksen sijainti ja koulutusohjelma (3 ensimmäistä merkkiä), opiskelijaryhmän aloitusvuosi ja lukukausi (4. ja 5. merkki) sekä tieto siitä, onko koulutus normaalia nuorisopuolen koulutusta vai aikuiskoulutusta. Saadakseni esiselvityksessä tarpeeksi tietoa ASIOssa olevasta tiedosta, konsultoin ASIOn toiminnoista vastaavaa pääkäyttäjää.

Esiselvityksessä ei kuitenkaan riitä, että on ymmärrys siitä, mitä lähdejärjestelmällä on tarjottavana ja kuinka se toimii, vaan on myös ymmärrettävä kohdejärjestelmää. Saadakseni lisää tietoa siitä, mitä IDM-järjestelmään tarvitaan ja missä muodossa, kävin keskustelua identiteetinhallintaa rakentavan konsultin kanssa. Varmistaakseni toimivan kokonaiskuvan pidimme yhteisen palaverin ASIOn pääkäyttäjien ja identiteetinhallintakonsultin kanssa, jolloin pystyimme selvittämään mahdolliset väärinymmärrykset ja selvittämään loput epäselvät asiat. Vaikka välillä esiselvitys saikin vaatimusmäärittelyn piirteitä, niin tärkeintä tässä vaiheessa oli kuitenkin selvittää, mitä eri termit tarkoittavat ja mitkä ovat niiden vastineet kohde- ja lähdejärjestelmissä.

Käsitemäärittelyn lopputuloksena syntyi taulukon 1 muotoinen taulukko (liite 1), jossa näkyy opiskelijahallintajärjestelmän termi, metahakemiston vastine termille sekä tarkempi selitys, mikäli se ei termistä yksiselitteisesti selviä.

TAULUKKO 1. Esimerkki käsitetaulukosta

Asio	**Metahakemisto**	**Selite**
Opiskelijatyyppi	MAMK: student type	A, E, N, V tai C (N = normaali)
Opiskelijanumero	MAMK: student id	Voi olla useita arvoja
Tilakoodi	MAMK: student status	Aktiivi, normi + 1v, harjoittelija, jatkoaika, keskeyttänyt, eronnut, poissaoleva, tilap, valmistunut, yli vuoden takainen opiskelija

Esiselvityksen tavoitteena oli myös kartoittaa, mitä opiskelijoita opiskelijahallintajärjestelmä kattaa, koska kaikki ammattikorkeakoulun opiskelija eivät ole ASIOssa tai eivät käytä sitä aktiivisesti opinnoissaan, ja mitkä opiskelijaryhmät haluttiin identiteetinhallinnan piiriin. Integraatio IDM-järjestelmään ei siis voinut käsittää kaikkia opiskelijoita, eikä se kaikille ryhmille ollut tarpeenkaan. Oli myös selvitettävä, vaikuttaako järjestelmäintegraatio opiskelijatietoja ylläpitävien tahojen käytäntöihin. Esiselvityksessä päädyttiin siihen, että IDM-integraatio tehtiin vain niille opiskelijaryhmille, jotka opiskelevat tutkintoon johtavassa koulutusohjelmassa ja käyttävät koulun sähköpostia. Nämä ryhmät käyttävät eniten palveluita, jotka ovat yhteydessä identiteetinhallintajärjestelmään ja voivat saada sieltä käyttäjätunnistuksen.

5.2 Tekninen ympäristö

Opiskelijahallintajärjestelmänä Mikkelin ammattikorkeakoululla on Oracle-tietokantaan (versio 9.2.0.8) ja PL/SQL ohjelmointikieleen perustuva ja internet-selaimella käytettävä ASIO. Palvelinalustana on toiminut Microsoftin Windows Server 2003, mutta internetpalvelimena ei ole Microsoftin Internet Information Server, vaan Oraclen mukana tuleva Apache. Syynä Apachen käyttöön on sen integroitu toiminta Oraclen oman PL/SQL-ohjelmoinnin kanssa, jolla opiskelijahallintajärjestelmä on toteutettu. ASIOn käyttökanava on pääsääntöisesti internetkäyttöliittymä, jonka kautta niin opiskelijat kuin ylläpitohenkilötkin käyttävät järjestelmää ja tietokannan tietoja.

Opiskelijahallintajärjestelmään on tehty myös joitain ODBC-rajapintaan perustuvia liittymiä, joilla haetaan yksittäisiä tietoja internetsivuille. Tällaisia ovat esimerkiksi opintojaksokuvaukset, jotka haetaan suoraan SQL-lausein opiskelijahallintajärjestelmästä osaksi sähköistä opinto-opasta. Tietojen haku ASIOsta ulkopuolisiin järjestelmiin ei siis ole aivan testaamaton toiminto ja näin ollen myös uskoin integraation identiteetinhallintajärjestelmään olevan mahdollinen ja toteutettavissa melko lyhyelläkin aikavälillä.

Opiskelijahallintajärjestelmän on toimittanut Asio-Data Oy, joka huolehtii järjestelmän päivittämisestä ja sen asianmukaisesta toiminnasta. Tämä kattaa järjestelmään kuuluvat PL/SQL-ohjelmat sekä tietovarastoina toimivat Oraclen tietokantataulut. Oraclen ylläpito, toiminnasta huolehtiminen sekä varmuuskopiointi kuuluvat kuitenkin käyttäjätaholle eli tässä tapauksessa Mikkelin ammattikorkeakoululle ja sitä kautta minulle. Käytettävissäni on siten Oraclen kaikki toiminnallisuudet pääkäyttäjän oikeuksin. Tämä mahdollistaa tarvittaessa monimutkaistenkin toimintojen, jotka hyödyntävät opiskelijahallintajärjestelmän tietorakenteita, rakentamisen tietokannan sisään sen omilla välineillä.

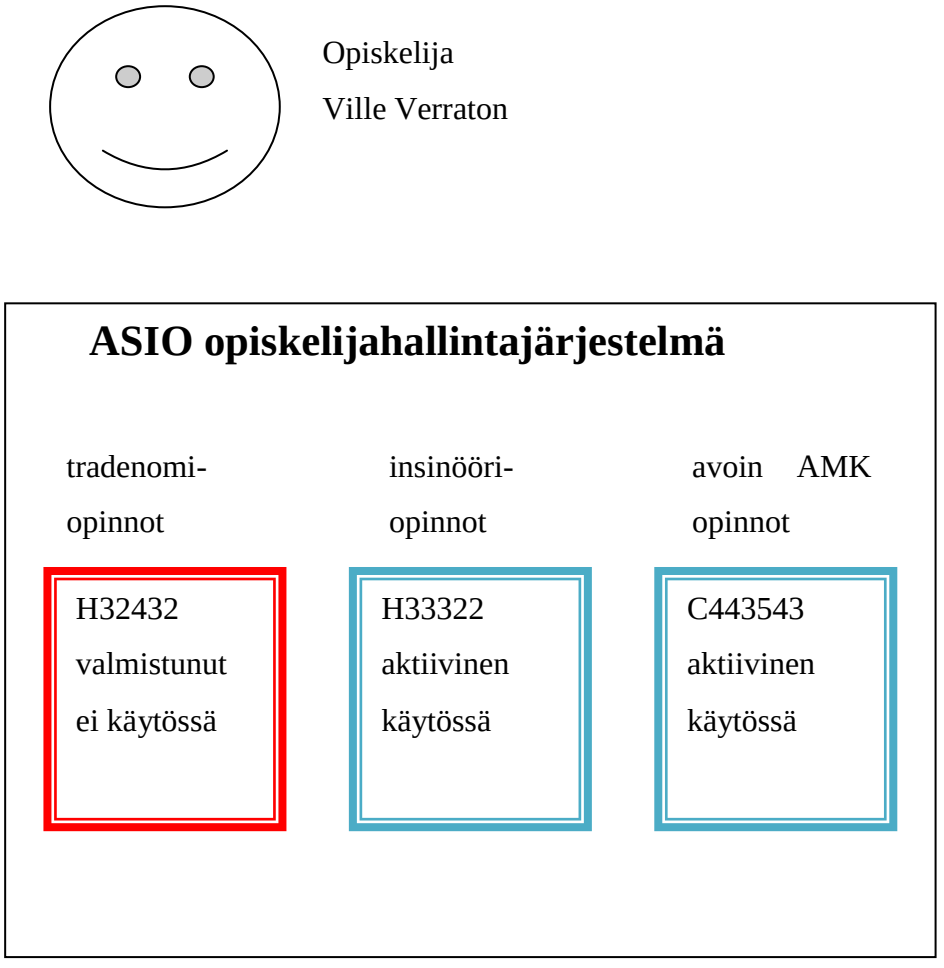
5.3 ASIOn perustoimintaidea

Jokaisella opiskelijalla on joko yksi tai useampi profiili eli niin sanottu opiskelijakortti opiskelijahallintajärjestelmä ASIOssa. Opiskelijan aloittaessa opintonsa Mikkelin

ammattikorkeakoulussa, hänelle luodaan uusi opiskelijakortti joko manuaalisesti tai automaattisesti opetusministeriön lähettämän siirtotiedoston tietojen perusteella.

Uusi kortti luodaan aina, vaikka opiskelija opiskelisikin ammattikorkeakoulussa. Esimerkiksi vuonna 2008 tradenomipaikan ja vuonna 2009 insinööripaikan vastaanottanut opiskelija on kahdella erillisellä opiskelijakortilla opiskelijahallintajärjestelmässä. Näin hänellä on myös kahdet erilliset tunnukset ja kaksi erillistä roolia (kuva 8). ASIO ei yhdistä näiden kahden roolin tietoja millään tavalla, vaikka kortteihin on toki mahdollista tehdä merkintöjä toisesta opintokortista.

Opiskelijahallintajärjestelmässä on oma opiskelijanumerointinsa, joka toimii myös opiskelijan käyttäjätunnuksena. Henkilötunnusta ei käytetä opiskelija tunnistamiseen, mikä lisää opiskelijan henkilösuojaa. ASIO-järjestelmän kaikki toiminnot perustuvat tähän opiskelijanumeroon, jonka perusteella opiskelijan tiedot kiinnitetään arvosanoihin, opintojaksoilmoittautumisiin ja muihin toimintoihin.



KUVA 8. ASIOn roolirakenne

Kun opiskelija lopettaa tiettyyn rooliin sidotut opintonsa eli valmistuu esimerkiksi insinööriksi, hänen tunnuksensa suljetaan. Tämä tarkoittaa, että valmistumisen yhteydessä hänen tilakoodikseen merkataan V, jolloin tunnukset eivät ole enää voimassa. Tunnukset poistetaan myös silloin, kun opiskelija keskeyttää joko pysyvästi tai määräaikaisesti, jolloin tilakoodiksi tulee K.

Opiskelijan opiskelijakorttia ei ainakaan toistaiseksi koskaan poisteta opiskelijahallintajärjestelmästä, mutta ne pyritään arkistoimaan järjestelmän sisäisesti 10 vuoden kuluessa. Tämän vuoksi opiskelijoilla voi olla ASIOssa niin aktiivisia kuin suljettujakin tunnuksia.

6 VAATIMUSMÄÄRITTELY

Vaatimusmäärittelyvaiheessa on tarkoitus selvittää ne asiat, jotka luovat rajat ja vaatimukset projektille. Määrittelyssä selvitetään ne sisällölliset, tekniset ja laadulliset vaatimukset, jotka projektissa on toteuduttava. Lisäksi vaatimusmäärittelyssä selvitetään rajoitukset, joita projektilla on. Rajoitukset voivat olla esimerkiksi taloudellisia, aikataulullisia kuin lakiinkin perustuvia.

Omassa projektissani keskityin vaatimusmäärittelyn osalta erityisesti sisällölliseen vaatimusmäärittelyyn, koska se oli luonteeltaan haasteellisin. Tekniset vaatimukset ja rajoitukset olivat selkeämmät, koska käytännössä vaihtoehtoisia järjestelmiä ei ollut, vaan käytettiin olemassa olevia ratkaisuja, joiden toiminta rajoituksineen oli jo tiedossa. Lain mukaan toimiminen on mielestäni itsestään selvää ja se on opiskelijahallintajärjestelmässä huomioitava koko ajan, koska opiskelijahallintajärjestelmä ASIO on virallinen henkilötietorekisteri. Pyrin hyvään tiedonhallintatapaan, minkä vuoksi huomioin vaatimusmäärittelyssä tietoturvallisuuden, joka hyvin monilta vaatimuksiltaan perustuu lain kirjaimeen ja erityisesti lain henkeen.

6.1 Sisällölliset vaatimukset

Identiteetinhallintajärjestelmä perustuu henkilön yksiselitteisesti identifioivaan tunnukseen. Aivan ensimmäisenä oli selvitettävä, mikä tämä tällainen tunnus tulisi olemaan. Opiskelijahallintajärjestelmässä itsessään on olemassa asiotunnukseksi kutsuttu merkkijono, jolla opiskelijahallintajärjestelmä tunnistaa käyttäjät. ASIO kuitenkin tunnistaa opiskelijat tarkemmalla tasolla kuin identiteetinhallinnassa oli tavoitteena. ASIOssa opiskelupaikka on merkityksellinen ja siksi opiskelijalla on jokaista opiskelijapaikkaansa kohti oma asiotunnus. Tämän vuoksi opiskelijalla voi olla useampia tunnuksia opiskelijanhallintajärjestelmässä. Identiteetinhallintaan oli kuitenkin tarkoitus tunnistaa opiskelija henkilönä.

Koska asiotunnuksen käyttö ei ollut mahdollinen, vaihtoehdoiksi jäi täysin uuden tunnuksen generointi tai henkilötunnuksen käyttö. Uuden tunnuksen luonti olisi täytynyt toteuttaa opiskelijahallintajärjestelmässä ja olisi perustunut henkilötunnuksen käyttöön, jolloin sen käyttö ei ollut järkevää. Oli siis yksinkertaisinta välittää ASIOsta

henkilötunnus, jonka avulla identiteetinhallintajärjestelmä sitten niputtaa opiskelijan eri opiskelijaroolit yhdeksi identiteetiksi.

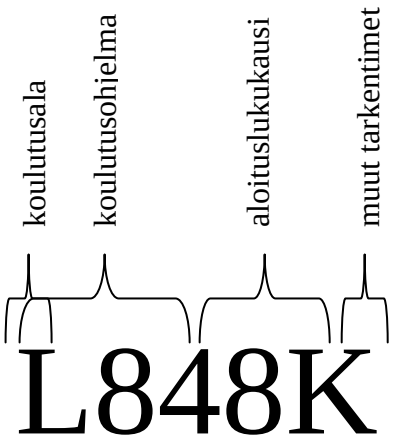
Kun identiteetinhallinnalla on tieto opiskelijoista, sen pitäisi myös tietää, miten kauan tunnuksia pidetään voimassa. Opiskelijanhallintajärjestelmän pitäisi siis kertoa, onko opiskelija vielä opiskelija vai ei. Identiteetinhallinta hallitsee opiskelijan pääsyä ASIOta lukuun ottamatta kaikkiin verkkopalveluihin sähköpostia myöten, niin oli erityisen tärkeää, että IDM:ssä olevan tunnuksen elinkaari oli sidottu opiskelijanhallinnassa olevaan aina ajan tasalla olevaan tietoon. Opiskelijapalvelut merkitsevät ASIOssa opiskelijan tietoihin, onko hänen opiskeluroolinsa tila aktiivinen, keskeyttänyt vai valmistunut. Linjaksi päätettiin, että aktiivisella opiskelijalla on pääsy järjestelmiin, mutta keskeyttäneellä, vaikka vain väliaikaisesti, ja valmistuneella ei ole IDM:n hallinnoimiin järjestelmiin kirjautumisoikeutta. Oli myös hyvä huomata, että opiskelijalla voi olla aktiivinen ja valmistunut rooli, jolloin hänellä täytyy olla pääsy verkkopalveluihin, vaikka hän toiselta rooliltaan onkin tunnuksettomuuden piirissä. Opiskelijahallintajärjestelmän täytyi siis lähettää opiskelijan tilan muutokset IDM:ään jokaisen roolin osalta.

Esiselvityksen aikana ja jo ennen projektin aloittamista käytiin runsaasti keskustelua siitä, missä IDM:n salasanaa hallitaan. Koska ASIOssa asiotunnuksen salasana on sellaisessa muodossa, että se on välitettävissä eteenpäin, ASIOn salasanan käyttö vaikutti opiskelijalle helpoimmalta ratkaisulta. Opiskelijahallintajärjestelmä on kuitenkin opiskelijan tärkein yksittäinen verkkopalvelu ja kaikkien muiden järjestelmien hallinnointi sen käyttäjätunnus ja salasana -parilla on helpointa muistaa. Näin ollen ASIO välittää opiskelijaroolikohtaisen asiotunnuksen lisäksi myös tätä tunnusta vastaavan salasanan. Tämä mahdollistaa sen, että opiskelija voi kirjautua järjestelmiin opiskelijaroolinsa mukaisilla tunnuksilla, jolloin hän saa esimerkiksi Moodlesta sen roolin mukaisen sisällön. Kirjautuessaan avoimen AMK:n asiotunnuksella opiskelija saa vain ja ainoastaan avoimen opintojakson kurssimateriaalin, eikä tutkintoon johtavan päiväopintojensa materiaaleja.

Toiminnallisuuden kannalta henkilötunnus, asiotunnus, salasana ja opintojen tila olivat riittävät tiedot ASIOsta välitettäväksi. Ylläpidon käytettävyyden kannalta ASIOsta päätettiin kuitenkin myös välittää opiskelijan perustietoja, kuten nimitiedot ja yhteystiedot. Identiteetinhallinnan ylläpidolla ei ole pääsyä tai pääsy on rajoitettua ASIOn

opiskelijarekisteriin, joten nimi- ja yhteystiedot olivat tarpeellisia mahdollisten vikatilanteiden varalle ja niiden selvittämiseksi.

Opiskelijaan liittyvät tiedot eivät kuitenkaan olleet ainoat tiedot, joita opiskelijahallintajärjestelmästä tarvittiin identiteetinhallintaan. Esiselvityksen palavereissa nousi esiin tarve saada myös tietoa siitä, millaisiin ryhmiin opiskelija organisaatiossa kuuluu. Ammattikorkeakoulussa ei ole varsinaisia opiskelijaluokkia, mutta opiskelija kuuluu kuitenkin tiettyyn opetusryhmään, joka on puolestaan osa tietyn koulutusalan laitoksen koulutusohjelmaa. Esimerkiksi minä kuulun opetusryhmään L848KA, joka on Sähköisen asioinnin ja arkistoinnin koulutusohjelmassa (L84). ASIOn opetusryhmän tunnus eli luokkatunnus (kuva 9) sisältää tiedon siitä, mihin koulutusohjelmaan se kuuluu ja sen määrittävät tunnuksen kolme ensimmäistä merkkiä. Loput merkit taas erottavat opetusryhmät toisistaan. Ensimmäinen merkki (L) itsekseen määrittää, mihin koulutusalaan opetusryhmä ja koulutusohjelma kuuluvat. Jotta identiteetinhallintajärjestelmään saadaan muodostettua riittävä organisaatiorakenne, opiskelijanhallintajärjestelmästä täytyy välittää tieto koulutusaloista, koulutusohjelmista ja opetusryhmistä. Organisaatiorakennetta hyödynnetään muun muassa sähköpostiryhmien muodostuksessa.



KUVA 9. Opetusryhmätunnuksen eli luokkatunnuksen rakenne

Organisaatiorakenteen siirtäminen opiskelijahallintajärjestelmästä vaikutti myös siihen, että opiskelijan tiedoista täytyi ottaa mukaan opiskelijan luokkatunnus. Näin opiskelija saadaan kytkettyä identiteetinhallinnan organisaatiorakenteessa omaan koh-

taansa. Samalla siirretään myös aloituslukukausimerkintä. Aloituslukukausi on erotettavissa luokkatunnuksesta, mutta koska se erillisenä tietona on olemassa, niin tieto päätettiin ottaa mukaan mahdollista tulevaa käyttöä varten, koska sen avulla voidaan opiskelijat tarvittaessa jakaa vuoden vaihteessa aloittaviin ja syksyllä aloittaviin.

Opiskelijan henkilötietojen ja luokkatietojen lisäksi opiskelijahallintajärjestelmästä tarvittavia tietoja katsottiin olevan myös opiskelijan tapahtumatiedoista löytyvä läsnä-oloilmoittautumiseen liittyvä tieto. Tämä tieto pitää sisällään sen, mille lukukausille opiskelija on ilmoittautunut poissa- ja läsnäolevaksi. Tapahtumatiedoista saatava opiskelijan roolin tila on tarkempi kuin henkilötiedoissa oleva aktiivinen, keskeyttänyt tai valmistunut merkintä, sillä se sisältää esimerkiksi syyn keskeyttämiseen tai määrittää onko keskeytys vain väliaikainen.

6.2 Tekniset vaatimukset ja rajoitukset

Teknisissä vaatimuksissa lähdettiin siitä, että tietojen siirto identiteetinhallinnan rajapintaan tulee olla mahdollista nykyisillä järjestelmillä. Käytännössä tämä tarkoitti siis Oracle 9i-tietokannan ja Windows 2003-palvelinympäristön tarjoamien välineiden käyttämistä. Opiskelijahallintajärjestelmää käytetään lähes ympäri vuorokauden ja hetkittäin, esimerkiksi opintojaksoilmoittautumisten aikana, sen käyttökuorma on suuri. Käyttökuormalla tarkoitan samanaikaisia käyttäjiä ja palvelimelle tulevia pyyntöjä, jotka ovat lähinnä monimutkaisia tietokantakyselyjä. Teknisenä pääkäyttäjänä en halunnut lähteä kuormittamaan järjestelmää uusilla ohjelmilla, jotka olisivat voineet heikentää palvelimen suoritustehoa, vaikuttaa opiskelijahallintajärjestelmän tietoturvaan ja laskea käytettävyyttä.

Identiteetinhallinnan kannalta tietoja olisi voinut välittää opiskelijahallintajärjestelmästä reaaliaikaisesti aina tiedon muuttuessa tai sitten eräajona esimerkiksi kerran päivässä aina muuttuneiden tietojen osalta. Kokonaisuuden kannalta toimivinta kuitenkin on, että tiedon muuttuessa vaikutus on välitön. Esimerkiksi salasanan vaihtuessa on käytettävyyden kannalta parempi, että uusi salasana on voimassa heti, eikä vasta seuraavana päivänä. Tämän vuoksi tekniseksi vaatimukseksi otettiin tietovirran reaaliaikaisuus.

Tärkein tekninen vaatimus oli kuitenkin se, että opiskelijahallintajärjestelmä ei saa kuormittua kohtuuttomasti, eikä tietojen siirtyminen identiteetinhallintaan saa vaikuttaa ASIOn käytettävyyteen ja toimintaan. Opiskelijoiden tietojen tulee pysyä muuttumattomina, tapahtuu identiteetin hallinnassa mitä tahansa IDM:ssä tai niissä opiskelijanhallintapalvelun toiminnoissa, jotka tietoa valuttavat IDM:ään. Jo esiselvityksen aikana päätettiin, että salasanoja voi muuttaa vain ASIOsta IDM:ään päin. Jos salasanaa joskus voi muuttaa jossain muussa järjestelmässä tai identiteetinhallinnassa, niin se ei vaikuta opiskelijanhallintajärjestelmän salasanaan.

6.3 Tietoturvallisuuden vaatimukset

Vaatimusmäärittelyssä, kuten koko projektissa, oli huolehdittava, että tietoturvallisuus säilyi ja sen kustannuksella ei tehty kompromisseja. Tein itselleni tietoturvallisuuden osatekijöiden mukaisen jaon ja hahmottelin jokaiseen kohtaan miten juuri kyseinen tietoturvallisuuden osa tuli käytännössä huomioida integraatioprojektissa.

Luottamuksellisuus

ASIO järjestelmässä tämä täytyi huomioida siten, että projektin aikana opiskelijahallintajärjestelmän tietoja ei paljastu ulkopuolisille. Vaikka integroinnissa IDM-järjestelmään organisaation ulkopuolisella konsultilla täytyikin olla pääsy tiettyihin tietoihin, niin suurimpaan osaan ASIOn tietoja hänellä ei ollut mitään syytä päästä ja täytyi selvittää mihin pääsy on taattava. Esiselvityksessä oli myös luottamuksellisuuden vuoksi selvitettävä, että opiskelijahallintajärjestelmästä ei siirretä turhia tietoja identiteetinhallintajärjestelmään, vaan ainoastaan sellainen tieto, joka on oleellista järjestelmää käyttöönotettaessa ja mahdollisesti tulevaisuudessa.

Identiteetinhallintajärjestelmää hallinnoidaan Mikkelin ammattikorkeakoulun omissa tiloissa oman henkilökunnan toimesta. Voikin kysyä, miksi on niin tärkeää, että ylimääräistä tietoa ei välitetä IDM:ään. Opiskelijahallintajärjestelmä on henkilörekisteri, joka sisältää runsaasti tietoja opiskelijoista ja heidän opinnoistaan. Hyvään tiedonhallintatapaan kuuluu, että järjestelmän ylläpitäjä jo suunnitellessa varautuu vikatilanteisiin ja ettei rekisteritietoa joudu vääriin käsiin. Samaa edellyttää myös henkilötietolaki. Siksi on tärkeää, että IDM-järjestelmään ei viedä ylimääräisiä tietoja, esimerkiksi

arvosanatietoja, jotka vikatilanteessa voisivat vaarantua ja jotka eivät ole olennaisia identiteetin hallintajärjestelmän toiminnan kannalta.

Käytettävyys

Esiselvityksen teknisissä vaatimuksissa kiinnitettiin erityistä huomiota juuri käytettävyyteen. Oli varmistettava, että uudet identiteetin hallintaan liittyvät toiminnallisuudet eivät vaikuta opiskelijanhallintajärjestelmän käyttöön hidastaen sitä tai jopa estäen toiminnan.

Identiteetinhallintajärjestelmän näkökulmasta täytyi myös huomioida käytettävyys. Tiedon opiskelijanhallintajärjestelmästä täytyy siirtyä IDM:ään koko ajan ilman suurta viivettä ja tiedon pitää olla aina samassa, etukäteen sovitussa ja suunnitellussa, muodossa.

Eheys

Opiskelijahallintajärjestelmän tiedot ovat ammattikorkeakoulun toiminnalle kriittisiä ja niiden on oltava oikein ja tallessa erityisesti sekä arvosanojen että suoritusten osalta. Koska identiteetinhallintajärjestelmä ei näitä tietoja tarvitse, oli pidettävä huolta, että IDM:llä ei ole mitään yhteyttä arvosanoihin tai suorituksiin, jotta ne eivät voisi tahattomastikaan muuttua. Tiedon eheyden turvaamiseksi esiselvityksen teknisissä vaatimuksissa rajattiin tietojen liikkuminen vain ASIOsta IDM:n suuntaan. Näin estetään tietojen korruptoituminen automaattisten toimintojen osalta. Tahallisten ja ihmisten tekemien muutosten osalta projektissa käytettävillä tunnuksilla ei saanut olla muutos-oikeuksia ASIOn järjestelmiin. Vaikka käyttöoikeudet joutuisivatkin vääriin käsiin, niillä ei voisi vaikuttaa tiedon eheyteen opiskelijahallintajärjestelmässä.

Tiedon eheyden säilyminen oli tietenkin tärkeää myös identiteetinhallintaan ja inhimillisten virheiden välttämiseksi tarkoituksena oli suunnitella täysin automatisoitu järjestelmä, joka tiedon muuttuessa automaattisesti välittää tiedon identiteetinhallintaan ja mahdollisesti tarkastaa tiedon eheyttä.

Kiistämättömyys

ASIO – IDM-integraatiossa kiistämättömyys täytyi huomioida niin, että ASIOn tietoja ei voida muuttaa mistään muualta kuin opiskelijahallintajärjestelmän hallintatoimintojen kautta kirjautumalla, joka myös tekee lokitietoa. Integraatio identiteetinhallintajärjestelmään ei saa heikentää kiistämättömyyttä eli tässä tapauksessa ASIOn tiedot eivät saa muuttua IDM:n suunnasta. Myös automaattisessa tiedonsiirrossa opiskelijahallintajärjestelmästä identiteetinhallintajärjestelmään oli tapahduttava kirjautuminen ja rajata yhteys vain IDM järjestelmälle, jolloin mikään muu, mahdollisesti ulkoinen, järjestelmä ei voi saada tietoja.

Kiistämättömyyden toteuttaminen vaati myös sitä, että vaikka suoria hallintayhteyksiä tietokantaan oli sallittava esimerkiksi konsulteille, niin oikeudet oli rajattava niin, että kirjautuminen tapahtuu oikeuksilla, jotka eivät vaaranna tietojen kiistämättömyyttä tai eheyttä. Käytännössä tämä tarkoittaa kirjautumista vain lukuoikeuksin.

Tietojen käytön valvonta auttaa myös virhetilanteiden selvittämisessä IDM:n ja ASIOn välillä, kun tiedot eivät syystä tai toisesta siirry. Kun lokitietojen käsittely on oikein tehty, on mahdollista selvittää mitä tietoa IDM on koettanut saada ja miksi se ei ole onnistunut. Tämä on kuitenkin toteutettu tässä projektissa IDM:n puolelle, joten en käsittele sitä tarkemmin.

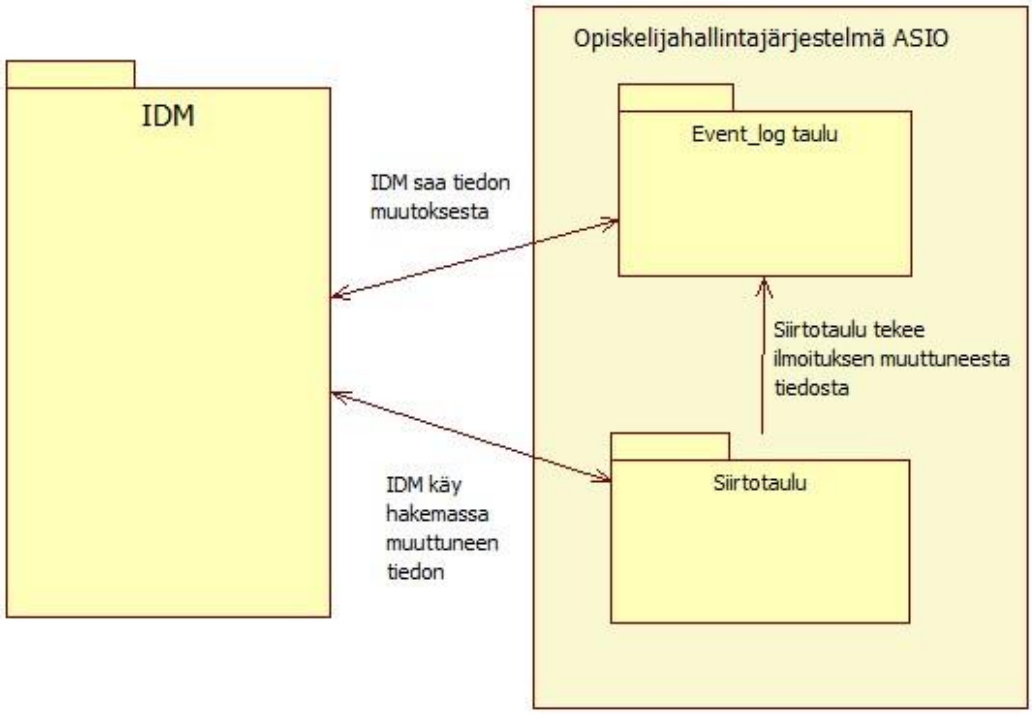
Pääsynvalvonta

Mikkelin ammattikorkeakoululla on palvelintilat, joissa on oma kulunvalvonta. Näihin tiloihin on pääsy vain palvelimien ylläpitäjillä ja tarvittaessa huoltohenkilöstöllä valvotusti. Fyysisesti niin identiteetinhallinta- kuin opiskelijanhallintapalvelimelle pääseminen on tietoturvallisuuden osalta kunnossa. Myös korkeakoulun verkko on sisäverkon osalta hyvin suojattu. Verkon kautta laitteille pääsy on rajattu muutamaan ihmiseen. Koska projektin aikana ei vaikutettu pääsyyn fyysisille laitteille, en perehdy siihen tarkemmin.

7 SUUNNITTELU

Lyhyen toteutusajanvuoksi suunnittelulle ei jäänyt paljoakaan aikaa. Jouduin viikossa suunnittelemaan, millä tavoin tietojen siirto opiskelijahallintajärjestelmästä identiteetinhallintajärjestelmään tehtäisiin. Koska aikaa ei ollut, päätin selvittää, onko vastaavia ratkaisuja olemassa jo valmiiksi. IDM-järjestelmänä Mikkelin ammattikorkeakoulussa oli käytössä Novell Identity Manager, joten keskityin etsimään Novellin suosittelemia ratkaisuja. Novellin ratkaisujen päättelin olevan toimivia myös siksi, että opiskelijahallintajärjestelmämme on Oracle–tietokannan päällä ja Novell on tehnyt paljon yhteistyötä Oraclen kanssa.

Hyvinkin nopeasti löysin IDM-konsulttimme avustuksella yksityiskohtaisen ohjeen Novellin ”Driver for JDBC Implementation Guide” oppaasta. Opas on tarkoitettu ohjeeksi identiteetinhallintajärjestelmän ylläpitäjälle, joka ohjelmoi ajurin hakemaan tietoja perusrekistereistä. Ajurin ohje ei antanut yksityiskohtaisia ohjeita tietokannan sisäiselle tiedon siirtämiselle, mutta se antoi yleiskuvan rajapinnan toiminnalle, sen vaatimuksille ja tarkat määritykset rajapintaan tulevalle lokitaululle, mikä on tiedonsiirron avainkohta. Tämän vuoksi puhunkin event log –tiedonsiirrosta (kuva 10).



KUVA 10. Event log -toimintaperiaate

Event log -tiedonsiirron perusidea toimii siten, että perusrekisterissä sijaitsee tietokantataulu, niin sanottu event_log taulu (kuva 11), johon välittyvät tiedot tietokannassa tapahtuvista muutoksista. Tarvittaessa vielä niin, että vain niistä muutoksista välitetään tieto, joita IDM tarvitsee. Kun IDM:n ajuri havaitsee, että event_log tauluun tulee muutos eli tietoa perusrekisterin tietojen muutoksesta, se käy lukemassa tauluun tulleen tiedon. Tämän tiedon perusteella ajuri saa tiedon, missä perusrekisterin tiedossa on tapahtunut muutos ja millainen se muutos on. Muutos voi olla tietueen lisäys, tietueen poisto tai tietueen kenttien arvojen muutos. Tiedon saatuaan ajuri menee muuttuneeseen perus-rekisterin tietoon, kopioi muutoksen identiteetinhallintajärjestelmään ja poistaa tämän jälkeen event_log taulusta tiedon muutoksesta tai merkitsee tietoihin virhemerkinnän, mikäli muuttuneen tiedon välittäminen IDM:ään ei onnistunut.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
RECORD_ID	NUMBER(10,0)	No	(null)	1	(null)
TABLE_KEY	VARCHAR2(100 BYTE)	No	(null)	2	(null)
STATUS	VARCHAR2(1 BYTE)	Yes	(null)	3	(null)
EVENT_TYPE	NUMBER(1,0)	No	(null)	4	(null)
EVENT_TIME	DATE	No	(null)	5	(null)
PERPETRATOR	VARCHAR2(50 BYTE)	Yes	(null)	6	(null)
TABLE_NAME	VARCHAR2(50 BYTE)	No	(null)	7	(null)
COLUMN_NAME	VARCHAR2(50 BYTE)	Yes	(null)	8	(null)
OLD_VALUE	VARCHAR2(100 BYTE)	Yes	(null)	9	(null)
NEW_VALUE	VARCHAR2(100 BYTE)	Yes	(null)	10	(null)

KUVA 11. Event_log taulu

Näkymät vai taulut?

Varsinainen rajapintatoiminto ei siis vaatinut suurempaa suunnittelua, koska sen toteuttamiseen oli jo olemassa tarkat ohjeet, myös event_log taulun rakentamiseen, ja toimiva ajuri. Tehtäväkseni jäi kuitenkin miettiä, miten tieto muuttuneesta tiedosta välitetään event log -tiedonsiirtoon. Vaihtoehtoja oli käytännössä kaksi. Relaatiotietokannoissa on olemassa näkymätoiminto (view), jonka avulla voidaan yhteen näkymään koostaa tietoja yhdestä tai useammasta tietokannan taulusta. Näkymä on sitä käyttäville palveluille kuin yksittäinen taulu, mutta oikeasti se on vain virtuaalinen kooste halutuista tiedoista (Hovi 2005, 14, 54; Rosenzweig 2003, 443 – 444). Näkymän hyvä puoli on sen virtuaalisuus. Tietoja ei kopioida useampaan paikkaan, vaan ne sijaitsevat aina alkuperäisessä tietokantataulussaan. Näkymä kuitenkin tekee aina sitä käytettäessä taustalla SQL-hakuja tietokantatauluihin, joista se koostetaan, mikä vie

tietokannan resursseja. Vaihtoehtona näkymälle on luoda tavallinen tietokantataulu, mihin tarvittavat tiedot kopioidaan ja event log –ajuri käy lukemassa tiedot sieltä. Tavallisen taulun hyvä puoli ovat sen irrallisuus alkuperäisistä tiedoista. Niitä voidaan käyttää ilman, että alkuperäiset tietokannan tiedot kuormittuvat. Huonoina puolina on tiedon kopiointi, mikä lisää aina hieman korruptoitumisen riskiä tietoja kopioitaessa. Lisäksi tiedot vievät tuplasti enemmän levytilaa.

Minulla oli käytössäni runsaasti levytilaa ja opiskelijahallinnalle erityisen tärkeää on se, että sen käyttö ei häiriinny. Osaksi tämän vuoksi päätin valita event log toiminnan tietolähteeksi erillisen tiedonsiirtotaulun, enkä suoraa näkymää alkuperäisiin tietokantatauluihin. Opiskelijajärjestelmässä tehdään runsaasti massa-ajoja, jolloin muutetaan paljon tietoja opiskelijatietoihin, joten en uskonut useita SQL-lauseita tekevän näkymän olevan käytettävyyden kannalta paras ratkaisu, koska se kuormittaisi varsinaisia tietolähteitä juuri samaan aikaan, kun niihin tehdään muutoksia. Haluan kuitenkin korostaa, että nykyaikaiset palvelinympäristöt ovat niin tehokkaita, että käytännössä molempien vaihtoehtojen ero kuormituksessa on häviävän pieni. Päätökseeni vaikutti vahvasti se, että kun IDM:ään välitettävä tieto on erillisessä taulussa, sen avulla voidaan helpommin testata yhteyden toimivuutta, kun tiedonsiirto alkaa alkuperäisten tietojen kopioista eikä alkuperäisistä. Lisäksi alkuperäisissä tiedoissa oli tietoa sellaisessa muodossa, että niiden käyttäminen IDM:ssä olisi ollut vaikeaa tai vaatinut ohjelmallista käsittelyä. Kopioitaessa tietoa alkuperäisistä tietokantatauluista siirtotauluun pystyin muokkaamaan tiedon valmiiksi helpommin käytettävään muotoon.

Tietokannan taulut ja herätteet

Suunnitteluvaiheen aluksi oli suunniteltava tietokannan taulut, joihin tieto opiskelijahallintajärjestelmä ASIOsta kopioidaan ja mistä IDM:n ajuri käy hakemassa muutokset ja uudet tiedot. Vaatimusmäärittelyn aikana oli selvitetty, että opiskelijahallintajärjestelmästä tarvitaan henkilötiedot, tapahtumatiedot ja organisaation koulutusrakenteen kuvaamiseen tarvittavat tiedot. Nämä kaikki sijaitsevat ASIOssa eri tauluissa, mutta oli loogisinta luoda jokaiselle IDMn tarvitsemalle tietoryhmälle oma taulunsa, joihin tieto ASIOn tauluista kopioidaan tarvittaessa tietoa yhdistäen.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
OPISKELIJANUMERO	VARCHAR2(1000 BYTE)	No	(null)	1	(null)
OPISKELIJATYYPPI	VARCHAR2(1000 BYTE)	Yes	(null)	2	(null)
SUKUNIMI	VARCHAR2(1000 BYTE)	Yes	(null)	3	(null)
ETUNIMI	VARCHAR2(1000 BYTE)	Yes	(null)	4	(null)
KUTSUMANIMI	VARCHAR2(1000 BYTE)	Yes	(null)	5	(null)
TILAKOODI	VARCHAR2(1000 BYTE)	Yes	(null)	6	(null)
HETU	VARCHAR2(1000 BYTE)	Yes	(null)	7	(null)
OPISKELUOSOITE	VARCHAR2(1000 BYTE)	Yes	(null)	8	(null)
OPISKELUPOSTIOSOITE	VARCHAR2(1000 BYTE)	Yes	(null)	9	(null)
PUHELIN1	VARCHAR2(1000 BYTE)	Yes	(null)	10	(null)
PUHELIN2	VARCHAR2(1000 BYTE)	Yes	(null)	11	(null)
KOTIKUNTAKOODI	VARCHAR2(1000 BYTE)	Yes	(null)	12	(null)
KOTIKUNTANIMI	VARCHAR2(1000 BYTE)	Yes	(null)	13	(null)
KOTILAANI	VARCHAR2(1000 BYTE)	Yes	(null)	14	(null)
OMA_EMAIL	VARCHAR2(1000 BYTE)	Yes	(null)	15	(null)
KOULUN_EMAIL	VARCHAR2(1000 BYTE)	Yes	(null)	16	(null)
KOTIOSOITE	VARCHAR2(1000 BYTE)	Yes	(null)	17	(null)
KOTIPOSTIOSOITE	VARCHAR2(1000 BYTE)	Yes	(null)	18	(null)
KOTIPUHELIN	VARCHAR2(1000 BYTE)	Yes	(null)	19	(null)
SALASANA	VARCHAR2(1000 BYTE)	Yes	(null)	20	(null)
MAA	VARCHAR2(1000 BYTE)	Yes	(null)	21	(null)
KIELI	VARCHAR2(1000 BYTE)	Yes	(null)	22	(null)
KANSALAISUUS	VARCHAR2(1000 BYTE)	Yes	(null)	23	(null)
KOULUSIVISTYSKIELI	VARCHAR2(1000 BYTE)	Yes	(null)	24	(null)
WEP_WAP	VARCHAR2(1000 BYTE)	Yes	(null)	25	(null)
POHJAKOULUTUS	VARCHAR2(1000 BYTE)	Yes	(null)	26	(null)
TUTKINNON_LAAJUUS	VARCHAR2(1000 BYTE)	Yes	(null)	27	(null)
LUPA_KOULUTUSTIEDOTUS	VARCHAR2(1000 BYTE)	Yes	(null)	28	(null)
LUPA_MARKKINOINTI	VARCHAR2(1000 BYTE)	Yes	(null)	29	(null)
LUPA_INTERNET	VARCHAR2(1000 BYTE)	Yes	(null)	30	(null)
LUPA_MATKAPUHELIN	VARCHAR2(1000 BYTE)	Yes	(null)	31	(null)
TUTKINTOKOODI	VARCHAR2(1000 BYTE)	Yes	(null)	32	(null)
TUTKINTO	VARCHAR2(1000 BYTE)	Yes	(null)	33	(null)
SUUNTAUTUMISKOODI	VARCHAR2(1000 BYTE)	Yes	(null)	34	(null)
SUUNTAUTUMINEN	VARCHAR2(1000 BYTE)	Yes	(null)	35	(null)
KOULUTUSOHJELMAKOODI	VARCHAR2(1000 BYTE)	Yes	(null)	36	(null)
KOULUTUSOHJELMA	VARCHAR2(1000 BYTE)	Yes	(null)	37	(null)
RYHMATUNNUS	VARCHAR2(1000 BYTE)	Yes	(null)	38	(null)
RYHMA	VARCHAR2(1000 BYTE)	Yes	(null)	39	(null)
ALOITUSVUOSI	VARCHAR2(1000 BYTE)	Yes	(null)	40	(null)
KOPIOT_TARKISTETTU	VARCHAR2(1 BYTE)	Yes	(null)	41	(null)
ALKUPERAISET_TARKISTETTU	VARCHAR2(1 BYTE)	Yes	(null)	42	(null)
EI_SAHKOPOSTIA	VARCHAR2(1 BYTE)	Yes	(null)	43	(null)
EI_VERKKOTUNNUSTA	VARCHAR2(1 BYTE)	Yes	(null)	44	(null)

KUVA 12. Siirtotaulu

Henkilötiedot kopioidaan siirtotauluun (kuva 12) ja kaikki tieto tuodaan ASIOn määrittelemänä tekstimuodossa, koska tietomuodolla ei ollut merkitystä identiteetinhallinnan kannalta.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
KOODI	VARCHAR2(10 BYTE)	No	(null)	1	(null)
KOODI2	VARCHAR2(10 BYTE)	Yes	(null)	2	(null)
SELITE	VARCHAR2(500 BYTE)	Yes	(null)	3	(null)
SELITE_ENG	VARCHAR2(500 BYTE)	Yes	(null)	4	(null)
PARENT	VARCHAR2(10 BYTE)	Yes	(null)	5	(null)

KUVA 13. Rakennetaulu

Koulutusrakenteen tiedot sijaitsevat ASIOssa kolmessa erillisessä taulussa, mutta koska niistä otettava koulutusrakenteen tieto on muodoltaan samanlainen, ne yhdistetään kolmesta taulusta yhteen rakennetauluun (kuva 13).

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
ID	VARCHAR2(10 BYTE)	No	(null)	1	(null)
OPISKELIJANUMERO	VARCHAR2(10 BYTE)	No	(null)	2	(null)
TAPAHTUMATUNNUS	VARCHAR2(10 BYTE)	Yes	(null)	3	(null)
TAPAHTUMASELITE	VARCHAR2(500 BYTE)	Yes	(null)	4	(null)
PVM	VARCHAR2(10 BYTE)	Yes	(null)	5	(null)
MUUTOSPVM	VARCHAR2(10 BYTE)	Yes	(null)	6	(null)
TAPAHTUMATILA	VARCHAR2(10 BYTE)	Yes	(null)	7	(null)

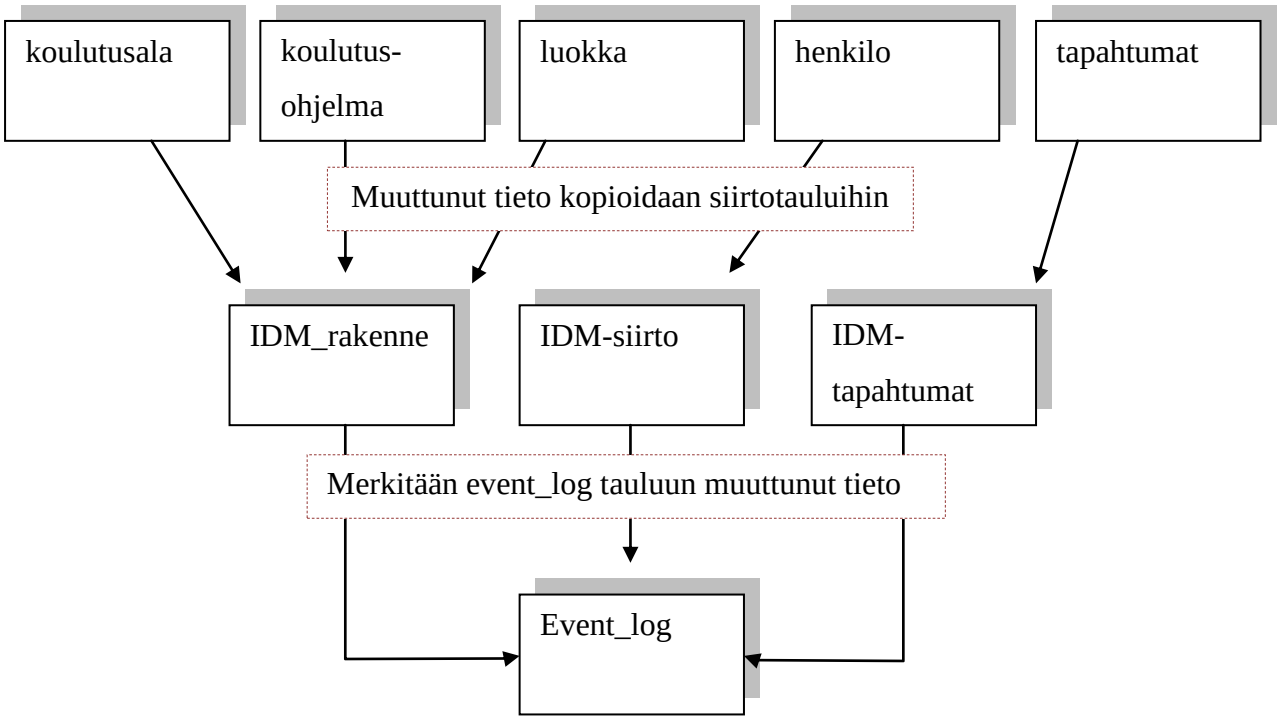
KUVA 14. Tapahtumataulu

Tapahtumat kopioidaan halutuin osin tapahtumatauluun (kuva 14), jolla on relaatio siirtotauluun, joka on tapahtumataulun äitiobjekti. Halusin varmistaa tällä yhteydellä sen, että henkilötietoja poistettaessa siirtotaulusta, poistuvat myös opiskelijan tapahtumamerkinnät tapahtumataulusta ilman erillistä poistokomentoa.

Seuraava vaihe oli suunnitella tiedon automaattinen kopiointi (kuva 15) opiskelijahallinnan tietokantatauluista identiteetinhallinnan siirtotauluihin tiedon muuttuessa. Koska en halunnut lähteä tekemään muutoksia opiskelijahallintajärjestelmään ja sen toiminnallisuuteen, jäi ainoaksi vaihtoehdoksi tietokannan omien toimintojen käyttäminen. Kaikista moderneista tietokannoista löytyy toiminto nimeltä trigger eli heräte. Heräte on toiminto, joka seuraa ennalta päätettyjä tietokannan toimintoja ja reagoi niihin tietyin tavoin (Hovi 2005, 14; Rosenzweig 2003, 441 – 442). Suunnitteluosuuden suurimmaksi työksi muodostuikin herätteiden suunnittelu – miten niiden pitäisi reagoida ja mihin.

Opiskelijahallintajärjestelmässä oli viisi tietokantataulua, joiden tietoja täytyi kopioida: henkilötiedot, tapahtumatiedot, koulutusalatieto, koulutusohjelmatiedot, ryhmätie-

to. Näihin kaikkiin täytyi suunnitella heräte, joka valvoo niihin kohdistuvia insert, update ja delete SQL-lauseita. Kun henkilötietoja lisätään, muutetaan tai poistetaan, täytyy vastaava toiminto suorittaa myös identiteetinhallinnan siirtotauluun. Opetusryhmä-, koulutusohjelma- tai koulutusalatietoja lisättäessä, muutettaessa ja poistettaessa sama muutos välitetään IDM:n rakenne -tauluun. Tapahtumataulun muutoksetkaan eivät olleet niin suoraviivainen juttu. Koska henkilöllä ja tapahtumilla on relaatio, niin tapahtumia lisättäessä on varmistettava, että henkilötiedot löytyvät siirtotaulusta ja jos niitä ei ole, on ne kopioitava ensin. Henkilötietojen lisäyksen yhteydessä oli kopioitava myös tapahtumataulun tiedot opiskelijahallintajärjestelmästä, vaikka ne eivät pakollisia siinä vaiheessa olekaan. Näin vältytään virhetilanteelta, missä lapsiobjektilla (tapahtumataulun tiedot) ei olekaan äitiobjektia (siirtotaulun henkilötietoja).



KUVA 15. Automaattinen tiedon kopioituminen

Opiskelijahallintajärjestelmä ASIO pitää sisällään paljon tietoa sellaisistakin opiskelijoista, joita ei ollut esiselvityksen perusteella tarkoitus ottaa mukaan identiteetinhallintaan. Herätteitä suunniteltaessa oli tämä otettava huomioon ja mietittävä, miten rajataan pois ne muutokset, jotka tapahtuvat niissä tiedoissa, jotka eivät ole identiteetinhallinnan piirissä. Minun täytyi löytää opiskelijoiden tiedoista jokin arvo, jonka perusteella oli mahdollista rajoittaa herätteen toimintaa. Opiskelijahallintajärjestelmässä jokaisella opiskelijan roolilla on olemassa opiskelijatyyppi ja tämän tiedon arvo osoit-

tautui sopivaksi rajoittimeksi. Kun opiskelijan opiskelijatyyppi on W (virtuaaliopiskelija), Y (yrityshautomon opiskelija) tai Z (kesälukukauden opiskelija), niin heräte ohittaa näiden opiskelijoidentiedoissa tapahtuvat muutokset. Kaikkien muiden opiskelijatyyppien opiskelijat tarvitsivat tavalla tai toisella identiteetinhallinnan tarjoamia palveluja.

Kun rajat herätteiden toiminnalle oli suunniteltu, oli seuraavaksi mietittävä, mitä heräte tekee. Jo aikaisemmin suunnitteluvaiheessa oli laadittu taulut ja päädytty siihen, että tieto kopioidaan ASIOn perustietotauluista erillisiin tauluihin, joista IDM noutaa tiedon. Perusidealtaan herätteen tuli toimia niin, että kun muutos halutuissa tiedoissa tapahtuu, niin heräte välittää muutostiedon uusin tauluihin. Pääosin muutokset toteutuvat yhden suhde yhteen eli esimerkiksi muutos sukunimitiedoissa päivittää samaa kenttää IDM:n siirtotauluissa. Herätteiden suunnittelussa oli kuitenkin huomioitava yksi poikkeus. Opiskelijahallinnan erilaiset lupakysymykset tallentuvat yhteen määrämittaiseen merkkimuotoiseen kenttään. Kun opiskelija on antanut luvan tietojensa julkaisuun internetissä, hänellä on tietokannassa kyseisessä lupakentässä X-merkki 4. merkin kohdalla. Jos lupaa ei ole, niin kyseisen merkin kohdalla on tyhjä. Lupakenttä saattaa siis näyttää tältä: XX X XX. Tällainen tieto välitettynä IDM:ään ei ole käytännöllinen, joten suunnittelin herätteeseen toiminnallisuuden, joka purkaa nämä lupatiedot omiin kenttiinsä IDM:n käyttämiin tietokantatauluihin. Näin voidaan tarvittaessa lukea helposti tietty lupa tai luvan muutos. Tein laajemmin herätteiden suhteen päätöksen, että mikäli tietoa on mahdollista muuntaa luettavampaan muotoon, niin se muutos tehdään herätteissä, jotka siirtävät tietoa opiskelijarekisteristä IDM:n siirtotauluiin.

Herätteitä oli suunniteltava myös muualle kuin muutoksiin opiskelijahallinnan tiedoissa. Kun tiedot on kopioitu identiteetinhallinnan siirtotauluihin (siirtotaulu, rakennetaulu ja tapahtumataulu), on tieto niihin tulevista muutoksista välitettävä identiteetinhallinnalle. Samalla tavoin kuin aikaisemmissakin herätteissä herätteet reagoivat tauluun kohdistuviin insert, delete ja update lauseisiin, mutta tässä tapauksessa mitään rajoitteita ei tarvinnut suunnitella, koska kaikki tieto on IDM:lle välitettävää. Herätteen tarkoitus ei kuitenkaan ole kopioida tietoa enää mihinkään, vaan informoida identiteetinhallinnan rajapinta-ajuria. Rajapinta-ajuri seuraa event_log taulua, joten toisen vaiheen herätteet syöttävät sinne tietoa muuttuneista tietokantariveistä siirtotauluissa. Suunnittelin siirtotaulujen herätteet niin, että ne kirjoittavat muutoksen yhteydessä

event_log tauluun rivin, missä on tieto siitä, mikä tieto siirtotauluissa on muuttunut, miten se on muuttunut ja mikä uusi ja vanha arvo on. Käytin suunnittelun pohjana Driver for JDBC Implementation Guide -opasta, joka antoi merkintätavoille yleisesti käytössä olevan standardin. Merkintätavasta sovittiin yhdessä IDM:ää rakentavan konsultin kanssa, jotta saatiin suoraan toimiva kommunikointikieli kahden järjestelmän välille. Tarkemmin herätteiden ohjelmointikoodista voi lukea liitteestä 2.

Tietokantaympäristön muutokset

Projektin vaatimuksiin kuului, että opiskelijahallintajärjestelmän toimintoihin ei puututa ja identiteetinhallintajärjestelmän vaatimat muutokset on pidettävä mahdollisimman selkeästi erillään ASIOn toiminnoista eikä ASIOn toiminta saa vaarantua tai heikentyä. Kun identiteetinhallintajärjestelmän vaatimat toiminnot ja muutokset oli suunniteltu, täytyi varautua niiden vaikutuksiin teknisessä ympäristössä - lähinnä tietokantaohjelmistossa.

Tietoturvallisuuden kannalta en halunnut ottaa mitään riskejä käyttäjätunnusten osalta. Identiteetinhallinnalle oli tarjottava jokin käyttäjätunnus, jolla se voi lukea event_log taulua sekä siirtotaulua, rakennetaulua ja tapahtumataulua. Mihinkään muualle käyttöoikeutta ei tarvinnut antaa. Suunnittelin tietokantaan tunnuksen ja ryhmän, jolla oli pääsy identiteetinhallinnan käyttämiin tietokannan tauluihin sekä ajaa niihin kohdistuvia herätteitä. Näin tällä tunnuksella ei edes vahingossa voi tehdä muutoksia ASIOn tietokantatauluihin. Opiskelijahallinnan tauluista tietoa kopioivat herätteet ovat ASIOn käyttäjätunnuksen omistuksessa, joten niitä ei ole mahdollista IDM:n tunnuksilla ajaa tai muuttaa, eikä näin ollen hakea sellaista tietoa, jota ei alun perinkään ole tarkoitusta välittää eteenpäin.

Käytettävyyden on tärkeää, että palvelu ei hidastele ja toimii virheettä. Uusien toimintojen suunnittelun yhteydessä tietokantapalvelimeen oli mietittävä niiden mahdollinen vaikutus käytettävissä oleviin resursseihin ja erityisesti muistin käyttöön. Normaalikäytössä identiteetinhallintajärjestelmän liittämisen opiskelijahallintapalveluun ei pitäisi näkyä suoritustasoa alentavasti, koska herätteet ovat tietokannan omaa toimintaa ja niiden suorittamat toimenpiteet varsin pieniä Oraclen kaltaiselle tietokannalle. On kuitenkin ajankohtia lukuvuoden aikana, jolloin tietokannan käyttökuorma kasvaa merkittävästi. Esimerkiksi uusien opiskelijoiden saapuminen ja opintojaksoilmoittau-

tumisten alkaminen kuormittaa tietokantaa normaalia enemmän. Jotta identiteetinhallinta ei alenna suoritustehoa näiden käyttöpiikkien aikanakaan, päätin suorittaa tietokantapalvelimessa viikon mittaisen analyysiajon ja määritellä tietokannan muistin käytön sen perusteella uudestaan. Tarkoitus oli tehdä analyysiajo sen jälkeen, kun opiskelijahallintajärjestelmä on liitetty identiteetinhallintaan ja se normaalikäytössä. Sillä oletuksella, että järjestelmä ei hyydy täysin heti alkuun, mihin en aikaisemmin mainitsemistani syistä uskonut.

8 TOTEUTUS, ONGELMAT JA KRIITTISET KOHDAT

Suunnitteluvaiheen jälkeen rakentelin tietokantataulut, herätteet ja muut tietokantaan tulevat pienemmät muutokset, esimerkiksi käyttöoikeuksissa, ensin opiskelijahallintajärjestelmämme testiympäristöön. Testiympäristö on pyritty rakentamaan siten, että ympäristö on muuten identtinen varsinaisen tuotantoympäristön kanssa, mutta laitteisto ei ole aivan yhtä tehokasta. Testiympäristöä ei myöskään ole varmennettu laitteiston, ohjelmistojen eikä siellä olevien tietojen osalta. Koska ympäristö oli tietokannaltaan ja siellä olevilta tiedoiltaan oikean opiskelijahallintajärjestelmän kopio, sinne tehdyt identiteetinhallintajärjestelmän osat olivat suoraan toimivia ja siirrettävissä myöhemmin suoraan tuotantokantaan.

Herätteiden ja taulujen luominen testiympäristöön sujui lähes ongelmitta. Ongelmia aiheutti lähinnä se, että en ollut koskaan ohjelmoinut PL/SQL-ohjelmointikielellä mitään, joten jouduin opettelemaan kaiken perusteista lähtien tehdessäni herätteiden toiminnallisuuksia. Käyttöoikeuksien ja taulujen luonti sekä herätteiden ohjelmointi vei pari viikkoa, jonka jälkeen testiympäristössämme oli valmis ”valumisketju”, missä muutos ASIOn henkilö-, luokka- ja tapahtumatiedoissa käynnistivät muutoksen kopioinnin IDM:n siirto-, rakenne- ja tapahtumatauluihin, minkä jälkeen siitä kirjoitettiin merkintä event_log tauluun. Testiympäristön käyttökuorma on käytännössä nolla, koska sitä käyttävät vain pääkäyttäjät (kaksi) omiin testauksiinsa. Koko tiedonvälitysketju ASIOn tietojen muutoksesta event_log tauluun kesti ainoastaan sekunnin osia, joten olin tyytyväinen herätteiden ohjelmointiin.

Kun perustoiminnallisuus tiedon välityksessä testitietokannassa oli kunnossa, päätimme identiteetinhallinnan ylläpidon kanssa liittää järjestelmät toisiinsa ja aloittaa kokonaisvaltaisen testauksen ja selvittää mahdolliset kriittiset kohdat ja niin sanotut pullonkaulat. Käytännössä liittäminen toisiinsa tarkoitti sitä, että identiteetinhallinnan rajapinta-ajurille määritettiin kohteeksi testitietokannan osoite ja ajurin toinen osa asennettiin testipalvelimelle, jolloin ajurin kaksi osaa välittävät tietoa keskenään ja tietokantapalvelimelle asennettu ajuri valvoo muutoksia event_log taulussa. Kun ajurit olivat paikoillaan ja tietoliikenteelle avattiin tarvittavat reiät palomuuriin, opiskelijahallinnan testiympäristö oli liitetty identiteetinhallintaan.

Varsinainen testaaminen aloitettiin pienillä muutoksilla testiympäristössä. Yhden opiskelijan, luokan ja tapahtuman tietoja muutettiin hieman. Samoin testattiin yhden tietueen lisäystä tai poistoa. ASIOn toiminnallisuuksissa oli pieniä virheitä, jotka korjasin. IDM:n päässä tehtiin enemmän pieniä ja isoja muutoksia, koska he pääsivät vasta tässä vaiheessa testaamaan oikeaa välitettyä tietoa. Pienen mittakaavan testauksessa ei havaittu suuria ongelmia, joten testauksessa siirryttiin suuriin määriin. Testikannasta ajettiin ohjelmallisesti kaikkien identiteetinhallinnan piiriin kuuluvien luokkien, opiskelijoiden ja tapahtumien lisäys identiteetinhallinnan tauluihin. Koska järjestelmässä ei ollut muuta rajoitusta tuotaville tiedoille kuin opiskelijatyyppi, opiskelijahallintajärjestelmästä siirtyi yli 18000 riviä opiskelijatietoa ja jokaisella opiskelijalla on tapahtumia muutamasta merkinnästä jopa pariin kymmeneen tapahtumamerkintään. Tällä massa-ajolla halusimme testata erityisesti identiteetinhallinnan kykyä selvitä suuresta määrästä samanaikaista tietoa, mutta samalla sain myös itse arvokasta tietoa tietokannan ja herätteiden toiminnasta. Oraclen tietokanta suoriutui massa-ajosta vaivattomasti. Muutamassa minuutissa kaikki välitetyt tiedot oli siirretty siirtotauluihin ja niistä oli kirjoitettu merkinnät event_log tauluun odottamaan IDM:n käsittelyä. Identiteetinhallintajärjestelmältä tiedon käsittely vei pari tuntia, mikä ei ollut yllättävää, koska siellä tietoa käsitellään monimutkaisemmin ja esimerkiksi luodaan tunnukset vain aktiivisille opiskelijoille tilakoodin perusteella. Massa-ajo tehtiin vielä muutaman kerran uudestaan niin, että IDM:n taulut tyhjennettiin, jolloin käynnistyi massapoisto IDM:ään ja sen jälkeen lisättiin taas kaikki tiedot ASIOsta. Myöhemmin massa-ajoa rajattiin niin, että se toi vain aktiivisten opiskelijoiden tiedot.

Testipalvelimella tehtyjen testausten aikana ei huomattu mitään erityisiä heikkoja kohtia järjestelmässä. Ainoa satunnaisesti tapahtuva virhetilanne syntyi siitä, kun tietoa välittävä rajapinta-ajuri syystä tai toisesta pysähtyi. Tämä vikaantuminen korjattiin sillä, että ajuri sammutetaan ja käynnistetään joka yö. Virhe ei vaikuttanut tietoihin, mutta ajurin ollessa pois päältä, muuttuneet tiedot eivät välittyneet identiteetinhallintaan.

Aikataulu oli erittäin tiukka, sillä identiteetinhallintajärjestelmä haluttiin ottaa opiskelijoiden käyttöön jo alkavan lukukauden alussa, joten päätin aloittaa IDM:n taulujen ja toiminnallisuuksien siirtämisen ASIOn tuotantoympäristöön varsin lyhyellä testauksella. Suurin osa selvitettävistä virheistä ja keskeneräisistä toiminnoista sijaitsi varsinaisessa identiteetinhallintajärjestelmässä, mikä ei estänyt ASIOn pään toimintojen

siirtämistä tuotantoon. Tässä vaiheessa oli selvää, että muutoksia oli kuitenkin tulossa. Identiteetinhallinnan piiriin haluttiin myös avoimen ammattikorkeakoulun opiskelijoita, joiden sijoittamista identiteetinhallintaan oli toivottu ja vastustettu koko projektin ajan. Ongelmia projektin kannalta aiheutti se, että avoimessa ammattikorkeakoulussa ASIOta käytettiin monilta osin väärin. Opiskelijoille ei esimerkiksi annettu valmistumis- tai keskeyttämismerkintöjä ja ASIOssa oleva vanha opiskelijarooli saatettiin muokata opiskelijalle käyttöön uudestaan toisiin opintoihin. Avoimen AMK:n osalta päätettiin tehdä tarvittavia muutoksia IDM:ssä ja välittää heidän tietonsa ASIOsta kuten muutkin, mutta vain viimeisen vuoden ajalta ja sen jälkeen heidän tuli laittaa opiskelijoilleen keskeyttämis- / valmistumismerkinnät opintojen päätyttyä, etteivät tunnukset jäisi aktiiviseksi identiteetinhallintaan.

9 KÄYTTÖÖNOTTO

ASIOn identiteetinhallintaosa otettiin käyttöön vain muutamia päiviä ennen opiskelijoiden tuloa ja mukaan otettiin myös avoimen ammattikorkeakoulun opiskelijat. Tarvittavien taulujen ja herätteiden siirto testiympäristöstä sujui vaivattomasti, sillä ne olivat siirrettävissä valmiina SQL-luontilauseina tietokannasta toiseen. Käyttöoikeudet tein varmuuden vuoksi käsin tuotantoon, vaikka nekin olisivat olleet siirrettävissä samalla tavoin kuin taulut ja herätteet. Lopuksi identiteetinhallinta liitettiin tuotantoon, kuten aikaisemmin testiympäristöön, ja luokka-, opiskelija- ja tapahtumatiedot ajettiin massa-ajoina IDM:n tauluihin.

ASIOn tuotantoympäristö on käyttöresursseiltaan reilusti testiympäristöä tehokkaampi ja massa-ajot eivät tuotantoympäristössäkään tuottaneet ongelmia. Tuotantoympäristössä huomasin kuitenkin hyvin pian vakavan virheen, mikä ei ollut tullut esiin testiympäristössä. Testiympäristössä varsinaista ASIOta ei käytä kukaan ja vaikka siellä muutamia testi-toimintoja testasinkin, ei minulla ollut ajallisesti mahdollista testata kaikkia toimintoja. Vakava virhe löytyi herätteestä, joka kopioi tietoa opiskelijan tapahtumista IDM:n tapahtumatauluun. Tapahtumataulun ja siirtotaulun, joka sisältää opiskelijan tiedot, välillä on relaatio ja tämän vuoksi tapahtumaa ei voi lisätä, ellei siirtotaulusta löydy tapahtuman omistajan opiskelijanumerolle opiskelijaa. Herätteitä suunnitellessani en ollut huomioinut mahdollisuutta, että opintosihteerit saattavat muokata sellaisten opiskelijoiden tapahtumia, joita ei ole siirretty siirtotauluun IDM:ään välitettäväksi, mutta joita ei kuitenkaan ole rajattu opiskelijatyyppinsä mukaan pois. Virheen seurauksena heräte antoi virheen ja virhe esti opintosihteereitä päivittämästä tapahtumatietoja. Kyseinen virhe aiheutti sen, mitä erityisesti tuli projektissa välttää eli opiskelijahallinnan käytön häiriintymisen. Virhe vaati herätteen ohjelmoinnin muuttamista siten, että se aina ensin tarkistaa onko opiskelijan tiedot jo siirtotaulussa, kun opiskelijan tapahtumia muutetaan. Lisäsin saman tarkastuksen myös opiskelijatietojen muutokseen, koska teoriassa oli mahdollista koettaa muuttaa sellaisen opiskelijan tietoja, joka ei ollut vielä siirtynyt identiteetinhallinnan siirtotauluun ja tällöin olisi syntynyt samanlainen virhetilanne. Näitä virhetilanteita olisi voinut syntyä erityisesti avoimen ammattikorkeakoulun opiskelijoiden kohdalla, joiden tietojen vaikutusta järjestelmään en osannut arvioida esiselvitys- ja suunnitteluvaiheessa.

Käyttöönottoa opiskelijoille ei lähdetty porrastamaan, vaan kaikki identiteetinhallinnan piiriin kuuluvat opiskelijat otettiin mukaan samanaikaisesti. Syksyllä 2009 saapuneet opiskelijat saivat kaikki identiteetinhallinnan kautta automatisoidusti luodut tunnukset verkkoon/tietokoneille, oppimisympäristö Moodleen ja sähköpostiin. Vanhoille opiskelijoille ei luotu uusia sähköpostilaatikoita, vaan he saivat pitää entiset laatikkonsa, mutta muuten heidän tunnuksensa ovat identiteetinhallinnan hallitsemat ja ASIOn tunnukseen ja salasanaan perustuvia. Kun käyttöönotto toteutettiin kaikille, se helpotti ohjeistuksen laatimista ja saatiin paljon erilaisia käyttötapauksia ja tilanteita, jotka auttoivat viimeisten virheiden löytämisessä. ASIOn toiminnan osalta ei käyttöönotossa aiemmin mainitsemani virheen lisäksi löytynyt virheitä tai toimintahäiriöitä. Identiteetinhallintajärjestelmässä oli jonkin verran virhetilanteita. Käyttöönoton suurimmat ongelmat syntyivät nopeasta käyttöönotosta, minkä vuoksi henkilökunnalla ei ollut tarpeeksi aikaa sisäistää uusia toimintatapoja ja -malleja tai tiedotus ei kaikkia ehtinyt kunnolla edes tavoittaa.

Muutaman kuukauden päästä ensimmäisestä käyttöönottopäivästä suoritin tietokantapalvelimelle 10 päivää kestävän analyysiajon, joka mittasi tietokannan käyttöastetta ja sen käyttämiä resursseja. Analyysi osoitti, että tietokannan kuormitus ei ollut juurikaan kasvanut identiteetinhallinnan integroinnin myötä, mutta koska muistin käyttö oli muutenkin lähellä annettua maksimia, lisäsin tietokannalle määritettyä muistin osuutta huomattavasti.

10 LOPPUTULOKSEN ARVIOINTI JA KEHITTÄMISIDEAT

Kokonaisuutena opiskelijahallintajärjestelmän integrointi identiteetinhallintajärjestelmään oli lyhyt ja nopeatempoinen projekti. Aikaa oli vähän ja jouduin tekemään suurimman osan suunnittelusta ja toteutuksesta yksin, mikä lisäsi haastetta erityisesti herätteiden osalta, joiden ohjelmointiin en ollut millään tavoin perehtynyt aikaisemmin. Projektia puolestaan helpotti oma vahva osaamiseni Oraclen puolelta eli tietokanta itsessään oli minulle tuttu.

Lopputulokseen olen erittäin tyytyväinen, vaikka matkan aikana virheitä muutama tulikin. Projekti piti aikataulunsa ja tavoitteisiin päästiin, vaikka välillä oli turhankin pitkiä hetkiä, jolloin projektia ei käytännössä johtanut kukaan. Asiantuntijaperusteisessa organisaatiossa on kuitenkin se hyvä puoli, että osaprojektit yleensä etenee, vaikka kokonaiskuva välillä olisikin hieman hukassa. Tällä hetkellä identiteetinhallinta toimii, vaikka muutamia virhetilanteita kuukausittain syntyykin, mutta nekään eivät ole opiskelijahallintajärjestelmän toiminnallisuudessa. Niin sanottu toinen käyttöönotto tapahtui tammikuussa 2010, jolloin tammikuussa aloittavat opiskelijat saapuivat. Henkilöstöä oli ehditty informoida riittävästi ja mahdollisiin ongelmiin oli osattu varautua etukäteen. Lähes kaikki IDM:n virheet oli korjattu ja avoimen ammattikorkeakoulun opiskelijatietojen käsittelyyn oli löydetty yhteisymmärrys, mikä on auttanut niiden tietojen välittymiseen.

Tulevaisuudessa ASIOin toiminnallisuutta on tarkoitus kehittää siten, että opiskelijan tiedoista välitetään myös tieto siitä, että hän ei tarvitse verkkotunnusta tai sähköpostia. Näin vältytään turhien tunnusten luomiselta identiteetinhallinnassa, vaikka opiskelija kuuluisikin opiskelijatyyppiin, jolle tunnukset automaattisesti tehdään. Lisäksi opiskelijahallintajärjestelmästä välitetään tulevaisuudessa tieto siitä onko opiskelijan todistukset ja todistusjäljennökset tarkistettu. Todistusten tarkistuksen yhteyteen on tarkoitus liittää henkilöllisyyden tarkistaminen. Kun opiskelijan henkilöllisyys on tarkastettu ja hänen tietoihinsa merkitään joko todistukset tai niiden jäljennökset tarkistetuksi, niin identiteetinhallintajärjestelmä vapauttaa opiskelijan tunnukset hänen käyttöönsä vasta sitten. Näin saavutetaan tunnusten osalta se turvallisuustaso, joka vaaditaan korkeakoulujen välisten järjestelmien yhteiskäyttöön. Näihin tulevaisuuden toimintoihin olen varautunut jo siten, että tiedot välitetään ASIOn perusrekisteristä siirtotauluun ja

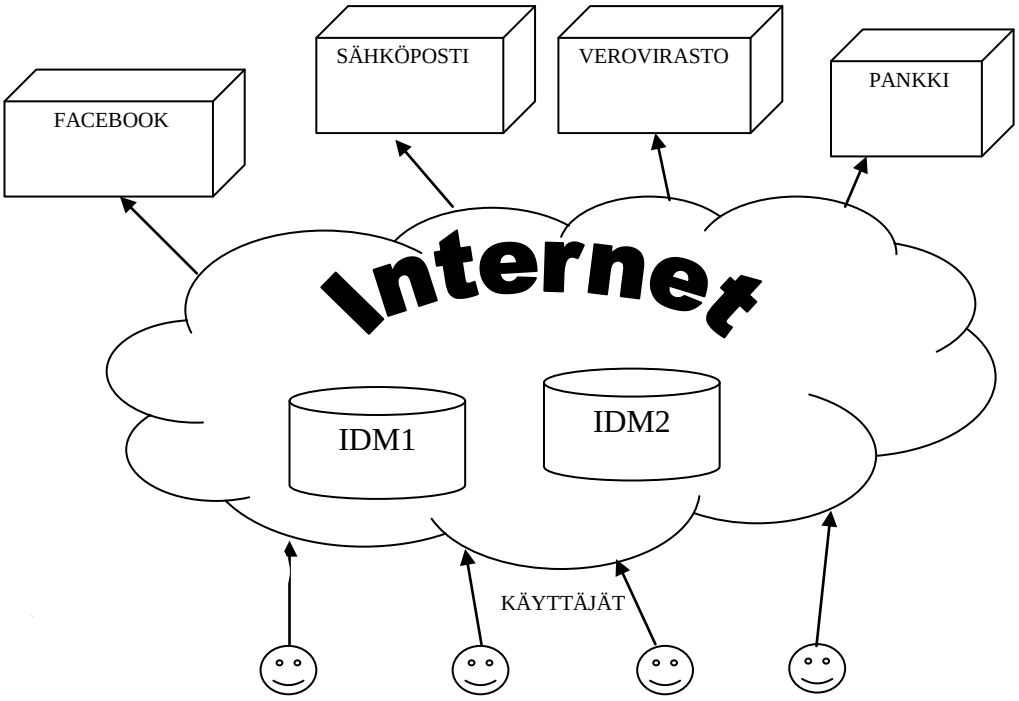
muutoksesta tehdään ilmoitus event_log tauluun, mistä rajapinta-ajuri sen lukee, mutta IDM ei vielä hyödynnä tietoja millään tavoin.

Järjestelmän oltua käytössä muutaman kuukauden huomasin, että tapahtumatiedot ovat tarpeettomia tietoja. Identiteetinhallinnan osalta olennaista on vain tieto siitä, onko opiskelija aktiivinen opiskelija vai ei. Ammattikorkeakoulussa tehtiin linjaus, että väliaikaisestikin keskeyttäneellä, esimerkiksi armeijan ajaksi, opiskelijalla ei ole käyttöoikeutta verkkoon tai koulun sähköpostiin. Tämän vuoksi tapahtumatiedoista saatavat syyt keskeyttämiselle eivät ole olennaisia. Opiskelijat, jotka merkitsevät itsensä poissaolevaksi tietylle lukukaudelle, merkitään myös keskeyttäneeksi, joten tämäkään tapahtumamerkintä ei ole merkityksellinen. Minusta tapahtumatietojen välittäminen on nykyisellään tarpeetonta. Pyrittäessä hyvään tiedonhallintatapaan se pitäisikin lopettaa ja tapahtumiin liittyvät herätteet ja taulut poistaa. Tapahtumiin liittyneet herätteet ja koodit olenkin jo poistanut käytöstä, mutta tarkempi siivous tapahtumien suhteen odottaa opiskelijahallintajärjestelmän huoltotaukoa ja keskustelua identiteetinhallinnan ylläpitäjien kanssa ja heidän näkemyksensä kuulemista tapahtumatietojen käyttötarpeesta tulevaisuudessa.

11 JÄLKISANAT

Identiteetinhallinta on sanana hieman mystinen ja hyvin usein olen huomannut oman alanikin ihmisissä pelon häivähdyksen, kun keskustelu siirtyy identiteetinhallinnan integrointiin johonkin toiseen järjestelmään, mikä vielä hyvin usein on jokin kriittinen perusrekisteri. Opiskelijahallintajärjestelmälle kaltaisia perusrekistereitä pidetään usein syystäkin hyvin arvokkaina ja niihin eikä niiden ympärille haluta tehdä suuria muutoksia. Identiteetinhallinta puolestaan mielletään vaikeaselkoiseksi ja monimutkaiseksi järjestelmäksi, missä vain liian moni asia voi mennä pieleen ja rikkoa kaiken. Asiaa ei auta se, että lopultakin IDM:ää kokonaisuutena käsittelevää kirjallisuutta on vähän ja se monesti mielletään vain pelkäksi hakemistopalveluksi.

Opinnäytetyöni tarkoitus onkin luoda pieni katsaus identiteetinhallintaprojektiin ja juuri siihen perusrekisterin osaan, jota tulee suojella kaikin tavoin. Haluan näyttää muille asiaa miettiville, että integraatio ei ole niin pelottava asia, miltä se monesti kuulostaa. Vaikka se vaatiikin hieman vaivaa, suunnittelua ja huolellisuutta, niin hyödyt ovat kuitenkin suuret. Hyvään lopputulokseen on mahdollista päästä kohtuullisella työmäärällä ilman, että perusrekisterin toiminta vaarantuu. Toivon tuoneeni esille sen, että valmiita ratkaisuja on olemassa, eikä kaikkea tarvitse keksiä itse. Ratkaisut ja toiminnot eivät aina ole monimutkaisia ja vaikeaselkoisia.



KUVA 16. Visio tulevaisuuden identiteetinhallinnasta

Tulevaisuudessa sosiaalisen median vaikutus tietoyhteiskunnassa kasvaa. Yhä useammat tekniset järjestelmät ja ratkaisut toimivat internetissä tai ne ovat ainakin käytettävissä internetin kautta. En usko, että identiteetinhallinta on tässä suhteessa poikkeus. Lähitulevaisuudessa tulemme näkemään kuvan 16 mukaisesti identiteetinhallintajärjestelmiä, joihin voidaan integroida muita järjestelmiä joko identiteetin lähteeksi tai käytettäväksi internetidentiteetin kautta. Internetin ja lähiverkkojen rajat hämärtyvät yhä enemmän, kun internetin anonyymia kaaosta koetetaan hillitä ulottamalla siellä oleviin palveluihin lähiverkoista jo tuttua identiteetinhallintaa. Facebookin kaltainen sosiaalinen nettimedia yhdistettynä helppokäyttöiseen ja vahvan tunnistautumisen kautta turvalliseen netti-identiteettiin on miljardien arvoinen kokonaisuus, mitä kohden isot informaatioteknologiayritykset Microsoftin johdolla ovat koettaneet päästä jo vuosia.

LÄHTEET

CSC. 2010. Haka-käyttäjätunnistusjärjestelmä. WWW-dokumentti. http://www.csc.fi/hallinto/haka. Luettu: 21.06.2010

ComputerWeekly.com. 2007. Identity management: the expert view. WWW-dokumentti. http://www.computerweekly.com/Articles/2007/11/23/225715/Identity-management-the-expert-view.htm. Päivitetty 23.07.2007. Luettu 20.07.2010

Engelberg, Paul. 2009. 5 Keys to a Successful Identity and Access Management Implementation. WWW-dokumentti. http://viewer.bitpipe.com/viewer/viewDocument.do?accessId=12890651. Päivitetty joulukuu 2009. Luettu 16.10.2010.

Haavisto, Heikki 2004. Järjestelmäintegraatio. Helsingin yliopiston tietotekniikka-osaston tiedotuslehti 1.

Hakala, Mika 2006. Tietoturvallisuuden käsikirja. Jyväskylä: Docendo

Hallituksen esitys 96/1998. WWW-dokumentti. http://www.finlex.fi/fi/esitykset/he/1998/19980096. Päivitetty: 24.07.1998. Luettu: 22.07.2010

Henkilötietolaki 523/1999. WWW-dokumentti. http://www.finlex.fi/fi/laki/ajantasa/1999/19990523#a523-1999. Päivitetty: 22.04.1999. Luettu: 22.07.2010

Hovi, Ari 2005. Tietokantojen suunnittelu & indeksointi. Jyväskylä: Docendo

Howes, Timothy A. 1999. Understanding and Deploying LDAP Directory Services. USA: Macmillan Computer Publishing

Jäppinen, Taija 2007. Korkeakouluopiskelijan henkilötietojen suoja. Turun yliopisto, oikeustieteellinen tiedekunta. Tutkielma.

Kaario, Kimmo 2008. Tiedonhallinta. Jyväskylä: WSOYpro / Docendo

Konstari, Timo 1992. Henkilörekisterilaki: säännökset ja käytäntö. Jyväskylä : Gummerus

Lahti, Kai 2003. Organisaation järjestelmäintegraatiot. Jyväskylän yliopisto. Tietotekniikan laitos. Pro gradu –tutkielma. PDF-dokumentti. https://jyx.jyu.fi/dspace/bitstream/handle/123456789/12521/G0000301.pdf?sequence=1. Päivitetty 6.10.2003. Luettu 16.10.2010.

Linden Mikael 2009. Organisational and Cross-Organisational Identity Management. Tampereen teknillinen yliopisto. Thesis for the degree of Doctor of Technology.

Litke, Hans-D 2004. Projektinhallinta. Helsinki: Oy Rastor AB

Microsoft. 2009. Windows Live ID- ja Microsoft Passport Network -palveluiden tietosuojaa koskeva liite. WWW-dokumentti . http://privacy.microsoft.com/fi-fi/windowsliveid.mspx. Päivi-tetty maaliskuu 2009. Luettu 19.07.2010

Microsoft. 2010. Volume Licensing Service Center. WWW-dokumentti. https://www.microsoft.com/licensing/servicecenter/Home.aspx. Luettu 19.07.2010

Murch Richard 2002. IT-projektinhallinta. Helsinki: Edita Publishing Oy

Mäenpää, Olli 2009. Julkisuusperiaate. Helsinki: WSOYpro

NextGov. 1997. Identity Management. WWW-dokumentti. http://www.nextgov.com/the_basics/tb_20080327_1273.php. Päivitetty 27.09.2008. Luettu 20.07.2010

Novell 2009. Driver for JDBC Implementation Guide. Novell Identity Manager. Päivitetty 23.7.2008. Luettu 16.10.2010.

Rosenzweig, Benjamin 2003. Oracle PL/SQL. New Jersey: Pearson Education Inc.

Saarenpää, Ahti 2007. Henkilö- ja persoonallisuusoikeus. PDF-dokumentti. http://www.uwasa.fi/midcom-admin/ais/midcom-serveattachment-4396/persvaa07.pdf. Luettu 22.07.2010

Tähtinen, Sami 2005. Järjestelmäintegraatio. Helsinki: Talentum

Valtionhallinnon tietoturvallisuuden johtoryhmän, VAHTI 2004. Tietoturvallisuus Suomen lainsäädännössä. WWW-dokumentti. http://www.yliopistojentt.fi/VAHTI-CD/Sivusto/lait/suomessa.htm. Luettu 26.07.2010.

Käsitematriisi

Asio – metahakemisto tiedon tuonti		
Asio	**Metahakemisto**	**Selite**
Läsnäolo-tieto	MAMK: student presence	?
Opiskelija-numero	MAMK: student id	Voi olla useita arvoja
Opiskelija-tyyppi	MAMK: student type	A, E, N, V tai C (N = normaali)
Tilakoodi	MAMK: student status	Aktiivi, normi + 1v, harjoittelija, jatkoaika, keskeyttänut, eronnut, poissaoleva, tilap, valmistunut, yli vuoden takainen opiskelija
Tapahtumat	MAMK: student log	Useita arvoja, 1, N, a, b, d, g, G, n
Opiske-luosoite	MAMK: student address	Opiskelijakatu 1 B 2
Tutkinto	MAMK: student programme	Tutkinnon selite
Tutkinto-koodi	MAMK: student programme code	Tutkinnon koodi
Suuntau-tuminen	MAMK: student orientation	Tutkinnon selite 2?
Suuntau-tumiskoodi	MAMK: student orientation code	Tutkinnon koodi 2?
Koulutusoh-jelma	MAMK: student degree	Koulutusohjelman nimi
Koulutusoh-jelman koodi	MAMK: student degree code	Nelimerkkinen koodi
Ala	MAMK: student line	Alan selite
Alan koodi	MAMK: student line code	Alan koodi
Kampus	MAMK: student campus	Kampuksen selite
Kampuksen koodi	MAMK: student campus code	Kampuksen koodi
Ryhmä	MAMK: student group	
Ryhmätun-nus	MAMK: student group code	Lasketaan tämän koodista muut
Aloitusvuosi	MAMK: student starting season	Esim 8S
Aloituspäi-vämäärä	MAMK: student start date	Päivämäärä
Opinto-oikeus	MAMK: student seasons	Lukukausien määrä, esim 8
Sukunimi	Surname	
Etunimet	Given Name	
Kutsuma-nimi	preferredName	
Hetu	MAMK: social security id	
Opiske-luosoite	Postal Address	Rakenteellinen attribuutti sisältää useamman kohdan
Postiosoite	Postal Address	

Käsitematriisi

Kotikunta (koodi)	L	Location, koodi
Kotipostiosoite	homePostalAddress	Rakenteellinen attribuutti sisältää useamman kohdan
Kotiosoite	homePostalAddress	
Kotipuhelin	MAMK: personal phone	
Oma email	MAMK: personal email	Henkkoht email
Kieli	language	Kielen koodi
Kansalaisuus	co	Country, virallinen numeerinen maakoodi
Koulun email	MAMK: school email	opnro@mail.mamk.fi
Koulutustiedotukset	MAMK: publish edu	Tietojani saa käyttää koulutustiedotukseen, true / false
Suoramarkkinointi	MAMK: publish com	Tietojani saa käyttää markkinointiin, true / false
Internet - julkaisu	MAMK: publish internet	Tietojani saa julkaista internetissä, true / false
Matkapuhelintiedotukset	MAMK: publish cellphone	Tietojani saa käyttää kännykkätiedotuksiin, true / false

****

ESIMERKKI ASIO-TRIGGEREISTÄ

****

// heräte, joka reagoi luokkatietojen muutoksiin luokka nimisessä taulussa

```
create or replace
trigger "asio"."idm_rakenne_update3" after
insert
or delete
or update of "luokkatun" on "asio"."luokka" for each row when (new.luokkatun is not
null or old.luokkatun is not null) begin
if inserting then
insert into "idm"."idm_rakenne" values(:new.luokkatun, null, null, null, sub-
str(:new.luokkatun,1,3));
end if;
if updating then
if not :new.luokkatun = :old.luokkatun then
update "idm"."idm_rakenne" set koodi=:new.luokkatun, par-
ent=substr(:new.luokkatun,1,3) where koodi =:old.luokkatun;
end if;
end if;
if deleting then
delete from "idm"."idm_rakenne" where koodi=:old.luokkatun;
end if;
end idm_rakenne_update;
```

```
*******************************
ESIMERKKI IDM-TRIGGEREISTÄ
*******************************

// heräte, joka reagoi idm:n rakennetaulun lisäyksiin

create or replace
trigger "idm"."event_log_rakenne_insert"
after
insert on "idm"."idm_rakenne" for each row declare
	opcode  numeric(1);
	date_format varchar2(64);
	table_name varchar2(64);
	key varchar2(64);
begin
	opcode := 5; /* insert row */
	table_name := 'idm_rakenne';
	key := ('koodi=' || :new.koodi);
	insert into idm_event_log(event_type, event_time, table_name, table_key, status)
		values(opcode, sysdate, table_name, key, 'n');
end;
```