

KARELIA-AMMATTIKORKEAKOULU
Tietojenkäsittelyn koulutus

Emmi Koistinen

JÄRJESTELMÄINTEGRAATIO JA DATAN AUTOMATISOINTI

Opinnäytetyö
Maaliskuu 2020



OPINNÄYTETYÖ
Maaliskuu 2020
Tietojenkäsittelyn koulutus

Tikkarinne 9
80200 JOENSUU
+358 13 260 600 (vaihde)

Tekijä Emmi Koistinen

Nimeke
Järjestelmäintegraatio ja datan automatisointi

Toimeksiantaja
Isolta Oy

Tiivistelmä

Opinnäytetyön tarkoituksena oli selvittää toimeksiantajalle, Isolta Oy:lle, kuinka järjestelmäintegraatio (EAI) ja ohjelmistorobotiikka (RPA) voivat helpottaa yrityksen käytössä olevien järjestelmien yhteen tuomista. Käytössä olevat ohjelmistot haluttiin yhdistää Google Data Studioon, ja saada data liikkumaan automaattisesti eri lähdejärjestelmistä kohteeseen.

Työssä tutustuttiin muun muassa järjestelmäintegraation hyödyntämiseen, suunnitteluun sekä erilaisiin integraatiomalleihin. Lisäksi perehdyttiin hieman ohjelmistorobotiikkaan ja tietoturvaan. Opinnäytetyössä tehtiin kolme erilaista integraatioesimerkkiä eri ohjelmistoja hyödyntäen. Ohjelmistot ovat Microsoft Visual Studio, Google Apps Script ja UiPath Studio. Työn esimerkkitietoina käytettiin OData.org:n ja Fixer.io:n tarjoamia JSON-tiedostoja sekä Karelia-ammattikorkeakoulun käytössä olevaa toiminnanohjausjärjestelmää Dynamics NAV-palvelua.

Opinnäytetyön esimerkit osoittavat, että integraatiota on mahdollista tehdä eri tavoilla ja eri ohjelmistoja hyödyntäen. Monet ohjelmistot ja palveluiden tarjoajat tarjoavat mahdollisuuksia erilaisiin integrointiratkaisuihin. Suunnitteluvaiheessa tulee käydä läpi tarkasti yrityksen tarpeet.

Kieli
suomi

Sivuja 36
Liitteet 1
Liitesivumäärä 1

Asiasanat

Järjestelmäintegraatio, EAI, ohjelmistorobotiikka, RPA



THESIS
March 2020
Degree Programme in Business
Information Technology

Tikkarinne 9
80200 JOENSUU
FINLAND
+ 358 13 260 600 (switchboard)

Author Emmi Koistinen

Title

Enterprise application integration and data automation

Commissioned By

Isolta Oy

Abstract

The purpose of this thesis, commissioned by Isolta Oy, is to investigate how enterprise application integration (EAI) and robotic process automation (RPA) can help companies to bring their software products together. More specifically, three different data sources are connected to Google Data Studio for automatic data flow between systems.

The theoretical part delves into planning and using enterprise application integration. Essential integration patterns, RPA and information security are examined. In the functional part, it is demonstrated how to integrate data between software products such as Microsoft Visual Studio, Google Apps Script and UiPath Studio into Google Data Studio. The data for the examples is from OData.org, Fixer.io and Karelia University of Applied Sciences enterprise resource planning (ERP) Dynamics NAV service.

The results suggest that there are many ways to achieve integration joints. Several software packages and service providers offer different integration options for companies.

Language

Finnish

Pages 36

Appendices 1

Pages of Appendices 1

Keywords

Enterprise application integration, EAI, Robotic process automation, RPA

Sisältö

1	Johdanto	7
2	Järjestelmäintegraatio	8
2.1	Järjestelmäintegraation perusteet	8
2.2	Järjestelmäintegraation suunnittelu	9
2.3	Järjestelmäintegraation hyödyt	10
2.4	Mahdolliset riskit	12
2.5	Integraatiomallit	13
2.6	Ohjelmistorobotiikka	15
2.7	Tietoturva.....	16
3	Opinnäytetyön toimeksianto.....	18
3.1	Isolta Oy:n lähtötilanne	18
3.2	Isolta Oy:n järjestelmien läpikäynti.....	19
4	Opinnäytetyössä käytetyt järjestelmät.....	21
4.1	Järjestelmät ja niiden yhdistämien	21
4.2	Microsoft Visual Studio	22
4.3	UiPath Studio.....	26
4.4	Google Apps Script.....	28
5	Tulokset	29
6	Pohdinta.....	32
	Lähteet.....	34

Liitteet

Liite 1 Apps Script -koodi

Sanasto ja lyhenteet

Access Token	Käyttöoikeustietue, jonka avulla voi päästä käsiksi johonkin.
API	Application programming interface. Ohjelmistorajapinta.
B2Bi	Yritykseltä yritykselle integrointi. Engl. Business-to-business integration.
CRUD-operaatiot	Create, retrieve, update, delete (luo, hae, päivitä ja poista).
CSV	Comma-separated values. Tiedostomuoto, jolla taulukkomuotoinen tieto tallennetaan tekstitiedostoon.
Dashboard	Visuaalinen raportointityökalu, missä esitetään yrityksen liiketoimintatietoja.
Datalähde	Opinnäytetyössä puhutaan Google Data Studion datalähteestä, jonka avulla voi yhdistää yhden tai useamman järjestelmän Google Data Studioon. Datalähde mahdollistaa omien datan siirtojen tekemisen.
EAI	Enterprise Application Integration. Järjestelmäintegraatio.
JSON	JavaScript Object Notation. Tiedostomuoto tiedonvälitykseen.
Metadata	Dataa kuvailevaa tai määrittelevää dataa. Yleisin kuvaus metadataasta on ”tietoa tiedosta”.
MS Dynamics Nav	Yrityskäyttöön tarkoitettu toiminnanohjausjärjestelmä.
NLP	Natural Language Processing. Luonnollisen kielen käsittely.
OData	Open Data Protocol. Mahdollistaa hyvin REST API:n rakentamisen ja hyödyntämisen.
REST	HTTP-protokollaan perustuva arkkitehtuurimalli. Engl. Representational State Transfer.
RPA	Robotic Process Automation. Ohjelmistorobotiikka.
SI	System integration. Sovellusintegraatio.
SPA	Smart Process Automation. Ohjelmistorobotiikka, joka käyttää luonnollisen kielen käsittelyä. Sopii monimutkaisempiin tehtäviin.
Terminaali	Tietokoneen komentoliittymä.

Toiminnanohjaus-järjestelmä	Tietojärjestelmä, jolla integroidaan erilaisia toimintoja yrityksessä. Esimerkiksi varastointia ja jakelua. Engl. Enterprise resource planning, ERP.
VPN	Julkisen verkon yli toimiva yksityinen erillisverkko. Engl. Virtual private network.
Web service	Ohjelmointirajapinta. Mahdollistaa sovellusten välisen vuorovaikutuksen tietoverkon yli.

1 Johdanto

Yritykset usein hyödyntävät monia erilaisia ohjelmistoja. Jotkut ohjelmistoista ovat pilvipalveluita, ja osa on asennettu paikallisesti. Ohjelmistot voivat olla moderneja ja osa perinteisiä järjestelmiä. Jokainen näistä ohjelmistoista mahdollisesti tuottaa ja kerää dataa suuria määriä päivittäin. Tiedot kuitenkin pysyvät omien ohjelmistojensa sisällä, jolloin sovellukset toimivat itsenäisesti, ja niiden tuottamaa dataa on vaikea verrata ja hyödyntää keskenään. Integroinnin avulla järjestelmät tuodaan yhteen, mikä mahdollistaa ohjelmistojen keräämän datan yhdistämisen ja vertailun. Järjestelmäintegraation on tarkoitus helpottaa ja nopeuttaa yrityksen toimintaa ja tämän myötä parantaa sen liiketoimintaa. Verkkokauppatilauksia ei tarvitse kirjata manuaalisesti toiminnanohjausjärjestelmään, vaan tieto liikkuu automaattisesti rajapintojen avulla järjestelmästä toiseen, kun asiakas klikkaa tilauksen eteenpäin.

Järjestelmäintegraation myötä datasta on helpompi rakentaa kattavia dashboardeja. Luvuista, statuksista ja muutoksista rakennetaan selkeitä diagrammeja, joita tarkastelemalla voidaan ymmärtää enemmän yrityksen tilanteesta.

Opinnäytetyössä esitellään järjestelmäintegraation ja ohjelmistorobotiikan perusteet. Tietopohjaa hyödyntämällä rakennettiin kolme erilaista esimerkkiä, joissa integroitiin dataa eri lähteistä. Työssä tutustutaan toimeksiantajan käytössä oleviin järjestelmiin, jotka haluttiin integroida yhteen. Yksi päätavoitteista opinnäytetyössä on tutkia erilaisia mahdollisuuksia, miten yrityksen asiakasdata saadaan integroitua yrityksen käytössä olevaan dashboardiin. Tietoturvasyistä esimerkit tehtiin internetistä löytyvien datalähteiden ja niiden avoimien rajapintojen avulla sekä Karelia-ammattikorkeakoulun MS Dynamics NAV -palvelua hyödyntäen.

Opinnäytetyön toimeksiantajana toimii helsinkiläinen ohjelmistoyritys Isolta Oy. Yrityksen markkinointitiimi käyttää Google Data Studiossa luotuja dashboardeja. Dashboardien data tulee monista eri yrityksen käytössä olevista ohjelmistoista. Yrityksen toiveena oli saada datan siirtovaiheet automaattisesti toimiviksi manuaalisen käsityön sijaan. Automaattiset työvaiheet ilman jatkuvia rutiininomaisia

käsityövaiheita vapauttaisi työntekijöitä muiden tehtävien pariin. Opinnäytetyön tarkoituksena oli löytää mahdollisia ratkaisuja lähdejärjestelmien yhdistämisestä kohteeseen.

2 Järjestelmäintegraatio

2.1 Järjestelmäintegraation perusteet

Järjestelmäintegraatiolla tarkoitetaan datan siirtoa järjestelmästä yhteen tai useampaan. Itsessään järjestelmäintegraatio ei kuitenkaan ole työkalu tai järjestelmä, vaan tapa ajatella. Sen avulla voidaan hahmottaa yrityksen tietoteknistä arkkitehtuuria. Integraatiosta voidaan käyttää kahta eri tason termiä. Sovellusintegraatio on enemmän teknisempi kuvaus sovellustason toiminnasta eli sovellusten tekemistä integraatiotehtävistä. Järjestelmäintegraatio kuvastaa ihmisistä ja sovelluksista koostuvaa suurempaa järjestelmää ja toimintatapaa. (Tähtinen 2005.)

Järjestelmäintegraatio pitää sisällään toimintatapoja, joiden tarkoituksena on tuoda yrityksen käytössä olevat järjestelmät yhteen. Yksi järjestelmäintegraation päätavoitteista on kehittää yrityksen liiketoimintaa. Sen avulla parannetaan sovelluskehitystä, minkä myötä irrallisista sovelluksista tehdään yksi suuri hallintakokonaisuus, ”supersovellus”. Irralliset ja erilaisesti toimivat ohjelmistot tuodaan yhteen yhdeksi hallittavaksi kokonaisuudeksi. Yhtä suurta hallintakokonaisuutta voidaan monitoroida ja hallita selkeämmin kuin montaa irrallista sovellusta. (Tähtinen 2005, 13–15.)

Integraationäkökulmaa voi lähestyä jo meille arkipäiväistyneiden esimerkkien kautta. Kaupunkien sekä maiden väliset tiet ja sillat mahdollistavat niin ihmisen kuin tavaran liikkumisen paikasta toiseen. Puhelinverkkojen myötä ollaan yhteydessä muihin ihmisiin, ja Google-tilille kirjautuneelle käyttäjälle tulee automaattisesti merkintä kalenteriin, kun hän varaa lomamatkan. Erilaisia integraatioita tapahtuu joka puolella. (Manouvrier & Menard 2008.)

Automatisointi on järjestelmäintegraatiossa liiketoimintaprosessien päämäärä ja sen yksi tärkeimmistä ominaisuuksista. Se nopeuttaa ja tehostaa erilaisia prosesseja. Työntekijöiden tekemä manuaalinen käsityö halutaan saada automaattisesti toimivien järjestelmien työtehtäväksi. Integroinnin tarkoitus on helpottaa ja tehostaa työntekoa, sillä data on usein hajautettuna useampaan eri järjestelmään. Mitä enemmän yhteistyötä yrityksen järjestelmät tekevät, sitä enemmän ne hyödyntävät yritystä. (Flashnode Oy 2019; Tähtinen 2005.)

2.2 Järjestelmäintegraation suunnittelu

Tarkkojen suunnitelmien avulla voidaan välttää mahdollisia turhia työvaiheita. Järjestelmäintegraatio voi olla suuri ja mahdollisesti hintava prosessi, jonka suunnittelee yleensä tietotekniikka-arkkitehti. Integraation elinkaarta on lähdettävä suunnittelemaan heti prosessin suunnittelun aloitusvaiheessa. Suunnitteluvaiheen voi aloittaa käymällä läpi kysymyspatteristoa, missä kartoitetaan integrointityön tärkeyttä ja tunnistamalla sen eri vaiheita. (JUHTA 2017; Tähtinen 2005.)

Vastauksia voidaan etsiä kysymyksiin:

- Miten suuri tarve integroinnilla on ja miksi?
- Mitkä ovat yrityksen teknologiset valinnat rajapintojen ja ohjelmistojen suhteen?
- Miten prosessi dokumentoidaan ja kuka on siitä päävastuussa?

(JUHTA 2017.)

Prosessin suunnitteluvaihe toteutetaan usein myös organisaation eri osajien näkökulmista. Johtoryhmä voi tarkastella prosessin tietovirtoja ja kokonaisuutta, kun taas esimerkiksi ohjelmoijat ja tietoturvavastaavat tutkivat prosessin tietoteknistä puolta. Jotta suunnitelmat olisivat yhtenäisiä, on tapana toteuttaa ne standardien ja viitekehysten mukaisesti. (JUHTA 2017.)

Järjestelmien väliset linkit ja yhteydet rajapintoineen tulee selvittää huolellisesti ja mikä integraatiomalli on yritykselle sopivin ratkaisu. On tärkeää miettiä myös,

kuinka ohjelmistoriippuvainen integraatiomalli on. Joustavuus ja muuttuvuuden sieto vähentävät vikatiloja. On varauduttava, että käytössä olevia järjestelmiä päivitetään ja uudet muutokset voivat rikkoa mahdolliset linkit toisiin järjestelmiin. (Tähtinen 2005.)

Järjestelmäintegraation toteutuksessa tulee huomioida erilaiset tietotyypit ja mahdolliset tietomuunnokset. Tiedon siirto monesta eri lähteestä yhteen kohteeseen vaatii usein tietotyypin muuttamista, jotta esimerkiksi verkkopalvelusta tuleva data ja CSV-muotoinen tieto pystyvät päivittymään samaan kohteeseen. Nykyisissä järjestelmäintegroiin tarkoitetuissa järjestelmissä tietomuutokset ovat helppo tehdä muutamaa asetusta muokkaamalla. Usein myös valmiin järjestelmän käyttö on suotavampaa ja turvallisempaa kuin omien skriptien tekeminen, jos kyseessä on monimutkaisempi integrointi. Valmiit integroiin tarkoitetut työkalut eivät välttämättä vaadi ollenkaan koodin kirjoittamista. (Microsoft 2018.)

Integroiminen voi tapahtua myös yritykseltä yritykselle. Kaksi tai useampi yritys kommunikoi keskenään, mikä mahdollistaa tehokkaan yhteistyön koskien yritysten asiakkaita ja tavarantoimittajia. Yrityksmaailmassa nopeus ja luotettavuus ovat tärkeässä asemassa, ja onnistunut yritysten välinen integrointi voi mahdollistaa tämän. (IBM 2020.)

2.3 Järjestelmäintegraation hyödyt

Järjestelmäintegraatio tulee tehdä sellaiseksi, että siitä hyöttyy konkreettisesti. On oltava selkeä suunnitelma, mitä sen avulla halutaan tavoittaa. Sen avulla voidaan esimerkiksi automatisoida informaationkäsittely, mikä johtaa kustannussäästöihin. Liiketoimintaprosessien tehostaminen on aina hyödyksi yritykselle. Järjestelmien yhdistämisen avulla eri ohjelmistojen liiketoimintaa parantava data saadaan nidottua yhteen. Yrityksen käytössä olevien yksittäisten ohjelmistojen erikseen käyttäminen vähenee ja järjestelmiä käsitellään yhtenä suurena organisoituna kokonaisuutena. (Manouvrier & Menard 2008; Tähtinen 2005, 23.)

Datan kerääminen hyödyntää yrityksen liiketoimintaa, sillä kerätyllä tiedolla voidaan huomata mahdollisia puutteita ja ongelmakohtia. Esimerkiksi asiakaspalvelija pystyy parantamaan asiakastyytyväisyyttä kerätyn datan avulla. Asiakkaan soittaessa yritykseen voi asiakaspalvelija löytää asiakkaan tiedot ja tapahtumahistorian nopeasti henkilötietojen perusteella. Näin ollen asiakkaan ei tarvitse kertoa samoja asioita toistamiseen. Datan hyvän hallinnan avulla myös yrityksen työntekijöiden tyytyväisyys kasvaa. (Väre 2019, 47.)

Järjestelmäintegraation avulla voidaan myös pienentää riskejä. Yrityksellä voi olla monia eri järjestelmiä käytössä, ja niiden dataa tarkastellaan yksi kerrallaan. Osasta ohjelmista siirretään dataa toiseen käsityönä. Tämä on hidasta ja mahdollistaa tietovirheet. Sairastapauksissa tiedon siirto voi kärsiä, sillä työtä ei välttämättä ole hajautettu monelle eri työntekijälle. Mahdolliset poissaolot eivät saisi vaikuttaa datan luotettavaan päivittymiseen. (Flashnode Oy 2019.)

Järjestelmäintegraation on tarkoitus helpottaa työntekoa siirtämällä dataa tietoturvallisesti, mikä on yritysmaailmassa erityisen tärkeää. Hyvä tiedonkulku ja -siirto ovat välttämättömiä ominaisuuksia. Datan automatisointi vapauttaa työntekijöitä esimerkiksi uusien tai taka-alalle jääneiden projektien pariin. Myös palkan maksua ajatellen on viisaampaa maksaa palkkaa työntekijälle asiantuntijatyöstä kuin rutiinitehtävien teosta. (Artsoft 2017; Väre 2019, 49.)

Järjestelmäintegraatio toimii yleensä yrityksen liiketoimintaprosessissa tärkeänä raportoinnin työvälineenä liiketoimintatiedon hallinnan kanssa (engl. Business intelligence, BI). Sen avulla voidaan siirtää yritykselle tärkeää dataa järjestelmistä raportointityökaluihin. Data itsessään ei kerro mitään, mutta metadatan avulla saadaan selville mitä eri luvut ja tiedot merkitsevät. Datan parissa työskentelevän henkilön tulee tietää, mikä tieto on tärkeää, jotta tiedetään, millaista dataa kerätään. Markkinoinnin tehostamisessa on hyvä seurata niin reaaliaikaisesti kuin historiallisesti asiakaskäyttäytymistä. Voidaan tutkia, onko kampanjoilla vaikutusta keneen ja miten. Työntekoa on helpompaa tehostaa oikeaan suuntaan, kun tiedetään, mitä tehdään. Oikeanlaisella datan esittämisellä mahdollistetaan asian oikein hahmottaminen. Liian vaikeasti tai epäselvästi kuvattu data voi johtaa väriin päätelmiin ja päätöksiin. (Tähtinen 2005; Vaala 2017.)

Datan hyvä integrointi parantaa yrityksen datan analysointia ja mahdollistaa tehokkaampien mittareiden tekemisen. Mitä enemmän tietoa yrityksellä on omasta toiminnasta eri asioihin, sitä paremmin niin sisäistä kuin ulkoista toimintaa voidaan kehittää. Yrityksen sisäisen toimintaympäristön strategioita osataan muuttaa ja tavoitteista pystytään tekemään saavutettavia, kun nähdään, mikä toimii ja mikä ei. Ulkoisen toiminnan kehittämisessä on tärkeää tietää asiakkaiden tarpeet ja kokemukset yrityksen tarjoamasta palvelusta tai tuotteesta. Luotettavasti eri lähteistä kerätty data mahdollistaa diagrammien tiedon ajantasaisuuden. Mittareiden avulla voidaan tehdä erilaisia hälytyksiä tai tutkia, onko tarkasteltavan kohteen tilanne huono, neutraali vai hyvä. (Pietikäinen 2016.)

2.4 Mahdolliset riskit

Huonosti ja nopeasti toteutettu integraatoratkaisu voi olla riski tietoturvan kannalta, sillä kyseessä on yksityistä dataa. Ratkaisujen käyttöönotossa tulee myös olla vankka tietämys, miten integroiminen käytännössä toimii ja onnistuu. Järjestelmäintegraatio on jatkuva prosessi ja sen käyttöönotto voi tulla kalliiksi kustannuksesi, jos integroinnin tarve ei ole välttämätön. Järjestelmien väliset yhteydet tulee toimia oikein, jotta siirtyvä informaatio on luotettavaa. Huonosti yhteen integroidut järjestelmät voivat näyttää väärää tietoa, ja aiheuttaa esimerkiksi tappiota yritykselle, jos väärän tiedon pohjalta lähdetään tekemään päätöksiä. Lisäksi kannattaa myös pohtia onko järkevintä ulkoistaa työnteko jo valmiita integraatoratkaisuja tarjoavalle yritykselle. (Flashnode Oy 2019; Tähtinen 2005.)

Ohjelmistojen välisien integraatiolinkkien tulee olla joustavia, sillä järjestelmät päivittyvät ja muuttuvat ajansaatossa, jolloin rajapintoihin voi tulla muutoksia. Integroidut järjestelmät voivat lakata toimimasta yhtäkkiä. Keskitetyssä järjestelmäintegraatiossa tieto kulkee lähdejärjestelmästä jonkin ohjelmiston tai esimerkiksi tietokannan kautta kohdejärjestelmään, jolloin vikatilanteissa esimerkiksi tietokantaan, lakkaavat kaikki yhteydet toimimasta oikein. (Tähtinen 2005.)

Onnistuneen integroinnin lisäksi dashboardien tulee olla laadukkaasti tehtyjä. Huonosti toteutettu dashboard voi viedä katsojan huomion väärään asiaan tai se voidaan ymmärtää väärin. Liika värien käyttö saattaa haitata diagrammien lukemista. Myös punavihersokeus tulee huomioida, jos käyttää punaista ja vihreää väriä keskenään. Raportti tulee osata myös kohdentaa halutulle yleisöille oikein. (Vaala 2017.)

Huonosti toteutettu B2Bi voi aiheuttaa riskejä. Yritysten välisen järjestelmäintegraation mahdollinen joustamattomuus luo haasteita. Haasteiksi lukeutuvat muun muassa toteutuksen nopeus ja mahdolliset tietoturvariskit. (Korhonen 2003.)

2.5 Integraatiomallit

Integraatiomallit (engl. Integration patterns) toimivat arkkitehtuuripohjana järjestelmäintegraatiolle. Mallin valitseminen riippuu yrityksen integraatiotarpeesta. Yrityksen kannattaa myös suunnitteluvaiheessa pohtia yrityksen tulevaisuutta ja uusia mahdollisia liitoksia integraatiomalliin. Nykyiset järjestelmät mahdollistavat monia erilaisia rajapintamahdollisuuksia datan siirtoon. Dataa pystytään lataamaan tietokannoista, järjestelmistä ja tiedostoista. Graafisten integrointiin tarkoitettujen työkalujen käyttö on käyttäjäystävällistä ja erilaisten integrointiratkaisujen rakentaminen selkeää. (Hohpe 2003; Microsoft 2018.)

Koska kyseessä on jatkuva liiketoimintaan vaikuttava prosessi, on tärkeää tehdä hyvä prosessisuunnitelma. Integroinnin on kuitenkin oltava yritykselle hyödyllinen. Järjestelmäintegraation vastuuhenkilön olisi suositeltavaa tehdä jatkuvaa raportointia yrityksen integroinnista ja muutoksista, jotta yrityksen muut työntekijät ymmärtävät yrityksen uuden prosessin. Selkokieliisyys ja mahdollisimman yksinkertainen ohjeistus raportoinnissa pitävät muut ajan tasalla. Itse järjestelmäintegraationkin suositellaan olevan mahdollisimman yksinkertaisesti toteutettu, jotta sen eri linkkejä ja rakennetta on helpompi ymmärtää, vaikka integraatioon tarkoitettujen järjestelmät pitävät huolen eri skeemoista mistä käyttäjän ei tarvitse huolehtia. (Tähtinen 2005.)

Järjestämästä järjestelmään siirtyvä, eli Point-to-point -malli, toimii pienissä integraatoratkaisuissa, jos täytyy integroida kahdesta neljään järjestelmää yhteen. Järjestelmien väliset linkit yhdistyvät toisiinsa rajapintojen kautta hajautetusti. Mitä enemmän järjestelmiä malliin tulee, sitä sekavampi ja sotkuisempi integraatio muodostuu. Tämä johtuu ohjelmistojen välisien linkkien eksponentiaalisesta kasvusta. Point-to-point -malli ei ole skaalautuva, ja integraation liittymiä täytyy hallinnoida yksi kerrallaan. Yrityksen on suositeltavaa miettiä ennen prosessin aloitusta, voiko järjestelmäintegraatioon mahdollisesti tulla lisää ohjelmistoja tulevaisuudessa. (Hohpe 2003; Flashnode Oy 2019.)

Keskitetyssä eli Hub-and-spoke -mallissa järjestelmien data siirtyy yhteen ohjelmistoon, jonka kautta ne integroituvat kohdejärjestelmiin. Keskitetty integraatiomalli on suositeltavaa, jos yrityksellä on monta eri järjestelmää integroitavana. Järjestelmiä on helppo lisätä lisää eikä linkkien määrä kasva eksponentiaalisesti. Ongelmaksi voi muodostua välikappaleena toimivan järjestelmän vikatila, mikä aiheuttaa järjestelmäintegraation pysähtymisen joka suuntaan. (Tähtinen 2005; Flashnode Oy 2019; Artsoft 2017.)

ESB (engl. Enterprise service bus) on Hub-and-spoke -mallia hajautetumpi. Reititys tapahtuu yhden välitystoiminnallisuuden sijasta monessa eri arkkitehtuuriverkon kohdassa. Eli ESB-alustassa on monta eri toiminnallisuutta toteuttamassa integraatiota. Se pystyy vastaanottamaan erilaisia teknologioita, kääntämään tietoa ja lähettämään sitä eteenpäin. (Flashnode Oy.)

Selkeänä esimerkkinä Tähtinen (2005) kuvailee kirjassaan point-to-point ja hub-and-spoke -integraatiomallien eron puhelinverkolla. Ennen nykyistä teknologiaa kulkivat sähkölinjat ristiin rastiin toisiinsa, jotta yhteyden muodostuminen onnistui. Jos paikasta X haluttiin soittaa paikkaan Y, oli näiden välillä oltava yksi linja. Kyseinen tapa kuvastaa Point-to-point -mallia. Mitä suuremmaksi verkot alkoivat laajeta, oli sähkölinjoja liikaa. Sen jälkeen kehittyivät puhelinkeskukset. Niiden avulla linjat olivat keskitetty yhteen kohteeseen, jonka kautta ne kulkivat toiseen. Keskitetty tapa integroida on Hub-and-spoke -mallin mukainen.

Asynkroninen eli ei-reaaliaikainen tiedonsiirto mahdollistaa sen, että lähettävä ja

vastaanottava järjestelmä eivät ole ajallisesti keskenään toisistaan riippuvaisia. Esimerkki tällaisesta on sähköpostin lähetys, missä viesti lähetetään kohteelle vasta, kun käyttäjä haluaa sen lähettää. Synkroninen tiedonsiirto toimii reaaliaikaisesti ja tieto liikkuu järjestelmien välillä heti. Puhelimella soittaminen on synkronisesti tapahtuva. Integraatiojärjestelmät mahdollistavat yleensä joko molemmat tai vähintään toisen tiedonsiirtotavan. (Hohpe 2003; Tähtinen 2005.)

2.6 Ohjelmistorobotiikka

Järjestelmäintegraatioon liittyy usein myös ohjelmistorobotiikka. Ohjelmistorobottille voi asettaa samoja työtehtäviä kuin ihmiselle. Sen avulla voidaan korvata ihmisen rutiinitehtäviä ja keskittyä muihin haasteisiin, jotka vaativat enemmän luovuutta, eivätkä ole säännönmukaisia. Rutiinin tulee olla tiettyyn sääntöön perustuva toiminto. On verrattu, että ihmisellä voi mennä yhden rutiininomaisen tehtävän tekemiseen noin 10–20 minuuttia, ja ohjelmistorobotilla alle minuutti. Robotiikan tarkoituksena on kasvattaa tuottavuutta, sillä pienemmillä resursseilla saadaan aikaan enemmän palveluita sekä hyödykkeitä. Ohjelmistorobotiikka ei vähennä organisaatiossa työntekoa, sillä se mahdollistaa uusien työtehtävien luomisen. (Kauhanen 2016; Kolehmainen 2016; Perttilä 2017; UiPath 2020.)

RPA:n lisäksi on myös älykkäämpää ohjelmistorobotiikkaa eli SPA:ta joka on tarkoitettu monimutkaisempiin tehtäviin. Kun RPA toimii pelkästään sille määrättyjen säännösten mukaan, pystyy SPA koneoppimaan ja toimimaan prosessien muutosten mukaan. Muun muassa NLP:n avulla SPA-ohjelmistorobotit pystyvät hyödyntämään uutta materiaalia. Molemmat ohjelmistorobotiikkavaihtoehdot ovat hyödyllisiä riippuen yrityksen tarpeista hyödyntää robotiikkaa. Ne toimivat myös hyvin yhteen, jolloin RPA tekee säännönmukaisia rutiinitehtäviä kuten raportteja ja siirtää dataa järjestelmästä toiseen. Samaan aikaan SPA:ta pystyy hyötykäyttämään esimerkiksi asiakaspalautteen läpikäynnissä, ja rakentamaan uutta algoritmia asiakkaiden toimintatapojen pohjalta. Esimerkiksi chatbotit ja virtuaaliset assistentit ovat SPA-ohjelmistorobotteja. (Haikonen 2016.)

Samalla tapaa kuin järjestelmäintegraatiossa, ohjelmistorobotiikka vähentää tietovirheitä. Se toimii aina sen säännön mukaan, mikä sille on määrätty. Robotin

ohjelmointi ei vaadi perinteisiä ohjelmointitaitoja, vaan yleiset tietotekniikan taidot riittävät. On kuitenkin oltava tarkkana, että määritelmät ovat oikein, ja robotti tekee pelkästään halutut asiat. Väärin toimiva ohjelmistorobotti voi synnyttää mahdollista tuhoa nopeasti. On arvioitu, että onnistuneesti tehtyjen ohjelmistorobottien kustannukset maksavat itsensä takaisin jo puolessa vuodessa. Mika Vainio-Mattila (2015) kuvailee, että suuri osa tehtävästä tietotyöstä johtuu puutteellisesta integraatiosta järjestelmien välillä. Sen myötä ihmiset joutuvat hoitamaan manuaalisesti järjestelmien väliset integraatiopulmat. (Kolehmainen 2016.)

Ohjelmistorobotiikka hyötykäyttää rajapintoja samalla tapaa kuin järjestelmäintegraatioon tarkoitettuja ohjelmistot. Ohjelmistorobotiikkaan suunnitellut ohjelmistot sisältävät työkaluja, minkä avulla robotille voidaan määritellä erilaisia tehtäviä ja säännöksiä. Ohjelmistorobotti pystyy yhdistämään monen järjestelmän tiedot yhteen kohdejärjestelmään sekä tiedostoon tai vastaavasti hajauttamaan tiedot eri kohteisiin. One-to-one -mallissa ohjelmistorobotti siirtää dataa kahden järjestelmän välillä, kun taas esimerkiksi many-to-one -mallissa data siirtyy monesta järjestelmästä yhteen kohdejärjestelmään. (Perttilä 2017; UiPath 2020.)

2.7 Tietoturva

Tietoturva on informaation luottamuksellisena pitämistä. Tiedon tulee olla saavutettavissa vain sitä tarvitseville työntekijöille ja järjestelmille. Datan tulee pysyä eheänä ja muuttumattomana informaation siirron aikana. Yrityksen oma salattu tietoverkko turvaa myös informaatiota, joten etätyöskentelyssä on suositeltavaa käyttää VPN-yhteyttä. (Tähtinen 2005.)

Henkilörekisterin asiakastiedot ovat yksityistä dataa, ja niiden tulee pysyä salassa. Asiakasdatasta tulee poimia vain yrityksen kasvua kiinnostava data, ei asiakkaan henkilökohtaisia tietoja. Jos ohjelmisto tarjoaa REST-rajapintaa järjestelmien väliseen yhteyteen, sen API-avain tulee pitää salassa. Avainta ei tule kirjata minnekään ylimääräiseen paikkaan, josta se voisi vuotaa eteenpäin. Datan käsittelyssä tulee noudattaa tietosuojalakea. Tietosuoja-asetus vaatii myös pitämään

datan laadun asianmukaisena ja henkilötietojen tulee olla ajan tasalla. Datan tulee olla tarpeellista ja hyvän datan hallinnan avulla ylimääräinen ja turha data tulee poistaa. (Intercom 2019; Rousku 2014, 130; Väre 2019, 51–52.)

Integraatioprosessisuunnitelmassa tulee huomioida, kuinka tieto päivittyy järjestelmissä. Tietokantojen päivitys käsin voi rikkoa tietokantarakenteen ja olla näin myös tietoturvaton. Erilaiset integraatioiden rakentamiseen mahdollistavat ohjelmistot toimivat rajapinnoille sopivilla tavoilla pitäen tiedon eheänä. Integraatio- tai ohjelmistorobotiikkasuunnitelman rinnalle tulee tehdä riskienhallintasuunnitelma, jos datan sisältö on arkaluonteista, kuten esimerkiksi henkilötiedot. Prosessisuunnitelmassa läpikäydään muun muassa mahdolliset riskit ja tunnistetaan ne, eli tehdään riskianalyysi. Tämän pohjalta voidaan toteuttaa toimintasuunnitelma. (Aihkisalo & Halén & Holmström & Jurmu & Kääriäinen & Matinmikko & Seppälä & Tihinen & Tirronen 2018; Hohpe 2003.)

Tiedon käsittely ja verkossa näkyvien rajapintojen yleistyminen on kasvattanut tietoturvauhkien määrää. Vuonna 2018 yksi yleisimmistä tietoturvauhkista on ollut SQL-injektiot eli tietokantoihin pääsyn mahdollistava tekniikka. Yleisesti ottaen ohjelmistorobotiikka on tietoturvallista, sillä robotti mahdollistaa tiedon eheyden ja tekee vain sille määrättyt tehtävät. Tärkeää on kuitenkin muistaa päivittää järjestelmät, sillä vanhentunut ohjelmisto voi tehdä tietoturva-aukon, mikä mahdollistaa hyökkäyksen kohteeksi joutumisen. (Aihkisalo ym. 2018.)

Ohjelmistorobotiikassa ja järjestelmäintegraatiossa on tärkeää huolehtia, kuinka prosesseja hallitaan. On selvitettävä miten ja kuka niitä ylläpitää ja kehittää. Lisäksi on varmistettava järjestelmien ajantasaisuus. Ohjelmistorobotiikassa on myös pidettävä huoli robottien hallinnasta. Tietoturvaa voi edistää ja hallita erilaisin keinoin. Järjestelmien automaatiota kannatta kehittää vain pienin askelin, jotta tila on stabiili joka muutosvaiheessa. Erilaiset integrointi- ja robotiikkavaiheet ovat suositeltavaa tallentaa lokihistoriaan, jotta muutoksia on mahdollista tarkastella myöhemmin. Työkalujen ja järjestelmien testaaminen on myös kannattavaa kehitys- ja käyttöönottovaiheessa. Erilaisiin protokolleihin ja suojaustyökaluihin on tärkeää tutustua ennen prosessin aloitusta. (Aihkisalo ym. 2018.)

3 Opinnäytetyön toimeksianto

3.1 Isolta Oy:n lähtötilanne

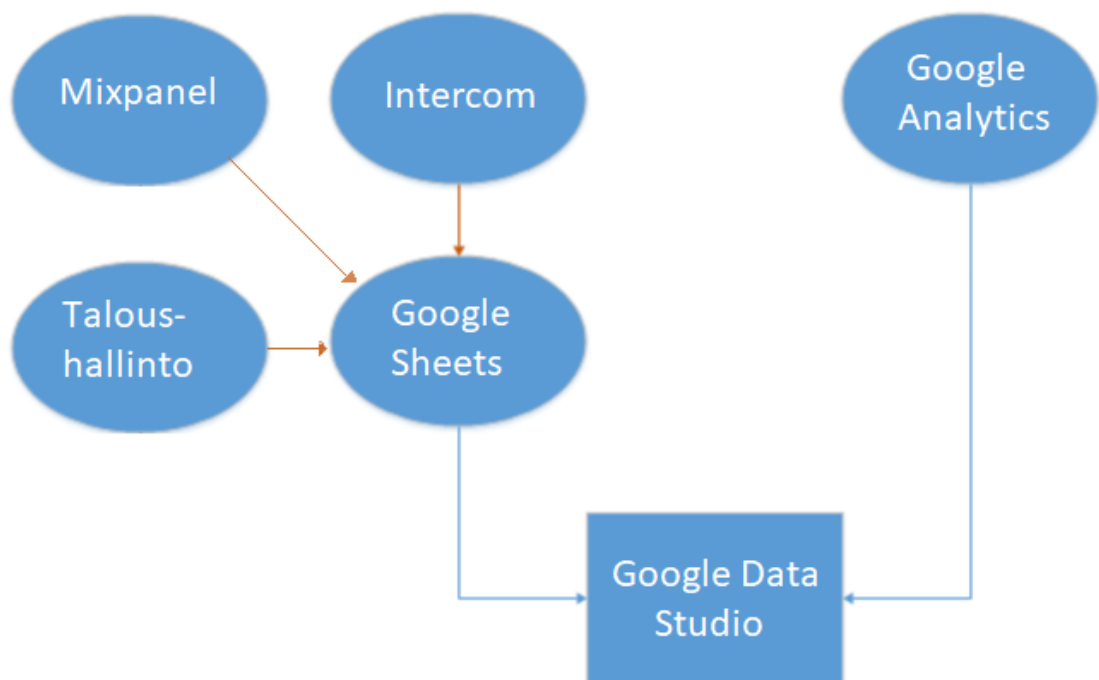
Lähtötilanteessa Isolta Oy:n käytössä on manuaalinen dataintegraatoratkaisu markkinoinnin datan hallinnassa. Yrityksen asiakasdata on eri järjestelmissä hajautetusti, mikä muodostaa yritykselle integraatiohaasteita. Opinnäytetyön tarkoituksena oli yhdistää ohjelmistot Google Data Studioon, niin että tieto liikkuisi järjestelmien välillä ilman ylimääräisiä käsityövaiheita. Lähtötilanteessa käytössä ei ollut suoraan järjestelmäintegraation tarkoitettuja työkaluja tai ohjelmistoja. Lähtötilanteessa ohjelmistojen erilaisten rajapintojen takia järjestelmien yhdistäminen kohdejärjestelmään suoraan ei ole onnistunut.

Yritys käyttää dashboardeja, jotka on tehty suosituksien mukaisesti, kuinka ja milaista dataa kannattaa kerätä, jotta siitä hyötyy mahdollisimman paljon. Ohjelmistot keräävät yritykselle arvokasta dataa, joten oli tärkeää myös mahdollistaa datan siirto mahdollisimman kätevästi kohteeseen, eli Google Data Studioon. Dashboardiin tarvitaan muun muassa tietoa asiakkaiden kokonaismäärästä sekä lajittelut asiakkaan statuksesta eli asiakkuudesta.

Yrityksen asiakasrekisterinä toimii Intercom ja Google Analytics näyttää yrityksen sivuston reaaliaikaista asiakaskäyttäytymistä. Analyticsin lisäksi käytetään Mixpanelia, jonka avulla tutkitaan asiakashistoriaa ja -käyttäytymistä. Ohjelmistoa hyödynnetään pidemmän aikavälin tutkimiseen ja asiakaskäyttökokemuksen parantamiseen. Yritys käyttää myös omaa taloushallinnon järjestelmää, jota kautta tarkastellaan maksuliikennettä.

Tässä opinnäytetyössä käsiteltiin tarkemmin vain yhden järjestelmän yhdistämistä kohteeseen. Selkeyden vuoksi yrityksen lähtötilannetta kuvaillaan vain yhden ohjelmiston näkökulmasta eikä kaikkien käytössä olevien järjestelmien. Opinnäytetyön malliesimerkit tehtiin kuitenkin muita datalähteitä hyödyntäen.

Lähtötilanteessa Intercomin data pitää ensin ladata manuaalisesti järjestelmästä omalle tietokoneelle, minkä jälkeen se siirretään Google-laskentataulukoon. Lopulta tieto ladataan datalähteen kautta Google-laskentataulukosta Google Data Studioon, missä tehdään yrityksen dashboardit. Googlen välisissä työkaluissa tieto liikkuu automaattisesti, sillä rajapinnat mahdollistavat yhteen toimivuuden. Kuvassa 1 näkyy yrityksen lähtötilanne. Punaisella merkityt nuolet merkitsevät datan siirron haastavia välejä.



Kuva 1 Yrityksen lähtötilanne.

Yrityksen käytössä olevien dashboardien päivittyminen haluttiin saada automatisoiduksi, jotta ne pystytään ottamaan jokapäiväiseen käyttöön. Automatisointi mahdollistaa luotettavan datan seurannan työntekijöille eikä vanhentunut tieto vaikuta mahdollisiin muutoksiin. Datan liikkuminen kohdejärjestelmään voi toimia asynkronisesti, mutta säännöllinen päivittyminen on kuitenkin tarpeen.

3.2 Isolta Oy:n järjestelmien läpikäynti

Markkinointitiimin käytössä on muutama eri järjestelmä, joiden avulla dataa kerätään. Kyseisiä järjestelmiä haluttiin integroida yhteen. Kappaleessa 3 tutustuttiin

yrittäjien lähtötilanteeseen ja käytössä oleviin järjestelmiin, mutta tässä luvussa käydään järjestelmiä läpi hieman tarkemmin.

Intercom tarjoaa chat-palvelumahdollisuuden yrityksen verkkosivuille sekä tallentaa asiakasdataa, eli se on henkilörekisteri. Asiakasdata pitää sisällään asiakkaan tiedot, mitkä ovat kirjattu rekisteröinnin yhteydessä. Näitä tietoja ovat muun muassa etu- ja sukunimi, sähköpostiosoite ja puhelinnumero. Jokaisella asiakkaalla on myös henkilökohtainen ID. (Intercom 2019.)

Intercomilla on REST -arkkitehtuurimalli, ja oman henkilökohtaisen datan integroimiseen tarvitaan yksityinen API-avain eli Access Token. REST on HTTP-protokollaan perustuva arkkitehtuurimalli, ja toimii HTTP-pyyntöillä (POST, GET, PUT ja DELETE). API on sovellusrajapinta, joka mahdollistaa sovellusten välisen vuorovaikutuksen. Data on Intercomissa JSON-tiedostomuodossa. Se on kevyt-rakenteinen ja helppolukuinen tiedostoformaatti tiedon säilyttämiseen ja lähettämiseen. (Intercom 2019; Rouse 2019; W3schools 2019.)

Yrityksen käytössä olevan Intercomin asiakasdata ladataan lähtötilanteessa manuaalisesti Google Sheetsiin. Google Sheets on Googlen tarjoama taulukkolaskentaohjelma. Yrityskäytössä Google tarjoaa muun muassa työkaluja tietoturvan hallinnointiin. Google-laskentataulukon on mahdollista ladata CSV-tiedostomuodon lisäksi myös muita datan muotoja liitännäistyökalujen avulla. Taulukko on yhdistetty datalähteen avulla Google Data Studioon. (Google G Suite 2019.)

Google Analyticsin voi yhdistää omalle verkkosivustolle seuraamaan kävijämäärää ja asiakaskäyttäytymistä. Kyseistä järjestelmää voidaan hyödyntää markkinoinnissa ja sivuston kehittämisessä. Yksittäisenä ohjelmistona datan seurantaan se ei kuitenkaan ole kovin kattava. (Google Marketing Platform 2019.)

Mixpanelin avulla voidaan tarkastella asiakaskäyttäytymistä pidemmällä aikavälillä ja analysoida erilaisia vaiheita. Ohjelmiston avulla voidaan kehittää asiakaskokemusta ja mitata muutoksia. Työkalun tarkoituksena on kehittää ja parantaa yritystoimintaa. Mixpanel tarjoaa erilaisia integrointimahdollisuuksia ja kirjastoja ohjelmointiin, mutta ei suoraa väylää Google Data Studioon. (Mixpanel 2020.)

Google Data Studio on ohjelmisto, joka on kehitetty visualisointia varten. Se tukee eri liittimiä eli datalähteitä. Google Sheets ja Google Analytics ovat valmiita Google-liittimiä, mitkä käyttäjä voi yhdistää suoraan toisiinsa. Google Data Studio tarjoaa yhteyksiä myös eri tietokantapalveluihin kuten tulevassa esimerkissä käytettävään PostgreSQL-tietokantaan. (Google Data Studio 2019.)

Google Data Studiolla on mahdollista tehdä erilaisia kaavioita ja diagrammeja, minkä avulla dataa voidaan visualisoida. Ihmisen aivot ymmärtävät ja prosessoivat paremmin visualisoitua dataa abstraktien lukujen ja arvojen sijaan. Onnistuneesti visualisoitu data on nopeasti ymmärrettävä ja muistettava. Lisäksi sen tulee olla täsmällinen ja asianmukainen eikä jättää kysymyksiä katsojalle. Jos diagrammia toi tulkita monella eri tapaa, on siinä jokin vialla. (TapClicks 2018.)

4 Opinnäytetyössä käytetyt järjestelmät

4.1 Järjestelmät ja niiden yhdistämien

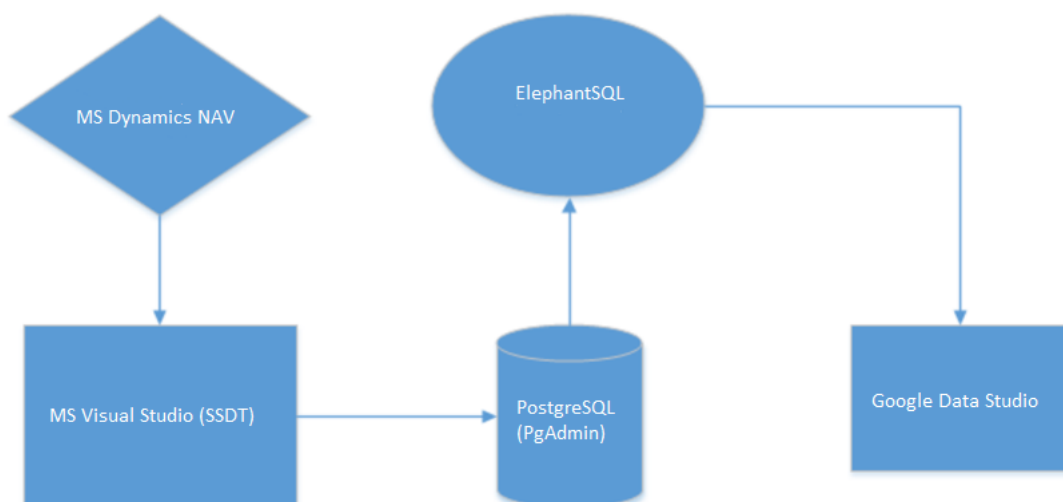
Tietoturvasyistä opinnäytetyön esimerkit pohjautuivat mallidataan, eikä kohdejärjestelmänä toimi yrityksen käytössä oleva Intercom. Ohjelmisto ei tarjoa ilmaiskokeilua, joten sen testaaminen ei opinnäytetyössä ollut mahdollista. Opinnäytetyön esimerkkien datana toimi OData.org:n ja Fixer.io:n tarjoamat JSON-tiedostot sekä Karelia-ammattikorkeakoulun NAV-toiminnanohjausjärjestelmä.

Esimerkeissä hyödynnettiin kolmea eri tapaa automatisoida datan liikkuminen kohteesta lähdejärjestelmään. Microsoft Visual Studiolla yhdistettiin toiminnanohjausjärjestelmä ja tietokanta. UiPath Studiolla rakennettiin ohjelmistorobotiikka-esimerkki ja Apps Script -ohjelmistossa tehtiin ohjelma, joka siirtää dataa Google Sheetsiin. Kaikkien esimerkkien data siirtyy eri tavoilla lopulta Google Data Studioon. Malliesimerkkien tarkoituksena on näyttää erilaisia ratkaisuja datan siirtoon. Järjestelmiä on mahdollista yhdistää eri tavoin, mutta opinnäytetyöhön valikoituivat seuraavat tavat toteuttaa integraatiot.

4.2 Microsoft Visual Studio

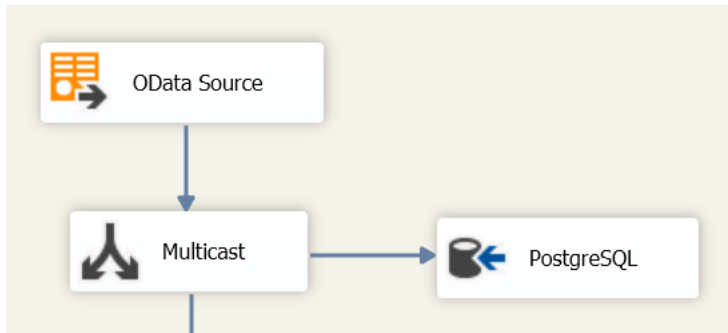
Esimerkissä käytettiin Microsoft Visual Studio 2017:ta, johon ladattiin SQL Server Data Tools -työkalupaketti (SSDT). Työkalupaketti mahdollistaa erilaisten integraatioiden tekemisen. Paketti ei kuitenkaan tarjoa PostgreSQL:n yhdistämistä MS Visual Studioon suoraan, joten järjestelmään ladattiin myös sen yhdistämiseen tarkoitettu ohjelmistopaketti. PostgreSQL tarvitsee Google Data Studioon yhdistykseen oman palvelinyhteyden, koska Google Data Studioon ei voi yhdistää paikallisesti toimivia tietokantoja. Tätä varten otettiin käyttöön palvelinyhteyden tarjoava ElephantSQL. Oman tunnuksen luonnin jälkeen ElephantSQL:n sivustolla luotiin uusi instanssi eli ilmentymä. Esimerkissä käytettiin ilmaista "Turtle" versiota. Henkilötiedoista löytyy oman henkilökohtaisen palvelimen tietokannan tunnukset. (ElephantSQL, 2020; Google G Suite, 2017.)

Kuvassa 2 on kaavio, jossa näkyy, kuinka datan oli tarkoitus liikkua lähdejärjestelmästä kohdejärjestelmään. Esimerkin data tulee Karelia-ammattikorkeakoulun toiminnanohjausjärjestelmästä NAV:sta, jossa on tuotetaulu polkupyöristä ja niiden osista.



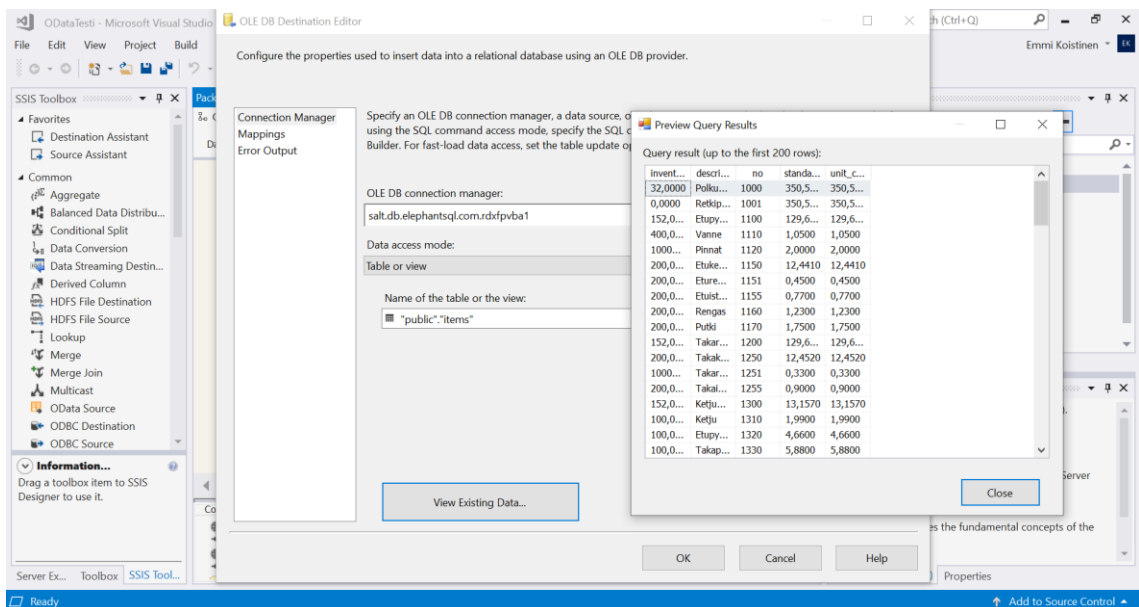
Kuva 2 Kaavio datan siirtämisestä MS Visual Studioon kautta.

Lähdejärjestelmän Dynamics NAV:n rajapintana toimii REST-rajapinta, joten MS Visual Studiossa käytettiin "OData Source" työkalua lähderajapintayhteyden muodostamisessa. (Microsoft 2017.)



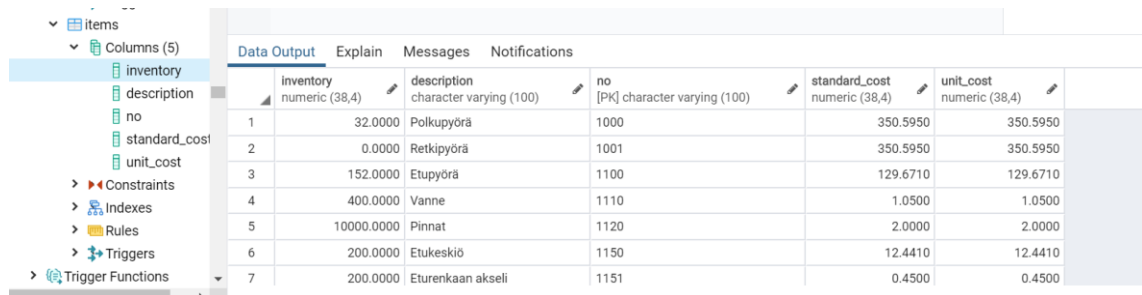
Kuva 3 MS Visual Studiossa datan siirto.

Dynamics NAV (Kuvassa 3 "OData Source") yhdistettiin MS Visual Studioon ja kohdejärjestelmäksi laitettiin PostgreSQL. Molempiin laitettiin tarvittavat todennukset, jotta halutut tiedot siirtyivät oikeaan kohteeseen. Todennuksien lisäksi täytyi valita haluttu taulu, joka siirrettiin kohteeseen. Kun ohjelma ajettiin onnistuneesti, siirtyi taulu ja sen tiedot PostgreSQL:n. Kuvassa 4 näkyy tuotelistan tiedot, jotka siirtyivät PostgreSQL-tietokantaan.



Kuva 4 MS Visual Studio ja tuotetaulu.

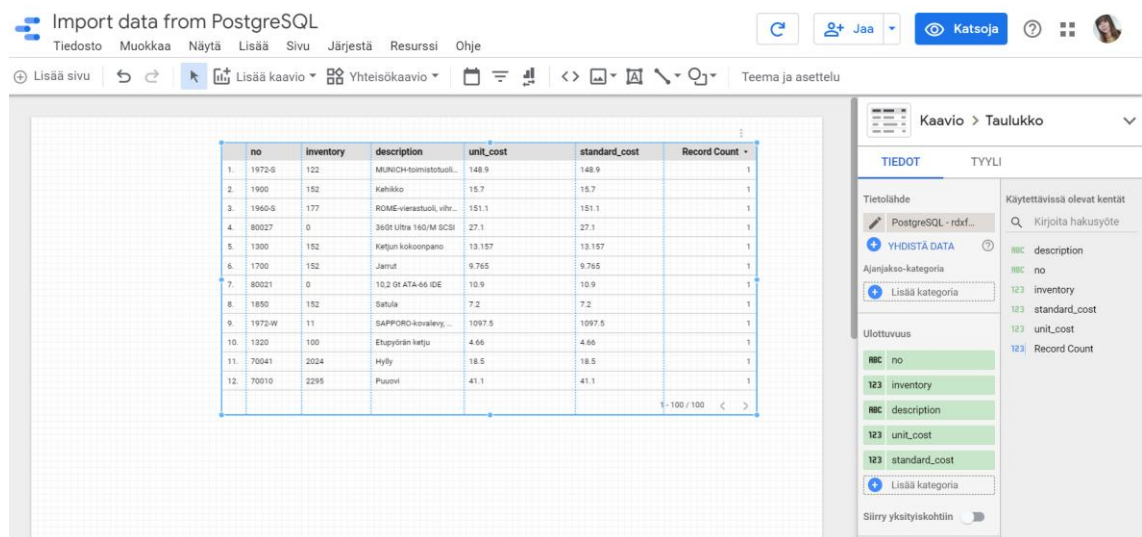
Kun integraatio onnistui NAV:n ja PostgreSQL:n välillä, tuli tuotetaulu näkyviin PgAdminiin, kuten kuvassa 5 näkyy. PgAdmin on työkalu PostgreSQL-tietokantojen käyttöön terminaalin lisäksi. Työkalu tulee PostgreSQL:n asennuksen mukana ja se avautuu selaimen.



	inventory	no	description	standard_cost	unit_cost
1	32.0000	1000	Polkupyörä	350.5950	350.5950
2	0.0000	1001	Retkipyörä	350.5950	350.5950
3	152.0000	1100	Etupyörä	129.6710	129.6710
4	400.0000	1110	Vanne	1.0500	1.0500
5	10000.0000	1120	Pinnat	2.0000	2.0000
6	200.0000	1150	Etuskiö	12.4410	12.4410
7	200.0000	1151	Eturenkaan akseli	0.4500	0.4500

Kuva 5 PgAdmin ja siirtynyt tuotetaulu.

Lopulta PostgreSQL yhdistettiin lopulliseen kohteeseen eli Google Data Studioon sen tarjoaman datalähteen kautta. Datalähteen tekemiseen tarvittavat tiedot löytyvät ElephantSQL:sta, omista henkilötiedoista. Kun yhteys onnistuu, tulee näkyviin omat tietokantataulut, mistä voi valita minkä haluaa yhdistää Google Data Studioon. Kuvassa 6 on lopullinen näkymä siirtyneestä tuotetaulusta.



	no	inventory	description	unit_cost	standard_cost	Record Count
1.	1972-S	122	MUNICH-komistotoulli	148.9	148.9	1
2.	1900	152	Kahikko	15.7	15.7	1
3.	1960-S	177	ROME-vierastuoli, vihr.	151.1	151.1	1
4.	80027	0	360t Ultra 160/M SCSI	27.1	27.1	1
5.	1300	152	Kirjan kokooppa	13.157	13.157	1
6.	1700	152	Jamut	9.765	9.765	1
7.	80021	0	10.2 Gt ATA-66 IDE	10.9	10.9	1
8.	1850	152	Satula	7.2	7.2	1
9.	1972-W	11	SAPPORO-kovalevy	1097.5	1097.5	1
10.	1320	100	Ehopyörän ketju	4.66	4.66	1
11.	70041	2024	Hyly	18.5	18.5	1
12.	70010	2285	Puuväli	41.1	41.1	1

Kuva 6 Google Data Studio ja PostgreSQL yhdistettynä.

Jos käyttää ElephantSQL:n ilmaisversioita, on yhteyksien määrä rajattu viiteen. PostgreSQL:n ja Google Data Studion yhdistämisessä voi mahdollisesti tulla virheviesti, yhteyksien liiallisesta määrästä ("FATAL: Too many connections for role "XX"). Kuvassa 7 on näkymä Google Data Studiosta, missä diagrammit eivät toimi oikein.



Kuva 7 "Käyttäjän määrittelyvirhe".

Ongelmana on, että PostgreSQL ja elephantSQL muodostavat koko ajan automaattisesti uusia yhteyksiä ja sallittu määrä ilmaisversiossa on viisi. Näin ollen virheviesti "Fatal: Too many connections for role "XX"" saattaa tulla, kun yhdistää PostgreSQL:n Google Data Studioon. Poistamalla ylimääräisiä yhteyksiä, saa yhteyden lopulta muodostettua Google Data Studioon. ElephantSQL:n tilastovälilehdeltä löytyy oman palvelimen yhteydet. Kuvassa 8 on esitelty yhteyksien näkymää.

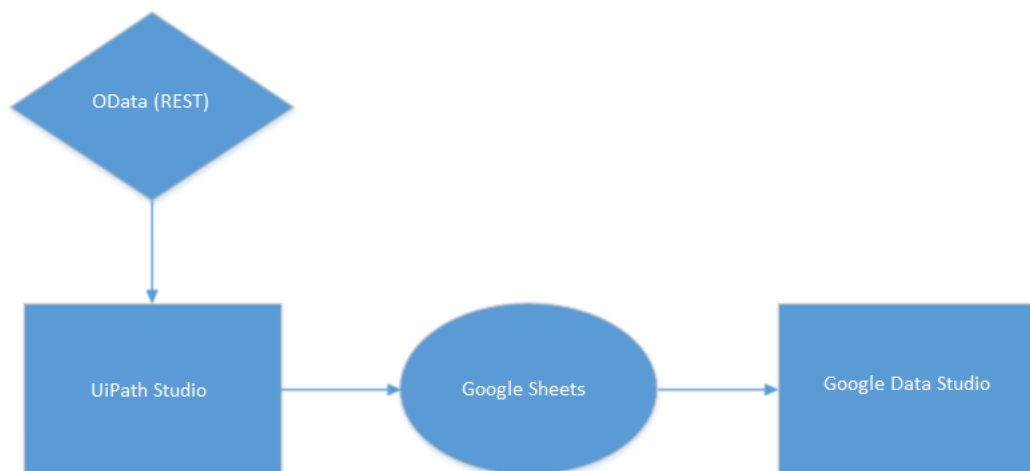
Connections				
3 currently open				
Since	Application name	Client address	State	Query
2019-12-18T11:42:12.852Z	pgAdmin 4 - DB:rdxfpvba	137.163.31.190	idle	/*pga4dash*/ SELECT 'session_stats' AS chart_name, row_to_json(t) AS chart_data FROM (SELECT (SELECT count(*) FROM pg_stat_activity WHERE datname = 'rdxfpvba') AS session_stats FROM pg_stat_activity WHERE datname = 'rdxfpvba')

Kuva 8 ElephantSQL yhteydet.

4.3 UiPath Studio

Järjestelmien integrointia ja automatisointia voidaan toteuttaa myös ohjelmistorobotiikan avulla. UiPath Studio on ohjelmistorobotiikkaan tarkoitettu ohjelmisto, jolla seuraava esimerkki tehtiin.

Roboteille luodaan erilaisia tehtäviä työkalujen avulla. Esimerkissä käytetty OData.org tarjoaa REST-rajapintaa ja haluttu data on JSON-tiedostomuotoinen. Rajapinnan käyttöä varten täytyi asentaa HTTP-integrointipaketti, mikä onnistui UiPath Studiossa. Kuva 9 esittää, kuinka lähdejärjestelmän datan oli tarkoitus siirtyä UiPath Studion kautta kohdejärjestelmään, Google Data Studioon. (UiPath 2019.)

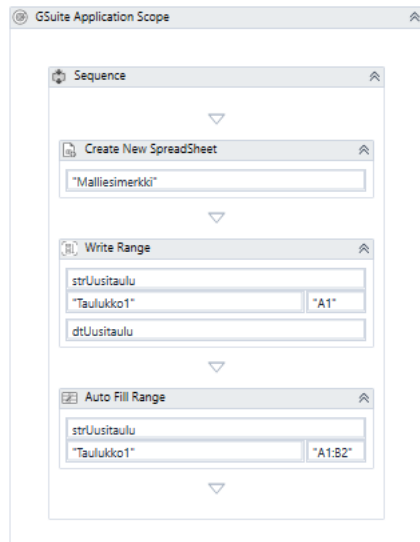


Kuva 9 Kaavio datan siirtämisestä UiPath Studion kautta.

Esimerkissä käytettiin lähteenä OData.org:n tarjoamaa REST APIa. UiPath Studio mahdollistaa rajapintojen käyttämisen erilaisilla todennustavoilla. Rajapinta voi toimia esimerkiksi yksityisellä acces tokenilla tai verkkopalvelun omilla tunnuksilla. (UiPath 2019.)

UiPath Studio mahdollistaa erilaisten työkalupakettien lataamisen ohjelmistorobotin käyttöön. Malliesimerkissä UiPath Studioon ladattiin GSuite-työkalupaketti, jonka avulla robotti pystyi tekemään Google-laskentataulukon. GSuite-paketin asentamisen jälkeen luotiin Google Sheetsiin rajapintayhteys "Google APIs" -sivustolla. Oman rajapinnan luomisen yhteydessä saa oman Client ID:n, jonka avulla yhteys muodostetaan. (Billakurthi 2019.)

Kun Googlerajapinta oli luotu, yhdistettiin se UiPath Studiossa tehtyyn ohjelmistorobottiin. Kuvassa 10 näkyy sovelluksen näkyvyysalue, missä luotiin uusi laskentataulukko ja siirrettiin JSON-tiedoston data sinne. (Billakurthi, 2019.)



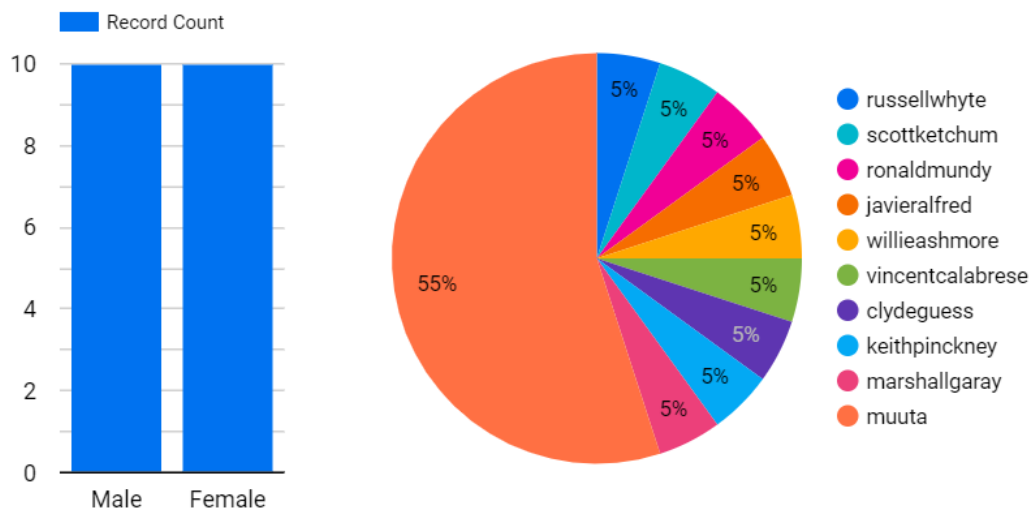
Kuva 10 UiPath Studio Application Scope.

Kuvassa 11 esitellään UiPath Studion ohjelmistorobotin tekemä laskentataulukko nimeltä "Malliesimerkki". JSON-tiedoston tiedot ovat omissa sarakkeissaan. Kyseinen taulukko yhdistettiin Google Data Studioon valmiin liittimen avulla ja kuvassa 12 näkyy taulukon sisältämää tietoa.

The screenshot shows a Google Sheet titled "Malliesimerkki". The data is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L
1	russellwhyte	Russell	Whyte		Male		Russell@example.com	Russell@contoso.c	Feature1	Feature1	Feature2	187 Suff
2	scottketchum	Scott	Ketchum		Male		Scott@example.com		Feature1			2817 Mil
3	ronaldmundy	Ronald	Mundy		Male		Ronald@example.com	Ronald@contoso.c	Feature1			187 Suff
4	javieralfred	Javier	Alfred		Male		Javier@example.com	Javier@contoso.c	Feature1			89 Jeffe
5	willieashmore	Willie	Ashmore		Male				Feature1			
6	vincentcalabrese	Vincent	Calabrese		Male		Vincent@example.com	Vincent@contoso.c	Feature1			55 Grizz
7	clydeguess	Clyde	Guess		Male				Feature1			
8	keithpinckney	Keith	Pinckney		Male		Keith@example.com	Keith@contoso.c	Feature1			55 Grizz
9	marshallgaray	Marshall	Garay		Male		Marshall@example.com	Marshall@contoso.c	Feature1			55 Grizz
10	ryantheriault	Ryan	Theriault		Male		Ryan@example.com	Ryan@contoso.c	Feature1			55 Grizz
11	elainestewart	Elaine	Stewart		Female		Elaine@example.com	Elaine@contoso.c	Feature1			55 Grizz
12	salliesampson	Sallie	Sampson		Female		Sallie@example.com	Sallie@contoso.c	Feature1			87 Polk
13	jonirosales	Joni	Rosales		Female		Joni@example.com	Joni@contoso.c	Feature1			55 Grizz
14	georginabarlow	Georgina	Barlow		Female		Georgina@example.com	Georgina@contoso.c	Feature1			55 Grizz
15	angelhuffman	Angel	Huffman		Female		Angel@example.com		Feature1			55 Grizz
16	laurelosborn	Laurel	Osborn		Female		Laurel@example.com	Laurel@contoso.c	Feature1			87 Polk
17	sandyosborn	Sandy	Osborn		Female		Sandy@example.com	Sandy@contoso.c	Feature1			87 Polk
18	ursulabright	Ursula	Bright		Female		Ursula@example.com	Ursula@contoso.c	Feature1			87 Polk
19	genevieveereves	Genevieve	Reeves		Female		Genevieve@example.com	Genevieve@contoso.c	Feature1			87 Polk

Kuva 11 Google Sheets "Malliesimerkki".



Kuva 12 Google Data Studiossa pylväsdiagrammi ja ympyrädiagrammi.

4.4 Google Apps Script

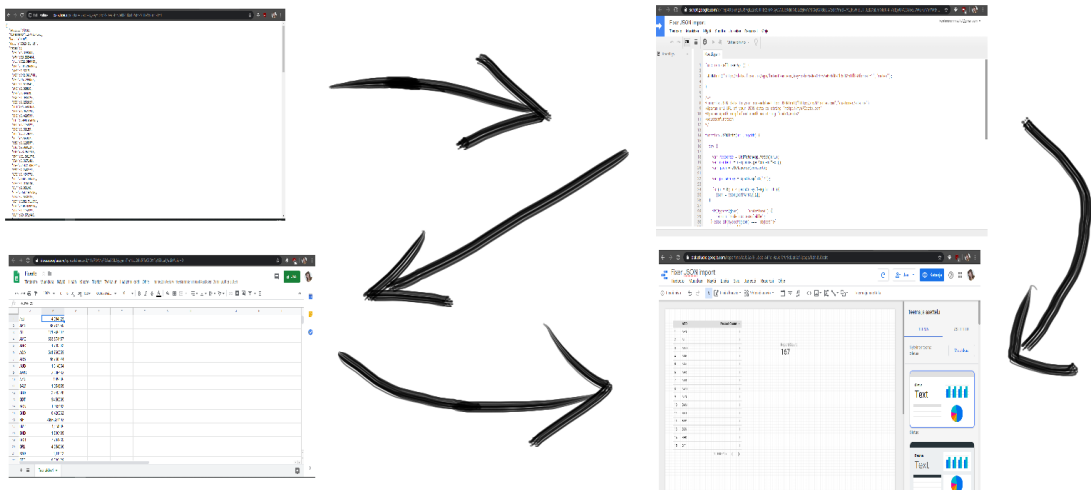
Google Apps Script on omien yhteyksien ja skriptien kirjoittamiseen tehty alusta. Se kuuluu Google G Suite -ohjelmistopakettiin. Esimerkissä luotiin oma yhteys Fixer.io:n tarjoamasta API:sta. Kyseiselle sivustolle tulee kirjautua, jotta saa oman access tokenin. Data on JSON-tiedostomuotoinen ja sisältää listan valuutoista ja niiden arvoista. Liitteestä 1 löytyy koodi, jonka avulla esimerkin data tulostettiin Google-laskentataulukon. Kuvassa 13 näkyy Apps Scriptissä ajettun koodin tulos. Taulukon tiedot päivittyvät valuuttakurssien myötä. Oma access token näkyy "access_key=" jälkeen. (Learn Google Spreadsheet 2017.)

The screenshot shows a Google Spreadsheet titled 'Fixerlo'. The formula bar contains the following code: `=JSONInit("http://data.fixer.io/api/latest?access_key=[redacted]&format=1";"rates")`. The spreadsheet displays the following data:

	A	B	C	D	E	F	G	H
1	AED	4,057271						
2	AFN	86,011704						
3	ALL	121,834108						
4	AMD	528,946878						
5	ANG	1,816879						
6	AOA	547,66788						
7	ARS	66,768135						

Kuva 13 Fixer.io:n JSON-tiedoston valuutat.

Kuvassa 14 esitellään vielä datan siirtymisen vaiheet käyttöliittymäkymässä. Google Apps Scriptin voi myös yhdistää suoraan Google Data Studioon, mutta esimerkissä yhdistettiin se ensin Google Sheetsiin.



Kuva 14 Datan siirtyminen.

5 Tulokset

Opinnäytetyön esimerkkien tarkoituksena oli saada toimivat integraatiot kohdejärjestelmään, Google Data Studioon. Esimerkit osoittivat, että eri ohjelmistojen välinen integroiminen on mahdollista tehdä monella eri tapaa. Omaan käyttöön ottamista varten täytyy huomioida yrityksen tarpeet ja budjetti integraation tekemiseen. Teoriaosuus oli enemmän yrityskäyttöön pohjautuvaa ja sen avulla pystyi toteuttamaan integraatioesimerkit helpommin. Oli tärkeä ymmärtää kokonaiskuva ennen toiminnallista osuutta.

Opinnäytetyön tuloksien tarkoituksena on esitellä järjestelmien integroimisen hyöty ja erilaiset mahdollisuudet toteuttaa se. Esimerkkien ohjelmistot valikoituivat toimeksiantajan tarpeiden pohjalta sekä omissa opinnoissa käytettävistä järjestelmistä. Integraatiota on kuitenkin mahdollista toteuttaa monella muulla eri ohjelmistolla tai tavalla. Nykyiset ohjelmistot tarjoavat monia erilaisia rajapintoja

joita voi hyödyntää. Työn tarkoituksena oli tarjota toimeksiantajalle mahdollisuus tutkia mahdollisuuksia yrityksen datan automatisointiin.

Aineistona oli erilaiset internetlähteet, kirjat sekä kurssit. Osaan internetlähteistä oli suhtauduttava varauksella, sillä niissä saattoi olla puolueellista tekstiä. Puolueelliset tekstit olivat joko liitoksissa opinnäytetyössä käytettyihin ohjelmistoihin tai erilaisiin järjestelmäintegraatiota tarjoaviin palveluihin. Lähdevertailun jälkeen koin lähteet sopiviksi, sillä opinnäytetyössä käytettävät lähdeviittaukset olivat myös yleistietoa mitä voi löytää myös muista lähteistä. Käsitteenä järjestelmäintegraatio ei ole liitoksissa mihinkään tiettyyn palveluun tai palveluntarjoajaan. Opinnäytetyön tarkoituksena ei ole suositella maksullisia ohjelmistoja, vaan näyttää esimerkkejä, joidenka pohjalta voi tutkia erilaisia mahdollisuuksia omaan käyttöön.

Esimerkeissä yksinkertaisin integraatoratkaisu oli Apps Script, sillä se ei vaatinut ylimääräisiä ohjelmistoja ladattavaksi ja toimeksiantaja Isolta Oy käyttää Google Sheetsiä jo valmiiksi. Toteutus sillä on myös ilmainen, joten ylimääräisiä kuluja ylläpidosta ei tule. Samaan taulukkoon on mahdollista tehdä monta omaa funktiota, joilla voi kutsua dataa samasta tai eri lähteistä. Esimerkin koodia tulee hienosäätää, jotta se toimii ongelmitta taulukon päivittämisen yhteydessä. Ohjelmaa ei ole testattu kuin yhdellä lähdejärjestelmällä, joten suuriin ja monimutkaisiin integrointeihin se ei välttämättä ole soveltuvin.

Liitteen 1 koodia pystyy muokkaamaan omien tarpeiden mukaan. Yrityksen käytössä olevan ohjelmiston Intercomin sivustolta löytyy kehittäjille tarkoitettu opastus, josta löytyy apua oman rajapintayhteyden luomisen. Sivustolla on listattu erilaisia malleja mitä esimerkiksi "user" JSON-tiedosto voi mahdollisesti pitää sisällään. Omaa dataa voi ennen integrointia tarkastella esimerkiksi Postman-ohjelmalla. Postman on tarkoitettu muun muassa rajapintayhteyksien testaamiseen.

Microsoft Visual Studio ja siihen asennettu integrointityökalupaketti SSDT takaa monen järjestelmän yhdistämisen. Sen avulla voi luoda käyttäjän näkökulmasta helpommin yhteyksiä kuin esimerkiksi ohjelmoimalla, jos sen osalta kokemusta

ei ole paljoa. Työkalupaketti mahdollistaa tietomuunnoksien tekemisen, jos esimerkiksi lähdejärjestelmän data on kohteeseen nähden väärässä muodossa. Esimerkin tekeminen ohjelmalla oli nopeaa ja data siirtyi ongelmitta PostgreSQL:n. Integroimiseen tarkoitettu työkalu on mahdollisesti toimivin ratkaisu, jos halutaan tehdä monimutkaisempia integraatioita.

PostgreSQL ja ElephantSQL vaativat kuukausimaksun, jos haluaa useamman kuin viisi palvelinyhteyttä. Yhteydet täyttyvät nopeasti, vaikka käytössä olisi vain yhden tietokannan taulu. Tämän myötä ilmaisversio on epävakaata ja yhteyksiä joutuu poistamaan manuaalisesti. Yleisimpiä virheviestejä Google Data Studioon yhdistämisessä ovat ”FATAL: too many connections for role”. Ilmaisversion toteutusta voisi kutsua väliaikaiseksi ratkaisuksi, sillä yhteys ei toimi stabiilisti, vaan aiheuttaa lisätöitä. Maksullinen versio on viisain ratkaisu, jos haluaa käyttää PostgreSQL:ää. Tietokantojen käyttäminen on kuitenkin suotava ja turvallinen tapa datan hallinnassa. Google Data Studio tarjoaa valmiin datalähteen myös MySQL-relaatiotietokantoihin.

UiPath Studio toimii ohjelmistorobotiikkaesimerkissä hyvin, jossa data integroidaan Google Sheetsiin. Harjoitteessa ei käydä läpi robotin tekemää päivitystyötä jatkoa ajatellen, ainoastaan yhden taulukon tekeminen. Ohjelmistorobotiikka helpottaa rutiinityötehtäviä, mutta ennen käyttöön ottamista tulee pohtia maksaako robotiikan käyttöön ottaminen itsensä takaisin yrityksen käytössä. UiPath Studion käyttöön ottaminen ei vaadi ohjelmointitaitoja.

Erilaisten integrointityökalujen ja ohjelmoinnin lisäksi monet yritykset tarjoavat erilaisia integraatiotehtäviä sekä niiden ylläpitoa. Joskus saattaa tulla halvemmaksi ja nopeammaksi ulkoistaa järjestelmäintegraation toteutus ja sen jälkeen keskittyä itse ylläpitotehtäviin. Osa ohjelmistoista tarjoaa myös valmiita mahdollisuuksia datan integroimiseen eri lähteistä sekä mahdollisuuden tehdä ohjelmiston sisällä erilaisia dashboardeja ja analytiikkatyötä. Valmiin kokonaisuuden ylläpitäminen ja ylläpitäminen on helpompaa ja tietoturvalisempää.

Ennen integraatioprosessin käynnistämistä on huomioitava mitkä kaikki ohjelmistot halutaan yhteen, sillä opinnäytetyön esimerkit tehtiin vain kahden järjestelmän

välille. On käytävä läpi mahdolliset vaihtoehdot, jotka ratkaisevat monen lähteen integroimisen yhteen mahdollisimman jouhevasti.

6 Pohdinta

Opinnäytetyön tavoitteena oli tutkia, kuinka datan liikkuminen saadaan automatisoitua. Oli löydettävä ratkaisu, kuinka toimeksiantajan eri järjestelmät saadaan toimimaan yhteen ilman rutiininomaisia käsityövaiheita. Aihetta tutkiessa vastaan tuli erilaisia ohjeita, kuinka dataa voidaan integroida eri kohteista toisiin järjestelmiin. Järjestelmäintegraatio käsitteenä oli minulle uusi, mutta mitä enemmän sitä tutki, sitä enemmän se alkoi vastaamaan mielessä pyöriviin kysymyksiin.

Aloitin syksyllä 2019 järjestelmäintegraatiokurssin, joka helpotti opinnäytetyön tekemistä. Ymmärrys aihetta kohtaan kasvoi ja pystyin tekemään käytännönharjoitteita erilaisten integrointitehtävien parissa. Tutustuin erilaisiin ohjelmistoihin ja integrointitapoihin, mitä kävin opinnäytetyössäni läpi niin teoriassa kuin toiminnallisessa osuudessa. Kurssilla käytiin myös ohjelmistorobotiikan perusteita läpi ja pääsin käyttämään UiPath Studiota. Sen myötä pystyin tekemään opinnäytetyöhön yhden malliesimerkin aiheeseen liittyen.

Kurssin ja erilaisten lähteiden lukemisen lisäksi hyödynsin oppimisessa videoita, sillä niiden avulla oppii myös paljon. Kävin myös opinnäytetyöhöni liittyen ilmaisen Apps Script -kurssin, joka herätti lisäkiinnostusta perehtyä lisää sen hyödyntämiseen.

Aluksi opinnäytetyön aihe tuntui haarautuvan joka suuntaan, sillä mieleen tuli erilaisia ideoita mitä haluan työssäni käydä läpi. Lopulta rajasin teoriaosuuden aiheen pelkästään järjestelmäintegraation ja ohjelmistorobotiikan perusteisiin. Karasin ylimääräisiä ideoita, jotta sain enemmän aikaa tehdä toiminnallista osuutta, johon lopulta paloi odotettua enemmän aikaa. Huomasin myös, että mitä selkeämmäksi aihe rajautui, sitä helpompi oli löytää lähteitä, joita tutkia.

Toiminnallisen opinnäytetyön yksi haasteista oli tasapainotella kirjoittamisen ja harjoitusten tekemisen välillä. Jälkeenpäin ajatellen on ollut opettavaista tehdä erilaisia virheitä ja kohdata haasteita, sillä niistä oppi hyvin. Myös aiheen yksinkertaistaminen tuotti omat haasteensa, sillä järjestelmäintegraatio pitää sisällään monia erilaisia vaiheita ja tietoja, mitä olisi voinut käydä läpi tarkemmin. Pyrin kuitenkin pysymään mahdollisimman lähellä toimeksiantajan tilannetta enkä käydä läpi kaikkia osa-alueita järjestelmäintegraatioon liittyen.

Tietoturvasyistä en käyttänyt yrityksen omaa dataa esimerkeissä, joten opinnäytetyön esimerkkien jatkokehitys voi mahdollisesti tapahtua yrityksen sisällä. Esimerkkejä voi hyötykäyttää ja jatkojalostaa myös omissa projekteissa ja muiden integraatioiden parissa. Aihetta opiskelleena olen ymmärtänyt kuinka jo pienet järjestelmäintegraatoratkaisut voivat hyödyntää yrityksen toimintaa. Myös jatkuvat uutiset siitä, kuinka yritykset hukkuvat datan määrään, nostattavat halua jatkaa opiskelua aiheen parissa. Data on arvokasta, joten sitä tulee ymmärtää ja käsitellä oikein.

Lähteet

- Aihkisalo, T & Halén, M & Holmström, H & Jurmu, P & Kääriäinen, J & Matinmikko, T & Seppälä, T & Tihinen, M & Tirronen J. 2018. Suomi: Valtioneuvoston kanslia. <http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/161123/65-2018-Ohjelmistorobotiikka%20ja%20tekoaly.pdf>.
- Artsoft. 2017. Mitä integraatio on ja miten se tehdään hyvin? <https://www.atrsoft.com/2017/11/14/mita-integraatio-on-ja-miten-se-tehdaan-hyvin/>. 13.1.2020.
- Billakurthi, V. 2019. UiPath Googlesuite gSuite Demo #vajrang talks.[video]. <https://www.youtube.com/watch?v=BnvhFOwKtOA>. 27.11.2019.
- ElephantSQL. 2019. Documentation. <https://www.elephantsql.com/docs/index.html>. 15.12.2019.
- Flashnode Oy. Itewiki. Integraatio-opas. <https://www.itewiki.fi/opas/integraatiot/>. 3.10.2019.
- Google G Suite. 2019. <https://gsuite.google.com/features/>. 3.10.2019.
- Google. 2019. Marketing Platform. <https://marketingplatform.google.com/about/analytics/>. 3.10.2019.
- Google Data Studio. 2019. <https://datastudio.google.com>. 3.10.2019.
- Haikonen, M. 2016. Tivi. Osa 2: Ohjelmistorobotiikka – mitä se on? <https://www.linkedin.com/pulse/osa-2-ohjelmistorobotiikkamist%C3%A4-siin%C3%A4-kyse-mika-haikonen.10.1.2020>.
- Hohpe, G. 2003. Enterprise integration patterns. <https://www.enterpriseintegrationpatterns.com>. 14.10.2019.
- Intercom Developer Hub. 2019. REST API. <https://developers.intercom.com/building-apps/docs/rest-apis>. 3.2.2020.
- Isolta Oy. 2020. <https://www.isolta.fi/>. 7.2.2020
- JUHTA. 2017. JHS 179 Kokonaisarkkitehtuurin suunnittelu ja kehittäminen. http://docs.jhs-suositukset.fi/jhs-suositukset/JHS179_liite8/JHS179_liite8.html. 17.12.2019.
- Kauhanen, A. 2016. Eva raportti. Robotit töihin – Uusi Työnjako. Taloustieto Oy. Helsinki: Nextprint Oy. <https://www.eva.fi/wp-content/uploads/2016/09/Robotit-t%c3%b6ihin.pdf>.
- Kolehmainen, A. 2016. Tivi. Ohjelmistorobotit mullistavat työelämän – ”Tulee vastaava taito kuin Excelistä”. <https://www.tivi.fi/uutiset/ohjelmistorobotit-mullistavat-tyoelaman-tulee-vastaava-taito-kuin-excelista/bdfeaa73-c8f7-3c29-b792-cd51a49e5d71>. 7.1.2020.
- Korhonen, J. 2003. Integroinnista tuli orkestrointia. <https://www.tivi.fi/uutiset/integroinnista-tuli-orkestrointia/61befc37-1b2c-39b4-922e-a2ecbd20448e>. 24.1.2020.
- W3schools. 2019. What is JSON? https://www.w3schools.com/whatis/whatis_json.asp. 17.10.2019.
- Learn Google Spreadsheet. 2017. Apps Script UrlFetchApp API, Get JSON data, Build Google Sheets Function, Advanced Tutorial.[video]. <https://www.youtube.com/watch?v=k0su6345KDI>. 14.1.2020.
- Manouvrier, B. & Menard, L. 2008. Application Integartion: EAI B2B BPM

- and SOA. New Jersey: John Wiley & Sons.
- Microsoft. 2017. Dynamics NAV Web Services overview.
<https://docs.microsoft.com/en-us/dynamics-nav/microsoft-dynamics-nav-web-services-overview>. 27.1.2020.
- Microsoft. 2018. SQL Server Integration Services.
<https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver15>. 29.1.2020.
- Mixpanel. 2020. Developer Documentation.
<https://developer.mixpanel.com/docs>. 6.2.2020.
- Perttilä, T. 2017. Robotic Process Automation - lyhyt oppimäärä.
<https://timopertila.com/2017/01/19/robot-process-automation-lyhyt-oppimaara/>. 12.12.2019.
- Pietikäinen, S. 2016. Organisaation toimintaympäristö. <https://www.vah-tiohje.fi/web/guest/4-organisaation-toimintaymparisto>. 7.1.2020.
- Rouse, M. 2019. What is RESTful API.
<https://searchapparchitecture.techtarget.com/definition/RESTful-API>. 16.10.2019.
- Rousku, K. 2014. Kyberturvaopas, Tietoturvaa kotona ja työpaikalla. Viro: Talentum.
- TapClicks. Innovative data visualization.
<https://www.tapclicks.com/innovative-data-visualization-examples/>. 10.06.2019.
- Tähtinen, S. 2005. Järjestelmäintegraatio. Helsinki: Talentum Media Oy.
- UiPath. 2020. Introduction.
<https://docs.uipath.com/activities/docs/>. 29.1.2020.
- Vaala, S. 2017. Datan visualisointi – viisi vinkkiä parempaan tarinan kerrontaan. <https://quru-analytics.com/fi/blogi-uutiset/datan-visualisointi-viisi-vinkkia-parempaan-tarinankerrontaan-2/>. 13.1.2020.
- Väre, T. 2019. Master Data, Helsinki: Alma Talent.

```
// Import data to Google Sheets

function callFixerApi() {
  JSONInit("http://data.fixer.io/api/latest?access_key=XXXXXXXXXXXX&format=1","rates");
}

function JSONInit(url, xpath) {

  try {

    var response = UrlFetchApp.fetch(url);
    var content = response.getContentText();
    var json = JSON.parse(content);

    var pathArray = xpath.split("/");

    for(i = 0; i < pathArray.length; i++){
      json = json[pathArray[i]];
    }

    if(typeof(json) === "undefined") {
      return "node not available";
    } else if(typeof(json) === "object"){
      var tempArr = [];

      for(var obj in json){
        tempArr.push([obj,json[obj]]);
      }
      return tempArr;

    } else if(typeof(json) !== "object"){
      return json;
    }

  }

  catch(err) {
    return "Error getting data";
  }
}
```