



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Riku Reiman

Selvitys verkkokauppa-alustojen integroitavuudesta 3PL-yrityksen iPaaS-väliohjelmistoon

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

11.12.2019

Tekijä Otsikko Sivumäärä Aika	Riku Reiman Selvitys verkkokauppa-alustojen integroitavuudesta 3PL-yrityksen iPaaS-väliohjelmistoon 57 sivua + 1 liite 11.12.2019
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	sähkö- ja automaatiotekniikka
Ammatillinen pääaine	automaatiotekniikka
Ohjaajat	lehtori Timo Tuominen kehityssuunnittelija Maarit Korhola
<p>Insinööriä oli luonteeltaan selvitys- ja tutkimustyö, jonka toimeksiantajana oli HUB logistics Finland Oy. Tarkoituksena oli selvittää Suomessa viiden yleisimmän verkkokauppa-alustan integroitavuutta kolmannen osapuolen logistiikkayrityksen (3PL) käyttämään väliohjelmistoon. Tavoitteena oli selvittää verkkokauppojen integraatioiden mahdollisia ongelmia. Väliohjelmisto oli pilvipohjainen iPaaS-alusta. Selvitystyöhön valikoitui seuraavat verkkokauppaohjelmistot: WooCommerce, MyCashflow, Magento, ePages/Vilkas ja Shopify.</p> <p>Hyvän pohjatiedon saamiseksi insinööriydessä perehdyttiin web-ohjelmistojen, joita myös verkkokaupat ovat, integraatioiden perusteisiin, eri rajapintatyyppeihin ja muihin näihin liittyviin tekniikoihin. Insinööriydessä tarkasteltiin HUB logisticsin ja verkkokauppojen välisiä prosesseja, ja näiden prosessien tarpeita integraation ja verkkokauppojen ominaisuuksien suhteen. Tämä jälkeen tutkittiin verkkokauppa-alustojen rajapintoja ja ominaisuuksia peilaten niitä näiden prosessien tarpeisiin. Työhön kuului myös väliohjelmiston kipupisteiden tarkastelu.</p> <p>Selvitystyön tuloksina saatiin tietoa integraatioiden mahdollisista ongelmakohdista. Verkkokauppa-alustojen ominaisuuksissa ja rajapinnoissa ilmeni joitain puutteita. Suurimmat puutteet liittyivät verkkokauppojen kassalla noutopisteitä tarjoavien lisäosien toimintaan. Yhden verkkokaupan rajapinnan puute liittyi noutopisteen tietojen hakuun. Toisen verkkokauppa-alustan ominaisuuden puute oli, ettei verkkokauppaan voi lisätä itsemääritettyjä tietokenttiä. Väliohjelmiston suhteen ei ilmennyt puutteita.</p>	
Avainsanat	verkkokauppa-alusta, integraatio, automaattivarasto, REST API

Author Title	Riku Reiman E-commerce Platforms' Integrability with 3PL Company's iPaaS Middleware
Number of Pages Date	57 pages + 1 appendix 11 December 2019
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Automation Technology
Instructors	Timo Tuominen, Senior Lecturer Maarit Korhola, Development Planner
<p>The thesis study was carried out as a research and study project. It was commissioned by HUB logistics Finland Oy. The purpose of the study was to find out how integrable five most common e-commerce platforms in Finland are with 3PL company's iPaaS middleware. The study aimed to expose potential problems with the integration between e-commerce platforms and the iPaaS middleware. Following e-commerce platforms were chosen for the research: WooCommerce, MyCashflow, Magento, ePages/Vilkas and Shopify.</p> <p>To get a good understanding of the subjects, integration of web software, such as e-commerce platforms are, different Web API styles, and technology related to these were studied. After that, different data interchange processes between HUB logistics and e-commerce platforms were examined. The processes were examined, keeping in mind the needs of the integration. Also the functionalities of e-commerce platforms were examined, considering the needs of the processes in the integration. Lastly, iPaaS middleware was explored to find potential problems.</p> <p>The results of the project help to understand the possible problems that might occur when integrating e-commerce platforms to 3PL company's middleware. Some deficiencies were noticed in the e-commerce functionalities and APIs. Most deficiencies were related to the e-commerce addons, which offer customers pick-up points at the web shop's checkout. One e-commerce platform's deficiency was a lack of capability to retrieve the pick-up point's information through the API. Another deficiency was in e-commerce platform's functionality. It was not possible to add custom information fields to orders, products and customers. No deficiencies were found with the iPaaS middleware.</p>	
Keywords	E-commerce platform, integration, automated warehouse, REST API

Sisällys

Lyhenteet ja käsitteet

1	Johdanto	1
1.1	Insinööriyön tausta	1
1.2	Toimeksiantaja	2
1.3	HUBin järjestelmien lyhyt kuvaus	3
1.4	Insinööriyön tavoitteet ja raja	5
1.5	Insinööriyön toteutus	6
2	Automaattivarasto	6
3	Verkkokaupankäynnin kasvu ja tietoa verkkokaupoista	9
3.1	Verkkokaupankäynnin kasvu viime vuosina ja merkitys 3PL-yritykselle	9
3.2	Yleisimmät verkkokauppa-alustat vuosina 2017–2019	12
3.3	Avoin ja suljettu lähdekoodi	13
3.4	WooCommerce/WordPress	14
3.5	MyCashflow	15
3.6	Magento	16
3.7	ePages/Vilkas	17
3.8	Shopify	17
3.9	Yhteenvedo verkkokauppa-alustoista	18
4	Integraatioihin liittyvä tekniikka	18
4.1	Integraatio käsitteenä	18
4.2	Web-rajapinta	20
4.3	Web Service	21
4.4	HTTP-protokolla	22
4.5	REST-arkkitehtuurityyli	25
4.6	REST-rajapinta	29
4.7	SOAP-rajapinta	31
4.8	GraphQL-rajapinta	33
5	Verkkokauppa-alustojen integraatioiden selvitystyö	34

5.1	Verkkokaupasta tehdyn tilauksen käsittelyprosessi	35
5.2	Verkkokauppoihin liittyvät muut prosessit	36
5.3	Varastonhallintajärjestelmän tarvitsemat tiedot	37
5.4	Automaattivaraston tarvitsemat tiedot	38
5.5	Verkkokauppa-alustojen tarpeelliset ominaisuudet	39
5.6	Noutopiste-sovellus	40
5.7	Verkkokauppa-alustojen rajapinnat	42
5.7.1	WooCommerce ja WordPress	43
5.7.2	MyCashflow	44
5.7.3	Magento 2	44
5.7.4	ePages / Vilkas Suite	45
5.7.5	Shopify	45
5.7.6	Yhteenveto verkkokauppa-alustojen rajapinnoista	46
5.8	iPaaS-väliohjelmistot	46
5.8.1	HUBin valitsema iPaaS-alusta	47
5.8.2	Väliohjelmiston käytön kipupisteet	49
6	Yhteenveto	50
	Lähteet	53
	Liitteet	
	Liite 1. Hyödyllisiä linkkejä verkkokauppa-alustoihin ja niiden integraatioihin liittyen	

Lyhenteet ja käsitteet

3PL	<i>Third Party Logistics</i> . Kolmannen osapuolen logistiikkapalvelu/-yritys.
API	<i>Application Programming Interface</i> . Ohjelmointirajapinta on ohjelman rajapinta, jonka avulla ohjelma pystyy kommunikoimaan toisen ohjelman kanssa.
Avoin lähdekoodi	Avoimen lähdekoodin ohjelmisto on avoimesti kenen tahansa nähtävissä, käytettävissä ja muokattavissa,
GraphQL API	Rajapinta, joka käyttää GraphQL-kyselykieltä.
HATEOAS	<i>Hypermedia as the Engine of Application State</i> . Yksi REST-arkkitehtuurityylin rajoitteista, joka määrittää hypermedian käytön yhdistämään resursseja toisiinsa.
HTTP	Webin käyttämä tiedonsiirtoprotokolla.
iPaaS	<i>Integration Platform as a Service</i> . Pilvipohjainen integraatioalusta/-palvelu.
Representaatio	Tiedon eli resurssin esitys REST-arkkitehtuurityylissä.
REST	<i>Representational State Transfer</i> . Hajautettujen hypermediajärjestelmien arkkitehtuurityyli, jonka Roy Fielding määritteli vuonna 2000 tohtorin väitöskirjassaan.
REST API	REST-tyyliä noudatteleva rajapinta.
Resurssi	Tieto-objekti REST-tyylissä.
SOAP	<i>Simple Object Access Protocol</i> . Tiedonsiirtoprotokolla, joka on suunniteltu hajautettujen järjestelmien tiedonvaihtoon.

SOAP API SOAP-protokollaa noudattava rajapinta.

Suljettu lähdekoodi

Suljetun lähdekoodin ohjelmisto on kaupallinen ohjelmisto, jonka lisenssi ei salli ohjelmiston muokkaamista ja levittämistä.

TMS *Transportation Management System*. Kuljetushallintajärjestelmä, esimerkiksi Unifaun TMS.

Unifaun PickUpLocator

Unifauniin tarjoama Web Service -palvelu noutopisteiden hakemiseksi.

Web *WWW* eli *World Wide Web*. Hajautettu hypertekstijärjestelmä eli toisin sanoen verkkosivujen kokoelma internetissä.

Web API Rajapinta, joka on käytettävissä webissä internetin kautta.

Web Service

SOAP rajapinta, joka tukee koneiden välistä automatisoitua tiedonsiirtoa ja on saatavissa webissä internetin kautta.

WMS *Warehouse Management System*. Varastohallintajärjestelmä/-ohjelmisto.

WSDL *Web Service Description Language*. XML-kielinen dokumentti, joka kuvaa Web Service -rajapinnan koneluettavalla tavalla.

1 Johdanto

Haluan kiittää HUB logistics -konsernia avoimesta asenteesta opiskelijoita kohtaan ja halusta tehdä yhteistyötä opiskelijoiden kanssa. Kiitos kuuluu myös yritysohjaajalleni Maarit Korholalle hänen antamastaan tuesta sekä avusta insinööriyön aiheen valinnassa.

1.1 Insinööriyön tausta

Insinööriyön idea syntyi työharjoitteluni loppupuolella HUB logistics Finland Oy:ssä. HUB logistics -konserni tunnetaan työntekijöiden ja asiakkaiden keskuudessa tuttavallisemmin HUBina. Viitataan yritykseen jatkossa tällä nimellä. Työharjoittelussani pääsin ensikosketuksiin integraatioiden kanssa. Integraatiot eivät ole kovin tuttu asia minulle, eikä varmaankaan kaikille automaatioinsinööriopiskelijoille ainakaan kovin syvällisellä tasolla, vaikkakin integraatiot liittyvät nykypäivänä lähes kaikkiin yritysten ohjelmistoihin ja järjestelmiin. Siksi integraatioihin liittyvä aihe tuntui sopivalta insinööriyöksi.

Perimmäinen syy insinööriyölle on ennalta selvittää yleisimpien verkkokauppa-alustojen integraatiovalmiutta HUBin käyttämään väliohjelmistoon ja siten HUBin varastonhallintajärjestelmään (WMS) ja automaattivarastoon. Tämä selvitystyö on ennakkointia tulevaisuudessa lisääntyviin verkkokauppoihin, joilla on tarve ulkoistaa myyviensä tuotteiden varastointi ja logistiikka. HUB ja HUBin automaattivarasto ovat sopiva ratkaisu verkkokauppojen logistiikan hoitoon.

HUBilla on Vantaalla Hakkilan logistiikkakeskuksessa automaattivarasto, jossa on jopa 400 000 varastopaikkaa. Automaattivarasto tarjoaa nopean ja virheettömän, jopa vain kolmen tunnin, palvelutason tuotteiden keräilylle ja pakkaamiselle. HUBin automaattivarasto on siten täydellinen ratkaisu verkkokauppojen tuotteille, jotka halutaan nopeasti toimittaa asiakkaille. Automaattivarastosta kerron lisää luvussa 2.

1.2 Toimeksiantaja

Insinööriyön toimeksiantajana oli HUB logistics Finland Oy, joka kuuluu HUB logistics -konserniin. HUB logistics -konserni on suomalainen 3PL-logistiikkayritys (Third Party Logistics) eli kolmannen osapuolen logistiikkayritys. Yrityksen palveluihin lukeutuu logistiikan ja toimitusketjun ulkoistukset, sisälogistiikan ratkaisut, Hakkilan automaattivarastopalvelu, logistiikan konsultointi, puupakkausten tuotanto ja pakkauspalvelut, sekä logistiikan lisäarvopalvelut. Logistiikan ulkoistuksia HUB toteuttaa sekä asiakkaiden että HUBin omissa tiloissa. [Yrityksen perustiedot 2019.]

HUB on kasvuyritys, joka pyrkii olemaan teknologisesti logistiikka-alan huipulla. Tästä osoituksena ovat esimerkiksi automatisoidut prosessit, Hakkilan automaattivarasto ja käynnissä oleva automaattitrukin hankintaprojekti. HUBin toiminnan arvoja ovat aktiivinen kehittäminen, ote toiminnasta, arvostus, vastuun kantaminen kaikissa tilanteissa ja ”win-win-win”-tunnuslause. ”Win-win-win”-tunnuslause tarkoittaa sitä, että toteuttamalla edellä mainittuja arvoja kaikki voittavat: HUB, HUBin asiakas sekä loppuasiakas. HUBin missiona on muuttaa logistiikkaa uusilla palveluinnovaatioilla. [HUB logistics 2019.]

HUBin juuret juontavat vuoteen 1992, jolloin Aarno Törmälä perusti osakeyhtiön tarjoamaan laatujärjestelmien koulutusta ja konsultointia logistiikka-alan yrityksille. Tuo asema antoi hyvän näköalapaikan logistiikkaan Suomessa. Ajatus logistiikan ulkoistamisesta palveluyritykselle syntyi Suomen toipuessa lamavuosista. HUB-nimi otettiin käyttöön vuonna 1997. Englanninkielinen sana ”hub” tarkoittaa keskiötä, keskusta ja napaa, jotka kuvaavat HUBin toimintaa. HUB logistics Oy toimii materiaalivirtojen ja tiedonhallinnan logistisena keskuksena. [HUB logistics 2019.]

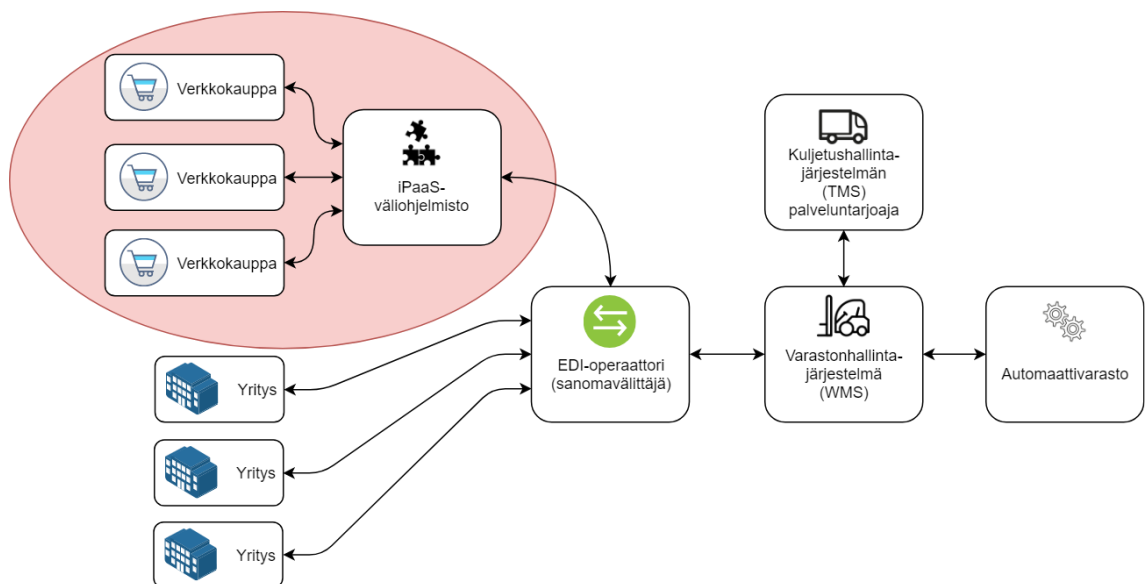
HUBilla on toimintaa yhteensä 15 toimipisteessä Suomessa ja Puolassa. Kuvassa 1 on HUBin Hakkilan logistiikkakeskus, joka toimii yrityksen pääkonttorina. Työntekijöitä on noin 650. Liikevaihto vuonna 2018 oli noin 50 M€. [Yrityksen perustiedot 2019.]



Kuva 1. HUB logistics -konsernin Hakkilan logistiikkakeskus. [Mansukoski 2017.]

1.3 HUBin järjestelmien lyhyt kuvaus

Avaan hieman HUBin järjestelmien kokonaisuutta ajatellen verkkokaupan integraatiota siihen. Tämä auttaa ymmärtämään insinööriyön osuutta ja rajausta. Kuvassa 2 on kuvattu järjestelmät ja niiden väliset yhteydet, joita tarvitaan verkkokaupasta tehdyn tilauksen toimittamiseksi. Kuvassa punaisella ellipsillä on merkitty osuus, jonka integraatioiden tarkastelu kuuluu insinööriyöhön.



Kuva 2. Kaavio HUBin käyttämistä tietojärjestelmistä ja niiden yhteyksistä. Vain punaisella ellipsillä merkityn alueen integraation tarkastelu kuuluu tähän insinööriyöhön.

HUBin omien järjestelmien sydämenä toimii ruotsalaisen Consafe Logisticsin toimittama ASTRO WMS -varastohallintajärjestelmä. Varastohallintajärjestelmä (WMS, Warehouse Management System) toimii kuvainnollisesti varaston aivoina. Järjestelmä ohjaa, aikatauluttaa ja optimoi varaston toimintoja, kuten vastaanottoa, hyllyttämistä, keräilyä ja pakkaamista. Toimiva varastohallintajärjestelmä tehostaa toimintaa muun muassa vähentämällä tyhjällä trukilla ajamista ja vähentämällä keräilyvirheitä. Varastohallintajärjestelmän tietojen avulla voidaan seurata varaston tehokkuutta erilaisten KPI-mittareiden avulla. [Consafe Logistics WMS 2019.]

Varastohallintajärjestelmään on suoraan integroitu saksalaisen KNAPP AG:n toimitama automaattivarasto oman ohjelmistonsa avulla. Automaattivarastosta kerron lisää luvussa 2.

Varastohallintajärjestelmästä on myös suora integraatio pilvipohjaiseen EDI-operaattoriin (Electronic Data Interchange), joka toimii sanomavälittäjänä. EDI-operaattori välittää sanomaliikenteen asiakkailta ja valvoo sitä. EDI-operaattori voi tehdä sanomamuunnoksia eri tiedostomuotojen välillä sekä rikastuttaa sanomia, eli lisätä sanomiin erilaista tietoa, esimerkiksi asiakkaisiin liittyen. EDI-operaattorin kautta kulkee kaikki HUBin automaattivaraston käsittelemät tilaukset. EDI-operaattoriin on integroitu eräs väliohjelmisto verkkokauppojen hallintaa varten. HUBin käyttämä väliohjelmisto on pilvipohjainen iPaaS-alusta, joka on suunniteltu nimenomaan verkkokauppojen yhdistämiseen logistiikan tarjoajalle. Insinööritö liittyvät tämän väliohjelmiston ja verkkokauppojen integraatioon selvittämällä verkkokauppa-alustojen rajapintoja ja ominaisuuksia. Työtä ei voida täysin rajata vain verkkokauppa-alustojen ja väliohjelmiston välille, sillä verkkokauppa-alustojen tutkimisessa pitää huomioida myös HUBin muiden tietojärjestelmien vaatimat ominaisuudet.

Varastohallintajärjestelmään on edellä mainittujen järjestelmien lisäksi integroitu pilvipohjainen kuljetustilausjärjestelmän palveluntarjoaja (TMS, Transportation Management System), joka hoitaa lähetysten kuljetustilaukset kuljetusliikkeille sekä välittää seuranta-koodit ja kollilaput varastohallintajärjestelmään.

1.4 Insinööriyön tavoitteet ja rajaus

Insinööriyö on luonteeltaan kvalitatiivinen selvitys- ja tutkimustyö. Tavoitteena on selvittää, kuinka joustavasti HUBin käyttämään väliohjelmistoon voidaan integroida uusia HUBin asiakkaiden verkkokauppoja, ja sitä myöten yhdistää nämä verkkokaupat HUBin varastohallintajärjestelmään ja automaattivarastoon. Lisäksi tavoitteena on kartoittaa, onko integraatioissa ja väliohjelmiston käytössä kipupisteitä. Selvitystyö keskittyy verkkokauppa-alustojen ominaisuuksiin ja toimintaan HUBin automatisoitujen prosessien kannalta.

Edellä mainituista tavoitteista voidaan muodostaa seuraavia tutkimuskysymyksiä:

- Löytyykö verkkokauppa-alustoista riittävät rajapinnat integraatioihin?
- Mikä on HUBin käyttämän väliohjelmiston valmius integraatioihin verkkokauppa-alustojen kanssa?
- Löytyykö verkkokauppa-alustoista riittävät tietokentät, joita tarvitaan tilausten käsittelyyn?
- Onko verkkokauppa-alustoissa riittävät toiminnallisuudet toimimaan automatisoidusti integroituna varastohallintajärjestelmään?

Insinööriyön tavoitteena on selvitystyön lisäksi perehdyttää kirjoittaja ja lukija verkkokauppojen sekä muiden verkossa olevien palveluiden integraatioiden perusteisiin ja käsitteisiin. Insinööriyössä tutustutaan myös yleisemmällä tasolla suosituimpiin verkkokauppa-alustoihin ja niiden ominaisuuksiin.

Insinööriyön osuus rajataan verkkokauppa-alustojen ja väliohjelmiston välisen integraation, verkkokauppa-alustojen rajapintojen ja ominaisuuksien sekä väliohjelmiston ominaisuuksien tutkimiseen. Tarkasteluun otetaan Paytrailin Suuren verkkokauppa-alustaraportin [2019: 4] viisi yleisintä verkkokauppa-alustaa Suomessa. Insinööriyöhön ei kuulu väliohjelmistosta Hubille päin olevien integraatioiden tarkastelu, mutta siihen kuuluu verkkokauppatilausten käsittelyprosessin kannalta oleellisten ominaisuuksien tutkiminen HUBin koko järjestelmässä. Kuvan 2 kaaviossa luvussa 1.3 näkyy insinööriyön rajaus punaisella ellipsillä merkittynä.

1.5 Insinööriyön toteutus

Selvitystyö aloitetaan tutustumalla verkkokaupankäyntiin ja selvittämällä yleisimmät verkkokauppa-alustat Suomessa. Verkkokauppa-alustoista valitaan viisi suosituinta selvitystyön tarkasteluun. Verkkokauppa-alustojen valinta löytyy luvusta 3.2. Hyvän pohjatiedon saamiseksi työhön kuuluu perehtymistä verkkokauppojen, sekä muiden verkossa olevien palveluiden, integraatioiden liittyvän tekniikan teoriaan. Teoriaan kuuluu muun muassa seuraavia käsitteitä: Web Service, API sekä REST-, SOAP- ja GraphQL-rajapinnat. Näitä käyn läpi luvussa 4.

Seuraavaksi työssä tarkastellaan verkkokauppojen ja HUBin järjestelmien välisiä prosesseja ja niiden tarpeita integraatiolle. Tämän jälkeen tutkitaan verkkokauppa-alustojen ominaisuuksia peilaten niitä näihin prosesseihin. Työssä myös tutustutaan verkkokauppa-alustojen rajapintoihin. Rajapinnoista löytyy parhaiten tietoa kunkin alustan omilta verkkosivuilta. Viimeisenä osuutena tutustutaan yleisesti väliohjelmistojen ominaisuuksiin sekä selvitetään HUBin käyttämän väliohjelmiston sopivuutta integraatioihin.

Työssä selvitettävät asiat pyritään jäsentelemään selkeiksi kokonaisuuksiksi, ja tärkeät asiat kootaan selkeiksi taulukoiksi. Verkkokauppa-alustojen ja väliohjelmiston integraatioita tarkastellaan eri näkökulmista, ja integraatioiden mahdolliset ongelmakohdat pyritään selvittämään.

2 Automaattivarasto

HUBin Hakkilan logistiikkakeskuksessa sijaitseva automaattivarasto liittyy olennaisesti insinööriyöhön, sillä se käsittelee lopulta verkkokaupoista tehtyjen tilausten toimitukset. Tässä luvussa esittelen automaattivaraston. Automaattivaraston tarvitsemia tietoja pohdin luvussa 5.4.

HUBin automaattivarasto on saksalaisen Knapp AG:n toimittama kokonaisuus. Automaattivarastossa on 50 000 laatikkopaikkaa, jotka voidaan jokainen jakaa kahdeksaan osaan, jolloin automaattivarastoon saadaan enimmillään jopa 400 000 varastopaikkaa.

Varastoautomaatissa on 46 sukkulaa, neljä hissiä, kaksi keräilyasemaa, viisi pakkausasemaa, neljä täyttöasemaa ja pahvilaatikon pystytyskone. Sukkulat kuljettavat laatikoita automaattivaraston sisällä pitkittäin hyllyväleissä. Hissit kuljettavat laatikoita hyllykerrosten välillä. Automaattivaraston hyllymoduulin ulkopuolelle laatikot kulkevat rullaratoja pitkin.



Kuva 3. Osanäkymä HUBin automaattivarastosta Hakkilan logistiikkakeskuksessa. [Koivistolainen 2018.]

Edut

Varastoautomaatin etuja ovat nopeus, tarkkuus ja tehokas tilankäyttö. Automaatin keruunopeus on jopa 3500 tuoteriviä päivässä, joka mahdollistaa kolmen tunnin palvelutason. Verrokkina keskiarvoisesti perinteisen varaston keruunopeus on vain noin 250 riviä päivässä, ja se on vähintään yhdeksän tuntia hitaampi kuin Hakkilan automaattivarasto. Automaattivaraston keräilytarkkuus on noin 99,99 %. Tämän mahdollistaa valo-ohjattu keräily yhdessä tuotteiden painontarkistuksen kanssa. Automaattivarasto punnitsee laatikon jokaisen kerätyn rivin jälkeen ja ilmoittaa, mikäli virhe tapahtuu. Halutessa tuotteiden viivakoodin luku on myös mahdollista, mikä lisää omalta osaltaan keräilytarkkuutta. [Hakkila Automated Warehouse Service: 11–18.]

Toiminta

Automaattivaraston jokaisella hyllykäytävän kerroksella on oma sukkula eli kahdella käytävällä 23 kerroksessa on yhteensä 46 sukkulaa. Sukkulat toimivat laatikoiden kuljettimina kerroksissa. Sukkulat hakevat laatikot varastopaikoilta ja kuljettavat ne hyllystön päässä oleville hisseille. Hissit sijaitsevat hyllyvälien toisessa päässä. Hissit kuljettavat laatikot rullaradan tasolle, josta laatikot lähtevät eteenpäin kohti keräilyasemia.

Tuotteiden keräily varastoautomaatin kanssa tapahtuu käsin valo-ohjauksella. Automaatti tuo varastolaatikoita keräilyasemalla oikeassa järjestyksessä ja näyttää valolla, mistä laatikosta kerätään ja mihin laatikkoon laitetaan. Tilausrivin keräily kuitataan napia painamalla, jolloin automaatti tuo uuden laatikon, josta voi kerätä. Keräilyasemalla on samanaikaisesti useampi laatikko, mistä kerätään ja mihin kerätyt tuotteet laitetaan. Keräilyasemalla on näyttö, joka näyttää kerättävän tuotteen tiedot, määrän ja mikäli tuote on sarjanumeroseurattu. Sarjanumeroseuratuissa tuotteissa keräilijä lukee sarjanumeron viivakoodinlukijalla. Keräilyasemalla laatikoiden päällä olevat valoverhot tunnistavat, jos tuotetta yritetään kerätä väärästä laatikosta, jolloin automaatti ilmoittaa virheestä välittömästi. Automaatti myös tarkistaa kerätyt tuotteet painon perusteella. Jokaisen kerätyn rivin jälkeen automaatti punnitsee laatikon ja ilmoittaa, jos keräilyssä tapahtuu virhe.

Automaattivarasto on älykäs, ja se optimoi toimintaansa. Se osaa sijoittaa useimmin kerätyt tuotteet mahdollisimman lähelle. Automaatti laskee, missä järjestyksessä laatikoita tuodaan keräilyasemalla, jotta keräily olisi sujuvaa. Automaatti suorittaa myös jatkuvaa inventointia. Tämä tarkoittaa sitä, että aina varastolaatikon tyhjennettyä se pyytää käyttäjältä kuittausta, että laatikko on oikeasti tyhjä. Laatikot, jotka eivät tyhjene tietyn ajan sisällä, automaattivarasto laittaa inventaariojonoon. Inventaariojono voidaan sitten ajaa keräilyasemalle, jossa inventointi suoritetaan.

3 Verkkokaupankäynnin kasvu ja tietoa verkkokaupoista

Tässä luvussa tarkastellaan verkkokaupan kasvua ja sen merkitystä 3PL-yritykselle. Luvussa valitaan insinööriyön tarkasteluun viisi yleisintä verkkokauppa-alustaa Suomessa viime vuosina sekä ne esitellään lyhyesti.

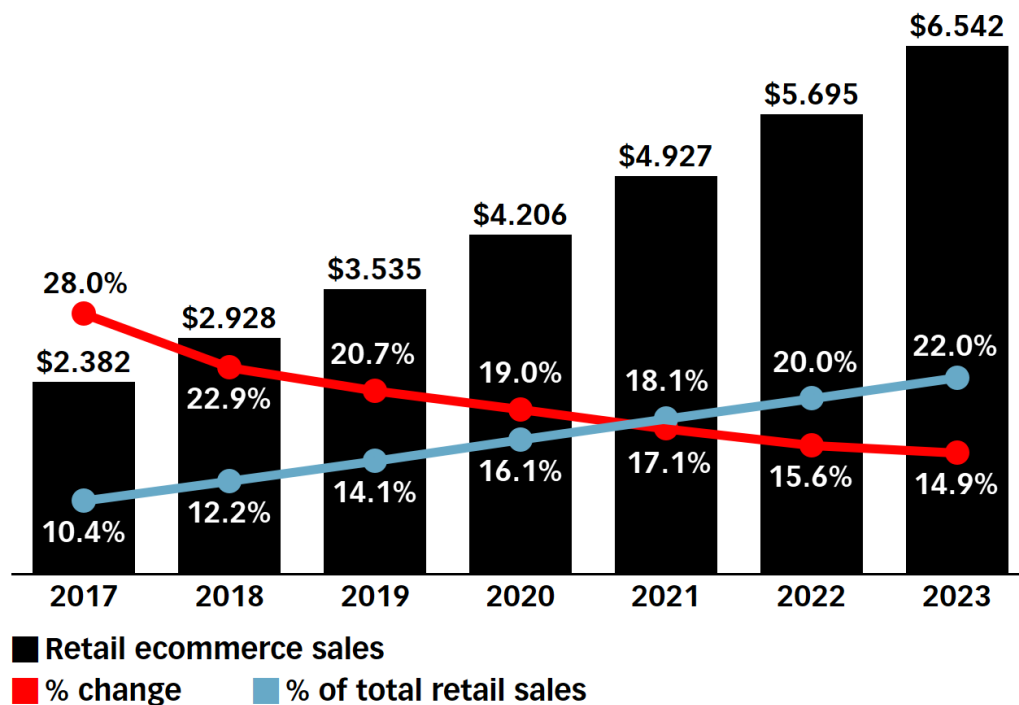
3.1 Verkkokaupankäynnin kasvu viime vuosina ja merkitys 3PL-yritykselle

Verkkokauppaostaminen kasvaa koko ajan niin Suomessa kuin maailmalla. Suomessa verkkokaupan liikevaihdon arvioidaan nousevan 13,8 miljardiin euroon vuonna 2019. Tämä tarkoittaa noin 1,6 miljardin euron kasvua vuodesta 2018. Paytrailin Verkkokauppa Suomessa raportin [2019: 4] mukaan yli 85 % suomalaisista tekee verkko-ostoksia. Näihin lukuihin kuuluvat kaikki verkkokaupan muodot eli tavaroiden lisäksi myös palvelut ja liput. Paytrail käyttää tutkimuksessa kolmea pääkategoriaa, joista matkailu on verkkokaupan suurin kategoria 42 % osuudella vuonna 2019. Tavarat ovat toiseksi suurin kategoria 37 % osuudella, ja palvelut ovat pienin kategoria 21 % osuudella. *Tavaroiden osuus euroissa on 5,1 miljardia.* Huomionarvoisena seikkana on, että päivittäistavaroiden ostaminen on lisääntynyt vuonna 2019 merkittävästi. Vuonna 2019 kuluttajista 24 % on ostanut päivittäistavaroita verkosta. Tämä on 52 % enemmän kuin viime vuonna. Raportissa mainitut vuoden 2019 luvut ovat arvioita. [Verkkokauppa Suomessa 2019: 6, 9–10.]

Maailmanlaajuisesti verkkokaupan liikevaihto on biljoonia euroja. Tänä vuonna se on arviolta 3,5 biljoonaa euroa ja noin 14 % kaikesta vähittäismyynnistä. Kasvua edelliseen vuoteen on noin 21 %. Kuvasta 4 nähdään, kuinka verkkokauppa on maailmanlaajuisesti kasvanut viimeisen kolmen vuoden aikana, sekä ennusteen seuraavan neljän vuoden kasvusta. Ennusteen mukaan verkkokaupan liikevaihto maailmanlaajuisesti kaksinkertaistuu vuodesta 2019 vuoteen 2023. [Lipsman 2019.]

Retail Ecommerce Sales Worldwide, 2017-2023

trillions, % change and % of total retail sales



Note: includes products or services ordered using the internet via any device, regardless of the method of payment or fulfillment; excludes travel and event tickets, payments such as bill pay, taxes or money transfers, food services and drinking place sales, gambling and other vice good sales

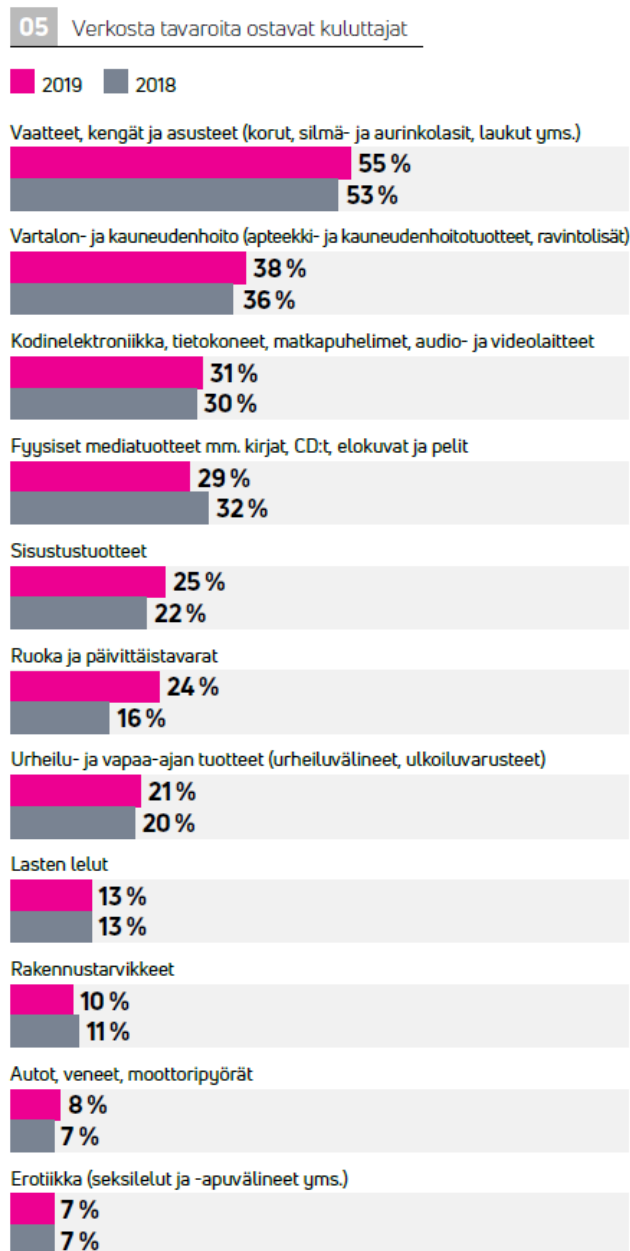
Source: eMarketer, May 2019

T10305

www.eMarketer.com

Kuva 4. Kaaviossa on maailmanlaajuisen verkkokaupan liikevaihto merkitty mustilla palkeilla. Punainen käyrä näyttää liikevaihdon kasvun edelliseen vuoteen. Sininen käyrä näyttää verkkokaupan vähittäismyynnin osuuden kaikesta vähittäismyynnistä. [Lipsman 2019.]

3PL-yrityksen kannalta mielenkiintoista tietoa ovat tavarat-kategorian luvut. Kuvan 5 kuvaajista nähdään tavarat-kategorian tuoteryhmien tarkemman erittelyn Suomen markkinoilla. Nämä ovat verkkokauppojen tuotteita, joiden logistiikan 3PL-yritys voisi hoitaa. Kuvaajassa näkyy prosentuaalisesti, kuinka suuri osa kuluttajista on ostanut mistäkin tuoteryhmästä tuotteita. Verkkokauppa Suomessa raportissa [2019] ei ilmennyt tavarat-kategorian alaisten tuoteryhmien liikevaihdon arvoa.



Kuva 5. Kuvaajassa näkyy kyseisiä tavaroita ostavien kuluttajien prosentiosuus Suomessa vuosina 2018 ja 2019. [Verkkokauppa Suomessa 2019: 10.]

Suurin osa Suomessa olevista verkkokaupoista tekee alle yhden miljoonan liikevaihtoa. Paytrailin Suuren verkkokauppa-alustaraportin [2019: 3] mukaan vain reilu 20 % verkkokaupoista Suomessa tekee yli yhden miljoonan euron liikevaihtoa. Tästä voidaan todeta, että suurin osa verkkokaupoista on melko pieniä.

3PL-yrityksillä on tyypillisesti ollut vain suuria asiakkaita. Tämä saattaa muuttua verkkokaupankäynnin kasvun myötä, jolloin 3PL-yrityksillä on mahdollisuus myös lukuisiin pienempiin verkkokauppa-asiakkaisiin. Integraatio ja automaatio ovat tässä tärkeässä roolissa. Tilausten tiedonkäsittelyn täytyy olla täysin automaattista työmäärän ja kulujen minimoimiseksi. Itse integraation täytyy olla toteutettavissa nopeasti ja edullisesti sekä toimia luotettavasti. Integraation apuna voidaan käyttää verkkokauppoihin ja logistiikkayrityksiin erikoistunutta iPaaS-väliohjelmistoa, josta kerrotaan luvussa 5.8.1.

3.2 Yleisimmät verkkokauppa-alustat vuosina 2017–2019

Suuren verkkokauppa-alustaraportin [2019: 4] tuloksista taulukossa 1 nähdään suosituimmat verkkokauppa-alustat Suomessa viimeisen kolmen vuoden aikana. Taulukkoon lisättiin sarake, josta näkee kolmen edellisen vuoden keskiarvon, joka tasaa yksittäisten vuosien trendejä. Suosituimpien verkkokauppa-alustojen valinta tehdään tämän keskiarvon perusteella. Toinen syy kolmen vuoden keskiarvon käyttöön on se, että 3PL-yrityksen potentiaalisilla asiakkailla saattaa jo olla verkkokauppa rakennettuna valmiina, joten on järkevää tarkastella vuoden 2019 lisäksi paria aikaisempaa vuotta. Mielestäni ei ole tarpeen tarkastella vuodesta 2017 taaksepäin, sillä ohjelmistot, etenkin verkkokauppa-ohjelmistot kehittyvät nopeasti.

Taulukon 1 mukaan insinööriyöhön valikoituvat seuraavat verkkokauppa-alustat:

- WooCommerce/WordPress
- MyCashflow
- Magento
- ePages/Vilkas
- Shopify.

”Oma”-verkkokauppa-alustat jätetään käytännöllisistä syistä pois tarkastelusta. Niistä on mahdotonta tehdä selvitystä, sillä jokainen alusta on oma toteutuksensa. ”Muu”-kategoriaan sisältyy lukuisia kyselyssä pienen prosenttiosuuden saaneita alustoja, joten myös tämä jätetään huomioitta, jolloin viidenneksi alustaksi valikoituu Shopify.

Taulukko 1. Suosituimmat verkkokauppa-alustat Suomessa vuosina 2017–2019 ja näiden vuosien keskiarvo. [Suuri verkkokauppa-alustaraportti 2019: 4.]

Alusta	Keskiarvo	2019	2018	2017
1. WooCommerce	17,3 %	18 %	18 %	16 %
2. MyCashflow	9,7 %	14 %	6 %	9 %
3. Magento	9,3 %	7 %	9 %	12 %
4. ePages/Vilkas	8,3 %	7 %	9 %	9 %
5. muu	8,3 %	9 %	3 %	13 %
6. oma	7,0 %	7 %	8 %	6 %
7. Shopify	6,3 %	6 %	8 %	5 %
8. Finqu	5,7 %	6 %	8 %	3 %
9. Clover Shop	5,0 %	8 %	5 %	2 %
10. Oscar eCommerce	3,3 %	4 %	2 %	4 %
11. Flowvy (ent. Omnisell) / Nethit	2,7 %	2 %	3 %	3 %
12. PrestaShop	2,3 %	2 %	3 %	2 %
13. OpenCart	2,2 %	2 %	2 %	3 %
14. Drupal	1,7 %	1 %	1 %	3 %
15. osCommerce	1,3 %	1 %	2 %	1 %
16. Holvi	0,8 %	2 %	1 %	1 %
17. Anders Inno	0,6 %	1 %	0 %	1 %

3.3 Avoin ja suljettu lähdekoodi

Avoin ja suljettu lähdekoodi ovat käsitteitä, jotka mainitaan lähes aina verkkokauppa-alustoista. *Avoimen lähdekoodin* ohjelmat ovat joustavampia ja paremmin muokattavissa kuin suljetun lähdekoodin ohjelmat. Avoimen lähdekoodin ohjelman lähdekoodi on nimensä mukaisesti avoimesti kenen tahansa nähtävissä, käytettävissä ja muokattavissa. Avoimen lähdekoodin ohjelmat ja niiden itsemuokatut versiot ovat vapaasti käytettävissä mihin tahansa tarkoitukseen. Niitä saa myös vapaasti kopioida ja levittää. Avoimen lähdekoodin vapaus mahdollistaa koodin kehittämisen julkisesti ja yhteisöllisesti. Yksi eduista on, että puuttuvan ominaisuuden voi kehittää joko itse tai palkata siihen pätevän henkilön. Avoimen lähdekoodin verkkokauppaohjelmistot asennetaan tyypillisesti itse omalle tai vuokratulle palvelimelle. On myös toimijoita, jotka toteuttavat verkkokauppoja

avoimen lähdekoodin ohjelmistoilla, jolloin toimijalla saattaa olla omat palvelimet. Joistakin avoimen lähdekoodin verkkokauppaohjelmistoista on myös kaupallinen versio, jonka saa täysin ylläpidettynä pilvipalveluna. [Avoin lähdekoodi 2019; Warnimont 2019.]

Suljetun lähdekoodin ohjelmiston lähdekoodi ei ole julkisesti saatavissa. Suljettu lähdekoodi tarkoittaa verkkokauppaohjelmistoissa lähes aina kuukausimaksullista SaaS-palvelua (Software as a Service), johon kuuluu palvelimen isännöinti (hosting-palvelu), ylläpito ja muut tukipalvelut. SaaS-palvelun eli pilvipalvelun etuna on se, että palveluntarjoaja kehittää, isännöi ja ylläpitää ohjelmistoa omilla palvelimillaan. Näin ollen käyttäjälle jää vain ohjelmiston käyttö ilman ylläpidollisia vastuita. Haittapuolena on käyttäjien rajoittuneisuus ohjelmiston suhteen. Ohjelmiston kehityksessä ja lisäominaisuuksissa pitää aina kääntyä ohjelmiston tekijän puoleen. Toinen haittapuoli on, että suljettu lähdekoodi sitoo yhteen toimittajaan. [Miten perustat menestyvän verkkokaupan? 2019.]

3.4 WooCommerce/WordPress

WooCommerce on ilmainen avoimen lähdekoodin verkkokauppaohjelmisto, joka julkaistiin vuonna 2011. Tarkemmin WooCommerce on WordPress-sisällönhallintaohjelmiston lisäosa (plugin). WooCommerce ja WordPress perustuvat avoimeen lähdekoodiin, ja ovat siksi täysin muokattavissa osaavissa käsissä. WooCommerce on maailmanlaajuisesti kaikista yleisin verkkokaupparatkaisu, jonka osuus kaikista verkkokaupoista maailmanlaajuisesti on 28 % vuonna 2019. [eCommerce for WordPress intro 2019.]

WordPress on ilmainen avoimen lähdekoodin ohjelmisto, jolla voidaan luoda verkkosivuja pienistä blogeista suuriin uutissivustoihin asti. WordPressin mukaan 34 % webistä käyttää WordPressiä. Referenssejä ovat muun muassa CNN, NY Times, Bloomberg, Mercedes Benz, Beyoncé ja Valkoinen talo. [Tämä on WordPress 2019.]

Ilmaisen WordPress-ohjelmiston voi ladata WordPress.org-sivustolta ja asentaa haluamalleen palvelimelle. WordPressin voi myös tilata kaupalliselta WordPress.com-sivustolta, jolloin palveluun kuuluu ylläpidetty palvelin ja tilauksen mukaiset ominaisuudet. Tilauksia löytyy viisi eri tasoista.

Vaikka WooCommerce ja WordPress ovat ilmaisia ohjelmistoja, niin se ei tarkoita ilmaista verkkokauppaa. Monet tarpeellisia lisäominaisuuksia tarjoavat lisäosat ovat maksullisia. Lisäksi verkkokaupan muokkaaminen halutunlaiseksi aiheuttaa kustannuksia, mikäli itsellä ei ole tarvittavia ohjelmointitaitoja.

WooCommerce sulautuu WordPressin hallintapaneeliin saumattomasti, ja on vaikea huomata, että se on WordPressin lisäosa, eikä WordPressin oma ominaisuus. WooCommerce säilyttää WordPressin kaikki ominaisuudet ja lisää sivustoon verkkokaupan toiminnallisuudet ja asetukset. WooCommerceen referenssejä ovat muun muassa Weber ja Singer.

Niin WooCommerceen kuin WordPressinkin ydin on toteutettu PHP-ohjelmointikielellä. Muita käytettyjä ohjelmointikieliä ovat HTML, CSS ja JavaScript. WooCommerceen ja WordPressin lähes kaikki lisäosat on toteutettu myös PHP-koodilla. Lisäosia voi kuka tahansa osaava tehdä. [Hayes 2019.]

3.5 MyCashflow

MyCashflow on kotimaisen Pulse247 Oy:n kehittämä verkkokauppaohjelmisto. MyCashflowlla on yli 1 500 aktiivista verkkokauppaa. Referenssejä ovat muun muassa Suojakalvotukku.fi, Fiksuruoka.fi ja Digitarvike.fi.

MyCashflow on täysin ylläpidetty pilvipalvelu, joka tarkoittaa sitä, ettei itse tarvitse huolehtia palvelimista, päivityksistä ja ylläpidosta. Tämä sopii hyvin asiakkaille, joilla ei ole aikaa tai teknistä tietotaitoa ylläpitämään verkkokauppaa. MyCashflow vaikuttaa helppokäyttöiseltä ja selkeältä verkkokauppaohjelmistolta. MyCashflowsta löytyy kolme eri tasoista palvelutilausta sekä täysin räätälöity verkkokaupparatkaisu. Peruspakettien hinnat ovat 49, 99 ja 149 euroa. [Perusta verkkokauppa 2019.]

MyCashflow'n suljettuun lähdekoodiin ei pääse käsiksi, joten sen toteutuksesta ei ole tarkempaa tietoa. MyCashflow'n ulkoasua voi muokata teemoilla. Teemoja voi räätälöidä HTML-koodilla ja CSS-tyylitiedostolla. MyCashflow perustuu HTML-kielisiin sivupohjiin. Sivupohjiin lisätään sisältöä MyCashflow'n Interface-merkkauksella eli eräänlaisilla tä-

geilla, jotka hakevat sisältöä sivuihin. Esimerkiksi sivupohjaan voisi lisätä seuraavan tagin: "{NewProduct}", joka tulostaisi sivulle uudet tuotteet HTML-muodossa. [Verkkokaupan suunnittelun helppous 2019.]

3.6 Magento

Magento verkkokauppaohjelmistosta on kolme erilaista versiota. Magento Commerce on kaupallisesti kehitettävä lisenssillinen versio, josta löytyy pilvipalveluversio ja asiakkaan palvelimelle asennettava versio. Magento Open Source (aikaisemmin Magento Community Edition) on ilmainen avoimeen lähdekoodiin perustuva versio. Adobe osti Magenton vuonna 2018, mutta ilmaisen version jakelu on silti jatkunut. Magentolla on yli 130 000 kauppiasta ympäri maailmaa. Magentosta on tullut uudistettu versio, Magento 2, vuosien 2016–2017 aikana. Tuki ensimmäiselle Magento-versiolle päättyy kesäkuussa 2020. Magento 2 -alustaa on kehitetty monipuolisemmaksi ja suunniteltu helpommin laajennettavaksi ja integroitavaksi muihin järjestelmiin. Magenton referenssejä ovat muun muassa Bauhaus, Canon, Festo, Helly Hansen ja Seat [Customer Case Studies 2019]. [Rogers 2019.]

Vuonna 2017 Magento Commerce -version lisenssin alkaen hinta on ollut Yhdysvalloissa 22 000 dollaria vuodessa. Lisenssimaksu määräytyy liikevaihdon mukaan. Magento on selvästi tarkoitettu suurien yritysten käyttöön. [Hughes 2017.]

Magenton Commerce -versioon kuuluu toiminnallisuuksia ja ominaisuuksia, joita ei ilmaisessa Open Source -versiossa ole. Näitä ovat esimerkiksi WYSIWYG-editori, erilaiset raportointi-, tilasto-, markkinointi- ja B2B-työkalut kuten tarjousten teko ja yritystilitt. Commerce-versioihin kuuluu tekninen tuki ja vianselvitys. Pilvipohjaiseen Commerce-versioon kuuluu myös hosting-palvelu. [Compare Open Source and Magento Commerce. 2019.]

Magenton ydin on WordPressin tavoin toteutettu PHP-kielellä. Magento-ohjelmisto koostuu moduuleista, teemoista ja kielipaketeista. Moduulit mahdollistavat jonkin toiminnallisuuden, ja ne voivat toimia yhdessä muiden moduuleiden kanssa. Moduuli voi sisältää käyttöliittymän käyttäjälle tai tiedon esittämiselle. Moduuli voi myös sisältää rajapinnan,

jota toinen moduuli voi kutsua. Modulaarinen rakenne mahdollistaa ohjelmiston laajentamisen ja kehittämisen. Teemat mahdollistavat verkkokaupan ja hallintapaneelin ulkoasun muokkaamisen. Kielipaketit tarjoavat kielikäännökset ohjelmiston teksteille, jotka näkyvät verkkokaupassa ja hallintapaneelissa. [PHP Developer Guide 2019.]

3.7 ePages/Vilkas

Vilkas Group Oy tarjoaa verkkokaupparatkaisua ePages-verkkokauppa-alustalle rakennettuna. Vilkas Group Oy on ePages:n virallinen yhteistyökumppani Suomessa. Vilkas on toteuttanut 1700 verkkokauppaa. Maailmanlaajuisesti ePages on käytössä yli 100 000 yrityksellä yli 70 maassa. Vilkas tarjoaa MyCashflow:n tavoin täysin ylläpidettyä pilvipalvelua. Verkkokaupat pyörivät DataCenter Finlandin konesalissa turvatuilla palvelimilla. [Usein kysyttyä 2019.]

Vilkaalla on tarjolla useampi palvelupaketti. Vilkas Now on uusin alusta, joka on helpokäyttöinen ja moderni. Vilkas Now -palvelusta on neljä eri palvelupakettia tarvittavan kapasiteetin ja ominaisuuksien mukaan. Vilkas Suite on heillä pidempään käytössä ollut alusta, joka tarjoaa enemmän ominaisuuksia myös isommalle yritykselle. Vilkas Suite -palvelusta on myös neljä palvelupakettia. Näiden lisäksi Vilkas tarjoaa täysin räätälöityä Vilkas Go -palvelua. Vilkas Go -palvelupaketin referenssejä ovat muun muassa Autoliitto, Marttiini, Puuilo ja Yliopiston verkkoapteekki. [Vilkas Go 2019; Hinnat 2019.]

3.8 Shopify

Shopify on pilvipohjainen, täysin ylläpidetty verkkokauppaohjelmisto. Shopify sai alkunsa vuonna 2004, kun Tobias Lütke, Daniel Weinand ja Scott Lake olivat perustamassa lumilautoja myyvää verkkokauppaa. Heidän tarpeilleen sopivaa verkkokauppaohjelmistoa ei ollut, joten he kehittivät verkkokaupallensa oman ohjelmiston. Shopify julkaistiin verkkokauppa-alustana vuonna 2006. Shopifyn omien sanojen mukaan Shopifylla on toteutettu yli 1 000 000 verkkokauppaa ja Built With -sivuston tilastojen mukaan tällä hetkellä Shopifylla toteutettuja sivuja on noin 910 000. Built With -sivuston käyrästä näkee kuinka Shopifyn suosio on noussut merkittävästi vuonna 2017 ja sen jälkeen. [Company Info 2019; About Us 2019; Shopify Usage Statistics 2019.]

Shopifylla on tarjolla viisi palvelupakettia. Kaikkiin muihin tilauksiin kuin Lite-tilaukseen kuuluu hosting-palvelu, rajaton kaistankäyttö ja rajaton määrä tuotteita. Shopify:n referenssejä ovat muun muassa Acon, Vallila, Reima ja Suomalainen Kirjakauppa.

3.9 Yhteenveto verkkokauppa-alustoista

Taulukossa 2 on yhteenveto edellä läpikäytyjen verkkokauppa-alustojen toteutusmaileista. Kaikilla alustoilla toteutetaan verkkokauppoja myös pilvipalveluna.

Taulukko 2. Verkkokauppa-alustojen toteutusmallit

	WooCommerce	MyCashflow	Magento	ePages/ Vilkas	Shopify
Avoim vai suljettu lähdekoodi	Avoim	Suljettu	Kumpikin	Suljettu	Suljettu
Pilvipalvelu vai paikallinen asennus	Kumpikin	Pilvipalvelu	Kumpikin	Pilvipalvelu	Pilvipalvelu
Ilmainen vai kaupallinen	Ilmainen	Maksullinen	Kumpikin	Maksullinen	Maksullinen
Myös kaupallinen versio	Ei	-	Kyllä	-	-
Sisältykö hosting-palvelu	Valittavissa	Sisältyy	Valittavissa	Sisältyy	Sisältyy

4 Integraatioihin liittyvä tekniikka

Luvussa käsitellään integraation perusteita, jotka auttavat ymmärtämään myöhemmin olevaa selvitystyötä verkkokauppojen rajapinnoista sekä sitä, kuinka verkkokauppa-alustoja integroidaan muihin järjestelmiin. Erilaisista web-rajapinnoista käyn syvällisemmin läpi REST-rajapintoja, sillä ne ovat tällä hetkellä yleisin käytössä oleva rajapinta monissa verkkokauppa-alustoissa. Muita web-rajapintoja käsittelen lyhyemmin.

4.1 Integraatio käsitteenä

Integraatio on kahden tai useamman järjestelmän tai ohjelmiston yhdistämistä toisiinsa siten, että ne pystyvät kommunikoimaan toistensa kanssa. Järjestelmien yhdistämiseen tarvitaan rajapintoja, jotka kuvainnollisesti toimivat järjestelmien korvina ja suuna. Rajapinnat eli API (Application Programming Interface) mahdollistavat järjestelmien välisen

tiedonsiirron ja kommunikoinnin. Monista järjestelmistä löytyy usein valmiina rajapinta, jonka kautta järjestelmään voidaan viedä dataa tai sieltä voidaan ottaa dataa ulos. Integraatio toimii näiden rajapintojen kautta. Integraatioissa käytetään usein väliohjelmistoa (Middleware), joka nimensä mukaisesti toimii kahden tai useamman järjestelmän välissä. Käytännössä integraatio on joukko toimintoja, kuten reitityksiä, muunnoksia, tiedostojen ja sanomien välitystä. Integraatioita on yksi- ja kaksisuuntaisia. Yksisuuntaisessa integraatiossa lähettävä osapuoli lähettää dataa vastaanottavalle osapuolelle. Kaksisuuntaisessa integraatiossa data kulkee molempiin suuntiin. [Integraatioiden perusteet 2019: 3, 8; Integraatio-opas 2019.]

Perimmäinen syy integraatioiden tarpeeseen on, että ohjelmistoja ei ole rakennettu toimimaan yhdessä. Onneksi monista ohjelmistoista kuitenkin löytyy rajapinta, jonka kautta integraatio muihin ohjelmistoihin voidaan toteuttaa. Integraation tarve syntyy siitä, että yrityksissä käytetään monia eri tietojärjestelmiä, jotka pitää yhdistää asiakkaiden ja toimittajien tietojärjestelmiin. Nykyään yrityksistä löytyy monia ohjelmistoja, joiden tietoja halutaan käyttää ristiin eri ohjelmistoissa. Tällaisia ohjelmistoja voivat olla esimerkiksi verkkokauppaohjelmisto, kassajärjestelmä, asiakkuudenhallintajärjestelmä (CRM), toiminnanohjausjärjestelmä (ERP), varastonhallintajärjestelmä (WMS) sekä kirjanpito-ohjelmisto. [Integraatioiden perusteet 2019: 5–6.]

Integraatio vähentää manuaalista työtä automatisoimalla tiedonsiirron järjestelmien välillä. Automatisointi poistaa raskaat ja aikaa vievät työtehtävät, joissa tietoa haetaan ja syötetään käsin järjestelmästä toiseen. Automatisointi luo henkilöstösäästöjä ja vapauttaa henkilöresursseja muihin tehtäviin. Automatisointi myös vähentää tietovirheiden määrää, esimerkiksi näppäilyvirheet poistuvat kokonaan. Oikein toteutetun integraation myötä yhteen järjestelmään syötetty tieto synkronoituu kaikille tiedon tarvitseville osapuolille. Näin yhteistyö sidosryhmien kanssa tehostuu ja muuttuu reaaliaikaiseksi. Yhdessä integraatiossa käytettyä väliohjelmistoa voidaan usein käyttää myös uusissa integraatioissa, jolloin uusien integraatioiden tekeminen helpottuu ja yksittäisen ohjelmiston vaihtaminen toiseen helpottuu. Syitä ohjelmiston vaihtoon voivat olla kustannukset tai puuttuvat ominaisuudet. [Integraatio-opas 2019: 6–9.]

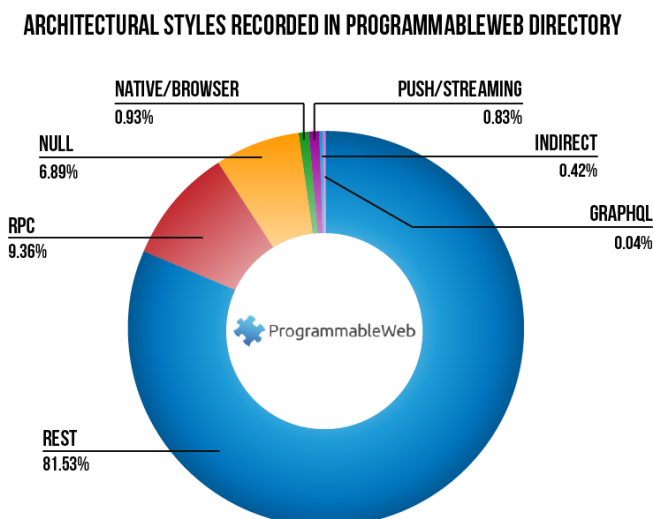
API (Application Programming Interface) on sovellusohjelmointirajapinta, joka mahdollistaa datan ottamisen ohjelmistosta tai viemisen sinne. ”API-rajapinta”-termiä yleistetään ja sitä käytetään usein kuvaamaan web-rajapinnan kokonaisuutta, johon kuuluvat ohjelmisto, palvelin ja tietokanta. Tarkasti ottaen API ei ole tietokanta eikä palvelin vaan ohjelmistokoodin se osa, joka hallitsee ohjelmiston liityntäpistettä ja määrittelee kommunikoinnissa käytetyt menettelytavat. [Eising 2017.]

4.2 Web-rajapinta

Web-rajapinnasta (Web API) puhuttaessa tarkoitetaan yleensä rajapintapalvelua kokonaisuudessaan, eli itse API-rajapintaa, ohjelmistoa sen taustalla sekä palvelinta. Web-rajapinta on rajapinta, joka on käytettävissä webissä internetin välityksellä. Kommunikointi web-rajapinnan kanssa tapahtuu internetin ja webin (WWW) käyttämillä protokollilla eli yleisimmin HTTP-tiedonsiirtoprotokollan avulla. Yksinkertaistettuna web-rajapinta toimii siten, että asiakas lähettää halutusta tiedosta kyselyn palvelimelle ja palvelin vastaa viestiin kysytyllä tiedolla. Rajapinnan käyttämä tiedonsiirtoprotokolla ja viestin tiedostoformaatti riippuvat rajapinnan tyypistä.

Web-rajapintoja nimetään usein joko sovelluksen nimen tai sen käyttämien tekniikoiden mukaan tai molempien mukaan. Muutamia esimerkkejä rajapintojen nimistä ovat ”Twitter API”, ”Facebook API”, ”Facebook Marketing API”, ”Wordpress REST API” ja ”Open Weather Map API”.

Web-rajapintojen arkkitehtuurisia tyylejä ovat muun muassa seuraavat: REST (Representational State Transfer), GraphQL, Falcor, RPC (Remote Procedure Call), SOAP (Simple Object Access Protocol) ja gRPC. Nykyään suurin osa web-rajapinnoista on toteutettu REST-tyylin mukaisesti. Programmable Web -sivuston hakemistoon lisätystä web-rajapinnoista lähes 82 % on REST-tyylisiä. Suurin osa verkkokauppa-alustojen rajapinnoista on myös REST-tyylisiä. Tämän insinöörityön selvityksen verkkokauppa-alustojen rajapinnoista neljä ovat REST-tyylisiä ja yksi SOAP-protokollan mukainen. Yhdellä on REST-rajapinnan lisäksi käytössä SOAP-rajapinta ja toisella GraphQL-rajapinta. Insinöörityön pienestä otannasta selviää kuitenkin REST-rajapintojen suosio, joka selviää myös kuvasta 6. Kuvassa 6 SOAP-rajapinta kuuluu RPC-arkkitehtuurin kategoriaan. Verkkokauppa-alustojen rajapintoja tutkitaan luvussa 5.7.



Kuva 6. Erilaisten web-rajapintojen osuus vuonna 2017. [Santos 2017.]

4.3 Web Service

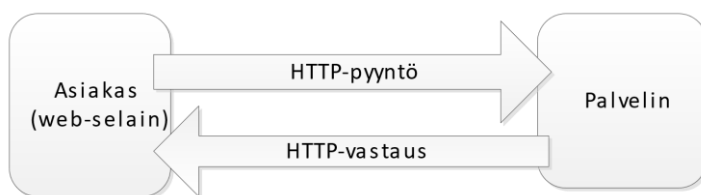
World Wide Web Consortium -yhteenliittymän (W3C) käyttämän määritelmän mukaan Web Service on ohjelmistojärjestelmä, joka on suunniteltu tukemaan koneiden välistä tiedonsiirtoa tietoverkon ylitse. Web Servicellä on rajapinta, joka on kuvattu koneluettavassa muodossa WSDL-dokumentilla (Web Service Description Language). Muut ohjelmistot ovat vuorovaikutuksessa Web Servicen kanssa käyttämällä XML-muotoisia SOAP-viestejä, jotka tyypillisesti lähetetään HTTP:n ylitse. [Web Services Architecture 2004.]

Web Servicet ovat siis webissä olevia ohjelmistoja, jotka pystyvät kommunikoimaan muiden ohjelmistojen kanssa automaattisesti internetin välityksellä. Web Service on web-rajapinta, joka käyttää SOAP-protokollaa ja WSDL-dokumenttia. Web Servicet ovat rajapintoja, mutta kaikki rajapinnat eivät ole Web Servicejä.

Web Servicen käsite tuntuu olevan määrittelykysymys. Web-rajapinnoista saatetaan puhua Web Servicenä, vaikka se ei noudattaisi SOAP-protokollan määrittelyä. Useampaan artikkeliin perehtymisen jälkeen, yleisin mielipide tuntuu olevan se, että Web Service on SOAP-rajapinta, joka on kuvattu WSDL-dokumentilla, jotta se olisi koneluettava ja koneiden käytettävissä. Tämä käy yhteen W3C:n käyttämän määritelmän [Web Services Architecture 2004] kanssa.

4.4 HTTP-protokolla

HTTP (HyperText Transfer Protocol) on webin käyttämä tiedonsiirtoprotokolla eli yhteyskäytäntö. Esimerkiksi verkkoselaimet ja palvelimet käyttävät HTTP-protokollaa tiedonsiirtoon. HTTP on sovelluskerroksen protokolla, joka toimii TCP-protokollan (Transmission Control Protocol) päällä. HTTP toimii asiakas-palvelinmallilla kuvan 7 mukaisesti. Asiakas-palvelinmallissa asiakas ottaa yhteyttä palvelimeen eli tekee pyynnön, ja palvelin vastaa siihen. [An overview of HTTP 2019.]



Kuva 7. Asiakas-palvelinmalli. [HTTP 2015.]

HTTP:llä on seuraavia ominaisuuksia [An overview of HTTP 2019]:

- *HTTP on yksinkertainen ja selkeä.* HTTP on suunniteltu yksinkertaiseksi ja ihmisen luettavaksi. HTTP viestien ihmisen luettava muoto helpottaa kehittäjiä testaamisessa.
- *HTTP on laajennettavissa.* HTTP:n otsikkotiedot mahdollistavat lisäominaisuuksien kehittämisen ja käyttöönoton. Asiakassovelluksen ja palvelimen kanssa sovittu uusi ominaisuus voidaan lisätä HTTP:n otsikkotietoihin.
- *HTTP on tilaton protokolla.* Tämä tarkoittaa sitä, että palvelin ei muista aikaisempia pyyntöjä, joten kunkin pyynnön pitää sisältää kaikki pyynnön käsittelyyn tarvittavat tiedot. Tilattomuuden tarkoitus on yksinkertaistaa palvelimen toteutusta ja parantaa skaalautuvuutta.

HTTP-protokollan kanssa tapahtuva viestintä asiakkaan ja palvelimen välillä tapahtuu seuraavan mukaisesti. Ensimmäiseksi asiakas aukaisee TCP-yhteyden palvelimeen. Tämä tapahtuu yhden tai useamman pyynnön ja vastauksen tuloksena. Asiakas voi aukaista yhden tai useamman uuden yhteyden tai käyttää jo aukaistua yhteyttä. Seuraavaksi asiakas lähettää pyynnön palvelimelle HTTP-viestinä, johon palvelin vastaa pyydetyllä tiedolla. Viimeiseksi yhteys katkaistaan tai sitä voidaan käyttää lisäpyyntöjen tekemiseen. [An overview of HTTP 2019.]

Osa HTTP-pyyntöistä sisältää vain pyyntörivin ja otsikkotiedot. Mutta esimerkiksi POST-metodin pyynnöt sisältävät myös runko-osan, joka sisältää palvelimelle tallennettavan tiedon vastaavanlaisessa muodossa kuin tieto olisi palvelimen vastauksessa. HTTP-pyyntöjen rakenne on seuraavanlainen:

- *Pyyntö-rivi (request line)* sisältää HTTP-metodin, haettavan resurssin URL:n ja HTTP:n version.
- *Otsikkotiedoissa (headers)* on avain-arvopareja, esimerkiksi: "Accept: text/plain".
- Tyhjä rivi otsikkotietojen ja rungon välissä.
- *Runko (message body)* voi olla mitä tahansa, esimerkiksi HTML-, XML- tai JSON-dokumentti, tekstiä, kuva tai video.

HTTP-pyyntöjen otsikkotiedoissa määritellään kohdepalvelimen osoite, joka on HTTP/1.1-protokollan ainoa pakollinen kenttä. Muita HTTP-pyyntöjen otsikkotietoja voivat olla tietoa asiakkaasta tai pyyntöjen lähettäjän URI:sta. HTTP:n otsikko sisältää myös HTTP-protokollan määrittelemiä avain-arvopareja. Avain-arvoparit voivat olla myös sovelluskohtaisia. Avain-arvopareilla voidaan määrittää muun muassa viestin rungon formaatti, välimuistin käyttäytyminen, autentikointitiedot ja paljon muita tietoja. [HTTP 2015.]

HTTP-vastauksen rakenne on seuraavanlainen:

- *Status-rivi (status)* sisältää HTTP:n version ja status-koodin selitystekstin kanssa, esimerkiksi: "HTTP/1.1 404 Not Found".
- *Otsikkotiedoissa (headers)* on avain-arvopareja, esimerkiksi: "Content-type: text/html; charset=UTF-8".
- Tyhjä rivi otsikkotietojen ja rungon välissä.
- *Runko (message body)* voi olla mitä tahansa, esimerkiksi HTML-, XML- tai JSON-dokumentti, tekstiä, kuva tai video.

HTTP-pyyntöjä on erilaisia. Pyyntöjen tyyppi määritetään pyyntöjen alussa olevalla HTTP-metodilla. Metodeista puhutaan myös verbeinä. HTTP/1.1-standardi [2014] määrittelee kahdeksan eri metodia, jotka ovat seuraavat:

- *GET* hakee resurssin representaation.
- *DELETE* poistaa resurssin.
- *POST* lähettää dataa palvelimelle, esimerkiksi luo uuden resurssin.

- *PUT* päivittää resurssin tai luo uuden, jos resurssia ei ole olemassa.
- *HEAD* hakee representaation mukana lähetettävät otsikkotiedot, mutta ei itse resurssin representaatiota.
- *OPTIONS* hakee resurssille käytössä olevat HTTP-metodit.
- *CONNECT* luo kaksisuuntaisen yhteyden valittuun resurssiin. Tämän avulla voidaan aukaista SSL-tunneli.
- *TRACE* lähettää viestisilmukatestin resurssin osoitteeseen. Tätä käytetään viankorjaukseen.

PATCH-metodi on lisätty HTTP-standardiin myöhemmin vuonna 2010. *PATCH*-metodilla voidaan päivittää resurssin yksittäisiä tietoja, kun taas *PUT*-metodin kanssa resurssi pitää päivittää kokonaisuudessaan eli korvata uudella. [PATCH Method for HTTP 2010.]

Metodit voidaan luokitella turvallisiin ja idempotentteihin metodeihin. Turvalliset metodit (safe methods) eivät muuta palvelimen tilaa. Turvallisia metodeja ovat GET, HEAD, OPTIONS ja TRACE. Idempotentin metodin (idempotent method) yhdellä pyynnöllä on sama vaikutus kuin usealla identtisellä peräkkäisellä pyynnöllä. Idempotentteja metodeja ovat PUT ja DELETE sekä kaikki edellä mainitut turvalliset metodit.

HTTP-vastaus sisältää statuskoodin (status code, response code), joka kertoo pyynnön onnistumisesta tai epäonnistumisesta numeerisella koodilla ja selitystekstillä. Statuskoodien luokka ja yleisimpiä statuskoodeja on esitetty taulukossa 3.

Taulukko 3. Statuskoodien luokat ja yleisimpiä statuskoodeja kustakin luokasta.

Statuskoodin luokka	Kuvaus	Esimerkki
1xx	Pyyntö kesken	100 - Continue
2xx	Pyyntö onnistui	200 - OK 201 - Created
3xx	Uudelleenohjaus	301 - Moved Permanently 302 - Found
4xx	Asiakkaan virhe	400 - Bad Request 404 - Not Found
5xx	Palvelimen virhe	500 - Internal Server Error 502 - Bad Gateway

HTTP-protokollan toinen versio, HTTP/2-protokolla, julkistettiin vuonna 2015. HTTP/2-protokollassa on paljon uutta, joka tekee tiedonsiirrosta nopeampaa ja luotettavampaa. HTTP/2 on hyvin yhteensopiva HTTP/1.1-protokollan kanssa, joten uusi versio ei muuta asioita kovinkaan paljon käyttäjiä ajatellen. HTTP/2-protokollan ominaisuuksia ovat muun muassa seuraavat [Tarvainen 2015]:

- Tiedonsiirto on tehokkaampaa, sillä sisältö siirretään binäärimuodossa ja otsikkotiedot pakataan.
- Useita viestejä voidaan lähettää rinnakkain samanaikaisesti samassa yhteydessä (multiplexing).
- Palvelin voi lähettää dataa selaimelle ilman selaimen erillistä pyyntöä. Ominaisuutta kutsutaan nimellä HTTP/2 Server Push. Tällaista dataa voi olla esimerkiksi verkkosivujen tyylitiedostot.

4.5 REST-arkkitehtuurityyli

REST (Representational State Transfer) on hajautettujen hypermediajärjestelmien arkkitehtuurityyli. REST ei ole standardi tai protokolla, joka määritteli tarkasti käytetyt tekniikat, vaan tyyli, joka antaa suuntaviivat RESTin mukaiselle toteutukselle. Roy Thomas Fielding määritteli REST-arkkitehtuurityylin tohtorin väitöskirjassaan vuonna 2000. REST-arkkitehtuurityyli on monista eri verkkopohjaisten ohjelmistojen arkkitehtuurityyleistä koostettu Fieldingin sanojen mukaisesti ”hybridityyli” [2000: 76].

Tunnetuin esimerkki REST-tyylin arkkitehtuurista on web (WWW eli World Wide Web), joka olikin REST-arkkitehtuurityylin kehittämisen suurin motivaattori. Web on hajautettu hypermediajärjestelmä, jonka toiminta perustuu REST-tyyliin ja webin standardeihin, joita ovat URI, HTTP ja HTML. Fieldingin REST-tyylin kehittäminen oli osa laajempaa työtä webin standardien kehitystyössä Internet Engineering Taskforce -organisaatiossa (IETF) ja World Wide Web Consortium -yhteenliittymässä (W3C). Fielding on ollut osallisena webin kehitystyössä jo vuodesta 1993 lähtien. RESTin ensimmäinen versio kehitettiin jo vuosien 1994–1995 aikana, joita seuraavien viiden vuoden aikana sitä iteratiivisesti paranneltiin. [Fielding 2000: 76, 107–109.]

Webistä tuli suosittu sen käytön helppoudesta. Kuka tahansa pystyi käyttämään sitä käytännössä ilman minkäänlaista opastusta. Tämä on suurelta osin sen ansiota, että web on kehitetty noudattamaan REST-tyylin periaatteita. Tavallisten www-sivujen selaamisen

lisäksi web on myös tehokas alusta hajautettuja ohjelmia varten. Tekniikka, jonka avulla kuka tahansa henkilö voi käyttää webiä, mahdollistaa myös webin käytön automatisoidusti ohjelmistoille. Web käyttää pääasiassa kolmea tekniikkaa: URL-osoitteita, HTTP-protokollaa ja HTML-dokumenttiformaattia. [Ruby ym. 2013: I. 1.]

REST-nimen (Representational State Transfer) on tarkoitus luoda mielikuva siitä, miten hyvin suunniteltu web-sovellus toimii. Web-sovellus on kokoelma sivuja, jotka ovat yhdistetty toisiinsa linkeillä. Kukin www-sivu esittää (represent) käyttäjälle sovelluksen yhden tilan (state). Sovelluksen tilasta toiseen siirrytään sivun linkkien avulla, ja kyseisen tilan esitys siirretään (transfer) käyttäjälle. [Fielding 2000: 109.]

REST-arkkitehtuurityylillä tavoitellaan kehittymiskykyisyyttä, suorituskykyä, skaalautuvuutta, yksinkertaisuutta, muunneltavuutta, läpinäkyvyyttä, siirrettävyyttä ja luotettavuutta [Fielding 2000: 76–86; Fielding 2013]. Näihin päämääriin päästään noudattamalla Fieldingin rajoitteita, jotka ovat

- asiakas-palvelinmalli (client-server)
- tilattomuus (stateless)
- välimuisti (cache)
- yhdenmukainen rajapinta (uniform interface)
- kerroksittainen järjestelmä (layered system)
- ladattava koodi (Code-On-Demand).

Asiakas-palvelinrajoitteen tarkoitus on yksinkertaistaa palvelimen toimintoja skaalautuvuuden parantamiseksi ja parantaa käyttöliittymän siirrettävyyttä eri alustoille. Usein tämä tapahtuu toimintoja jakamalla siten, että asiakassovellus huolehtii rajapinnan toiminnallisuuksista ja palvelin tiedon käsittelystä tietokannassa. Tämä jako mahdollistaa asiakas- ja palvelinpuolen ohjelmistojen itsenäisen kehittämisen. [Fielding 2000: 78.]

Tilattomuusrajoite tarkoittaa sitä, ettei palvelin tallenna asiakassovelluksen tiloja tai pyyntöjä. Jokainen pyyntö on yksittäinen pyyntö, ja sen tulee sisältää kaikki pyyntöön tarvittavat tiedot. Jos on tarvetta tallentaa tiloja, niin sen tekee asiakassovellus. Tilattomuuden etuja ovat palvelimen skaalautuvuus, monitoroinnin helpottuminen ja luotettavuuden parantuminen. Tilattomuuden haittapuolena on verkon mahdollinen kuormittuminen, sillä jokaisen pyynnön pitää olla itseselitteinen. [Fielding 2000: 78–79.]

Välimuistin tarkoitus on kompensoida tilattomuusrajoitteesta aiheutuvaa verkon kuormitusta. *Välimuistirajoite* vaatii, että palvelimen vastauksessa on oltava selkeä tieto siitä, että saako asiakas tallentaa vastauksen datan asiakasohjelman välimuistiin. Jos tallentaminen on sallittu, niin asiakasohjelman voi hakea kyseisen tiedon välimuististaan lähettämättä uutta samanlaista pyyntöä kyseisen tiedon hakemiseksi. Tämä parantaa suorituskykyä, skaalautuvuutta ja käyttäjäkokemusta vähentämällä tiedonsiirron keskimääräistä viivettä. Haittapuolena on luotettavuuden vähentyminen, sillä välimuistissa voi olla vanhentunutta tietoa. [Fielding 2000: 79–81.]

Yhdenmukainen rajapinta ohjelmistojen välillä erottaa oleellisesti REST-tyylin muista ohjelmistoarkkitehtuureista. REST-tyyli korostaa, että ohjelmistojen välisten rajapintojen tulee olla yhdenmukaisia. Tämä yksinkertaistaa kokonaisarkkitehtuuria ja parantaa ohjelmistojen välisen vuorovaikutuksen näkyvyyttä. Tämä mahdollistaa myös ohjelmistojen itsenäisen kehityksen, sillä ohjelmistojen toteutus on erotettu niiden tarjoamista palveluista ja rajapinnoista. Yhdenmukaisen rajapinnan haittapuolena on suorituskyvyn heikkeneminen, sillä data siirretään aina samassa muodossa, eikä sovelluksen tarpeiden mukaisessa muodossa. Yhdenmukainen rajapinta määritetään neljällä rajapintarajoitteella, joita ovat *resurssien tunnistaminen*, *resurssien käsittely representaatioiden avulla*, *itseselitteiset viestit* ja *hypermedia sovelluksen tilakoneena*. [Fielding 2000: 81–82.]

Resurssi on tiedon keskeinen käsite REST-tyylissä. Resurssilla tarkoitetaan mitä tahansa tietoa, joka voidaan nimetä. Toisin sanoen resurssi voi olla mikä tahansa käsite, johon voidaan viitata hypertekstissä. Resurssi voi olla dokumentti, kuva, palvelu (esimerkiksi päivän sää Helsingissä), kokoelma muita resursseja tai todellinen asia (esimerkiksi henkilö). Käytännössä jokainen linkki verkkosivuilla tunnistaa jonkin resurssin antamalla sen sijainnin webissä. Resurssit voivat muuttua ajan myötä. Ainoastaan resurssin nimen pitää pysyä samana, sillä se erottaa resurssin muista resursseista. Resurssi tunnistetaan URI:n (Uniform Resource Identifier) avulla, joka siis on webissä URL-osoite. [Fielding 2000: 88–89.]

Representaatio on palvelimen tarjoama esitys resurssista eli tiedosta. Representaatio on siis palvelimen lähettämä dokumentti resurssista. Representaatio koostuu datasta ja metadatasta, joka on sarja avain-arvopareja. Representaation dataformaatti tunnetaan ni-

mellä mediatyyppi. Mediatyyppejä löytyy erilaisia, joista toiset soveltuvat paremmin automatisoituihin prosesseihin, ja toiset on tarkoitettu renderöitäväksi ihmiskäyttäjän luettavaksi. *Resursseja muokataan representaatioiden avulla.* Asiakassovellus lähettää muokatun representaation palvelimelle, joka sitten hyväksyy tai hylkää pyynnön. Asiakassovellus ei koskaan pääse muokkaamaan resurssia suoraan. [Fielding 2000: 90–92.]

Itseselitteinen viesti sisältää kaiken tarpeellisen tiedon viestin käsittelemiseksi. Viestien itseselitteisyys mahdollistaa välikomponenttien käytön. Viestien pitää olla itseselitteisiä myös sen takia, että palvelin ei tallenna asiakas-palvelinistunnon dataa tilattomuusrajoitteen vuoksi. [Fielding 2000: 121–125.]

Hypermedia sovelluksen tilakoneena (Hypermedia as the Engine of Application State, lyhenteenä HATEOAS) tarkoittaa sitä, että palvelin osaa kertoa asiakassovellukselle mitä seuraavaksi voi tehdä ja miten. Tämä tapahtuu palvelimen vastauksessa olevien linkkien avulla, joissa on toisilleen relevanttien resurssien URL-osoitteet ja resurssien hyväksymät metodit. Toisin sanoen palvelin johdattaa asiakassovellusta tilasta toiseen representaatioissa olevien linkkien avulla. [Jedrzejewsk 2018.]

Kerroksittainen järjestelmä koostuu hierarkkisista kerroksista, jotka siirtävät tietoa toisensa välillä. Tähän liittyvä rajoite rajoittaa kerroksen tiedonsiirron vain välittömästi ylä- ja alapuolella oleviin kerroksiin. Näin yksittäisen järjestelmän osan eli kerroksen toiminnallisuutta voidaan yksinkertaistaa. Kerroksilla voidaan muun muassa kapseloida vanhoja järjestelmän osia ja parantaa tietoturvaa. Kerroksittaisen järjestelmän haittapuolena on datan käsittelyn lisääntymisen aiheuttama viive. Viivettä voidaan kuitenkin kompensoida kerroksittaisen järjestelmän välimuistin käytön avulla. [Fielding 2000: 82–84.]

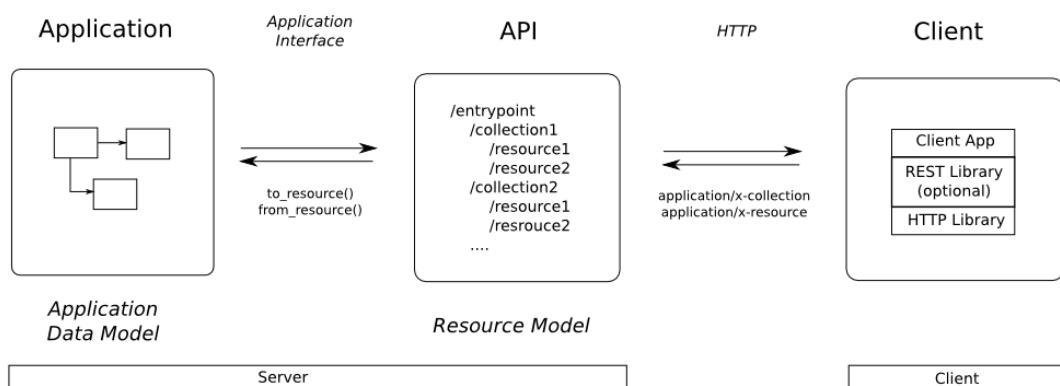
Ladattavan koodin avulla voidaan suorittaa toiminnallisuuksia asiakasohjelman päässä. Palvelin lähettää koodin vastauksen mukana, ja koodi suoritetaan asiakkaan päässä. Tämä vähentää asiakasohjelman toiminnallisuuksien tarvetta ja mahdollistaa ohjelman toiminnallisuuksien laajentamisen käyttöönoton jälkeen. Ladattava koodi on REST-tyylin valinnainen rajoitus. [Fielding 2000: 84–86.]

4.6 REST-rajapinta

REST-rajapinnat (myös RESTful API, REST API -rajapinnat) noudattavat REST-tyylin rajoitteita. Termiä REST-rajapinta käytetään melko usein väärin ja löyhästi kuvaamaan web-rajapintaa välittämättä siitä, kuinka tarkasti rajapinta noudattaa REST-tyylin periaatteita. Nykyään tilanne on hieman parantunut ja REST-termiä käytetään oikeammin kuvaamaan rajapintoja.

Käytännössä mikään niin sanottu REST-rajapinta ei täysin täytä Fieldingin rajoitteita. HATEOAS-rajoite on vaikeimmin ja vähinten toteutettu ominaisuus REST-rajapinnoissa. Tosin rajapinnoissa ei aina ole tarvetta antaa vastauksessa linkkejä muihin resursseihin. Richardsonin kypsyysmalli (The Richardson Maturity Model) luokittelee web-rajapinnat eri luokkiin sen mukaan, miten hyvin ne täyttävät REST-tyylin määitykset. En käy tarkemmin Richardsonin kypsyysmallia, mutta mainitsen tämän viitetietona, sillä Richardson on tarkastellut noin sataa eri web-rajapintaa peilaten niiden ominaisuuksia REST-tyyliin. [Richardson 2008.]

Tyypillinen REST-rajapinta käyttää useita resurssien päätepisteitä eli URL-osoitteita ja useita eri HTTP-metodeja resurssien käsittelyyn. REST-rajapinnassa asiakas lähettää pyynnön HTTP-protokollaa käyttäen URL-osoitteeseen, jossa sijaitsee haluttu resurssi. Palvelin vastaa GET-pyyntöön resurssin representaatiolla. Tyypillisen REST-rajapinnan resurssin representaatio on yleensä XML- tai JSON-dokumentti. Onnistuneeseen POST-pyyntöön palvelin vastaa yleensä statuskoodilla "201 Created". Käytetyimpiä HTTP-metodeja REST-rajapintojen yhteydessä ovat GET, POST, DELETE, PUT ja PATCH. Kuvassa 9 on kaavio web-rajapinnan toiminnasta. Kaaviossa asiakas lähettää pyynnön web-sovelluksen rajapinnalle (API), joka hakee pyydetyn tiedon sovelluksen tietokannasta ja vastaa sen jälkeen asiakkaalle pyydetyllä tiedolla.



Kuva 8. Web-sovellus, sovelluksen REST API -rajapinta sekä rajapintaa käyttävä asiakas. [Jan- sen 2011.]

REST-rajapinnan määrittelevät Fieldingin rajoitteet, jotka ovat vain suuntaviivoja toteu- tukselle. Konkreettisemmalla tasolla ne tarkoittavat seuraavia hyviä käytäntöjä [Maciej 2019]:

- Rajapinta käyttää salattua HTTPS-protokollaa. Web-käyttää HTTP-pro- tokollaa ja siksi on myös käytännöllistä käyttää HTTP:tä REST-rajapinnassa. REST ei kuitenkaan sido käyttämään tiettyä tiedonsiirtoprotokollaa [Fiel- ding 2000: 100].
- REST sallii useiden tiedostoformaattien käytön, mutta JSON on yleisesti paras vaihtoehto etenkin julkisissa web-rajapinnoissa. JSON on ohjelmoin- tikieliriippumaton formaatti. JSON on sekä koneluettava että ihmisen luet- tavissa.
- Resurssit nimetään substantiivien eikä verbien avulla. Resurssien sijainnit eli päätepisteet osoitetaan URL-osoitteilla. URL:ssä kategorioiden nimien tulee olla monikossa. Esimerkiksi `www.example.com/lomakkeet/1` eikä `www.example.com/lomake/tayta-lomake1`, missä numero "1" on resurssi ja "lomakkeet" on resurssin kategoria.
- Rajapinta käyttää useita HTTP-metodeja yhteen resurssiin, eikä esimer- kiksi vain POST-metodia tiedon hakuun ja tallentamiseen kuten SOAP-ra- japinta tekee.
- Rajapinta tukee virhekäsittelyä. Tämä voidaan toteuttaa valmiiden HTTP- status-koodien avulla. Status-koodin lisäksi vastaukseen olisi hyvä liittää tarkempi sanallinen kuvaus virheestä.
- Rajapinta tukee suodatusta, lajittelua ja tulosten näyttämisen rajaamista. Näitä ominaisuuksia käytetään lisäämällä kyselyn päätepisteen perään ky- seisen ominaisuuden hakuparametri. Esimerkiksi `"GET /users?country=FI"` voisi hakea ainoastaan Suomessa asuvat käyttäjät.

- Rajapinta tukee versiointia. Tämä mahdollistaa rajapinnan kehittämisen ominaisuuksilla, jotka eivät välttämättä ole taaksepäin yhteensopivia. Tällöin on hyvä ylläpitää myös vanhaa API-versiota.
- Rajapinnasta on tärkeää olla julkinen dokumentaatio. Dokumentaatiota käyttää kehittäjien lisäksi myös potentiaaliset käyttäjät, jotka haluavat nähdä, mitä rajapinnan avulla on mahdollista tehdä. Dokumentaation tulee sisältää päätepisteet ja niiden metodit, esimerkkejä pyynnöistä ja vastauksista, statuskoodit, käyttäjien tunnistus- ja autentikointitavat sekä mahdolliset rajoitukset.
- Palvelimen vastaus sisältää linkkejä muihin resursseihin (HATEOAS). Monessa rajapinnassa tälle ei välttämättä ole tarvetta tai sitä ei osata hyödyntää. Siksi se ei ole yleisesti käytössä.

4.7 SOAP-rajapinta

Microsoft julkaisi SOAP-protokollan (Simple Object Access Protocol) vuonna 1999. Julkaisun jälkeen Microsoft luovutti SOAP-protokollan Internet Engineering Taskforce -organisaatiolle (IETF) standardoitavaksi. SOAP on standardoitu tiedonsiirtoprotokolla, jota nykyään ylläpitää ja kehittää World Wide Web Consortium -yhteenliittymä (W3C). Tiedonsiirtoon SOAP-protokolla käyttää tarkasti määriteltä XML-viestiä, jota kutsutaan SOAP-viestiksi. Tiedonsiirtotapana SOAP-protokolla käyttää useimmiten HTTP-protokollaa, mutta SOAP-protokolla voi käyttää mitä tahansa sovelluskerroksen protokollaa kuten SMTP-, JMS-, TCP- tai UDP-protokollaa. [Monus 2018.]

SOAP-rajapinnat käyttävät SOAP-protokollaa tiedonsiirtoon. SOAP-rajapinta käyttää vain yhtä päätepistettä ja ainoastaan POST-metodia sen käyttämiseen. Viestin sisältö määrää, miten palvelin käsittelee viestin, toisin kuin REST-rajapinnassa HTTP:n metodi määrää viestin käsittelytavan. SOAP-rajapinnan toiminta perustuu eri nimisiin toimintoihin, esimerkiksi "luo käyttäjä". REST-rajapinta taas perustuu eri nimisten resurssien käsittelyyn, esimerkiksi "käyttäjät". [Monus 2018.]

SOAP-protokolla on suunniteltu hajautettujen ohjelmistojen kommunikoimiseen menetämättä tietoturvaa ja hallintaa. SOAP-protokolla mahdollistaa viestin reitittämisen ja käsittelyn usean eri palvelun kautta ennen lopullista vastaanottajaa. SOAP-protokolla tarjoaa turvallisen ja yhteensopivan yhteyden, luotettavan viestin toimituksen ja vikatilanteista palautumisen sekä tuen dynaamiselle palvelun löytämiselle. SOAP-rajapintojen

käyttökohteita ovat tyypillisesti suurien konsernien järjestelmät, pankkijärjestelmät ja vanhat taustajärjestelmät, niin sanotut legacy-järjestelmät. [Monus 2018; Rouse 2019.]

REST-rajapintaan verrattuna SOAP-rajapinnan arkkitehtuuri on raskaampi ja monimutkaisempi, jonka vuoksi SOAP-rajapinta käyttää enemmän kaistaa eikä ole niin suorituskykyinen kuin REST-rajapinta. SOAP-protokollassa on paljon standardoituja, sisäänrakennettuja turvallisuuteen, autentikointiin ja virhekäsittelyyn liittyviä ominaisuuksia. Nämä ominaisuudet tekevät siitä monimutkaisemman, mutta myös turvallisemman. [Monus 2018.]

SOAP-viesti on XML-dokumentti. XML on kaikkien ohjelmointikielten ymmärrettävissä, jonka vuoksi XML soveltuukin hyvin koneiden väliseksi kieleksi. XML mahdollistaa monimutkaiset rakenteet ilman tietosisällön muuttumista. SOAP-viestin rakenne on seuraavanlainen:

- *Kirjekuori (Envelope)* -elementin nimiavaruus määrittää viestin SOAP-viestiksi. Kirjekuori-elementti toimii SOAP-viestin juurielementtinä, joka kertoo viestin alun ja lopun.
- *Otsikkotiedot (Header)* sisältävät metatietoa viestistä, mikä kertoo, miten viestiä käsitellään ja reititetään. Otsikkotiedot eivät ole pakollinen osa viestiä.
- *Runko (Body)* sisältää viestin tietosisällön, joka on tarkoitettu lopulliselle vastaanottajalle. Vastauksen rungossa voi tietosisällön sijaan olla viestin käsittelyn virhetieto. Tietosisältöä ovat esimerkiksi Web Servicen kutsu, sen parametrit sekä muu XML-dokumentti. Runko on pakollinen osa viestiä.

Esimerkkikoodissa 1 on SOAP-viesti, joka noudattaa edellä kerrottua rakennetta.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <n:alertcontrol xmlns:n="http://example.org/alertcontrol">
      <n:priority>1</n:priority>
      <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
  </env:Header>
  <env:Body>
    <m:alert xmlns:m="http://example.org/alert">
      <m:msg>Pick up Mary at school at 2pm</m:msg>
    </m:alert>
  </env:Body>
</env:Envelope>
```

Esimerkkikoodi 1. SOAP-viesti. [SOAP Version 1.2 2007.]

SOAP-rajapinnan kuvaukseen käytetään usein WSDL-dokumenttia (Web Services Description Language). WSDL on XML-dokumentti, joka kuvaa Web Service -rajapinnan ja sen sijainnin. WSDL-dokumenttia ei ole tarkoitettu ihmisten vaan ohjelmien luettavaksi. WSDL on vaikeasti ymmärrettävä dokumentti, mutta se sisältää paljon hyödyllistä tietoa. WSDL-dokumentilla voidaan kuvata rajapinnan tietoelementit, toiminnot ja sijainti verkossa sekä tiedonsiirrossa käytettävien viestien muodon. WSDL-dokumenttia voidaan käyttää Web Service -rajapinnan käyttöönotossa tai WSDL-dokumentti voidaan luoda jo toteutetusta rajapinnasta. [Sormunen 2019.]

4.8 GraphQL-rajapinta

GraphQL-rajapintoja ei käsitellä kovin syvällisesti, sillä aiheesta voisi kirjoittaa oman insinööriyön, ja ainoastaan yhdellä selvitystyön verkkokauppa-alustalla on käytössä GraphQL-rajapinta. GraphQL-rajapinnoista löytyy lisää tietoa muun muassa Laajan insinööriyöstä [2019], jossa hän on vertaillut GraphQL- ja REST-rajapintoja.

GraphQL on uusin tulokas rajapintojen toteutustavoissa. GraphQL on alun perin Facebookin kehittämä kyselykieli. Vuonna 2012 rajapinnat eivät olleet enää riittäviä Facebookin mobiilisovellusten web-sisällön esittämiseen. Facebookin mobiilisovellukset kärsivät huonosta suorituskyvystä ja kaatuivat usein. Facebook kehitti GraphQL-kielen mobiilisovelluksiaan varten. Facebookin mobiilisovellukset ovat käyttäneet GraphQL:ää jo vuodesta 2012 lähtien. GraphQL:stä tehtiin avoin standardi vuonna 2015, mistä lähtien sen kehityksestä on vastannut GraphQL Foundation. GraphQL Foundation koostuu ympärimaailmaa olevista teknologia-alan yrityksistä. GraphQL-rajapintoja käyttävät muun muassa Airbnb, Audi, GitHub, PayPal, Shopify ja Twitter. [Byron 2015.]

GraphQL on kyselykieli rajapinnoille ja palvelimien ajonaikaisille ympäristöille kyselyjen suorittamiseksi. GraphQL ei ole sidottu mihinkään ohjelmointikieleen, tietokantaan tai tiedontallennusjärjestelmään, joten sitä voidaan käyttää halutun ohjelmointikielen ja jo olemassa olevan tietokannan kanssa. GraphQL-rajapinta tarjoaa kaiken tiedon yhden päätepisteen kautta. Tämä tapahtuu tyypillisesti HTTP:n kautta yhdestä URL-osoitteesta. Tämä on yksi ero REST-rajapintaan, joka tarjoaa tietoa monesta eri päätepisteestä. GraphQL perustuu graafiteoriaan, joka kuvaa tiedon toisiinsa kytkeytyneinä tietokalkioina. [GraphQL specification 2018.]

GraphQL-kielen merkittävä ominaisuus on se, että yhdellä kyselyllä voidaan hakea kaikki haluttu tieto yhdestä tai useammasta eri resurssista. Tämän mahdollistaa yhden päätepisteen käyttö ja haettavien tietokenttien määrittely kyselyssä. Tämä ominaisuus vähentää asiakassovelluksen ja verkon kuormitusta, sillä sen avulla vältetään turhan tiedon siirto. Toisaalta pitkät kyselyt saattavat kuormittaa palvelinta. Esimerkkikoodissa 2 on esimerkki GraphQL-kielen kyselyn (pyynnön) ja vastauksen yksinkertaisuudesta. Kysely on vasemmalla puolella ja vastaus oikealla. Kyselyssä pyydetään ainoastaan henkilön nimi ja pituus. Vastauksessa on ainoastaan juurikin nämä tiedot.

<pre>{ human(id: "1000") { name height } }</pre>	<pre>{ "data": { "human": { "name": "Luke Skywalker", "height": 1.72 } } }</pre>
--	--

Esimerkkikoodi 2. GraphQL-kysely argumenteilla ja vastaus. [Queries and Mutations 2019.]

GraphQL-rajapinta voidaan toteuttaa ilman versiointia. Rajapintaan voidaan lisätä uusia tietotyyppejä ja -kenttiä ilman, että rajapinta rikkoutuu. Rajapinnan rikkoutumisen estää se, että rajapinnan vastauksissa on vain kyselyssä määritetty tieto. Vanhentuvat kentät voidaan poistaa käytöstä ja piilottaa työkaluilta.

Lyhyellä perehtymisellä GraphQL vaikuttaa hyvin lupaavalta tekniikalta, jossa on omat etunsa etenkin mobiilisovelluksien kanssa.

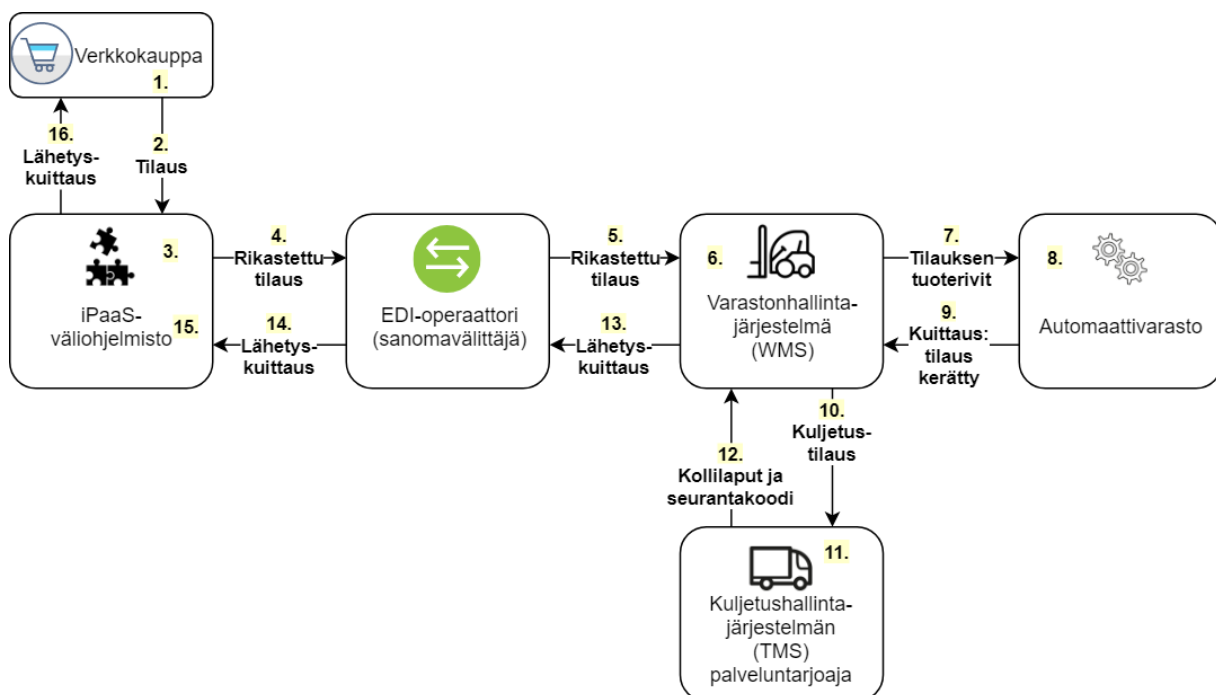
5 Verkkokauppa-alustojen integraatioiden selvitystyö

Selvitystyö koostuu neljästä osa-alueesta, jotka ovat verkkokauppoihin liittyvät prosessit, verkkokauppa-alustojen ominaisuudet, verkkokauppa-alustojen rajapinnat ja iPaaS-väliohjelmisto. Näitä osa-alueita tutkimalla pyritään selvittämään verkkokauppa-alustojen ja väliohjelmiston integraatioiden ongelmakohdat.

5.1 Verkkokaupasta tehdyn tilauksen käsittelyprosessi

Aluksi tarkastellaan HUBin asiakkaan verkkokaupasta tehdyn tilauksen käsittelyprosessia. Tämä käsittelyprosessi on kuvattu alla kronologisessa järjestyksessä sekä kuvan 9 kaaviossa. Käsittelyketju saattaa vaikuttaa turhan monimutkaiselta, mutta jokaisella järjestelmällä on oma osansa prosessissa. EDI-operaattoria voidaan luulla ylimääräiseksi komponentiksi tässä prosessissa, sillä se ei muuta tässä tapauksessa välitettävää tietoa ollenkaan. EDI-operaattoria käytetään siksi, että se on jo valmiiksi integroitu HUBin varastohallintajärjestelmään, ja sitä käyttämällä vältetään yksi työläs suora integraatio. Verkkokauppatilauksen käsittelyprosessi on seuraava:

- 1. Asiakas tekee tilauksen verkkokaupassa ja maksaa sen.
- 2. Väliohjelmisto noutaa maksetun tilauksen tiedot verkkokaupan rajapinnan kautta.
- 3. Väliohjelmisto muuntaa tilauksen XML-sanomaksi ja rikastaa tilauksen tietoja verkkokauppakohtaisilla tiedoilla sekä kuljetuksiin liittyvillä tiedoilla.
- 4. & 5. Väliohjelmisto lähettää tilauksen XML-muodossa eteenpäin EDI-operaattorin läpi varastohallintajärjestelmään.
- 6. Varastohallintajärjestelmä vastaanottaa tilauksen ja aikatauluttaa toimituksen.
- 7. Varastohallintajärjestelmä lähettää tilauksen tuoterivit automaattivaraston ohjelmistoon.
- 8. Automaattivarasto kerää tilauksen tuotteet.
- 9. Automaattivarasto ilmoittaa varastohallintajärjestelmälle, kun tilaus on kerätty.
- 10. Varastohallintajärjestelmä tekee kuljetustilauksen kuljetushallintajärjestelmän (TMS) palveluntarjoajalle.
- 11. TMS-palvelu vastaanottaa kuljetustilauksen ja tekee kuljetustilauksen kuljetusliikkeelle sekä muodostaa lähetyksen kollilaput ja seurantakoodin.
- 12. TMS-palvelu lähettää kollilaput ja seurantakoodin varastohallintajärjestelmälle.
- 13. & 14. Varastohallintajärjestelmä lähettää lähetyskuittauksen EDI-operaattorin läpi väliohjelmistoon.
- 15. & 16. Väliohjelmisto muuttaa tilauksen tilan lähetetyksi ja lisää tilaukselle seurantakoodin sekä omassa järjestelmässä että verkkokaupassa.



Kuva 9. Kaavio verkkokaupan tilausprosessista ja sen etenemisestä eri järjestelmissä.

5.2 Verkkokauppoihin liittyvät muut prosessit

Tilausprosessin lisäksi verkkokauppoihin liittyy tuotetietojen synkronointi verkkokaupasta varastohallintajärjestelmään sekä varastosaldojen synkronointi automaattivarastosta verkkokauppaan.

Tuotetiedot pitää viedä verkkokaupasta varastohallintajärjestelmään silloin, kun uusi tuote luodaan tai tuotteen tietoja päivitetään verkkokaupan hallinnassa. Tuotteista ei tarvitse viedä kaikkia tietoja. Kaikissa tuotteissa tarpeellisia tietoja ovat ainoastaan tuotekoodi, tuotenimi ja EAN-koodi. Tuotekoodi ja -nimi ovat tarpeellisia tuotteen tunnistamisessa, kun tuotteita käsitellään. EAN-koodi ei ole pakollinen tieto, mutta koodin lukeminen viivakoodinlukijalla nopeuttaa tuotteiden käsittelyä varastolla. Muita tarpeellisia tietoja tuotetietojen synkronoinnissa ovat tuotekohtaiset lisätiedot, joita voiva olla VAK-tuotteissa (vaaralliset aineet) VAK-koodi, elintarvikkeissa parasta ennen -päiväys ja toimituksissa EU:n ulkopuolella tullikoodi.

Tuotteiden varastosaldon eli varastomäärän tiedon synkronointi on yksi huomioitava asia. *Varastosaldotieto päivitetään aina varastohallintajärjestelmästä verkkokauppaan eikä toisinpäin.* Tähän on yksinkertainen syy, joka on, että varaston päässä on tuotteiden todelliset varastosaldot. Varastosaldoa lisätään, kun tavara on vastaanotettu ja vähennetään tilauksen toimituksessa. Varastosaldoa pitää päivittää palautuksien yhteydessä ja mahdollisesti inventaariota tehdessä.

5.3 Varastohallintajärjestelmän tarvitsemat tiedot

Varastohallintajärjestelmä tarvitsee verkkokauppatilauksesta erilaisia tietoja, joita ovat esimerkiksi vastaanottajan yhteystiedot, toimitusosoite, tuoterivit ja toimitustapa. Tuotetietojen synkronoinnissa verkkokaupasta varastohallintajärjestelmään tarvitaan tuotekoodi, tuotenimi ja EAN-koodi sekä tuotekohtaiset lisätiedot, joka voi olla esimerkiksi VAK-koodi. Edellä mainituilla tiedoilla pitää olla tietokentät verkkokaupassa. Lienee turha selvittää löytyykö selvityksen kaikista verkkokauppa-alustoista kaikki mahdolliset tietokentät, mutta on hyvä selvittää, pystyykö verkkokaupoissa lisäämään itsemääritettyjä attribuutteja eli lisätietokenttiä tuote-, asiakas- ja tilaustietoihin. Seuraavassa on selvitetty lisäkenttien lisäämisen mahdollisuus verkkokauppa-alustoihin:

- *WooCommerce-verkkokaupassa* voi lisätä tuotteisiin, asiakkaisiin ja tilauksiin itsemääritettyjä tietokenttiä (custom fields) lisäosien avulla tai koodia muokkaamalla. Lisäosat ovat yleensä maksullisia. Koodin muokkaamiseen löytyy verkosta ohjeita. Tuotteiden lisäkentät saa näkyviin sekä hallintapuolella että halutessa verkkokaupassa.
- *MyCashflowssa* ei pysty lisäämään itsemääritettyjä tietokenttiä. Tuotteisiin liittyvää lisätietoa voi lisätä ainoastaan valmiisiin tuotekuvaus-, tarkat tuotetiedot tai avainsanatkenttiin. Avainsanat ovat tägejä, joita voi käyttää tuotehaussa sekä hallinnan että kaupan puolella. Tägät sopivat tuotteiden kategorioimiseen.
- *Magento 2 -verkkokauppaan* saa lisättyä itsemääritettyjä lisäkenttiä tuotteisiin. Magentossa lisäkentät ovat nimeltään attribuutteja. Itsemääritetyistä lisäkentistä voi tehdä ryhmiä. Tuotetta luodessa valitaan lisäkenttien ryhmä, jolloin ryhmän lisäkentät tulevat tuotteelle näkyviin. Tilauksiin saa lisättyä lisäkenttiä koodia muokkaamalla. Tähän löytyy verkosta ohjeita.
- *Vilkas Suite -verkkokaupassa* voi itse määrittää lisäkenttiä asiakastietoihin sekä tuotetietoihin tuotekategoria kohtaisesti. Lisäkenttien tietoja voi käyttää hauissa suodattimena. Lisäkenttien lisääminen on hyvin selkeästi toteutettu ja onnistuu ilman ohjeita.

- *Shopify*ssa onnistuu lisäkenttien lisääminen lisäsovelluksen avulla, esimerkiksi *Shopify FD*:llä [ShopifyFD 2019], joka on Chrome-selaimen laajennus. *Shopify*ssa lisäkenttiä kutsutaan metakentiksi (metafield). Metakenttiä voi lisätä tuotteille, asiakkaille ja tilauksille. Lisäkentän saa halutessa näkyviin verkkokaupan julkisivuun muokkaamalla teematiedostoa. Metakentillä on joitakin rajoituksia. Metakenttiä ei voi käyttää hallintapaneelissa haku- tai suodatusarvoina. *Shopify*n analytiikka ja raportointi ei sisällytä metatietoja, eikä *Shopify*n skriptit tue metakenttiä. *Shopify*ssa on myös niin kutsuttuja tägejä, joita voi suoraan käyttää *Shopify*n hallintapaneelissa haku- ja suodatusarvoina. Tägit ovat pilkulla erotettuja merkkijonoja. Lisäsovellukset tukevat natiivisti tägejä. Tägejä voi käyttää tuotteiden, asiakkaiden ja tilauksen kategorioimiseen, esimerkiksi tägillä voisi merkitä tuotteen valmistajan.

5.4 Automaattivaraston tarvitsemat tiedot

Automaattivaraston tarvitsemia tietoja ovat tuotteiden paino- ja mittatiedot, sekä halutessa sarjanumerot.

Automaattivarasto käyttää painotietoja tuotteiden keräilyvaiheessa keräilyn virheettömyyden tarkistamiseksi. Paino on yksi verkkokauppojen tuotetietojen kenttä, *mutta sitä ei saa synkronoida verkkokaupasta varastohallintajärjestelmään*. Tuotteiden paino mitataan varastolla erityisellä mittalaitteella, jotta painotieto olisi riittävän tarkka automaattivaraston toiminnalle. Mittalaite skannaa tuotteiden mitat noin 1 millimetrin tarkkuudella ja punnitsee tuotteet noin 0,1 gramman tarkkuudella. Mittalaitteesta mittatiedot tallentuvat varastohallintajärjestelmään, josta ne välittyvät automaattivaraston järjestelmään. Automaattivarasto käyttää tuotteiden mittatietoja laskentaan, minkä avulla automaattivarasto määrittää, kuinka paljon tuotteita mahtuu varastopaikkoihin ja kerättävään laatikkoon.

Jos HUBin automaattivaraston asiakas haluaa tuotteiden sarjanumeroseurattavuutta, niin silloin myös sarjanumero kuuluu tarvittaviin tietoihin. Sarjanumerot luetaan tuotetta kerättäessä automaattivarastosta sekä asiakkaan halutessa tuotteen vastaanottovaiheessa. Jos tuotteen sarjanumero on luettu jo vastaanottovaiheessa, sillä voidaan myös validoida tuotteiden keräystä.

Loppujen lopuksi automaattivarasto itsessään ei aseta montaa ehtoa verkkokauppojen integraatiolle. Automaattivaraston kanssa pitää kuitenkin huomioida, että soveltuuko verkkokaupan tuote sinne varastoitavaksi fyysisten ominaisuuksien kannalta.

5.5 Verkkokauppa-alustojen tarpeelliset ominaisuudet

Verkkokauppa osaa poistaa tuotteen myynnistä, jos tuotetta ei ole varastossa. Osatoimitukset ovat hankalia, siinä mielessä, että osatoimitusten hallinnan pitäisi tapahtua verkkokaupassa. Kaikissa verkkokaupoissa ei kuitenkaan ole mahdollista hallita osatoimituksia, joten, jos halutaan yhtenäinen käsittelyprosessi eri verkkokauppojen tilauksille, niin osatoimitusten synty kannattaa estää kokonaan. Osatoimituksien synty voidaan estää poistamalla tuote myynnistä eli verkkokaupan asiakkaan näkyvistä silloin, kun tuotetta ei ole varastossa. Osatoimitukset pitää kuitenkin miettiä verkkokauppiaskohtaisesti, sillä joissain verkkokauppa-alustoissa osatoimitus on mahdollista kuten esimerkiksi Vilkas Suite -verkkokaupassa.

Verkkokaupassa tilauksen tila muuttuu käsittelyn eri vaiheissa. Tämä on yksi prosessien automatisoinnin edellytys. Tilauksen tilaa voidaan hyödyntää eri toiminnoissa. Tilauksen tilaa voidaan käyttää esimerkiksi ehtona, jolloin väliohjelmisto hakee tilauksen verkkokaupasta käsiteltäväksi eteenpäin. Tällöin ehtona on, että tilauksen tilana on maksettu.

Maksetun tilauksen tuoterivien muuttaminen on mahdollista jälkikäteen verkkokaupan hallinnassa. Tämä ei ole haluttu ominaisuus, sillä se hankaloittaa tilauksen käsittelyprosessia. Tilauksen muuttaminen jälkikäteen aiheuttaisi sekaannuksia tilauksen käsittelyssä. Tilaus voisi olla jo kerätty ja odottamassa kuljetusta, kun siihen tehtäisiin muutoksia. Tällaisen tilanteen selvittäminen olisi aikaa vievää ja hankalaa. Siksi on parempi estää kokonaan mahdollisuus tällaiseen sekaannukseen estämällä tilauksen muuttaminen jälkikäteen.

Tilauksen seurantakoodi olisi hyvä saada näkyviin verkkokaupan hallinnassa, jotta verkkokaupan myyjä pääsee siihen helposti käsiksi, ja jotta seurantakoodi voidaan lähettää asiakkaalle sähköpostitse tilausvahvistuksen yhteydessä. Verkkokaupoissa, joissa on tietokenttä seurantakoodille, seurantakoodi on usein linkitetty tilausvahvistuksen sähköpostissa näkyväksi.

Noutopisteen valinta verkkokaupan kassalla ja noutopisteen tallennus tilaustietoihin on tärkeä ominaisuus, jota käyn läpi seuraavassa luvussa.

Edellä käsitellyt ominaisuudet löytyvät verkkokaupoista taulukon 4 mukaisesti.

Taulukko 4. Verkkokauppojen ominaisuuksia.

	WooCommerce	MyCashflow	Magento 2	ePages/ Vilkas Suite	Shopify
Ei myyntiä, jos ei varastossa	On valittavissa	On valittavissa	On valittavissa	On valittavissa	On valittavissa
Tilauksen eri tilat (status)	On / muokattavissa lisäosalla	On	On / muokattavissa	On	On
Maksetun tilauksen rivien muuttaminen jälkikäteen	Ei	* On mahdollista	Ei	* On mahdollista	* ** Tulossa 2019/2020
Tilauksen seurantakoodi verkkokaupassa	Metatieto tai lisäsovellus	On	On	On	On
* Tilausrivien muuttamisesta asiakkaalle lähetetään hyvitys- tai lisälasku.					
** Ominaisuuden on kerrottu tulevan vuoden 2019 lopussa.					

5.6 Noutopiste-sovellus

Yksi ilmennyt ongelma liittyy tilauksiin, joissa verkkokaupan asiakas noutaa tuotteensa kuljetusliikkeen noutopisteestä. Ongelmallista tässä on se, että jos verkkokauppa-alustaan ei löydy sopivaa noutopisteen tarjoavaa lisäosaa valmiina, niin se pitää kehittää, ja se voi olla kallista ja aikaa vievää.

Verkkokauppoihin löytyy melko paljon erilaisia lisäosia, joiden avulla verkkokaupan kassalla asiakkaalle voidaan tarjota noutopisteitä. Osa lisäosista on kuljetusliikekohtaisia, ja osa lisäosista tarjoaa Unifaunin tai jonkin muun toimijan palvelusta löytyvät kuljetustavat. Kuljetusliikekohtaiset sovellukset tarjoavat vain kyseisen kuljetusliikkeen noutopisteitä. Unifaunia käyttävät sovellukset pystyvät tarjoamaan useamman kuljetusliikkeen noutopisteitä verkkokaupan asiakkaalle. Suurin osa noutopistelisiä lisäosista tekee kuljetustilauksen suoraan jonkin toimijan, esimerkiksi Unifaunin järjestelmään, jolloin kuljetustilauksesta palautuu kollilaput ja seurantakoodi takaisin verkkokauppaan. Usein 3PL-yritysten

varastonhallintajärjestelmät tekevät kuljetustilaukset itse tilauksen pakkaamisen yhteydessä. Noutopistetilauksen kuljetustilauksen tekemiseksi tarvitaan toimitustavan tiedoissa noutopisteen tunnus eli noutopiste-id. Noutopistelisäosassa tarvitaan siis ominaisuus, joka tarjoaa noutopisteitä asiakkaalle ja tallentaa asiakkaan valinnan tilauksen tietoihin sisältäen tuon noutopiste-id:n. Tämän noutopiste-id:n pitää olla lisäksi haettavissa verkkokaupan rajapinnan kautta.

Verkkokauppojen noutopistelisäosat käyttävät enimmäkseen joko Unifaun PickUpLocator-palvelua tai Unifaun DeliveryCheckOut-palvelua. Mikäli on tarvetta vain noutopisteiden haulle, niin Unifaun PickUpLocator-palvelu tarjoaa sen. Unifaun DeliveryCheckOut on huomattavasti kalliimpi palvelu. Näin ollen voi olla kannattavaa etsiä verkkokauppoihin PickUpLocator-palvelua käyttäviä lisäosia.

Unifaun PickUpLocator on Web Service -palvelu, joka hakee lähimmät noutopisteet maakoodin, postinumeron ja kuljetusliikkeen tunnuksen perusteella. PickUpLocator-palvelussa on seuraavien kuljetuspalveluiden noutopisteet: Matkahuolto Lähellä-paketti, Posti Economy ja Posti SmartPOST, PostNord MyPack sekä DB Schenker Noutopistepaketti.

Shopify-verkkokauppaan löytyi kuitenkin valmis noutopistesovellus, Nordic Shipping App [2019], joka sopii sellaisenaan HUBin käyttöön. Sovelluksen ilmainen versio antaa verkkokauppa-asiakkaan valita noutopisteen ja tallentaa noutopiste-id:n tilauksen metatietoihin. Metatiedoissa oleva noutopiste-id on haettavissa muiden tilaustietojen tavoin Shopify:n rajapinnan kautta.

Vilkas Suite -verkkokauppoihin löytyy Posti SmartShip- ja Unifaun-lisäosat [2019]. Unifaun-lisäosa on kiinnostavampi, sillä se tarjoaa usean kuljetusliikkeen palvelut kerralla. Unifaun-lisäosa vaatii Unifaun ERP Connect -palvelun tilauksen sekä käytettäessä noutopistehakua Unifaun PickUpLocator -palvelun tilauksen. Vilkkaan verkkokaupoissa PickUpLocator-palvelu pitää itse tilata Unifaunilta. Palvelu on maksullinen, mutta melko edullinen. Vilkkaan asiakastuesta kysymällä selvisi, että Unifaun-lisäosa tallentaa noutopiste-id:n UnifaunPickDropPointId-attribuuttiin. Kyseinen attribuutti ei kuitenkaan ole tällä hetkellä saatavissa rajapinnan kautta, mutta Vilkas voi tarvittaessa ottaa kehityksen työn alle.

WooCommerce-verkkokauppaan löytyy WooCommerce Noutopistehaku -lisäosa [2019], jonka on tehnyt Markup.fi. Lisäosa tarjoaa haun seuraavien kuljetusliikkeiden noutopisteille: Posti, Matkahuolto, DB Schenker ja PostNord. Sovelluksen saa käyttöön maksamalla 49 euron kertalisenssimaksun. Lisäosa tallentaa noutopisteen tiedot, myös noutopiste-id:n, ja nämä tiedot voi hakea rajapinnan kautta. Tähän löytyy suoraan ohjeet kehittäjän sivuilta [WooCommerce Noutopistehaku / Ohjeet 2019]. Sovelluksen kehittäjä vastasi nopeasti sähköpostikyselyyni, ja vaikuttaa aktiiviselta sovelluksen tuessa ja kehityksessä WooCommerce Noutopistehaku -lisäosan [2019] verkkosivuilla olevien asiakaspalautteiden perusteella.

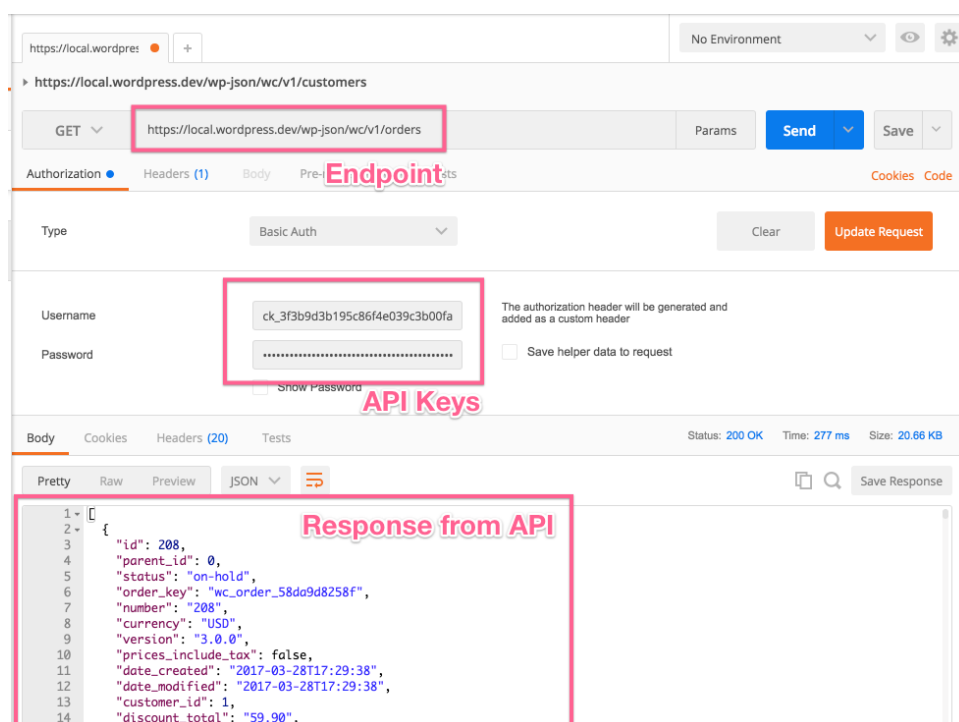
Magento 2 -verkkokauppaan löytyy kuljetuslisäosia muun muassa saman tekijän toimesta kuin WooCommerceen. Magento 2 verkkokauppaan löytyy seuraavat heidän tekemät lisäosat: Shipit, Posti SmarthShip, Matkahuolto, GLS ja DB Schenker. Kehittäjän mukaan lisäosissa voi käyttää pelkkää noutopisteen hakua ilman kuljetustilauksen tekemistä. Tämä ei ole suositeltavaa Shipit-lisäosan kanssa ilman sopimusta asiasta heidän kanssaan. Edellä mainitut lisäosat maksavat kukin 249 euroa. Magento 2 on yleisesti ottaen kallis verkkokauppa-alusta, joten voisi olettaa verkkokaupan toimittajan tekevän myös paljon räätälöintejä ja integraatioita halutuilla ominaisuuksilla. Magentoa toimittavat kumppanit mainitsevat integroinneista Unifaunin kanssa, mutta heidän verkkosivuiltaan ei sen tarkempaa tietoa löydy. Jos 3PL-yrityksen asiakkaaksi päätyy Magentoa käyttävä verkkokauppa, niin kannattanee selvittää verkkokaupan toimittajalta, löytyykö heiltä valmiiksi sopiva integraatio Unifaun kanssa.

MyCashflow-verkkokauppaan löytyy Unifaun eCom LITE -lisäosa, jossa on myös noutopisteiden haku. Lisäosa käyttää Unifaun PickUpLocatoria. MyCashflow veloittaa lisäosan käytöstä 0,40 euroa jokaisesta lähetyksestä ilman kuukausimaksua. MyCashflow'n lisäosan tarkemman toiminnan selvittämiseen ei valitettavasti selvitystyön aika riittänyt.

5.7 Verkkokauppa-alustojen rajapinnat

Verkkokauppa-alustojen rajapinnoista löytyy hyvät dokumentaatiot kunkin alustan verkkosivulta. Rajapintojen päätepisteet ja toiminnallisuudet on selkeästi kerrottu kustakin alustasta. SOAP-rajapinnoille löytyy myös koneluettavat WSDL-dokumentit. Postman-sovellus on hyvä tapa tutustua ja testata verkkokauppojen rajapintoja. Postmanilla voi

testata REST-rajapinnan ja API-tunnusten toimivuuden. Jos rajapinta ja tunnistautuminen toimii Postman-sovelluksella, mutta ei kehitettävällä asiakassovelluksella, niin vika on asiakassovelluksen koodissa. Kuvassa 10 on Postman-sovelluksen käyttöliittymä. Endpoint-kenttään syötetään haettavan tiedon päätepisteen URL-osoite ja API Keys -kenttiin API-avaimet. Ennen päätepistettä on kyselyn metodi, joka kuvassa 10 on GET. Kyselyn vastaus näkyy alimmassa ikkunassa (Response from API). Postman-sovellus toimii myös SOAP-rajapinnan testaamiseen. Tällöin kyselyyn pitää muokata oikean muotoinen SOAP-viesti, joka myös onnistuu Postman-sovelluksessa.



Kuva 10. Kuvassa Postman-sovelluksen käyttöliittymä ja esimerkki kysely WooCommerce-kauppaan. Sovellusta voi käyttää esimerkiksi testaamaan verkkokaupan rajapinnan toimivuuden. [Shellberg 2019.]

5.7.1 WooCommerce ja WordPress

WooCommerce on täysin integroitu WordPressiin ja käyttää WordPressin REST API -rajapintaa. Vanhemmissa WooCommerce-versioissa (ennen versiota 2.6) on ollut oma REST API -rajapinta, jota on kutsuttu Legacy API -rajapinnaksi. Legacy API -rajapinta on nyt poistumassa kokonaan. Jokaisella WordPressin REST-rajapinnan versiolla on oma URL-osoitteensa. Uusin versio on v3. Rajapintaa käytettäessä, kesto-linkkien muoto

tulee vaihtaa niin sanottuun hakukoneystävälliseen muotoon (pretty permalinks) Wordpressin hallintapaneelistä, jotta rajapinnan pääte pisteet toimivat. [Woo Rest API 2019.]

REST API -rajapinnan vastaukset ovat JSON-formaatissa. Kyselyt käyttävät JSON-formaattia resurssien luomiseen ja päivittämiseen. Wordpressin REST-rajapinnassa on vain neljä statuskoodia virhetilanteille: 400 Bad Request, 401 Unauthorized, 404 Not Found ja 500 Internal Server Error. Onnistuneen pyynnön vastauksen status-koodi on "200 OK". Rajapinta tukee kyselyparametrejä. Rajapinta tukee myös JSONP-formaattia (JSON with Padding). JSONP-formaatti pitää määritellä kyselyn otsikkotiedoissa. [Woo Rest API 2019.]

WooCommercessa on Webhook-ominaisuus. Webhook lähettää halutusta tapahtumasta ilmoituksen valittuun URL-osoitteeseen. Tällainen tapahtuma voi olla tilauksen, tuotteen tai asiakkaan lisääminen, muokkaaminen tai poistaminen.

5.7.2 MyCashflow

MyCashflow:ssa on REST API -rajapinta. MyCashflow tarjoaa rajapinnan käytön Advanced-, Pro- ja Enterprise-palvelupaketeissa, mutta ei halvimmassa Basic-paketissa. Rajapinnan käyttö ei aiheuta lisäkustannuksia.

MyCashflow'n rajapinta on versioitu. Rajapintaversiot ovat yhteensopivia saman sarjan versioissa taaksepäin, esimerkiksi v1.1.2-versio on yhteensopiva v1-version kanssa. MyCashflow'n rajapinnan vastaukset ovat JSON-formaatissa ja kysytyn resurssin data on data-elementin sisällä. Virhetilanteessa vastauksessa on otsikkotiedoissa virheen status-koodi ja rungossa virheen aiheuttama elementti ja sanallinen kuvaus virheestä. Rajapinnasta löytyy webhook-ominaisuus. [MyCashflow API 2019.]

5.7.3 Magento 2

Magentossa on REST API- ja SOAP-rajapinnat. Magento 2:ssa REST- ja SOAP-rajapinnat kattavat samat toiminnallisuudet. REST-rajapinta on kuvattu OpenAPI-määrittelyn

mukaisella skeemalla. Skeema mahdollistaa rajapinnan ominaisuuksien koneluettavuuden. SOAP-rajapinta on kuvattu koneluettavalla WSDL-dokumentilla. Magento luo WSDL-dokumentin vain rajapinnan palveluille, joita käytetään.

Magenton rajapintojen vastaus on joko XML- tai JSON-formaatissa. Vastausmuoto määritellään pyynnön otsikkotiedoissa. Magenton rajapinta tukee vastauksen tietokenttien suodatusta eli rajapinnasta voidaan hakea vain haluttu tieto, jolloin verkon kuormitus vähenee etenkin mobiilikäytössä. Rajapinnoista löytyy Event Observer -ominaisuus, joka on hieman webhook-ominaisuutta vastaava ominaisuus. Event Observer seuraa haluttuja verkkokaupan tapahtumia, joiden seurauksena voidaan ajaa oma koodipätkä. [Getting Started with Magento Web APIs 2019.]

5.7.4 ePages / Vilkas Suite

Vilkas Suite -verkkokauppa tarjoaa SOAP-rajapinnan käytön kalleimmissa Pro-palvelupaketeissa sekä täysin räätälöidyssä Go-palvelupaketissa. Rajapinta on maksullinen lisäpalvelu, joka maksaa 50 euroa kuukaudessa. SOAP-rajapinta on aina yhteensopiva taaksepäin. Uuden verkkokauppaversion päivitys ei vaadi rajapinnan päivittämistä, mutta se voi olla hyödyllistä rajapinnan uusien ominaisuuksien vuoksi. Rajapinnan palvelut on kuvattu WSDL-dokumenteilla.

5.7.5 Shopify

Shopifysta löytyy REST API- ja GraphQL -rajapinnat.

REST-rajapinnan vastaukset ovat JSON-formaatissa. Shopify luo automaattisesti API-avaimet asiakassovellukselle, kun Shopifyihin lisätään niin sanottu yksityinen sovellus. Yksityinen sovellus on vain yhteen verkkokauppaa kehitetty asiakassovellus, joka ei ole julkisesti saatavissa Shopifyyn sovelluskaupan kautta.

GraphQL-rajapinta tarjoaa REST-rajapinnan toiminnallisuudet sekä enemmän yhden päätepisteen kautta. GraphQL-rajapintaa voi testata Shopifyyn omalla GraphQL-sovelluksella, joka asennetaan Shopify-verkkokauppaan, ja jota käytetään hallintapaneelin kautta.

5.7.6 Yhteenveto verkkokauppa-alustojen rajapinnoista

Taulukossa 5 on esitetty verkkokauppojen rajapintatyypit ja perusominaisuudet.

Taulukko 5. Verkkokauppa-alustojen rajapinnat.

	WooCommerce/ WordPress	MyCashflow	Magento 2	ePages/ Vilkas Suite	Shopify
REST-rajapinta	X	X	X	-	X
SOAP-rajapinta	-	-	X	X	-
GraphQL-rajapinta	-	-	-	-	X
Tiedostoformaatti	JSON	JSON	JSON / XML	XML	JSON
Webhook-ominaisuus	X	X	*X	X	X
Dokumentaatio	Tekstikuvaus	Tekstikuvaus	Rest: OpenAPI skeema SOAP: WSDL	WSDL	REST: tekstikuvaus GraphQL: Skeema
* Magentossa on Event Observer -ominaisuus, joka vastaa webhookkia.					

5.8 iPaaS-väliohjelmistot

HUBin verkkokauppojen kanssa käytettävä väliohjelmisto on iPaaS-alusta (integration Platform as a Service). iPaaS-alustat ovat pilvipohjaisia integraatioalustoja, jotka toimivat järjestelmien ja sovellusten välikerroksina ja mahdollistavat niiden välisen reaaliaikaisen tiedonsiirron. iPaaS-alustalla voidaan toteuttaa integraatioita yrityksen järjestelmien ja pilvipalveluiden välille tai niiden eri yhdistelmille. Integraatioalustat tarjoavat seuraavia toiminnallisuuksia jopa monimutkaisten integraatioiden toteuttamiseksi; siltana toimiminen eri protokollien välillä, viestien välittäminen ja reitittäminen, dataformaattien kääntäminen ja muuntaminen, palvelujen visualisointi, erilaisten sovitustekniikoiden käyttäminen, integraatioiden hallinta ja skaalautuminen tarpeen mukaan. [Inside the iPaaS 2019: 5–6.]

iPaaS-palvelut ovat yleistyneet yhdessä muiden pilvipalvelujen kanssa 2010-luvulla. iPaaS-alustat on suunniteltu integraatioiden helpottamiseksi ja ne sisältävät usein graafisen käyttöliittymän integraatioiden toteuttamiseksi. iPaaS-palveluita löytyy niin suurien monialaisten organisaatioiden kuin tarkemmin rajattujen toimialakohtaisten toimijoiden

integraatioiden toteuttamiseen. Tunnetuimpia iPaaS-alustoja tarjoavat yritykset ovat Informatica, Dell Boomi, MuleSoft, Snaplogic, Oracle, Jitterbit, Microsoft ja SAP.

iPaaS-palveluiden yleisiä etuja ovat seuraavat [Thoo ym. 2019]:

- Pienet aloituskustannukset.
- Integraation nopea toteutus ja käyttöönotto.
- Pilvipalvelua käytettäessä ei ole tarvetta ostaa ohjelmistoa eikä kalliita lisenssejä. Palveluntarjoaja hoitaa palvelimien ja ohjelmiston ylläpidon, päivitykset ja muut tukipalvelut. Palvelun kehittyessä uudet ominaisuudet tulevat kaikkien palvelua käyttävien saataville. Myös jonkun toisen asiakkaan vuoksi tehdyt ominaisuudet tulevat muiden saataville.
- Skaalautuvuus. Palvelu skaalautuu viestiliikenteen määrän mukaan.
- Reaaliaikaisuus. Tiedonsiirron reaaliaikaisuuden mahdollistaa sanoma-, tapahtuma- ja pyyntöperustainen liikenne ja niiden sekoitus. Toki iPaaS-alusta pystyy ajamaan myös perinteiset ajastetut massa-ajot.
- Monipuolisuus datan käsittelyssä. Alusta pystyy käsittelemään kaikenlaisia dataformaatteja ja muuntamaan niitä toiseen formaattiin. Tietoa voidaan myös validoida, poimia, yhdistää ja rikastaa.
- Tietoturvallisuus. Pilvipalvelu on palveluntarjoajan ydinbisnestä, joten tietoturva on kärjessä heidän tärkeysjärjestyksessään.
- iPaaS-palvelut tarjoavat usein monipuolista monitorointia ja raportointia. Palvelu osaa ilmoittaa virhetilanteista tukitiimille.
- Työkalut ohjelmistojen välisiin integraatioihin mahdollistavat eri ohjelmistojen välisen tiedonsiirron, palvelujen käytön ja liiketoimintaprosessien automatisoinnin.
- Työkalut tietovarastojen integraatioihin mahdollistavat tietovarastojen synkronoinnin, datan siirtämisen toiseen sijaintiin, ja datan kokoamisen ja yhdistämisen eri sijainneista.
- Rajapintojen hallinta eli rajapintojen luonti, käyttöönotto ja monitorointi.

5.8.1 HUBin valitsema iPaaS-alusta

HUBin valitsema iPaaS-palvelu on suunniteltu 3PL- ja 4PL-logistiikkayrityksille, jotka haluavat kasvaa verkkokauppojen markkinoilla. Palvelu mahdollistaa jopa satojen verkkokauppojen integraation ja niistä tulevien tilausten käsittelyn yhden käyttöliittymän kautta. HUBin valitsema iPaaS-alusta mainostaa itseään joustavana palveluna ja edullisilla in-

tegraatiokustannuksilla. Pienempien verkkokauppojen kanssa edulliset integraatiokustannukset ovat merkittävä asia. Aikaisemmin suuret integraatiokustannukset ovat rajoittaneet nämä pienemmät verkkokauppa-asiakkaat pois.

Kyseinen iPaaS-alusta mahdollistaa tilausten tarkastelun palvelussa ilmaiseksi. Yleensä vastaava palvelu on maksullinen varastohallintajärjestelmässä. Tarkasteltavia tietoja voivat olla esimerkiksi tilauksen käsittelyn tila, tuotteiden varastomäärien tiedot, tilausten ja palautusten määrät. Tilauksen tilan tarkistaminen antaa verkkokaupan myyjälle mahdollisuuden itse tarkistaa tilauksen tila iPaaS-palvelusta, eikä hänen tarvitse soittaa tai laittaa sähköpostia logistiikkapalvelun tarjoajalle. Näin verkkokaupan myyjällä on reaaliaikaiset tiedot ja muiden ihmisten työmäärä vähenee.

HUBin käyttämän iPaaS-palvelun ominaisuuksia ovat seuraavat:

- Useita verkkokauppoja pystyy hallitsemaan yhtä helposti kuin yhtä.
- Alustan joustavuus mahdollistaa monimutkaisetkin integraatiot helposti. Esimerkiksi verkkokauppa tilauksissa iPaaS-palvelu hakee verkkokaupasta tilauksen tiedot, rikastaa tilauksen tietoja verkkokauppiaan tiedoilla ja lopulta yhdistää tiedot HUBin hyväksymään muotoon XML-dokumentiksi.
- Yhden verkkokaupan kanssa tehtyä integraatiomallia voidaan hyödyntää muiden verkkokauppojen kanssa, jolloin uusien verkkokauppojen integraation toteutus nopeutuu.
- iPaaS-palvelu skaalaa itsensä verkkokauppojen ja tilausten määrän mukaan.
- Palvelu päivittyy automaattisesti.
- Palveluun pääsee mistä tahansa verkkoselaimen kautta.

Sähköpostikyselyn perusteella kyseinen iPaaS-palvelu on toteuttanut integraatioita WooCommerce- ja Magento-verkkokauppa-alustoille. iPaaS-alustasta saatavien tietojen perusteella näyttää siltä, että alustalla on valmiudet integraatioihin kaikkien verkkokauppojen kanssa. HUBin järjestelmän ja verkkokauppojen integraatioiden haasteena saatetaan pikemminkin olla verkkokaupan jonkin ominaisuuden puuttuminen tai rajapinnan puute tietyn tietokentän hakemiseksi kuin iPaaS-alustan puute.

5.8.2 Väliahjelmiston käytön kipupisteet

Selvitystyötä tehdessä esille tuli kysymyksiä integraation kipupisteisiin liittyen. Suurim-
paan osaan kysymyksiin on myös löytynyt vastaus selvitystyön aikana. Seuraavassa on
esille nousseet kysymykset ja vastaukset niihin:

- *Rikkooko muutokset verkkokauppojen rajapinnoissa integraation? Seuraako väliahjelmisto rajapintojen muutoksia?* Väliahjelmisto monitoroi verkkokauppojen rajapintoja ja seuraa niiden versiointia. Muutokset huomataan ja niitä korjataan usein.
- *Onko väliahjelmiston monitorointi luotettavaa? Huomataanko virhetilanteet?* Väliahjelmisto käyttää palvelun toiminnan ja integraation monitorointiin Prometheus-alustaa. Prometheus-alusta on vuonna 2012 julkistettu avoimen lähdekoodin monitorointialusta. Väliahjelmisto kertoo huomavansa virhetilanteen.
- *Onko väliahjelmisto tietoturvallinen?* Yleisesti ottaen pilvipalvelut ovat tietoturvaltaan korkeatasoisia, sillä pilvipalvelut käyttävät tietoturvaan paljon resursseja. Tietoturva on tai ainakin pitäisi olla erittäin korkealla pilvipalvelun tarjoajien tärkeysjärjestyksessä.
- *Integraatiotestauksen pitää olla prosessin kaikkien osien kattava, jotta esille tulee mahdolliset ongelmat jo testausvaiheessa.* Integraation testaus on lopulta HUBin vastuulla ja HUBilla on hyvässä tiedossa kyseinen prosessi. Tämän osalta kaikki on kunnossa.
- *Verkkokauppojen rajapinnoissa on tietyt rajat rajapinnan kyselyiden määrälle. Voiko tämä olla ongelma?* Väliahjelmistolla on kyselyille jonotusjärjestelmä. Jos kyselyraja tulee vastaan, niin kyselyitä viivästetään.

6 Yhteenveto

Insinööriyö aloitettiin valitsemalla viisi käytetyintä verkkokauppa-alustaa Suomessa viimeisen kolmen vuoden aikana ja tutustumalla näihin verkkokauppoihin. Tämän jälkeen tutkittiin web-ohjelmistojen, joita myös verkkokaupat ovat, integraatioita, rajapintatyyppejä ja niihin liittyviä tekniikoita. Lopulta tarkasteltiin prosesseja, joita on verkkokauppojen ja HUBin järjestelmien välillä. Näihin prosesseihin peilaten tutkittiin verkkokauppojen ominaisuuksia ja integraation tarpeita tiedon suhteen.

Insinööriyön alussa esitettiin seuraavat tutkimuskysymykset:

- Löytyykö verkkokauppa-alustoista riittävät rajapinnat integraatioihin?
- Mikä on HUBin käyttämän väliohjelmiston valmius integraatioihin verkkokauppa-alustojen kanssa?
- Löytyykö verkkokauppa-alustoista riittävät tietokentät, joita tarvitaan tilausten käsittelyyn?
- Onko verkkokauppa-alustoissa riittävät toiminnallisuudet toimimaan automatisoidusti integroituna varastohallintajärjestelmään?

Verkkokauppa-alustat ovat kehittyneet paljon viime vuosina ja niissä on huomioitu tarpeet integraatioille muihin ohjelmistoihin kehittämällä rajapintoja. Etenkin suuremmissa verkkokauppaohjelmistoissa rajapinnat ovat monipuolisia ja tarjoavat päätepisteen lähes kaikkeen saatavaan tietoon. MyCashflow oli selvitystyön ainut pienempi verkkokauppaohjelmisto, joten sen rajapinnan kanssa voi ehkä ilmetä haasteita. Myös Vilkkaan Suite-verkkokaupassa ilmeni yksi puute rajapinnan kanssa noutopistelisiä osaa käytettäessä. Vilkaassa noutopiste-id:n hakeminen ei ole toistaiseksi mahdollista rajapinnan kautta. Havaitut puutteet rajapinnoissa olivat kuitenkin melko vähäisiä.

HUBin käyttämä iPaaS-väliohjelmisto vaikuttaa riittävän joustavalta ja monipuoliselta verkkokauppojen integraatioiden toteuttamiseen ja hallinnoimiseen. Tätä tukee se, että kyseinen iPaaS-ohjelmisto on suunniteltu 3PL-yrityksien tarpeisiin. Väliohjelmistosta ei löytynyt paljoa tietoa ja dokumentaatiota, mutta ohjelmiston kehittäjät ovat vastanneet kaikkiin heille esitettyihin kysymyksiin.

MyCashflow-verkkokauppa on selvitystyön ainut ohjelmisto, johon ei voi lisätä itsemääritettyjä tietokenttiä, esimerkiksi VAK-koodia, tullikoodia ja parasta ennen -päiväystä.

Näitä tietoja vaativien tuotteiden osalta voi ilmetä haasteita. HUBin käyttämän väliohjelmiston avulla pystyy myös hallitsemaan tuotteita, joten ehkä siellä tuotteisiin voisi lisätä tarvittavat lisätiedot. Se olisi toki työlästä suuren tuotevalikoiman kanssa. Muissa selvityksen verkkokauppaohjelmistoissa lisäkenttiä saa lisättyä suoraan hallintapaneelistä, lisäosan avulla tai koodia muokkaamalla.

Viimeinen tutkimuskysymys oli, että löytyykö verkkokaupoista riittävät ominaisuudet toimimaan automatisoidusti HUBin järjestelmien kanssa. Haasteita ilmeni noutopistelisäösin kanssa. Shopify- ja WooCommerce verkkokauppoihin löytyy sopivat lisäosat halutulla toiminnallisuudella. Vilkas Suite -verkkokauppaan löytyy sopiva sovellus, mutta noutopiste-id:n saatavuudessa rajapinnan kautta on vielä puute. Magento 2 -verkkokauppaan löytyy kuljetusliikekohtaiset lisäosat, joilla toiminnallisuuden saa toteutettua. Magentoon ei kuitenkaan löytynyt sopivaa lisäosaa, jolla saisi käyttöön kerralla useamman kuljetusliikkeen noutopisteet. MyCashflow-verkkokaupan noutopistelisäösin selvitykseen ei työn aikataulu riittänyt.

Selvitystyön rajallisella ajalla ei ehtinyt tutustua verkkokauppojen ominaisuuksiin tarpeeksi syvällisesti, jotta integraatioiden kaikki ongelmat olisivat tulleet ilmi. Selvitystyössä käsiteltiin verkkokauppoja melko yleisellä tasolla, joskin joihinkin verkkokauppojen ominaisuuksiin syvennyttiin enemmän. Keskittymällä yhteen verkkokauppa-alustaan, tutkittaviin asioihin olisi voinut perehtyä syvällisemmin ja kattavammin. Toisaalta silloin ei olisi saatu yleistä käsitystä verkkokauppojen integroitavuudesta, joka oli insinöörityön tarkoitus. Yhtenä haasteena työssä oli verkkokauppa-alustojen lukuisat eri versiot ja palvelutilaukset. Verkkokauppa-alustojen eri versioilla voi olla hieman eri ominaisuuksia, joten työssä tutkitut tiedot eivät välttämättä päde verkkokauppa-alustojen kaikkiin eri versioihin. Esimerkiksi Vilkaan verkkokaupoista tarkasteltiin vain Vilkas Suite -verkkokauppa.

Insinöörityön aikana heräsi ajatus, kannattaisiko muidenkin alan toimijoiden kehittää vaihtoehtoisia ratkaisuja integraatioiden toteuttamiseen verkkokauppoihin, kuten Shipit ja NettiVarasto ovat tehneet. Heillä on joihinkin verkkokauppoihin valmiit integraatiosovellukset, joiden avulla verkkokauppiat voivat helposti liittyä heidän palvelujen käyttö-

jiksi. Tällaisilla lisäsovelluksilla voisi saada näkyvyyttä, etenkin jos lisäsovellukset löytyvät verkkokauppojen verkkosivuilta ja lisäsovelluskaupoista. Tämä voisi olla hyvä lisätutkimuksen aihe alan toimijoille.

Insinöörityön aihealueet olivat minulle hyvin pitkälti uusia asioita ja niiden teoriaan opiskeluun meni paljon aikaa. Insinöörityön aikana opin paljon uusia asioita verkkokaupoista, web-tekniikoista, rajapinnoista ja integraatioista. Koen insinöörityöstä opittujen asioiden olevan minulle hyödyllisiä tulevaisuudessa.

Liitteeseen 1 olen koonnut listan hyödyllisistä linkeistä liittyen verkkokauppojen integraatioihin sekä insinöörityössä käsiteltyihin verkkokauppoihin ja käsitteisiin.

Lähteet

Aalto, Olli. 2019. REST-arkkitehtuurin hyödyntäminen web-rajapinnan kehittämisessä. Insinöörityö. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.

About Us. 2019. Verkkoaineisto. Shopify. <<https://www.shopify.ca/about>>. Luettu 31.10.2019.

An overview of HTTP. 2019. Verkkoaineisto. Mozilla Corporation. <<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>>. 29.9.2019. Luettu 7.11.2019.

Avoin lähdekoodi. 2019. Verkkoaineisto. Suomen avoimien tietojärjestelmien keskus - COSS ry <<https://coss.fi/avoimuus/avoin-lahdekoodi/>>. Luettu 12.11.2019.

Byron, Lee. 2015. GraphQL: A data query language. <<https://graphql.org/blog/graphql-a-query-language/>>. 14.9.2015. Luettu 12.11.2019.

Company Info. 2019. Verkkoaineisto. Shopify. <<https://news.shopify.com/company-info>>. Luettu 31.10.2019.

Compare Open Source and Magento Commerce. 2019. Verkkoaineisto. Magento. <<https://magento.com/compare-open-source-and-magento-commerce>>. Luettu 23.11.2019.

Consafe Logistics WMS. 2019. Verkkoaineisto. Consafe Logistics. <<https://www.consafe-logistics.com/products/wms/>>. Luettu 24.10.2019.

Customer Case Studies. 2019. Verkkoaineisto. Magento. <<https://magento.com/case-studies>>. Luettu 23.11.2019.

eCommerce for WordPress intro. 2019. Verkkoaineisto. WooCommerce. <<https://woocommerce.com/guides/ecommerce-for-wordpress-intro/>>. Luettu 29.10.2019.

Eising, Perry. 2017. What exactly IS an API? Verkkoaineisto. <<https://medium.com/@perrysetgo/what-exactly-is-an-api-69f36968a41f>>. 7.12.2017. Luettu 23.11.2019.

Fielding, Roy. 2000. Architectural Styles and the Design of Network-based Software Architectures. Filosofian tohtorin väitöskirja. Kalifornian yliopisto, Irvine. Verkkoaineisto. <https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf> Luettu 14.10.2019.

Fielding, Roy. 2013. Scrambled Eggs. Diaesitys. Verkkoaineisto. <https://www.sli-deshare.net/evolve_conference/201308-fielding-evolve>. Luettu 8.11.2019.

Getting Started with Magento Web APIs. 2019. Verkkoaineisto. Magento. <<https://dev-docs.magento.com/guides/v2.3/get-started/bk-get-started-api.html>>. 10/2019. Luettu 18.11.2019.

GraphQL Specification. 2018. Verkkoaineisto. GraphQL Foundation. <<https://graphql.github.io/graphql-spec/June2018/>>. 6/2018. Luettu 12.11.2019.

Hakkila Automated Warehouse Service. 2018. Pdf-esite. Hub logistics Group.

Hayes, David. 2019. The Four Languages You Must Know to Understand WordPress. Verkkoaineisto. <<https://wpshout.com/wordpress-programming-language/>>. Luettu 29.10.2019.

Hinnat. 2019. Verkkoaineisto. Vilkas Group Oy. <<https://www.vilkas.fi/hinnoittelu>>. Luettu 20.11.2019.

HTTP request methods. 2019. Verkkoaineisto. Mozilla Corporation. <<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>>. 23.3.2019. Luettu 25.10.2019.

HTTP. 2015. Verkkoaineisto. Tampereen yliopisto. <<http://www.cs.tut.fi/~seitti/2015/kalvot/http/all.html>>. Luettu 7.11.2019.

HUB logistics. 2019. Verkkoaineisto. Hub logistics -konserni. <<https://www.hub.fi/yri-tys/>>. Luettu 16.10.2019.

Hypertext Transfer Protocol (HTTP/1.1). 2014. Verkkoaineisto. Internet Engineering Task Force (IETF). <<https://tools.ietf.org/html/rfc7231>>. 6/2014. Luettu 7.11.2019.

Inside the iPaaS. 2019. E-kirja. Youredi.

Integraatioiden perusteet. 2019. Verkkoaineisto. Flashnode. <<https://www.flash-node.com/fi/oppaat/integraatioiden-perusteet>>. Luettu 17.10.2019.

Integraatio-opas. 2019. Verkkoaineisto. Alfame. <<https://www.alfame.com/integraatio-opas-opas-sujuvaan-tietojarjestelmaintegraatioon>>. Luettu 28.10.2019.

Jansen, Gert. 2011. The Job of the API Designer. Verkkodokumentti. <<https://restful-api-design.readthedocs.io/en/latest/scope.html>>. Luettu 24.11.2019.

Jedrzejewski, Bartosz. 2018. HATEOAS – a simple explanation. Verkkoaineisto. <<https://www.e4developer.com/2018/02/16/hateoas-simple-explanation/>>. 16.2.2018. Luettu 24.11.2019.

Koivistolainen, Jani. 2018. HUB logistics -konserni.

Laajala, Riku. 2019. GraphQL- ja REST-rajapintojen vertailu. Insinöörityö. Jyväskylän Ammattikorkeakoulu. Theseus-tietokanta.

Lipsman, Andrew. 2019. Global Ecommerce 2019. Verkkoaineisto. <<https://www.emarketer.com/content/global-ecommerce-2019>>. 27.6.2019. Luettu 24.10.2019.

Maciej. 2019. 9 Best Practices to implement in REST API development. Verkkoaineisto. <<https://www.merixstudio.com/blog/best-practices-rest-api-development/>>. 7.6.2019. Luettu 25.11.2019.

Mansukoski, Pentti. 2017. Elfhill Oy.

Metafields. 2019. Verkkoaineisto. Shopify. <<https://help.shopify.com/en/manual/products/metafields>>. Luettu 18.11.2019.

Miten perustat menestyvän verkkokaupan? Verkkoaineisto. Paytrail Oy. <<https://www.paytrail.com/miten-perustat-menestyvan-verkkokaupan>>. Luettu 14.11.2019.

Monus, Anna. 2018. SOAP vs REST vs JSON comparison. Verkkoaineisto. <<https://raygun.com/blog/soap-vs-rest-vs-json/>>. 2.8.2018. Luettu 8.11.2019.

MyCashflow API. 2019. Verkkoaineisto. Pulse 247 Oy. <<https://support.mycashflow.com/fi/kayttoopas/mycashflow-api>>. Luettu 18.11.2019.

Nordic Shipping App. 2019. Verkkoaineisto. Woolman Oy. <<https://woolman.fi/pages/nordic-shipping-app>>. Luettu 26.11.2019.

PATCH Method for HTTP. 2010. Verkkoaineisto. Internet Engineering Task Force (IETF). <<https://tools.ietf.org/html/rfc5789#section-2>>. 3/2010. Luettu 24.11.2019.

Perusta verkkokauppa. 2019. Verkkoaineisto. Pulse247 Oy. <<https://www.mycashflow.fi/perusta-verkkokauppa/>>. Luettu 29.10.2019.

PHP Developer Guide. 2019. Verkkoaineisto. Magento. <<https://devdocs.magento.com/guides/v2.3/extension-dev-guide/bk-extension-dev-guide.html>>. 17.10.2019. Luettu 23.11.2019.

Queries and Mutations. 2019. Verkkoaineisto. GraphQL Foundation. <<https://graphql.org/learn/queries/>>. Luettu 12.11.2019.

Richardson, Leonard. 2008. Act Three: The Maturity Heuristic. Verkkoaineisto. <<https://www.crummy.com/writing/speaking/2008-QCon/act3.html>>. 16.1.2009. Luettu 25.11.2019.

Rogers, Paul. 2019. Magento Open Source vs Magento Commerce (& on-premise vs cloud). Verkkoaineisto. <<https://paulnrogers.com/magento-open-source-vs-magento-commerce/>>. 24.3.2019. Luettu 23.11.2019.

Rouse, Margaret. 2019. SOAP (Simple Object Access Protocol). Verkkoaineisto. <<https://searchapparchitecture.techtarget.com/definition/SOAP-Simple-Object-Access-Protocol>>. Luettu 11.11.2019.

Ruby, Sam; Amundsen, Mike & Richardson, Leonard. 2013. RESTful Web APIs. E-kirja. O'Reilly Media, Inc.

Santos, Wendell. 2017. Which API Types and Architectural Styles are Most Used? Verkkoaineisto. <<https://www.programmableweb.com/news/which-api-types-and-architectural-styles-are-most-used/research/2017/11/26>>. 26.10.2017. Luettu 31.10.2019.

Shellberg, Thomas. 2019. Getting started with the REST API. Verkkoaineisto. <<https://github.com/woocommerce/woocommerce/wiki/Getting-started-with-the-REST-API>>. 19.8.2019. Luettu 18.11.2019.

Shopify Usage Statistics. 2019. Verkkoaineisto. Built With Pty Ltd. <<https://trends.builtwith.com/shop/Shopify>>. Luettu 23.11.2019.

ShopifyFD. 2019. Verkkoaineisto. Freakdesign. <<http://shopifyfd.com/>>. Luettu 26.11.2019.

SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). 2007. Verkkoaineisto. W3C. <<https://www.w3.org/TR/soap12/>>. 27.4.2007. Luettu 11.11.2019.

Sormunen, Marko. 2019. Johdatus web-palveluihin. Kuopion yliopisto, tietotekniikka keskus. Verkkoaineisto. <http://www.oppi.uef.fi/uku/serapi/menetelmat/johdatus_web-palveluihin.ppt>. Luettu 14.11.2019.

Suuri verkkokauppa-alustaraaportti. 2019. Verkkoaineisto. Paytrail Oyj. <https://www.paytrail.com/hubfs/images/eBook_Folder/Suuri-verkkokauppa-alustaraaportti-2019.pdf>. Luettu 14.10.2019.

Tarvainen, Jani. 2015. Mikä on HTTP/2 ja kuinka se otetaan käyttöön? Verkkoaineisto. <<https://www.symfony.fi/artikkeli/mika-on-http-2-ja-kuinka-se-otetaan-kayttoon>>. 24.10.2015. Luettu 24.11.2019.

Thoo, Eric ym. 2019. Magic Quadrant for Enterprise Integration Platform as a Service. Verkkoaineisto. <<https://www.gartner.com/doc/reprints?id=1-6KE0RUF&ct=190424&st=sb>>. 23.4.2019. Luettu 23.11.2019.

Tämä on WordPress. 2019. Verkkoaineisto. WordPress. <<https://fi.wordpress.org/>>. Luettu 18.10.2019.

Unifaun. 2019. Verkkoaineisto. Vilkas Group. <<https://www.vilkas.fi/tuki/unifaun>>. Luettu 21.11.2019.

Usein kysyttyä. 2019. Verkkoaineisto. Vilkas Group Oy. <<https://www.vilkas.fi/usein-kysyttya>>. Luettu 16.20.2019.

Warnimont, Joe. 2019. 11 Best Open Source and Free Ecommerce Platforms for 2019. Verkkoaineisto. <<https://ecommerce-platforms.com/articles/open-source-ecommerce-platforms>>. 10.11.2019. Luettu 14.11.2019.

Web Services Architecture: What is a Web service? 2004. Verkkoaineisto. W3C. <<https://www.w3.org/TR/ws-arch/#whatis>>. 11.2.2004. Luettu 14.11.2019.

Verkkokaupan suunnittelun helpous. 2019. Verkkoaineisto. Pulse247 Oy. <<https://www.mycashflow.fi/blog/882-verkkokaupan-suunnittelun-helppous/>>. 1.10.2011. Luettu 29.10.2019.

Verkkokauppa Suomessa. 2019. Verkkoaineisto. Paytrail Oy. <<https://www.paytrail.com/hubfs/Verkkokauppa-Suomessa-2019.pdf>>. Luettu 16.10.2019.

Vilkas Go. 2019. Verkkoaineisto. Vilkas Group Oy. <<https://www.vilkas.fi/vilkas-go>>. Luettu 20.11.2019.

Woo Rest API. 2019. Verkkoaineisto. WooCommerce. <<https://woocommerce.github.io/woocommerce-rest-api-docs/?shell#introduction>>. Luettu 18.11.2019.

WooCommerce Noutopistehaku / Ohjeet. 2019. Verkkoaineisto. Markup.fi. <<https://markup.fi/products/woocommerce-noutopistehaku/guide#heading-6>>. Luettu 21.11.2019.

WooCommerce Noutopistehaku. 2019. Verkkoaineisto. Markup.fi. <<https://markup.fi/products/woocommerce-noutopistehaku>>. 26.11.2010. Luettu 21.11.2019.

Yrityksen perustiedot. 2019. Yrityksen sisäinen dokumentti. HUB logistics -konserni.

Hyödyllisiä linkkejä verkkokauppa-alustoihin ja niiden integraatioihin liittyen

Verkkokauppa-alustojen ohjesivut:

WooCommerce <https://docs.woocommerce.com/>

MyCashflow <https://www.mycashflow.fi/ohjeet-ja-tuki/>

Magento 2 https://docs.magento.com/m2/ce/user_guide/getting-started.html

ePages/Vilkas Suite <https://www.vilkas.fi/tuki>

Shopify <https://help.shopify.com/en/>

Verkkokauppa-alustojen rajapintojen dokumentaatiot:

WooCommerce <https://docs.woocommerce.com/document/woocommerce-rest-api/>

MyCashflow <https://www.mycashflow.fi/docs/mycashflow-api/>

Magento 2 <https://devdocs.magento.com/guides/v2.3/get-started/bk-get-started-api.html>

ePages/Vilkas Suite <https://www.vilkas.fi/tuki/integraatiot-ulkopuolisiin-jarjestelmiin>

Shopify <https://help.shopify.com/en/api/getting-started>

Verkkokauppa-alustojen online-demoja:

Magento 2 <https://bsscommerce.com/blog/try-the-latest-magento-2-demo-free-with-sample-data/>

WooCommerce <https://premmerce.com/premmerce-woocommerce-demo/>

Noutopistelisiäosia:

WooCommerce <https://markup.fi/products/woocommerce-noutopistehaku>

MyCashflow <https://www.mycashflow.fi/verkkokaupan-ohjelmisto/palvelut/logistiikka/unifaun-ecom-lite/>

Magento 2 https://markup.fi/products?collection_id=3

ePages/Vilkas Suite <https://www.vilkas.fi/tuki/unifaun>

Shopify <https://woolman.fi/pages/nordic-shipping-app>

Rajapintatyylien määritelmät:

REST https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

SOAP <https://www.w3.org/TR/soap12/>

GraphQL <https://graphql.org/>

Webin tekniikkaa:

<https://developer.mozilla.org/en-US/docs/Web>