

Autotask PSA – Netbaron ERP -integraatio



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutus, Hämeenlinnan korkeakoulukeskus
Vuosi 2021

Pertti Suutarinen

Tekijä	Nimi	Vuosi 2021
Työn nimi	Autotask PSA – Netbaron ERP laskutusintegraatio	
Ohjaajat	Lasse Seppänen	

TIIVISTELMÄ

Opinnäytetyön tarkoituksena oli selvittää ja toteuttaa järjestelmien välille laskutusaineiston siirtoon tarvittava integraatiomoduli, jonka avulla manuaalinen laskutusaineiston siirto voidaan automatisoida. Automaation tarkoituksena on vapauttaa laskutukseen kuluva aikaa, sekä poistaa manuaalisesta työstä johtuvat satunnaiset virheet. Opinnäytetyön toimeksiantajana oli IT-Infotec Oy, jonka toisena yrittäjänä toimin.

Opinnäytetyön teoriaosuudessa käsitellään yleisellä tasolla järjestelmien välistä tiedonsiirtoa ja erilaisten integraatioiden toteutustapoja sekä integraation tarvitsemien tekniikoiden kuvaukset. Teoriaosuudessa käsitellään myös toteutuksessa käytettyjen SOAP- ja REST-rajapintojen toimintaa sekä tiedonsiirrossa käytetyn Finvoice-XML-sanoman rakennetta. Opinnäytetyön tyyppi on toiminnallinen.

Opinnäytetyön lopputuloksena syntyneen laskutusaineiston siirtomodulin johtopäätöksenä voidaan todeta projektin olleen onnistunut. Laskuttamiseen kuluva aika on lyhentynyt murto-osaan ja samalla laskujen sisältö on vakioitunut, helpottaen vastaanottajan työtä. Koska kohdejärjestelmän API-rajapinta ei tue kaikkia tarvittavia ominaisuuksia, joudutaan muodostettuja laskuja vielä manuaalisesti editoimaan. Tämän osalta toivomme toimittajan kehittävän rajapintaansa, jolloin prosessi nopeutuisi entisestään. Asiakasyritys on erittäin tyytyväinen kehitystyön tuloksiin.

Avainsanat integraatio, sovelluskehitys, API-rajapinnat

Author Pertti Suutarinen

Year 2021

Subject Autotask PSA – Netbaron ERP invoicing integration

Supervisors Lasse Seppänen

ABSTRACT

The purpose of this thesis was to develop an integration application between two systems, Autotask PSA and Netbaron ERP. Autotask PSA is the system IT-Infotec Oy uses for ticketing, labor documentation and to keep track of recurring billing and contracts. Netbaron ERP is used for all invoicing and accounting. The thesis was commissioned by IT-Infotec Oy where I am entrepreneur. The thesis is practical.

The research questions for the thesis are: How to develop the integration application and how does the integration shorten invoicing time?

The theoretical chapter discuss the main functions of the systems and concepts needed developing the integration application. In the practical part the stages of developing the application are explained and some of the code examples are reviewed as well.

As a result of the thesis, developed integration application is in use, and it has dramatically shortened the time needed for invoicing. Due to the limitations in target systems API, there is still some manual labor needed. Discussions with the manufacturer if they would improve their API are ongoing.

Keywords integration, software development, API-interface

Pages 26 pages and appendices 1 pages

Sanasto

Integraatio	Järjestelmien välisen tiedonsiirron toteutus
ERP	Yrityksen toiminnanohjausjärjestelmä
PSA	Asiantuntijayrityksen toiminnanohjausjärjestelmä
XML	Siirrettävän tiedon kuvaustapa
Python	Työssä käytetty ohjelmointikieli

Sisällys

1	Johdanto	1
2	Teoria.....	2
2.1	IT-Infotec Oy ja sen käyttämät toiminnanohjausjärjestelmät	2
2.2	Netbaron ERP	3
2.3	Autotask PSA	3
2.3.1	Tiketöinti ja työn kirjaus.....	4
2.3.2	Palvelutuotteiden sopimuslaskutus	5
2.3.3	Autotask PSA laskutusyhteenveto	6
2.4	Python-ohjelmointikieli.....	7
2.5	Ohjelmointirajapinnat.....	7
2.5.1	REST-API-rajapinta	8
2.5.2	SOAP-rajapinta	8
2.6	Finvoice	9
2.6.1	Finvoice-SOAP sanoman rakenne	10
2.6.2	Työssä käytetty SOAP-sanoman rakenne	11
2.6.3	Finvoice-määrittelyksiä	12
2.6.4	Finvoice-sanoman merkistö	12
2.6.5	Lukuarvojen ilmoittaminen Finvoice-sanomalla.....	13
2.6.6	Finvoice-laskusanomaesimerkki	13
3	Järjestelmien API-rajapinnat	15
3.1	Netbaron Web Services API	15
3.2	Autotask REST rajapinta.....	16
3.2.1	Autotask PSA-REST -API turvallisuus ja autentikointi	16
3.2.2	Projektissa käytetyt Autotask PSA Entiteetit	17
3.2.3	Invoices-entiteetti	17
3.2.4	Companies-entiteetti	18
3.2.5	BillingItems-entiteetti	19
4	Menetelmät ja kehitystyöt	20
5	Integraatiosovelluksen suunnittelu ja toteutus	21
5.1	Ohjelman perusrakenne	21
5.2	Datan lukeminen lähdejärjestelmän REST-API-rajapinnasta.....	22
5.3	Datan kirjoittaminen kohdejärjestelmän SOAP API -rajapintaan.....	23
5.4	Finvoice XML -tietueen muodostaminen.....	24

5.5	PDF-liitteen muodostaminen	24
5.6	Finvoice-SOAP-sanoman lähettäminen	25
6	Johtopäätökset ja pohdinta.....	26
6.1	Haasteita ja ratkaisuja.....	26
6.2	API-rajapinnan rajoitteet	26
7	Yhteenveto	28
	Lähteet.....	29
	Base64encode. (n.d.) Verkkopalvelu base64 enkoodaukseen ja dekkoodaukseen.	29
	Datto. (2021). <i>Investor Presentation</i>	29
	Datto. (toukokuu 2021). Luettelo Autotask PSA:n ominaisuuksista.	29
	Finanssiala. (n.d). Verkkolaskutus(Finvoice).	29
	Finanssiala. (23.5.2011) FInvoice 1.3 Soveltamisohje	29
	Netbaron. (n.d.-a) NetBaron ratkaisut. Haettu 28.11.2021 osoitteesta.....	29
	Netbaron. (n.d.-b) NetBaron WS-Laskutus-rajapintakuvaus NDA:n alainen dokumentti, ei julkisesti saatavana.....	30

Kuvat, ohjelmakoodit ja taulukot

Kuva 1	Esimerkki Autotask PSA -tiketistä	5
Kuva 2	Esimerkki Autotask PSA -sopimuslaskusta	6
Kuva 3	Esimerkki Autotask-laskutusraportista	7
Kuva 4,	SOAP-sanoman rakenne	9
Kuva 5	Finvoice XML sanoman rakenne yhdessä SOAP kehyksen kanssa (Finanssiala, 23.5.2011 s.15)	10
Kuva 6,	Autentikointi Autotask PSA-API -rajapintaan Postman-ohjelmistolla	16
Kuva 7,	Ohjelman vuokaavio	22
Kuva 8,	Finvoice laskusanoman muodostaminen	23
Taulukko 1,	HTTP-Methodien toiminnot	8
Taulukko 2	Finvoicen versiot julkaisuaikoinen	9
Taulukko 3,	Merkkien korvaaminen entiteeteillä	12
Taulukko 4,	Projektissa käytetyt entiteetit ja niiden kuvaus	17
Taulukko 5,	lähettävän sanoman osat	25

Liitteet

Liite 1 Aineistonhallintasuunnitelma

1 Johdanto

Yrityksissä käytetään eri toimintoihin erilaisia tietojärjestelmiä sen mukaan, mikä parhaiten tukee yrityksen liiketoimintaa. Tyypillisesti pienten ja keskisuurten yritysten käyttämät geneeriset ERP-, eli toiminnanohjausjärjestelmät on tehty auttamaan mahdollisimman laajan käyttäjäkunnan tarpeita, mutta vähänkään erityisemmät tarpeet jäävät toteuttamatta, tai ne on toteutettu tavalla, joka ei tuo kilpailuetua kaikille yrityksille.

Yllä mainitusta syystä tilaajalla on toiminnanohjaukseen IT-palvelutuotantoa varten räätälöity sovellus, Autotask PSA (Professional Services Automation), joka tukee palvelutuotantoa monin eri tavoin. Koska PSA-järjestelmä ei sovellu sellaisenaan kirjanpidon, laskutuksen tai logistiikan toimintoihin, käytetään Netbaron ERP -järjestelmää niiden toteuttamiseen.

Liiketoiminnan kasvaessa ja laskutettavien palveluiden lisääntyessä tilaajalla on havaittu manuaalisen tietojensiirron PSA-järjestelmästä laskutukseen olevan aikaa vievää sekä virhealtista. Tässä lopputyössä toteutetaan järjestelmien välille integraatiomoduli, jonka avulla laskutusaineiston siirto automatisoidaan.

Opinnäytetyön tutkimuskysymykset ovat:

- Kuinka integraatio eri järjestelmien välille rakennetaan?
- Minkälainen vaikutus integraatiolla on laskutuksen vaatimaan ajankäyttöön?

2 Teoria

Teoriaosuudessa esitellään tilaajayritys ja sen käyttämien järjestelmien yleisimmät toiminnot. Tämän jälkeen esitellään integraatiomodulin tarvitsemat toiminnallisuudet ja tekniikat, jolla työ saadaan toteutettua.

2.1 IT-Infotec Oy ja sen käyttämät toiminnanohjausjärjestelmät

IT-infotec Oy on Hyvinkää-Riihimäki talousalueella toimiva ICT-palveluita tarjoava yritys, jonka pääkohderyhmänä on alueen pk-yritykset kokoluokassa 10-200 henkeä. Yrityksen pääliiketoiminta tulee IT-ulkoistuspalveluista, joissa yritys huolehtii asiakasyritysten tietotekniikkatarpeista kuten laitetoimituksista, käyttöönotosta sekä muista elinkaaren aikaisista palveluista. Osana palvelutarjontaa on myös erilaiset pilvi- sekä konosalipalvelut. (Suutarinen, 2021)

IT-Infotec Oy:n liiketoiminta on ajan myötä muuttunut entistä enemmän palvelupohjaiseksi, jossa laite ja sovelluskauppa on muuttunut kertalaskutettavista jaksoittain laskutettaviin palveluihin, kuten kuukausittain veloittavat tietoturvapalvelut. Jotta erilaisten palveluiden hallinta ja työnohjaus on tehokasta, käytetään ERP-järjestelmän rinnalla erillistä tiketointi ja palveluidenhallintajärjestelmää, jossa töiden ja palveluiden laskutusta hallitaan. (Suutarinen 2021)

IT-Infotec Oy käyttää toiminnassaan kahta pääjärjestelmää:

Netbaron ERP	Laskutus, logistiikka, palkka- ja taloushallinto
Autotask PSA	Sopimuslaskutus, työnohjaus ja töiden raportointi

Kumpikin järjestelmä toimii pilvipalveluna toimittajiensa datakeskuksissa, eikä kummassakaan ole suoraa pääsyä esim. tietokantaan, vaan ulkoinen tiedonsiirto täytyy toteuttaa valmistajien API-rajapintojen kautta. Autotask PSA:han on mahdollista saada lisäpalveluna datawarehouse-lisäosa, jonka kautta tietokantayhteydet ovat mahdollisia.

2.2 Netbaron ERP

Netbaron ERP -järjestelmä on suomalainen, suomalaisille yrityksille ja tilitoimistoille kehitetty pilvipalvelu, jonka avulla kaikki yrityksen talouden ja toiminnanohjauksen päätoiminnot saadaan tehtyä. Netbaron ERP soveltuu esimerkiksi seuraaville toimialoille (Netbaron, n.d.-a):

- Tukku- ja vähittäismyynti
- Huolto ja kunnossapito
- Hotelli ja ravintola
- Kone- ja varaosamyynti sekä huolto
- LVIS
- Rakentaminen
- Tilitoimistot

Netbaron-järjestelmän laskutuonnin rajapinta vastaanottaa Finvoice 1.3 -tyyppisiä sanomia (Netbaron, n.d.-b), joka on suomalaisten pankkien määrittelemä XML-kuvauskieli laskujen sähköisessä muodossa (Finanssiala, n.d-a).

2.3 Autotask PSA

Autotask PSA on IT-asiantuntijaorganisaatioille suunnattu toiminnanohjausjärjestelmä, joka ottaa huomioon alan erityispiirteet. Koska järjestelmä on erityisesti suunniteltu IT-alan palveluyrityksille, siitä löytyvien toimintojen avulla saavutetaan etuja, joita generisellä ERP-järjestelmällä ei saavuteta. Autotask on nykyisin osa Datto Inc. yhtiötä, joka on yhdysvaltalainen pörssi-yhtiö. Yhtiö on perustettu vuonna 2007 ja sillä on 2000 työntekijää (Datto, 2021, s. 4).

Jotta käyttökokemus pysyy hyvänä, löytyy Autotask PSA:n konesaleja useista eri paikoista ympäri maailmaa. Etäisyyden kasvaessa tiedonsiirron viive, eli latenssi kasvaa, joten käyttäjä tyypillisesti käyttää sijaintiaan lähinnä olevaa konesalia. Toimeksiantajan Autotask-instanssi siirrettiin Iso-Britanniasta Saksaan vuoden 2020 aikana, jotta voimassa oleva GDPR-tietosuojasetus täyttyy.

Autotask PSA -järjestelmän ydinominaisuudet

- Tiketöinti
- Sopimustenhallinta
- Projektinhallinta
- Töiden aikataulutus ja raportointi
- Töiden ja sopimusten laskutusraportointi
- Integraatio Datto RMM etävalvonta ja -hallintajärjestelmään

Luettelo Autotask PSA:n ominaisuuksista löytyy Datton verkkosivuilta (Datto, n.d.)

Autotask PSA -järjestelmän tuottama laskutusraportti ei sellaisenaan sovellu asiakkaalle lähetettäväksi, vaan raportin osoittama laskutusaineisto täytyy viedä erikseen laskutusjärjestelmään, eli tässä tapauksessa Netbaron ERP-järjestelmään.

2.3.1 Tiketöinti ja työn kirjaus

Autotask PSA -järjestelmän toiminnanohjauksen ytimenä on työtilaukset eli tiketit. Raportoitava työ sidotaan aina liittymään tikettiin, eikä työsuoritetta voi kirjata ilman sitä.

Tiketti voidaan tarpeen mukaan linkittää laitekorttiin (Configuration Item, CI), jolloin laitteen työhistoria saadaan helposti selville. Lisäksi tiketti voidaan linkittää asiakkaan palvelusopimukseen, jota kautta asiakkaalle sovitut laskutussäännöt ja hinnoittelu astuvat voimaan. Työtyypistä ja asiakassopimuksesta riippuen työ joko laskutetaan erikseen tai sisältyy kiinteään sopimukseen. Kuvassa 1 näkyy esimerkki Autotask-tiketistä.

Kuva 1 Esimerkki Autotask PSA -tietistä

The screenshot displays the Autotask PSA interface for a service request ticket. The ticket is titled 'Testiticketti' and is associated with account 'Testiasiakas' (ID: T20201115.0002). The ticket was created on 15-11-2020 at 06:58 by Pertti Suutarinen. The description is 'Testataan API laskutusta'. The ticket is currently assigned to Pertti Suutarinen (Huoolto). The interface shows various sections including 'TICKET INFORMATION', 'ASSIGNMENT', 'DESCRIPTION', 'TIMELINE', 'RESOLUTION', 'ACTIVITY', 'CHARGES & EXPENSES', 'SERVICE CALLS & TO-DOS', 'ACCOUNT/CONTACT', 'TIME SUMMARY', and 'ASSET'. The 'TIME SUMMARY' section indicates a 5m remaining time for the ticket.

2.3.2 Palvelutuotteiden sopimuslaskutus

Työkirjausten laskuttamisen lisäksi Autotask PSA sisältää useita muita tapoja laskuttaa palveluita jaksoittain. Näitä ovat esimerkiksi ylläpito- ja tukipalvelut, pilvipalvelut, tietoturvapalvelut ja muut jaksoittain laskutettavat sopimukset.

Sopimuksia voi yhdellä asiakkaalla olla useita, joista jokaisella oma laskutusjaksonsa ja palvelunsa. Sopimuksen laskutusyksi voi olla kuukausi, neljännesvuosi, puolivuosi tai vuosi. Laskutus voi tapahtua joko ennakoon, tai laskutettavan jakson päättyessä. Tyypillisesti ennakoon laskutetaan laitevuokrapalvelut ja jakson päättyessä esimerkiksi tuntiperusteiset palvelut. Kuvassa 2 on esimerkki palveluiden sopimuslaskun sisällöstä.

Autotask PSA jyvittää kesken laskutuskautta lisättyjen tai poistettujen palveluiden hinnan käytettyjen päivien perusteella, jolloin toteutetaan pilvipalvelussa yleisesti käytettyä, kustannus käytön mukaan -määrittelyä.

Kuva 2 Esimerkki Autotask PSA -sopimuslaskusta

Contracts esim. Infotec kk laskutettavat - Google Chrome
 www19.autotask.net/Contracts/Views/ContractView.aspx?contractID=296834458&%20ContractType=7

SERVICE BUNDLES & SERVICES - esim. Infotec kk laskutettavat (IT-Infotec Oy) Contract Period 01-04-2021 - 30-04-2021

Summary: + New Service Bundle, + New Service, Apply Overall Discount, 0 Units, Show Data for 05-04-2021

Service or Bundle Name	Vendor Name	Period Type	Unit Price (List)	Discount	Unit Price (this Period)	Assets	Units	Extended Price
DG Palvelimen perusylläpöpalvelu		Monthly	€150.00	0.00%	€150.00	0	1	€150.00
DG Työaseman perusylläpöpalvelu		Monthly	€15.00	0.00%	€15.00	0	10	€150.00
Dynamics 365 for Sales, Enterprise		Monthly	€64.00	0.00%	€64.00	0	1	€64.00
Exchange Online		Monthly	€5.00	0.00%	€5.00	0	5	€25.00
IDID -Infonäyttöpalvelu		Monthly	€35.00	0.00%	€35.00	0	2	€70.00
M365 Business Standard		Monthly	€12.70	0.00%	€12.70	0	10	€127.00

What is this? Estimated Monthly Price: €586.00

Note: Extended Price may not equal Units x Unit Price. If there have been adjustments in the contract period. If the service or bundle period type differs from the contract period type, the Unit Cost, Unit Price (List), Unit Price, and Extended Price will reflect the period type with the longer duration. Discount may be slightly different than the discount value you entered. Due to rounding. For additional information, click the Help icon.

PRORATED TRANSACTIONS

Service or Bundle Name	Quote Name (Number)	Unit Change	Effective Date	Prorated Cost Amount	Prorated Price Amount
No Transactions Available					

2.3.3 Autotask PSA laskutusyhteenvedo

Asiakkaalle toimitetut palvelut voidaan laskuttaa joko yksittäisinä, esim. tiktin sulkemisen jälkeen, tai koontilaskuna, joka on tyypillinen tapa laskuttaa sopimusasiakkaita.

Asiakaslaskutusta varten Autotask PSA tekee yhteenvedoraportin laskutusvuorossa olevista palveluista. Tätä yhteenvedoraporttia käytetään varsinaisen Netbaronin muodostaman laskun liitteenä laskutusperusteen tarkistamista varten. Kuvassa 3 on esimerkki Autotask PSA:n koostamasta laskutusraportista.

Kuva 3 Esimerkki Autotask-laskutusraportista

Bill To		Date	
Testiasiakas Testikatu 1 Toinenkatu 2 05800 Testikaupunki		05.04.2021	
		Numero: 2385	
		Ajalta: 05.03.2021 - 05.04.2021	

Päivä	Kuvaus	Tyyppi	Suorittaja	Määrä	Veloitus	Laskutetaan
05.04.2021	T20201115.0002 Task/Ticket: Testiticketti Pertti Suutarinen Työ2, Palvelin, verkko ja järjestelmät Lähityö Työseloste: Internetyhteys ei toiminut. Käyty paikan päällä ja vaihdettu varalaitte rikkiöntyneen palomuurin tilalle. Testattu yhteys, kaikki toimii kuten pitää. Toimitetaan palomuuuri takuuvaihtoon ja käydään vaihtamassa kun palaa takaisin.	Labour	Suutarinen, Pertti	3.00	105.00/tunti	315.00
05.04.2021	Veloitus: Matkakulu sis. matka-aika T20201115.0002 Kulun peruste:	Ticket Charge		22.00	1.00	22.00
01.04.2021	Service: DG Palvelimen perusylläpöpalvelu [01.04.2021 - 30.06.2021] Contract Name: Testiasiakas Oy palvelusopimus	Service		1.00	450.00	450.00
01.04.2021	Service: DG Työaseman perusylläpöpalvelu [01.04.2021 - 30.06.2021] Contract Name: Testiasiakas Oy palvelusopimus	Service		10.00	45.00	450.00
01.04.2021	Service: Dynamics 365 for Sales, Enterprise [01.04.2021 - 30.06.2021] Contract Name: Testiasiakas Oy palvelusopimus	Service		1.00	192.00	192.00
01.04.2021	Service: Exchange Online [01.04.2021 - 30.06.2021] Contract Name: Testiasiakas Oy palvelusopimus	Service		5.00	15.00	75.00
01.04.2021	Service: iDiD -infonyttöpalvelu [01.04.2021 - 30.06.2021] Contract Name: Testiasiakas Oy palvelusopimus	Service		2.00	105.00	210.00
01.04.2021	Service: M365 Business Standard [01.04.2021 - 30.06.2021] Contract Name: Testiasiakas Oy palvelusopimus Sis. Office + Pilvipalvelut	Service		10.00	38.10	381.00
01.04.2021	Contract Name: Testiasiakas Oy palvelusopimus Setup Fee	Setup Fee		1.00	0.00	150.00
Total Billable Hours					3.00	
Total Billable Amount					€2,245.00	

2.4 Python-ohjelmointikieli

Python-ohjelmointikieli on avoimen lähdekoodin lisenssillä käytettävä, korkean tason yleiskäyttöinen ohjelmointikieli, jonka erityisvahvuutena on koodin luettavuus ja helppo oppittavuus. Pythonin opetteluun löytyy runsaasti sekä ilmaista, että maksullista verkkosisältöä ja kursseja.

Python valikoitui tämän projektin ohjelmointikieleksi sen kattavan rajapintojen käsittelyyn tarkoitettujen moduulien vuoksi, ja koska tavoitteena oli opetella kielen perusteet mahdollisia tulevia tarpeita varten.

2.5 Ohjelmointirajapinnat

API (Application Programming Interface) tarkoittaa ohjelman tai järjestelmän tarjoamaa liityntää, joka tarjoaa palveluita ulkoisille ohjelmistoille.

Tässä projektissa hyödynnetään lähdejärjestelmän REST-rajapintaa, sekä kohdejärjestelmän SOAP-rajapintaa.

2.5.1 REST-API-rajapinta

Representational state transfer, eli REST on ohjelmointirajapintojen toteuttamiseen tarkoitettu arkkitehtuurimalli, jossa tiedonsiirto tapahtuu HTTP-protokollan avulla (Tutorialspoint, n.d.-a).

REST-rajapinnan resursseilla on yleensä palvelun kautta saatavaa tietoa kuvaava juuriosoite, esimerkiksi */companies*, jonka kautta saadaan asiakasyrityksiin liittyvää dataa, kuten nimi ja osoitetiedot. Resurssi palauttaa määritellyn muotoista (esim. HTML, JSON, XML) tietoa, jotta sitä pystytään hyödyntämään asiakasjärjestelmässä.

REST-rajapintaa hyödynnettäessä HTTP-protokollan metodi määrittää ollaanko tietoa lukemassa, päivittämässä, poistamassa vai lisäämässä (Taulukko 1).

Taulukko 1, HTTP-Metodien toiminnot

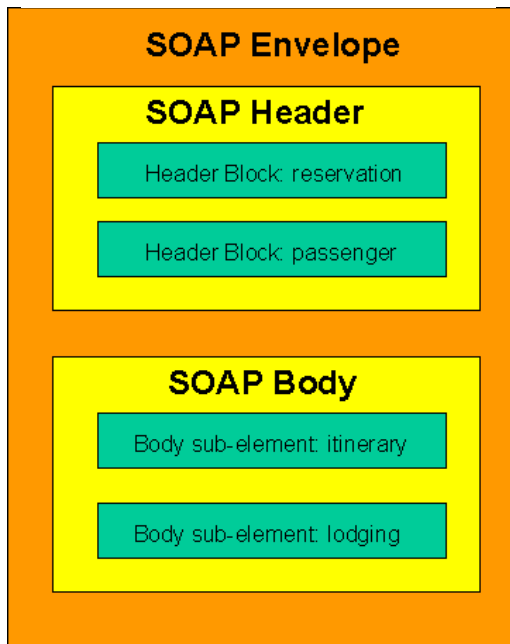
HTTP-metodi	Käyttötapa
GET	Tiedon lukeminen
PUT	Tiedon päivittäminen
DELETE	Tiedon poistaminen
POST	Tiedon lisääminen

2.5.2 SOAP-rajapinta

Simple Object Access Protocol eli SOAP on sanomien välitysprotokolla, jonka avulla voidaan siirtää tietoa järjestelmien välillä. Tiedon siirrossa käytetään XML-kuvauskieltä ja tyypillisesti HTTP-protokollaa (W3.Org, n.d).

Kuvassa 4 esitetään SOAP-sanoman rakenne, joka koostuu SOAP-kehyksestä (Envelope), Otsaketiedoista (header) sekä varsinaisen tiedon kuljettavasta Body-osiosta.

Kuva 4, SOAP-sanoman rakenne



Kuva: W3.org

2.6 Finvoice

Finvoice on suomalaisten pankkien määrittelemä XML-muotoinen sanoma, jonka avulla laskutus ja muu vastaava tieto, kuten tarjoukset, tilaukset, tilausvahvistukset ja hinnastot voidaan välittää organisaatioiden välillä. Finvoice-muotoisia sanomia välittävät pankit ja Finvoice-välittäjäoperaattorit. (Finanssiala, n.d)

Finvoice-määritelmän uusin versio on 3.0, mutta Netbaron tukee vain versiota 1.3. Finvoicen eri versiot ja niiden päämuutokset on esitetty taulukossa 2.

Taulukko 2 Finvoicen versiot julkaisuaikoinen

vuosi	versio
2003	1.0
2004	1.1
2005	1.2
2008	1.3
2012	2.0
2017	3.0

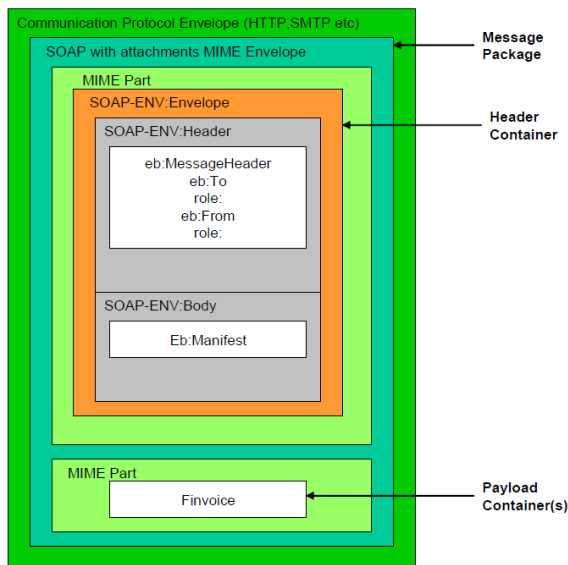
Koska tässä yhteydessä Finvoicea käytetään sisäisesti, sovelletaan sitä vain tarvittavin osin, eikä määrittelyssä huomioida Finvoice-sanoman tiukkaa validointia.

2.6.1 Invoice-SOAP sanoman rakenne

Finvoice-sanoman siirrossa käytetty SOAP-sanoma on MIME Multipart -tyyppinen sanoma, jossa SOAPille, headerille ja Finvoicelle on oma osansa (Finanssiala 23.5.2011, s.15).

Finvoice XML -viesti lähetetään liitettynä SOAP-kirjekuoreen omana osanaan. SOAP-sanoman Body-osassa kerrotaan viestiin liittyvän Finvoice-laskusanoman, joka tuodaan omassa MIME-osassaan. Viestin header-osiossa määritellään lähetettävän laskun reititystiedot. Kuvassa 5 on esitetty Finvoice-laskusanoman sisältävän SOAP-viestin rakenne.

Kuva 5 Finvoice XML -sanoman rakenne yhdessä SOAP-kehyksen kanssa (Finanssiala, 23.5.2011 s.15)



SOAP-ENV:Header-tagien sisällä välitetään sanoman lähettäjä(From), vastaanottaja(To) sekä mahdollinen välittäjä(tieto)(Intermediator). Envelope-kehyksen avulla vastaanottava taho osaa käsitellä ja reitittää viestin oikealla tavalla.

SOAP-ENV:Body-tagien sisällä olevassa eb:Manifest-osiossa ilmoitetaan tiedot sanomassa välitettävästä laskusta, joka sijoitetaan omaan MIME-osioonsa.

Finvoice-sanoma sisältää seuraavat tiedot (Finanssiala, 23.5.2011)

- Myyjän tiedot <SellerPartyDetails>
- Laskun lähettäjän tiedot <InvoiceSenderPartyDetails>
- Laskun vastaanottajan tiedot <InvoiceRecipientDetails>
- Ostajan tiedot <BuyerPartyDetails>
- Toimitusosapuolen tiedot <DeliveryPartyDetails>
- Laskun tiedot <InvoiceDetails>
- Laskurivin tiedot <InvoiceRow>
- Myyjän antamat tiedot maksutoimeksiantoa varten <EpiDetails >

Lähetettäessä Finvoice-sanoman liitetään siirtokehys, jota käytetään laskun reitittämiseen vastaanottajalle.

2.6.2 Työssä käytetty SOAP-sanoman rakenne

Netbaron WS-Laskurajapinnan toteutus poikkeaa hiukan Finanssialan Finvoice-määrittämisestä, joten työssä käytetty SOAP-rakenne ei ole määrittämisen mukainen. Tällä ei ole käytännössä merkitystä, koska rajapinnan kautta muodostetaan laskuja sisäiseen järjestelmään. Myöskään laskujen reitityksestä ei tarvitse välittää, koska vastaanottava järjestelmä on itse suoraan kohdejärjestelmä. Itse laskun lähettäminen pankin tai operaattorin kautta soveltaa Finvoice-määrittäystä.

Ohjelmakoodissa 1 esitellään työssä käytetty SOAP-sanomarakente. Ero Finanssialan Finvoice-määrittämisestä on se, että Finvoice-sanoma on suoraan Body-elementissä, eikä sen ulkopuolella omana osanaan, kuten määrittelyssä ohjeistetaan.

Ohjelmakoodi1, Työssä käytetty SOAP-rakenne

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns1="urn:NBWS"
xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
<env:Header>
    <wsse:Security env:mustUnderstand="true">
        <wsse:UsernameToken>
            <wsse:Username>WStesti</wsse:Username>
            <wsse:Password>xxxxxxxxxxxxxxxxxxxx</wsse:Password>
```

```

                </wsse:UsernameToken>
            </wsse:Security>
</env:Header>
<env:Body>
    <ns1:put>
    <Request>
    <InvoiceBill>>false</InvoiceBill>
    <Finvoices>
    <Finvoice1>
    <FileName>e-lasku</FileName>
    <File>Finvoice-sanoma base64 enkoodattuna</File>
        </Finvoice1>
    </Finvoices>
    </Request>
    </ns1:put>
</env:Body>
</env:Envelope>

```

2.6.3 Finvoice-määrittäjiä

Finvoice-laskusanoman elementeillä on Finvoice-tietoluettelomäärittelyn mukaiset minimi- ja maksimipituudet. Kentät, joissa ei ole tietoa, tulee jättää kokonaan pois.

Yhdellä rivillä voi olla vain yksi elementti alku- ja lopputageineen. Alku- ja lopputagin välissä olevan arvon edessä tai lopussa ei saa käyttää välilyöntejä. Tiedon sisennysmerkinä ei voi käyttää tabulaattoria, ainoastaan välilyönti on sallittu.

Finvoice 1.3 -välityspalvelussa ei välitetä liitteitä. Netbaron itsessään sallii kuitenkin liitteiden käsittelyn rajapintakuvauksessa kuvatulla tavalla, koska liitteet siirretään sisäisesti ilman välittäjää.

2.6.4 Finvoice-sanoman merkistö

Finvoice-sanomilla käytetään ISO-8859-15-merkistöä. Finvoice-verkkolaskussa XML-standardin mukaisesti korvataan seuraavat merkit entiteeteillä:

Taulukko 3, Merkkien korvaaminen entiteeteillä

Merkki	Entiteetti
&	&
<	<

>	>
”	"
'	'

2.6.5 Lukuarvojen ilmoittaminen Finvoice-sanomalla

Finvoice-määrittelyssä kuvataan lukuarvojen ilmoittaminen seuraavasti:

- Desimaalit erotetaan aina pilkulla kokonaisluvuista
- Numerotiedoissa ei ole etunollia
- Prosentit esitetään 1-3 desimaalilla
- Valuuttakurssit esitetään 6 desimaalilla
- Summan kokonaislukuosaan on sisällytettävä vähintään yksi numero
- Luvussa oleva pilkku lasketaan mukaan kentän maksimipituuteen

2.6.6 Invoice-laskusanomaesimerkki

Tässä esimerkki testauksessa käytetystä Finvoice-laskusanomasta. Sanomalla testattiin laskun muodostaminen kohdejärjestelmään Postman-sovelluksen avulla. (<https://www.postman.com>)

```

Koodiesimerkki 1, Finvoice-laskusanoma
<Finvoice Version="1.3">
  <BuyerPartyDetails>
    <BuyerPartyIdentifier>1234567-8</BuyerPartyIdentifier>
    <BuyerOrganisationName>Firma Oy Ab</BuyerOrganisationName>
    <BuyerPostalAddressDetails>
      <BuyerStreetName>katuosoite 9</BuyerStreetName>
      <BuyerPostCodeIdentifier>05800</BuyerPostCodeIdentifier>
      <BuyerTownName>Hyvinkää</BuyerTownName>
      <BuyerContactPersonName>Teuvo Testaaja</BuyerContactPersonName>
      <BuyerEmailAddressIdentifier>teuvo.testaaja@gmail.com</BuyerEmail
addressIdentifier>
      <InvoiceFreeText>Joulukuun palvelulaskutus</InvoiceFreeText>
      <OrderIdentifier>Teuvo Testaaja</OrderIdentifier>
    </BuyerPostalAddressDetails>
  </BuyerPartyDetails>
  <InvoiceDetails>
    <InvoiceTypeCode>INV01</InvoiceTypeCode>
    <AmountCurrencyIdentifier>EUR</AmountCurrencyIdentifier>
    <SalesPersonName>Pertti Suutarinen</SalesPersonName>
    <InvoiceDate Format="CCYYMMDD">20210228</InvoiceDate>
  </InvoiceDetails>
</InvoiceAttachments>
  <InvoiceAttachment>

```

```
<AttachmentName>liite.pdf</AttachmentName>
<AttachmentMimeType>application/pdf</AttachmentMimeType>
<AttachmentContent>base64_liite_tähän</AttachmentContent>
</InvoiceAttachment>
</InvoiceAttachments>
<InvoiceRow>
  <ArticleIdentifier>9876</ArticleIdentifier>
  <ArticleName>testituote</ArticleName>
  <UnitPurchaseAPriceAmount>300</UnitPurchaseAPriceAmount>
  <DeliveredQuantity>5</DeliveredQuantity>
  <UnitPriceAmount>123</UnitPriceAmount>
</InvoiceRow>
<InvoiceRow>
  <ArticleIdentifier>111222</ArticleIdentifier>
  <ArticleName>Ääkköstuote2</ArticleName>
  <UnitPurchaseAPriceAmount>300</UnitPurchaseAPriceAmount>
  <DeliveredQuantity>12</DeliveredQuantity>
  <UnitPriceAmount>567</UnitPriceAmount>
</InvoiceRow>
</Finvoice>
```

3 Järjestelmien API-rajapinnat

Työssä toteutettava itegraatiomoduuli käyttää lähde- ja kohdejärjestelmien API-rajapintoja tiedonsiirron toteutukseen. Autotaskin tapauksessa rajapintana on REST-rajapinta ja Netbaronin tapauksessa SOAP-rajapinta. Autotaskiin on toteutettu myös SOAP-rajapinta, mutta se on ilmoitettu poistuvan käytöstä vuoden 2021 alkupuolella, joten valinta luonnollisesti kohdistui REST-rajapintaan.

3.1 Netbaron Web Services API

Netbaron Web Services API tarjoaa tavan integroida ulkoisia järjestelmiä SOAP-protokollaa käyttäen. Rajapinta on toteutettu sovelluskohtaisesti ja jokaiselle on tehty oma rajapintakuvauksensa. Sopimus ja hinnoittelu rajapinnan käytöstä on moduulikohtainen ja jokaisella moduulilla on myös omat kirjautumistunnukset sekä sallittujen IP-osoitteiden listansa. Nämä vaatimukset tulee huomioida rajapintoja hyödynnettäessä, jotta yllätyksiä esim. kustannusten tai käyttöpaikan suhteen ei syntyisi.

Koska Netbaron API -rajapinnan hyödyntäminen edellyttää toimittajan NDA-sopimuksen tekemistä, en avaa rajapinnan ominaisuuksia tässä raportissa tarkemmin.

Netbaron WS -rajapinnat löytyvät seuraaviin ohjelmistomoduuleihin:

- WS-asiakasrekisteri
- WS-tuoterekisteri
- WS-myyntilaskutus
- WS-ostolaskutus
- WS-myyntitilaus
- WS-ostotilaus
- WS-kalenteri
- WS-varauskalenteri
- WS-palaute
- WS-huolto
- WS-kirjanpito

- WS-palkka
- WS-Kello

Tässä työssä käytettiin Netbaron WS-laskutus -rajapintaa.

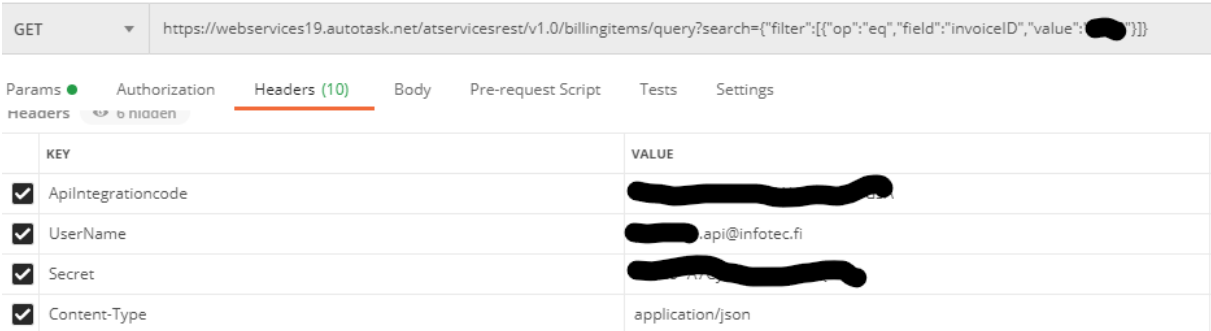
3.2 Autotask REST rajapinta

Autotask REST -rajapinta on melko uusi, sen ensimmäinen versio on julkaistu 18.5.2020. Projektissa käytetty API-versio on 1.0 ja se on julkaistu 9.12.2020. Autotask PSA:n API-rajapintakuvaus on avoin ja sen voi ladata Autotaskin verkkosivuilta (Autotask, n.d.)

3.2.1 Autotask PSA-REST -API turvallisuus ja autentikointi

Rajapinta edellyttää TLS 1.2 -protokollaa tiedon suojaukseen ja API-rajapintaan pääsee käsiksi vain käyttäjätunnuksella, joka kuuluu Autotaskissa määritettyyn API-user -käyttäjryhmään. Autentikaatio toteutetaan käyttäjätunnuksen ja siihen liittyvän salasanan, sekä ApiIntegrationcode-avaimen avulla. Käyttäjäautentikointitieto välitetään http-sanoman Header-tietueessa (Kuva 6). Rajapinnan käyttö on avointa kaikkialta internetistä, joten esim. mobiiliapplikaatiot on mahdollista toteuttaa sen avulla.

Kuva 6, Autentikointi Autotask PSA-API -rajapintaan Postman-ohjelmistolla



KEY	VALUE
<input checked="" type="checkbox"/> ApiIntegrationcode	[REDACTED]
<input checked="" type="checkbox"/> UserName	[REDACTED].api@infotec.fi
<input checked="" type="checkbox"/> Secret	[REDACTED]
<input checked="" type="checkbox"/> Content-Type	application/json

Autotask PSA:n käyttäjähallinnassa voidaan tarkemmin määritellä API-käyttäjätunnuksen oikeudet eri sovellusmoduuleihin. Oletuksena pääsy on kaikkialle muualle paitsi käyttöliittymään.

3.2.2 Projektissa käytetyt Autotask PSA Entiteetit

Taulukossa 4 luetellaan tässä projektissa käytetyt Autotask PSA -entiteetit ja niiden kuvaus. Entiteettejä järjestelmästä löytyy kymmenittäin, tarkempi lista löytyy API-dokumentaatiosta (Autotask, n.d.-b).

Taulukko 4, Projektissa käytetyt entiteetit ja niiden kuvaus

Entiteetti	Kuvaus
BillingItems	This entity describes an approved and posted billable item in Autotask. A billing item may or may not be included in an invoice and billed to the customer.
Companies	This entity describes an Autotask Company. It represents any organization with which you do business.
Invoices	This entity describes an Autotask Invoice. Invoices include Billing Items that have been approved and posted and are being billed to a customer or presented for information purposes only.

3.2.3 Invoices-entiteetti

Autotaskin Invoices-entiteetti sisältää laskun muodostamisessa käytettävät tiedot. Invoices-entiteetti sisältää laskutapahtumasta vain perustiedot, joiden avulla voidaan kysyä yksityiskohdat niistä huolehtivilta muilta rajapinnoilta. Invoices-entiteetin REST-rajapintaa kutsutaan omasta rajapinnastaan (Autotask, n.d.-c).

Invoices-rajapinta palauttaa invoiceID:n perusteella vain perustietoja, kuten:

- Eränumeron (batchID)
- Asiakas ID:n (companyID)
- Laskunumeron (invoiceID)
- Laskun loppusumman (invoiceTotal)

Invoices-rajapinta ei siis sisällä laskun yksityiskohtia, kuten asiakkaan nimeä tai osoitetta, eikä myöskään laskurivitietoja. Jotta integraatioprojektissa tarvittava yksilöity laskuerittely asiakastietoineen voitiin muodostaa, pitää yksityiskohdat noutaa omista rajapinnoistaan.

3.2.4 Companies-entiteetti

Companies entiteetti sisältää asiakasyritysten tarkemmat tiedot ja sitä kutsutaan omasta URL osoitteestaan (Autotask, n.d.-d).

Companies-rajapinta palauttaa companyID:n perusteella asiakkaan asiakaskortille tallennetut yksityiskohtaiset tiedot, kuten:

- Nimi
- Osoite
- Y-tunnus
- Puhelinnumero jne.

Ohjelmakoodissa 2 on esimerkki rajapinnasta saadusta JSON koodista, kun companyID on 223.

Ohjelmakoodi 2, Esimerkki rajapinnan palauttamasta JSON koodista.

```
"item": {
  "id": 223,
  "address1": "Yritystie 7",
  "address2": null,
  "alternatePhone1": "",
  "city": "HYVINKÄÄ",
  "classification": null,
  "companyCategoryID": 1,
  "companyName": "Yritys Oy",
  "isActive": true,
  "ownerResourceID": 4,
  "phone": "0194 600 300",
  "postalCode": "05460",
  "taxID": "1234567-8",
  "webAddress": "www.yritys.fi",
}
```

Koodista on jätetty luettavuuden parantamiseksi pois harvemmin käytettäviä kenttiä.

3.2.5 BillingItems-entiteetti

BillingItems-entiteetti sisältää laskun yksityiskohtaiset tiedot, eli käytännössä laskurivit.

Rajapinnalle löytyy oma URL-osoitteensa (Autotask, n.d.-e).

Ohjelmakoodissa 3 on esimerkki rajapinnan palauttamasta JSON-koodista.

Ohjelmakoodi 3, Esimerkki billingItems-rajapinnan palauttamasta JSON-koodista.

```
"items": [{
  "id": 6394,
  "billingItemType": 6,
  "companyID": 247,
  "contractID": 29683381,
  "contractServiceID": 118,
  "contractServicePeriodID": 2756,
  "description": "domainin nimipalvelu (2 x 2.0000 = 4.0000)",
  "extendedPrice": 4.0000,
  "invoiceID": 1495,
  "itemDate": "2020-12-01T00:00:00Z",
  "itemName": "Nimipalvelu DNS",
  "lineItemFullDescription": "Service: Nimipalvelu DNS
[01.12.2020 - 31.12.2020] ",
  "nonBillable": 0,
  "ourCost": 2.0000,
  "postedDate": "2020-12-30T00:00:00Z",
  "postedOnTime": "2020-12-30T12:06:51.593Z",
  "quantity": 2.0000,
  "rate": 2.0000,
  "serviceID": 16,
  "sortOrderID": 1,
  "subType": 12,
  "taxDollars": 0.9600,
  "totalAmount": 4.0000,
}]
```

Koodista on jätetty luettavuuden parantamiseksi pois harvemmin käytettyjä kenttiä.

4 Menetelmät ja kehitystyöt

Opinnäytetyön tavoitteena on kehittää integraatiosovellus, jota hyödynnetään kuukausittain sopimuslaskutuksessa. Sovelluksen kehityksessä sovellettiin ketterää Scrum-menetelmää, jonka avulla sovelluksen ominaisuuksia lisättiin kehityksen aikana. Lopputuloksena on työn teoriaosuuteen pohjautuva sovellus, joka sovittaa lähde- ja kohdejärjestelmien erilaiset rajapinnat ja tietokentät yhteen muodostaen lähdejärjestelmän laskutusaineistoa vastaavan laskun kohdejärjestelmään pdf liitetiedostoineen.

Ohjelmiston kehityksessä käytettiin Visual Studio Codea (VS Code), joka on avoimeen lähdekoodiin pohjautuva koodieditori. Ohjelmointikieleksi valittiin Python 3.

Rajapintojen testaamista varten käytettiin Postman-sovellusta ja testauksenaikaiseen base64-enkoodaukseen ja dekodaukseen verkkopalvelua (`base64encode`, n.d)

5 Integraatiosovelluksen suunnittelu ja toteutus

Integraatiosovelluksen kehittäminen aloitettiin tutustumalla lähde- ja kohdejärjestelmien rajapintoihin, tutoriaaleihin sekä esimerkkikoodeihin. Koska projektin sivutavoitteena oli samalla opetella Python ohjelmointikieli, tuli siitä samalla iso osa projektin ajankäyttöä.

Itse sovellus rakentui projektissa osa-alue kerrallaan. Kun yhden osa-alueen toiminnallisuus pääpiirteittäin saatiin toimivaksi, siirryttiin seuraavaan osa-alueeseen. Esimerkiksi, kun lähdejärjestelmän REST-rajapinnasta saatiin JSON-muotoinen vastaus konsoliin kaiutettuna, lähdettiin tutkimaan, kuinka sen sisältämiä avain-arvopareja pystytään lukemaan muuttujiin ja sitä kautta käyttämään ohjelman toimintalogiikassa.

Lisäksi kehitysprosessia vietiin eteenpäin kohdejärjestelmän suunnasta katsoen, tutkimalla seuraavaa kysymystä:

- Mitä tietoa pitää lähdejärjestelmästä saada, jotta kohdejärjestelmään voidaan muodostaa lasku asiakkaalle lähetettäväksi?

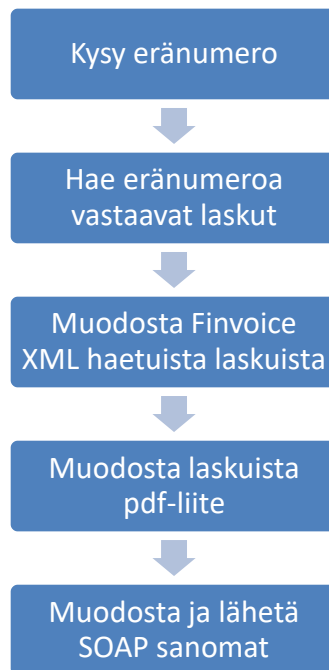
Kun listaus lähdejärjestelmästä tarvittavan tiedon osalta oli valmis, tutkittiin seuraavaksi keinot tiedon noutamiseksi ja lukemiseksi ohjelman hyödynnettäväksi.

5.1 Ohjelman perusrakenne

Ohjelman rakenne on suoraviivainen. Kun laskutettavaksi halutut palvelut on Autotask PSA:n omalla laskutustoiminnolla ennakkoon prosessoitu, käytetään siinä saatua eränumeroa sovelluksen lähtötietona. Autotask PSA:n REST-rajapinnan invoices-resurssista haetaan lista ko. eränumeroon kuuluvista laskuista ja laskujen yksityiskohdat, eli käytännössä rivitiedot omasta billingItems-resurssistaan.

Saadusta laskudatasta muodostetaan Finvoice-XML-sanoma sekä pdf-liite, jotka lähetetään osana SOAP-sanomaa kohdejärjestelmän rajapintaan. Kuvassa 7 on esitetty ohjelman vuokaavio.

Kuva 7, Ohjelman vuokaavio



5.2 Datan lukeminen lähdejärjestelmän REST-API-rajapinnasta

Kun laskun muodostamisessa tarvittava data oli selvillä, tutkittiin millä tavoin se on luettavissa lähdejärjestelmän REST-API-rajapintaa käyttäen. Tässä vaiheessa ei kiinnitetty huomiota kohdejärjestelmän SOAP-API-rajapintaan, tai siihen miten se toimii. Fokus oli selvittää ainoastaan se, kuinka Python-ohjelmalla voidaan tehdä toiminto REST API -rajapinnan kutsumiseen ja sen palauttaman datan käsittelyyn konsolissa luettavaksi.

Testauksessa korvaamattomana apuna oli Postman-sovellus, jolla rajapintaa voitiin kutsua erilaisin parametrein ja saada visuaalinen palaute kutsun onnistumisesta ja halutun datan saamisesta. Tämä vaihe vaati satoja erilaisia kokeiluja, ennen kuin halutun muotoinen kutsu saatiin aikaiseksi.

Koska REST API -rajapinta palauttaa tiedot JSON-muodossa, piti tutkia keinot sen sisältämien avain/arvoparien löytämiseen ja tarvittavien arvojen tallentamiseen muuttujiin jatkokäyttöä varten. Itse JSON-data tallennettiin yksinkertaisuuden vuoksi kokonaisuutena yhteen muuttujaan, josta haluttu tieto haettiin funktioiden avulla. Funktiot tehtiin asiakas-, lasku- ja rivitietojen hakuun.

5.3 Datan kirjoittaminen kohdejärjestelmän SOAP API -rajapintaan

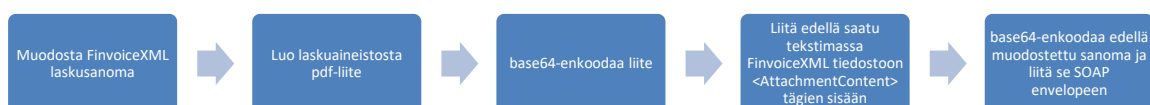
Kun lähdejärjestelmästä oli saatu luettua tarvittavat tiedot, alettiin tutkimaan kohdejärjestelmän SOAP API -rajapinnan toimintaa ja tapoja sen hyödyntämiseen. Netbaron järjestelmän käyttämä SOAP API -rajapinta muodostaa laskun Finvoice 1.3 määrittymisen mukaisesta sanomaliitteestä, joka täytyy olla base64-enkoodattuna. Tämä tarkoittaa sitä, ettei Postman-sovelluksella lähetetyt rajapintakutsut olleet ihmisen ymmärtämässä muodossa, vaan enemmänkin sekalaista merkkimassaa. Tämän vuoksi SOAP-rajapinnan testaaminen oli Autotask PSA:n käyttämään REST-rajapintaan verrattuna hankalampaa.

Netbaron API -rajapintakuvauksessa olleet esimerkit olivat PHP-kielellä toteutettuja, joten niitä ei suoraan pystynyt hyödyntämään tässä projektissa. Lisäksi saatavilla ei ollut suoraan Postmanilla tehtyjä esimerkkejä, joita olisin voinut hyödyntää rajapinnan testaamisessa. Netbaronin SOAP-API-rajapinnan testaamisessa suuri osa ajasta kuluikin kokonaisuuden toimintojen testaamisessa sekä SOAP-sanoman määrittelyssä. Kokonaisuuteen kuului esimerkiksi autentikointiheaderien määrittely ja kuinka Finvoice XML -tietue saadaan oikealla tavalla koodattua ja liitettyä SOAP-sanomaan.

Kun viimein Postman sovelluksen avulla saatiin Netbaroniin testilasku muodostumaan, oli aika selvittää, kuinka laskuun voidaan lisätä liite. Koska rajapintakuvauksessa prosessia ei oltu kovin tarkasti kuvattu, kului prosessin selvittämiseen useita viikkoja erilaisine kokeiluineen. Loppujen lopuksi valmistajan tuen avulla liitteet saatiin toimimaan ja päästiin eteenpäin työssä.

Kuvassa 8 esitellään vuokaaviona Finvoice-laskusanoman muodostaminen.

Kuva 8, Finvoice-laskusanoman muodostaminen



Kohdejärjestelmän vaatiman Finvoice-sanoman muodostamiseen tarvittava prosessi on monimutkainen ja esimerkkitoteutusten puuttumisen vuoksi erilaisia testejä jouduttiin tekemään sadoittain oikean rakenteen löytämiseksi.

5.4 Finvoice XML -tietueen muodostaminen

Jotta kohdejärjestelmän vaatima Finvoice-tyyppinen tietue saatiin muodostettua, oli tutkittava, kuinka Pythonilla rakennetaan XML-markup-määrittymisen mukaisia tietueita. Metodina Finvoice-sanoman muodostamiseen oli tutkia esimerkkinä saadun XML-laskun koodia ja pyrkiä ohjelmallisesti tuottamaan samanlainen lähdejärjestelmästä saatujen tietojen perusteella. XML-muotoisen tietueen käsittelyyn löytyi Python-yhteisön verkkokeskusteluja selailemalla ElementTree-moduuli, joka on tehty XML-tietueiden lukemiseen ja kirjoittamiseen.

XML-tietue alustetaan Element() metodilla ja sen alle lisätään alielementtejä SubElement() metodin avulla. Ohjelmakoodissa 5 on esimerkki ElementTree-moduulin käytöstä ja lopputulos ohjelmakoodissa 6.

Ohjelmakoodi 5. Esimerkki ElementTree-moduulin käytöstä

```
root = Element('Finvoice', version='1.3')
BuyerPartyDetails = SubElement(root, ' BuyerPartyDetails ')
BuyerPartyIdentifier = SubElement(root, ' BuyerPartyIdentifier ')
BuyerPartyIdentifier.text = '1234567-8'
```

Ohjelmakoodi 6. Edellä olevan koodin avulla muodostettu XML-tietue

```
<Finvoice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
Version="1.3">
  <BuyerPartyDetails>
    <BuyerPartyIdentifier>1234567-8</BuyerPartyIdentifier>
  </BuyerPartyDetails>
</Finvoice>
```

5.5 PDF-liitteen muodostaminen

Lähdejärjestelmästä luetun datan perusteella muodostetaan pdf-muotoinen tiedosto, joka viedään laskun liitteeksi. Seuraavaksi oli siis selvitettävä, kuinka PDF-muotoinen tiedosto saadaan tehtyä.

Python kehittäjä sivustoja tutkittuani löysin FPDF-moduulin, johon löytyi myös muutamia tutoriaaleja, joten valitsin sen pdf-tiedostojen muodostamiseen. (PyFPDF, n.d.-a)

Moduuli on viimeksi päivitetty vuonna 2012, ja sen käytettävyys oli suhteellisen vaatimaton, mutta sillä saa perusmuotoisen dokumentin tehtyä.

FPDF-modulia käytetään pääosin kertomalla koordinaateilla minkä verran edellisestä sijainnista siirrytään, ja lisäämällä teksti uuteen sijaintiin. Kun dokumentti on valmis, tallennetaan se levyllä `pdf.output('tiedostonimi', 'F')` komennon avulla.

Muodostettu dokumentti enkoodataan (base64) ja liitetään Finvoice-sanoman osaksi `<file></file>` tágien sisään.

5.6 Finvoice-SOAP-sanoman lähettäminen

Kun yllä olevan prosessin lopputuloksena saadaan valmis Finvoice-sanoma, se enkoodataan (base64) ja liitetään Finvoice SOAP -sanomaan. Sanoman lähetys tapahtuu komennolla:

```
requests.put(url, data, headers)
```

Sanoma lähetetään Python requests -moduulin put-metodin avulla. Muut sanoman osat on kuvattu taulukossa 5.

Taulukko 5, lähetettävän sanoman osat

atribuutti	Kuvaus
url	Kutsuttavan API rajapinnan http-osoite
data	Sanomassa lähetettävä hyötykuorma. Tässä tapauksessa Finvoice-laskusanoma
headers	Sanoman lähettämisessä tarvittavat otsaketiedot. Tässä tapauksessa otsakkeessa lähetetään autentikoinnissa tarvittavat tiedot, kuten käyttäjätunnus ja salasana.

6 Johtopäätökset ja pohdinta

Integraatiomoduurin suunnittelu ja toteutus oli vaatavuudesta huolimatta kiinnostava projekti ja sen hyödyt osoittautuivat selkeiksi: ohjelmalla laskut muodostuivat n. 3 minuutissa, kun manuaalisesti tekemällä laskutukseen kuluu useita tunteja. Mikäli laskutiedot saataisiin rajapinnan läpi kaikkine tarpeellisine tietoineen, olisi laskujen muodostaminen äärimmäisen tehokasta.

6.1 Haasteita ja ratkaisuja

Koska tämä projekti oli ensimmäinen Python-ohjelmointikielellä tekemäni sovelluskehitysprojekti, riitti ohjelmointitekniisiä haasteita joka tilanteeseen, esim. kuinka lähdejärjestelmästä luetusta JSON-vastauksesta saadaan poimittua oikeat avain-arvoparit? Tai kuinka XML-muotoinen Finvoice-lasku muodostetaan?

Useampaan kertaan ohjelmointiosaamisen karttuessa huomasin jollekin ohjelman osalle löytyneen tehokkaamman toteutustavan, jolloin edellinen koodi kannatti poistaa ja korvata uudella. Suurin osa ohjelman koodista syntyi erilaisten kokeilujen kautta, eikä sitä ole optimoitu käytännössä lainkaan toimivan toteutustavan löytyttyä.

Tämänhetkisessä koodissa ei myöskään ole käytössä kattavaa virheidenhallintaa, vaan ohjelma kaatuu poikkeustilanteissa virheilmoitukseen. Tällainen tilanne syntyi esimerkiksi, kun asiakasyrityksen nimessä oli tiedostojärjestelmään kelpaamaton merkki. Vastaava kaatuminen merkistöongelman vuoksi syntyi, kun ohjelmaan käytettäväksi osoitettu fonttikirjasto ei tukenut skandinaavisia merkkejä. Näistä selvittiin lisäämällä komponentti, joka korvaa tiedostojärjestelmälle kelpaamattomat merkit toisilla ja vaihtamalla fonttikirjastoa.

6.2 API-rajapinnan rajoitteet

Projektissa törmättiin myös useisiin API-rajapintojen puutteisiin liittyviin haasteisiin, joita joudutaan vielä myöhemmin ratkomaan rajapintojen kehittyessä. Esimerkiksi muodostetulle laskuriville ei pystytty rajapinnan kautta kirjoittamaan katelaskennassa käytettävää

nettohintaa, vaan ongelmaan täytyi tehdä manuaalinen kiertotie: jokaiseen laskuun kirjoitettiin erillinen rivi, jossa kerrotaan nettohinta. Nettohinta täytyy tämän jälkeen käydä laskukohtaisesti tallentamassa laskuriville ja tämä nettohintarivi luonnollisesti poistaa. API-rajapinnan kautta tallennettuun laskuun ei myöskään saada myyjätietoa vietyä, sekin on manuaalisesti käytävä lisäämässä. Näistä puutteista neuvotellaan järjestelmätoimittajan kanssa, josko ne saataisi lisättyä rajapinnan toimintoihin tavoitellun laskutusautomaation saavuttamiseksi.

7 Yhteenveto

Projektin oppina sain hyvän käsityksen järjestelmäintegraation toteuttamisesta API-rajapintoja hyödyntämällä, sekä hyvän perusosaamisen Python-ohjelmoinnista. Vastaavanlaisia integroimattomia järjestelmiä löytyy monista yrityksistä, joten tässä projektissa saatu oppi on hyödynnettävissä myös tulevilla asiakasprojekteilla.

Tutkimuskysymykset saatiin myös selvitettyä. Itse integraation toteutusmalli löytyi ja toimiva ohjelma saatiin toteutettua. Ohjelmiston avulla tehty laskutusaineiston siirto tapahtui murto-osassa manuaaliseen tapaan verrattuna. Tiedonsiirron automaation avulla säästetään työaikaa ja vähennetään manuaalisista kirjauksista johtuvia virheitä.

Työn tuloksena syntynyttä ohjelmistoa tullaan kehittämään jatkossa oman osaamisen sekä API-rajapintojen kehittymisen myötä. Tavoitteena on saavuttaa täysi automaatio, jolloin laskuja ei tarvitsisi enää siirron jälkeen manuaalisesti käsitellä lainkaan. Lisäksi PDF-tiedostojen ulkoasuun tullaan tekemään parannuksia kehittyneemmän PDF-lisäosan löydyttyä, jotta sen ulkoasu saadaan entistä ammattimaisemmaksi.

Lähteet

Autotask. (n.d.-a). Autotask PSA-API -rajapintakuvaus.

https://www.autotask.net/help/developerhelp/Content/APIs/REST/REST_API_Home.htm

Autotask. (n.d.-b). Autotask PSA -entiteetit.

<https://www.autotask.net/help/developerhelp/Content/APIs/REST/Entities/EntitiesOverview.htm>

Autotask. (n.d.-c). Autotask PSA -invoices-rajapinta.

<https://webservices17.autotask.net/at-services-rest/v1.0/invoices/>

Autotask. (n.d.-d). Autotask PSA -companies-rajapinta.

<https://webservices17.autotask.net/at-services-rest/v1.0/companies/>

Base64encode. (n.d.) Verkkopalvelu base64 enkoodaukseen ja dekkoodaukseen.

<https://www.base64decode.org/>

Datto. (2021). *Investor Presentation*.

https://s26.g4cdn.com/278347101/files/doc_presentations/2021/11/Datto-Investor-Presentation-Nov_2021_VFINAL.pdf

Datto. (toukokuu 2021). Luettelo Autotask PSA:n ominaisuuksista.

https://www.datto.com/product-assets/autotask-psa/May-2021-Autotask_PSA_Editions_US.pdf

Finanssiala. (n.d). Verkkolaskutus(Finvoice).

<https://www.finanssiala.fi/aiheet/verkkolaskutus-finvoice/#/>

Finanssiala. (23.5.2011) Finvoice 1.3 Soveltamisohje

https://file.finanssiala.fi/finvoice/FINVOICE_1_3_soveltamisohje_23052011.pdf

Netbaron. (n.d.-a) NetBaron ratkaisut. Haettu 28.11.2021 osoitteesta

<https://www.netbaron.fi/ratkaisumme-sinulle/>

Netbaron. (n.d.-b) NetBaron WS-Laskutus-rajapintakuvaus
NDA:n alainen dokumentti, ei julkisesti saatavana

Pypdf. (n.d) Pypdf-moduulin verkkosivusto

<https://pyfpdf.readthedocs.io/en/latest/>

Tutorialspoint. (n.d.-a). RESTful Web Services - Introduction. Haettu 28.11.2021 osoitteesta

https://www.tutorialspoint.com/restful/restful_introduction.htm

W3.Org. (n.d). Simple Object Access Protocol (SOAP) 1.1.

<https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

Liite 1: Aineistonhallintasuunnitelma

Projektin aikana kerätään kehitysideoita ja koodiesimerkkejä OneNote-työkirjaan aihealueittain koottuina. Itse dokumentaatio, lähdekoodi ja muu projektiin liittyvä materiaali tallennetaan Onedrive-pilvipalveluun, joka on erikseen varmuuskopioitu tiedon katoamisen estämiseksi.

Lisäksi ohjelmakoodia päivitetään säännöllisesti GitHub-tiliin sen ominaisuuksien oppimiseksi.

Opinnäytetyön materiaali säilytetään vähintään 1 vuosi valmistumisen jälkeen.