



samk



Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

LAURI VIITANIEMI

Veroilmoitus-rajapinnan toteutus

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN
TUTKINTO-OHJELMA
2021

Tekijä(t) Viitaniemi, Lauri	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Toukokuu 2021
	Sivumäärä 24	Julkaisun kieli Suomi
Julkaisun nimi Veroilmoitus-rajapinnan toteutus		
Tutkinto-ohjelma Sähkö- ja automaatiotekniikka		
<p>Tämän opinnäytetyön aiheena oli ohjelmoida toiminnallisuus, jolla voidaan lähettää veroilmoitusaineisto suorasiirtona verohallinnon ApitamoPKI-rajapinnalle, ilman välissä olevaa TYVI-operaattoria. Aiemmin veroilmoitusaineisto lähetettiin TYVI-operaattorin kautta. Työn toimeksiantajana toimi CGI Suomi Oy, jonka ohjelmistoon toiminnallisuus rakennettiin.</p> <p>Työ koostui pääpiirteissään veroilmoitusaineiston lähettämisestä ja muodostamisesta, jotka koodattiin X-ohjelmointikielellä. Muodostettava aineisto on muotoa MTOM with XOP, joka sisältää XML-osuuden ja veroilmoituksen.</p> <p>Työssä esitellään aluksi yleisesti rajapintoja ja projektin vaiheita, jonka jälkeen kerrotaan määrittelystä ja toteutuksesta.</p>		
Asiasanat Tietokoneohjelmointi, Rajapinta, XML, Veroilmoitus		

Author(s) Viitaniemi, Lauri	Type of Publication Bachelor's thesis	Date May 2021
	Number of pages 24	Language of publication: Finnish
Title of publication Implementation of tax return -interface		
Degree program Electrical and automation engineering		
<p>Subject of this thesis was to develop functionality that can send tax return –material as direct transfer to tax administrations ApitamoPKI-interface, without TYVI-operator. Previously tax return –material was sent through TYVI-operator. Principal of work was CGI Finland, to whose software the functionality was built.</p> <p>Task consisted mainly of tax return –materials sending, receiving and forming functionalities, which were coded using X programming language. The material to be formed was MTOM with XOP type, which contains XML-part and tax return.</p> <p>In this task, interfaces and project phases are being presented generally. After that, specification and implementation are presented.</p>		
<p><u>Key words</u> Computer programming, Application Programming Interface, XML, Income tax return</p>		

SISÄLLYS

1 JOHDANTO	6
2 RAJAPINTA	7
2.1 Ohjelmointirajapinta	7
2.2 Ohjelmointirajapinnan hyödyt	8
2.3 Ohjelmointirajapinta tyypit	9
2.3.1 Open API	9
2.3.2 Partner API	9
2.3.3 Internal API	9
2.3.4 Composite API.....	9
2.4 XML.....	10
2.5 Web service.....	10
2.5.1 SOAP.....	11
2.5.2 REST.....	11
2.6 ApitamoPKI-rajapinta	12
3 PROJEKTI	13
4 MÄÄRITTELY	14
4.1 Vaatimusmäärittely	14
4.2 Veroilmoitus-rajapinnan vaatimusmäärittely.....	14
5 TOTEUTUS	16
5.1 Veroilmoitusaineiston muodostus.....	17
5.2 Veroilmoituksen lähetys.....	19
5.3 Veroilmoituksen vastaanotto.....	21
5.4 Veroilmoituksen lähetyksen ohjaustiedot	22
5.5 Tietokantataulu.....	23
5.6 Historia.....	23
6 YHTEENVETO	24
LÄHTEET	

SYMBOLI- JA LYHENNELUETTELO

SOAP	Simple Object Access Protocol, tietoliikenneprotokolla
XML	Extensible Markup Language, tiedonvälitysformaatti
MTOM	Message Transform Optimization Mechanism
XOP	XML-binary Optimized Packaging
API	Application Programming Interface
UTF-8	Unicode Transformation Format - 8-bit, merkistö
TYVI	Tietovirrat Yrityksiltä Viranomaisille; palvelu, jonka kautta voi lähettää tietoja viranomaisille
WWW	World Wide Web
REST	Representational State Transfer, arkkitehtuurimalli
RPC	Remote Procedure Call, tietoliikenneprotokolla
HTTP	Hypertext Transfer Protocol, selaimien käyttämä tiedonsiirtoprotokolla
WS-Security	Web Service Security

1 JOHDANTO

Kuvitellaan tilanne, jossa kaksi henkilöä keskustelee keskenään, mutta eivät ymmärrä toistensa kieltä. Sen sijaan heillä olisi tulkit, jotka kääntäisivät toistensa puheet. Tällaisia tilanteita voi olla oikeassa elämässä, mutta keskustelu ei ole kovin jouhevaa ja aikaa kuluu paljon. Lopputuloksena molemmat henkilöt varmasti toki ymmärtävät toisiaan.

Sitten kuvitellaan kaksi sovellusta. Molemmat sovellukset tarvitsevat tietoa toisiltaan, mutta ne eivät kykene lähettämään sitä toisilleen suoraan. Tarvitaan taas tulkki, tässä tapauksessa jokin henkilö, joka lukee tiedot sovelluksesta ja syöttää ne manuaalisesti toiseen sovellukseen. Tiedot menevät kyllä perille, mutta siihen kuluu aikaa ja vaivaa.

Edellä olevia tilanteita helpottaisi rajapinta, jonka avulla osapuolet voisivat keskustella luontevasti. Heillä pitäisi vain olla yhteinen sopimus siitä, miten he kommunikoivat. Ensimmäisessä esimerkissä rajapintana toimisi yhteinen kieli ja toisessa ohjelmointirajapinta, joka välittää tietoa sovellusten välillä. Tässä opinnäytetyössä pureudutaan tilanteeseen, joka muistuttaa jälkimmäistä.

Opinnäytetyön tavoitteena oli luoda toiminnallisuus, jolla voidaan lähettää veroilmoitusaineisto verohallinnolle ApitamoPKI-rajapinnan kautta. Veroilmoitus voi olla esimerkiksi elinkeinotoiminnan veroilmoitus. Työhön kuului koodata oikeanlaiset lähetys- ja sanomanmuodostustoiminnot. Koodaamisessa noudatettiin Verohallinnon teknistä ohjeistusta, jossa kuvataan mm. minkä muotoisia sanomia rajapinnalle voi lähettää. Työn toimeksiantajana toimi CGI Suomi Oy, jonka ohjelmistoon toiminnallisuus tehtiin.

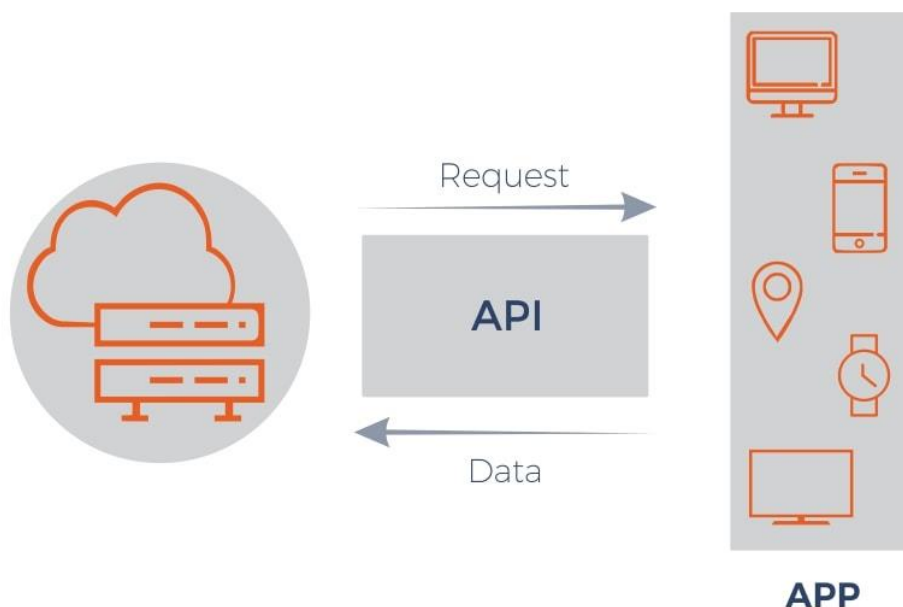
Opinnäytetyössä käydään aluksi läpi yleistä tietoa rajapinnoista: hyödyistä ja tyypeistä. Tämän jälkeen kerrotaan projektin vaiheista ja määrittelystä. Lopuksi kerrotaan varsinaisesta toteutuksesta ja vedetään johtopäätökset.

2 RAJAPINTA

2.1 Ohjelmointirajapinta

Jotta kahden sovelluksen välinen kommunikointi olisi mahdollista, tarvitaan välikäsi, joka tulkitsee tietoa toisilleen. Ohjelmointirajapinta (API) toimii välikätenä, joka mahdollistaa sovellusten välisen kommunikoinnin. Käytännössä sovellukset voivat pyytää ja vaihtaa dataa. Ohjelmointirajapinta on sopimus, joka kertoo, miten sovellusten tulee keskustella. Se koostuu erilaisista säännöistä ja protokollista. Rajapinnasta hyötyvät niin kolmannet osapuolet, yhteistyökumppanit kuin yrityksen sisäiset yksiköt. (IBM, 2020)

Ohjelmointirajapinnan toimintaa on havainnollistettu kuvassa 1. Sovellus (APP) lähettää rajapinnalle pyynnön (Request), jonka rajapinta (API) käsittelee. Jos pyyntö on validi, API ottaa yhteyttä ulkoiseen sovellukseen tai palvelimeen (kuvassa 1. vasemmalla). Sovellus tai palvelin lähettää rajapinnalle vastauksen, joka pitää sisällään pyydettyä dataa. Lopuksi rajapinta välittää datan sovellukselle, joka sen alun perin tilasi. (IBM, 2020)



Kuva 1. API havainnollistettuna. (QATestLab, 2017)

2.2 Ohjelmointirajapinnan hyödyt

Rajapinnoista on paljon hyötyä niin käyttäjille kuin kehittäjille. Käyttäjät eivät ehkä tulekaan ajatelleeksi, kuinka usein he ovat tekemisissä rajapintojen kanssa. Jos haetaan osoitetta Google Mapsista tai jaetaan päivitys Twitterissä, ollaan tekemisissä rajapinnan kanssa. Maailma ei pyörisi samalla tavalla kuin nyt ilman rajapintoja.

Rajapintojen avulla eri sovellukset ja ohjelmat voivat jakaa dataa toisilleen suoraan ilman mitään välikäsiä, josta koituu säästöjä. Kehittäjien työ helpottuu myös, koska heidän ei tarvitse tietää, miten rajapinta on toteutettu. He vain käyttävät rajapintaa kommunikoidakseen muiden palveluiden kanssa. Uusien kumppanuuksien solmiminen on myös helpompaa. (IBM, 2020)

Datan hyödyntäminen on myös tärkeä hyöty. Rajapintojen avulla dataa voidaan kerätä ja yritykset voivat hyödyntää sitä suunnittelussa ja analytiikassa. Analytiikan avulla voidaan parantaa yrityksen tehokkuutta ja tuottavuutta ja luoda parempia palveluita. (Sim, 2019)

Rajapinnat tarjoavat yhteistyölle hyvän perustan. Jaetuilla resursseilla on mahdollista päästä parempiin tuloksiin, kuin sillä, että sovellukset toimisivat yksin ilman vuorovaikutusta. Rajapinnat tarjoavat luotettavaa ja ajantasaista tietoa, sekä vähentää manuaalista työtä, virheiden määrää ja duplikaattitietoa. Yritys voi myös keskittää resurssejaan yhteen paikkaan, josta voi kätevästi löytää hakemaansa tietoa. (Sim, 2019)

Erilaisten teknologien kehittyessä jossain vaiheessa saattaa tulla vastaan haaste, mitä vanhalle teknologialle tehdään. Vanhan ohjelman uudelleen kehittäminen ei välttämättä tule kysymykseen, jos vanhaa koodia on todella paljon. Rajapintojen avulla kuitenkin vanha teknologia voidaan yhdistää uuteen. (Sim, 2019)

Tietoturvanäkökulmasta rajapinnat ovat myös merkittävä asia. Tietoturvariskejä voi aiheutua mm. puutteellisesta koodaamisesta, validoinnista tai tunnistautumisongelmista. Rajapinta toimii eräänlaisena suojakerroksena, joka vaatii tunnistautumisen, jotta dataan pääsee käsiksi. (Sim, 2019)

Rajapinnat mahdollistavat nopeamman sovelluskehityksen. Enää yrityksen ei tarvitse alusta saakka kehittää omia kuvaus-, maksu- tai kommunikointivälineitä. Yritys voi käyttää rajapintaa sisällyttääkseen halutun toiminnallisuuden sovellukseensa. Näin pystytään myös välttämään turhaa toistoa, jossa tehdään samanlaista koodia. (Sim, 2019)

2.3 Ohjelmointirajapinta tyypit

2.3.1 Open API

Open API on avoin rajapinta, joka saatavilla kehittäjille ja muille käyttäjille, minimalistisilla rajoitteilla. Open API voi joko vaatia tunnistautumisen API-avaimella tai voi olla täysin avoin. Se on tarkoitettu ulkoisille käyttäjille, kuten kehittäjille muissa yrityksissä, jotta he pääsevät käsiksi tietoihin. (Stoplight, 2021)

2.3.2 Partner API

Partner API on samankaltainen, kuin open API, mutta partner API:in on rajoitettu pääsy. Se on yleensä tarkoitettu johonkin tiettyyn tarkoitukseen, esimerkiksi se voi tarjota pääsyn maksujenvälitysjärjestelmään. (Stoplight, 2021)

2.3.3 Internal API

Verrattuna avoimiin rajapintoihin, internal API on piilotettu ulkoisilta käyttäjiltä. Sitä käytetään yrityksen sisällä tiedon jakamiseen. Internal API mahdollista eri tiimien tai yksiköiden käyttää toistensa työkaluja, tietoja ja ohjelmia. Internal API:a voidaan pitää parempana siksi, koska siinä on mm. parempi tietoturva. (Stoplight, 2021)

2.3.4 Composite API

Composite API mahdollistaa kehittäjien kutsua useampaa päätelaitetta yhdellä kutsulla. Composite API:t ovat hyödyllisiä mikropalveluarkkitehtuureissa, jossa käyttäjä

saattaa tarvita tietoa usealta palvelulta. Composite API vähentää myös palvelimen kuormitusta ja parantaa sovelluksen suoritusnopeutta. (Stoplight, 2021)

2.4 XML

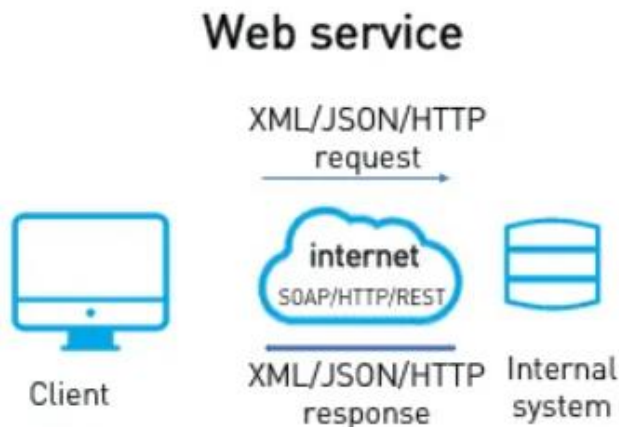
XML on yksinkertainen tekstipohjainen formaatti järjestetyn tiedon esittämiseen. XML on oleellinen formaatti tähän työhön liittyen, koska sitä käytetään SOAP-sanomissa. Toinen vastaavanlainen formaatti on JSON, mutta sitä ei käydä lävitse tässä työssä. XML:ää käytetään tiedon siirtoon eri ohjelmien välillä. XML-dokumentti koostuu alku- ja lopputägeistä, joiden väliin tulee informaatiota. Alku- ja lopputägi kertovat, mistä informaatiosta on kyse. Täten XML on helposti luettavaa. (Quin, 2016)

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DeliveryDataSendRequest xmlns="http://www.vero.fi/xmlschema/ApiTaMo">
      <Language>fi</Language>
      <BackgroundProcessing>false</BackgroundProcessing>
      <Emails>
        <Email>testi@testi.fi</Email>
      </Emails>
      <ReportingMaterial>
        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" />
      </ReportingMaterial>
    </DeliveryDataSendRequest>
  </soap:Body>
</soap:Envelope>
```

Kuva 2. XML-esimerkki.

2.5 Web service

Web service on rajapinta, jonka kautta kaksi laitetta voi kommunikoida tietoverkon kautta. Web service koostuu standardeista ja protokollista, joiden mukaan laitteet keskustelelevat. Yleisesti laitteet keskustelelevat WWW:n välityksellä, käyttämällä XML-, JSON-, tai http-formaatteja tiedon välittämiseen. Pääasiallisesti käytössä on kaksi web service -tyyppiä: SOAP ja REST. Ne eivät kuitenkaan ole itsessään web service -rajapintoja, vaan tapoja toteuttaa web service. (Javatpoint, 2018)



Kuva 3. Web service. (RapidApi, 2020)

2.5.1 SOAP

SOAP on tiedonvälitykseen tarkoitettu XML-pohjainen protokolla, jolla pääsee yhteyteen web servicen kanssa. SOAP toimii http-protokollan yli. Protokolla on sopimus tai standardi, joka määrittää miten kahden laitteen tulisi kommunikoida. SOAP koostuu kolmesta osasta, joiden nimet ovat SOAP envelope, SOAP encoding rules ja SOAP RPC representation. Envelope sisältää varsinaisen viestin tai tiedon, joka halutaan välittää rajapinnan kautta. Esimerkiksi tässä opinnäytetyössä envelope sisältää veroilmoituksen vastaanottajan sähköpostin ja kieliasetuksen. Encoding rules määrittelee, miten alkuperäinen data esitetään xml:nä. RPC representation määrittelee sopimuksen, jota käytetään etäproseduureihin. Etäproseduuria käytetään hajautetun järjestelmän toteuttamiseen. (World Wide Web Consortium (W3C), 2000)

SOAP:issa tiettyjä etuja, jotka kannustavat sen käyttöön. Sen käytössä on parempi tietoturva, koska se käyttää sekä TLS- että WS-Security –protokollaa. Nämä protokollat ovat salausprotokollia, joita käytetään IP-verkkojen yli tapahtuvan tietoliikenteen suojaamiseksi. (Upwork, 2017)

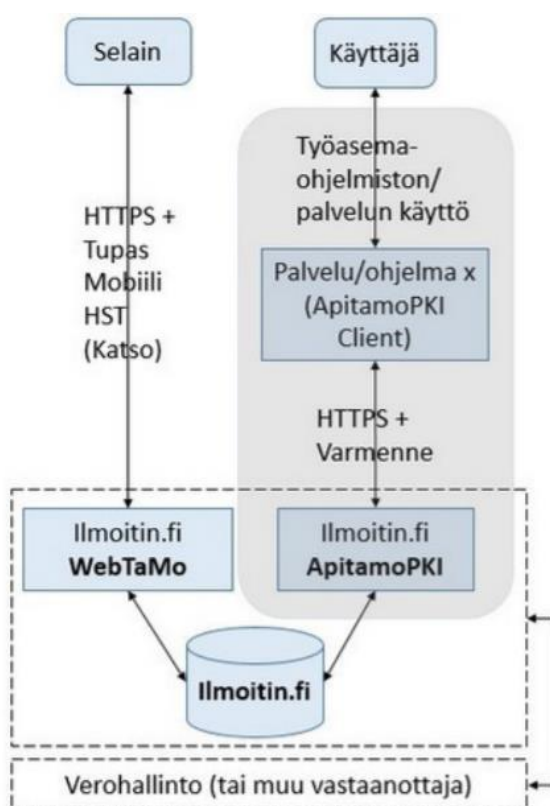
2.5.2 REST

Verrattuna SOAP-protokollaan, REST on arkkitehtuurimalli, joka perustuu http-protokollaan. REST-rajapinnan avulla pääsee myös käsiksi web serviceen. REST on tarkoitettu hajautettuihin järjestelmiin. REST on joustavampi ratkaisu kuin SOAP, koska

REST ei ole ohjelmointikielestä tai alustasta riippuvainen. REST voi käyttää HTML-, XML- ja JSON-formaattia. Web service –rajapintoja, jotka toteuttavat REST-kriteerit, kutsutaan RESTful-rajapinnoiksi. Näitä kriteereitä ovat mm. yhtenäinen käyttöliittymä, tilattomuus ja välimuistiin talletettavuus. (Fielding, 2000)

2.6 ApitamoPKI-rajapinta

Verohallinnolla on rajapinta veroilmoituksen ilmoittamiseen, nimeltään ApitamoPKI, joka on oleellinen osa tätä opinnäytetyötä. Se tarjoaa Web Services –rajapinnan sovelluksille, jonka kautta voidaan tarkistaa mm. vuosi-ilmoituksia ja tuloveroilmoituksia ja lähettää niitä eteenpäin verohallintoon. ApitamoPKI on osa ilmoitin.fi-palvelukokonaisuutta, joka koostuu kahdesta osasta: selainkäyttöisestä WebTamo:sta ja edellä mainitusta Web Services –rajapinnasta. Ilmoitin.fi on palvelu, jonka avulla voidaan tarkastaa ohjelmien tuottamia ilmoituksia ja testata ilmoitusten oikeamuotoisuuksia. (Verohallinto, 2020)



Kuva 4. Ilmoitin.fi-palvelukokonaisuus ja ApitamoPKI. (Verohallinto, 2020)

3 PROJEKTI

Projekti käynnistyi tammikuussa 2021, koronapandemian aikana, joten kaikki työt ja yhteydenpidot hoidettiin etäyhteyksien avulla. Etäyhteyksien kanssa toimimisessa ei ollut silti ongelmia. Työ toteutettiin yhdessä vanhemman ohjelmoijan kanssa, joka auttoi selvittämään verorajapintaan liittyviä teknisiä asioita ja neuvoi ymmärtämään CGI:n ohjelmiston logiikkaa.

CGI:n ohjelmistossa oli olemassa jo vastaavantyyppinen toiminnallisuus, josta alettiin ottaa mallia. Näin tehtiin, jotta ulkoasu ja lähdekoodi olisivat yhdenmukaista. Turha myöskään on pyörää uudelleen keksiä. Toiminnallisuuksissa oli kuitenkin eroja, joten selviteltäviä asioita oli paljon. Keskeisin ohje toiminnallisuuden tekemistä varten oli ApitamopKI:n tekninen kuvaus, jossa kerrotaan toimintaprosessi, viestiliikenne ja sanomakuvaukset. Tämän kuvauksen avulla koodattiin mm. oikeanlaiset lähetys- ja vastaanottotoimet ja sanomanmuodostukset. Verohallinnolta kysyttiin tarvittaessa myös sähköpostitse lisätietoja.

Projektilla ei ollut tarkkaa aikataulua, muuta kuin sen tuli olla valmis kevään aikana. Se helpotti työtä, koska asia oli vierasta ja selvittämiseen kului paljon aikaa. Työssä oppi paljon asiaa sekä rajapinnoista että CGI:n ohjelmistosta.

4 MÄÄRITTELY

4.1 Vaatimusmäärittely

Yleisesti ohjelmistoprojekteissa laaditaan vaatimusmäärittely, jossa kuvataan ohjelmiston tai toiminnallisuuden vaatimukset. Vaatimusmäärittelyn laatiminen on tärkeä tehdä, jotta tiedetään tarkasti mitä ollaan tekemässä, ettei ruveta tekemään mitään epäoleellista. Vaatimusmäärittely jaetaan yleisesti kahteen osaan, toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Toiminnallisissa vaatimuksissa kuvataan, miten ohjelman tulee reagoida käyttäjän syötteisiin ja miten ohjelma käyttäytyy erilaisissa tilanteissa. Ei-toiminnallisissa vaatimuksissa taas kuvataan ohjelman sisäisiä toimintoja. Vaatimusmäärittely koostuu pääpiirteissään toiminnon tai ohjelman esittelystä, käyttäjävaatimuksista ja järjestelmävaatimuksista. (Sommerville, 2016)

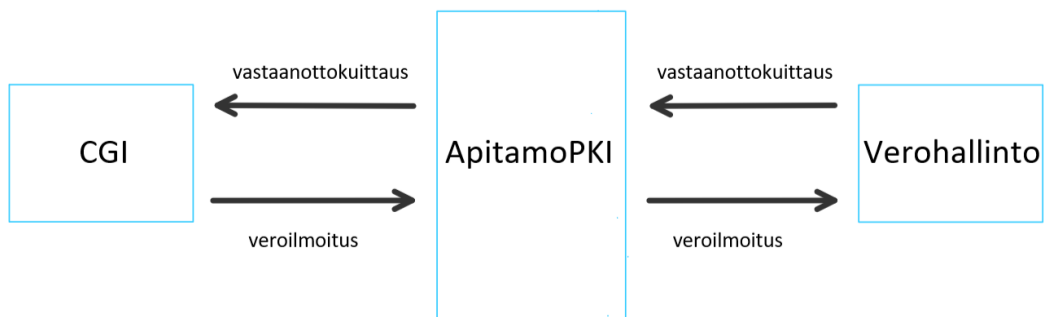
4.2 Veroilmoitus-rajapinnan vaatimusmäärittely

Toimittajaorganisaatio CGI:n ohjelma muodostaa veroilmoitustaineiston, joka toimitetaan TYVI-palvelun kautta verottajalle. TYVI-palvelun kautta voi lähettää tietoja viranomaisille. Työn tavoitteena oli luoda toiminto, jolla veroilmoitustaineistot voidaan lähettää suorasiirtona verottajalle ilman TYVI-palvelua, ApitamoPKI-rajapinnan kautta.

Toiminto koostuu lähetys-, vastaanotto- ja sanomanmuodostustoimista. Lähetystoimen tehtävä on lähettää veroilmoitus ja varmenne ApitamoPKI-rajapinnalle. Lähetystoimi hakee varmenteen ja sitä tarvitaan rajapinnalle tunnistautumiseen. Vastaanotto toimi hakee veroilmoituksesta saatavan palautteen, joka kertoo, oliko veroilmoitus validi vai ei. Sanoman muodostustoimi muodostaa MTOM with XOP –liitteen, joka sisältää veroilmoituksen. Sanomanmuodostustoimea kutsutaan lähetystoimen kautta. Verohallinnon teknisessä kuvauksessa kerrottiin yleisesti toimintaprosessi, miten veroilmoitus tulee lähettää ja minkä muotoista sanomaa rajapinnalle voidaan lähettää. Toiminnallisuuteen sisältyy myös historiaselailu, josta voi katsella eri veroilmoitusten tilaa. Verohallinto ei ole aiemmin mahdollistanut veroilmoituksen suorasiirtoa, joten

nyt sellainen päätettiin tehdä. Kuvassa 5. toimintaa on havainnollistettu. Pääpiirteisään toiminnallisuus on seuraava:

- ohjelma muodostaa veroilmoitusaineiston
- ohjelma lähettää varmenteen ja veroilmoitusaineiston ApitamoPKI-rajapinnalle
- rajapinta tarkastaa veroilmoituksen oikeellisuuden ja lähettää sen verottajalle
- vastauksena verottajalta saadaan vastaanottokuittaus tai mahdolliset tiedot virheellisyyksistä
- hyväksytty aineisto siirtyy verottajan järjestelmiin
- CGI:n ohjelman puolella aineiston tila, eli lähettäminen eri vaiheineen viedään historiatauluun



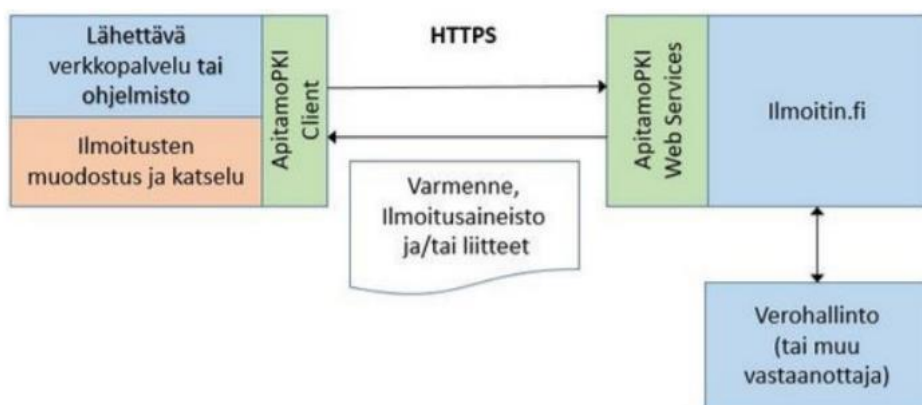
Kuva 5. Toiminnallisuus havainnollistettuna.

5 TOTEUTUS

Varsinainen toteutus tehtiin X-ohjelmointikielellä, joka on CGI:n kehittämä kieli. X-kieli on proseduraalinen kieli, joka muistuttaa C-kieltä. Proseduraalisissa ohjelmoinnissa ohjelman on jaettu prosedureihin eli aliohjelmiin, jossa ohjelman suoritus etenee. CGI:n ohjelmistoon liittyen prosedureja kutsutaan termillä toimi. Esimerkiksi ”Veroilmoituksen lähetys”-toiminnossa on päätoimi, joka kutsuu alitoimia. Päätoimessa alustetaan alitoimet, joissa tehdään muuttuja-alustuksia ja asetetaan käyttöliittymässä näkyviä arvoja. Ohjelman suoritus alkaa päätoimesta ja etenee alitoimiin.

Jotta verottajan rajapintaan saatiin yhteys, oli luettava huolella ApitamoPKI:n teknistä kuvausta. Veroilmoitusaineisto lähetetään ApitamoPKI-rajapinnalle MTOM with XOP-liitteenä, joka sisältää SOAP-sanoman. Liite koostuu XML-osasta ja tunnus:tieto-pari -osuudesta. Yleisesti veroilmoituksen lähettäminen tapahtuu seuraavasti:

- käyttäjä tunnistautuu ohjelmistoon ja valitsee mitä hän haluaa ilmoittaa
- ohjelmisto lähettää ApitamoPKI-rajapinnalle kutsun ja rajapinta tarkastaa ilmoituksen ja sen lähettäjän oikeuden
- ohjelmisto näyttää vastauksen käyttäjälle ja aineisto siirtyy verohallinnon järjestelmiin käsiteltäväksi



Kuva 6. ApitamoPKI periaatekuva. (Verohallinto, 2020)

Veroilmoituksen lähettämistoiminto jaettiin neljään osaan: veroilmoitusaineiston muodostukseen, lähetykseen, palautesanomien vastaanottoon ja ohjaustietoihin. Lähetystoiminnossa lähetetään varsinainen veroaineisto ApitamoPKI-rajapinnalle, vastaanotossa luetaan vastaus, ohjaustiedoissa annetaan hakemisto, jonne aineisto tallentuu ja veroilmoituksen muodostus –toiminto tekee MTOM with XOP –liitteen eli käytännössä veroilmoitusaineiston.

Työt aloitettiin tutustumalla verohallinnon ohjeisiin ja toiseen CGI:n ohjelmassa olevaan toiminnallisuuteen, jossa oli sama idea. Varsinaiset kehitystyöt aloitettiin veroilmoituksen muodostustoiminnon kehittämällä. Verohallinnon teknisessä ohjeessa oli tarkka kuvaus, mitä muodostustoimen muodostaman sanoman tulee sisältää. Muodostustoimen kehittämisen jälkeen ryhdyttiin tekemään lähetystoimea ja lopuksi tehtiin vastaanotto toimi. Toimia ei tehty kerralla valmiiksi vaan niihin palattiin tarvittaessa uudelleen, toisin sanoen niitä kehitettiin limittäin. On järkevämpää tehdä pieni pätkä koodia ja testata sen toimivuus, kuin että tehdään pitkä pätkä ja aletaan etsiä tuskallisesti virheitä.

5.1 Veroilmoitusaineiston muodostus

Veroilmoituksen muodostus –toiminnon tarkoitus on luoda MTOM with XOP –liite, joka koostuu kahdesta osasta: XML-osasta ja tunnus:tieto-parista. XML-osa sisältää rajapinnalle välitettävää tietoa, kuten sähköpostiosoitteen, kieliasetuksen ja osoitteen. Tunnus:tieto-pari –osuus sisältää veroilmoituksen. Kun ohjelma on muodostanut veroilmoituksen, Veroilmoituksen lähetys –toiminto lähettää sen verottajalle.

XML-osan muodostukseen käytettiin X-kielillä luotuja valmiita funktioita. Kuvassa 7. kutsutaan funktiota, joka muodostaa uuden XML-elementin. Sille annetaan kolme parametria eli välitettävää tietoa, joiden perusteella funktio luo elementin. Tässä tapauksessa ensimmäinen parametri on tiedosto, johon kaikki XML-elementit kootaan; toinen parametri on Email-elementti ja kolmas parametri on Email-elementille annettava arvo. Funktioita kutsutaan sen verran, kuinka paljon elementtejä tarvitaan.

```
if(!XMLLisääElementti(XMLTIED,"Email","testi@testi.fi"))
```

Kuva 7. XML-funktio.

Tunnus:tieto-parin muodostus tehdään jo olemassa olevassa Veroilmoitusten siirto – toimessa, josta se liitetään XML:n perään. Kuvassa 8. on Veroilmoitusten siirto –ikkuna, jossa on Vie veroilmoitusten lähetystoimintoon –kohta. Aktivoimalla tämän kohdan, muodostettava veroilmoitus siirtyy Veroilmoitusten lähetystoimintoon, jota kautta se liitetään osaksi veroilmoitusaineistoa.

Veroilmoitusten siirto

Valinnat	
Tilivuosi	<input type="text" value="2018"/>
Kausi	01 - <input type="text" value="12"/>
Otsikko	<input type="text" value="01.01.2018 - 01.01.2018"/>
	<input type="button" value="Valuutta"/> EUR <input type="checkbox"/> Vie veroilmoitusten lähetystoimintoon
Tiedosto	
Kansio ja siirtotiedosto	<input type="text"/> <input type="button" value="P"/>

Kuva 8. Veroilmoitusten siirto.

Tunnus:tieto-pareista koostuva veroilmoitus sisältää kolminumeroisen tunnuksen ja sitä vastaavan tiedon, jotka erotetaan toisistaan puolipisteellä. Esimerkiksi kuvassa 9. tunnusta 000 vastaa tietuetunnuksen nimi, tunnusta 020 vastaa yrityksen nimi ja tunnusta 010 vastaa Y-tunnus.

```
000:VSY06B13
020:CGI 17
010:1234567-8
054:01012013-31122013
478:0,00
999:1
```

Kuva 9. Tunnus:tieto-pareja sisältävä veroilmoitus.

Kuvassa 10. on valmis veroilmoitusaineisto, jonka CGI:n ohjelma lähettää ApitaMoPKI-rajapinnalle. Ensimmäinen elementti kertoo XML-version ja käytettävän merkistön, tässä tapauksessa UTF-8. Body-elementti sisältää tiedot veroilmoituksen tarkastukseen ja lähettämiseen. Bodyn perässä on tunnus:tieto-parit eli veroilmoitus.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <DeliveryDataSendRequest xmlns="http://www.vero.fi/xmlschema/ApiTaMo">
      <Language>fi</Language>
      <BackgroundProcessing>false</BackgroundProcessing>
      <Emails>
        <Email>testi@testi.fi</Email>
      </Emails>
      <ReportingMaterial>
        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" href="cid:316b1781-ead8-4671-bef5-d2b3b6073414-5@www.vero.fi">
        </xop:Include>
      </ReportingMaterial>
    </DeliveryDataSendRequest>
  </soap:Body>
</soap:Envelope>
--uuid:56b148e5-4de1-4929-a0b1-aebb3853030a
Content-Type: text/plain
Content-Transfer-Encoding: binary
Content-ID: <316b1781-ead8-4671-bef5-d2b3b6073414-5@www.vero.fi>
Content-Disposition: attachment;name="JokuTiedostonNimi.txt"
000:VSY06B18
048:Sonet
198:30032021103236
020:CGI 17
010:1234567-8
054:01012018-01012018
001:0
478:0,00
999:1
--uuid:56b148e5-4de1-4929-a0b1-aebb3853030a--
```

Kuva 10. Veroilmoitusaineisto MTOM with XOP –muodossa.

5.2 Veroilmoituksen lähetys

Kuten jo aiemmin mainittiin, veroilmoitus lähetetään Veroilmoituksen lähetys –toiminnolla. Jotta veroilmoitus voidaan lähettää toiminnolla, pitää tunnus:tieto-pari -muotoinen veroilmoitus olla muodostettu jo aiemmin. Kuvassa 11. on veroilmoituksen lähetysikkuna, josta veroilmoitus voidaan lähettää. Muodostuspvm-kentän avulla voidaan hakea tiettyinä päivämääränä muodostettuja veroilmoituksia ja kun ollaan valittu päivämäärä, painetaan Hae tiedot –painiketta. Tämän jälkeen veroilmoitukset tulevat

scrolliin eli selailuun, josta voidaan valita veroilmoitusaineisto ja lähettää se ApitamopKI-rajapinnalle painamalla Lähetä-painiketta. Historia-painikkeen pääsee selailemaan aineistojen tilaa eli missä vaiheessa ne ovat.

Jotta veroilmoituksen lähettäminen on ylipäättään mahdollista, tarvitaan varmenne, joka haetaan asiakkaan y-tunnuksen mukaan. Varmenne on veroilmoituksen lähetyksessä oleva identifioiva tunnus, joka oikeuttaa lähettää veroilmoituksia ApitamopKI-rajapinnan kautta.

Lomake-sarake kertoo, mistä veroilmoituksesta on kyse: tässä yhteydessä esiintyy 6B-lomake eli elinkeinotoiminnan veroilmoitus. Tunnustiedot kertovat jakson, jolta ajalta veroilmoitusaineisto on lähetetty. Tyyppi kertoo, missä tilassa veroilmoitus on: onko se uusi, lähetetty, kuitattu vai hylätty. Pvm-sarakkeet kertovat milloin veroilmoitus on muodostettu, lähetetty, kuitattu ja valmis. Aineistotunnus-sarake kertoo veroilmoituksen sijainnin tietokoneen levyllä tai palvelimella. Veroaineisto ilmoittaa valmiin veroilmoitusaineiston sijainnin tietokoneella.

Veroilmoitusten lähetykset

Rajaus								
Muodostuspvm		<input type="text" value="6B"/>	Lähetä		Hae palaute		Historia	
Hae tiedot								
Lomake	Tunnustiedot	Tyyppi		Muodostuspvm	Lähetykspvm	Valmistuspvm	Aineistotunnus	Veroaineisto
6B	01122021	5	Hylätty	8.4.2021	8.4.2021		c:\oma\vero2021.txt	
6B	01122013	5	Hylätty	1.4.2021	1.4.2021		C:\temp\vero2013	
6B	01122014	5	Hylätty	31.3.2021	31.3.2021		c:\temp\vero.txt	
6B	01122017	5	Hylätty	31.3.2021	30.3.2021		c:\temp\vero.txt	
6B	01122015	5	Hylätty	30.3.2021	30.3.2021		C:\temp\vero2016	
6B	01122016	5	Hylätty	30.3.2021	30.3.2021		C:\temp\vero2016	
6B	01122018	5	Hylätty	30.3.2021	30.3.2021		C:\temp\verotesti2	
6B	01122019	5	Hylätty	30.3.2021	30.3.2021		c:\temp\vero2019.txt	

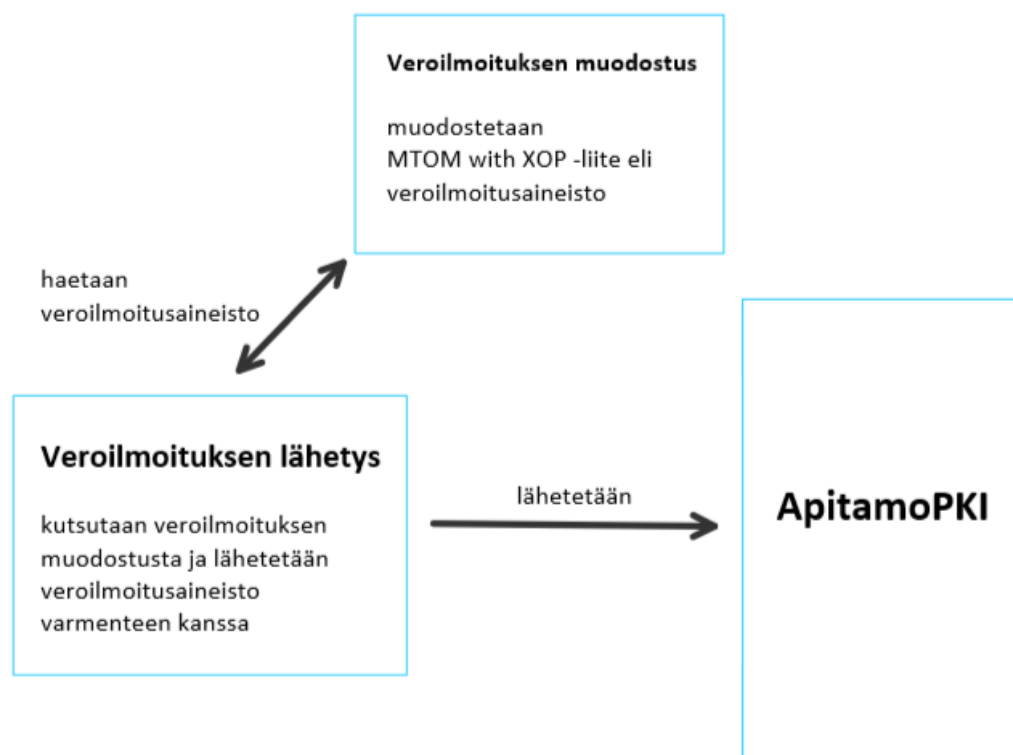
Kuva 11. Veroilmoituksen lähetyksikkuna

Ohjelma antaa erilaisia virheilmoituksia tilanteen mukaan, kun aineistoa ollaan lähettämässä. Jos veroilmoituksen tyyppi on 2, 3 tai 4; ohjelma ilmoittaa ”Aineisto on jo lähetetty.” Jos tyyppi on 5, ohjelma ilmoittaa ”Aineisto on hylätty.” Jos tyyppi on 1, ilmoitusta ei anneta, koska on kyseessä uusi veroilmoitus ja lähetykset voidaan suorittaa.

Tyyppi muuttuu scrolliin automaattisesti sitä mukaa, missä tilanteessa veroilmoitus on kyseiselläkin hetkellä.

Koodin suoritus etenee yleisellä tasolla seuraavasti:

- Lähetä-painikkeen painamisen jälkeen lähetys käynnistyy ja ensimmäiseksi haetaan varmenne
- muodostetaan veroilmoitusaineisto eli MTOM with XOP –liite kutsumalla Veroilmoitusaineiston muodostus –toimintoa
- aineisto lähetetään ApitamoPKI-rajapinnalle.



Kuva 12. Veroilmoituksen lähetys havainnollistettuna.

5.3 Veroilmoituksen vastaanotto

Kun veroilmoitusaineisto on lähetetty verohallinnolle ApitamoPKI-rajapinnan kautta, sieltä saadaan vastaus, oliko aineisto validi vai ei. Vastaus haetaan painamalla kuvassa 11. olevaa Hae palaute –painiketta. Toiminto lähettää XML-muotoisen kyselysanoman. Jos vastaus tulee hyväksyty-tilassa, aineisto päivitetään valmis-tilaan. Vastavasti jos vastaus oli virheellinen, syitä voi olla:

- veroilmoitusaineistosta puuttuu jotakin tai se ei mene tarkastuksesta läpi
- virheellistä tietoa, xml-tiedoston rakenne on virheellinen
- ei saa yhteyttä rajapintaan

5.4 Veroilmoituksen lähetyksen ohjaustiedot

Ohjaustiedot määrittävät, voidaanko rajapinnalle lähettää veroilmoituksia. Rajapinta ohjaus –kohdassa on siihen kolme vaihtoehtoa, jotka näkyvät kuvassa 13. Jos rajapinta on käytössä, veroilmoituksia voidaan lähettää. Rajapinnan ollessa testikäytössä, voidaan veroilmoituksia lähettää verohallinnon testiympäristöön, johon tarvitaan testitunnukset. Jos rajapinta ei ole käytössä, varmennetta ei ole haettu ja täten lähetystä ei voida tehdä. Käytännössä Veroilmoituksen lähetyksen ikkunaan ei tällöin päästä. Veroilmoituksen siirto –toiminnossa muodostetut veroilmoitukset tallentuvat näytöllä olevan Aineistojen hakemisto –kentän sijaintiin tietokoneen levyllä.

Veroilmoituksen lähetyksen ohjaustiedot

Rajapinta ohjaus

Tekninen rajapinta on käytös...
 Tekninen rajapinta on käytössä testissä (poikkeustilanteet)
 Tekninen rajapinta ei ole käytössä

Huom: Testi-valintaa käytetään vain tilanteessa, jossa teknistä rajapintaa testataan tulorekisterin kanssa. Testiympäristön käyttö edellyttää testitunnusten käyttöä. Ne saa Tulorekisteristä testisopimuksen teon yhteydessä.

Hakemisto ohjaus

Aineistojen hakemisto (kaikki aineistot muodostuvat tähän palvelimella olevaan hakemistoon)

Testiympäristön osoite:

<https://apitesti.ilmoitin.fi/wsapp/apitamopki?wsdl>

Kuva 13. Ohjaustiedot.

Ohjelmassa oli jo olemassa varmenteen hakuun tarkoitettu toiminto. Tulorekisterin nettisivuilta, joka kuuluu käytännössä verohallintoon, tehdään hakemus teknisen rajapinnan käyttöönottamiseksi. Sinne kirjaudutaan yrityksen tunnuksilla. Hakemuksessa

valitaan kanava (Web Service), rajapinnan käyttötarkoitus ja tekninen asiantuntija, joka tekee varmenteen hakemuksen.

5.5 Tietokantataulu

Muodostuneen veroilmoitusaineiston tiedot talletetaan tietokantatauluun. Taulussa olevat tiedot vastaavat Veroilmoitusten lähetys –ikkunassa olevia sarakkeiden nimiä.

Tietokantataulu

Taulun nimi BASE TABLE

<input type="checkbox"/>	Nro	Sarakenimi	Tyyppi
<input type="checkbox"/>	0	YRTUN	CHAR
<input type="checkbox"/>	1	MAGDA_SQ	VARBINARY
<input type="checkbox"/>	15	VL_LOMAKE	CHAR
<input type="checkbox"/>	16	VL_TUNNUSTIEDOT	CHAR
<input type="checkbox"/>	18	VL_TYYPPI	CHAR
<input type="checkbox"/>	20	VL_MUODOSTUSPVM	TIMESTAMP
<input type="checkbox"/>	21	VL_LAHETYS PVM	TIMESTAMP
<input type="checkbox"/>	24	VL_VALMISTUSPVM	TIMESTAMP
<input type="checkbox"/>	27	VL_SELITE	CHAR
<input type="checkbox"/>	29	VL_AINEISTO	CHAR
<input type="checkbox"/>	30	VL_KUITTAUSPVM	TIMESTAMP

Kuva 14. Tietokantataulu.

5.6 Historia

Veroilmoitusaineiston tila (uusi, valmis, lähetetty tai hylätty) viedään historiatauluun, joka muistuttaa kappaleessa 5.5 mainittua tietokantataulua, mutta on kuitenkin eri. Historiataulu tehtiin, jotta asiakas pääsee näkemään aineiston historian eli miten käsittely on edennyt. Historiataulu toimii lokina, josta näkee, kuka on suorittanut aineiston eri käsittelyvaiheet.

6 YHTEENVETO

Rajapinnan käyttö helpottaa huomattavasti veroilmoituksen lähettämistä, koska erillistä TYVI-operaattoria ei sitten enää tarvita. Toiminnallisuutta tullaan jatkokehittämään tulevaisuudessa lisää, erilaisten veroilmoitusten yhteyteen. Lisäksi veroilmoitusten mukana xml-sanomien yhteydessä pystytään tätä kautta jatkossa toimittamaan niihin kuuluvat liitteet. Tässä työssä toiminnallisuus tehtiin vain elinkeinotoiminnan veroilmoituksen yhteyteen. Mielestäni rajapintojen hyödyntäminen on oikein hyvä asia, koska ne suoraviivaistavat kaikkea toimintaa, niin yrityksen kuin käyttäjänkin kannalta. Tietoa on helpompi välittää järjestelmien välillä, palveluita pystytään tehostamaan ja ihmisten arki helpottuu. Rajapintojen tekeminen saattaa olla hidasta ja aikaa vievää, mutta lopussa se kantaa hedelmää.

Tässä opinnäytetyössä luotiin toiminnallisuus, jolla voidaan lähettää veroilmoitus ApitamoPKI-rajapinnan kautta verohallinnolle. Työ koostui veroilmoituksen lähetys-, vastaanotto- ja muodostustoimintojen kehittämisestä. Kehittämiseen käytettiin X-ohjelmointikieltä. Lisäksi tutkittiin yleistä asiaa ohjelmointirajapinnoista ja niiden hyödyistä.

Haasteena projektissa oli veroilmoituksen lähettäminen. Oli epäselvää, pystytäänkö CGI:n ohjelmasta lähettämään veroilmoitusaineisto jo olemassa olevilla funktioilla. Verohallinnon teknisessä kuvauksessa kaikki ohjeet eivät olleet tyhjentyviä, joten lisätietoa kysyttiin verohallinnon tekniseltä puolelta.

Projekti oli opettava kokemus siitä, miten ohjelmistoon luodaan kokonainen uusi toiminnallisuus. Tällaisen tekeminen toimii hyvänä starttina työelämään, jos aikoo vastaavia toiminnallisuuksia tehdä. Projektin kautta X-kieli tuli paremmin tutuksi ja bisneslogiikan ymmärrys kasvoi. Rajapinta-asian kautta tutuiksi tulivat rajapintojen peruserätyypit, hyödyt ja tyypit.

LÄHTEET

Fielding, R. T. 2000. Architectural Styles and the Design of Network-based Software Architectures. Irvine: University of California

IBM. 2020. Application Programming Interface. Viitattu 09.03.2021.

<https://www.ibm.com/cloud/learn/api>

Javatpoint. 2018. What is Web Service. Viitattu 25.03.2021.

<https://www.javatpoint.com/what-is-web-service>

QATestLab. 2017. API (Application Programming Interface). Viitattu 25.03.2021.

<https://qatestlab.com/resources/knowledge-center/application-programming-interface/>

Quin, L. 2016. Extensible Markup Language. Viitattu 18.02.2021.

<https://www.w3.org/standards/xml/core>

RapidApi. 2020. API vs Web Service: What's the Difference? Viitattu 26.03.2021.

<https://rapidapi.com/blog/api-vs-web-service/>

Sim, I. 2019. The 6 Key Benefits of Being an API-Centric Organization. Viitattu 13.03.2021.

https://uploads-ssl.webflow.com/595bd9c5fbd4b66110fd46d3/5db3569fe7717a87345d0f1b_The%206%20Key%20Benefits%20of%20Being%20an%20API-Centric%20Organzation%20by%20Radical%20IO.pdf

Sommerville, I. 2016. Software Engineering. Englanti: Pearson Education.

Stoplight. 2021. Types of APIs and Popular REST API Protocol. Viitattu 12.03.2021.

<https://stoplight.io/api-types/>

Upwork. 2017. SOAP vs. REST: A Look at Two Different API Styles. Viitattu 7.4.2021.

https://www.upwork.com/resources/soap-vs-rest-a-look-at-two-different-api-styles?vt_cmp=11384804789&vt_adg=113089129402&vt_src=google&vt_kw=&vt_device=c&utm_source=google&utm_campaign=11384804789&utm_medium=paid_search&gclid=Cj0KCQjwmIuDBhDXARIsAFITC_5BhwW

Verohallinto. 2020. Tekninen kuvaus, ApitamopKI-rajapinta. Viitattu 18.02.2021.

https://www.vero.fi/globalassets/tietoa-verohallinnosta/ohjelmistokehittajille/verohallinto_apitamopki_tekninen_kuvaus_v1.4_fi.pdf

W3C. 2000. Simple Object Access Protocol (SOAP). Viitattu 26.03.2021.

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