Web-maksupalvelujen suunnittelu ja toteutus PSD2-direktiivin mukaisesti

Diplomityö
Turun yliopisto
Tulevaisuuden teknologioiden laitos
Tietotekniikka
2020
Niklas Byggmästar

TURUN YLIOPISTO
Tulevaisuuden teknologioiden laitos

NIKLAS BYGGMÄSTAR: Web-maksupalvelujen suunnittelu ja toteutus PSD2-direktiivin mukaisesti

Diplomityö, 63 s.
Tietotekniikka
Joulukuu 2020

Toisen maksupalveludirektiivin (engl. Second Payment Services Directive, PSD2) tultua voimaan vuonna 2016 on maksupalveluiden kehittämisen ja tarjoamisen mahdollisuuksissa näkynyt muutoksia. Maksupalveluja tarjoavien yritysten, kuten pankkien, on tämän direktiivin nojalla tarjottava kolmansille osapuolille mahdollisuudet käyttää turvallisesti ja luotettavasti pankin palveluita ja dataa siihen luvan antaneen loppuasiakkaan puolesta. Uudistus tuo mukanaan myös paremmat mahdollisuudet lukea verkkopankkien tiliotteita ja lähettää tilisiirtokomentoja vakioitujen rajapintojen avulla.
PSD2-direktiivin keskeisimpiä tavoitteita on luoda Euroopan Unionin alueelle yhtenäisempi, standardisoitu maksualue sekä loppuasiakkaille että maksupalveluiden tarjoajille. Tällä pyritään myös edistämään kilpailua maksupalveluiden alalla, luoden parempia mahdollisuuksia ja valinnanvaraa palveluiden käyttäjille.
Tässä diplomityössä esitellään PSD2-direktiiviä hyväksi käyttävän maksupalvelukomponentin kehitystä ja arkkitehtuuria, sekä tutkitaan PSD2-direktiivin vaikutuksia yleisesti maksupalveluiden kehitykseen ja muutoksia aiempiin säädöksiin. Työssä esitellään myös arviointia suunnitellun ja kehitetyn maksupalvelun toteutuksesta sekä uuden direktiivin tarjoamista tavoista edesauttaa sen tavoitteiden toteutumista.

Asiasanat: PSD2, Toinen maksupalveludirektiivi, Maksupalvelut, Maksupalvelun tarjoajat, Ohjelmistoarkkitehtuuri, Tietoturva, ASP.NET Core, Docker

Sisällys

1 Johdanto

Verkossa maksaminen on muun muassa internetin ja mobiilimaksamisen kehityksen myötä yleistynyt nopeasti kasvavalla tahdilla, ja tullut jo tavalliseksi osaksi ostosten tekoa ympäri maailmaa. Tämän kehityksen myötä eri alojen kauppiaiden verkkokaupat ovat lisääntyneet, ja tämän kautta puolestaan kaikenlaiset verkossa maksamiseen liittyvät järjestelmät yleistyvät ja kehittyvät jatkuvasti. Näitä ovat esimerkiksi kaikenlaiset maksupalvelut ja maksupalvelun tarjoajat, joiden avulla voidaan tehdä verkkomaksamisesta helpompaa sekä kauppiaalle että palvelua käyttävälle asiakkaalle.

Tässä diplomityössä esitellään toista maksupalveludirektiiviä (engl. Second Payment Services Directive, PSD2) hyväksi käyttävää projektia, jonka tarkoituksena on luoda yksinkertainen ja helppokäyttöinen maksupalvelukomponentti, nimeltään Ajas Payments. Ohjelmiston osalta arvioidaan ja käydään läpi sen kehitystä ja arkkitehtuuria, sekä tutkitaan PSD2-direktiivin vaikutuksia maksupalveluiden kehitykseen. Työ tehdään ohjelmistoyritykselle nimeltä Eneroc Oy, joka kehittää muun muassa omaa tuotettaan, Ajas-ajanvarauspalvelua, jossa Ajas Payments -maksupalvelua tullaan ensisijaisesti käyttämään.

Maksupalvelulla tarkoitetaan yleisesti palveluntarjoajaa, joka ottaa vastaan asiakkaidensa toimeksiantoja varojen siirtämiseksi toisille osapuolille. Maksupalveluihin kuuluvat esimerkiksi erilaiset maksutilien väliset tilisiirrot, ostokset maksukorteilla, suoramaksut, rahanvälitys ja matkapuhelimella tehdyt maksut kolmansille osapuolille. Maksupalveluja ohjaavat Suomessa erilaiset lait, kuten maksupalvelulaki sekä maksulaitoslaki.

Maksupalvelun tarjoajalla tarkoitetaan tyypillisesti sellaista kolmatta osapuolta, joka

toimii maksajan ja maksunsaajan välikätenä, eli siirtää varoja osapuolten välillä heiltä saadun toimeksiannon perusteella. Maksupalvelun tarjoajat tuovat esimerkiksi verkkokaupan ylläpitäjille ja näiden asiakkaille sähköiset mahdollisuudet haluttujen maksupalvelujen käyttöön ja maksujen tekoon verkon välityksellä. Maksupalvelun tarjoaja tyypillisesti tarjoaa käyttäjälle mahdollisuuden useisiin eri maksutapavaihtoehtoihin, kuten eri pankkien verkkomaksuihin, luottokorttimaksuihin tai mobiilimaksuihin. Maksupalveluja saa tarjota Suomessa ainoastaan ne toimijat, joille Finanssivalvonta on myöntänyt toimiluvan, joten toimiala on luonnollisesti hyvin säädeltyä ja valvottua.

Työssä viitataan maksupalvelun asiakkaisiin, eli maksupalvelua käyttäviin toisiin yrityksiin joko sanoilla *asiakas* tai *kauppias*. Maksupalvelun *loppukäyttäjä* tai *loppuasiakas* on jokin henkilö, joka käyttää tietyn yrityksen maksujärjestelmää joko Ajas-ajanvarausjärjestelmässä tai -verkkokaupassa. Täten Eneroc Oy:n asiakkaita ei tässä työssä siis pidetä tuotteen loppukäyttäjinä.

Ottaakseen maksuominaisuudet käyttöön, tulee Ajas-palvelua käyttävän yrityksen tällä hetkellä hankkia itse erikseen maksupalvelu esimerkiksi Checkout Finlandilta. Tämä on ollut usein asiakkaille turhan vaikeaa, eikä erityisen joustavaa. Maksupalvelujen toteutukset ovat olleet aiemmin myös kohtalaisen työläitä ja kalliita toteuttaa, johtuen pitkälti siitä, että eri pankkien rajapintojen toiminnassa on ollut paljon eroja. Tämän lisäksi maksualan lakisäätely on ollut voimakasta, minkä takia omien maksupalveluiden tekeminen ei ole ollut kovin yleistä. Aiemmin verkkopankkeihin siirtoja tehdäkseen on täytynyt toteuttaa eri verkkopankkeihin maksurajapintaan oma integraatio.

Toteutettavan maksupalvelun yhtenä tarkoituksena on pyrkiä saamaan palvelun integrointi yrityksen omaan Ajas-ajanvarausjärjestelmään ja -verkkokauppaan käyttäjälle mahdollisimman yksinkertaiseksi. Maksupalvelu on tarkoitus pystyä integroimaan muihin verkkokauppoihin myöhemmissä sovelluksen kehityksen vaiheissa, mutta tämän diplomityön osalta tavoitteena on integrointi ainoastaan Ajas-tuotteisiin. Maksupalvelussa on tarkoitus käyttää hyväksi uutta PSD2-maksupalveludirektiiviä muun muassa helpomman

käyttäjäkokemuksen luomiseksi. Tuotteen markkinointimalli on yritysmarkkinointi (engl. business-to-business, B2B), eli tuotetta kaupataan ainoastaan toisille yrityksille.

Tavoitteena on suunnitella ja toteuttaa maksupalvelu, jonka asiakas voi milloin tahansa lisätä ostamaansa verkkokauppaan tai ajanvarausjärjestelmään. Tällöin loppuasiakas maksaa Eneroc Oy:lle, josta Eneroc tilittää provision jälkeisen osuuden palveluntarjoajalle uuden PSD2-direktiivin mukanaan tuomien automaattisten pankkiyhteyksien kautta. Ratkaisu ei ota kantaa siihen, mitä kautta loppuasiakkaalta veloitus tehdään, jolloin moderneja maksutapoja voidaan lisätä helposti järjestelmään.

Toisen maksupalveludirektiivin (PSD2) tultua voimaan vuonna 2016 on maksupalveluja tarjoavien yritysten, kuten pankkien, tämän direktiivin nojalla tarjottava kolmansille osapuolille mahdollisuudet käyttää turvallisesti ja luotettavasti pankin palveluita ja dataa siihen luvan antaneen loppuasiakkaan puolesta. Uudistus tuo mukanaan muun muassa paremmat mahdollisuudet lukea verkkopankkien tiliotteita ja lähettää tilisiirtokomentoja vakioitujen rajapintojen avulla, ja näitä ominaisuuksia on tässä työssä käytetty hyväksi. [1]. Uuden direktiivin myötä byrokratia keventyy ja uusien maksupalveluiden kehittäminen on täten helpompaa.

Työn toisessa luvussa käsitellään tarkemmin palvelun vaatimuksia ja arkkitehtuuripäätöksiä, sekä käytettäviä teknologioita ja niiden valinnan perusteluita. Kolmannessa luvussa käsitellään web-pohjaisia maksupalveluja ja niiden tarjoajia, sekä avataan tarkemmin näihin liittyviä toimilupia, sekä näiden hyötyjä eri osapuolille. Neljännessä luvussa esitellään työn osalta suuressa roolissa olevaa PSD2-direktiiviä yksityiskohtaisemmin, ja käydään läpi direktiivin edeltäjää ja sen sisältämiä parannustarpeita. Viidennessä luvussa käsitellään tarkemmin itse käytännön työn suunnitelmaa ja implementaatiota. Pääasiallisina aiheina tässä luvussa ovat tietoturva, maksujen suoritusprosessit ja palvelun jatkokehitys.

Tämän diplomityön tutkimuskysymyksenä on tutkia, mitkä ovat olleet PSD2-direktiivin vaikutukset web-pohjaisten maksupalveluiden kehittämiseen, sekä miten maksupalvelui-

den alalla tapahtuva kilpailu on kehittynyt tämän direktiivin myötä.

Kuudennessa luvussa arvioidaan toteutetun työn eri laatuattribuuttien toteutumista ja esitetään mahdollisia parannusehdotuksia. Näiden lisäksi luvussa pyritään vastaamaan diplomityön tutkimuskysymykseen arvioimalla käytetyn PSD2-direktiivin vaikutuksia maksupalveluiden kehitykseen.

2 Vaatimukset ja arkkitehtuuripäätökset

Työn soveltavan osuuden tavoitteena on suunnitella ja toteuttaa uusi maksupalvelujärjestelmä, joka tulisi pystyä mahdollisimman helposti integroimaan Ajas-ajanvarauspalveluun tai -verkkokauppaan. Kehitettävän maksupalvelun avaamisen jälkeen tulisi riittää, että kaupan ylläpitäjä syöttää pankkitilinsä numeron ja yrityksensä tarvittavat tiedot. Tämän jälkeen tarkoituksena on, että maksusivu voidaan ottaa käyttöön esimerkiksi kauppiaan verkkokaupassa helpon rajapinnan avulla. Mihin tahansa verkkokauppaan tai maksusivulle voidaan jatkossa lisätä Ajas Payments -palvelu. Ensisijaisesti palvelun integrointi ainoastaan Ajas-tuotteiden omiin maksujärjestelmiin on kuitenkin riittävä.

2.1 Toiminnalliset vaatimukset

Maksupalvelussa tulisi toteuttaa seuraavat ominaisuudet ja yksityiskohdat. Maksupalvelun rakenne tulisi perustua siihen, että Ajas Payments käyttää olemassa olevia maksurajapalveluita, ja aluksi käytettävä maksupalvelu on Checkout Finland, jonka avulla Ajas Payments veloittaa loppuasiakasta. Kun loppuasiakkaan suorittama maksu on veloitettu, maksusuorituksen tiedot kirjataan Ajas Payments -järjestelmän tietokantaan. Suoritetusta maksusta järjestelmä suorittaa tapahtuman, joka laskee tilisiirrot eteenpäin. Ajas Payments ottaa maksun kokonaissummasta tietyn provision, jonka jälkeen jäljelle jäävä osa siirretään suoraan palveluntarjoajan tilille seuraavan siirto-operaation aikana. Maksupal-

velu käsittelee ainoastaan euromääräisiä SEPA-maksuja. Toteutetaan alussa prototyyppinä yhteys Osuuspankkiin, jolla saadaan SEPA-maksutiedostot vietyä tilityksiä varten.

2.2 Laatuattribuutit

2.2.1 Muunneltavuus ja modulaarisuus

Tärkeää maksupalvelun toteutuksessa on myös sen helppo muunneltavuus, vaikka alkuvaiheessa toteutus tehdään vain Ajas-tuotteisiin integrointiin keskittyen. Kuitenkin kehityksen myöhemmässä vaiheessa Ajas Paymentsia olisi tarkoitus alkaa käyttämään myös muilla sivustoilla maksupalveluna. Muunneltava ohjelmisto on tyypillisesti helpompi ylläpitää, ja erityisesti muokata, sillä tätä periaatetta noudattavassa ohjelmakoodissa riittää tyypillisesti jonkin yksittäisen osion muokkaus ilman, että tarvitaan muutoksia muualla, missä kyseistä osiota käytetään.

Ohjelmiston ja sen koodin muunneltavuus on sen laadun ja jatkokehityksen kannalta erittäin tärkeää ottaa huomioon. Jotta ohjelmistosta saadaan helposti muunneltavaa, on seurattava tiettyjä standardeja sen arkkitehtuurissa. Hyviä esimerkkejä muunneltavuuden saavuttamiseksi on esimerkiksi kovakoodattujen arvojen välttäminen ja ohjelmiston käyttämän datan muokkausmahdollisuus itse ohjelmiston avulla ilman ohjelmakoodin muokkausta. [2]

Näiden lisäksi myös ohjelmiston modulaarisuus edesauttaa muunneltavuutta. Modulaarisuus tarkoittaa sitä, että ohjelmistokoodia on jaoteltu pienempiin kokonaisuuksiin niin, että kyseisten kokonaisuuksien välillä ei ole vahvoja riippuvuussuhteita ja kokonaisuuksilla on selkeät ja samantyyppiset rajapinnat. Heikot riippuvuussuhteet komponenttien välillä tarkoittaa, että komponentteja voidaan helposti käyttää täysin tai lähes itsenäisinä ohjelminaan, niitä ympäröivästä kontekstista välittämättä. [3]

2.2.2 Skaalautuvuus

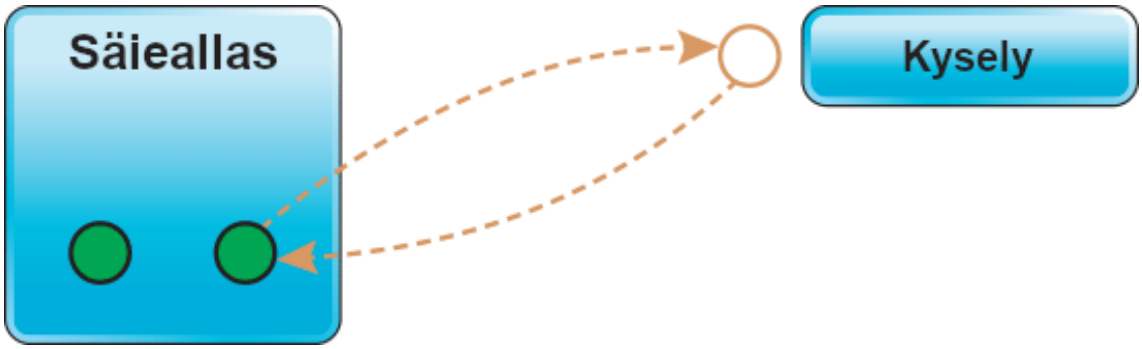
Työssä on myös tärkeää ottaa huomioon palvelun skaalautuvuus, joka toisaalta koskee yhtä lailla palvelun muunneltavuutta ohjelmistokoodin ja sen kehityksen osalta, mutta myös koskee ajan myötä lisääntyvää käyttäjien ja datan määrää. Skaalautuvuudella tarkoitetaan, että palvelun tulisi pystyä palvelemaan käyttäjää samalla tavalla ja yhtä tehokkaasti, riippumatta siitä kuinka paljon dataa on tietokannassa, tai kuinka moni muu käyttää palvelua samanaikaisesti kerrallaan. Palvelun hyvä skaalautuvuus puolestaan edesauttaa palvelun saatavuutta, eli miten hyvin palvelun ohjelmisto pysyy toiminnassa, eikä esimerkiksi kaadu.

Palvelu tullaan todennäköisesti viimeistään tarkemmassa testausympäristön luonnissa sijoittamaan jollekin pilvialustalle, kuten Amazonin tarjoamalle AWS-alustalle (Amazon Web Services). AWS tarjoaa muun muassa hyvät mahdollisuudet palvelun suorituskyvyn ja tapahtumien valvonnalle, kustannustehokkuudelle, turvallisuudelle ja skaalautuvuudelle [4]. Amazonin tarjoamaa pilvipalvelua käyttämällä tehtäisiin siis tällöin sovelluksen tarjoamisen lisäksi myös muun muassa sen suorituskyvyn valvonta.

Työnä kehitetyn maksupalvelun vasteaikojen tavoitteena on pidetty noin yhtä sekuntia. Palvelun kehitys on kuitenkin vielä varsin alkuvaiheessa, eikä tätä olla vielä määritelty tarkemmin. Kun palvelu lähestyy tuotantokäyttöä ja palvelutasosopimusten tekoa palvelua käyttävien kauppiaiden kanssa, tullaan viimeistään silloin käsittelemään asiaa tarkemmin.

Palvelun skaalautuvuutta saadaan toteutettua esimerkiksi asynkronisilla HTTP-kyselyillä. Asynkronisissa kyselyissä tiettyä kyselyä käsittelevä säie ei jää varattuun tilaan odottamaan esimerkiksi API-kyselyn valmistumista, vaan vapautuu muiden kutsujen käsittelyä varten. Lopulta kun käynnistetty kysely on valmis, määrätään kyselylle uudelleen oma säie, joka käsittelee esimerkiksi kyselyn palauttamat tulokset. Kun kysely on käsitelty loppuun, palautetaan säie takaisin säiealtaaseen, josta se voidaan taas ottaa käyttöön tarvittaessa. Kuvassa 2.1 esitetään yksinkertaistettuna, miten asynkronisissa kutsuis-

sa otetaan säiealtaasta säie, joka heti kyselyn suorituksen jälkeen palautetaan altaaseen ja palautuu jälleen vapaaksi säikeeksi.



Kuva 2.1: Säikeiden toiminta asynkronisessa kyselyssä

2.2.3 Helppokäyttöisyys

Yksi olennaisimmista projektin ei-teknisistä vaatimuksista on sen helppokäyttöisyys sekä asiakkaalle että loppukäyttäjälle. Asiakkaalle, eli pääasiallisesti jonkin alan kauppiaalle helppokäyttöisyys näkyisi siten, että maksupalvelukomponenttia pitäisi pystyä käyttämään mahdollisimman yksinkertaisella tavalla. Työssä tämä haluttaisiin ensisijaisesti toteuttaa niin, että komponentin käyttäjälle, kauppiaalle tai myyjälle, riittäisi käyttöönottamisen yhteydessä vain oman pankkitilin numeron ja yhdistettävän yrityksen tietojen syöttäminen erillisen Ajas-järjestelmään helpon käyttöliittymän ja rajapinnan kautta. Tämän maksusivun lisäämisen jälkeen kauppiaan ei tarvitse enää tehdä asialle mitään.

2.2.4 Tietoturva

Edellä mainittujen laatuattribuuttien lisäksi myös tietoturva ja sen oikeanlainen toteutus ovat luonnollisesti hyvin oleellisia osia projektissa. Tämä johtuu siitä, että maksupalvelussa käsitellään hyvin arkaluonteisia, henkilökohtaisia tietoja, sekä käsitellään rahaa ja niiden siirtoja. Tietoturvan toteutukseen liittyvät muun muassa käyttäjän vahva todennus,

joka on myös osa PSD2-direktiivin vaatimuksia, sekä esimerkiksi Ajas Paymentsin palvelinpuolen API-kyselyjen todentaminen ja validointi. Myös erilaisilla palvelussa käytettävillä sertifikaateilla sekä tiedon ja yhteyden salauksilla saadaan tehtyä palvelusta turvallisempi käyttää.

2.3 Valitut ohjelmointikielet ja teknologiat

Työssä tullaan toteuttamaan Ajas Payments -maksupalvelulle sekä käyttöliittymä että palvelinpuoli, joka käsittelee kaikki maksuihin liittyvät tapahtumat. Sovelluksen palvelinpuoli on sen toiminnallisuuksien kannalta olennaisin sekä tämän työn suurimman mielenkiinnon ja tarkastelun kohteena.

Sovelluksen palvelinpuolen toteutuksessa käytetään ensisijaisesti Microsoftin kehittämää, C#-ohjelmointikielellä kirjoitettua ASP.NET Core -ohjelmistokehystä. ASP.NET Core on ilmainen avoimen lähdekoodin ohjelmistokehys, joka on tehty sekä Windows-, Linux- että macOS-käyttöjärjestelmille, joten se on myös järjestelmäriippumaton. Ohjelmistokehyksellä voidaan myös kehittää web-sovellusten lisäksi myös perinteisiä työpöytäsovelluksia Windows-käyttöjärjestelmälle, sekä komentorivisovelluksia ja ohjelmistokirjastoja. ASP.NET Core on laajalti käytetty kehys, tyypillisesti erityisesti suuremman kokoluokan ohjelmistoprojekteissa. [5]

Sovelluksen alkusuunnitteluissa päädyttiin nopeasti kahteen ohjelmointikielen vaihtoehtoon palvelinpuolen toteutusta varten, jotka olivat PHP ja C#. PHP-ohjelmointikieltä on käytetty yrityksessä tyypillisesti, minkä takia se myös tässä tapauksessa nousi hyväksi vaihtoehdoksi työn toteutusta varten.

2.3.1 C# ja ASP.NET Core -ohjelmistokehys

Projektin pääasialliseksi ohjelmointikieleksi päädyttiin lopulta valitsemaan C# ja tarkemmin ASP.NET Core -ohjelmistokehys, pitkälti niiden hyvän web-sovellusten kehityk-

sen tuen ja yleisen arkkitehtuurirakenteen vuoksi. Suorituskyvyltään C# ja PHP vaikuttivat keskimäärin melko tasavertaisilta, joten tällä ei ollut juuri vaikutusta päätökseen [6]. Kuitenkin mahdollisesti tärkein tekijä, joka vaikutti ohjelmointikielen ratkaisuun, oli ASP.NET Core -ohjelmistokehyksen tarjoama korkea turvallisuus, joka on luonnollisesti erityisesti tämän projektin kaltaisissa maksupalveluohjelmistoissa erityisen tärkeässä asemassa.

Valitussa ASP.NET Core-ohjelmistokehyksessä on tehty paljon valmiuksia tietoturvan toteuttamiseksi. Esimerkiksi käyttäjän todentaminen ja valtuuttaminen tiettyihin toimintoihin on tehty kehittäjälle varsin suoraviivaiseksi erilaisilla moduuleilla tai valmiilla komennoilla. Kehys tarjoaa myös helpot puitteet sovellussalaisuuksien turvalliseen säilyttämiseen sovelluksen kehitysvaiheessa, jolloin ohjelmakoodiin ei siis tarvitse kirjoittaa suoraan esimerkiksi tietokannan kirjautumistietoja tai API-avaimia. [7]

Erityisesti puhtaaseen PHP-koodiin verrattuna valittu ohjelmistokehys, ASP.NET Core, sisältää jonkin verran enemmän valmiiksi toteutettua turvallisuutta [8]. PHP:n ja C#:n yksi turvallisuuteen liittyvä esimerkkitapaus koskee ohjelmakoodin muokattavuutta palvelimella sovelluksen julkaisun jälkeen, jossa C#-koodia ei helposti pystytä muokkaamaan, kun taas PHP:n koodi on suoraan selkokielisenä muokattavissa palvelimella. Ohjelmakoodin muokattavuus palvelimen päässä saattaa aiheuttaa uhkia ohjelmiston turvallisuudelle, asettamalla sen enemmän alttiiksi palvelinpuolen hyökkäyksille. [9]

2.3.2 Konttiteknologia ja Docker

Sovelluksen asiakas- ja palvelinpuolten suorittamisessa käytetään myös Docker-teknologiaa, jossa ohjelmisto jaetaan useampaan eristettyyn osaan, niin sanottuun konttiin (engl. container), jotka sisältävät tyypillisesti yhden määritellyn osakokonaisuuden sovelluksesta. Tämän työn osalta sovellus jaetaan kolmeen pääasialliseen osaan: itse maksupalvelu, MySQL-tietokanta sekä Nginx-web-palvelin, jonka kautta tullaan tarjoamaan käyttäjän näkemä käyttöliittymä ja siihen liittyvät tiedostot ja ohjelmistokirjastot.

Dockeria päädyttiin tässä työssä käyttämään muun muassa sen tarjoaman skaalautuvuuden ja modulaarisuuden vuoksi. Dockerin kontteja käyttämällä saadaan maksupalvelukokonaisuus helposti jaettua useaan erilliseen, itsenäiseen osaan, joka helpottaa erityisesti sovelluksen jatkokehitystä ja testausta, sekä yleisesti sovelluksen myöhempiä päivityksiä.

Dockerin käyttämä konttiteknologia on ohjelmistotekniikassa käytetty tapa liittää sovellus tai palvelu, sekä kaikki siihen liittyvät riippuvuudet ja määrittelyt yhteen pakettiin, niin sanottuun levykuvaan (engl. image). Täten kontissa oleva sovellus voidaan suorittaa ja testata omana itsenäisenä yksikkönään, sekä ottaa suoraan käyttöön uudessa käyttöjärjestelmässä, tyypillisesti ilman erillisiä määrittelyjä tai riippuvuuksien asentamista. Tämän lisäksi kontit sovelluksineen ovat täysin eristyksissä muista konteista, mutta käyttävät yhteistä käyttöjärjestelmää. Muun muassa konttien kesken jaettu käyttöjärjestelmä tekee konteista tämän vuoksi tehokkaamman kuin virtuaalikoneiden käytöstä samaan tarkoitukseen. [10]

2.3.3 Tietokantarakenteet ja MySQL-relaatiotietokanta

Ajas Paymentsin tietojen tallennukseen käytettiin tietokantana MySQL-relaatiotietokantaa. MySQL valittiin projektissa tietokannaksi erityisesti sen todetun turvallisuuden ja suorituskyvyn, sekä suosion tuoman suuren tukiyhteisön takia. Tietokannan tyypiksi valittiin relaatiotietokanta, sillä sen arkkitehtuurirakenteen koettiin soveltuvan paremmin maksupalvelun kaltaisen ohjelmiston tietokantatarpeisiin kuin dokumenttipohjaisen tietokannan.

Relaatiotietokannan valinnan eräs suurimmista syistä on sen tapa käsitellä tietokantaan tallennettua dataa. Tässä hyvänä esimerkkinä maksupalvelun kaltaisen järjestelmän tietokantaan tallennettavista tietueista on tietyn verkkokaupan asiakkaan tiedot, kuten nimi, osoite, puhelinnumero ja sähköpostiosoite, jotka tallennettaisiin aina kun tietty henkilö tekee ostoksen kauppiaan verkkokaupassa.

Verrattuna dokumenttitietokantoihin (NoSQL) tämän kaltaisessa projektissa, tilanne ei vielä alussa olisi ongelmallinen, mutta tiedon lisääntyessä saman henkilön tehdessä uudelleen ostoksia loppuasiakkaan tiedot tallennettaisiin turhaan aina uudelleen, ellei duplikaattien välttämistä käsitellä jotenkin erikseen. Ajan kuluessa ja tietokannan koon kasvaessa duplikaatit voivat aiheuttaa pahimmillaan suuria määriä turhaa dataa, ja täten viedä turhaan järjestelmän resursseja ja heikentää yleistä suorituskykyä. Tästä voidaan nähdä, että dokumenttitietokannat ovat huomattavasti relaatiotietokantoja joustavampia muun muassa datan muodon suhteen. Kuitenkin joustavuus ja sääntöjen vähäisyys on tämän työn luonteen ja vaatimusten vuoksi jopa haitallista. [11]

Relaatiotietokannoissa sen sijaan ei pystytä luomaan täysiä duplikaatteja tallennettavasta datasta, sillä se kuuluu relaatiokantojen ideologiaan. Relaatiotietokannoissa tieto, kuten loppuasiakkaan henkilötiedot, voidaan tallentaa kerran tietokannan asiakastauluun, jonka jälkeen uusissa tilauksissa voidaan vain viitata jo luotuihin loppuasiakkaan tietoihin. Tämän avulla voidaan helpommin sekä säästyä turhilta duplikaateilta että pitää tieto varmemmin oikeana. [12]

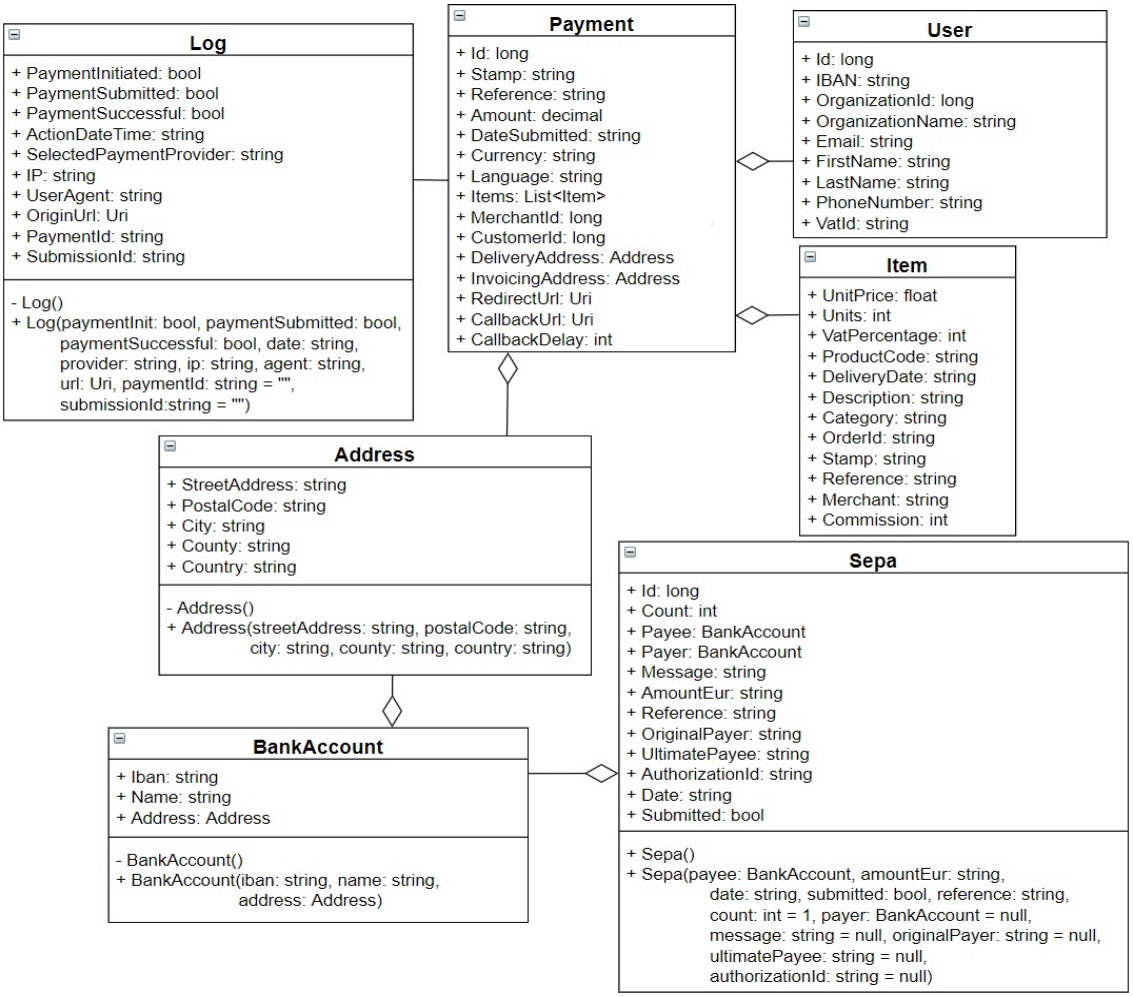
Tässä tehtyä eri taulun riveihin viittaamista varten relaatiotietokannoissa käytetään niin sanottuja pää- ja viiteavaimia. Pääavainta voidaan käyttää tietyn tietokannan tietueen yksilöivänä tunnisteena, kuten esimerkiksi tässä projektissa tietyn loppuasiakkaan yksilöllisellä tunnistekoodilla. Viiteavainta sen sijaan voidaan käyttää jossain toisessa tietokannan taulussa yhtenä sen attribuuteista. Samassa tätä projektia koskevassa esimerkissä voidaan ostostietojen taulussa viitata tiettyyn loppuasiakkaaseen viiteavaimella, jonka arvo on tässä tapauksessa sama kuin loppuasiakkaan pääavain. [13]

Näiden ominaisuuksien lisäksi relaatiotietokannan eri taulujen relaatioita voidaan tutkia myös toiseen suuntaan. Täten juuri mainitussa loppuasiakkaan ja ostostietojen esimerkissä voidaan myös helposti hakea kaikki tietyn loppuasiakkaan tekemät ostokset. Tätä ominaisuutta voidaan tarvittaessa käyttää projektin myöhemmissä kehityksen vaiheissa hyödyksi, muttei vielä alkuvaiheessa ole ollut kovin oleellinen yksityiskohta. Tätä

voidaan myöhemmin käyttää esimerkiksi analyyttisen datan keräämiseen eri loppuasiakkaiden ostoksista, ja tämän tiedon mahdolliseen hyötykäyttöön. [12]

Relaatiotietokanta ollaan näiden perusteella todettu olevan oikea valinta tälle maksupalvelulle. Nämä edellä mainitut edut ovat myös saavutettavissa dokumenttipohjaisella tietokantarakenteella, mutta vaatisivat kuitenkin enemmän työtä saman tuloksen saavuttamiseksi. Tämän lisäksi itse toteutettuna nämä tietokannan ominaisuudet olisivat myös todennäköisemmin virhealttiimpia, joka taas voi vaarantaa palvelun tietoturvan tasoa.

Kuvan 2.2 kaaviossa esitetään Ajas Paymentsin oleelliset luokat attribuutteineen, sekä näiden väliset suhteet.



Kuva 2.2: UML-diagrammi projektin oleellisista luokista ja näiden välisistä suhteista

Ohjelmiston luokkien välillä on siis jonkin verran riippuvuussuhteita, mutta ne käyttävät toisiaan lähinnä omien attribuuttiensa tarkan tyypin määrittelyssä. Esimerkiksi Sepa-luokan esittämissä SEPA-maksutiedoissa sekä maksaja- että maksunsaaja-attribuutit ovat tyyppiä BankAccount, jonka attribuutit tyyppeineen taas ovat määriteltynä omassa luokassaan. Tällä tarkalla attribuuttien tyypityksellä saadaan aikaiseksi yhtenäisempi ja turvallisempi kokonaisuus, sillä jokaisessa luokassa voidaan erikseen määrätä jokaisen attribuutin tarkka tyyppi, mitkä niistä ovat pakollisia luokan olioinstanssin luonnissa, sekä mahdollinen attribuuttiarvon formaatti, kuten merkkijonon pituuden. Tällöin luokkien pohjalta luodut olioinstanssien sisällöt tulevat samalla validoitua, eli tieto on oikeassa muodossa, joka puolestaan edistää koko palvelun vakautta.

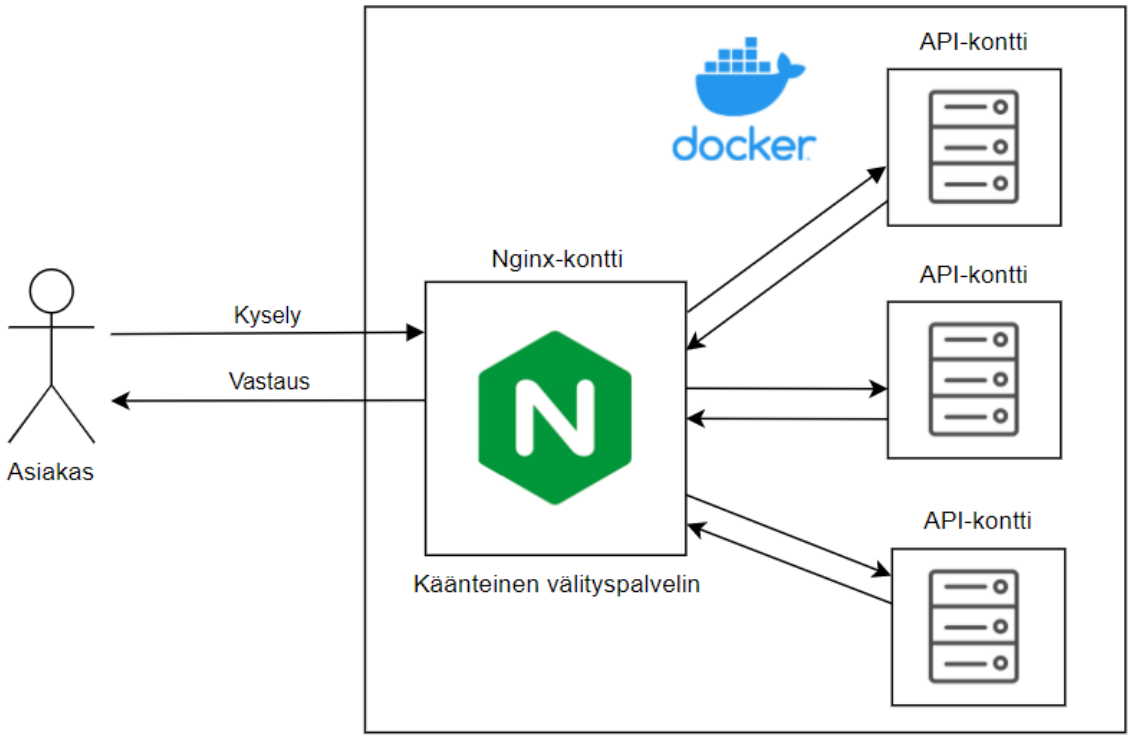
2.3.4 Nginx-web-palvelin

Ajas Paymentsin asiakaspuolen sovellus ja web-sivulla esitettävä käyttöliittymä suoritetaan Nginx-nimisellä web-palvelimella, joka muiden palvelun alisovellusten tavoin suoritetaan omassa Docker-kontissaan erillään muista sovelluksen kokonaisuuksista. Nginx on avoimen lähdekoodin web-palvelin, jota voidaan käyttää helposti myös käänteisenä välityspalvelimena (engl. reverse proxy), joka mahdollistaa myös muun muassa HTTP-välimuistin (engl. HTTP-cache) ja kuormituksen tasaajan (engl. load balancer) käytön.

Nginx oli melko helppo valinta sovelluksen asiakaspuolen suorittamiseen, sillä edellä mainittujen ominaisuuksien lisäksi sen etuja ovat muun muassa tyypillisesti vähäinen muistin käyttö ja hyvät valmiudet korkealle samanaikaisuudelle, sekä muun muassa näiden kahden hyödyn johdannaisena hyvä suorituskyky. Näiden lisäksi Nginx-web-palvelin on hyvin helppo käyttöönottaa muun muassa Docker-konteissa. Palvelimelle löytyy myös verkosta suuri tukiyhteisö, erityisesti Docker-konttien kanssa käytettynä, joka helpottaa mahdollisten ongelmatilanteiden ratkaisua.

Nginx käyttää myös asynkronista tapahtumapohjaista (engl. event-driven) arkkitehtuuria. Tämä tarkoittaa, että kaikki kutsut suoritetaan yhdessä säikeessä omissa proses-

seissaan niin, etteivät kutsut kuitenkaan tuki palvelinta, vaan myös muita kutsuja voidaan käsitellä samanaikaisesti. Muun muassa tällä ominaisuudella pystytään saamaan palvelulle korkeampaa saatavuusastetta, ja palvelun käyttö on tällöin vakaampaa. Kuvassa 2.3 esitetään yksinkertaistettuna Nginx-kontin toimintaa käänteisenä välityspalvelimena.



Kuva 2.3: Nginx-kontin käyttö käänteisenä välityspalvelimena Docker-ympäristössä

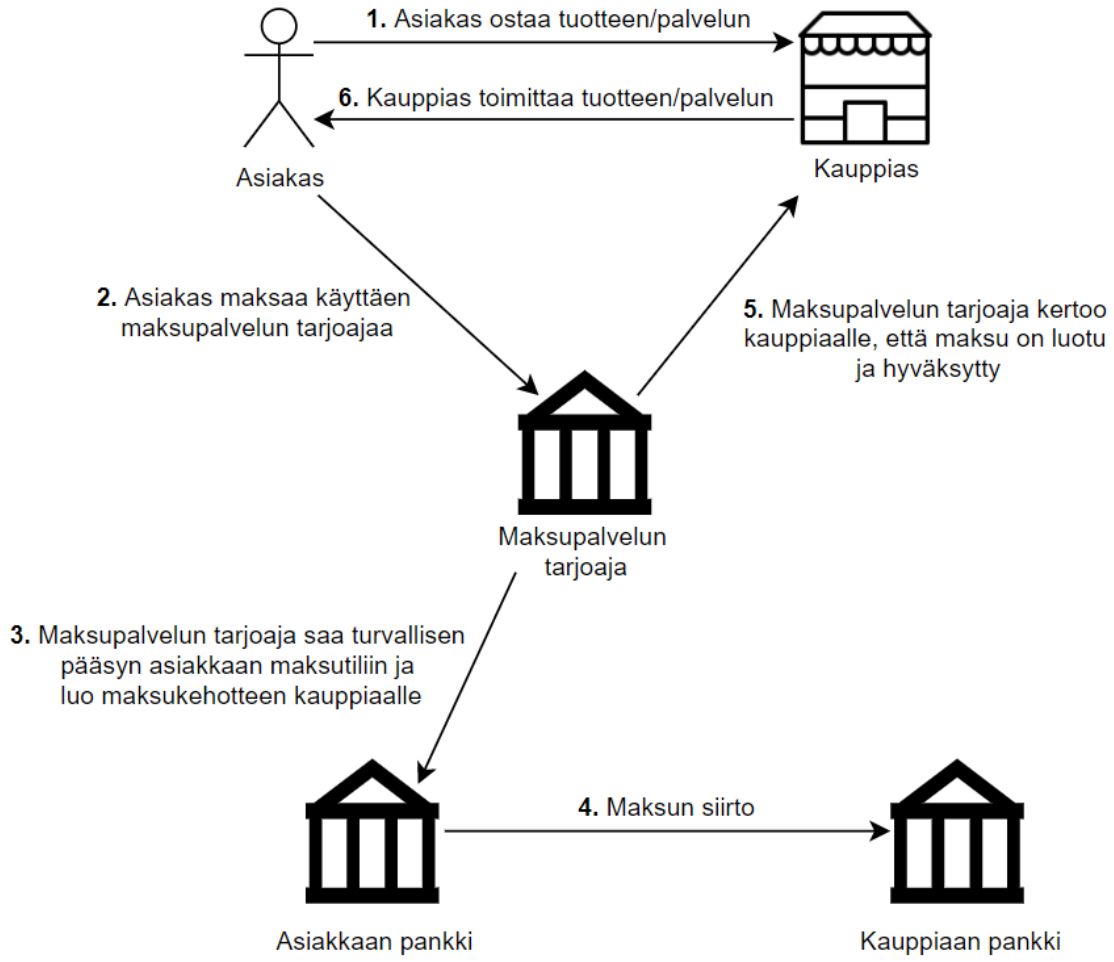
Tässä projektissa on vielä toistaiseksi käytetty Nginx-palvelinta ainoastaan yksinkertaisena web-palvelimena, jotta palvelun asiakaspuolen sovellus pystytään suorittamaan. Kuitenkin myöhemmässä työn kehitysvaiheessa on tarkoitus käyttää sitä myös käänteisenä välityspalvelimena, joka mahdollistaa muun muassa HTTP-välimuistin käytön esimerkiksi maksupalvelun tarjoajien API-kyselyissä. Näiden lisäksi tarkoituksena on hyödyntää palvelimen tarjoamaa kuormituksen tasauksen käyttöä, jotta palvelimien resurssit pystytään optimoimaan parhaalla mahdollisella tavalla suurien käyttäjämäärien aikaan, ja tällä tavoin parantamaan palvelun skaalautuvuutta sekä saatavuutta.

3 Web-pohjaiset maksupalvelut ja maksupalvelun tarjoajat

Maksupalvelut tarkoittavat sellaisia palveluntarjoajia, jotka vastaanottavat asiakkaidensa toimeksiantoja, jotka koskevat varojen siirtämistä, pääasiallisesti erilaisia verkkopankkimaksuja ja tilisiirtoja. Maksupalvelujen käyttötarkoituksia ovat siis muun muassa kaikenlaiset maksutilien väliset tilisiirrot sekä ostokset maksukorteilla. Maksupalveluihin liittyy luonnollisesti tarkat lainsäädännöt eri maissa, ja esimerkiksi Suomessa maksupalveluja ohjaavat erilaiset niihin liittyvät lait, kuten maksupalvelulaki sekä maksulaitoslaki.

Erona maksupalvelun ja maksupalvelun tarjoajan välillä on se, että maksupalveluja käyttävät esimerkiksi pankin asiakkaat, ja näihin palveluihin kuuluvat muun muassa edellä mainitut maksutilien väliset tilisiirrot ja ostokset maksukorteilla. Maksunpalvelun tarjoajat puolestaan ovat palveluntarjoajia, jotka toimivat maksajan ja maksunsaaja välissä, siirtäen varoja osapuolten välillä saatujen toimeksiantojen perusteella. [14] Esimerkkinä maksupalveluiden tarjoajasta on myös tässä diplomityön käytännön toteutuksessa käytetty Checkout Finland.

Kuvassa 3.1 on kuvaaja, jolla esitetään pääpiirteittäin vaiheet, kun loppuasiakas ostaa verkkokaupasta ja maksaa kauppiaalle jotain maksupalvelun tarjoajaa käyttäen.



Kuva 3.1: Maksupalvelun tarjoajan rooli loppuasiakkaan ja kauppiaan välisessä maksamisessa

3.1 Hyödyt kauppiaille ja loppuasiakkaille

Maksupalvelun tarjoajat tuovat esimerkiksi verkkokaupan ylläpitäjille ja näiden loppuasiakkaille sähköiset mahdollisuudet haluttujen maksupalvelun käyttöön ja maksujen tekoon verkon välityksellä. Tällä tavoin pystytään sallimaan jonkin alan kauppiaalle myytävien tuotteiden ja palveluiden helpompi myynti ja rahaliikenteen hallinta verkossa.

Maksupalvelun tarjoaja tyypillisesti tarjoaa käyttäjälle mahdollisuuden useisiin eri maksutapavaihtoehtoihin, kuten eri pankkien verkkomaksuihin, luottokorttimaksuihin tai

mobiilimaksuihin. Tämän tarkoituksena on luonnollisesti pyrkiä saamaan asiakkaalle, eli loppukäyttäjälle, mahdollisimman laajan valikoiman eri vaihtoehtoja maksutavoille, jotta käyttö olisi mahdollisimman helppoa.

Maksupalveluita käyttämällä voidaan taata, että verkossa tehdyt maksut kauppiaille tapahtuvat turvallisesti, ja että maksutapahtumasta jää aina jälki. Jokaisesta maksutapahtumasta jäävä jälki toteuttaa kyseiselle palvelulle myös kiistämättömyyden, joka on yksi tärkeä osa yleistä ohjelmistojen tietoturvaa, ja on erityisen tärkeä maksupalveluiden kaltaisissa rahaa käsittelevissä järjestelmissä. Järjestelmään jääviä jälkiä voidaan muun muassa käyttää ongelmatilanteiden ratkaisemisessa tai mahdollisten petosyritysten tapahtuessa. Esimerkkinä tästä voisi olla tilanne, jossa maksupalvelua käyttävä kauppias väittää, ettei hän ole saanut loppuasiakkaalta maksua tuotteistaan tai palveluistaan, tai loppuasiakas väittää maksaneensa, vaikkei olisikaan suorittanut maksua.

Maksupalvelun tarjoajia käyttämällä pystytään myös helpottamaan huomattavasti loppuasiakkaan käyttämän sovelluksen käyttöä, sillä esimerkiksi perinteisempään suoraan tilisiirtoon verrattuna loppuasiakkaan ei tarvitse huolehtia maksun summan, kauppiaan tilinumeron, nimen tai viitenumeron kirjoittamisesta.

3.2 Palveluntarjoajien toimiluvat

Maksupalveluja saavat tarjota Suomessa ainoastaan ne toimijat, joille Finanssivalvonta on myöntänyt toimiluvan, joten toimiala on luonnollisesti hyvin säädeltyä ja valvottua. Vaadittavan toimiluvan puolestaan voivat saada ainoastaan ne palveluntarjoajat, jotka täyttävät tietyt maksulaitoslaissa säädellyt edellytykset. [14]

Toimiluvan saannin edellytyksiä Suomessa ovat esimerkiksi palveluntarjoajan todettu luotettavuus, maksulaitoksen toiminnalle ja taloudelliselle asemalle säädettyjen vaatimusten täyttäminen, sekä palveluntarjoajan kotimaisuus. Palveluntarjoajan luotettavuuden toteamisessa tarkistetaan kyseisen palveluntarjoajan tai maksulaitoksen perustajan,

sekä mahdollisten vähintään kymmenen prosenttia osakkeista omistavien osakkaiden luotettavuus. Luotettavana ei esimerkiksi voida pitää henkilöitä, jotka ovat tulleet tuomituiksi vankeusrangaistukseen viiden edellisen vuoden ajalta, sakkorangaistukseen rikoksesta kolmen edellisen vuoden ajalta, tai jotenkin muuten osoittanut aiemmalla toiminnallaan olevansa sopimaton tähän rooliin. [15]

4 Maksupalvelulait

Maksupalvelulait koskevat kaikenlaisia sähköisen maksamisen eri tapoja, joita ovat: tilisiirrot, suoraveloitukset, korttimaksut, sekä mobiili- ja nettipankkimaksut. Maksupalvelut ja niiden tarjoajien toiminta on varsin tiukkaan säädeltyä erilaisilla maksupalveluja koskevilla lakipykälillä, ylimpänä Euroopan unionin tasolta.

Maksupalvelulakien (ensimmäinen ja toinen maksupalveludirektiivi) yhtenä tarkoituksena on saada Euroopan unionin alueelle yhtenäiset maksupalveluja säätelevät lait, jotta EU:n sisällä tapahtuvat maksut ja niitä hoitavat maksupalvelut pystyisivät toimimaan samojen standardien perusteella. Näillä lakipykälillä on tarkoitus saada aikaan selkeät, yhtenäiset maksutiedot, nopeat maksujen suoritukset, parempi kuluttajansuoja ja laajemmat valinnan mahdollisuudet maksupalveluille. Euroopan unioni pyrkii täten luomaan yhden standardoidun maksualueen, jossa esimerkiksi eri EU-maiden välillä tapahtuvat maksut voidaan tehdä aivan yhtä helposti ja turvallisesti, sekä samaan hintaan, kuin kotimaan sisäiset maksutkin. [16]

4.1 SEPA-maksut

Ensimmäinen maksupalveludirektiivi toimi alkuna yhtenäisen euromaksualueen synnylle (engl. Single Euro Payments Area, SEPA) luomalla sille hyvän juridisen pohjan. SEPA-alueen pääasiallisena tarkoituksena on helpottaa Euroopan unionin sisällä eri maiden välillä tapahtuvien sähköisten maksujen lähetystä ja vastaanottoa, jotta maiden väliset maksut ovat yhtä helppoja ja turvallisia kuin maksut kotimaan sisällä.

SEPA-maksuihin kuuluvat tilisiirrot, suoraveloitukset ja pankkikorttimaksut, jotka ovat olleet tyypillisimpiä sähköisen maksamisen tapoja. [17] SEPA-maksuja voidaan tehdä siis ainoastaan SEPA-alueeseen kuuluvien maiden välillä ja ainoastaan euromääräisinä. SEPA-maksun vaaditut tiedot ovat melko minimaaliset; maksun suoritus vaatii ainoastaan maksajan ja maksunsaajan kansainväliset IBAN-muotoiset tilinumerot, maksunsaajan pankin yksilöivän BIC-koodin, sekä maksettavan summan euroina. [18]

4.2 Ensimmäinen maksupalveludirektiivi (PSD)

Euroopan unioni laati alun perin ensimmäisen maksupalveludirektiivin (PSD, engl. Payment Service Directive) alkamaan vuonna 2007, ja tulemaan voimaan joko täysin tai osittain EU:n jäsenvaltioiden lakisäädöksissä vuonna 2009. Laaditun direktiivin tarkoituksena oli yhtenäistää EU-maiden keskenään varsin eriävät lait ja säädökset, jotta EU:n sisällä maksupalveluita koskevat lait olisivat samat kaikissa jäsenmaissa. Yhtenäisen maksupalvelujen lakisäädösten alueen tarkoituksena oli edesauttaa lakien selkeyttä ja erityisesti maksupalvelujen markkinoiden tehokkuutta. Toinen tavoite tällä direktiivillä oli luoda lisää kilpailua eri maksunpalvelun tarjoajien välille EU:n alueella, ja samalla antaa kuluttajille suuremman valinnanvapauden. [19], [20] Kilpailua lisättiin sillä, että nyt muutkin toimijat kuin ainoastaan pankit saivat tarjota maksupalveluja asiakkailleen. [21]

4.2.1 Maksupalveluiden valtuutus

Direktiivin yhtenä tärkeimmistä vaatimuksista oli, että jokaisen maksupalveluja tarjoavan yrityksen tulee olla paikallisen valtion säätelijän valtuuttama. Valtuutusta hakevien yritysten tuli direktiivin mukaan olla muun muassa seuraavat: vahvat hallintovalmiudet maksupalveluiden tarjoamiselle, tarpeeksi pääomaa ja todisteet siitä sekä kattava liiketoimintasuunnitelma. [19] Esimerkiksi Suomessa näistä maksupalvelujen valtuutuksista vastaa Finanssivalvonta.

4.2.2 Tarve uudistukselle ja aiemmat rajoitteet

Ensimmäinen maksupalveludirektiivi oli tullut voimaan jo vuonna 2007 ja täytäntöönpantavaksi ainakin osittain kaikissa EU:n jäsenvaltioissa vuonna 2009. Tästä huolimatta vielä 2010-luvun alussa ensimmäinen maksupalveludirektiivi ei kuitenkaan ollut saanut kovinkaan suurta liikettä aikaiseksi, sillä vielä vuoden 2012 loppupuoliskolla EU:n alueella oli vain 568 valtuutettua maksuinstituutiota (engl. authorized payment institution, API). Tämän lisäksi direktiivin tuomista säädöksistä huolimatta eri maksupalvelun tarjoajien rakenteiden välille oli edelleen jäänyt suuria keskinäisiä eroavaisuuksia, tyypillisesti eri EU:n toimivaltojen välillä. Myöskään direktiivin tavoitteena ollut maksupalvelun tarjoajien välinen kilpailu ei ollut nähtävästi juurikaan kasvanut, sillä valtaosa vuoden 2012 valtuutetuista maksuinstituutioista oli perustettu jo ennen direktiivin voimaantuloa, eli uusia yrittäjiä ei ollut juurikaan tullut alalle. [22]

4.3 Toinen maksupalveludirektiivi (PSD2)

Toinen maksupalveludirektiivi (PSD2) on Euroopan parlamentin ja neuvoston laatima direktiivi, joka koskee sähköisiä maksupalveluja ja direktiiviä noudattavia pankkeja. Direktiivi laadittiin tarkoituksena parantaa ja uudistaa aiempaa maksupalveludirektiiviä, eli pääkohteena parantaa EU-maiden sisäisiä markkinoita maksupalveluille. Direktiivin tavoitteena on saada yhä useammat maksupalvelut sääntelyn piiriin, sekä tehdä näiden maksupalvelujen sääntelystä paremmin nykyisiä markkinoita vastaavaa. [23] Säännös astui voimaan vuoden 2016 alussa, ja tuli kaikille EU-maille pakolliseksi osaksi jokaisen maan lakia vuoden 2018 alussa.

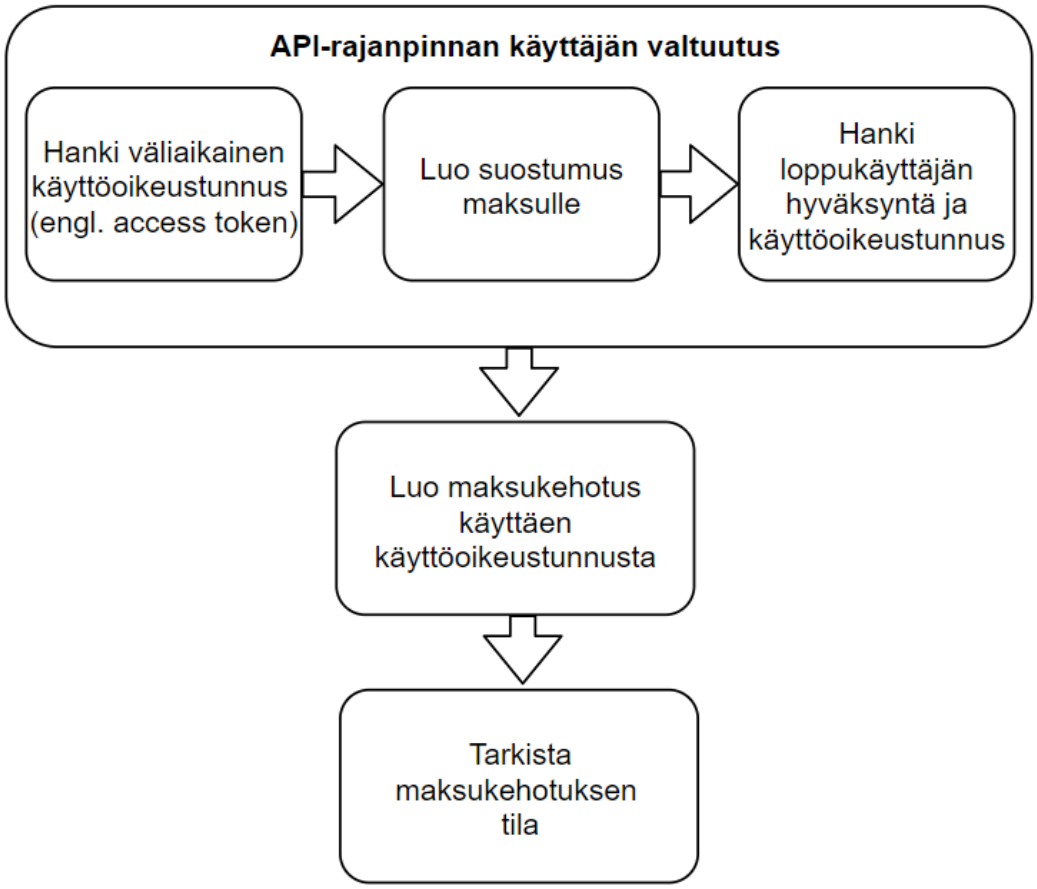
Toinen maksupalveludirektiivi luo lailliset puitteet kaksille uudentyyppisille maksupalveluille, jotka ovat maksutoimeksiantopalvelun tarjoajat sekä tilitietopalvelun tarjoajat. [24] Jotta pankki saisi tarjota näille kolmansille osapuolille pääsyoikeudet loppuasiakkaiden tileille, tulee loppuasiakkaan ensin suostua tietojen jakamiseen. Tämän lisäksi

maksutoimeksiantopalvelun ja tilitietopalvelun tarjoajat pystyvät käyttämään hyväkseen pankin loppuasiakkaalle tarjoamia vahvan tunnistautumisen menettelyjä. [23]

Maksupalveluita tarjoavien yritysten valtuutus ei muuttunut kovinkaan merkittävästi aiemmasta maksupalveludirektiivistä, eli yritysten tulee edelleen tämän direktiivin puitteissa saada valtuutus paikalliselta valtion säätelijältä. Kuitenkin uuden direktiivin mukaan maksutoimeksiantopalveluita tai tilitietopalveluita tarjoavilla yrityksillä tulee nyt myös olla ammatillinen vastuuvakuutus. Uutena direktiivi tuo myös säännöksiä valtuutettujen yritysten valvontaan, sekä menettelytapoja vaatimusten mahdolliseen vastaiseen toimintaan. [25]

Tässä ohjelmistoprojektissa tullaan hyödyntämään erityisesti kyseistä direktiiviä maksujen siirroissa sekä loppuasiakkaalta Enerocille, että lopulta Enerocilta kauppiaalle. Direktiiviä hyödyntämällä tarkoituksena on saavuttaa uusien pankkiyhteyksien helpompi luominen, täten helpottaen lopulta sekä ohjelmiston kehittäjiä, että palvelua käyttäviä kauppiaita.

Kuvassa 4.1 olevassa kuvaajassa esitetään tärkeimmät vaiheet maksukehotuksen luomisesta ja suorittamisesta käyttäen PSD2-direktiiviä noudattavaa maksupalvelun tarjoajan rajapintoja. Tämä askelten malli on PSD2-direktiivissä standardina, eli esimerkiksi jokaisen pankin tulee noudattaa tätä.



Kuva 4.1: Pankkien tarjoamien PSD2-rajapintojen käyttöönoton askeleet

4.3.1 Uudet mahdollisuudet

Ensimmäisen maksupalveludirektiivin tuomien säännösten parannusten lisäksi PSD2-direktiivi avaa EU-maiden markkinoita myös kokonaan uusille palveluille ja palveluntarjoajille. Uusi direktiivi pyrkii tällä tavoin lisäämään maksupalveluiden alalla sekä kilpailua, että kuluttajille valinnanvaraa.

Uudet mahdolliset palvelut ja niiden tarjoajat ovat maksutoimeksiantopalvelun tarjoajat ja tilitietopalvelun tarjoajat. [25] Direktiivissä toinen hyvin tärkeä uudistus on, että maksutiliä hallinnoivat maksupalveluntarjoajat, eli pankit, velvoitetaan antamaan tilitie-

topalveluiden ja maksutoimeksiantopalveluiden tarjoajille pääsyoikeudet käyttäjän verkkomaksutileille ilmaiseksi. [24]

Tilitietopalveluita pystytään tämän myötä käyttämään maksupalveluissa niin, että maksupalvelun käyttäjä pystyy saamaan omat tilitietonsa, kuten tiliensä saldot, maksupalvelun kautta milloin tahansa. Käyttäjän ei siis tällöin tarvitse tarkistaa tilitietojaan erikseen suoraan pankkinsa kautta, jonka tarkoituksena on helpottaa käyttäjän omien talousasioiden hallintaa. Maksutoimeksiantopalveluita taas voidaan käyttää luomaan uusia maksutoimeksiantoja maksupalvelun käyttäjän pyynnöstä toisille maksutileille. [25]

Verkkomaksutilien pääsyoikeuksien jako ei ollut aiemmin maksupalvelulakien vaatimuksena, eikä tästä johtuen ollut myöskään kovin yleistä, että pankit antoivat muille palveluntarjoajille pankin asiakkaiden tilitietoihin. Muun muassa tämän koettiin hidastavan esimerkiksi uusien maksupalveluiden ja -tarjoajien syntyä ja palveluiden välistä kilpailua. Tilitietojen mahdollinen jakaminen ei myöskään ollut minkään yleisen standardin mukaista, eli esimerkiksi eri pankkien tarjoamien ohjelmointirajapintojen (eng. application programming interface, API) välillä oli selkeitä eroja. Tämä puolestaan teki uusien pankkiyhteyksien luomisesta aina työläämpää, sillä jokaiselle pankille tuli tehdä omat integraatiot erikseen.

4.3.2 Vahvan tunnistautumisen asetus

Asiakkaan vahva tunnistautuminen (engl. Strong Customer Authentication, SCA) on yksi hyvin olennainen osa toista maksupalveludirektiiviä ja sen toteuttamista. Vahvan tunnistautumisen asetus tuli uudeksi osaksi PSD2-direktiivin sääntelyjä vasta myöhemmässä direktiivin voimassaolon vaiheessa, vuoden 2019 syyskuussa. Vaatimuksia ei vielä ole määrätty valvottavaksi kaikissa EU-alueen maissa, mutta tämä tapahtuu lopuissakin maissa vielä vuoden 2020 aikana. [26]

Tämä on uusi sääntelyvaatimus (engl. European regulatory requirement), jonka tarkoituksena on vähentää petosten mahdollisuutta, ja täten tehdä verkkomaksuista ja maksu-

palveluiden käytöstä entistäkin turvallisempaa. Vahva tunnistautuminen -asetuksen noudattamisen olennaisena ehtona on käyttää loppuasiakkaan tunnistamiseen vähintään kahta eri tunnistautumistapaa kolmesta eri mahdollisesta vaihtoehdosta, ja vasta niillä onnistuneen tunnistautumisen jälkeen antaa loppuasiakkaan esimerkiksi suorittaa maksuja. [22], [27]

Vahvaa tunnistautumista ei vielä alkuun olla toteutettu Ajas Payments -tuotteessa, sillä projekti ei ole vielä tuotannossa käytössä, vaan vielä kehitysvaiheessa. Vahva tunnistautuminen tulee kuitenkin olemaan pakollinen osa maksupalveluja viimeistään vuoden 2020 loppuun mennessä kaikissa EU-alueen jäsenmaissa, joten luonnollisesti vaatimus tullaan täyttämään vielä. Sääntelyvaatimuksessa on oleellista myös se, että se koskee ainoastaan loppuasiakkaan toimeksi panemia maksuja. Tämä tarkoittaa, että vaatimus pätee esimerkiksi loppuasiakkaan ostaessa tuotteita verkkokaupasta, mutta ei esimerkiksi silloin, kun jokin yritys siirtää varoja toiselle yritykselle. Projektin alkuvaiheessa käytetään vielä Checkout Finland -maksupalvelua varojen siirtoon loppuasiakkaalta Enerocille, joten vahvaa tunnistautumista ei vielä ole tarpeen kehittää.

Kolme vaihtoehtoa tunnistautumiseen

Vahvan tunnistautumisen sääntelyvaatimus vaatii, että maksupalveluihin tullaan jatkossa rakentamaan kaksiosainen tunnistautuminen. Kaksiosaisessa tunnistautumisessa käyttäjä pystyy itse valitsemaan haluamansa kaksi tunnistautumistapaa kolmesta alla olevasta eri vaihtoehdosta. [27]

- Jokin minkä käyttäjä tietää, esimerkiksi salasana tai pin-koodi
- Jotain mitä käyttäjä omistaa, esimerkiksi puhelin tai muut laitteet
- Jotain mitä käyttäjä on, kuten käyttäjän sormenjälki tai kasvotunnistus

3D Secure 2 -standardi

3D Secure 2 -standardi on yksi tapa toteuttaa vahvan tunnistautumisen asetuksen vaatimukset Euroopassa, ja täten tehdä maksuista samalla turvallisempia sekä helppokäyttöisempiä. Standardin odotetaan nousevan pääasialliseksi tavaksi täyttää tunnistautumisasetuksen vaatimukset.

Tyypillinen tapa 3DS2-standardissa on käyttää hyväksi älypuhelimia osana tunnistautumisprosessia maksujen yhteydessä. Aiemmin puhelimeen on tavallisesti saatu jokin salasana tai koodi, jolla vahvistetaan identiteetti. Kuitenkin älypuhelinten kehityksen myötä on voitu alkaa myös käyttämään ihmisen ominaisuuksia tunnistautumiseen, kuten sormenjälkeä tai kasvojentunnistusta. Tällä tavoin maksujen teko helpottuu, ja tämän standardin toteutuksen odotetaankin koko ajan yleistyvän esimerkiksi pankeissa. [28]

4.3.3 Vaaditut eIDAS-sertifikaatit

Kaikessa maksupalveluihin liittyvässä verkon yli tapahtuvassa kommunikaatiossa tulee olla erityisen varovainen, ja turvallisuuteen tulee kiinnittää suurta huomiota. Syinä tälle on luonnollisesti henkilöiden arkaluontoiset tiedot sekä rahan siirrot.

Loppuasiakkaan vahvan tunnistautumisen lisäksi PSD2-direktiivissä vaaditaan, että kaikki rajapinnat käyttävät niin sanottuja eIDAS-sertifikaatteja. eIDAS (electronic IDentification, Authentication and trust Services) on Euroopan unionin säännös, ja eIDAS-sertifikaatteja käytetään Euroopan unionin sisämarkkinoilla elektronisten valuuttasiirtojen elektroniseen tunnistamiseen sekä luottamuspalvelujen valvontaan. [29]

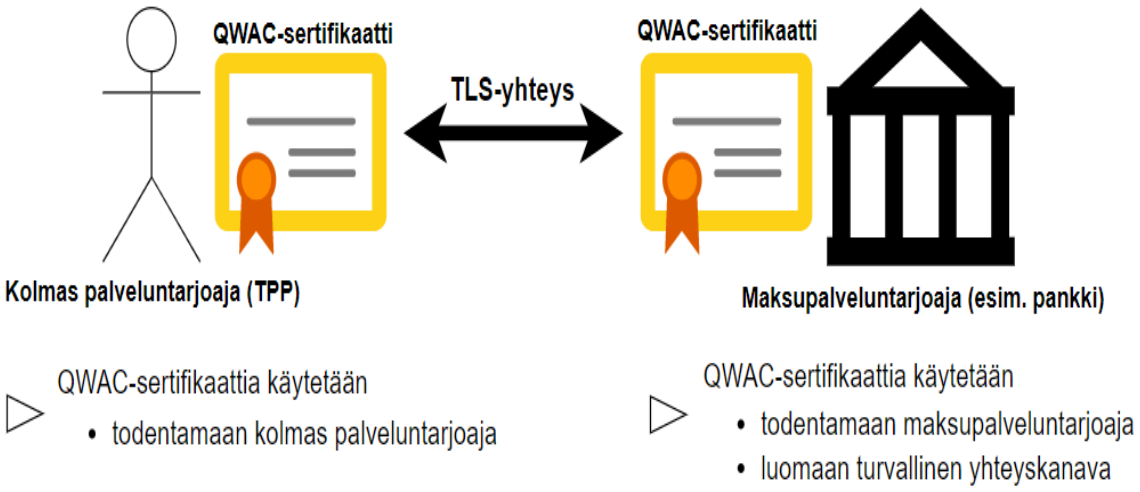
Näitä sertifikaatteja ei kuitenkaan vielä tarvittu projektin prototyypin kehitysvaiheessa. Tämä johtui siitä, että Osuuspankin huomattiin tarjoavan mahdollisuuden Osuuspankin verkkopalvelujen rajapintojen käyttöön testiympäristöissä ilman Finanssivalvonnan myöntämiä virallisia sertifikaatteja.

Sertifikaattityypit QWAC ja QCSEAL

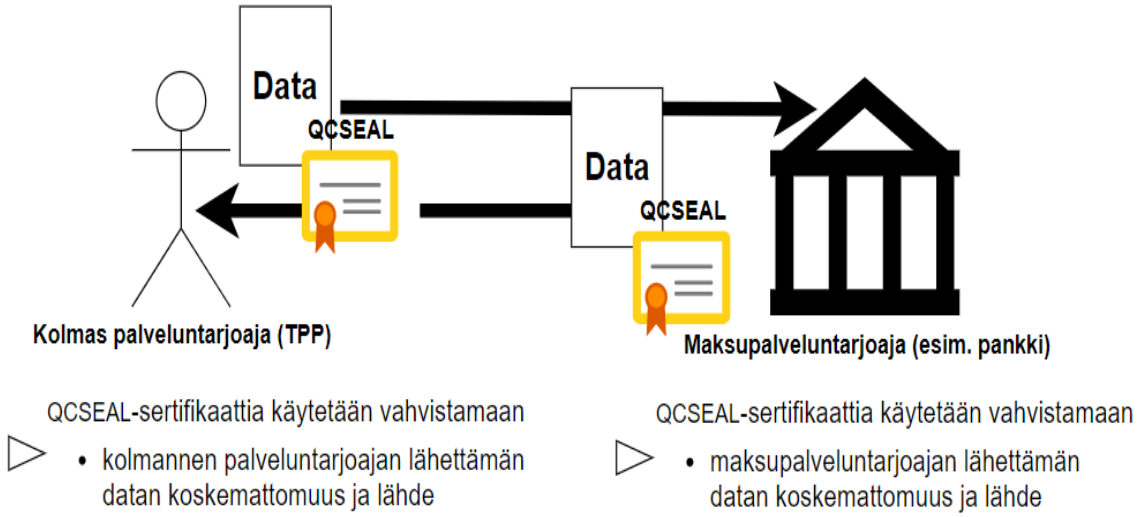
eIDAS-sertifikaatteja on kahta eri tyyppiä: QWAC (Qualified Website Certificate) ja QC-SEAL (Qualified Certificate for Electronic Seal).

Näissä sertifikaateissa tulee säännösten mukaisesti olla seuraavat tiedot ja ominaisuudet: kolmannen palveluntarjoajan valtuutusnumero, päätösvaltaisen valtuuttajan nimi, johon palveluntarjoaja on rekisteröity, sertifikaatin omistajan nimi, pätevät luottamuspalvelun tarjoajan nimi (engl. Qualified Trust Service Provider, QTSP), voimassaoloaika eli alkamis- ja päättymispäivämäärät, sekä palveluntarjoajan rooli. Palveluntarjoajan rooli on oltava yksi seuraavista: tilipalvelu, maksutoimeksiantopalvelu, tilitietopalvelu tai korttipohjaisten maksuvälineiden liikkeeseenlaskupalvelu. [23], [29]

Kaavioissa 4.2 ja 4.3 avataan näiden kahden eri sertifikaatin toimintaa ja käyttöä kolmansien palveluntarjoajien ja maksupalveluntarjoajien välillä.



Kuva 4.2: QWAC-sertifikaatin käyttö ja tarkoitukset



Kuva 4.3: QCSEAL-sertifikaatin käyttö ja tarkoitukset

Toisen maksupalveludirektiivin vaatimien QWAC- ja QCSEAL-sertifikaattien tarkoituksena on taata molemmille keskenään kommunikoiville osapuolille luottamuksen toisen osapuolen oikeaan identiteettiin, mahdollistaa turvallisen väylän osapuolten kommunikaatiolle, sekä tarjota laillisesti pätevät todisteet tapahtuvista rahansiirroista.

Sertifikaattien välillä ei ole erityisen suuria eroja, mutta ne ovat silti erilaiseen käyttöön tarkoitettuja. QWAC-sertifikaattia voidaan käyttää molemminpuolisen tunnistautumisen ja todennuksen suorittamiseen yhteyden luonnin aikana. QCSEAL-sertifikaatti sen sijaan takaa osapuolten yhteyden ja datan yhtenäisyyden ja koskemattomuuden, sekä todisteen yhteyden lähteestä. Molempia sertifikaatteja käyttäessä tulee kommunikoinnin tapahtua salatun TLS-yhteyden välityksellä. [29]

5 Työn suunnitelma ja implementaatio

Ajas Payments -maksupalvelukomponentti on kaksiosainen, ja se koostuu asiakas- ja hallintapuolesta. Tämän diplomityön osalta on keskitytty ainoastaan asiakaspuolen toteutukseen, eli yksinkertaistettuna loppuasiakkaan valitseman maksupalvelun avulla maksujen suorittaminen, tietojen kirjaamiseen ja maksuosuuksien vientiin lopulta kauppiaalle.

Ajas Paymentsin asiakaspuoleen kuuluvat kaikki näkymät ja toiminnot, joita kauppiaalta ostava loppuasiakas käyttää ja näkee. Toisin sanoen loppuasiakas valitsee haluamansa maksutavan yksinkertaiselta sivulta, jonka jälkeen loppuasiakas ohjataan suorittamaan maksun valitun maksupalveluntarjoajan sivuilla. Kun maksu on suoritettu ja onnistunut, ohjataan loppuasiakas vielä Paymentsin kiitos-sivulle, josta loppuasiakas jatkaa eteenpäin pois maksupalvelun sivulta.

Komponentin hallintapuoli, eli esimerkiksi verkkokauppaa ylläpitävän kauppiaan näkymät ja toiminnot sijaitsevat erillisenä Ajas Touch -nimisessä ajanvarausten hallintajärjestelmässä. Kauppiaalle on pyritty tekemään mahdollisimman helpoksi ottaa Ajas Payments -komponentti käyttöön kauppiaan verkkokaupassa tai ajanvarausjärjestelmässä. Tarkoituksena on, että kauppiaan tulee vain täyttää pankkitilinsä numeron ja yrityksensä perustiedot Ajas Touchissa. Tämän jälkeen maksusivu voidaan lisätä kauppiaan sivuille helpon rajapinnan avulla.

Tämän diplomityön osalta palvelua on kehitetty vielä ainoastaan paikallisessa kehitysympäristössä käyttäen Docker-kontteja. Dockeria on käytetty palvelun eri moduulien suorittamiseen muun muassa sen vuoksi, että tällä tavoin saadaan samat tulokset ja suo-

ritusympäristö, kuin jollain ulkoisella palvelimella suoritettuna. Tällä tavoin saadaan vähennettyä huomattavasti esimerkiksi työvaihe- tai tuotantopalvelimella olevan palvelun testausta, ja täten keskittyä enemmän itse palvelun kehitystyöhön. Ajansäästön lisäksi tämän ratkaisun tarkoituksena on säästää rahaa ja työvoimaresursseja.

5.1 Tietoturva ja sen toteutus

Tässä diplomityössä kehitettävän maksupalvelun kaltaisissa ohjelmistoissa on erittäin tärkeää ottaa tietoturva tarkasti huomioon sen käsittelemien hyvin arkaluontoisten tietojen vuoksi. Turvallisuus on tässä työssä pyritty pitämään mielessä heti palvelun kehityksen alkuvaiheesta lähtien, eli seuraamaan niin sanottua Secure by Design -periaatetta. Tämän kehitysperiaatteen etuja ovat esimerkiksi kehityskulujen väheneminen, teknisen velan minimointi ja yleisesti riskien väheneminen [30]. Tämän lisäksi on ollut tärkeää, että tietoturva ja yhtenäisyys pidetään mielessä jokaisella ohjelmiston osa-alueella, kaikissa toiminnoissa.

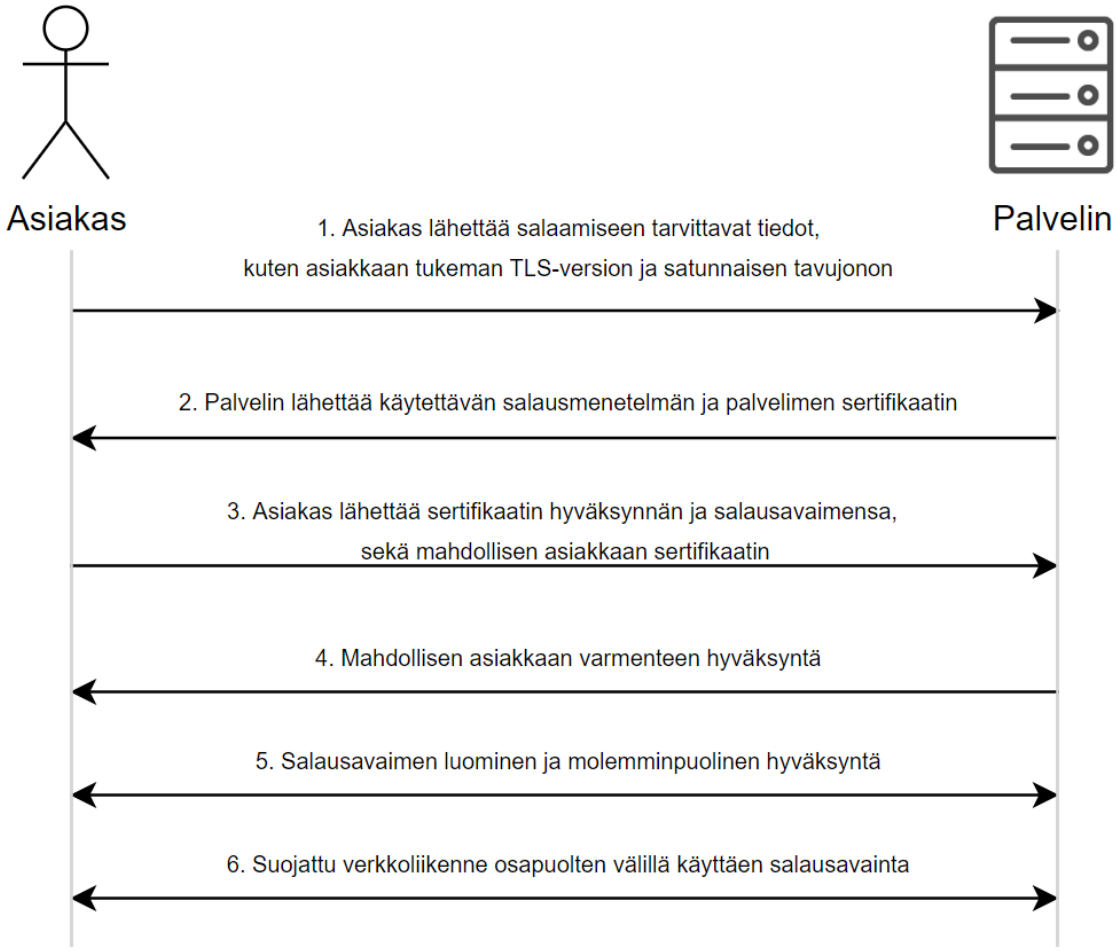
5.1.1 HTTPS- ja TLS-protokollat

HTTPS-protokolla on pitkälti sama kuin HTTP-protokolla (engl. Hypertext Transfer Protocol), mutta tässä on otettu turvallisuus ja yhteyden salaus paremmin huomioon. HTTPS-protokollassa käytetään TLS-salausprotokollaa (engl. Transport Layer Security) suojaamaan HTTP-kutsuissa tapahtuvaa liikennettä salaamalla osapuolten välillä liikkuva data useilla eri keinoilla. Tähän kuuluvat esimerkiksi palvelin- ja asiakaspuolen sertifikaatit ja salausavaimet, ja niiden käyttö yhteisen salaus- ja salauksen poisto -avainten luomiseksi, jota käyttämällä voidaan asiakkaan ja palvelimen välillä keskustella turvallisesti. [31]

TLS-protokolla toimii verkkoliikenteessä verkko- ja sovelluskerrosten välissä, ja perustuu symmetriseen ja epäsymmetriseen salausalgoritmiin sekä tiivistealgoritmiin. Näiden algoritmien tarkoituksena on saavuttaa verkkoliikenteessä luottamuksellisuus ja mo-

lemminpuolinen osapuolten ja liikenteen todentaminen. TLS käyttää niin sanottua X.509-sertifikaattia, joka perustuu tiettyyn julkisen avaimen algoritmiin, jota käytetään muiden muassa symmetrisen salausavaimen ja koskemattomuuden takaavan tiivistealgoritmin luomiseen. Prosessissa tapahtuu myös samanaikaisesti asiakkaan ja palvelimen välinen identiteettien todennus, joista ainoastaan palvelimen todennus on tyypillisesti pakollista. Tässä työssä tullaan kuitenkin käyttämään molemminpuolista identiteettien todennusta, jolla pystytään rajaamaan paremmin palvelimen rajapintojen kutsumisen mahdollisuuksia minimiin. [32]

Kuvassa 5.1 nähdään sekvenssikaavio, joka esittää asiakkaan ja palvelimen käymän kommunikoinnin askeleet salatun TLS-yhteyden muodostamiseksi.



Kuva 5.1: Yhteyden muodostuksen vaiheet TLS-protokollassa

5.1.2 Pankkien vaatimat sertifikaatit ja käyttäjän todennukset

Tässä projektissa oli alkuperäisenä tarkoituksena toteuttaa prototyyppinä käytettävä pankkiyhteys Enerocilta Ajas Paymentsia käyttävälle kauppiaalle Danske Bankin ohjelmointirajapintoja käyttäen, mutta työssä päädyttiin lopulta valitsemaan Osuuspankin rajapinnat. Suurimpana syynä tähän oli pankkien väliset eroavaisuudet siinä, miten kehitysvaiheessa olevan maksupalvelun tunnistautuminen ja valtuutus oli tehtävissä. Tässä tapauksessa Danske Bank olisi vaatinut virallisten sertifikaattien anomista jo kehitysvaiheessa tapahtuvaa rajapintojen käyttöä varten, ja tämän olisi koettu vievän liian paljon aikaa varsinai-

sesta prototyypin kehityksestä. [33]

Sen sijaan lopulta valitulla Osuuspankilla käytäntö oli erilainen. Osuuspankin rajapintaa sai kehitysvaiheessa käyttää rekisteröimällä sovellus Osuuspankin OP Developer -palvelussa, luomalla kehitystä varten sertifikaatit ja avainparit Osuuspankin tarjoamalla sertifikaattien generointiohjelmalla. Näiden lisäksi tuli rekisteröidä oma palvelu kolmantena palveluntarjoajana Osuuspankin tarjoamassa URL-osoitteessa. [34]

Kuitenkin kaikkien pankkien ja rahalaitosten rajapintojen käyttöä varten tarvitaan viimeistään tuotantokäytössä samanlaiset QWAC- ja QCSEAL-sertifikaatit ja luvat joltain varmennetulta auktoriteetilta, kuten tässä tapauksessa Finanssivalvonnalta.

5.1.3 API-rajapintojen rajoitukset, varmenteet ja valtuutukset

Koko maksupalvelun ja siihen liittyvän datan arkaluontoisuuden vuoksi ovat palvelun tietoturva ja datan eheys erittäin tärkeissä asemissa. Tämän vuoksi on tietoturvan edistämiseksi pyritty kehittämään myös itse Ajas Payments -palvelun sisälle mahdollisimman hyviä periaatteita noudattavaa ohjelmakoodia.

Maksupalvelun palvelinpuolelle tehtäviin kutsuihin on esimerkiksi otettu käyttöön sertifikaatteja käyttävä varmennustapa. Tämä tarkoittaa sitä, että Ajas Paymentsin palvelinpuolen rajapintaa voidaan kutsua ainoastaan, jos kutsun mukana toimitetaan oikeanlainen sertifikaatti asiakkaan selaimesta. Toisin sanoen tämän tarkoituksena on pystyä rajaamaan rajapinnan kutsuminen tapahtuvan ainoastaan Ajas Paymentsin asiakaspuolen sovelluksen kautta.

Ainakin vielä palvelun kehityksen alkuvaiheessa tullaan käyttämään itse luotuja sertifikaatteja tähän tarkoitukseen, mutta viimeistään tuotantoon siirtyessä tullaan hankkimaan sertifikaatit luotetulta varmenteiden myöntäjältä (engl. Certificate authority, CA). Syynä tähän on se, etteivät itse luodut sertifikaatit pysty tarjoamaan yhtä varmaa turvallisuutta kuin varmenteiden myöntäjän vahvistamat sertifikaatit.

5.1.4 Käyttäjän varmennus JWT-menetelmällä

Tämän työn osalta ainakin Osuuspankki käyttää API-rajapintojensa kutsuissa niin sanottua JWT-menetelmää (engl. JSON Web Token) käyttäjän ja hänen käyttöoikeuksiensa varmentamiseksi. JWT-menetelmässä luodaan kaksi peräkkäistä JSON-objektia, jotka erotetaan keskenään pisteellä. Objektit ovat otsikkotiedot (engl. header) ja tietosisältö (engl. payload), ja nämä molemmat salataan ylätunnuksessa määritellyllä algoritmilla. Tämän jälkeen salatut JSON-objektit muunnetaan Base64-koodauksella merkkijonoiksi. JWT-tunnus sisältää vielä kolmannen osan, joka on allekirjoitus (engl. signature). Allekirjoitus luodaan käyttämällä koodattua ylätunnistetta ja tietosisältöä sekä salasanaa, ja salaamalla näistä määritellyllä algoritmilla allekirjoitusta esittävä merkkijono. Täten lopullinen JWT-tunnus on nämä kolme koodattua merkkijonoa yhdessä, erotettuna toisistaan pisteillä. [35] JWT-tunnusta käytetään tyypillisesti varmistamaan, onko käyttäjällä oikeus tiettyyn haluttuun operaatioon. Tämän työn tapauksessa Osuuspankin rajapintaa käyttäessä JWT-tunnusta käytetään loppuasiakkaan tunnistautumiseen ja valtuuttamiseen, sekä tämän jälkeen loppuasiakkaan pankkitunnuksia maksuoperaation vahvistamiseksi. [34]

5.1.5 Salaisten tietojen talletus ja käyttö

Osana Secure by Design -periaatetta, jossa heti kehityksen alusta alkaen otetaan turvallisuus huomioon, on tässä projektissa pyritty pitämään muun muassa kaikenlaiset salaiset avaimet, sertifikaatit, salasanat ja tunnukset poissa ohjelmakoodista jo kehitysvaiheessa. Sen sijaan aikaa on käytetty jo alusta alkaen siihen, että pyritään löytämään hyviä tapoja ja paikkoja säilöä kyseisiä arkaluontoisia tietoja. Tällä on pyritty saamaan hyvä pohjarakenne salaiselle tiedolle, sekä pyritään paremmin välttämään esimerkiksi vahingossa tapahtuneita salaisuuksien julkaisuja versionhallintaan.

Projektissa on tähän mennessä käytetty pääasiassa Dockerin omaa toteutusta salaisuuksien talletukselle, nimeltä Docker Secrets. Jotta voidaan käyttää Secrets-palvelua, on Dockeria käytettävä niin sanotussa parvitilassa, nimeltään Docker Swarm. Parvitilassa

Docker-sovelluksia suorittavat fyysiset tai virtuaaliset koneet ovat yhdessä ryhmässä, joita voidaan helpommin hallita yhdessä. Tätä hyödyntämällä voidaan käyttää sovelluksen eri osien välillä samoja salaisia tietoja niin, ettei niitä kuitenkaan tarvitse säilöä esimerkiksi ohjelmakoodissa. Secrets-palvelun käyttö on myös esimerkiksi yksinkertaista erillistä JSON-tiedostoa parempi säilömistapa, sillä palveluun säilötyt salaisuudet ovat salattuja sekä tallennettuina että lähetyksen aikana. Tämän lisäksi Dockerilla täytyy eksplisiittisesti määrittää tietyt suoritettavat palvelut, jotka saavat käyttää mitäkin salaisuuksia. Tällä pyritään pitämään kiinni vähimpien oikeuksien periaatteesta, ja tällä tavoin suojaamaan ohjelmistokokonaisuuden käyttämiä salaisia tietoja entistä paremmin. Periaatteen tarkoituksena on minimoida mahdolliset ohjelmiston vikatilanteista tai hakkereiden hyökkäyksistä johtuvat haitat. [36]

5.2 Maksun suorituksen prosessit ohjelmistossa

Maksupalvelun odotusten mukainen prosessi on hyvin yksinkertainen loppukäyttäjän näkökulmasta, ja helppokäyttöisyys onkin luonnollisesti yksi palvelun tärkeistä tavoitteista. Kuitenkin palvelussa tapahtuu jo tässä kehityksen prototyyppivaiheessa paljon, suurimmaksi osaksi sovelluksen palvelinpuolella.

Vaikka tässä työssä on toistaiseksi toteutettu suora pankkiyhteys ainoastaan Osuuspankille, tulisi työssä käytettävän PSD2-direktiivin perusteella olla muidenkin pankkien yhteyksien luonti samankaltainen, sillä yksi direktiivin tarkoituksista oli standardisoida esimerkiksi pankkien rajapinnat ja niiden toteutukset.

5.2.1 Maksun luominen loppukäyttäjällä

Palvelun käyttö alkaa loppukäyttäjällä siitä, kun käyttäjä on asioinut jossain verkkokaupassa ja haluaa maksaa ostoksensa. Verkkokaupasta siirrytään Ajas Payments -maksupalvelun asiakassovellukseen, jonka kutsun yhteydessä toimitetaan kyseisen tilauksen ostos-

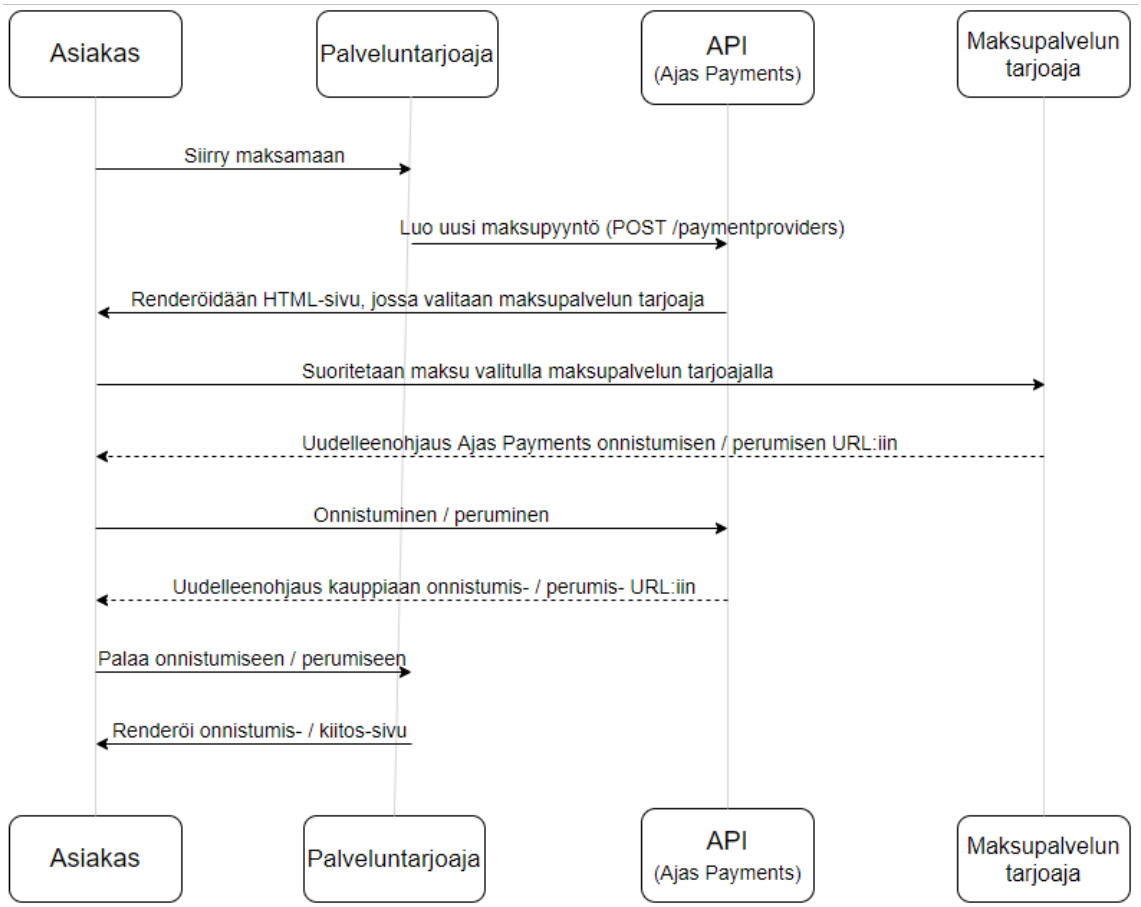
ten ja käyttäjän omat tiedot. Tähän kuuluvat muiden muassa tuotteiden määrä ja tuotekoodit, loppusumma, viitenumero ja mahdollinen toimitusosoite. Asiakassovellus puolestaan kutsuu välittömästi Paymentsin rajapintaa, johon maksun tiedot edelleen toimitetaan.

Tämän jälkeen palvelinpuoli kutsuu Checkout Finland -maksupalvelun rajapintaa, jonne maksutiedot toimitetaan myös. Checkoutin rajapinta palauttaa listan eri maksupalvelun tarjoajista asiakassovellukselle, joka puolestaan renderöi kaikki maksuvaihtoehdot käyttäjän valittavaksi. Checkout Finlandin käyttö tässä kohtaa on ainoastaan käytössä vielä prototyypissä, mutta ei tule enää olemaan käytössä tuotantoversiossa, sillä Checkoutin käyttö on testiympäristön ulkopuolella maksullista, eikä tähän olla pyrkimässä muutenkaan.

Loppukäyttäjä valitsee selaimensa asiakasnäkymässä palvelujen listasta haluamansa maksutavan ja vahvistaa siirtyvänsä maksamaan ja tunnistautumaan. Asiakaspuolen palvelu kutsuu jälleen palvelinta, mukanaan maksun aiemman datan lisäksi nyt haluttu maksutapa. Tässä kohtaa palvelin käyttää joko Checkout Finlandin palvelua suoraan välikätenä, tai Osuuspankin tapauksessa kutsutaan sille tehtyä rajapinnan kutsun toteutusta. Maksu tulee loppuasiakkaalta Enerocille, joka myöhemmin puolestaan pienen provision ottamisen jälkeen siirtää rahat edelleen itse kauppiaalle.

Checkout Finlandia käyttäessä kaikki maksamiseen tarvittavat tiedot löytyvät JSON-objekteista, jotka myös renderöitiin asiakassovelluksen näkymään. JSON-objektissa on siis esimerkiksi oikeat URL-osoitteet, tunnukset ja maksun yleiset tiedot valmiina, ja valitun maksutavan JSON-objekti lähetetään HTTP POST -pyynnön mukana oikeaan URL-osoitteeseen. Kutsu ohjaa käyttäjän maksutavan, esimerkiksi pankin, tunnistautumisnäkymään, jossa käyttäjä syöttää normaaliin tapaan pankkitunnuksensa ja hyväksyy maksun.

Kuvassa 5.2 on sekvenssikaavio, jonka tarkoituksena on esittää maksuihin liittyvät tapahtumat loppuasiakkaan ja eri järjestelmien tai toisten osapuolten välillä. Kaaviossa käydään läpi maksupalvelun käytössä tapahtuvat prosessit korkealla abstraktiotasolla.

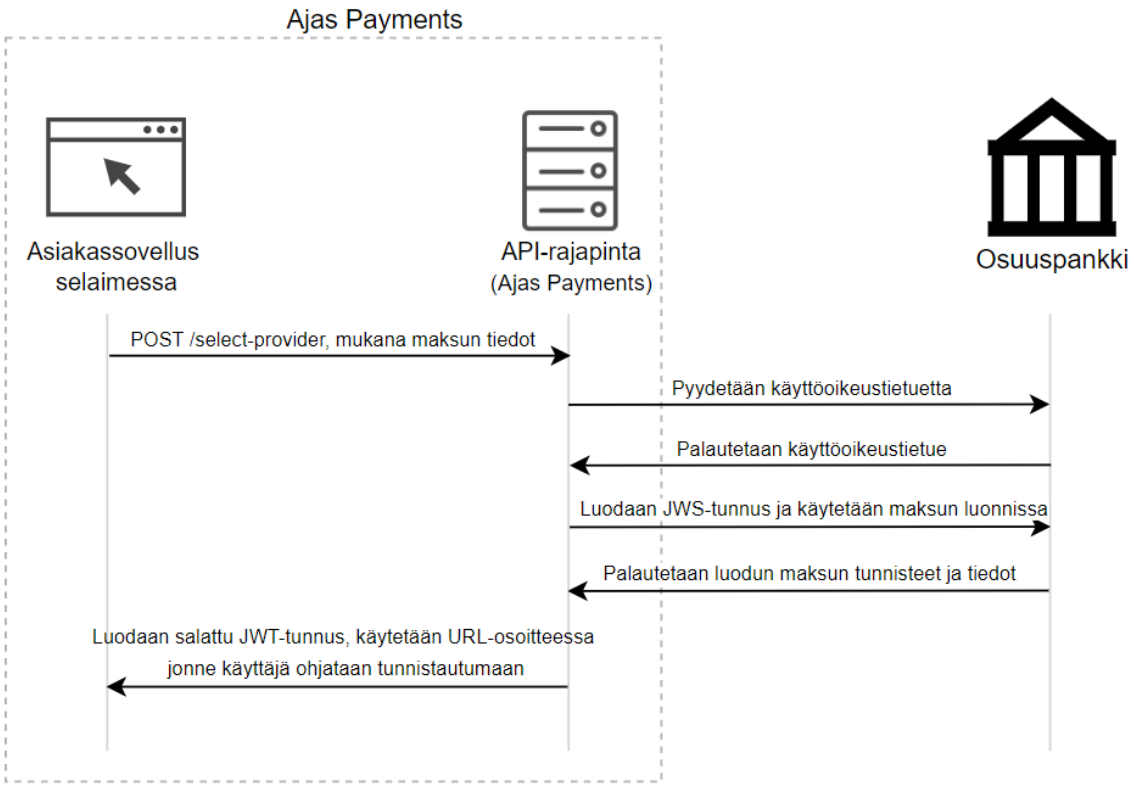


Kuva 5.2: Sekvenssikaavio loppuasiakkaan ja eri osapuolten välisistä toiminnoista

Suoran pankkiyhteyden tapauksessa, kuten Osuuspankilla tuotteen prototyypissä, palvelimelle viedään kutsun mukana ainoastaan tilauksen loppusumma euroina, ja tätä käytetään rakentamaan yksinkertainen JSON-objekti, jossa on maksunsaajan (Enerocin) tiedot, summa ja aiemmin saatu viitenumero. Osuuspankilla maksukehotuksen luomista varten tulee ensin itse luoda käyttöoikeustietue (engl. access token) kutsumalla Osuuspankin rajapintaa HTTP POST -pyynnöllä, jonka mukana toimitetaan muun muassa halutut käyttöoikeudet, Enerocin asiakastunnus ja -salaisuus Osuuspankissa. Tämän jälkeen luodaan niin sanottu JWS-tunnus (engl. JSON Web Signature), joka on salattu merkkijono, joka on muodostettu muun muassa maksun JSON-muotoisista tiedoista ja salausavaimesta. Maksun tiedot on täten turvallista viedä seuraavaan vaiheeseen, jossa luodaan uusi maksu-

kehote jälleen Osuuspankin rajapinnan kautta. Lopuksi palvelin luo vielä JWT-tunnuksen (engl. JSON Web Token) ES256-salausalgoritmilla, käyttäen aiemmin saatua valtuutustunnusta ja jälleen Enerocin asiakastietoja Osuuspankissa. Tätä JWT-tunnusta käytetään yhdessä Osuuspankin tarjoaman URL-osoitteen kanssa, johon käyttäjä ohjataan tunnistautumaan omilla pankkitunnuksillaan.

Kuvan 5.3 sekvenssikaavio esittää suoran pankkiyhteyden muodostuksen päätapahtumia PSD2-direktiiviä noudattavaan rajapintaan, kuten työssä esimerkkinä olevaan Osuuspankkiin.



Kuva 5.3: Sekvenssikaavio asiakkaan ja eri osapuolten välisistä toiminnoista

5.2.2 Tapahtumat maksun suorituksen jälkeen

Tähän mennessä on käyty prosessia läpi siihen asti, että maksupalvelua käyttänyt loppukäyttäjä on maksanut ostoksensa Ajas Paymentsin kautta ja lopettanut omalta osaltaan

palvelun kanssa vuorovaikutuksen. Kuitenkin tässä kohtaa rahat ovat vielä Enerocilla hallussa, eivätkä itse kauppiaalla.

Kun maksu on luotu ja tämän jälkeen vahvistettu ja suoritettu, haetaan vielä maksun suorituksen tiedot joko Checkoutin tai Osuuspankin rajapintojen kautta. Jos kaikki on kunnossa, on maksu tullut Enerocille onnistuneesti. Tämän jälkeen luodaan maksukehotus Osuuspankin rajapintaa käyttäen kauppiaalle, kuitenkaan sitä vielä lähettämättä, sekä tallennetaan luodun SEPA-maksun tiedot Ajas Paymentsin tietokantaan myöhempää varojen siirtoa varten. Tässä kohtaa luodusta maksusta otetaan Enerocille tietty provisio-osuus. Ennen maksun tallennusta tietokantaan tarkistetaan, onko kyseisenä päivänä ollut aiempia maksuja tallennettu. Jos aiempia maksuja on ollut, korvataan SEPA-maksun tietoihin sen päivän aiempien maksujen mukainen valtuutustunnus (engl. Authorization ID). Täten tietyn päivän kaikilla tietokantaan tallennetuilla SEPA-maksuilla on sama valtuutustunnus, jota voidaan myöhemmin käyttää hyväksi lopullisessa varojen siirrossa kauppiaille.

Maksupalvelun prototyypissä maksujen eteenpäin vienti tapahtuu kutsumalla Paymentsin rajapintaa, jonka avulla pystytään suorittamaan sen päivän kaikki maksut kerralla. Kutsun saatuaan palvelin hakee tietokannasta viimeisimmän SEPA-maksun ja tarkistaa onko sen päivämäärä sama kuin tänään. Jos on, otetaan talteen SEPA-maksun tiedoista sen valtuutustunnus, jota käyttämällä voidaan Osuuspankin rajapinnan ja tunnistautumissivun avulla helposti suorittaa kaikki päivän maksut kauppiaille.

5.3 Palvelun jatkokehitys

Ajas Payments jäi tämän diplomityön osalta prototyyppivaiheeseen, eli palvelu ei ole päätynyt vielä yksinkertaisia beta-testauksia pidemmälle. Palvelu on vaiheessa, jossa sitä voidaan testiympäristössä jo helposti testata satunnaisesti luodulla testimaksudatalla ja keksityillä kauppiaan tiedoilla.

Tämän vuoksi palvelussa on luonnollisesti vielä paljon uutta kehitettävää ja parannettavaa ennen siirtymistä tuotantoon. Osa tässä aliluvussa esitellyistä parannusten tarpeista johtuvat yksinkertaisesti joko ajan puutteesta tai odottamattomista esteistä, tai siitä, että parannus ei ollut vielä tässä prototyypissä tarpeeksi olennainen ominaisuus.

5.3.1 Suorat yhteydet maksupalvelun tarjoajiin

Tässä työssä on toteutettu toistaiseksi suora pankkiyhteys ainoastaan Osuuspankille prototyyppinä. Osuuspankin rajapintoja on käytetty suorana pankkiyhteytenä, kun loppukäyttäjältä siirretään varoja Enerocin tilille, sekä kun Enerocilta lopulta siirretään varsinaiselle kauppiaalle kuuluva osuus maksusta. Kuitenkin kaikki muut loppukäyttäjän maksutapavaihtoehdot käyttävät vielä prototyypissä Checkout Finland -maksupalvelua välikätenä, ja tästä on tarkoitus vähitellen luopua kokonaan.

Työssä hyödynnetyn PSD2-direktiivin perusteella maksuyhteyksien luominen muihin rajapintoihin pitäisi seurata samoja direktiivin luomia standardeja. Tämä tarkoittaa, että kaikkiin eri maksupalvelun tarjoajien PSD2-direktiiviä noudattaviin rajapintoihin yhdistäminen tulisi tapahtua noudattaen samoja askeleita ja käyttäen samanlaisia tietoja.

5.3.2 Kauppiaan tilisiirtojen automatisointi

Tällä hetkellä lopulliset tilisiirrot Enerocilta kauppiaalle tehdään vielä manuaalisesti kutsumalla Paymentsin rajapintaa. Rajapinta palauttaa URL-osoitteen, jossa manuaalisesti tunnistautumalla voidaan hyväksyä sen päivän kaikki tilisiirrot niille kuuluville kauppiaille. Tähän ollaan kuitenkin myöhemmässä kehitysvaiheessa tulossa tekemään muutos niin, että maksut siirtyvät kauppiaille automaattisesti yksittäisten ostosten jälkeen. Toinen vaihtoehto tähän voisi olla toteuttaa samankaltainen kerran päivässä tapahtuva kaikkien maksujen yhtäaikainen siirto, mutta automatisoituna.

Kauppiaan tilisiirtoihin tullaan todennäköisimmin myös jatkossa käyttämään Osuuspankin palveluja, joka tarjoaa niin sanotun Web Services -kanavan käytön muun muassa

maksujen automatisoituun lähetykseen. Tämän palvelun avulla tunnistautuminen pystytään suorittamaan ohjelmistossa käyttäen maksajan toimittamia salausavaimia ja sertifikaatteja, sekä Osuuspankin kautta saatavia käyttöoikeustietueita ja niiden perusteella luotuja JWS-tunnuksia todennusta varten. [37]

5.3.3 Kauppiaan hallintamoduuli

Tässä prototyypissä on toteutettu tähän mennessä ainoastaan loppuasiakkaalle maksupalvelun tarjoajien valinta ja maksujen vienti Enerocin kautta lopulliselle kauppiaalle. Palvelun tulevaisuuden suunnitelmaan kuitenkin kuuluu myös kauppiaan oma hallintamoduuli, jossa kauppias pystyy helpon rajapinnan kautta antamaan yrityksensä tiedot maksupalvelun käyttöönottamiseksi. Hallintamoduulin on tarkoitus tulla osaksi Enerocin omaa jo olemassa olevaa tuotetta, Ajas-ajanvarauspalvelun Touch-nimistä hallintasovellusta.

Ajas Paymentsin hallintamoduuli olisi siis upotettuna Ajas Touch -palveluun. Myös hallintamoduulin tietokanta on suunniteltu olevan osana Touch-palvelun tietokantaa, eikä Paymentsin, sillä kauppiaiden joitain käyttäjätietoja on jo valmiina tässä tietokannassa. Tämän ajateltiin selkeyttävän lopulta enemmän Paymentsin rakennetta, sillä nyt kauppiaiden tiedot ovat samassa tietokannassa, ja jättää Paymentsin tietokannan yksinkertaisemmaksi.

5.3.4 Suorituskyvyn ja saatavuuden kehitys

Palvelun kehityksessä on alusta alkaen pyritty kiinnittämään huomiota sen suorituskykyyn ja saatavuuteen. Näistä kahdesta suorituskykyä on pyritty testaamaan hieman jo kehitysvaiheessa, mutta saatavuusasteen arviointia ei olla tehty. Kuitenkin yksi syy miksi palvelun taustalle on esimerkiksi valittu Dockerin konttiteknologia, on Dockerin kyky skaalata kontteja huomattavasti helpommin kuin esimerkiksi tyypillisessä virtuaalikoneiden tapauksessa. Suorituskykyä pystytään parantamaan oikeastaan palvelun jokaisen osa-alueen puolesta, mutta tällä hetkellä suurimmat kohteet tässä ovat itse Dockerin kontit,

sekä Paymentsin palvelinpuolen sovellus ja sen tarjoamat rajapinnat.

Yleisen suorituskyvyn ja palvelun skaalautuvuuden lisäksi oleellista on myös palvelun saatavuus, joka puolestaan riippuu melko paljon suorituskyvystä ja skaalautuvuudesta. Nämä kaikki laatuattribuutit ovat tämän työn osalta tärkeitä yksityiskohtia, mutta niiden tarkempi kehitys on vielä toistaiseksi jäänyt sivummalle, ja tulevat ajankohtaisemmaksi vasta lähempänä tuotantoon siirtymistä. Eräs tärkeä mahdollinen ongelmakohta, joka tulee ottaa huomioon tässä työssä, on useat yhdenaikaiset käyttäjät. Tällöin palvelun skaalautuvuus on hyvin tärkeässä roolissa, ja sen tärkeys kasvaa samanaikaisten käyttäjien määrän kasvaessa. Korkea samanaikainen käyttöaste voi heikentää palvelun suorituskykyä sekä saatavuutta, sillä palvelun resursseja käytetään tällöin enemmän, joka taas voi helposti hidastaa suoritusaikoja. Skaalautuvuutta ja saatavuutta pystytään parantamaan käyttämällä ohjelmistossa esimerkiksi kuormantasaajia eri kontti-instansseille sekä käyttämällä välimuistia tehokkaasti niissä paikoissa, missä sitä voidaan tehdä.

5.3.5 Siirto pilvialustalle

Suurimmat pilvipalvelut ovat jo erittäin luotettavia ja vakaita, joten maksupalvelun on todettu olevan mahdollinen julkaista pilvessä. Palvelu tullaan julkaisemaan todennäköisimmin Amazonin tarjoaman AWS-pilvialustan avulla. Pilvialusta on todennäköisin vaihtoehto palvelun testi- ja tuotantoversioiden julkaisuun, sillä esimerkiksi palvelun skaalaaminen on erittäin helppoa ja resurssien käyttö on optimoitu varsin hyvin, joka vähentää palvelusta aiheutuvia kuluja. Pilvipalveluiden tietoturva on myös nykyään korkealla tasolla, ja esimerkiksi AWS tarjoaa hyvät valmiudet tietoturvan toteutukselle. [38] Vaikka tietoturva on pilvipalveluissa otettu hyvin huomioon, maksuihin ja asiakkaisiin liittyvät tiedot tullaan todennäköisesti pitämään tietokannoissa yrityksen hallinnassa sen omalla palvelimella. Tällä pyritään lisäämään palvelun tietoturvaa entisestään. Yrityksen palvelinta on harkittu myös palvelun varapalvelimeksi, mikäli jotain yllättävää tapahtuu pilvialustalla. Varapalvelimella pystyttäisiin lisäämään palvelun saatavuutta.

6 Työn arviointi ja pohdintaa

Tässä luvussa arvioidaan käytännön työnä suunniteltua ja kehitettyä palvelua eri näkökulmista. Tarkastelun kohteena ovat pääasiassa eri laatuattribuuttien toteutuminen todellisuudessa. Luvussa käsitellään laatuattribuuttien arvioinnin ohessa myös pintapuolisesti tuotteen jatkokehitystä niiden osalta ja mahdollisia uusia tulevia haasteita ja tarpeita. Tuotteen laatuattribuutteja ovat muunneltavuus ja modulaarisuus, skaalautuvuus, helppokäyttöisyys sekä tietoturva. Nämä termit on esitelty tarkemmin luvussa 2.2.

Työssä yksi tärkeimmistä teemoista on ollut PSD2-direktiivi ja sen tuomat muutokset maksupalvelujen kehitykseen. Tässä luvussa pyritäänkin edellä mainittujen aiheiden lisäksi vastaamaan työn tutkimuskysymyksiin tutkimalla sitä, miten työssä käytetty PSD2-direktiivi vaikuttaa yleisesti maksupalveluiden kehitykseen, ja miten direktiivin muutokset näkyivät tässä työssä. Tässä luvussa pyritään myös tutkimaan käytetyn PSD2-direktiivin metodeja, joilla edistetään palveluiden tietoturvaa, palveluiden kehittämistä ja maksupalveluiden välistä kilpailua.

6.1 Käytettävyys ja saatavuus

Ajas Payments -palvelun suunnitteluvaiheessa ja tämän tutkielman luvun 5 alussa määriteltiin, että palvelun käyttö tulisi saada mahdollisimman helpoksi ja luontevaksi sekä kauppiaalle että lopulliselle verkkokaupan loppuasiakkaalle. Näihin projektin vaatimuksiin liittyvät pääasiallisesti tuotteen käytettävyys ja sen saatavuus.

Ohjelmiston käytettävyys on tyypillisesti yksi erittäin tärkeä mittari ohjelmiston koko-

naisvaltaisen laadun ja täten sen onnistumisen suhteen. Käytettävyydellä voidaan tarkoittaa monia eri asioita ohjelmistotuotantoon liittyen, sillä käytettävyyttä voidaan esimerkiksi tarkastella useasta eri näkökulmasta. Esimerkiksi tässä työssä käytettävyyttä voitaisiin tarkastella asiakkaana toimivan loppukäyttäjän ja verkkokauppaa pitävän kauppiaan näkökulmista erikseen. [39]

Maksupalvelun käytettävyyttä ei vielä olla testattu oikeilla mahdollisilla käyttäjillä, eli kauppiailla tai loppukäyttäjillä. Käytettävyystestausta on kuitenkin tarkoitus tehdä palvelun kehityksen myöhemmässä vaiheessa, jotta pystytään saamaan palautetta oikeilta käyttäjiltä, ja tämän perusteella mahdollisesti kehittää palvelua tiettyyn suuntaan. Tässä kohtaa olisi myös tarkoitus määrittää palvelulle taattu saatavuusprosentti tarkasti, mutta alkuperäisenä tavoitteena on pidetty noin 99.95 % saatavuutta. Tämä vastaisi esimerkiksi yhtä päivää kohden keskimäärin yhteensä 43 sekunnin katkosta. Kuitenkin palvelun saatavuusprosentti on myös hyvin paljon riippuvainen sen julkaisuun käytetystä alustasta, eli tässä työssä todennäköisimmin AWS-pilvipalvelusta.

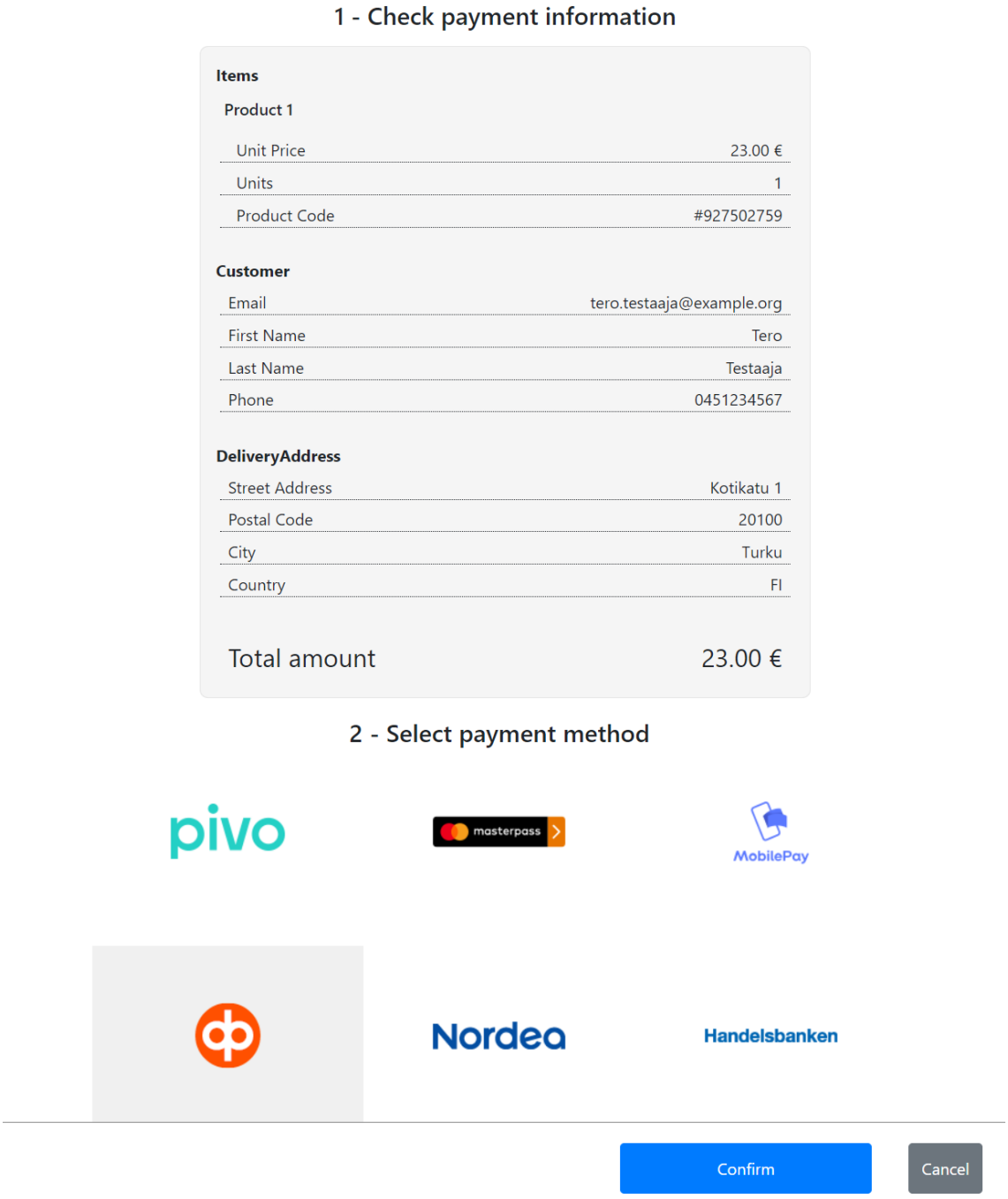
6.1.1 Käytettävyys loppukäyttäjälle

Kehitetyn maksupalvelun loppukäyttäjän näkemä käytettävyys ja yleinen käyttäjäkokemus alkaa siitä, kun käyttäjä siirtyy kauppiaan verkkokaupasta maksamaan tuotteitaan, ja päättyy kun käsiteltävä maksutapahtuma on joko onnistuneesti suoritettu tai keskeytetty.

Käyttäjän näkemässä käyttöliittymässä on pyritty vahvasti yksinkertaisuuteen ja intuitiivisuuteen. Asiakaspuolen käyttöliittymä on ulkoasultaan varsin yksinkertainen ja koostuu kahdesta eri osasta, jotka ovat yhteenveto maksettavan maksun tiedoista, kuten valitut tuotteet tai palvelut, asiakastiedot ja mahdolliset toimitusosoitteet, sekä listatut vaihtoehdot eri maksutavoille. Loppuasiakas voi näkymässä ensin tarkistaa, että maksun tiedot ovat oikein ja tämän jälkeen valita haluamansa maksutavan.

Kuvassa 6.1 esitetään loppukäyttäjän näkemää käyttöliittymää. Kuvassa nähdään vain kaksi ensimmäistä riviä palveluntarjoajista, mutta niitä löytyy todellisuudessa enemmän

sivua alas vierittämällä.



Kuva 6.1: Loppuasiakkaan näkemän maksusivun käyttöliittymän komponentit

Käyttöliittymää pystyisi kuitenkin vielä yksinkertaistamaan ja täten selkeyttämään en-

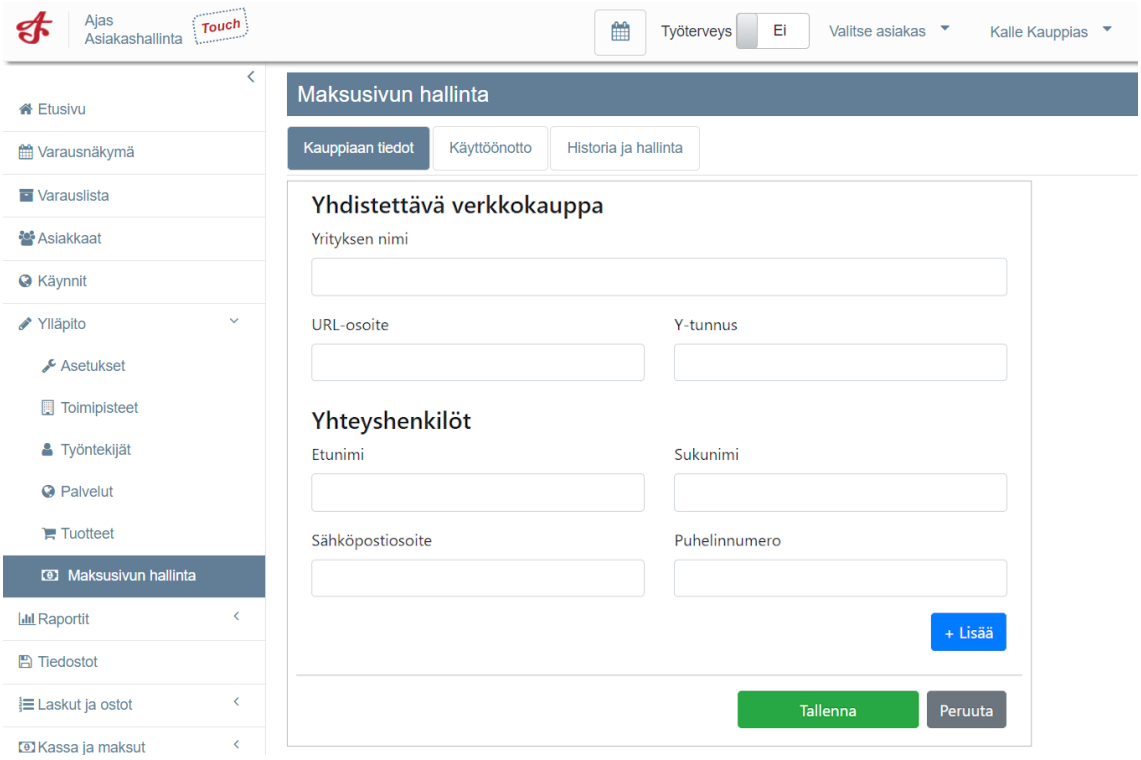
tuudestaan ainakin muokkaamalla maksutietojen yhteenvetoa. Yhteenvedon tietoja voisi esimerkiksi vähentää huomattavasti niin että näkymässä esitettäisiin vaikka ainoastaan maksun summa ja verkkokaupan nimi. Yhteenvedon pystyisi mahdollisesti myös jopa poistamaan kokonaan tästä näkymästä, sillä useissa verkkokaupoissa on oletuksena jo itse toteutettu yhteenvetonäkymä tilauksesta, jolloin tässä palvelussa tällä hetkellä oleva yhteenveto olisi vain turhaa toistoa. Tämän perusteella voisi olla parasta, että maksupalvelua käyttävä kauppias saisi itse määritellä hallintasivun asetuksissa, haluaako esittää maksun yhteenvetotiedot loppuasiakkaalle, ja mitä tietoja sivulla näytettäisiin. Tällöin kauppias voisi halutessaan toteuttaa yhteenvetotietojen esittämisen omassa verkkokaupassaan tai jättää sen maksupalvelun tehtäväksi.

6.1.2 Käytettävyys kauppiaalle

Kauppiaan näkemää hallintamoduulia ei vielä tämän diplomityön kirjotushetkellä olla kehitetty. Moduulin osalta on kuitenkin jo suunniteltu sen toiminnallisuudet ja käyttöliittymän ulkoasua.

Tarkoituksena on, että maksusivun hallintamoduuli jaettaisiin kolmeen osaan, esimerkiksi välilehdillä, joita käyttöliittymän tämänhetkisessä suunnitelmassa on käytetty. Yhdellä välilehdellä kauppias voi lisätä tai muokata yrityksensä tietoja ja yhteyshenkilöitä, toisella kauppias voi aktivoida maksupalvelun verkkokaupassaan ja nähdä yleiset ohjeet maksupalvelun käyttöönotolle. Kolmannella välilehdellä kauppias pystyy tarkastelemaan maksupalvelun käytön tilastoja, esimerkiksi maksupalvelun käytön historiasta ja tapahtuneista maksutoimeksiannoista.

Kuvassa 6.2 nähdään tämänhetkinen suurpiirteinen suunnitelma siitä, miltä kauppiaan käyttämä hallintamoduulin käyttöliittymä tulisi näyttämään ja mitä tietoja maksupalvelun käyttöönotto vaatisi.



Kuva 6.2: Karkea suunnitelma kauppiaan käyttämästä hallintamoduulista

Tämän moduulin suunnitelma on ollut vielä melko vähällä huomiolla, eikä ole siis välttämättä sitä mitä lopulta tullaan kehittämään osaksi palvelua. Esimerkiksi käyttöliittymässä moduulin jako välilehdillä kolmeen osaan saattaa olla huono ratkaisu käytettävyyden kannalta, sillä esimerkiksi kauppiaan syöttämät tiedot ja käyttöönotto voisivat olla allekkain samassa näkymässä. Tällä tavoin voitaisiin vähentää ylimääräisiä klikkauksia eri välilehtien välillä. Kuitenkin Ajas-palvelussa on jo entuudestaan suosittu useissa paikoissa välilehtien käyttöä kuvan 6.2 esittämällä tavalla, ja tämä myös vaikutti tässä tapauksessa päätöksen tekoon suunnitelmassa.

6.1.3 Palvelun saatavuus ja sen takaaminen

Saatavuudella tarkoitetaan sitä, kuinka suuren osan ajasta ohjelmisto tai palvelu on käynnissä ja käytettävissä. Ohjelmistokehityksessä pyritään luonnollisesti mahdollisimman

korkeaan saatavuuteen. Ohjelmiston saatavuus voidaan helposti liittää sen käytettävyyteen, sillä ohjelmiston saatavuus vaikuttaa siitä välittyvään käyttäjäkokemukseen. Tämä puolestaan tätä kautta vaikuttaa usein myös käyttäjän käsitykseen ohjelmiston laadusta ja luotettavuudesta, ja ohjelmiston saatavuutta kutsutaankin monesti myös loppuasiakkaan kokemaksi luotettavuudeksi tai luotettavuudeksi huollettavuudella. Saatavuusasteen määrittämisessä tulee huomioida käynnissä olon ajan lisäksi myös järjestelmän palautukseen kuluva sammuksissa olon aika. [40]

Ajas Payments -palvelun todellista toteutunutta saatavuutta ei olla vielä tällä hetkellä testattu käytännössä. Saatavuus on kuitenkin pyritty ottamaan ohjelmiston kehityksessä huomioon heti alusta alkaen esimerkiksi eri teknologioita valittaessa. Yleisesti saatavuutta tavoiteltaessa tulee ottaa huomioon useita eri osa-alueita ja tapoja. Hyviä asioita ottaa huomioon tätä ajatellessa ovat esimerkiksi ohjelmissa tapahtuvat vikatilat, skaalautuvuus, riskien vähentäminen ja lieventäminen, saatavuuden valvonta ja saatavuusongelmien käsittely selkeästi määritetyillä tavoilla. [41]

Työssä on tarkoituksena käyttää esimerkiksi kuorman tasaajia verkkoliikenteen jakamiseksi eri sovellusinstanssien välillä niin, ettei yhdelläkään suoritettavalla Docker-kontilla ole liikaa kyselyjä hoidettavanaan kerrallaan. Docker-kontit on sijoiteltu palvelussa omiin virtuaalikoneisiinsa ja ovat aiemmin mainitussa parvitilassa (Docker Swarm), jolloin niiden käyttöä voidaan helposti hallita kuormantasaajalla. Palvelun alijärjestelmiä suorittavia kontteja pystytään myös skaalaamaan, eli esimerkiksi suoritukseen vaadittavia resursseja pystytään automaattisesti lisäämään tarpeen tullen, ja toisaalta myös vähentämään, kun tarve laskee taas. Resurssien käyttö pyritään siis optimoimaan palvelun käytön tarpeiden mukaisesti.

Toinen työssä tähän mennessä erityisesti huomioitu asia on virhetilojen hallinta erinäisillä tavoilla. Mahdollisesti eniten riskialttiutta esiintyy maksupalvelun asiakas- ja palvelinsovellusten useassa kohtaa suorittamissa HTTP-kyselyissä. Kun palvelu suorittaa HTTP-kutsun johonkin ulkopuoliseen rajapintaan, on maksupalvelun toimivuus tällöin

riippuvainen myös näistä kutsutuista rajapinnan tarjoajista, kuten tässä työssä pankeista, sekä väliin luodun yhteyden toimivuudesta.

Tämän lisäksi, vaikka maksupalvelu on jaettu useaan itsenäiseen osaan, ne kutsuvat toisiaan verkon yli. Tällöin on mahdollista saada tilanne, jossa esimerkiksi asiakassovellus toimii normaalisti, mutta palvelinsovellus on hetkellisesti kaatunut, joka taas aiheuttaa virhetilanteen asiakassovelluksessa. Kaikenlaiset virhetilanteet ovat näiden vuoksi pyritty käsittelemään erikseen ohjelmakoodissa, jotta voidaan välttyä vakavilta virheiltä ja virheeseen johtaneet jäljet voidaan kirjata lokiin. Virheiden käsittely sivuaa myös hieman jo käsiteltyä käytettävyyttä, sillä palvelun käytön estävien virhetilanteiden tapahtuessa on hyvä esittää käyttäjälle jokin virheilmoitus ja esimerkiksi kehottaa yrittämään hetken kuluttua uudelleen.

6.2 Suorituskyky ja sen toteutus

Toteutetun ohjelmiston suorituskyky on myös varsin suuressa roolissa sen laadun ja onnistumisen kannalta, ja erityisesti suorituskyvyn kestäminen myös suuremmilla käyttäjämäärillä on hyvin tärkeää, joka taas koskee ohjelmiston skaalautuvuutta. Toisin sanoen palvelun tulee olla myös tarpeeksi hyvin skaalautuva. Suorituskyky on pyritty huomioimaan mahdollisimman laajasti ympäri koko maksupalvelun, lähtien teknologiavalinnoista aina ohjelmakoodissa oleviin hyvien käytäntöjen noudattamiseen. Suorituskykyyn vaikuttavia teknologiavalintoja on käyty tarkemmin läpi tutkielman luvussa 2.3.

6.2.1 Singletonit

Ohjelmakoodissa on pyritty käyttämään niin sanottuja singletoneja siten, että niiden käytöllä on suorituskykyä parantava vaikutus. Singletoni on olio-ohjelmointikielissä sellainen ohjelmistossa olevan luokan olion staattinen instanssi, joka luodaan vain kerran, ja käytetään tätä samaa instanssia koko suorituksen ajan. Singletonit eivät kuitenkaan sovi joka

paikkaan, ja väärin käytettyinä voivat olla jopa antisuunnittelumalleja, eli ovat vastoin hyväksi todettuja käytäntöjä ohjelmakoodissa. [42]

Tässä työssä singletoneja käyttämällä pyrittiin välttämään varsin yleistä huonoa käytäntöä, jossa ohjelmassa saatetaan päätyä luomaan valtavia määriä turhia kaksoiskappaleita käyttämällä aina tiettyä luokkaa tarvittaessa luokan olioinstanssin luovaa new-avainsanaa. Jos turhia kaksoiskappaleita on tarpeeksi paljon, alkaa se vaikuttamaan negatiivisesti ohjelmiston suorituskykyyn. Esimerkiksi tässä työssä käytetään C#-kielen HttpClient-luokasta yhtä ainutta singleton-instanssia, sillä tätä luokkaa käytetään HTTP-kyselyiden tekoon esimerkiksi pankkien rajapinnoille, ja kyselyitä tapahtuu tyypillisesti useampi maksun suorituksen aikana.

6.2.2 Asynkroniset kutsut

Erityisesti edellä mainittuihin HTTP-kyselyihin liittyy myös kyselyiden asynkroninen suoritustapa. Asynkronisella suorituksella tarkoitetaan yksinkertaistettuna sitä, että suoritus tapahtuu ikään kuin ohjelman taustalla ja kestää ajan, joka on suorituksen alussa vielä tuntematon. Asynkronisen kutsun aikana ohjelman muu suoritus jatkuu, eikä jää odottamaan kutsun valmistumista, ellei niin erikseen määritellä. Kuitenkin joidenkin asynkronisten HTTP-kyselyjen tulosta täytyy odottaa, jotta kutsuttu osapuoli ehtii palauttaa jonkin vastauksen. Tällöin ohjelma vastaanottaa takaiskutsufunktion (engl. callback) ja voi käsitellä palvelimelta vastaanotetun vastauksen ja mahdollisesti saadun datan. [43]. Esimerkki tästä on tässä työssä loppuasiakkaan näkemien maksuvaihtoehtojen haku.

Asynkroniset kutsut ovat osaltaan välttämättömiä, kuten HTTP-kyselyiden suorittamisessa, mutta asynkronisia kutsuja voidaan suorittaa myös, jos halutaan jonkin tapahtuvan ohjelmassa taustalla, samalla kun käyttäjä tekee jotain muuta. Työssä näitä kutsuja on käytetty muun muassa lokien kirjaamisessa ja maksutietojen tallennuksissa tietokantaan.

Esimerkiksi lokitietoja kirjataan neljässä kohtaa yhden maksusuorituksen prosessin aikana ja erilaisia maksutietoja kirjoitetaan kaksi kertaa. Suorituskykyä hidastaisi huo-

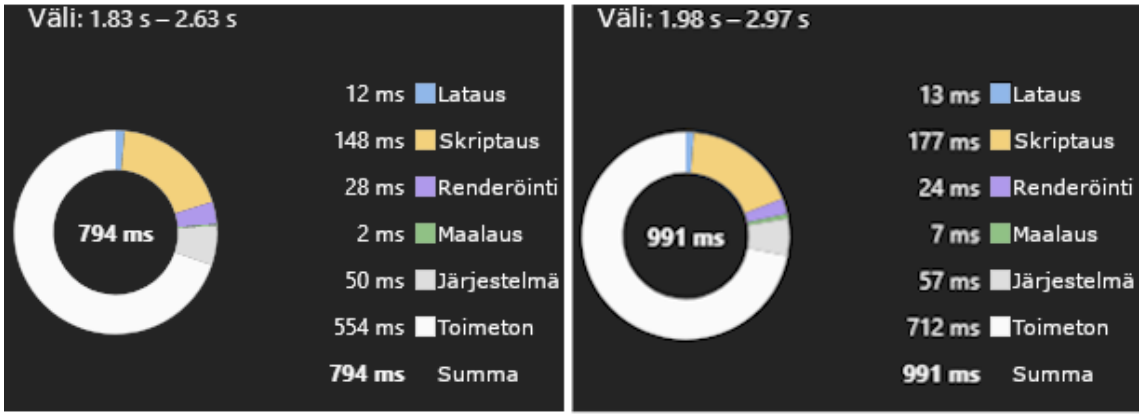
mattavasti, jos tietokantaan kirjoittamisen valmistumista jäätäisiin jokaisessa kohtaa odottamaan, ennen kuin suoritusta jatkettaisiin eteenpäin. Nämä kutsut pystytään jättämään taustaprosesseiksi, sillä kumpienkaan sisältämiä tietoja ei tarvitse hakea tietokannasta välittömästi, jotta ohjelma voitaisiin suorittaa onnistuneesti loppuun. Kuitenkin sekä lokiettä maksutietoja tullaan tarvitsemaan myöhemmin, jolloin ne haetaan määrätyistä tietokannoista. Tällöin voidaan olettaa, että tiedot on jo kirjattu tietokantaan, sillä esimerkiksi päivän aikana tulleet maksut, jotka siirretään eteenpäin kauppiaalle, tapahtuu vain kerran päivässä sen päätteeksi.

6.2.3 Suorituskyvyn mittaus

Maksupalvelun suorituskyky näkyy käyttäjälle pääasiassa sivulatauksien keston perusteella, oli kyseessä sitten loppuasiakas tai kauppias. Palvelun loppuasiakkaan näkemän maksunäkymän täysi lataaminen testidatan kanssa verkkoselaimella tapahtuu melko nopeasti, keskimäärin noin yhden sekunnin ajassa, jos sivun välimuisti on tyhjennetty. Testausten perusteella myös muut operaatiot, kuten maksupalvelun valinnan jälkeinen tunnistautumissivulle ohjaaminen ja onnistuneen maksun jälkeinen tietojen kirjaus, olivat keskimäärin noin sekunnin mittaisia. Palvelulle ei ole vielä määritelty tarkkoja vaatimuksia vasteajoille, mutta noin sekunnin suoritusaika kutsuissa on todettu olevan riittävän hyvä tulos, ja tarkoitus on pitää vasteajat jatkossakin tällä tasolla.

Molempien testien suoritukset on tehty manuaalisesti. Sivunlataustestin ajanotossa käytettiin hyväksi Google Chrome -selaimen suorituskyvyn kehittäjätyökalua, ja varsinaisissa palvelinkutsujen testeissä käytettiin C#-kielen DateTimeOffset-tietueesta (engl. struct) saatavan Unix-aikaleiman arvoja aikaeron mittaukseen.

Kuvassa 6.3 näkyy kaksi eri mittausta maksusivun latausajasta. Vasemmanpuoleisessa kuvassa on mitattu aika joka kuluu siihen, että asiakassovellus kutsuu palvelinta ja saa tältä vastauksen. Oikeanpuoleisessa kuvassa sen sijaan on otettu tähän aikaan mukaan myös sivun tietojen ja maksuvaihtoehtojen käsittely ja lopullinen renderöinti.



Kuva 6.3: Tyypilliset maksusivun latausajat eri pisteisiin saakka

Kuvassa 6.4 nähdään sadan peräkkäisen testimaksun perusteella tehtyjen mittausten tulokset. Kuvassa näkyy ensin muutamia yksittäisiä mittaustuloksia esimerkkinä, ja lopuksi keskiarvot tuloksista. Jokaisella kierroksella on mitattu kaksi eri arvoa: maksupalvelun valinnan jälkeinen tunnistautumissivulle ohjaaminen ja onnistuneen maksun jälkeinen tietojen kirjaus.

```
[TUNNISTAUTUMISEEN ---- 1411 ms]
[TUNNIST. JALKEEN ---- 1040 ms]
================================
[TUNNISTAUTUMISEEN ---- 914 ms]
[TUNNIST. JALKEEN ---- 1266 ms]
================================
[TUNNISTAUTUMISEEN ---- 908 ms]
[TUNNIST. JALKEEN ---- 992 ms]
================================
[TUNNISTAUTUMISEEN ---- 914 ms]
[TUNNIST. JALKEEN ---- 1205 ms]
================================
[TUNNISTAUTUMISEEN ---- 1164 ms]
[TUNNIST. JALKEEN  ---- 989 ms]
================================
[TUNNISTAUTUMISEEN ---- 1170 ms]
[TUNNIST. JALKEEN ---- 1460 ms]
```

```
[TUNNISTAUTUMISEEN ---- 931 ms]
[TUNNIST. JALKEEN ---- 1013 ms]
================================
[TUNNISTAUTUMISEEN ---- 912 ms]
[TUNNIST. JALKEEN ---- 1031 ms]
================================
[TUNNISTAUTUMISEEN ---- 858 ms]
[TUNNIST. JALKEEN ---- 1351 ms]
================================
[TUNNISTAUTUMISEEN ---- 1774 ms]
[TUNNIST. JALKEEN ---- 1152 ms]


KESKIARVOT:
[TUNNISTAUTUMISEEN ---- 866 ms]
[TUNNIST. JALKEEN ---- 1186 ms]
```

Kuva 6.4: Esimerkki suorituskyvyn mittaustuloksista ja lasketuista keskiarvoista

6.3 Muunneltavuus ja jatkokehitys

Kehitetty maksupalvelu on tällä hetkellä vielä prototyypin vaiheessa, ja tulee täten luonnollisesti vielä muuttumaan ja kehittymään huomattavasti ajan myötä. Palvelu pitäisi myös pystyä integroimaan verkkokauppoihin, kuten Ajas-ajanvarauspalvelun omaan verkkokauppaan ja myöhemmin muihin, ulkopuolisiin verkkokauppoihin. Tämän vuoksi palvelun kehityksessä on otettu huomioon sen muunneltavuus, joka edesauttaa palvelukokonaisuuden helpompaa jatkokehitystä.

Muunneltavuutta on pyritty saavuttamaan useilla ratkaisuilla, joista yksi tärkeimmistä lienee koko järjestelmän hajautettu rakenne. Esimerkiksi palvelun asiakaspuolen sovellus olisi helppo korvata kokonaan uudella sovelluksella, mahdollisesti eri tekniikoilla toteutettuna. Tämä johtuu siitä, että palvelin on irrallaan asiakaspuolesta ja asiakaspuolen ainoa tehtävä on kutsua palvelimen rajapintoja. Kuitenkin palvelua voitaisiin edelleen hajauttaa erillisiin, itsenäisiin sovelluksiinsa, jolloin arkkitehtuurityyppi olisi lähempänä mikropalveluarkkitehtuuria. Tällöin voisi uudistustarpeiden sattuessa riittää entistä pienempien kokonaisuuksien uudistaminen, jolloin kehitys olisi helpommin hallittavaa ja testattavaa.

Toinen muunneltavuutta edistävä ominaisuus työssä on tarvittavien muuttujien arvojen kovakoodauksen välttäminen. Sen sijaan ohjelmistossa on pyritty käyttämään mahdollisimman usein erillistä JSON-tyyppistä määrittelytiedostoa, ja tätä kautta hakemaan eri muuttujille niiden arvot. Nämä määrittelyt olisi kuitenkin tarkoitus siirtää oman rajapintansa taakse, josta tiedot voidaan hakea aina tarvittaessa. Muutoin näitä tietoja käytettäisiin samalla tavalla kuin nykyisessäkin määrittelytiedostossa.

6.4 Tietoturva

Maksupalveluissa on erittäin tärkeää ottaa koko palvelun tietoturva tarkasti huomioon jokaisesta näkökulmasta. Osittain tietoturvan vaatimukset tulevat aina Euroopan Unionin

säädösten ja direktiivien tasolta, ja osa taas palvelua kehittävän yrityksen sisäisistä tietoturvavaatimuksista.

Maksupalveluja koskevien lakien ja maksualaa sääntelevien direktiivien määräämien tietoturvatoimenpiteiden lisäksi tietoturvaa on pyritty parantamaan myös palvelun sisäisillä keinoilla. Palvelussa käytetään kaikissa verkon yli kulkevissa yhteyksissä TLS-salausprotokollaa, jonka avulla yhteys on turvallinen, ja ainoastaan määrätty vastaanottaja voi lukea välitetyn kutsun ja sen sisältämän datan. Tämä koskee sekä palvelun eri Docker-konttien välillä, että kolmansien palveluntarjoajien rajapintojen välillä tapahtuvia kyselyitä. Palvelussa käytettäviä erillisiä Docker-kontteja on myös pyritty turvaamaan käyttämällä ohjelmiston sisältämille salaisille tiedoille, kuten API-avaimille ja erinäisille käyttäjätunnuksille, jo aiemmin mainittua Docker-parvien tarjoamaa Docker Secrets -ominaisuutta.

Käyttäjäsertifikaatit tulevat myös olemaan tärkeä osa maksupalvelua. Käyttäjäsertifikaateilla pystytään lisäämään koko järjestelmän turvallisuutta, sillä niitä käyttämällä voidaan myös järjestelmän palvelinrajapinnassa varmentaa sitä kutsuva käyttäjä. Tällöin voidaan tarkempien lokikirjausten lisäksi rajoittaa palvelimen mahdolliset kutsujat niihin, joilla on oikeanlainen sertifikaatti mukana kutsussa. Tällöin esimerkiksi palvelimen rajapintojen kutsut voidaan sallia tulevan ainoastaan maksupalvelun asiakassovelluksen kautta. Tällä tavoin pystytään paremmin rajoittamaan palvelinrajapinnan käyttöoikeudet vaadittuun minimiin, jonka avulla voidaan välttää useita turvallisuusongelmia ja vaikeuttaa esimerkiksi hyökkäysten tekoa asiakassovelluksen ulkopuolelta.

Kaikenlaiset lakisäädökset, kuten tässä työssä melko suuressa roolissa ollut PSD2-direktiivi, ovat vaikuttaneet tämän työn suunnittelussa ja toteutuksessa paljon muun muassa niiden tietoturvakysymyksiin. Esimerkiksi pankkien rajapintayhteyksien toteuttamiseksi tulee maksupalvelun tarjoajalla olla viimeistään tuotantoympäristössä viralliset sertifikaatit, sekä pankkiyhteyksiä varten tunnistautuessa tullaan vuoden 2020 loppuun mennessä vaatimaan kaksivaiheista tunnistautumista kaikissa PSD2-direktiivin alaisissa maissa.

[26] Maksupalveluiden tarjoajia koskevat vaatimukset ovat myös tiukempia kuin ennen, mutta samalla myös yhtenäisempiä EU:n alueella, joka helpottaa maiden välisten toiminnan saavuttamisessa.

6.5 PSD2-direktiivin vaikutukset maksupalveluihin

Käytännön työnä kehitetyn maksupalvelun tukena on käytetty uuden PSD2-direktiivin tuomia uusia ominaisuuksia muun muassa pankkiyhteyksien luomisessa. Yhteyksien luontiin liittyvät tärkeimpinä pankkien rajapintojen uudet toteutustavat, yhteyksiin vaadittavat sertifikaatit ja käyttäjän sekä palveluntarjoajan tunnistautumiset ja valtuutukset. Direktiivillä on ollut maksupalveluiden historian perusteella lähinnä parantavia vaikutuksia sekä tietoturvan että ohjelmistojen kehitettävyyden osalta. Direktiivin mukaisten rajapintojen käyttö ja toteutus, sekä direktiivin noudattaminen, olivat tämän työn kehityksessä monelta osin varsin suoraviivaisia ja rajapinnat vaikuttivat hyvin standardoiduilta.

6.5.1 Erot aiempaan maksupalveludirektiiviin

Toisella maksupalveludirektiivillä ja sen johdannaisilla on useita eri vaikutuksia maksupalveluiden alaan ja tarjoajiin. Yksi PSD2-direktiivin tavoitteista oli luoda EU-alueelle yhtenäinen alue maksamiselle ja maksupalvelun tarjoajille, jota oli yritetty jo ensimmäisen maksupalveludirektiivin kohdalla, mutta ei onnistuttu tarpeeksi hyvin. Syynä alkuperäisen direktiivin puutteellisuuteen oli todennäköisesti se, ettei siinä otettu tarpeeksi hyvin huomioon alalla tapahtuvaa jatkuvaa kehitystä. Tämän vuoksi alkuperäisen PSD-direktiivin jälkeen maksupalveluiden ala oli nopeasti varsin epäjohdonmukainen ja huonosti standardoitu. [44], [45]

Uusi direktiivi myös sisältää aiemman direktiivin tavoin vaatimukset informaation läpinäkyvyydelle, mutta käyttäjien ja palveluiden valtuutuksia koskevien tietojen vaatimusta on kasvatettu, samalla nostaen informaation läpinäkyvyyden tasoa määrittelemällä tar-

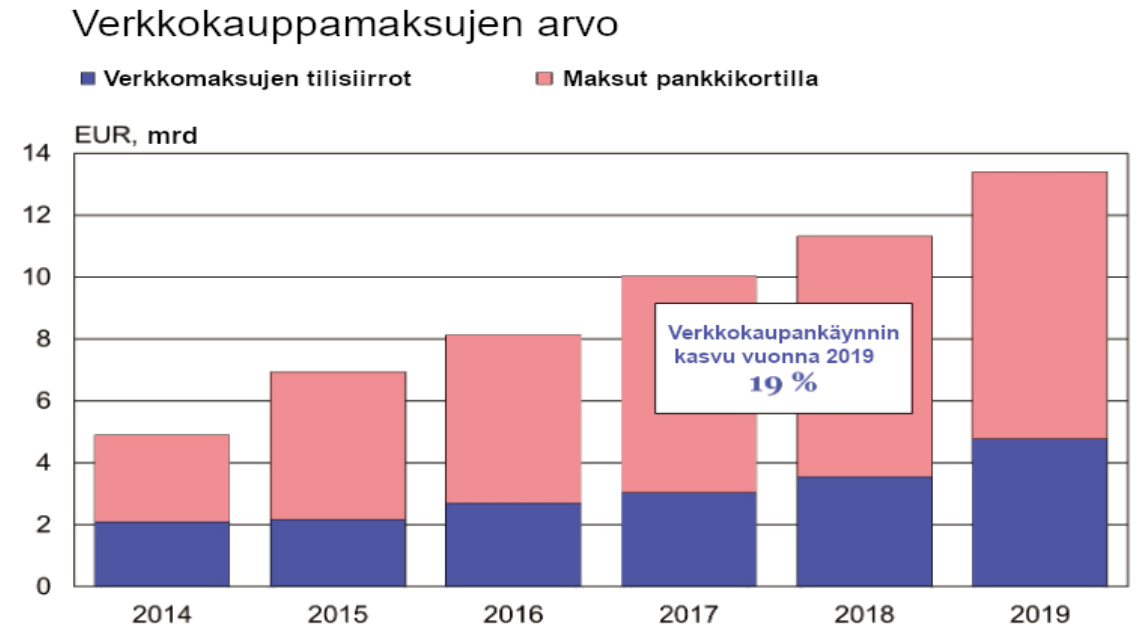
kat viitteet sille, mitä tietoja palveluntarjoajien tulee paljastaa. Maksupalveluiden käyttäjien, kuten loppuasiakkaiden tai kauppiaiden, valtuutus ja rekisteröinti on pidetty voimassa, mutta mukaan on lisätty Euroopan pankkiviranomaisten ylläpitämä rekisteri valtuutetuista palveluntarjoajista. Olennaisena yksityiskohtana myös loppuasiakkaan vahva tunnistautuminen on uutta vanhaan direktiiviin verrattuna, ja tällä pyritään edistämään entuudestaan maksupalveluiden turvallisuutta. [45]

Uusi PSD2-direktiivi on näiden muutostensa perusteella melko pitkälti samanlainen kuin alkuperäinen PSD-direktiivi, mutta omilla parannuksillaan. Uutta direktiiviä kutsutaankin myös uudistetuksi tai korjatuksi maksupalveludirektiiviksi (engl. revised payment service directive) [46].

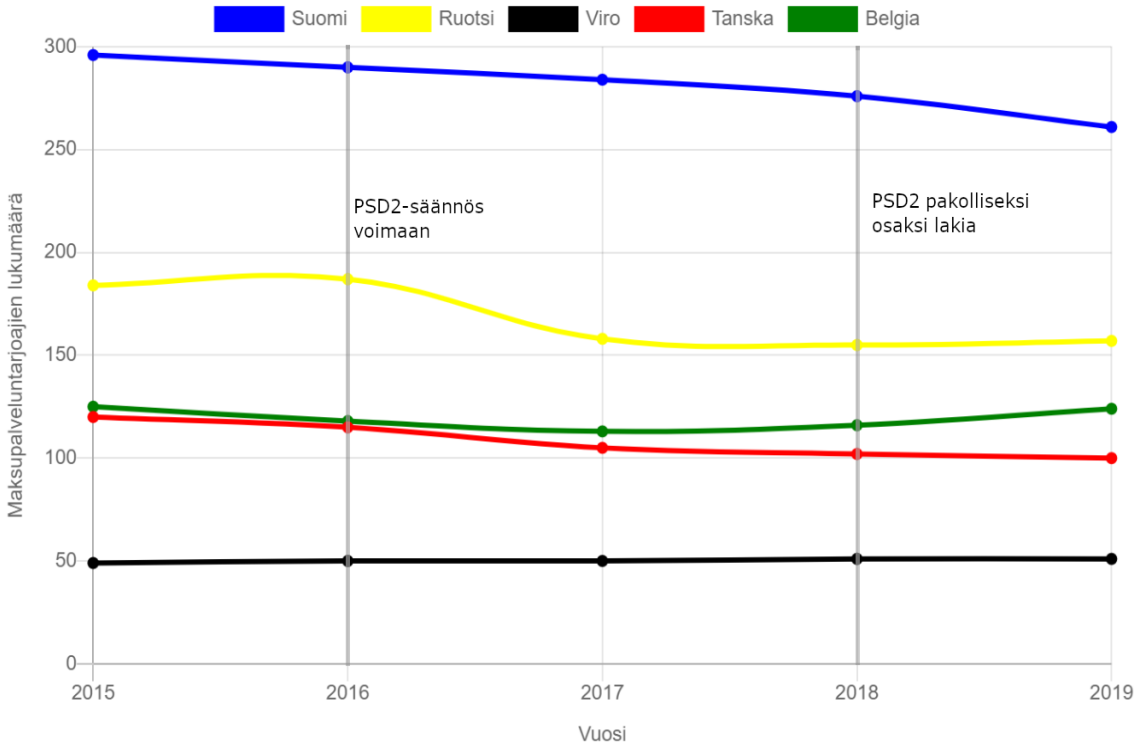
6.5.2 Maksupalveluiden yleistyvyys ja kilpailu

Eräs PSD2-direktiivin tärkeimmistä tavoitteista on kasvattaa kilpailua maksupalveluiden alalla. Tällöin odotettavissa voisi olla myös maksupalveluiden yleistyvyyttä kuluttajien käytettäväksi. Maksupalveluiden ala on nykyään huomattavasti yhtenäisempää EU:n alueella, kuin esimerkiksi vielä ensimmäisen maksupalveludirektiivin aikana. Tällä pitäisi olla maksupalveluiden kehitystä sekä eri maiden välillä tapahtuvaa toimintaa helpottavia vaikutuksia, jolloin palveluita on helpompi kehittää.

Maksupalveluiden yleistymisen syitä voidaan löytää myös PSD2-direktiivin tai muiden säädösten ulkopuolelta. Eräät suurimmista syistä kasvulle ovat yksinkertaisesti palvelujen digitalisaatio sekä teknologian jatkuvasti kiihtyvä kehitys ja uudet innovaatiot, jotka tekevät verkossa maksamisesta yleisempää. Kehitystä voidaan nähdä esimerkiksi Suomen Pankin keräämistä verkkomaksujen tilastoista Suomessa, sillä esimerkiksi vuosien 2014 ja 2019 välillä verkkomaksuissa kulkeva vuosittainen rahasumma on kasvanut arviolta yli kahdeksalla miljardilla eurolla. Tämän lisäksi pelkästään vuosien 2018 ja 2019 välillä havaittiin lähes 19 prosentin kasvu verkkomaksuissa. [47] Kuvan 6.5 kaaviossa esitetään tarkemmin verkkomaksujen kokonaissummien kehitystä vuosien 2014 ja 2019 välillä.



Kuva 6.5: Verkkomaksujen kokonaissummat Suomessa vuosien 2014 ja 2019 välillä [47]



Kuva 6.6: Maksupalveluja tarjoavien säätiöiden lukumäärät [48]

Kuvan 6.6 kaaviossa sen sijaan tarkastellaan maksupalveluiden tarjoajien lukumääriä eri maissa. Kaaviosta 6.5 ja 6.6 voidaan nähdä, että vaikka esimerkiksi Suomessa verkkomaksujen kokonaissummat ovat kasvaneet jatkuvasti vuodesta 2014 lähtien, lukumäärällisesti maksupalveluiden tarjoajat ovat useissa EU-alueen maissa pysyneet joko melko samoina tai jopa laskeneet. Esimerkiksi Suomessa maksupalvelun tarjoajien lukumäärä on vuodesta 2015 lähtien ollut melko tasaisessa laskussa. Näistä esimerkkimaista ainoastaan Belgiassa voidaan nähdä pientä kasvua maksupalvelun tarjoajien lukumäärässä PSD2-direktiivin tultua voimaan.

Kilpailu maksupalveluiden alalla ei ole siis nähtävästi noussut ainakaan kilpailijoiden lukumäärän perusteella, mutta maksupalveluiden käyttö on muuten kasvanut huomattavasti. Jatkuvasti kasvava käyttömäärä tällä alalla voisi kuitenkin tarkoittaa myös kasvavaa kilpailua, erityisesti kun maksupalveluiden käyttäjäkunta laajenee, ja verkkomaksuista tulee entistä enemmän normi. Toisaalta alhaisempi kilpailijoiden määrä on Ajas Paymentsin kannalta hyvä asia, ja jos kilpailu tulee esimerkiksi seuraavina vuosina kasvamaan, on Paymentsilla jo todennäköisesti etumatkaa maksupalveluiden keskuudessa.

6.5.3 Direktiivin sisältämät tietoturvariskit

Toinen maksupalveludirektiivi sisältää useita muutoksia aiempaan direktiiviin verrattuna, ja pyrkii luonnollisesti parantamaan maksupalveluiden toimintaa. Kuitenkin kysymykseksi nousee, onko direktiivin tavoitteissa onnistuttu täysin, ja olisiko niissä ollut jotain parannettavaa entistä paremman tuloksen saavuttamiseksi. Olisiko esimerkiksi sen tietoturvaa pystytty edistämään entistä paremmin?

Tällä hetkellä PSD2-direktiivin tuoma vaatimus avata muun muassa pankkien maksurajapintoja kolmansien osapuolten käytettäväksi tuo tarkoitettujen hyötyjen lisäksi myös erityisesti tietoturvaan liittyviä kysymyksiä, sillä pankkiyhteyksien avaaminen pankin ulkopuolisten osapuolten käyttöön luo uuden väylän hyökkäyksille ja väärinkäytölle. Kuitenkin maksupalvelua kehittäessä havaittiin, että esimerkiksi pankkien uusia avattuja raja-

pintoja käyttääkseen maksupalvelua kehittäessä tuli luoda muun muassa oikeanlaiset asiakassertifikaatit ja jokaiselle maksulle uniikit tunnisteet, eli direktiivin tuomia vaatimuksia ei suoranaisesti voinut kiertää. Palvelun kehityksen osalta tietoturvan toteutus myös tapahtui melko yksinkertaisesti seuraamalla esimerkiksi Osuuspankin tarjoamia ohjeiden vaiheita yhteyden muodostamiseksi ja maksujen suorittamiseksi.

Tästä huolimatta, esimerkiksi pelkästään jo pankin asiakkaiden asiakastiedot voivat olla järjestelmään hyökkääjien mielenkiinnon kohteena. Esimerkiksi asiakkaan ostostietoja ja taloudellisia tuloja voidaan käyttää hyväksi. Myös asiakkaan fyysiseen sijaintiin liittyviä tietoja voidaan nyt helpommin saada selville lähes reaaliajassa, esimerkiksi tarkastelemalla pankkikortilla tehtyjen maksujen saajia ja selvittämällä niiden fyysiset sijainnit. [49] Pelkästään jo näitä tietoja voidaan käyttää hyväksi rikollisen toiminnan helpottamiseksi tai käyttää tietoja muutoin vääriin tarkoituksiin.

Rikolliset toimijat voivat myös esimerkiksi perustaa FinTech-yrityksen (finanssiteknologiayritys) ja hakea Suomen tapauksessa toimilupaa Finanssivalvonnalta. Jos toimiluvan saanti onnistuu, yrityksellä on pääsy sertifikaatteihin ja lupiin käsitellä heidän palveluaan käyttävien asiakkaiden pankkitietoja jopa kahden vuoden ajan [49]. Asiakkaan tulee toki ensin antaa lupa tietojensa käyttöön erikseen tunnistautumalla hänen pankkinsa verkkosivuilla, mutta yrityksen toiminta voidaan naamioida helposti esimerkiksi vastaamaan tavallista maksupalvelua, eikä asiakas voi tietää mitä palvelun taustalla tapahtuu. Tällöin maksupalveluita tarjoavien yritysten valtuutuksista vastaaville toimijoille jää suuri vastuu siitä, että valtuutettavat osapuolet ovat varmasti luotettavia.

Luonnollisesti myös hyökkäykset maksupalveluja ja niiden tarjoajia kohtaan, kuten tässä projektissa kehitettyä Ajas Payments -palvelua, ovat täysin mahdollisia. Tämän vuoksi varsinaisten pankkien tuomien tietoturvatoimien lisäksi on erittäin tärkeää pitää myös oma maksupalvelujärjestelmä sisäisesti turvallisena ja vakaana, jotta hyökkäyksiä voidaan vaikeuttaa ja minimoida mahdolliset vahingot.

7 Yhteenveto

Maksupalveluihin ja niiden tarjoajiin liittyy paljon erilaisia lakeja, ja palveluiden toiminta on tarkkaan säädelty sekä virallisia lupia ja sertifikaatteja vaativa kokonaisuus. Verkkomaksamisen yleistyessä myös lait ovat muuttuneet, pyrkimyksenään vastata nykyisiä ja tulevia alan tarpeita ja vaatimuksia. Yksi viimeisimmistä tärkeistä säädöksistä on Euroopan Unionin toinen maksupalveludirektiivi (PSD2), jolla pyritään muun muassa avaamaan pankkien toimintaa erilaisten rajapintojen kautta myös kolmansien osapuolten käyttöön. Direktiivillä on myös pyritty saamaan aikaseksi muun muassa turvallisempi ja yhtenäisempi ympäristö verkkomaksamiselle esimerkiksi erilaisilla maksupalveluiden käyttöön vaadituilla sertifikaateilla ja käyttäjien parannetuilla tunnistautumistavoilla.

Tämän diplomityön tutkimuskysymyksenä oli tutkia PSD2-direktiivin tuomia vaikutuksia web-pohjaisten maksupalveluiden kehittämiseen eri näkökulmista, sekä miten tämän direktiivin tavoittelema maksupalveluiden kilpailun edistys on todellisuudessa onnistunut. Työssä tutkittiin toisen maksupalveludirektiivin vaikutuksia maksupalveluiden kehitykseen ja tämän kautta maksupalveluiden käyttöön ja tulevaisuuteen. Tutkittavana oli myös maksupalveluiden määrän kehitys ja uuden direktiivin mahdollinen vaikutus siihen, sekä direktiivin tuomien tietoturvariskien arviointi. Työssä käsiteltiin ja arvioitiin myös käytännön työnä kehitetyn maksupalvelun toteutusta, sen arkkitehtuuria ja sitä, miten PSD2-direktiivi näkyi ohjelmiston kehityksessä.

Eräs toisen maksupalveludirektiivin tavoitteista on ollut standardisoida esimerkiksi pankkien rajapintojen toteutus, sekä tätä kautta maksupalveluissa käytettävien tunnis-

tautumisten prosessi. Aiemmin pankeilla on tyypillisesti ollut jotkin omat vaaditut askeleet integraation toteuttamiseksi, joka on tehnyt pankkiyhteyksien luomisesta selvästi työläämpää maksupalveluiden kehittäjille.

Työn luvussa 6.5 saatiin selville, että vaikka PSD2-direktiivin yksi tavoite oli lisätä kilpailua maksupalveluiden alalla, nähtiin Euroopan keskuspankin tarjoamasta taulukosta, että maksupalvelun tarjoajien lukumäärissä ei olla nähty viime vuosien aikana eri EU-alueen maissa juuri ollenkaan kasvua. Sen sijaan, useissa näissä maissa trendi on ollut laskeva, ja esimerkiksi Suomessa maksupalvelun tarjoajien määrä on vähentynyt tasaista vauhtia viimeisten noin viiden vuoden aikana. Maksupalveluiden kilpailu ei siis ole ainakaan palveluntarjoajien lukumäärän perusteella kasvanut, vaikka tämän oletettiin olevan direktiivin seuraus.

Samassa luvussa 6.5 havaittiin tietoturvan osalta, että esimerkiksi pankkien tarjoamien PSD2-rajapintojen käyttöönotto vaatii varsin korkean turvallisuuden takaavia toimia kehittäjältä ja maksupalvelua tarjoavalta yritykseltä. Ilman näitä toimia pankkiyhteyksien käyttö ei olisi mahdollista, eli toimia ei ainakaan helpolla pystytä ohittamaan. Kuitenkin direktiivin julki avaamat uudet rajapinnat tuovat uusia portteja erilaisille hyökkäyksille tai muutoin tiedon väärinkäytölle. Väärinkäyttöä voidaan myös päästä mahdollisesti tekemään, mikäli rikolliset toimijat perustavat maksupalvelua tarjoavan yrityksen, joka väärin perustein saa toimiluvan ja sertifikaatit asianmukaiselta valtuuttajalta. Tällöin yritys voi saada täyden pääsyn palveluaan käyttävien asiakkaiden tietoihin, joita taas voidaan käyttää väärin erinäisiin tarkoituksiin.

Maksupalvelua kehittäessä havaittiin muun muassa käytännön kautta, että PSD2-direktiivin mukaisten rajapintayhteyksien toteuttaminen on useista vaadituista tietoturvatoimista huolimatta melko suoraviivaista ja helppoa, kunhan seuraa esimerkiksi yhdistettävän pankin tarjoamia ohjeita. Tämän lisäksi maksupalvelun kehitystä helpottaa direktiivin tuoma rajapintojen standardisointi, jolloin rajapintayhteyksien luominen myös muihin pankkeihin pitäisi olla samanlainen prosessi. Kuitenkin pankkien toteuttamista tie-

toturvatoimista huolimatta on maksupalvelun tarjoajalla myös suuri vastuu tietoturvasta, sillä esimerkiksi vääriin käsiin joutuneet sertifikaattitiedot tai vuodot asiakkaan tiedoissa voivat aiheuttaa suuria ongelmia. Näiden lisäksi eri valtuutuksia hallinnoivilla viranomaisilla on suuri vastuu siitä, että valtuutusta hakevat maksupalvelua tarjoavat yritykset ovat luotettavia.

Tässä työssä keskityttiin melko pintapuolisesti PSD2-direktiivin tuomiin vaikutuksiin, ja näistä monet havainnot perustuivat käytännön työn tuomiin henkilökohtaisiin kokemuksiin. Itse tutkimus olisi siis voinut olla pidemmälle viety. Työn osalta jäi esimerkiksi selvittämättä, millaisia eroja eri pankeilla on direktiiviä noudattavien rajapintojen toteutuksissa, jolloin direktiivin vaatiman rajapintojen standardisoinnin toteutumista olisi pystytty paremmin arvioimaan.

Vaikka maksupalveluiden tarjoajien välinen kilpailu ei ole merkittävästi kasvanut, tullaan tulevaisuudessa mitä luultavimmin olemaan aina vain enemmän riippuvaisia maksupalveluista maksamisen siirtyessä entistä enemmän verkkoon internetin ja teknologian kiihtyvän kehityksen myötä. Todennäköisen maksupalveluiden käytön kasvun myötä myös PSD2-direktiivin edistämät pankkien rajapintojen avaamiset ja standardisoinnit, sekä direktiivin tietoturvan tason vaatimukset tulevat olemaan varmasti myös tulevaisuudessa hyvin tärkeässä roolissa. Teknologian ja maksupalveluiden kehityksen myötä saatetaan aikanaan kuitenkin havaita uusia haasteita, joihin PSD2-direktiivi ei enää pysty vastaamaan. Tällöin maksupalveluihin liittyviä direktiivejä ja lainsäädäntöjä tulee korjata vastaamaan sen ajan tarpeita.

Lähdeluettelo

[1] T. Arminen, ”Toisen maksupalveludirektiivin (PSD2) vaikutukset kuluttajille”, 2018. url: https://www.theseus.fi/bitstream/handle/10024/155405/Tuuli_Arminen.pdf.

[2] M. G. Christiansen, S. N. Delcambre, E. Demirors, O. Demirors ja M. M. Tanik, ”Software development with transformable components”, teoksessa *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, vol. ii, 1992, 558–559 vol.2. DOI: 10.1109/HICSS.1992.183302.

[3] K. J. Sullivan, W. G. Griswold, Y. Cai ja B. Hallen, ”The structure and value of modularity in software design”, *ACM SIGSOFT Software Engineering Notes*, vol. 26, nro 5, s. 99–108, 2001.

[4] A. W. Services. (2020). ”Benefits at a Glance”, url: https://aws.amazon.com/application-hosting/benefits/.

[5] V. Davis. (2019). ”Why ASP.Net Core is the best choice to build enterprise web applications [Interview]”, url: https://hub.packtpub.com/why-asp-net-core-is-the-best-choice-to-build-enterprise-web-applications-interview/.

[6] K. Bounnady, K. Phanthavong, S. Pathoumvanh ja K. Sihalath, ”Comparison the processing speed between PHP and ASP. NET”, teoksessa *2016 13th International Conference on Electrical Engineering/Electronics, Computer,*

Telecommunications and Information Technology (ECTI-CON), IEEE, 2016, s. 1–5.

[7] Microsoft. (2018). ”Overview of ASP.NET Core Security”, url: https://docs.microsoft.com/en-us/aspnet/core/security/?view=aspnetcore-3.1.

[8] A. Mishra, ”Critical Comparison of PHP and ASP .NET for Web Development”, *International Journal of Scientific & Technology Research*, vol. 3, nro 7, s. 331–333, 2014.

[9] S. Sagayaraj ja M. S. Kumar, ”Performance Evaluation of Web Services in C#, JAVA, and PHP”, *International Journal of Computer Science and Business Informatics*, vol. 7, nro 1, 2013.

[10] Microsoft. (2018). ”Introduction to Containers and Docker”, url: https://docs.microsoft.com/en-us/dotnet/architecture/microservices/container-docker-introduction/.

[11] IBM. (2019). ”NoSQL Databases”, url: https://www.ibm.com/cloud/learn/nosql-databases.

[12] C. J. Date, *SQL and relational theory: how to write accurate SQL code*. "O’Reilly Media, Inc.", 2011.

[13] IBM. (2019). ”Relational Databases”, url: https://www.ibm.com/cloud/learn/relational-databases.

[14] Finanssivalvonta. (2018). ”Maksupalvelun tarjoajat”, url: https://www.finanssivalvonta.fi/pankki/maksupalvelun-tarjoajat/.

[15] Finlex. (2010). ”Maksulaitoslaki”, url: https://finlex.fi/fi/laki/ajantasa/2010/20100297.

[16] E. Commission. (2020). ”Payment services”, url: https://ec.europa.eu/info/business-economy-euro/banking-and-finance/consumer-finance-and-payments/payment-services/payment-services_en.

[17] ——, (2018). ”Single euro payments are (SEPA)”, url: https://ec.europa.eu/info/business-economy-euro/banking-and-finance/consumer-finance-and-payments/payment-services/single-euro-payments-area-sepa_en.

[18] Finanssivalvonta. (2018). ”Mitä ovat maksupalvelut?”, url: https://www.finanssivalvonta.fi/kuluttajansuoja/kysymyksia-ja-vastauksia/maksupalvelut/.

[19] E. Commission. (2016). ”Payment services in the EU”, url: https://eur-lex.europa.eu/legal-content/EN/LSU/?uri=CELEX:32007L0064.

[20] I. Romānova, S. Grima, J. Spiteri ja M. Kudinska, ”The Payment Services Directive 2 and competitiveness: the perspective of European Fintech companies”, *European Research Studies Journal*, vol. 21, nro 2, s. 5–24, 2018.

[21] J. Reijers, B. Jacobs ja I. E. Poll, ”Payment Service Directive 2”, tohtorinväitöskirja, Thesis for the Degree of Master of Science in Information Sciences at the . . ., 2016.

[22] A. Brener, ”Payment Service Directive II and Its Implications”, teoksessa *Disrupting Finance*, Palgrave Pivot, Cham, 2019, s. 103–119.

[23] Finanssivalvonta. (2019). ”Toinen maksupalveludirektiivi – Payment Services Directive, PSD2”, url: https://www.finanssivalvonta.fi/saantely/saantelykokonaisuudet/psd2/.

[24] P. Wolters ja B. Jacobs, ”The security of access to accounts under the PSD2”, *Computer law & security review*, vol. 35, nro 1, s. 29–41, 2019.

[25] P. Office. (2019). ”Revised rules for payment services in the EU”, url: https://eur-lex.europa.eu/legal-content/EN/LSU/?uri=CELEX:32015L2366.

[26] Stripe. (2020). ”Strong Customer Authentication (SCA) enforcement date”, url: https://support.stripe.com/questions/strong-customer-authentication-sca-enforcement-date.

[27] ——, (2020). ”Strong Customer Authentication - What internet businesses need to know about the new European regulation”, url: https://stripe.com/en-fi/guides/strong-customer-authentication.

[28] ——, (2019). ”3D Secure 2 - A new authentication standard”, url: https://stripe.com/en-fi/guides/3d-secure-2.

[29] B. Jones. (2018). ”Are eIDAS certificates the answer to PSD2 open banking TPP verification?”, url: https://www.fintechfutures.com/2018/06/are-eidas-certificates-the-answer-to-psd2-open-banking-tpp-verification/.

[30] S. P. Kadam ja S. Joshi, ”Secure by design approach to improve security of object oriented software”, teoksessa *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, 2015, s. 24–30.

[31] E. Rescorla. (2000). ”HTTP Over TLS”, url: https://tools.ietf.org/html/rfc2818.

[32] J. Chen, F. Miao ja Q. Wang,
”SSL/TLS-based Secure Tunnel Gateway System Design and Implementation”,
teoksessa *2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID)*, IEEE, 2007, s. 258–261.

[33] D. Bank. (2020). ”Danske Bank - Authorization to APIs”,
url: `https://developers.danskebank.com/documentation`.

[34] V. Rajakannas. (2020). ”Workflow for PSD2 Payments API”,
url: `https://op-developer.fi/p/paymentauthorizationflow`.

[35] JWT.io. (2020). ”Introduction to JSON Web Tokens”,
url: `https://jwt.io/introduction/`.

[36] D. documentation. (2020). ”Manage sensitive data with Docker secrets”,
url: `https://docs.docker.com/engine/swarm/secrets/`.

[37] Osuuspankki. (2020). ”OP Corporate Payment API (1.0)”,
url: `https://op-developer.fi/docs/api/3J0IyOCOAoE7Y18DRUYJeI/OP%20Corporate%20Payment%20API`.

[38] A. W. Services. (2020). ”AWS Cloud Security”,
url: `https://aws.amazon.com/security/`.

[39] H. Röder, ”Specifying usability features with patterns and templates”,
teoksessa *2012 First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE)*, IEEE, 2012, s. 6–11.

[40] K. Tokuno ja S. Yamada, ”Software availability theory and its applications”,
teoksessa *Handbook of Reliability Engineering*, Springer, 2003, s. 235–244.

[41] L. Atchison,
Architecting for Scale: High Availability for Your Growing Applications.
"O’Reilly Media, Inc.", 2016.

[42] F. A. Fontana, J. Dietrich, B. Walter, A. Yamashita ja M. Zanoni, ”Antipattern and Code Smell False Positives: Preliminary Conceptualization and Classification”, teoksessa *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, vol. 1, 2016, s. 609–613.

[43] MDN. (2019). ”Synchronous and asynchronous requests”, url: https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Synchronous_and_Asynchronous_Requests.

[44] C. Westermeier, ”Money is data - the platformization of financial transactions”, *Information, Communication & Society*, s. 1–17, 2020.

[45] P. Valcke, N. Vandezande ja N. Van De Velde, ”The evolution of third party payment providers and cryptocurrencies under the EU’s upcoming PSD2 and AMLD4”, 2015.

[46] E. C. Bank. (2018). ”The revised Payment Services Directive (PSD2) and the transition to stronger payments security”, url: https://www.ecb.europa.eu/paym/intro/mip-online/2018/html/1803_revisedpsd.en.html.

[47] S. Pankki. (2020). ”Payments statistics”, url: https://www.suomenpankki.fi/en/Statistics/payments-statistics/payments-statistics/.

[48] S. D. Warehouse. (2020). ”Institutions offering payment services to non-MFIs”, url: https://sdw.ecb.europa.eu/reports.do?node=1000001384.

[49] F. Hacquebord, R. McArdle, F. Mercês ja D. Sancho, ”Ready or Not for PSD2”, 2019. url: https://documents.trendmicro.com/assets/white_papers/wp-PSD2-The-Risks-of-Open-Banking.pdf.