



**TURUN
YLIOPISTO**

Matkustusoikeuden offline-tarkastus tunnistepohjaisessa lippujärjestelmässä

Ohjelmistotekniikka
Tietotekniikan tutkinto-ohjelma
Tietotekniikan laitos, Teknillinen tiedekunta
Diplomityö

Laatija:
Mikael Janhonen

Ohjaajat:
Lauri Koivunen
Ville Leppänen

Kesäkuu 2021

Diplomityö
Tietotekniikan laitos, Teknillinen tiedekunta
Turun yliopisto

Oppiaine: Ohjelmistotekniikka

Tutkinto-ohjelma: Tietotekniikka

Tekijä: Mikael Janhonen

Otsikko: Matkustusoikeuden offline-tarkastus tunnistepohjaisessa lippujärjestelmässä

Sivumäärä: 68 sivua, 2 liitesivua

Päivämäärä: Kesäkuu 2021

Suomen joukkoliikenteessä käytettävät lippu- ja maksujärjestelmät ovat siirtymässä tunnistepohjaisiin järjestelmiin, joissa matkustajan omistamia lippuja säilytetään älykorttien sijaan taustajärjestelmässä olevilla matkatileillä. Muutoksen myötä matkustusoikeuden tarkastukseen tarvitaan taustajärjestelmän tilaa, ja tarkastusta varten suoritetaan yleensä rajapintakutsu taustajärjestelmään. Tarkastuksen suorittavat laitteet ovat usein kuitenkin mobiilitiedonsiirtoyhteyksien päässä, eikä jatkuva yhteys taustajärjestelmään ole taattu, mikä saattaa johtaa matkustusoikeuden tarkastuksen estymiseen ja sitä myötä mahdollisiin lipputulotappioihin ja huonoon käyttökokemukseen. Järjestelmässä on kuitenkin mahdollista toteuttaa matkustusoikeudelle offline-tarkastus, jossa matkustusoikeus tarkastetaan paikallisesti päätelaitteella.

Työssä tutkittiin matkustusoikeuden offline-tarkastuksen tarpeellisuutta sekä toteutuksen mahdollisuuksia ja haasteita. Tarpeellisuutta tutkittiin lähinnä nykyisen rajapintakutsuihin perustuvan online-tarkastuksen teknisen toimivuuden kannalta. Tutkimus tapahtui toimialan kirjallisuuden ja Pusatec Oy -yrityksen tunnistepohjaisen lippujärjestelmälustan offline-toiminnallisuutta käsittelevien menetelmien avulla. Työn aikana ajoneuvolaitteiden verkkoyhteyksiä ja online-tarkastuksen toimintaa tutkittiin analysoimalla lippujärjestelmälustan ja ajoneuvolaitteiden tallentamaa lokimateriaalia. Työn aikana muodostettiin myös offline-tarkastuksen osittainen prototyyppitoteutus, jonka avulla toteutukseen liittyviä haasteita pyrittiin selvittämään.

Työssä havaittiin, että online-tarkastus saattaa toimia alueesta riippuen riittävän hyvin, eikä offline-tarkastuksen toteutus ole välttämättä kriittistä järjestelmän toiminnan kannalta. Toteutuksella voidaan kuitenkin ehkäistä pelkkään online-tarkastukseen liittyvää lipputuloriskiä ja parantaa käyttökokemusta nopeuttamalla leimaustapahtumaa. Offline-tarkastuksen toteutuksessa päätelaitteille etukäteen replikoitavat tiedot havaittiin käyttökelpoiseksi ja parhaaksi toteutusmalliksi, osittain siksi, että toteutus toimii yhtäläisesti kaikille tunnistemedioille, kuten älykortteille ja mobiilisovelluksille. Myös mobiilisovelluksella käytettävien lippujen tunnisteen yhteyteen liitetyt tiedot voivat olla, lippujärjestelmän säännöistä riippuen, hyvä ratkaisu mobiililippujen offline-tarkastukseen.

Asiasanat: tunnistepohjainen lippujärjestelmä, matkustusoikeuden tarkastus, lipuntarkastus, whitelist

Sisällys

1	Johdanto	1
2	Joukkoliikenteen lippu- ja maksujärjestelmät	3
2.1	Lippu- ja maksujärjestelmien toiminta	3
2.2	Lippujärjestelmäarkkitehtuurit	6
2.2.1	Korttipohjainen arkkitehtuuri	6
2.2.2	Tunnistepohjainen arkkitehtuuri	7
2.3	Lippu- ja tunnistemediat	10
2.3.1	Älykortit	10
2.3.2	Mobiililaitteet	10
2.3.3	Paperiliput	13
2.3.4	Lähimaksulliset pankkikortit	14
2.4	Joukkoliikenne Suomessa	15
2.4.1	Julkisen henkilöliikenteen lainsäädäntö	15
2.4.2	Lippujärjestelmien yhteentoimivuus	16
2.4.3	Joukkoliikennejärjestelmät ja matkustajamäärät	18
3	Tunnistepohjainen matkustusoikeuden validointi	19
3.1	Online-validointi	19
3.2	Offline-validointi	20
3.2.1	Validointiin tarvittavat tiedot	21
3.2.2	Älykortteille tallennettavat offline-tiedot	21
3.2.3	Mobiili- ja paperilippujen offline-tiedot	22
3.2.4	Päätelaitteille jaettavat offline-listat	23
3.2.5	Estolistat	25
4	Offline-validointi Pusatecin lippujärjestelmäalustassa	27
4.1	Nykyinen lippujärjestelmäalusta	27
4.2	Toteutuksen alustava suunnitelma	28
4.3	Vaatimukset	28
5	Offline-validoinnin toteutuksen suunnittelu	31
5.1	Ympäristön asettamat rajoitukset	31
5.1.1	Päätelaitteet	31
5.1.2	Tiedonsiirtoyhteydet	32

5.1.3	Taustajärjestelmän suoritusympäristö	37
5.2	Toteutusmallin valinta	38
5.2.1	Offline-validoinnin tarpeellisuus.....	38
5.2.2	Offline-validointiin tarvittavat tiedot.....	39
5.2.3	Mallin valinta	40
5.3	Listojen jakelun suunnittelu.....	42
5.3.1	Listojen muoto ja muodostus	42
5.3.2	Listojen replikaatio.....	43
5.3.3	Ensisijaisten listojen säilytys	46
5.3.4	Tiedon koodaus.....	47
5.3.5	Jakelurajapinta	48
5.3.6	Välimuistissa olevat listat.....	49
5.4	Vaatimusten ratkaisut.....	50
6	Toteutus.....	52
6.1	Listojen konfigurointi	52
6.2	Ensisijaisten listojen muodostus ja säilytys	54
6.3	Listojen jakelu	55
7	Toteutuksen evaluaatio	57
7.1	Järjestelmän oikeellinen toiminta.....	57
7.2	Järjestelmän tehokkuus ja skaalautuvuus	58
7.3	Haasteet ja rajoitukset.....	60
8	Yhteenveto	62
	Lähdeluettelo	65
	Liite 1 - Pusatec PT10 eX ja PT11 eX -ajoneuvolaitteiden tiedot	1
	Liite 2 - Yhteyskatkot Jyväskylän alueella	2

1 Johdanto

Suomessa vuonna 2018 voimaan astunut liikennepalvelulaki (24.5.2017/320) velvoittaa liikkumispalveluita tuottavia toimijoita avaamaan tietojärjestelmien rajapintoja, ja ohjaa toimialan järjestelmiä avoimempaan suuntaan. Rajapintojen avaaminen aiheuttaa uudistusta lippu- ja maksujärjestelmien toteutukselle, sillä useimmat joukkoliikenteen käytössä olleista lippujärjestelmistä ovat aiemmin perustuneet suljettuihin teknisiin ratkaisuihin (Liikennevirasto, 2018). Myös kehittyvä teknologia on keskeisessä osassa ajamassa liikkumispalveluiden uudistumista, ja varsinkin tietoliikenneyhteyksien kattavuuden ja nopeuksien kasvu sekä pilvipohjaisten ratkaisujen yleistymisen ovat tuoneet uudenlaisia mahdollisuuksia lippu- ja maksujärjestelmien toteutukseen.

Yksi Suomen joukkoliikenteessä meneillään oleva muutos onkin pilvipohjaisten ratkaisujen hyödyntäminen matkustajien omistamien lippu- ja maksutietojen säilöntään ja leimaustapahtumien käsittelyyn. Uudenlaisessa tunnistepohjaisessa järjestelmässä matkustajien tiedot ja heidän ostamansa liput säilötään matkustajien hallussa pitämien älykorttien sijaan taustajärjestelmässä säilytettäviin matkatileihin. Matkan yhteydessä matkustaja tunnistautuu matkatiliin liitetyn tunnisteiden avulla, ja matkustusoikeus voidaan tarkastaa matkatiliin säilöttyjen tietojen perusteella rajapintakutsulla.

Joukkoliikenteen käytössä olevat ajoneuvolaitteet ovat suureksi osaksi mobiiliverkkoyhteyksien päässä, ja matkan aloituksen yhteydessä tapahtuvan matkustusoikeuden tarkastuksen, eli validoinnin, luotettava toiminta on erittäin kriittistä järjestelmän toiminnan kannalta. Tämä vuoksi matkustusoikeuden validoinnin luotettavuutta tai nopeutta lisäämään voidaan toteuttaa offline-validointi, jossa matkustusoikeuden validointi tapahtuu ilman leimaushetkellä toimivaa tiedonsiirtoyhteyttä. Perinteiset älykortteihin perustuvat järjestelmät ovat olleet käytössä pitkään ja toimivat luotettavasti. Tunnistepohjaisuuteen siirtyminen on laaja ja haastava järjestelmä uudistus, varsinkin kun tämänhetkiseen laadukkaaseen ja nopeaan käyttökokemukseen on totuttu ja vastaavanlaatuista toimintaa odotetaan myös tulevaisuudessa.

Tämän työn tarkoitus on tutkia matkustusoikeuden offline-validoinnin tarpeellisuutta ja toteutusmahdollisuuksia joukkoliikenteen tunnistepohjaisissa lippujärjestelmissä. Tarpeellisuutta tarkastellaan online-validoinnin tämänhetkisen toimivuuden ja offline-validoinnin tuomien etujen avulla. Erilaisia toteutusmahdollisuuksia puolestaan etsittiin toimialan kirjallisuudesta.

Työssä pyritään vastaamaan seuraaviin tutkimuskysymyksiin.

TK 1. Mikä on offline-validoinnin tarve ja siitä saatavat hyödyt tunnistepohjaisessa lippujärjestelmissä?

TK 2. Miten offline-validointi kannattaa toteuttaa, ja mitä haasteita ja ratkaisuja tähän liittyy?

Työn toimeksiantaja on Pusatec Oy, ja työn tutkimusmenetelmissä hyödynnetään Pusatec Oy:n tunnistepohjaisessa lippujärjestelmälustaa. Pusatec Oy on joukkoliikenteen tietojärjestelmiä kehittävä yritys, joka toimittaa muun muassa lippujärjestelmälustaa ja päätelaiteratkaisuja useisiin Suomessa operoiviin joukkoliikennejärjestelmiin. Yrityksellä on meneillään tunnistepohjaisen lippujärjestelmälustan kehitys, johon halutaan toteuttaa myös offline-validointiin liittyvää toiminnallisuutta. Työssä käytetään konstruktivistista tutkimusotetta, jossa offline-validointiin liittyvää toiminnallisuutta suunnitellaan ja osittain myös toteutetaan osaksi Pusatecin lippujärjestelmälustaa. Toteutuksessa keskitytään erityisesti päätelaitteille jaettavien offline-validointiin tarvittavien tietojen jakelurajapinnan prototyypin toteutukseen, sekä toteutuksen evaluointiin. Offline-validoinnin tarpeellisuutta ja toteutusmahdollisuuksia tutkitaan myös ajoneuvolaitteiden lokimateriaalin, ja siitä johdettavien metriikoiden avulla.

Työn sisältö jakaantuu seuraavasti. Luvussa 2 vertaillaan eri lippujärjestelmäarkkitehtuureita ja sitä, miksi Suomen järjestelmät ovat osittain siirtymässä tunnistepohjaiseen arkkitehtuuriin. Tässä yhteydessä myös esitetään nykyiset yleisimmät käytössä olevat lippumediat. Luvussa 3 esitetään eri keinoja tunnistepohjaisessa leimauksessa suoritettavaan matkustusoikeuden validointiin, erityisesti offline-validoinnin osalta. Luvuissa 4–7 käsitellään offline-validointia Pusatecin tunnistepohjaisen lippujärjestelmässä. Luvussa 4 annetaan taustatietoa Pusatecin lippujärjestelmälustasta ja esitetään offline-validoinnilta haluttua toiminnallisuutta ja sille asetettuja vaatimuksia. Luvussa 5 analysoidaan järjestelmän teknisiä rajoituksia ja toteutusmahdollisuuksia rajoitusten ja vaatimusten asettamissa rajoissa. Luvussa 6 kuvataan toteutettu offline-listojen jakelujärjestelmän prototyyppi, ja luvussa 7 evaluoidaan toteutetun järjestelmän toiminnan oikeellisuutta ja tehokkuutta sekä toteutusvaiheen haasteita. Luvussa 8 esitetään työn yhteenveto ja tulokset.

Työssä käytetään Liikenneviraston julkaiseman ”Henkilöliikenteen palveluiden sanasto” -nimisen oppaan (1/2018) suosittelimia termejä ja niiden merkityksiä. Kyseisen oppaan tarkoitus on yhtenäistää henkilöliikenteen toimijoiden näkemyksiä eri käsitteiden merkityksistä. Vaikka tutkielmassa lippu- ja maksujärjestelmiä käsitellään liikkumispalveluiden, erityisesti Suomen joukkoliikenteen, näkökulmasta, suuri osa sisällöstä soveltuu hyvin myös muihin lippujärjestelmiin.

2 Joukkoliikenteen lippu- ja maksujärjestelmät

Julkinen liikenne on kaikille avointa henkilöliikennettä. Suuri osa julkisesta liikenteestä tapahtuu joukkoliikenteenä, jossa henkilöiden kuljettaminen toteutetaan suuria matkustajamääriä kuljettavilla liikennevälineillä, kuten busseilla tai junilla. Joukkoliikenteen tavoitteena on suurien ihmismäärien sujuva ja tehokas liikkuttaminen, ja hyvin toteutetun joukkoliikenteen avulla kaupunkien liikenteestä voidaan vähentää huomattavasti ympäristö- ja viihtyvyysoongelmia. *Joukkoliikennejärjestelmällä* tarkoitetaan mahdollisesti useasta eri liikkumispalvelusta muodostuvaa palveluverkostoa, jossa on tyypillisesti käytössä yhteisiä lippu- ja tietojärjestelmiä. Esimerkiksi asutuskeskusten yhteyteen on muodostunut omia seudullisia joukkoliikennejärjestelmiä, jotka erottuvat toisistaan brändäyksen ja lippujärjestelmien osalta. Tällaisia ovat esimerkiksi Turun seudun Föli tai Tampereen seudun Nysse.

Liikkumispalvelut perivät yleensä niiden käytöstä veloituksen matkustajalta kustannuksien ainakin osittaiseksi kattamiseksi. Järjestelmän toiminnan sujuvuuden takaamiseksi toimiva ja luotettava lippu- ja maksujärjestelmä on erittäin oleellinen osa joukkoliikennejärjestelmien toimintaa. Lippujärjestelmän tehtävä on määrittää järjestelmässä käytettävät lipputuotteet ja mahdollistaa tehokkaat ja luotettavat matkustusoikeuksien hankinta- ja tarkastustoimet. Maksujärjestelmä puolestaan vastaa lipputuotteiden myyntiin ja matkojen muunlaiseen rahastukseen liittyvästä rahaliikenteestä. Lippu- ja maksujärjestelmät ovat hyvin riippuvaisia toisistaan ja järjestelmien toiminta on osittain päällekkäistä ja rajat saattavat olla häilyviä.

2.1 Lippu- ja maksujärjestelmien toiminta

Joukkoliikenteessä käytetään lippujärjestelmiä matkustusoikeuksien hankinta- ja valvontaprosesseihin. Matkustajat voivat ostaa erilaisia lipputuotteita, jotka oikeuttavat tiettyjen liikkumispalveluiden käyttöön tietyillä säännöillä. Lippujärjestelmä voi palvella samanaikaisesti useita eri tavoilla liikkuvia ihmisryhmiä, kuten säännöllisesti matkustavia työmatkalaisia tai kaupungissa vierailevia turisteja. Moderneissa lippujärjestelmissä suuri osa toiminnoista, kuten lippujen myynti ja tarkastus, on automatisoitu niin, että käyttäjät ovat suoraan vuorovaikutuksessa tietoteknisten päätelaitteiden ja verkko- ja mobiilipalveluiden kanssa. Modernit lippujärjestelmät hyödyntävät ja ovat hyvin riippuvaisia laajoista tietoteknisistä järjestelmistä.

Lippujärjestelmissä käyttäjien matkustusoikeutta kuvataan lippujen avulla. *Lippu* on käyttäjän omistuksessa oleva todiste palvelun maksun suorituksesta, jonka avulla käyttäjän matkustusoikeus voidaan tarkastaa. Tyypillisessä joukkoliikennejärjestelmässä on useita erilaisia *lipputuotteita*, jotka antavat käyttäjille erilaisia oikeuksia liikkumispalveluiden käyttöön. Lipputuotteisiin liittyvät oikeudet määritellään tarkasti

lippujärjestelmän säännöissä, jotta niiden toiminta on selvää kaikille osapuolille. Tyypillisiä tuotetyyppejä ovat kausiliput, jotka oikeuttavat palveluiden käyttöön tietyinä aikavälinä, ja kerta- ja sarjaliput, jotka oikeuttavat tiettyyn määrään matkoja. Usein lippuun liittyy myös *vaihto-oikeus*, joka oikeuttaa sen keston ajan veloituksettomiin vaihto-yhteyksiin.

Pitkän matkan liikenteessä käyttäjien tulee usein varata paikka tietylle matkalle ennen matkaa. Etukäteen hankittavat kertaliput soveltuvat tämänkaltaiseen käyttöön hyvin, sillä ne toimivat samalla maksun sekä paikan varauksen todisteena. Paikalliset ja alueelliset liikkumispalvelut tarjoavat käyttäjille kuitenkin joustavampaa liikkumispalvelua, jossa käyttäjät voivat matkustaa vapaammin ilman etukäteen varattavaa paikkaa. Tällöin lippujärjestelmää voidaan hyödyntää erilaisten liikkumispalvelukokonaisuuksien hankintaan, ja käyttäjille voidaan tarjota eri käyttötapoihin soveltuvia lipputuotteita, jotka oikeuttavat palveluverkoston tietynlaiseen käyttöön.

Liikkumispalveluista voidaan veloittaa myös tapahtuvan käytön perusteella ilman etukäteen hankittavaa lippua. Matkan yhteydessä tapahtuvaan maksuun perustuvasta palvelun käytöstä käytetään usein englanninkielistä ilmaisua ”Pay as you go”. Veloitus voi tapahtua esimerkiksi käteismaksulla, tai muulla maksutavalla. Käyttäjälle voidaan myös myydä palveluissa maksutapana kelpaavaa arvoa etukäteen, josta matkan hinta veloitetaan matkan yhteydessä, mikä esimerkiksi älykorttien avulla toteutettuna nopeuttaa kyytiin nousua. Matkan aikana ostetun matkustusoikeuden osoitukseksi käyttäjä saa yleensä haltuunsa lipun tai muun tosittien, josta matkustusoikeus voidaan tarkastaa koko matkan ajan, myös mahdollisissa vaihtoyhteyksissä. Toinen tapa käyttöön perustuvaan veloitukseen on rekisteröidä käyttäjän tekemät matkat järjestelmään, päätellä niiden hinta myöhemmin taustajärjestelmässä, ja veloittaa se automaattisesti jälkikäteen käyttäjän hyväksymästä maksukanavasta. Käyttöön perustuvaan hinnoitteluun voidaan määrittää erilaisia dynaamisia hinnoittelusääntöjä, joiden avulla käyttäjälle voidaan automaattisesti laskea paras mahdollinen hinta. Tiettyinä aikana tapahtuneeseen käyttöön voidaan esimerkiksi asettaa hintakatto, jonka täytyessä järjestelmä ei veloita käyttäjältä enempää kyseisenä aikavälinä. Käyttöön perustuva hinnoittelu on käyttäjille usein joustavampi ratkaisu, sillä heidän ei tarvitse tietää matkustustarpeitaan etukäteen.

Matkalipun *leimauksella* tarkoitetaan tässä työssä matkan yhteydessä tapahtuvaa toimintoa, jossa käyttäjä asettaa digitaalisen lippumediensa vuorovaikutukseen lukijalaitteen kanssa. Leimauksessa käyttäjän matkustusoikeus tarkastetaan, eli *validoidaan*, ja matka rekisteröidään järjestelmään. Tämän lisäksi lipusta tai maksutavasta riippuen voidaan suorittaa muita toimintoja, kuten lipun voimassaolon aktivoiminen, kertalipun merkkäminen käytetyksi, tai etukäteen ladatusta arvosta tehtävä veloitus. Leimauksen yhteydessä leimauksen tulos, eli tieto matkustusoikeuden kelpoisuudesta ja mahdollisesti muista tiedoista, kuten jäljellä olevasta arvosta tai voimassaoloajasta, ilmoitetaan käyttäjälle. Matkustusoikeuden puuttuessa käyttäjän matkan aloitus voidaan pyrkiä estämään kuljettajan tai automaattisten porttien toimesta. Leimaus voi tapahtua myös kahdessa osassa, matkan aloituksen ja lopetuksen yhteydessä, jolloin järjestelmä tietää automatisoidusti koko matkan pituuden, ilman että pituus tai vyöhykeväli pitää syöttää matkan aloituksen

yhteydessä. Tämänkaltaiseen ”check in – check out” -menettelyyn ei kuitenkaan ole Suomessa nähty edellytyksiä (Kanerva ym. 2020, s. 4), vaan leimaus tapahtuu yleensä ainoastaan matkan aloituksen yhteydessä. Matkustusoikeuden voi tarkastaa myös lipuntarkastaja matkan aikana, jolloin ilman oikeutta matkustavilta henkilöiltä veloitetaan sakkomaksu, ja tämän vuoksi matkustusoikeuden täytyy olla todennettavissa koko matkan ajan.

Tietyissä tapauksissa käyttäjän matkustusoikeutta ei välttämättä validoida ollenkaan matkan aloituksen yhteydessä. Tällöin luotetaan siihen, että lipuntarkastuksen ja sakkomaksun uhka riittää ohjaamaan käyttäjiä matkustusoikeuden hankintaan. Järjestelmän käyttö on tällöin hyvin sujuvaa, sillä erillistä leimausta ei tarvitse suorittaa matkan aloituksen yhteydessä, vaan matkustajat voivat kulkea vapaammin sisään ja ulos ajoneuvoista. Matkaliput voidaan tarkastaa myös esimerkiksi kuljettajan visuaalisella tarkastuksella, ilman erillistä leimausta lukijalaitteen kanssa. Tämä on mahdollista esimerkiksi paperisten tai mobiililippujen osalta, joissa lipun tiedot voidaan esittää visuaalisesti. Ihmisen suorittama visuaalinen lipun validointi ei kuitenkaan välttämättä ole yhtä täsmällinen kuin päätelaitteiden kanssa tehtävä validointi. Tässä työssä tarkastellaan ainoastaan matkan aloituksen yhteydessä tapahtuvaa lippujen validointia päätelaitteiden kanssa.

Matkustusvyöhykkeet ovat alueita, joiden avulla eri pituiset ja eri alueille sijoittuvat matkat voidaan hinnoitella eri tavoilla. Leimauksen matkustusvyöhykkeet voidaan päätellä järjestelmän toimesta automaattisesti, mutta käyttäjä tai kuljettaja voi joutua joskus myös syöttämään ne itse, mikä hidastaa leimaustapahtumaa. Matkustajien täytyy tällöin tietää matkansa vyöhykkeet ennen leimausta, jotta hinnoittelu tapahtuu oikein.

Lippujärjestelmän säännöistä vastaa tyypillisesti joukkoliikennejärjestelmän liiketoimintaa operoiva taho. Lippujärjestelmälustan toimittaja voi tarjota työkaluja tuotteiden sääntöjen määrittämiseen, jotka huomioidaan myynnin ja matkustusoikeuden validoinnin yhteydessä. Tuotteiden sääntöjen määrittämisä kutsutaan tässä työssä tuoteparametreiksi, ja ne voivat sisältää esimerkiksi tuotteiden hintojen, kelpoisuusaikojen ja vyöhykkeiden määrittäykset. Leimauksessa matkustajan käyttämää liikkumispalvelua kuvataan puolestaan leimausparametrien avulla, joilla voidaan määrittää esimerkiksi palvelun tyyppi, leimauksen sijainti, ja matkan vyöhykeväli tai pituus. Nämä leimausparametrit tulevat päätelaitteelta, ja joskus matkustaja tai kuljettaja syöttää osan niistä päätelaitteiden käyttöliittymän avulla.

Liikkumispalveluita tuottavat toimijat voivat hyväksyä samoja lipputuotteita, jolloin puhutaan järjestelmien yhteistoiminnasta ja lippujärjestelmien *yhteentoimivuudesta*. Jos lippujen myynti ja käyttö tapahtuu eri toimijoiden toimesta, lipputulojen jakamista varten voidaan sopimuksista riippuen tarvita saatavien selvittelyprosessi, josta käytetään termiä clearing (Liikennevirasto, 2018, s. 38). Esimerkiksi käyttäjälle ennakoon myydystä arvosta saadut tulot voidaan jakaa eri palveluntarjoajien kesken tapahtuneen käytön perusteella maksamalla liikkumispalvelun suorittaneelle taholle nousukorvaus. Myös kausilippujen ja muiden etukäteen maksettujen lippujen tulot voidaan jakaa eri toimijoiden kesken käyttöön

perustuvilla säännöillä. Lippujärjestelmän lipputulosten clearing-menettelyn, tilastollisen analysoinnin ja järjestelmän muun toiminnan vuoksi on tärkeää, että jokaisesta matkasta välittyy tieto järjestelmään.

2.2 Lippujärjestelmäarkkitehtuurit

2.2.1 Korttipohjainen arkkitehtuuri

Joukkoliikenteen automatisoitu lippujärjestelmä on perinteisesti toteutettu niin, että käyttäjällä on hallussaan tallennusmedia, tavallisimmin etäluettava älykortti, johon käyttäjän ostamat liput tallennetaan myynnin yhteydessä. Leimauksessa kortin sisältö luetaan päätelaitteen toimesta, ja laitteen ohjelma päättelee matkustusosoikeuden kelpoisuuden ja tarvittaessa kirjoittaa kortille lippujen uuden tilan. Tällaista arkkitehtuuria, jossa liput kulkevat käyttäjien lippumedian mukana ja leimaus tapahtuu vain lippumedian ja leimauspäätteen välillä, kutsutaan *korttipohjaiseksi* arkkitehtuuriksi (engl. card-based systems). (Calypso, 2017, s. 4–5)

Leimauspäätte suorittaa leimauksen paikallisesti laitteen ohjelmakoodin, älykortilla säilytettävien tietojen ja taustajärjestelmästä ladattujen tuoteparametrien avulla. Leimauksen aikana laite lukee käyttäjän esittämältä kortilta lippujen tiedot, validoi matkustusosoikeuden, ja ilmoittaa sen käyttäjälle ja mahdollisille muille pääsyä valvoville tahoille, kuten ajoneuvon kuljettajalle. Etukäteen hankittavien lippujen lisäksi älykortteille voidaan tallentaa myös maksutapana kelpaavaa arvoa, josta palvelun hinta veloitetaan leimauksen aikana. Korteilla voidaan säilyttää myös käyttäjän henkilötietoja, joiden avulla kortin haltija voidaan tunnistaa. Tällä pyritään estämään henkilökohtaisten matkakorttien ja lippujen väärinkäyttöä. Matkakortille tallennetaan usein myös leimaushistoriaa, joka voidaan tarvittaessa tulostaa, tai jonka avulla lipuntarkastaja voi tarkastaa käyttäjän oikeuden matkaan esimerkiksi kortille ladatulla arvolla suoritettuna leimauksen jälkeen. Leimaustuloksen muodostaminen paikallisesti leimauslaitteella ja kortilla olevilla tiedoilla on nopeaa, ja tarjoaa hyvän käyttökokemuksen.

Korttipohjaisessa arkkitehtuurissa lippujen ensisijaiset tilat ovat siis asiakkaiden hallussa pitämällä lippumedialla. Älykortin hukkuessa tai vaurioituessa käyttäjä ei enää pysty käyttämään hankkimaansa liikkumispalvelua. Lippujen lukeminen ja uusien lippujen kirjoittaminen täytyy tehdä kortin ollessa lukijassa, mikä tarkoittaa, että tuotteiden käyttämisen ja uusien tuotteiden ostojen yhteydessä käyttäjillä tulee olla kortti mukana. Järjestelmän ylläpitäjällä ei ole mahdollisuutta vaikuttaa korttien sisältöön etänä, mikä tekee virheellisten tilojen korjauksesta haastavaa, ja virheelliset tilat täytyykin yleensä tulla korjaamaan paikan päälle palvelupisteisiin.

Järjestelmän toimintalogiikka ja lippujärjestelmän säännöt on hajautettu suurelle määrälle päätelaitteita. Muutokset lippujärjestelmän sääntöihin täytyy jakaa laitteille tyypillisesti uusien

tuoteparametrien tai ohjelmakoodin muodossa. Tämänkaltaisen hajautetun järjestelmän operointi ja kehittäminen voi olla huomattavasti vaikeampaa kuin keskitettyjen järjestelmien.

Korttipohjaisessa lippujärjestelmässä yhteentoimivuus eri toimijoiden järjestelmien välillä on haastavaa. Korteille tallennettujen tietojen rakenne ja validointilogiikka ovat usein järjestelmän toimittajien itse kehittämiä ratkaisuja, joiden ymmärtäminen muiden toimijoiden osalta vaatii yhteistyötä. Kortin tietojen tulkitseminen ja uusien tietojen kirjoittaminen kortille oikein on vaikeaa ja riskialtista, eikä korttipohjaisen järjestelmän kehittäjä välttämättä halua sallia muille mahdollisuutta muuttaa korttien tiloja. Korttien käsittelyyn tarvitaan yleensä kryptograafisia avaimia, joiden jakaminen saattaa olla riskialtista ja vaatii valvontaa ja luottamista muihin toimijoihin. Modernit älykorttistandardit tukevat kyllä useiden eri avaimilla salattujen sovellusten säilyttämisen samalle älykortille, mikä mahdollistaisi saman älykortin käyttämisen eri toimittajien järjestelmissä, mutta tietyn lippujärjestelmän sovelluksen tiedot on kuitenkin edelleen vain avaintenhaltijan muokattavissa.

Korttipohjaisiin lippujärjestelmiin liittyy yleensä myös laajoja taustajärjestelmiä, joita käytetään toiminnan ylläpitoon, saatavien selvitykseen, käytön analysointiin ja muun lisätoiminnallisuuden tuottamiseen. Myynti- ja leimauslaitteet ovat ainakin ajoittain tiedonsiirtoyhteyksien päässä, jolloin tapahtuneet myynti- ja leimaustapahtumat voidaan välittää päätelaitteiden muistista taustajärjestelmään. Järjestelmässä toimivista korteista ja niillä olevista tuotteista voidaan ylläpitää tietokantaa taustajärjestelmässä, jolloin esimerkiksi kortin kadotessa käyttäjälle voidaan luoda kadonneen kortin tilaa vastaava uusi kortti. Kortille ladattavia lipputuotteita myydään myös verkkokaupoissa, joissa lippuja voidaan ostaa ilman, että käyttäjän täytyy mennä älykortin kanssa myyntipisteeseen. Uudet verkkomyynnit välitetään korteille leimauslaitteiden kautta; Laitteiden muistiin välitetään lista odottavista myynneistä, josta ne ladataan korteille seuraavan leimauksen yhteydessä.

Laajoistakin taustajärjestelmistä huolimatta korttipohjaiset liput ovat aina käyttäjän hallussa, ja matkustusoikeuden validointi ei ole riippuvainen jatkuvasta tiedonsiirtoyhteydestä. Yleensä riittää, että leimauslaitteet ovat yhteydessä taustajärjestelmään esimerkiksi ajon aloituksen ja lopetuksen yhteydessä, ja ne voivat toimia ilman tiedonsiirtoyhteyksiä suurimman osan ajasta ilman ongelmia. Laitteet keräävät tapahtumat omaan muistiinsa, ja lähettävät ne taustajärjestelmään tiedonsiirtoyhteyksien salliessa. Tapahtumien reaaliaikainen välitys taustajärjestelmään ei siis ole välttämätöntä järjestelmän toiminnan ja tulevien leimausten kannalta, sillä leimauksessa käytetään aina kortilla ajan tasalla olevaa tilaa.

2.2.2 Tunnistepohjainen arkkitehtuuri

Mobiilitiedonsiirtoyhteyksien kehittyessä yhä kattavammiksi ja nopeammiksi, julkisen liikenteen lippujärjestelmät voivat perusteellisemmin käyttää toiminnassaan jatkuvaa tiedonsiirtoyhteyttä myös mobiiliverkossa olevilla päätelaitteilla. Tämä mahdollistaa uudenlaisen taustajärjestelmää keskeisesti

hyödyntävän lippujärjestelmän toteutuksen, jossa käyttäjillä on hallussaan ainoastaan tilinsä tunniste, ja kaikki käyttäjän ja lippujen tiedot säilytetään taustajärjestelmässä. Tällaista järjestelmän arkkitehtuuria kutsutaan *tunnistepohjaiseksi* arkkitehtuuriksi (engl. ID-based ticketing, account-based ticketing, ABT, server-centric ticketing), ja leimausta tunnistepohjaiseiksi leimaukseksi. Tunnistepohjaisessa järjestelmässä käyttäjien tileille voidaan myydä ennen matkaa hankittavia lippuja tai arvoa, samalla tavalla kuin korttipohjaisessakin järjestelmässä. Tämän lisäksi tiliin voidaan liittää myös erilaisia maksutapoja, joista veloitus tehdään automaattisesti käytön perusteella. (Calypso, 2017; Kanerva ym., 2020, s. 4)

Leimauksessa käyttäjät esittävät digitaalisesti luettavan tunnisteensa päätelaitteelle, joka validoi matkustusoikeuden ja raportoi nousutapahtuman taustajärjestelmään. Leimauksen aikana päätelaitteet voivat validoida käyttäjien matkustusoikeuden taustajärjestelmästä, jolloin käyttäjän matkustusoikeus validoidaan taustajärjestelmässä ja validoinnin tulos palautetaan päätelaitteelle. Matkustusoikeuden validointi voi tapahtua myös paikallisesti päätelaitteella luvussa 3.2 esitetyillä offline-validointi menettelyillä. Toisin kun korttipohjaisessa järjestelmässä, jossa päätelaitteet saattavat kirjoittaa älykortteille uutta tilaa leimauksen aikana, tunnistepohjaisessa järjestelmässä muutokset lippujen tiloihin tapahtuvat vasta leimauksen jälkeen taustajärjestelmässä leimaustapahtuman käsittelyn yhteydessä. Päätelaitteen vastuulle leimauksessa jää siis ainoastaan matkustusoikeuden validointi, ja tapahtuman rekisteröinti järjestelmään.

Tietojen säilyttäminen korttien sijaan taustajärjestelmässä lisää mahdollisuuksia asiakas- ja lipputietojen hallintaan. Käyttäjien lippuja voidaan muokata milloin tahansa ilman fyysistä yhteyttä älykorttien kanssa. Sen jälkeen, kun käyttäjälle on luotu tili ja hän on saanut haltuunsa luettavan tunnisteena, voidaan lippuja myydä ja palauttaa ilman kirjoitusoperaatioita fyysisen lippumedian kanssa. Käyttäjän ostamat liput tulevat myös käytettäväksi välittömästi hankinnan jälkeen. Myös leimaukset muuttavat tilin tilaa välittömästi, ja käyttäjälle voidaan tarjota palveluita, joissa he näkevät omien lippujensa tilat ja matkustushistorian reaaliaikaisesti. (Calypso, 2017)

Tunnistepohjaisessa leimauksessa käyttäjän hallussa pitämältä medialta luetaan ainoastaan tilin tunniste, jonka avulla loput tiedot löydetään taustajärjestelmästä. Kirjoitusmahdollisuutta lippumedialle ei tarvita. Tämä tarkoittaa, että järjestelmä ei ole enää samalla tavalla riippuvainen lippumedian ominaisuuksista kuin korttipohjainen järjestelmä. Käyttäjän tunnisteena voi toimia mikä tahansa ajoneuvossa kätevästi luettavissa ja autentikoitavissa oleva tunniste, kuten älykortti, mobiilisovelluksesta näytettävä kuviokoodi, älykellon NFC-ominaisuus, tai matkustajan biometrinen tunniste. Tilin tunnisteena voidaan käyttää myös esimerkiksi ajoneuvon rekisteritunnusta, jonka jälkeen tilin lippuja ja maksutapoja voidaan käyttää pysäköintipalveluissa automaattisten rekisterikilvenlukijoiden avulla (Calypso, 2017, s. 9). Tunnisteen väärentäminen ja kopiointi halutaan kuitenkin estää väärikkäytön ehkäisemiseksi, ja tämän vuoksi tunnisteet yleensä autentikoidaan päätelaitteilla tai taustajärjestelmässä leimauksen yhteydessä (Kanerva ym., 2020, s. 4). Tunnisteen kadotessa käyttäjän tiliin voidaan liittää uusi tunniste, jolloin käyttäjän omistamat tuotteet

säilyvät täysin ennallaan taustajärjestelmässä. Useita tunnisteita ja tilejä voidaan myös linkittää yhteen, mikä mahdollistaa esimerkiksi uudenlaisten perhe- tai yritystilien tarjoamisen. Tilit eivät kuitenkaan aina vaadi henkilökohtaisten tietojen liittämistä järjestelmään ja tilit voivat olla myös täysin anonyymejä.

Tunnistepohjaisten lippujärjestelmien toiminta on korttipohjaista järjestelmää riippuvaisempi jatkuvista tiedonsiirtoyhteyksistä. Varsinkin linja-autoliikenteessä, jossa päätelaitteet ovat mobiiliverkossa ja liikkuvat jatkuvasti, tietoliikenneyhteyksien riittämätön toiminta saattaa estää järjestelmän oikeellisen toiminnan ja aiheuttaa lipputulotappioita. Leimaustapahtumat on tärkeä välittää nopeasti taustajärjestelmään, sillä tulevien tapahtumien käsittely riippuu muuttuneesta matkatilin tilasta. Jos käyttäjällä on ostettuna esimerkiksi yhteen matkaan oikeuttava kertalippu, on tieto lipun leimauksesta ja sen aiheuttamat muutokset tilin tilaan saatava taustajärjestelmään ja seuraavien leimaustapahtumien käytettäväksi ennen kuin käyttäjä ehtii käyttämään kyseistä tuotetta uudestaan.

Tunnistepohjainen toteutus mahdollistaa avoimemman ja modulaarisemman lippujärjestelmän kuin korttipohjainen järjestelmä. Päätelaitteiden ei enää tarvitse ymmärtää lippujärjestelmän sääntöjä ja lippujen validointilogiikkaa, vaan käsittely tapahtuu keskitetysti taustajärjestelmässä. Tämän ansiosta päätelaitteet voivat yleensä olla toteutukseltaan huomattavasti kevyempiä ja halvempia kuin korttipohjaisessa järjestelmässä. Niiden ohjelmia tai leimaukseen tarvittavia tuoteparametrejä ei myöskään tarvitse kehittää ja päivittää lippujärjestelmän sääntöjen muuttuessa samalla tavalla kuin korttipohjaisessa järjestelmässä. Matkustusosoikeuden validointiin riittää, että luetulla tunnisteella ja leimausparametreillä kutsutaan oikeaa validointirajapintaa. Sama validointilaitte voi olla yhteydessä usean eri lippujärjestelmän taustajärjestelmään, ja validoida matkustusosoikeuden eri rajapinnasta luetun tunnisteiden perusteella. Rajapintojen avulla myös muut toimijat voivat melko helposti liittyä lippujärjestelmään joko lippuja myyvinä tai validoivina tahoina. Liiketoiminnan säännöt on ohjelmoitu keskitetysti taustajärjestelmän palveluihin, ja niiden kehittäminen voi olla huomattavasti helpompaa kuin hajautetussa korttipohjaisessa järjestelmässä.

Tunnistepohjaisessa järjestelmässä voidaan tarjota samanlaisia etukäteen ostettavia lipputuotteita kuin korttipohjaisessakin, kuten kausi- ja sarjalippuja, jotka voidaan leimauksen yhteydessä validoida rajapintakutsulla tai offline-menettelyllä. Tunnistepohjaisuus tuo kuitenkin uusia mahdollisuuksia käyttöön perustuvaan veloitukseen. Käyttäjien tileihin voidaan liittää automaattisia maksutapoja, joista tapahtuneet matkat veloitetaan automaattisesti. Myös etukäteen tileille ladattuun arvoon voidaan kytkeä automaattinen lisälataus, jonka avulla valitusta maksutavasta ladataan tarvittaessa automaattisesti tilille lisää arvoa. Järjestelmä voi seurata tapahtuneita leimaustapahtumia, ja päätellä niiden perusteella dynaamisilla hinnoittelusäännöillä käyttäjälle parhaan mahdollisimman hinnan, joka veloitetaan automaattisesti ennakkoon ladatusta arvosta tai automaattisesta maksukanavasta. Dynaamisia hinnoittelusääntöjä voivat olla esimerkiksi tiettyyn aikaväliin asetettu kattohinta, jonka täytyessä järjestelmä ei veloita käyttäjältä enempää maksua kyseisenä aikavälinä. Tämä palvelee samaa käyttäjien

tarvetta kuin etukäteen myytävät kausiliput, joiden avulla säännöllisesti matkustaville käyttäjille tarjotaan edullisempaa hintaa.

Lippujärjestelmässä voi olla samanaikaisesti korttipohjaisen ja tunnistepohjaisen arkkitehtuurin tuotteita ja kortteja. Olemassa olevia laajoja korttipohjaisia järjestelmiä voi olla hyvin raskas lähteä uudistamaan tunnistepohjaisiksi yhdellä kertaa. Uusien tunnistepohjaisten tuotteiden tuominen osaksi olemassa olevia korttipohjaisia lippujärjestelmiä voi tapahtua vähitellen, tuomalla uudet tuotteet tunnistepohjaiset lippu- ja maksuvaihtoehdot nykyisten korttipohjaisten tuotteiden rinnalle.

2.3 Lippu- ja tunnistemediat

Tässä luvussa esitetään erilaisia lippumedioita, joita voidaan käyttää matkustusoikeuksien validoinnissa. Eri lippumedioiden toimintaa kuvataan sekä korttipohjaisessa että tunnistepohjaisessa lippujärjestelmässä. Lippujärjestelmissä käynnissä oleva trendi on siirtyminen joukkoliikenneoperaattorien itse liikkeelle laskemasta lippumediasta, kuten järjestelmäkohtaisista älykorteista, kohti käyttäjällä valmiiksi hallussa olevaa lippumediaa, kuten matkapuhelinta tai EMV-maksukorttia (Whitaker, 2016).

2.3.1 Älykortit

Etäluettavat älykortit ovat hyvin yleinen tallennusmedia lippujen tai tunnisteiden säilyttämiseen. Niiden käyttö ajoneuvossa on erittäin helppoa ja tiedonsiirto kortin ja lukijalaitteen välillä on nopeaa tähän käyttötarkoitukseen. Leimaustilanteessa korttien tietoja voidaan sekä lukea että kirjoittaa. Kortteja voidaan käyttää perinteisellä korttipohjaisella tavalla, jolloin käyttäjän omistamat liput ja muu tieto ovat kortilla, tai tunnistepohjaisessa järjestelmässä, jolloin kortilla on käyttäjän tilin tunniste.

Älykorttien sisältöä suojataan kryptograafian avulla, korttien kopioimisen ja luvattoman muokkaamisen estämiseksi. Luku- ja kirjoitusoperaatioita varten tarvitaan usein symmetrisiä salausavaimia, ja sisältöä voidaan muokata vain avainten haltijoiden toimesta. Leimauslaitteet voivat myös avainten avulla olla varmoja tiedon alkuperästä ja eheydestä. Modernien älykorttistandardien, kuten Mifare DESfiren, tietoturvan taso on korkea, ja korttien jäljittely ja kloonaus on erittäin vaikeaa. DESfire-korttien valmistaja suosittelee kuitenkin ympäröivälle järjestelmälle tarkkoja tietoturvakäytäntöjä, joilla ehkäistään kortin tietoturvan mahdollisista murtamisista aiheutuvia riskejä. (NPX Semiconductors, 2010)

2.3.2 Mobiililaitteet

Modernit joukkoliikenteen lippu- ja maksujärjestelmät tarjoavat käyttäjille usein matkapuhelimella, tai muilla mobiililaitteilla kuten älykelloilla, käytettäviä ratkaisuja matkustusoikeuden hankintaan. Matkapuhelimet soveltuvat tähän erittäin hyvin, sillä ne kulkevat valmiiksi matkustajien mukana

ja mahdollistavat tuotteiden oston laitteen omaa verkkoyhteyttä käyttäen. Mobiililaitteiden käyttämiseen lippumedianä tai maksutapana on monia eri tapoja, kuten lippujen myynti ja validointi itsepalvelun mahdollistavan sovelluksen avulla, älykorttien emulointi ja EMV-maksaminen. (Cheng, 2017)

Mobiilisovellusten avulla voidaan myydä ja validoida järjestelmässä toimivia lippuja. Lippu voidaan validoida esimerkiksi pelkästään visuaalisen tarkastuksen avulla, jossa lipun autenttisuus käy ilmi lipuntarkastajalle sovelluksen käyttöliittymästä. Tämä työ käsittelee kuitenkin ainoastaan päätelaitteen kanssa tapahtuvaa automaattista lipun validointia, jossa leimauksen yhteydessä puhelin välittää tietoja päätelaitteelle, esimerkiksi optisesti luettavan kuviokoodin, kuten QR- tai Aztec-koodin, avulla. Leimauslaitteen optinen lukija pystyy lukemaan kuviokoodin sisältämän datan puhelimen näytöltä ja suorittamaan leimauksen sen avulla. Optisesti luettavien koodien leimaaminen on havaittu suunnilleen yhtä nopeaksi ja helppokäyttöiseksi kuin älykorttien leimaaminen (Gooch, 2019). Tiedonsiirtoon voidaan käyttää myös NFC-tekniikka, mutta se ei kuitenkaan ole saatavilla kaikissa puhelimissa, joten pääasiallisena kanavana käytetään yleensä optista koodia. Tiedon välitys optisen lipun avulla mahdollistaa ainoastaan yksisuuntaisen tiedonvälityksen puhelimelta leimauslaitteelle. Tämä tarkoittaa, että lippujen tai tunnisteiden lukeminen ja niiden kelpoisuuden tarkastaminen voidaan toteuttaa, mutta uuden tilan kirjoitusta takaisin mobiililaitteelle leimauksen aikana ei voida suorittaa.

Mobiililiput voidaan toteuttaa sekä korttipohjaisella, eli tässä tapauksessa yleisemmin lippumediapohjaisella, tai tunnistepohjaisella arkkitehtuurilla. Lippumediapohjaisuudella tarkoitetaan tässä yhteydessä siis sitä, että lipun ensisijainen tietosisältö on käyttäjän hallussa olevalla medialla eli matkapuhelimella. Tällöin leimauksen aikana leimattavan lipun tiedot välitetään leimauslaitteelle, joka validoi paikallisesti niiden kelpoisuuden. Liput voi ostaa mobiililaitteelle myyntirajapintojen avulla, jolloin leimauksessa käytetty lipun tietosisältö siirtyy mobiililaitteelle. Tiedot voidaan allekirjoittaa lipun liikkeellelaskijan yksityisavaimella niin, että kaikki toimijat voivat vahvistaa niiden eheyden ja autenttisuuden.

Koska optisesti luettavien koodien avulla toteutettu tiedonsiirto puhelimelta leimauslaitteelle on vain yksisuuntainen, kaikkia korttipohjaisten lippujen toiminnallisuuksia ei voida toteuttaa samalla tavalla mobiililipuilla kuin älykorteille tallennettaville lipuille. Esimerkiksi etukäteen ladatun arvon automaattista veloitusta matkan yhteydessä ei voida toteuttaa. Myöskään sarjalippujen automaattinen kulutus tai kausilippujen aktivointi ei ole mahdollista leimaustapahtuman aikana. Mobiililaitteet voivat tarjota käyttäjälle kuitenkin älykorteista poiketen interaktiivisen käyttöliittymän, jonka avulla käyttäjä voi manuaalisesti tehdä vastaavia toimintoja, kuten ostaa kertalipun tai aktivoida sarjalipusta yhden käyttökerran omaksi lipuksi.

Osa Android-puhelimista kykenee emuloimaan älykorttien toimintaa niin, että puhelimia voidaan käyttää lippumedianä tavallisten älykorttien tapaan. Samanlaisen tietoturvatason saavuttaminen vaatii kuitenkin erillisen tietoturvaelementin (engl. Secure Element, SE) hyödyntämisen. Elementti voi olla mobiililaitteella tai SIM-kortilla, ja sen toimintaa rajoitetaan Android-käyttöjärjestelmän toimesta, mikä

puolestaan rajoittaa tekniikan avointa hyödyntämistä sovelluskehittäjien osalta. SE-elementin avulla toteutetun korttiemulaation avulla puhelinta voidaan käyttää korttipohjaisen arkkitehtuurin lippumedianan samalla tavalla ja samaa infrastruktuuria hyödyntämällä kuin älykorttejakin, mikäli infrastruktuuri ja tietoturvaelementti ymmärtävät samaa standardia. Älypuhelinlusta Androidin versiossa 4.4 lisättiin myös ominaisuus, jossa Android-aplikaatio voi emuloida ISO/IEC 14443-4 standardien mukaisten älykorttien toimintaa Host Card Emulation (HCE) -nimisellä tekniikalla (Google, 2021a). Tällöin tiedonsiirto NFC-lukijan kanssa tapahtuu suoraan sovellusprosessin kanssa ilman tietoturvaelementtiä. HCE:n hyödyntämisessä osa tietoturvavastuusta siirtyy kuitenkin tietoturvaelementistä palvelimelle, mikä tarkoittaa, että HCE-aplikaatioiden hyödyntäminen älykorttien tavoin vaatii korttipohjaiseen järjestelmään uudenlaista taustajärjestelmätoteutusta. HCE-tekniikkaa voidaan kuitenkin käyttää lippujen ja tunnisteiden välittämiseen sovelluksesta päätelaitteelle, eli samaan käyttötarkoitukseen kuin optisesti luettavia kuviokoodejakin käytetään. Tällöin leimaus on mahdollista myös ilman sovelluksen avaamista. Yhtenä ongelmana HCE-tekniikan käytössä on rajoitettu puhelinjoukko, jossa tekniikan hyödyntäminen on mahdollista, sillä IOS-laitteet eivät ainakaan tämän työn kirjoitushetkellä mahdollista applikaatioiden kehittäjille kyseisen tekniikan hyödyntämistä, sillä se mahdollistaisi Apple Payn kanssa kilpailevien maksusovellusten kehittämisen (GoToTags, 2019.) (OSPT Alliance, 2016; Cheng, 2017)

Tunnistepohjaisessa järjestelmässä mobiililaitte saa taustajärjestelmästä tilin tunnisteiden ja välittää sen leimauksessa leimauslaitteelle. Sovellus voi myös antaa käyttäjän valita minkä lipun hän haluaa leimata ja välittää leimauslaitteelle sekä tilin että valitun lipun tunnisteet, jolloin leimaus kohdistetaan kyseiseen lippuun. Tunnisteiden välitys voi tapahtua optisesti luettavien koodien tai NFC-tiedonsiirron avulla. Myös ”Bluetooth Low Energy” -tekniikan käyttöä matkustajien automaattisempaa tunnistamista ja matkojen seuraamista varten on tutkittu (Ferreira, Dias & Cunha, 2020).

Älykorteilta luettavien lippujen autentikointi ja tiedon eheyden tarkastus tapahtuu kortilla ja leimauslaitteilla olevien kryptograafisten avainten avulla. Mobiililaitteilta välitettävä koodi voidaan myös autentikoida kryptograafisilla keinoilla. Koodi voidaan salata symmetrisellä tai asymmetrisellä salauksella, jolloin salauksen purkaja varmistuu siitä, että koodi on varmasti luotu avainten hallussapitäjän toimesta. Salauksen sijaan koodi voidaan myös allekirjoittaa lisäämällä sen loppuun samankaltaisten salausalgoritmien tuottaman vahviteen, jolloin koodin sisältö voidaan ymmärtää myös ilman salauksen purkuun tarvittavia avaimia. Symmetrisessä salauksessa purkuun tarvittavan avaimen vuotaminen mahdollistaa lippujen väärentämisen, joten symmetrisen avaimen jakaminen on riskialtista. Asymmetrisessä salauksessa julkista avainta voidaan puolestaan jakaa vapaasti, minkä vuoksi asymmetrisen salaus on parempi vaihtoehto, mikäli koodien validointi tapahtuu useissa eri paikoissa, kuten useilla leimauslaitteilla tai muiden toimijoiden järjestelmissä. (Whitaker, 2008)

Mobiililippujen yhtenä haasteena on leimauksessa välitettävien koodien kopioinnin estäminen. Mobiililaitteen näytölle piirtyvän kuviokoodin kopioimista ei voida täysin estää. Väärinkäyttäjät

voivat esimerkiksi toteuttaa oman puhelinosovelluksen, joka lukee koodin aidosta sovelluksesta, jakaa sen eteenpäin muille käyttäjille, ja esittää kopioidun QR-koodin aidon sovelluksen kaltaisella näkymällä. Tällöin edes ihmisen on vaikea havaita väärinkäyttöä. Edellä mainittu tiedon autentikointi varmistaa tiedon eheyden ja alkuperän, mutta ei auta kuviokoodien kopioinnin tunnistamisessa ja ehkäisyssä.

Kopioinnin avulla tapahtuvan väärinkäytön estämiseksi voidaan käyttää esimerkiksi jatkuvaa koodigenerointia (Kanerva ym., 2020, s. 4). Leimauksessa välitettävään tietosisältöön voidaan asettaa viimeinen validisuusajka, jonka jälkeen validointi ei enää hyväksy kyseistä tietoa. Tällöin QR-koodeista tehdyt kopiot ovat käytettävissä vain hyvin rajallisen ajan, ja niiden väärinkäyttäminen kopioinnin avulla on vaikeampaa. Aidot lippusovellukset voivat päivittää koodien sisältö jatkuvasti, jolloin niillä on aina käytössä validi tietosisältö. Koodien sisällön päivitys voi olla uusien sisältöjen hakemista taustajärjestelmästä, tai siinä voidaan hyödyntää esimerkiksi aikaan perustuvaa salasanojen generointia (engl. time-based one-time password).

2.3.3 Paperiliput

Käyttäjille voidaan myydä myös paperisia tai esimerkiksi sähköpostiin lähetettäviä lippuja, jotka on varustettu optisesti luettavalla kuviokoodilla. Tällöin koodi on luotava ennen lipun tulostusta tai



Kuva 1 - QR-koodilla varustettu paperinen kertalippu

lähetystä, eikä sitä voida muuttaa jälkeenkäin. Tämänkaltaisten lippujen kopioimista ei voida estää, minkä vuoksi niiden toiminta on rajattava niin, että kopioimisella ei saavuteta merkittävää hyötyä. Korttipohjaisessa järjestelmässä tämä on hyvin vaikeaa, ja lippuja voidaan myydä riskittömästi vain tietyille matkalle ja paikalle myytäviin lippuihin, joissa paikka vahvistetaan lipun validoinnin yhteydessä. Tietyille matkalle myytyjä lippuja validoivat leimauslaitteet voivat myös pitää paikallista listaa validoiduista lipuista ja näin tunnistaa kopioidut liput.

Tunnistepohjaisessa järjestelmässä paperisissa lipuissa voidaan käyttää muitakin tuotetyyppejä, ilman että niiden väärinkäyttö kopioimalla on mahdollista. Koska lippujen leimaukset käsitellään reaaliaikaisesti taustajärjestelmässä, lipputuotteista, jotka vähentävät arvoa tai kuluttavat jäljellä olevia matkoja jokaisella leimauksella, ei saada hyötyä kopioimalla. Jos liput kuitenkin oikeuttavat lisäveloituksettomiin matkoihin tietynä voimassaoloaikana, on edelleen riski, että lippuja kopioimalla useampi henkilö matkustaa järjestelmässä saman lipun avulla. Mahdolliset lipputuloriskit voidaan kuitenkin hyväksyä, mikäli ne havaitaan riittävän mitättömäksi.

2.3.4 Lähimaksulliset pankkikortit

cEMV, eli "contactless EMV", -pankkikortit ovat EMV-standardin (Europay, Mastercard and Visa) mukaisia maksamiseen käytettäviä sirukortteja, jotka tukevat myös kontaktittomia lähimaksutapahtumia. Pankkikortteja voidaan käyttää maksuvälineenä käteisen sijasta vähittäiskaupassa, esimerkiksi matkalippuja etukäteen ostettaessa. Tämän lisäksi pankkikortteja on alettu hyödyntää tunnistepohjaisessa lippujärjestelmässä käyttäjän tunnisteenä ja käyttöön perustuvan veloituksen maksukanavana. Korttien toiminta pohjautuu samoihin ISO/IEC 7816 ja ISO/IEC 14443-4 -standardeihin kuin monien muidenkin modernien älykorttien toiminta, ja niiden leimaaminen tapahtuu käyttäjän näkökulmasta samalla tavalla viemällä kortti lukijalaitteen lähelle. Korttien lisäksi EMV-lähimaksaminen on nykyään mahdollista myös useilla matkapuhelimilla ja älykelloilla.

Rahoitusalan yhdistys UK Finance määrittää 3 eri mallia cEMV-korttien käytölle liikkumispalveluiden lippumedianana (UK Finance, 2017):

Malli 1: Pankkikortteja käytetään matkan maksutapana käteisen sijasta samalla tavalla kuin vähittäiskaupassakin. Tarvetta erilliselle tunnistepohjaiselle taustajärjestelmälle ei ole, vaan matkat veloitetaan suoraan itsenäisillä transaktioilla EMV-maksuverkostosta. Tarjottava toiminnallisuus ja matkojen hinnoittelu on kuitenkin hyvin rajoittunutta, sillä matkan hinta täytyy tietää matkan aloituksessa tapahtuvan maksutapahtuman yhteydessä. Käyttäjä voi saada tapahtumasta tulostetun lipun, tai digitaalisen kuitin, joka voidaan tarkastaa lipuntarkastuksen yhteydessä. (UK Finance, 2017; Gooch, 2020)

Malli 2: Leimauspäätteen ja EMV-maksuverkoston välissä on tunnistepohjainen taustajärjestelmä, joka kokoaa ja käsittelee keskitetysti pankkikortilla tapahtuneet leimaukset ja hallinnoi EMV-verkostosta tehtäviä maksutapahtumia. Veloitettava hinta voidaan johtaa useista tapahtuneista matkoista dynaamisilla hinnoittelusäännöillä, kuten kattohinnoilla. EMV-verkosta tehtävä maksutransaktio voidaan suorittaa esimerkiksi kerran vuorokaudessa, jolloin myös transaktiokustannukset ovat mallia 1 pienemmät. (UK Finance, 2017; Gooch, 2020)

Malli 3: cEMV-kortit toimivat pelkästään tunnistepohjaisen lippujärjestelmän tunnisteina. Ennen matkustusta käyttäjien tulee luoda järjestelmään etukäteen tili, liittää pankkikortti tilin tunnisteeksi, ja ostaa tilille liikkumispalvelun käyttöön oikeuttava lipputuote. Tunnisteena toimiva cEMV-kortti voidaan myös liittää tilin maksutavaksi, mutta tämä ei tapahdu automaattisesti kuten mallissa 2. (UK Finance, 2017; Gooch, 2020)

EMV-pankkikorttien käytöllä voidaan korvata joukkoliikennejärjestelmien itse ylläpidettävää lippumediaa ja lipunmyynti-infrastruktuuria. Korttien käyttö on käyttäjille hyvin helppoa, sillä mallien 1 ja 2 mukaisissa toteutuksissa pankkikortin omistavien käyttäjien ei tarvitse tehdä mitään toimia ennen matkaa (UK Finance, 2017). Kaikilla joukkoliikennettä käyttävillä henkilöillä ei kuitenkaan ole hallussaan pankkikortteja, tai he eivät halua niitä käyttää, joten täydellinen siirtyminen cEMV-kortteihin ei todennäköisesti ole mahdollista (Zamer 2018). Etukäteen maksettavat EMV-standardia tukevat Prepaid-kortit saattavat olla ratkaisu näissä tilanteissa (CPI Card Group, 2016).

2.4 Joukkoliikenne Suomessa

2.4.1 Julkisen henkilöliikenteen lainsäädäntö

3. joulukuuta 2009 Euroopan unionissa astui voimaan palvelusopimusasetus (EY) N:o 1370/2007, joka ei salli liikkumispalveluiden kilpailun rajoittamista, ellei se ole tarpeellista palveluiden määrän tai laadun varmistamiseksi. Muutoksille määrättiin asetuksessa siirtymäaika, joka päättyi vuonna 2019. Suomessa palvelusopimusasetus aikaansai joukkoliikennelain (869/2009) uudistuksen, joka mullisti bussien kaukoliikenteen järjestämismallin, ja on lisännyt kilpailua ja tehnyt kaukoliikenteestä entistä houkuttelevampaa. (Valmari, 2019)

Myöhemmin 1.7.2018 lähtien joukkoliikennelain tilalle tuli voimaan uudistunut liikennepalvelulaki, joka asettaa lisää veloitteita liikkumispalveluita tuottavien toimijoiden tietojärjestelmien avoimuudelle ja yhteentoimivuudelle. Liikennepalvelulain pykälissä 154 § - 161 § lippu- ja

maksujärjestelmästä vastaavalle toimijalle asetetaan velvoite avoimien rajapintojen tarjoamisen muille toimijoille (Laki liikenteen palveluista, 24.5.2017/320). Muille liikkumispalveluita tarjoaville tahoille on tarjottava mahdollisuus vähintään perushintaisen kertamatkaan oikeuttavan lipputuotteen hankkimista avoimen rajapinnan kautta (155 §). Matkustusoikeuksien on myös oltava avoimesti validoitavissa. Muiden palveluntarjoajien tietojärjestelmille on myös tarjottava mahdollisuus asioida rajapintojen kautta asiakkaan puolesta (156 §). Näissä pykälissä mainittujen rajapintojen yhteentoimivuudesta on huolehdittava ja rajapinnoissa on käytettävä yleiskäyttöisiä teknologioita (157 § ja 159 §). Lakiuudistuksen tavoitteena on paremman palvelutason tarjoaminen ja valinnanvapauden lisääminen matkustajille (Liikenne- ja viestintäministeriö, 2018).

Palvelulain uudistukseen liittyy oleellisesti liikkumispalveluiden tarjoamisen muuttunut ajattelumalli. Liikenteen palveluita käyttävän henkilön tavoite on päästä määränpään parhaalla mahdollisella tavalla. Avoimien tietojärjestelmien ja älykkään tiedonkäsittelyn avulla digitaaliset ja fyysiset palvelut voidaan koota paremmin matkustajien tarpeita vastaaviksi palvelukokonaisuuksiksi. Useiden eri liikkumismuotoja ja liikennejärjestelmien palveluita yhdistämällä matkustajille voidaan tarjota tehokkaampia ja helppokäyttöisempiä tapoja löytää paras matkaketju lähtöpisteen ja määränpään välillä. Tämä matkaketju voi koostua useista eri liikkumismuodoista ja palveluntarjoajista, joiden tuotteet voidaan yhdistää älykkäiden ja avoimien tietojärjestelmien avulla. Tämän kaltaisesta ajattelutavasta käytetään yleisesti termiä liikkuminen palveluna (engl. mobility as a service, MaaS). (Liikennevirasto, 2018, s. 9-10)

Palvelulain asettamat vaatimukset ohjaavat joukkoliikenteen lippujärjestelmiä tunnistepohjaisiksi. Koska esimerkiksi kertalippuja täytyy pystyä ostamaan avoimien rajapintojen avulla, täytyy ne toteuttaa tunnistepohjaisesti, jotta esimerkiksi lippujen väärinkäyttö niitä kopioimalla ei ole mahdollista. Järjestelmän avoimuusvaatimukset lippujen hankintaan ja validointiin liittyen ovat yksi keskeinen syy järjestelmien siirtymiselle luvussa 2.1.5 esitettyyn tunnistepohjaiseen järjestelmään.

2.4.2 Lippujärjestelmien yhteentoimivuus

Liikennepalvelulain vaatimat avoimet rajapinnat mahdollistavat eri palveluntarjoajien liikkumispalveluiden hankkimisen käyttäjän puolesta kolmannen osapuolen toimesta. Tämä mahdollistaa erilaisten yhdistämispalveluiden tuottamisen, jossa käyttäjille voidaan tarjota laajoja palvelukokonaisuuksia yhden palvelun kautta. Esimerkki tämänkaltaisesta yhdistelmäpalvelusta on vuonna 2017 lanseerattu Whim-palvelu, jonka avulla käyttäjä voi tarkastella useiden liikkumismuotojen palveluita kätevästi yhden sovelluksen kautta. Sovelluksella voi myös hankkia matkustusoikeuden koko matkaketjun osamatkoille. (Salminen, 2018; Pesonen, 2017)

Eri liikkumispalveluiden käyttämisestä saman sovelluksen kautta tai useiden tuotteiden paketoimisesta yhdeksi tuotteeksi seuraa kuitenkin vain rajallista etua käyttäjälle helppokäyttöisemmän

palvelun muodossa. Enemmän hyötyä seuraa järjestelystä, jossa yksi lipputuote hyväksytään useiden eri toimijoiden liikkumispalveluissa, ja lipputulot jaetaan oikeudenmukaisesti joko arvioperusteisesti tai matkustusmäärät huomioivan clearing-prosessin avulla. Käyttäjä on kiinnostunut lähinnä matkaketjun kokonaishinnasta, ja tämän kaltaisella yhteistoiminnalla käyttäjille voidaan tarjota edullisempia liikkumispalveluita. Tätä varten eri palveluntarjoajien on kyettävä validoimaan yhteiskäyttöisiä lippuja. (Kanerva ym., 2020)

Yhteentoimivuus on ollut eri toimijoiden suljettujen järjestelmien takia haastavaa. Suomessa eri lippujärjestelmillä ei ole ollut yhtenäistä standardia lippujen tietosisällölle ja niiden validointiin tarvittavalle logiikalle. Esimerkiksi Ruotsissa maan julkisen liikenteen toimintaa koordinoiva Samtrafiken on kehittänyt avoimen ”The National Ticket Standard” -lippustandardin (BoB), jota noudattamalla eri toimijat pystyvät ymmärtämään ja validoimaan toistensa liikkeelle laskemia lippuja. BoB-standardi määrittää lipun voimassaolon ja kelpoisuussijaintien määrittämiseen käytettävän ”Tickle”-määrityskielen. Standardi tarjoaa myös ohjelmakirjaston lipputietojen validointia varten. Yhtenäinen standardi lippujen tietosisällölle ja validointilogiikalle mahdollistaa eri järjestelmien yhteistoimivuuden. (Samtrafiken, 2016–2021)

Lippujärjestelmien omat säännöt ja varsinkin omat suljetut tekniset toteutukset ovat tehneet yhteistoiminnasta haastavaa. Tähän on kuitenkin yritetty saada muutosta avoimempien järjestelmien avulla. Useat Suomen joukkoliikennejärjestelmät ovat ottaneet käyttöön LMJ:n (TVV lippu- ja maksujärjestelmä Oy) ylläpitämän valtakunnallisen Waltti-lippujärjestelmän tuotteita. LMJ on 21 kaupunkiseudun ja Helsingin seudun liikenteen omistama IT- hankinta ja palveluyhtiö, jonka tehtävänä on tuottaa innovatiivisia joukkoliikenteen palveluita, ja toimia eri toimijoita yhdistävänä tahona. Waltti on LMJ:n ylläpitämä maksu- ja lippujärjestelmä, joka tarjoaa tällä hetkellä kortteja ja tuotteita yli 20 kaupunkiseudun alueille. Samat Waltti-kortit ja kortteille ladattava arvo toimivat kaikilla Waltti-alueilla, mutta kausiliput toimivat aina vain tietyllä alueella. Useat eri kaupunkiseudut ovat korvanneet omia lippujärjestelmiään ja tuotteitaan Waltti-tuotteilla. Aiemmin korttipohjaisesti toiminutta lippu- ja maksujärjestelmää ollaan muuttamassa tunnistepohjaiseksi (Kanerva ym., 2020, s. 11). (TVV lippu- ja maksujärjestelmä Oy, ei pvm)

Tunnistepohjaisuuteen siirtyminen helpottaa eri lippujärjestelmien teknisen yhteentoimivuuden saavuttamista. Lipputuotteen validointia varten leimauslaitteiden ja muiden matkustusoikeutta validoivien tahojen ei tarvitse enää ymmärtää muiden lippujärjestelmien lipputuotteiden sääntöjä ja lipun tietojen määrittämiä ja koodauksia, eikä laitteille tarvitse ladata tuoteparametreja. Mikäli validointi tapahtuu taustajärjestelmässä, ei ole tarvetta yhteiselle standardille lipun sisällöstä ja validointilogiikasta, tai validoinnista vastaavien ohjelmakirjastojen ja tuoteparametrien jakamiselle muille toimijoille. Matkustusoikeuden validoinniksi riittää tunnisteen lukeminen ja validoinnista vastaavan järjestelmän rajapinnan kutsuminen. Traficomien julkaisemassa selvityksessä (Kanerva ym., 2020) selvitettiin tunnistepohjaisten lippujärjestelmien yhteentoimivuuteen liittyviä haasteita. Selvityksessä todettiin, että

koordinoinnin ja kehityssuunnan puute on teknistä yhteentoimivuutta suurempi este järjestelmien yhteistoiminnalle.

2.4.3 Joukkoliikennejärjestelmät ja matkustajamäärät

Liikenne- ja viestintävirasto Traficom in toimeksi antamasta ja Tilastokeskuksen tuottamasta ”Julkisen liikenteen suoritetilasto 2018” -julkaisusta voidaan tarkastella Suomen joukkoliikenteen matkustajamääriä. Suomen joukkoliikenteessä tehtiin vuonna 2018 573 miljoonaa matkaa. Keskiarvoisesti tämä tarkoittaa 1,57 miljoonaa matkaa päivässä, vaikkakin matkojen jakaantuminen päivien kesken ei ole tasaista. Matkustajamäärät lasketaan tapahtuneiden nousujen perusteella.

Helsingin seudun liikenne (HSL) raportoi (2019a) vuoden 2018 matkustajamääräksi 387 miljoonaa. Tähän lukuun laskettiin mukaan koko HSL-alueella tapahtuneiden nousujen määrä. HSL:n lippujärjestelmässä voidaan käyttää korttipohjaisesti toimivia älykortteja, tai mobiilisovelluksesta tai lippuautomaatista ostettavia lippuja. Mobiili- ja paperilippuja ei leimata lukijalaitteiden kanssa matkan aikana, mutta ne voidaan visuaalisesti validoida esimerkiksi kuljettajan toimesta. Osassa liikennevälineitä lippuja ei validoida muuten kuin lipuntarkastajan toimesta. (HSL, 2021a) Järjestelmässä luotetaan siis siihen, että lipuntarkastuksen ja sakkomaksun uhka riittää ohjaamaan käyttäjiä lippujen hankintaan. Vuonna 2018 HSL:n lipuntarkastuksessa lippu puuttui 2,94 % tarkastetuilta matkustajilta (HSL, 2019b). HSL on ilmoittanut (2021b) tunnistepohjaisen taustajärjestelmän kehittämistä; tavoitteena on, että järjestelmä on käytössä vuonna 2022.

Tampereen seudun alueella operoiva Nysse on raportoinut (2020) vuoden 2019 matkojen määräksi 41 miljoonaa. Lippujärjestelmässä toimivat älykortit ovat toimineet korttipohjaisesti. Tavoitteena on kuitenkin uuden tunnistepohjaisen lippujärjestelmän käyttöönotto, ja mobiilisovelluksella ostettavat liput toimivat jo tunnistepohjaisina. (Elli Kotakorpi, 2020)

Turun seudun alueella operoiva joukkoliikennejärjestelmä Föli raportoi (Föli, 2020), että järjestelmässä tehtiin vuonna 2019 26,3 miljoonaa matkaa. Fölin lippujärjestelmän toteutus on saksalaisen INIT-yrityksen toimittama, ja järjestelmä on ollut täysin tunnistepohjainen jo vuodesta 2016. Lippujärjestelmän säännöt ovat melko yksinkertaiset; lippujärjestelmässä ei ole käytössä esimerkiksi matkustusvyöhykkeitä, vaikka INIT:in lippujärjestelmäalusta niitä tukeekin. (INIT, 2016)

Tässä luvussa esitetyt matkustajamäärät antavat tietynlaista käsitystä Suomen joukkoliikenteen lippu- ja maksujärjestelmiin kohdistuvasta rasituksesta. Selkeästi HSL:n lippujärjestelmän käyttö on muihin Suomen joukkoliikennejärjestelmiin verrattuna huomattavasti suurempaa.

3 Tunnistepohjainen matkustusoikeuden validointi

Alaluvussa 2.2.2 esiteltiin tunnistepohjainen lippujärjestelmäarkkitehtuuri. Kyseisen arkkitehtuurin toiminnan yksi keskeisimmistä toiminnoista on tunnistepohjainen leimaus, joka tarkoittaa nousun yhteydessä tapahtuvaa tunnistemedian lukemista, matkustusoikeuden validointia, ja nousutapahtuman rekisteröintiä järjestelmään. Leimauksen tehokas ja luotettava toteutus on erittäin tärkeää koko joukkoliikennejärjestelmän käytön sujuvuuden kannalta. Tässä luvussa esitetään eri tapoja tunnistepohjaiseen matkustusoikeuden validointiin sekä online- että offline-tilanteissa.

Käyttäjän matkustusoikeus riippuu käyttäjän omistamista lipuista ja tilin maksukyvyistä, joka voi olla tilille etukäteen ladattua arvoa tai tiliin kytkettyjen maksukanavien maksukykyä. Tunnistepohjaisessa järjestelmässä nämä tiedot ovat lippumedian sijaan taustajärjestelmässä. Tämän vuoksi matkustusoikeuden validoinnin täytyy tapahtua taustajärjestelmässä, tai sen suorittamiseen tarvittavat tiedot on tuotava leimauslaitteelle.

Mikäli matkustusoikeutta ei voida validoida nousuhetkellä tai validointi on virheellinen, tästä saattaa syntyä lipputulotappioita joukkoliikenteen operaattoreille. Kaikki nousutapahtumat välitetään joka tapauksessa taustajärjestelmään, mutta jos käyttäjä matkusti ilman oikeutta, eikä taustajärjestelmä pysty veloittamaan matkan hintaa, jää matkan lipputulot saamatta. Lipputulosten kiertämiseen liittyvät riskit halutaan tietenkin minimoida, vaikkakin tietty lipputuloriski saatetaan hyväksyä, mikäli se esimerkiksi tarkoittaa huomattavasti kevyempää lippujärjestelmätoteutusta. Ilman matkustusoikeutta tehdyt matkat voidaan myös tietyissä tapauksissa jättää asiakkaalle velaksi, ja huomioida esimerkiksi tulevien arvon latausten yhteydessä.

3.1 Online-validointi

Tuotteiden leimaus voidaan toteuttaa tiedonsiirtoverkon yli suoritettulla rajapintakutsulla, jota kutsutaan tästä eteenpäin online-validoinniksi. Kutsussa taustajärjestelmään välitetään leimattavan käyttäjän tunniste ja mahdolliset leimausparametrit, ja taustajärjestelmä palauttaa leimauksen tuloksena matkustusoikeuden kelpoisuuden ja mahdollisesti muita leimaukseen liittyviä tietoja, kuten jäljellä olevan kukkaron arvon tai kausilipun viimeisen voimassaoloajan. Online-validoinnissa käytetään taustajärjestelmän uusinta tilaa ja leimaustapahtumat voidaan käsitellä ja välittää eteenpäin välittömästi taustajärjestelmän sisällä. Myös tunnisteiden autentikointi voidaan tehdä palvelimen puolella. Online-validointiin vaaditaan kuitenkin toimiva tiedonsiirtoyhteys.

Tiedonsiirto-ongelmien lisäksi online-validoinnin ongelmaksi voi myös muodostua sen hitaus, sillä laitteen on odotettava taustajärjestelmässä tapahtuvan validoinnin tuloksia ennen kuin se voi ilmoittaa

käyttäjälle tiedon matkustusoikeuden kelpoisuudesta. Kohtalaisen pienilläkin viivästyksillä voi olla oleellinen vaikutus suurten ihmismäärien liikkumisen sujuvuuteen, etenkin ruuhkaisissa tilanteissa. Matkustajat ovat tottuneet korttipohjaisen arkkitehtuurin erittäin nopeisiin leimauksiin, joten huomattavasti pidemmältä tuntuva leimaus saattaa heikentää käyttökokemusta oleellisesti.

3.2 Offline-validointi

Offline-validoinnilla tarkoitetaan matkustusoikeuden validointia paikallisesti leimauslaitteella ilman taustajärjestelmään suoritettavaa rajapintakutsua. Koska validointi tapahtuu vain leimauslaitteen sisäisellä prosessoinnilla, ei taustajärjestelmään suoritettavaa rajapintakutsua tarvitse odottaa ja leimaustapahtuma on online-validointia nopeampi; yhtä nopea kuin korttipohjaisessakin järjestelmässä. Käyttäjän matkustusoikeus riippuu käyttäjän omistamista lipuista ja maksukyvyistä, joita säilytetään taustajärjestelmässä. Koska offline-validointi tapahtuu leimauslaitteilla, täytyy validoinnin mahdollistavat tiedot siirtää taustajärjestelmästä leimauslaitteille niin, että ne ovat välittömästi käytettävissä leimaushetkellä. Tietoja voidaan ladata laitteille ennakoon tiedonsiirtoverkon ylitse, tai niitä voidaan kuljettaa käyttäjän tunnistemedian avulla. Myös tunnisteen auktorisointi täytyy suorittaa päätelaitteella, ja tämä voidaan tehdä esimerkiksi tunnisteen RSA-allekirjoituksen varmennuksella liikkeellelaskijan julkisella avaimella.

Leimauslaitteiden ei välttämättä tarvitse tukeutua ainoastaan joko online- tai offline-validointiin, vaan laitteet voivat suorittaa validoinnin tilanteesta riippuen eri keinoilla. Mikäli online-validointi havaitaan riittävän nopeaksi, voidaan sitä käyttää pääsääntöisenä validointi-keinona, ja tukeutua offline-validointiin vasta tiedonsiirtoyhteyden ollessa poikki tai muissa ongelmatilanteissa. Toisaalta offline-toiminta tarjoaa nopeamman käyttökokemuksen ja sietää tiedonsiirtoyhteyksien arvaamatonta toimintaa paremmin kuin online-validointi, joten mikäli offline-validoinnista saadaan riittävän täsmällinen ja luotettava, kannattaa sitä hyödyntää ensisijaisesti.

Offline-validointia hyödyntävässä järjestelmässä tietoa leimauksesta ei välttämättä saada välitettyä taustajärjestelmään heti nousun tapahduttua tiedonsiirtoyhteyden ollessa poikki. Tästä voi seurata tilanne, jossa matkustaja nousee ajoneuvon kyytiin ja leimaa lippunsa onnistuneesti, jonka jälkeen lipuntarkastaja tarkastaa matkustusoikeuden ennen kuin tieto leimauksesta on ehtinyt välittyä taustajärjestelmään. Tämänkaltaisen tilanne on ongelmallinen, sillä tarkastaja ei tiedä, johtuuko puuttuva leimausmerkintä käyttäjän tarkoituksenmukaisesta vilpistä vai järjestelmän teknisestä ongelmasta. Lipuntarkastaja voi kuitenkin tarkastaa, että matkustajalla on matkustusoikeus järjestelmässä. Matkustajalle ikävien tilanteiden välttämiseksi lipuntarkastajille on kuitenkin tärkeä tiedottaa, että järjestelmä ei ole täysin reaaliaikainen leimaustapahtumien suhteen. Sama tilanne on mahdollinen myös online-validointia käyttävässä järjestelmässä, mikäli järjestelmässä käyttäjien matka sallitaan validoinnin epäonnistuessaa.

Tilanne saattaa tosin olla sekä matkustajalle että kuljettajalle läpinäkyvämpi, mikäli esimerkiksi päätelaite ilmoittaa yhteysvirheestä.

Tässä alaluvussa käydään läpi offline-validoinnin eri toteutusmahdollisuuksia. Toteutukset perustuvat siihen, että taustajärjestelmän tietoja, tai tiedoista johdettuja koostetietoja, saadaan taustajärjestelmästä kuljetettua leimauslaitteille joko tunnistemedian mukana tai tiedonsiirtoyhteyksien yli ennen leimaustapahtumaa. Offline-validoinnin ratkaisuja etsittiin toimialan kirjallisuudesta ja olemassa olevien lippu- ja maksujärjestelmien kuvauksista. Vaikka eri toteutusmallit yrittävät samaa asiaa, eli matkustusoikeuden offline-validointia, eivät mallit kuitenkaan ole toisiaan poissulkevia, vaan useita niistä voidaan mahdollisesti käyttää samanaikaisesti.

3.2.1 Validointiin tarvittavat tiedot

Matkustusoikeuden validointiin tarvittavat tiedot riippuvat lippu- ja maksujärjestelmän säännöistä. Käyttäjän omistamat liput voivat sisältää erilaisia oikeuksia, joiden perusteella liput kelpaavat tiettyinä ajankohtina tiettyihin liikkumispalveluihin. Käyttöön perustuvassa hinnoittelussa taas jäljellä oleva maksukyky määrittää käyttäjän oikeuden erilaisiin liikkumispalveluihin; matkustusoikeus riippuu siitä, riittääkö maksukyky leimauksen hintaan. Lippujen ja maksukyvyn lisäksi validoinnin tulokseen saattaa vaikuttaa myös esimerkiksi menneet nousut, joihin saattaa sisältyä tietynmittainen vaihto-oikeus.

Taustajärjestelmän tiedoista voidaan johtaa kevyemmät koostetiedot, joiden perusteella matkustusoikeus voidaan validoida. Yksinkertaisimmissa lippujärjestelmissä koostetiedoksi riittää binäärinen arvo; joko käyttäjällä on oikeus liikkumispalveluihin tai hänellä ei ole oikeutta. Tämä voi olla tilanne silloin, jos liput oikeuttavat aina koko palveluverkoston käyttöön, tai kaikki palvelut veloittavat saman hinnan. Mikäli kuitenkin leimauksen parametrit, kuten palvelun tyyppi, ajettava linja, vyöhykeväli tai matkan pituus tai ajankohta vaikuttavat validoinnin tulokseen, täytyy koostetiedot olla sellaisessa muodossa, että validoinnissa voidaan huomioida leimausparametrien määrittelemät eri tilanteet.

Leimauksen yhteydessä käyttäjälle voidaan haluta näyttää matkustusoikeuden hyväksynnän lisäksi myös tietoja lippujen tiloista tai jäljellä olevasta arvosta. Online-validoinnissa nämä tiedot saadaan rajapintakutsun vastauksessa, mutta offline-validoinnissa näytettävät tiedot täytyy sisällyttää offline-koostetietoihin.

3.2.2 Älykorteille tallennettavat offline-tiedot

Jos käyttäjien tunnistemediana toimii sekä kirjoitus- että lukuoperaatioita tukeva lippumedia, kuten älykortti, voidaan lippumedian mukana kuljettaa tunnisteen lisäksi offline-validointiin tarvittavia tietoja. Leimauksen yhteydessä nämä offline-tiedot voidaan lukea kortilta, ja niiden perusteella voidaan suorittaa matkustusoikeuden offline-validointi.

Keskeinen haaste älykorteilla säilytettävissä offline-tiedoissa on kuitenkin niiden ylläpidettävyyden vaikeus. Älykorttien tietoja voidaan muuttaa vain korttien ollessa lukijassa, joten esimerkiksi verkkopalveluissa tileille tehtyjä myyntejä ei saada suoraan päivitettyä kortin offline-tietoihin. Tämän takia kortin offline-tiedot saattavat vanhentua, eivätkä ne vastaa enää taustajärjestelmässä olevan tilin tilaa. Ylläpitämiseen voidaan käyttää esimerkiksi tekniikkaa, jossa lista muuttuneista offline-tiedoista ladataan leimauslaitteille etukäteen, ja tiedot päivitetään kortille seuraavan leimauksen yhteydessä. Jos laitteet käyttävät online-validointia ensisijaisena metodina, voidaan offline-tietojen päivitys suorittaa myös online-validointikutsun vastauksessa saatujen tietojen perusteella.

Yksi esimerkki tämänkaltaisesta toteutuksesta löytyy OSPT Alliance -yhdistyksen kehittämästä CIPURSE-määrittelystä, joka kuvaa lippumedian yhteentoimivuuteen ja tietoturvaan liittyviä avoimia määrittelyksiä. OSPT Alliance ehdottaa ”Customer Convenience Register” (CCR) -rekisterien käyttöä tunnistepohjaisen järjestelmän offline-toiminnan varmuuden lisäämiseksi. Kortilla olevaan CCR-rekisteriin tallennettaisiin tietoja matkustusoikeuden kannalta viimeisimmästä oleellisesta tapahtumasta, kuten arvon lataamisesta, ja tämän tiedon perusteella leimauslaitteet päättävät, liittyykö käyttäjän matkustusoikeuden hyväksyntään matala vai korkea lipputuloriski. Jos käyttäjä esimerkiksi lataa arvoa kortille itsepalvelukioskista juuri ennen matkan aloitusta, voivat leimauslaitteet lukea kortilta viimeisimmän lataustapahtuman ajankohdan, ja käyttää sitä matkustusoikeuden päättelyssä. Tällöin ladattu arvo on välittömästi käyttäjän käytettävissä jo ennen kuin tapahtumaa on ehditty käsittelemään taustajärjestelmässä. (OSPT Alliance, ei pvm)

Edellä kuvatussa toiminnallisuudessa kortin offline-tiedot voidaan päivittää vain kortin ollessa lukijalaitteessa kiinni. Tämä voi olla hyvä menettely, jos esimerkiksi arvon lataus kortille itsepalvelukioskilaitteella ennen matkaa on järjestelmässä yleistä. Jos taas pääasiallisena myyntikanavana toimii esimerkiksi verkkopalvelu, ei kortin tietoja voida päivittää eikä kortilla ylläpidettävistä offline-tiedoista saada samanlaista hyötyä.

3.2.3 Mobiili- ja paperilippujen offline-tiedot

Luvussa 2.3.2 esitettiin, miten mobiililaitteita voidaan käyttää tunnistepohjaisena lippumedianä. Mobiilisovellukset voivat hakea taustajärjestelmästä kryptograafisesti allekirjoitettuja koodeja, jotka sisältävät tilien ja lippujen tunnisteita. Koska koodit haetaan taustajärjestelmästä, voidaan niihin lisätä myös offline-validointiin tarvittavia tietoja, jotka välitetään leimauksessa lukijalaitteelle tunnisteiden mukana. Offline-tietoja voidaan hakea taustajärjestelmästä tietyn väliajoin, jotta ne eivät pääse vanhenemaan. Varsinkin jos mobiilisovelluksessa käytetään jatkuvaa koodien uudelleenhakemista kopioinnin ehkäisemiseksi, taustajärjestelmästä haettuun koodiin lisätyt offline-tiedot saadaan varmasti pysymään ajan tasalla taustajärjestelmän tilan kanssa. Tiedot saattavat kuitenkin olla taustajärjestelmän

tilasta jonkin verran jäljessä, riippuen koodigeneroinnissa käytetystä jaksonajasta ja myös mobiililaitteen tiedonsiirtoyhteydestä, joka saattaa estää uuden koodin hakemisen taustajärjestelmästä.

Jos leimaus suoritetaan pelkällä tilin tunnisteella, täytyy offline-tietoista olla johdettavissa koko tiliä koskeva matkustusoikeus. Jos taas leimaus kohdennetaan tiettyyn lippuun, ja mobiilisovellus on hakenut taustajärjestelmästä kyseisen lipun tunnisteeseen sisältävän koodin, on tämän tunnisteeseen olemassaolo jo oikeastaan merkki matkustusoikeudesta. Lipun tunnisteeseen yhteyteen voidaan kuitenkin liittää lipun tietoja, jotta tiedot voidaan näyttää käyttäjälle leimauksen yhteydessä.

Myös luvussa 2.3.3 esitettyihin kuviokoodillisiin paperilippuihin lisätään taustajärjestelmässä luotu ja allekirjoitettu tunnisteeseen sisältävä koodi. Tähänkin koodiin voidaan lisätä offline-tietoja samalla tavalla kuin mobiilisovelluksissa käytettyihin koodeihin. Paperisten lippujen koodeja ei voida kuitenkaan generoida uudestaan, joten offline-tietojen vanhenemista ei voida estää ja tiedot vastaavat taustajärjestelmän lipun luontihetken tilaa. Lipputuotetyypistä riippuen vanhentuneisiin offline-tietoihin perustuva validointi saattaa johtaa virheelliseen tulokseen. Esimerkiksi yhteen matkaan oikeuttavaa kertalippua voitaisiin käyttää useita kertoja, sillä tieto sen käytöstä ei välity offline-tietoihin.

Mobiili- ja paperilippujen leimauksessa koodien tiedonsiirtoon käytetään usein optisesti luettavia kuviokoodeja. Koodien tietomäärät halutaan pitää pieninä, sillä liian suuri määrä dataa heikentää optisen luvun onnistumisen mahdollisuutta. Tunnisteeseen yhteydessä välitettyjen offline-tietojen datamäärä ei siis saa olla liian suuri.

3.2.4 Päätelaitteille jaettavat offline-listat

Offline-validointiin tarvittavia tietoja voidaan myös siirtää taustajärjestelmästä tiedonsiirtoverkkojen ylitse päätelaitteille valmiiksi tulevia validointeja varten. Leimauksen yhteydessä näistä offline-tiedoista haetaan luettua ja autentikoitua tunnistetta vastaavat tiedot, joiden avulla matkustusoikeus validoidaan paikallisesti päätelaitteella. Taustajärjestelmän tiedoista siis replikoidaan matkustusoikeuden validointiin tarvittavia tietoja päätelaitteiden muistiin. Suoraan taustajärjestelmästä ladattujen tietojen etuina ovat, että tiedot voidaan pitää ajan tasalla taustajärjestelmän kanssa, ja että offline-validointi ei ole riippuvainen tunnistemediasta. Tässä työssä leimauslaitteille ladattuja offline-validointiin käytettäviä tietoja kutsutaan *offline-listaksi*.

Listojen tietosisältö riippuu matkustusoikeuden validointiin tarvittavista tiedoista, joita kuvattiin luvussa 3.2.1. Koska offline-listojen tiedot jaetaan leimauslaitteille taustajärjestelmästä, voidaan eri laitteille jakaa eri tietosisältöä. Tällöin esimerkiksi eri liikkumispalveluiden laitteille voidaan jakaa vain niiden validointi-toimintaan tarvittavia tietoja. Tämä voi esimerkiksi tarkoittaa sitä, että laitteille ei tarvitse ladata ollenkaan lippujärjestelmän tuoteparametrejä, joiden avulla kyseiseen palveluun kelpaavat liput voidaan

suodattaa kaikkien lippujen joukosta, vaan suodatus tehdään jo taustajärjestelmässä listojen jakelun yhteydessä.

Yksi offline-listan muoto on sallittujen tunnisteiden lista, eli whitelist¹. Listalle tuodaan kaikkien niiden tilien tunnisteet, joilla on kelpaava matkustusoikeus. Offline-validointi suoritetaan tällöin yksinkertaisesti tarkastamalla, löytyykö lukijaan asetettu tunniste whitelistilta. Täsmällisen validoinnin suoritukseksi matkustusoikeus ei saa tällöin riippua vasta leimaushetkellä tiedettävistä parametreista, vaan se täytyy olla pääteltävissä etukäteen taustajärjestelmässä.

Mikäli tietyllä leimauslaitteella tehdyn validoinnin leimausparametrit, kuten vyöhykeväli tai matkan pituus, tiedetään vasta leimaushetkellä, ei validoinnin tulokselle voida muodostaa yksiselitteistä arvoa etukäteen taustajärjestelmässä. Yksinkertainen whitelist-toteutus ei tällöin riitä täsmällisen offline-validoinnin suorittamiseksi. Kattavampaa validointia varten offline-listalle täytyy tuoda luvussa 3.2.1 mainittuja koostetietoja, joiden pohjalta validointi voidaan tehdä eri leimausparametriyhdistelmille. Leimauksessa tunnistetta koskevat koostetiedot luetaan listalta, ja matkustusoikeus johdetaan niiden avulla. Leimauslaitteelle tarvitaan tällöin siis yksinkertaista whitelist-ratkaisua enemmän logiikkaa, jonka avulla päättely voidaan tehdä. Myös siirrettävät tietomäärät ovat suurempia.

Koska käyttäjien matkustusoikeus voi muuttua uusien myynti- ja leimaustapahtumien myötä, täytyy leimauslaitteiden offline-listoja päivittää jatkuvasti vastaamaan taustajärjestelmän uusinta tilaa. Matkustusoikeus voi muuttua myös ajan kuluessa, esimerkiksi kausilipun voimassaolon päättyessä, ja myös tämänkaltaiset muutokset matkustusoikeuksiin pitää huomioida ja päivittää offline-listoille. Tieto voimassaolosta voidaan vaihtoehtoisesti myös sisällyttää offline-listan tietoihin ja huomioida leimauslaitteen validointilogikassa. Tapahtuneet muutokset koskevat aina tiettyä tiliä, joten päivityksen yhteydessä koko offline-listaa ei tarvitse hakea uudestaan, vaan riittää, että päätelaitteille ladataan edellisen päivityksen jälkeen tapahtuneet muutokset. Lippujärjestelmän käyttäjien määrästä ja offline-tietojen muodosta riippuen tietomäärät saattavat olla suuria, joten leimauslaitteilta vaaditaan tietyn verran suorituskykyä ja muistia offline-listojen hallintaan.

Luvussa 2.4.3 mainittu INIT-yritys kertoo (2018, s. 3) toteuttaneensa tunnistepohjaisen offline-validoinnin päätelaitteille ladattavien listojen avulla. INITin lippujärjestelmän toteutus on käytössä Turun seudulla toimivassa Föli-joukkoliikennejärjestelmässä, ja Fölin palvelu- ja myyntiohjeessa (2017, s. 16) kuvataan sen käyttöä. Ohjeessa mainitaan rahastuslaitteelle ladattava whitelist-luettelo, jota päivitetään noin minuutin välein. Viimeisin päivitysaika näytetään kuljettajalle rahastuslaitteen käyttöliittymässä, minkä avulla mahdolliset ongelmatilanteet tiedonsiirron kanssa voidaan havaita. Mikäli listaa ei saada päivitettyä, saattaa validointi hylätä juuri ennen nousua hankitun matkustusoikeuden. Näissä tilanteissa kuljettajaa

¹ Alan kirjallisuudessa whitelist-termiä käytetään yleisesti tarkoittamaan myös laajempia offline-tietoja sisältäviä listoja. Tässä työssä whitelist-termillä tarkoitetaan kuitenkin vain hyväksytyjä tunnisteita sisältävää listaa, ja laajempaa sisältöä sisältävistä listoista käytetään termiä offline-lista.

ohjeistetaan myymään matkustajalle kertalipun, jonka matkustaja voi saada korvauksen asiakaspalvelussa asioimalla.

Myös luvussa 2.4.2 mainittuun Samtrafikenin ylläpitämään lippustandardiin kuuluu tunnistepohjaisiin lippuihin liittyviä määräytyksiä, joissa kuvaillaan myös offline-validointiin liittyvää toiminnallisuutta. Samtrafikenin määrätyksissä matkustusoikeuksia kuvataan ”Mobile ticket bundle” (MBT) – tietopakettien avulla, joiden tietosisällön perusteella matkustusoikeus voidaan validoida. Offline-toiminnallisuutta varten validoinnin suorittavalle laitteelle voidaan ladata lista käyttäjien tunnisteisiin liittyvistä MBT-paketeista. Jokaisella listan rivillä on versionumero, ja taustajärjestelmästä voidaan hakea tietyn version jälkeen tapahtuneet muutokset rajapintakutsulla. Rivit sisältävät versiotiedon lisäksi tunnisteiden julkisen avaimen, tunnisteeseen liittyvän MTB-paketin, sekä rivin voimassaoloajan. Validointi tapahtuu lukemalla listalta tunnisteeseen liittyvät MBT-paketit, ja suorittamalla niille standardin mukaisen validoinnin. Samtrafikenin määrätyksessä samoja MBT-paketteja voidaan kuljettaa myös lippumedian, kuten älykorttien, mukana, joten validointi voi tapahtua korttipohjaisten ja tunnistepohjaisten lippujen osalta samalla tavalla, ainoastaan tiedon lähteenä on nyt lippumedian sijaan laitteelle ladattu offline-lista. Määrityksissä mainitaan myös tilanne, jossa tunnistepohjaiset liput ostetaan käyttäjien tunnistemediana käyttämien laitteiden, eli esimerkiksi puhelimien, avulla. Näissä tilanteissa suositellaan, että sama MBT-paketti säilytetään taustajärjestelmän lisäksi myös tunnistemediana toimivalla laitteella, josta sitä voidaan tarvittaessa käyttää validointiin offline-tilanteissa. Tämä vastaa edellisessä alaluvussa 3.2.3 kuvattua toteutusmallia, ja samassa järjestelmässä voidaan siis hyödyntää molempia offline-validoinnin toteutusmalleja. (Samtrafiken, 2018)

3.2.5 Estolistat

Yksi offline-listan muoto on *estolista* (engl. deny-list blacklist), joka sisältää vain ne tunnisteet, joilla matkustaminen on estetty. Leimauslaitteilla tapahtuva offline-validointi voi tällöin oletusarvoisesti hyväksyä luetun ja autentikoidun tunnisteiden, ellei sitä löydy estolistalta. Estolistan sisältöä hallitaan taustajärjestelmästä käsin, ja listojen sisältöä päivitetään jatkuvasti leimauslaitteille.

Estolista voi soveltua erityisesti sellaisiin tilanteisiin, joissa käyttäjän matkustusoikeus riippuu lippujärjestelmäalustan ulkopuolisista tiedoista, eikä matkustusoikeutta tiedetä ennen leimausta. Matkustajat voidaan hyväksyä tällöin oletusarvoisesti ensimmäiselle matkalle, mutta taustajärjestelmä tarkastaa matkustusoikeuden ensimmäisen leimauksen jälkeen, ja estää tulevat matkat estolistan avulla siihen saakka, kunnes matkustaja on korvannut velaksi jääneen matkan. Tämä on tilanne esimerkiksi silloin, jos käyttäjän maksukyky tulee järjestelmään kytketystä maksutavasta, jonka maksukyky voi muuttua ilman että tietoa tästä saadaan lippujärjestelmään saakka.

UK Finance yhdistys suosittelee estolistojen käyttöä cEMV-pankkikorteilla tehtyjen matkojen oikeuksien validoinnissa. Luvussa 2.3.4 esitettiin pankkikorttien käyttämisen 3 eri mallia, joista mallissa 2 käyttäjien tekemät matkat välitetään taustajärjestelmään, joka hoitaa veloituksen EMV-maksuverkosta. Vähittäiskaupassa korttien maksukyky vahvistetaan yleensä maksutransaktion online-auktorisoinnilla. Mallin 2 mukaisessa toteutuksessa maksukyvyyn vahvistusta ei kuitenkaan suositella suoritettavan leimauslaitteella, vaan taustajärjestelmä tarkastaa sen tietyissä tilanteissa EMV-maksuverkostosta, ja maksukyvyyn puuttuessa lisää kortin estolistalle. Estettyjä kortteja voidaan yrittää palauttaa käyttöön yrittämällä veloittaa velaksi jäänyttä summaa automaattisesti maksuverkostosta, tai käyttäjän sovittaessa velan asiakaspalvelun tai verkkopalveluiden avulla. Tämänkaltaiseen toteutukseen liittyy lipputuloriskiä, sillä käyttäjät voivat vielä matkustaa maksukyvyyn puuttuessa sen aikaa, ennen kuin kortti saadaan estolistalle. UK Finance esittää, että lipputuloriskit jaetaan liikkumispalvelun operaattoreiden ja korttien liikkeellelaskijoiden välillä riskinjakosopimuksilla. (UK Finance, 2017)

Vasta leimauksen jälkeen tehtävästä maksukyvyyn vahvistuksesta aiheutuvasta lipputuloriskistä käytetään termiä "first ride risk". Pankkikortteja käyttävissä järjestelmissä riski on kuitenkin havaittu hyvin vähäiseksi. Suurin osa pankkikortteja käyttävistä henkilöistä lataavat tileihin nopeasti lisää arvoa, ja velaksi jääneet matkat saadaan yleensä perittyä automaattisella takaisinperintäprosessilla. (Steering & Douglas, 2016)

4 Offline-validointi Pusatecin lippujärjestelmäalustassa

Tässä luvussa esitetään Pusatec Oy:n tunnistepohjaisen lippujärjestelmäalustan nykyistä tilannetta ja offline-validoinnin toteutukselta haluttua toiminnallisuutta. Luvun lopuksi listataan toteutettavaa offline-validointia koskevat vaatimukset. Luvussa kuvattua toiminnallisuutta ja asetettuja vaatimuksia käytetään pohjana seuraavien lukujen offline-validoinnin toteutuksessa.

4.1 Nykyinen lippujärjestelmäalusta

Pusatec Oy on joukkoliikenteen tietoteknisiä ratkaisuja tuottava yritys. Se kehittää ja ylläpitää lippujärjestelmäalustaa, joka on käytössä useassa Suomen eri joukkoliikennejärjestelmissä. Pusatec kehittää joukkoliikenteen tarpeisiin myös laiteratkaisuja, esimerkiksi kuljettajapäätelaitteita ja erillisiä lukijalaitteita.

Kokonaisuudessaan joukkoliikennejärjestelmien käyttämät tietojärjestelmät voivat olla hyvin laajoja, useiden eri toimijoiden ylläpitämistä tietojärjestelmistä koostuvia kokonaisuuksia, ja Pusatec voi vastata vain tietyistä järjestelmäkokonaisuuden osista. Eri tietojärjestelmät keskustelevat keskenään rajapintojen avulla. Pusatecin lippujärjestelmässä toimivia tuotteita myydään useissa muiden toimittajien ylläpitämissä palveluissa, ja Pusatecin lippujärjestelmä tarjoaa rajapinnat tuotteiden myyntiin, matkustusosoikeuden validointiin ja tapahtumatietojen välitykseen.

Pusatecin lippujärjestelmäalusta on aikaisemmin toiminut korttipohjaisella arkkitehtuurilla. Toimialan muuttuneen ympäristön vuoksi järjestelmään on kehitetty korttipohjaisen mallin rinnalle myös tunnistepohjaisia matkatilejä ja lipputuotteita tukevaa toiminnallisuutta. Tunnistepohjaisuutta varten on perustettu uusi tietojärjestelmän osa, joka vastaa matkatilien ja lippujen säilönnästä, tunnistepohjaisesta logiikasta sekä rajapintojen tarjoamisesta päätelaitteille, asiakaspuolen ohjelmille, ja muille tietojärjestelmille. Suuri osa toiminnallisuudesta, kuten lipputuotteiden konfiguraatio sekä tapahtumien seuranta, välitys ja raportointi tapahtuu kuitenkin yhtenäisesti korttipohjaisen järjestelmän kanssa, ja samassa lippujärjestelmässä voidaan ottaa käyttöön sekä korttipohjaisia että tunnistepohjaisia tuotteita. Käytössä olevat älykortit sisältävät korttipohjaisia tuotteita, mutta ajan myötä ainakin osa korteista halutaan muuttaa tunnistepohjaisiksi. Tällöin taustajärjestelmään luodaan asiakkaalle tili, kortille tallennetaan tilin tunniste, ja kortilla olevat liput siirretään taustajärjestelmään perutetulle tilille.

Järjestelmässä käytettäviä lippumedioita ovat älykortit, mobiililaitteet ja paperiset QR-koodilla varustetut liput. Tunnistepohjaisina näistä toimivat tällä hetkellä mobiililiput, paperiset QR-koodiliput ja osa älykorteista. Tapahtuneet leimaukset välitetään taustajärjestelmään mahdollisimman nopeasti

tiedonsiirtoyhteyksien niin salliessa. Järjestelmässä tuetaan erilaisia tuotteita, kuten kerta- sarja- ja kausilippuja. Lippuihin voi liittyä tietynmittainen vaihtoaika ja matkustusvyöhykkeet. Tiliin kytkettävät maksutavat ja arvon lataukset tulevat muiden toimijoiden järjestelmäintegraatioiden kautta, ja Pusatecin järjestelmässä niitä käsitellään vain käyttäjän tilin maksukykyä.

Lippujärjestelmälustaa käytetään useassa eri joukkoliikennejärjestelmässä, joilla on yleensä toisistaan erilliset lipputuotteet. Lippujen kelpaavuutta eri tilanteissa hallitaan leimauskanavien avulla. Kanavat määrittävät leimauksessa kelpaavat lipputuotteet, ja kanaville on asetettu niiden toimintaa ohjaavat tuoteparametrit. Matkustusoikeuden validointi suoritetaan aina tietyn leimauskanavaan sidottujen sääntöjen mukaisesti.

Tunnistepohjaiset mediat validoidaan tällä hetkellä ensisijaisesti online-validoinnilla. Mobiili- ja paperilippujen taustajärjestelmästä haettujen tunnisteiden yhteyteen liitetään kuitenkin muitakin lipun tietoja, joita voidaan käyttää aluvun 3.2.3 esittämällä tavalla myös offline-validointiin. Validoinnin suorittavilla päätelaitteilla on järjestelmän tuoteparametrit, joita voidaan käyttää myös offline-validoinnin logiikassa, mutta tämä ei kuitenkaan välttämättä ole tilanne muiden toimittajien päätelaitteilla.

4.2 Toteutuksen alustava suunnitelma

Matkustusoikeuksien validoinnin tulee olla nopeaa ja luotettavaa. Varsinkin kun pitkään toiminnassa olleen ja luotettavan korttipohjaisen järjestelmän älykortteja siirretään tunnistepohjaisiksi, online-validointi ei välttämättä riitä halutun käytettävyytensä ja lipputuloriskin saavuttamiseksi. Suuri osa Pusatecin päätelaitteista on mobiilitiedonsiirtoyhteyksien päässä, ja tiedonsiirtoyhteyksiin liittyvä epävarmuus ja hitaus aiheuttavat haasteita tunnistepohjaiselle järjestelmälle. Mikäli nykyinen online-validointi havaitaan liian hitaaksi tai epäluotettavaksi, halutaan toimintavarmuutta lisätä offline-validoinnin avulla.

Offline-validointi on alustavasti suunniteltu toteuttaa luvussa 3.2.4 esitettyjen offline-listojen avulla. Listojen avulla matkustusoikeus voidaan validoida kaikille tunnistemedioille yhtenäisesti. Luvussa 2.4.2 esitettiin lippujärjestelmien yhteentoimivuuden liittyviä huomioita. Yksi syy tunnistepohjaisuuteen siirtymiseen on juuri teknisen yhteentoimivuuden helppouden lisääminen. Tämä otetaan huomioon myös offline-validoinnin osalta, ja muille toimijoille halutaan tarjota mahdollisuus offline-validoinnin hyödyntämiseen myös toimijoiden omilla päätelaitteilla ja tietojärjestelmillä. Tämän vuoksi mahdollinen offline-listojen jakelurajapinnan toteutus tulee tehdä yleisillä tekniikoilla ja hyvillä käytännöillä.

4.3 Vaatimukset

Luetellaan seuraavaksi offline-validoinnin toteutusta koskevat vaatimukset. Nämä asettavat korkean tason odotuksia toteutettavalle järjestelmälle, ja jotta toteutusta voidaan pitää onnistuneena,

toteutuksen on vastattava vähintään näitä odotuksia. Vaatimukset on asetettu sillä olettamuksella, että online-validointi havaitaan riittämättömäksi.

I. Tunnistemedian riippumattomuus

Offline-validointi tulee olla mahdollista käytettävästä tunnistemediasta riippumatta. Validoinnissa oletetaan, että päätelaite tietää ja on autentikoinut tilin yksilöivän tunnisteen ennen validointia.

II. Muuttuneiden matkustusoikeuksien reaaliaikainen huomiointi

Tilien muuttuvat matkustusoikeudet tulee olla validoitavissa oikein mahdollisimman nopeasti muutosten jälkeen. Tilanteessa, jossa käyttäjä ostaa lippuotteensa verkkokaupasta esimerkiksi mobiililaitteellaan, validoinnin tulisi hyväksyä lippu 2 minuutin kuluessa. Tästä aikavaatimuksesta voidaan poiketa mahdollisissa ongelmatilanteissa, kuten tiedonsiirtoyhteyksiin liittyvissä ongelmissa. Myös leimauslaitteiden käynnistyessä niillä tehtävien validointien tulee onnistua 5 minuutin kuluttua.

III. Avoimuus muille toimijoille

Lippujärjestelmän tilien matkustusoikeuksien validointi tulee olla mahdollista muille toimijoille. Offline-validointi tulee toteuttaa yleisillä, mahdollisimman useita eri laiteratkaisuja ja ympäristöjä tukevilla tekniikoilla. Laiteratkaisuissa huomioidaan kuitenkin vain kohtalaisen modernit ratkaisut; ei ole tarvetta tukea selvästi vanhentuneita laiteratkaisuja. Toteutuksessa halutaan käyttää moderneita, luotettavia ja tietoturvallisia tekniikoita, jotta jakelujärjestelmän toteutus on pitkäikäinen, ja se tukee myös tulevaisuudessa kehitettäviä tietojärjestelmiä ja laitteita.

IV. Tietomäärien rajoittaminen

Validoinnissa päätelaitteille mahdollisesti välitettävien tietojen määrää tulee rajoittaa niin, että tiedot voidaan siirtää nykyisten mobiiliverkkojen ylitse, ja ne saadaan mahtumaan erilaisten laiteratkaisujen muistiin. Tietomäärät halutaan myös muuten minimoida vain siihen tietomäärään, mikä on järjestelmän toiminnan kannalta tarpeellista.

V. Tehokkuus, skaalautuvuus ja luotettavuus

Offline-validoinnin toteutuksen tulee pystyä toimimaan suurella määrällä leimauslaitteita. Toteutuksessa halutaan varautua myös lippujärjestelmän toiminnan laajentumiseen. Offline-validointiin liittyvän toiminnallisuuden tulisi skaalautua tehokkaasti järjestelmän käytön kasvaessa. Resurssivaatimusten tulee skaalautua palveltavien laitteiden määrän kasvaessa mahdollisimman lineaarisesti. Validoinnin tulee toimia niin luotettavasti, että sen toiminnasta syntyy mahdollisimman vähän ongelmia käyttäjille ja muille osallisille tahoille, ja siksi validointien tulosten tulee olla riittävän täsmällisiä.

Käytössä varaudutaan tyyppillisten Suomen joukkoliikennejärjestelmien aiheuttamaan rästitukseen. Samassa lippujärjestelmässä voi olla useita satoja tuhansia matkustusoikeuden omistavia käyttäjiä, ja matkustusoikeuksia muuttavia tapahtumia voi päivän tapahtua myös satoja tuhansia. Taustajärjestelmän tulee varautua samanaikaisesti palvelemaan useita tuhansia päätelaitteita.

VI. Toteutus osaksi nykyistä järjestelmää

Pusatec haluaa pitää yhtenäisen linjan käytettyjen tekniikoiden, toteutusratkaisujen ja tuotantoympäristöjen osalta, ja siksi myös offline-validoinnin toteutuksessa ei tule poiketa nykyisistä tekniikoista ilman riittävää syytä. Offline-validointiin liittyvät taustajärjestelmän toiminnot halutaan toteuttaa osaksi nykyistä tunnistepohjaista lippujärjestelmäalustaa.

5 Offline-validoinnin toteutuksen suunnittelu

5.1 Ympäristön asettamat rajoitukset

Tässä aluvussa tarkastellaan nykyisen tietojärjestelmäkokonaisuuden asettamia rajoituksia toteutukselle. Rajoituksia asettavat päätelaitteiden resurssit ja tiedonsiirtoyhteydet, lippujärjestelmän säännöistä riippuva validointiin tarvittavien tietojen muoto, sekä tämänhetkinen tunnistepohjainen taustajärjestelmä. Aluvussa käsitellään myös sitä, onko offline-validoinnille yläpäättään tarvetta, vai riittääkö pelkkä online-validointi halutun toiminnallisuuden tuottamiseen.

5.1.1 Päätelaitteet

Offline-validointi toteutetaan ensisijaisesti Pusatecin uusimman sukupolven päätelaitteille, joita ovat kuljettajan maksupäätte PT10eX ja lukijalaitte PT11eX. Vanhojen sukupolvien laitteita on edelleen operoinnissa, mutta niitä ei huomioida offline-validoinnin toteutuksessa.



Kuva 2 - Pusatec PT10eX (vasemmalla) ja PT11eX (oikealla) ajoneuvolaitteet

(Lähde <https://www.pusatec.fi>)

PT10eX on tarkoitettu ensisijaisesti kuljettajan käyttöön. Siihen on kuitenkin mahdollista liittää QR-koodi- ja RFID/NFC-lukija matkustajien lippumedian leimaamista varten. PT11eX -laitetta käytetään ensisijaisesti lippujen leimaamiseen, ja se voidaan sijoittaa kuljettajapäätettä joustavammin. Ajoneuvolaitteiden tekniset tiedot on listattu liitteessä 1. Validoinnin toteutuksessa tulee näiden laitteiden lisäksi huomioida kuitenkin mahdollisimman helppo käyttöönotto myös muiden valmistajien laitteille ja tietojärjestelmille, kuten vaatimuksessa III todetaan.

Pusatecin leimauslaitteet käyttävät Linux-käyttöjärjestelmää 64-bittisen ARM-arkkitehtuurin päällä. Ohjelmakoodit kirjoitetaan suurimmaksi osaksi C++ tai Rust -ohjelmointikielillä, jotka soveltuvat hyvin järjestelmäohjelmien kirjoittamiseen. Koska varsinkin C++ -kieli on todella laajasti käytetty kieli, oletetaan, että muiden laitevalmistajien mahdollinen toteutus voi hyödyntää Pusatecin laitteiden ohjelmia vastaavia ohjelmaratkaisuja.

Laiteresurssien osalta prosessointikyky ei todennäköisesti tuota ongelmia offline-validoinnin osalta, sillä validointiin ei tarvita nykyistä käyttöä monimutkaisempaa tai raskaampaa prosessointia. Muisti sen sijaan täytyy ottaa tarkemmin huomioon. Laitteilla on dynaamista muistia käytössä 1 Gt verran, mutta sitä ei voida kokonaan varata offline-listan käyttöön. Tämän lisäksi laitteella on massamuistia, joka on laajennettavissa 256 Gt asti. Käyttötapaus huomioiden tietoon suoritettavien operaatioiden latenssi ei vaikuta oleellisesti järjestelmän toimintaan, ja offline-validointiin tarvittavat tiedot voidaan tarvittaessa tallentaa massamuistiin. On myös edullista, että tiedot säilyvät laitteen uudelleenkäynnistyksen ylitse, jotta niitä ei tarvitse siirtää laitteelle alusta asti jokaisen uudelleenkäynnistyksen yhteydessä.

5.1.2 Tiedonsiirtoyhteydet

Päätelaitteet ovat suurimmaksi osaksi Pusatecin hallitsemien mobiilitiedonsiirtoyhteyksien päässä. Jos tiedonsiirtoyhteydet toimivat riittävällä luotettavuudella, ja verkon yli suoritettavat kutsut taustajärjestelmään ovat riittävän nopeita, voidaan kaikki validoinnit toteuttaa online-validoinnilla, eikä tarvetta offline-validoinnille ole. Päätelaitteet tukevat 3G ja 4G -yhteyksiä, ja yhteyksissä käytetyt liittymät voidaan valita tarpeen mukaan, mutta ylimääräisiltä kustannuksilta on hyvä välttyä.

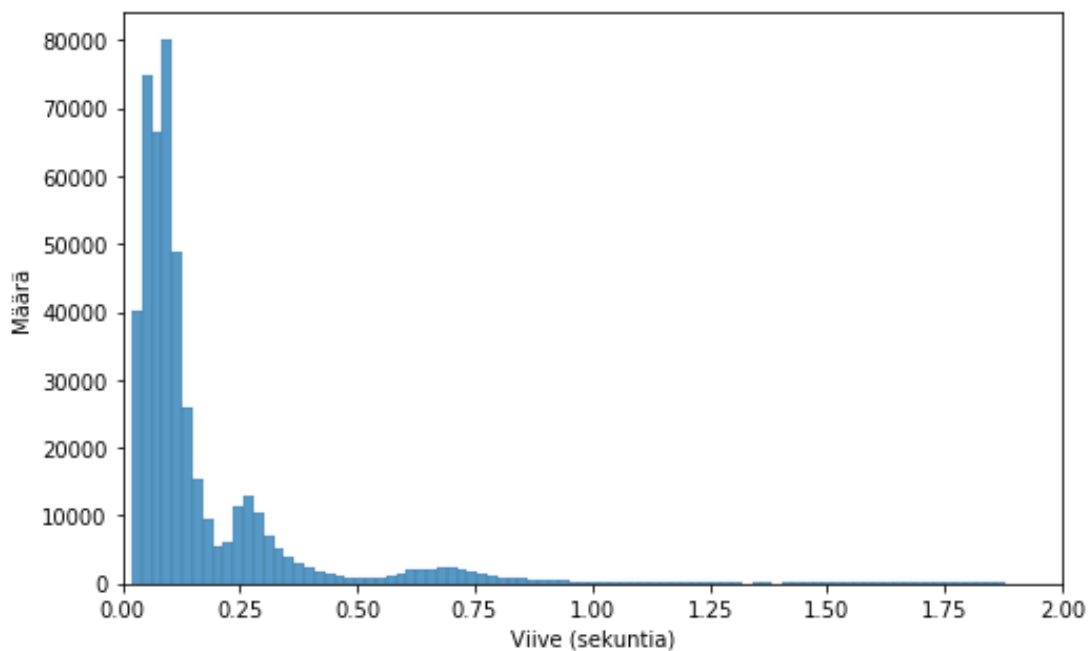
Mikäli offline-validointi toteutetaan taustajärjestelmästä laitteille ladattujen offline-listojen avulla, ovat laitteiden tiedonsiirtoyhteyksien nopeus ja luotettavuus, eli se kuinka suuren osan ajasta yhteys toimii, keskeisiä rajoitteita myös listojen jakelun kannalta. Yhteyksien nopeus rajoittaa offline-listan muutosten jatkuvaa välittämistä laitteille. Tiedonsiirron estyminen puolestaan johtaa offline-listojen vanhentumiseen, ja saattaa aiheuttaa virheellisiä validointituloksia.

Ajossa olleiden laitteiden nauhoittamasta lokimateriaalista voidaan kerätä tietoa verkkoyhteyksien tiloista, tapahtuneista yhteyshäiriöistä ja laitteiden suorittamista validointikutsuista. Pusatecin ajoneuvolaitteet kirjoittavat lokiin säännöllisesti tiedonsiirtoyhteyden tilasta, laitteen sijainnista,

ajettavasta linjasta sekä muista tiloista, joita voidaan käyttää analyysiin. Työn aikana lokien perusteella tehtiin analyysiä online-validointien tämänhetkisestä tilanteesta, ja offline-listojen jakeluun liittyvistä rajoituksista. Analysoinnissa huomioitiin vain linja-ajon aikana kerätyt lokit, jotta data vastaa todellisia tilanteita mahdollisimman hyvin. Analyysia varten kirjoitettiin ohjelma, joka käy lokirivejä järjestyksessä läpi ja poimii samalla lokeista tutkimuksen kannalta oleellisia tapahtumia ja lukuarvoja ylös.

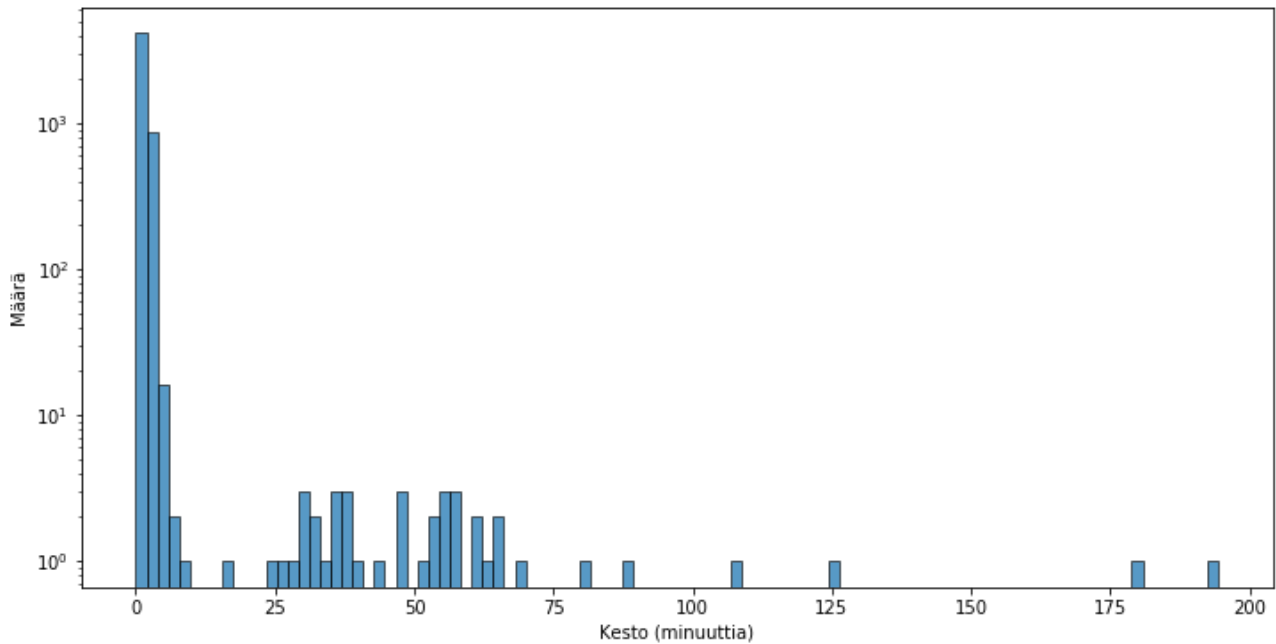
Lokien analysointiin käytettiin 66 eri Jyväskylän alueella ajavan ajoneuvolaitteen lokeja heinä-elokuulta 2020. Lokit kattavat yhteensä noin 16 484 tuntia linja-ajoa. Analyysi tehtiin kolmella eri tavalla. Ensimmäisessä tavassa lokeista etsittiin kaikki ping-komennot, joita laitteet ovat suorittaneet säännöllisin väliajoin. Ping-komentoja löydettiin 474 897 kappaletta, joista 472 863, eli 99,57 % oli onnistunut saamaan yhteyden. Kuvassa 3 on esitetty Ping-komentojen viiveiden jakauma.

Toisessa tavassa laitteen lokeja käytiin aikajärjestyksessä läpi, ja samalla pyrittiin seuraamaan laitteen senhetkistä tilaa. Lokiriveistä havaittiin tiedonsiirtoyhteyden tilan muutokset, jotka päivitettiin analyysin suorittavan ohjelman muistiin ja kirjattiin ylös erilliseen tiedostoon. Näistä muutoksista johdettiin lista jaksoja, joiden aikana tiedonsiirtoyhteys on pysynyt joko yhdistettynä tai katkenneena. Analyysissä havaittiin, että verkkoyhteys oli ollut yhdistettynä 99,42 % kokonaisajasta. Yhteyshäiriöitä löydettiin yhteensä 5 060 kappaletta. Kuvassa 4 on visualisoitu tapahtuneiden katkojen kestojen jakauma. Kuvassa 5 on sama visualisointi, mutta vain alle 6 minuutin katkojen osalta. Kuvaajista nähdään, että valtaosa katkoista oli kestoltaan alle 4 minuuttia, ja tästä pidempiä katkoja voidaan pitää poikkeuksellisina, vaikkakin muutamia todella pitkiäkin yhteyshäiriöitäkin löytyy tuloksista.

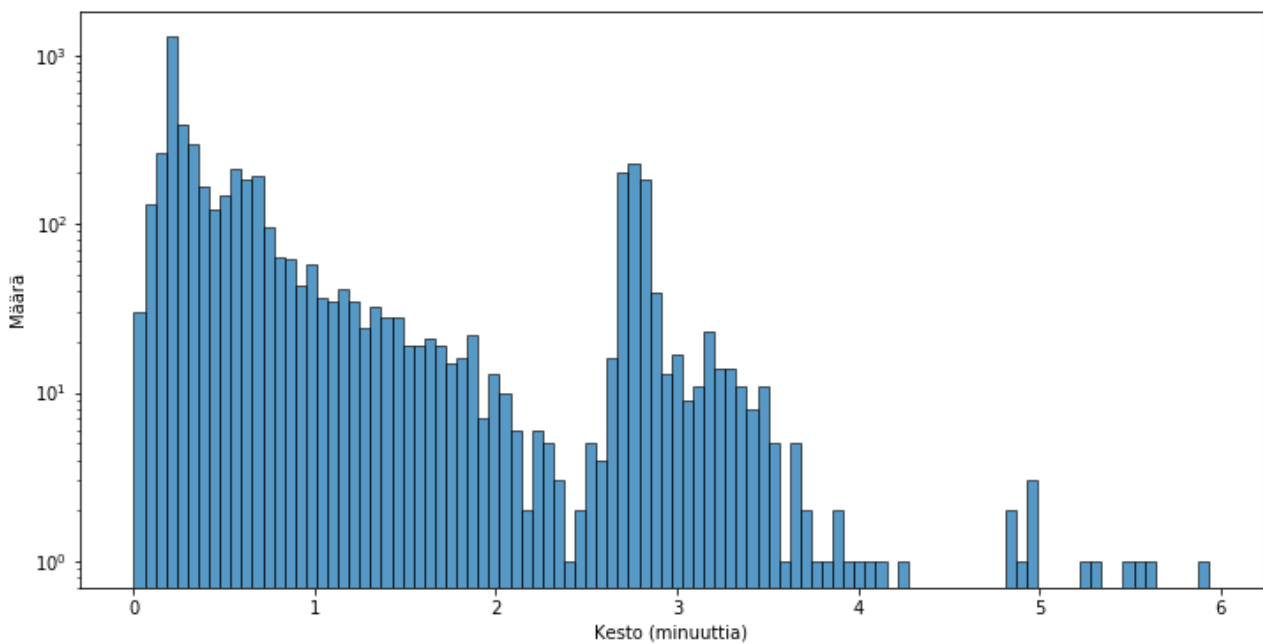


Kuva 3 - Ping-komentojen viiveiden jakauma

Liitteen 2 kuvassa on esitetty yhteyskatkosten geograafiset sijainnit Jyväskylän alueen kartalla. Kuvasta nähdään, että yhteyskatkot ovat usein keskittyneet tietyille alueille, kuten kaupungin keskustaan. Nämä keskittymät saattavat johtua kiireellisistä linjareiteistä, joilla on paljon liikennettä ja kerättyä dataa, tai jonkinlaisesta ongelmallisesta sijainnista kuten katvealueesta. Lyhyitä yhteyskatkoja tapahtuu kuitenkin

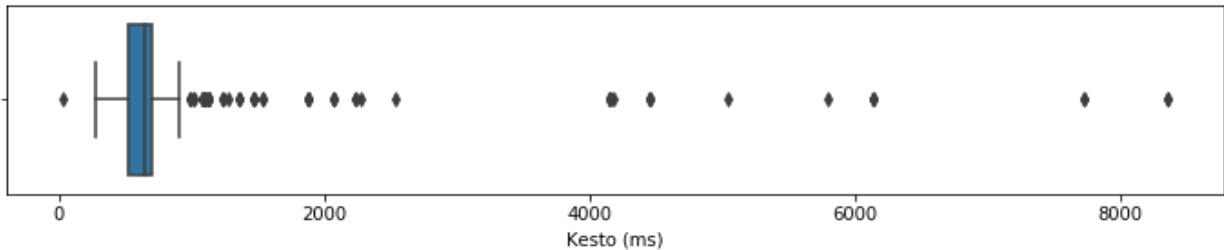


Kuva 4 - Yhteyskatkojen kestojen jakauma



Kuva 5 - Yhteyskatkojen kestojen jakauma, vain alle 6 minuutin katkot

< 2 s	94 %
< 1 s	90 %
< 800 ms	85 %
< 600 ms	43 %
< 400 ms	5 %



Kuva 6 – Taustajärjestelmään tehtyjen online-validointikutsujen kestojen jakauma

myös laajalti ympäri koko linja-autoliikenteen aluetta, eikä yhteyshäiriöitä voida aina laittaa tietyn sijainnin tai katvealueen syyksi. Online-validointia ei voida suorittaa yhteyshäiriön aikana, joten lyhyetkin häiriöt ovat sen osalta ongelmallisia.

Kolmannessa tavassa löydettiin kaikki online-validointikutsut, ja niissä tapahtuneet virheet ja kutsujen kestot otettiin ylös. Rajapintakutsun kesto mitattiin juuri ennen kutsua kirjoitetun rivin ja kutsun jälkeisen rivin aikaleimojen erotuksesta. Rajapintakutsujen kesto vaikuttaa leimaustapahtuman nopeuteen, ja sitä kautta käyttökokemukseen. Kutsuja löydettiin 230 kpl ja niistä vain 1 oli epäonnistunut rikkinäisen verkkoyhteyden vuoksi. Kutsuista onnistui siis 99,57 %. Ajoneuvolaitteiden suorittamien validointikutsujen kestojen jakauma on esitetty kuvassa 6. Kutsujen kestojen mediaani oli 646 ms. Otanta validointikutsujen osalta on kuitenkin melko pieni, eikä näitä tuloksia voi siksi pitää kovin luotettavina.

Validointikutsujen kestot havaittiin oletettua pidemmiksi, ja tämän vuoksi asiaa tarkasteltiin tarkemmin. Havaittiin, että laiteohjelma luo verkkokutsun suorittavan ohjelmakirjaston instanssin uudestaan joka kutsua varten. Tästä pääteltiin, että myös TLS-käyttely suoritetaan uudestaan joka kutsulla. TLS-käyttely on TLS-protokollalla suojatun tiedonsiirron alussa tapahtuva toiminto, jossa tiedonsiirtoon osallistuvat tahot muodostavat keskenään turvallisen yhteyden. Käyttelyn aikana tahot välittävät useita viestejä toisilleen, mikä hidastaa suoritettavaa rajapintakutsua. (Chris Kemmerer, 2015)

Ajoneuvolaitteelle kirjoitettiin uusi ohjelmaversio, jossa ohjelmakirjaston instanssi luodaan vain yhden kerran, ja samaa instanssia käytetään jokaisella rajapintakutsulla. Muutoksen vaikutusta selvitettiin testillä, jossa rajapintakutsuja suoritettiin kahdella ohjelmaversiolla, ja kutsujen kestoja ohjelmaversioiden välillä verrattiin. Testi toteutettiin suorittamalla 10 online-validointirajapintakutsua 10 sekunnin välein sekä vanhalla että uudella korjauksen sisältävällä ohjelmaversiolla. Molempien versioiden

kutsut suoritettiin täysin samassa ympäristössä, ohjelmaversiota lukuunottamatta. Versioiden välillä ei ole muuta tiedonsiirtoon vaikuttavaa muutosta kuin edellä mainittu ohjelmakirjaston instanssin luomisen eroavaisuus. Testissä kutsujen kohteena ollut taustajärjestelmää suoritettiin samassa lähiverkossa kuin kutsut suorittanut ajoneuvolaite. Testissä mitatut rajapintakustujen kestot on esitetty taulukossa 1. Esitetyistä kutsujen kestoista on vähennetty taustajärjestelmän käsittelyyn kulunut aika, joka luettiin erikseen jokaiselle kutsulle taustajärjestelmän lokeista.

Taulukko 1 – Testissä mitatut rajapintakustujen kestot

VANHA OHJELMAVERSIO	UUSI OHJELMAVERSIO
177 ms	109 ms
200 ms	13 ms
193 ms	18 ms
182 ms	13 ms
197 ms	19 ms
177 ms	14 ms
186 ms	14 ms
173 ms	17 ms
179 ms	24 ms
202 ms	13 ms

Testin tuloksista huomataan, että ohjelmaversiossa tehdyllä muutoksella on todella suuri vaikutus kutsujen keston. Testissä rajapintakutsut suorittanut laite oli kiinteällä Ethernet-yhdeydellä kiinni samassa lähiverkossa kuin kutsujen kohteena ollut palvelin, ja vaikutus saattaa olla huomattavasti suurempi ajossa olevien ajoneuvolaitteiden yhteyksissä, joissa viiveet ovat suurempia. Tämä tarkoittaa, että edellä esitetyn lokeista tehdyn analyysin tulokset eivät validointikutsujen osalta välttämättä pidä paikkaansa uudella ohjelmaversiolla, vaan kutsut voivat olla tuloksissa esitettyjä kestoja huomattavasti nopeampia. Uuden ohjelmaversioon vaikutusta kanattaisi tutkia ajossa olevien laitteiden lokien perusteella. Työn aikana ohjelmaversioon muutosta ei kuitenkaan ehditä ottaa käyttöön, joten uusi lokeista tehtävä analyysi jätetään työn ulkopuolelle. Uuden ohjelmaversioon tuloksista huomataan myös, että ensimmäinen rajapintakutsu kesti huomattavasti muita kutsuja pidempään. Tähän saattaa olla syynä ensimmäisen kutsun yhteydessä tehtävästä TLS-käittelystä syntyvä viive.

Online-validoinnin onnistumismäärää tutkittiin myös tunnistepohjaisen taustajärjestelmän lokeista aikaväliltä 23.4.2021 klo 3:00 – 9.5.2021 klo 3:00. Lokien perusteella taustajärjestelmään suoritettiin 37 220 online-validointikutsua. Taustajärjestelmään välitettiin myös 201 leimaustapahtumaa, joille online-

validointi oli epäonnistunut, joten validointi onnistui 99,46 % yritetyistä validoinneista. Näistä 201 tapahtumasta 46 tapahtui ilman kelpavaa matkustusosoikeutta. Todennäköisesti kaikki rajapintaan tulleista validointikutsuista on peräisin ajossa olevilta ajoneuvolaitteilta, mutta joukossa saattaa olla myös muista lähteistä tulevia validointikutsuja. Näiden osuus katsotaan kuitenkin tulosten kannalta merkityksettömäksi, ja tulos kuvastaa hyvin ajoneuvolaitteiden tiedonsiirtoyhteyksien tilannetta.

5.1.3 Taustajärjestelmän suoritusympäristö

Pusatec käyttää Googlen pilvialustapalveluita (Google Cloud Platform, GCP) tietojärjestelmien suoritukseen ja operointiin. Tunnistepohjaista taustajärjestelmää suoritetaan Kubernetes-alustalla, joka on tarkoitettu palvelinohjelmien operoinnin automatisoituun hallintaan. Kubernetesen avulla voidaan hallita laajan ohjelmistokokonaisuuden suoritusta, jossa järjestelmän eri osia on jaettu itsenäisiin kontteihin (engl. container). Kuberneteselle kerrotaan deklaratiivisilla konfiguraatioilla, mitä palveluita halutaan ajaa, ja Kubernetes orkestroi automatisoidusti niitä ajavien konttien luomisen ja suorituksen, sekä järjestelmäkokonaisuuden muun hallinnan.

Pusatecin tunnistepohjainen lippujärjestelmä koostuu monesta pienemmästä palvelusta, jotka hoitavat tiettyä rajattua toiminnallisuutta. Kyseessä on siis eräänlainen mikropalvelu-arkkitehtuuri. Palvelut ovat itsenäisiä prosesseja, joita ajetaan konteissa, ja samasta palvelusta voi olla useampi kontti samanaikaisesti käynnissä. Kontteja suoritetaan virtuaalikoneiden päällä, joita kutsutaan Kubernetes-noodeiksi. Rajapintojen kautta taustajärjestelmään tulevat kutsut jaetaan Googlen Cloud Load Balancing -tekniikan avulla eri konttien välillä kuormien tasoittamiseksi. Tämänkaltainen jaettu arkkitehtuuri tekee taustajärjestelmästä paremmin skaalautuvan ja viansietoisemmän, sillä yhden kontin kaatumisen aiheuttavan vian ei pitäisi suoraan vaikuttaa muihin kontteihin. Kontit käyttävät osittain kuitenkin jaettuja resursseja, kuten prosessointikykyä ja muistia, joten resurssien ylikäyttö saattaa vaikuttaa muiden palveluiden suoritukseen. Jaettu järjestelmä tuo myös haasteita toteutukseen esimerkiksi tiedon yhteneväisyyden (engl. consistency) osalta. Myös esimerkiksi tunnisteiden luomisessa ei voida hyödyntää kontin muistissa olevaa nousevaa arvoa, vaan uniikit tunnisteet täytyy luoda yhteisessä palvelussa tai käyttää itsenäisesti luotavia tunnisteita, joiden päällekkäisyys on kuitenkin riittävän epätodennäköistä, kuten globaalisti uniikiksi tarkoitettuja GUID-tunnisteita.

Taustajärjestelmä käyttää palveluiden väliseen viestintään ja yhteneväisyyden varmistukseen Googlen Pub/Sub-viestintäkanavaa, jossa eri palvelut voivat välittää asynkronisesti viestejä toisilleen. Pub/Subin avulla palvelut voivat julkaista viestejä tietylle aiheelle, ja viestit välitetään asynkronisesti kaikille aiheen tilanneille palveluille. Pub/Sub huolehtii myös siitä, että aiheen tilaavat palvelut varmasti käsittelevät viestin, välittämällä tilaajalle saman viestin uudestaan, mikäli viestiä ei kuitattu käsitellyksi. Kun operaatio on hyväksytty järjestelmään ja siitä on luotu Pub/Sub-viesti, voidaan olla varmoja, että kaikki hajautetun

järjestelmän palvelut käsittelevät viestin lopulta, vaikka käsittely ei tapahtuisikaan välittömästi. Tämän avulla huolehditaan siitä, että järjestelmään hyväksytyt operaatio välittyy lopulta kaikille sitä tarvitseville palveluille, eivätkä palvelut jää epäyhtenäiseen tilaan esimerkiksi palveluiden kaatuessa. (Google, 2021b)

5.2 Toteutusmallin valinta

5.2.1 Offline-validoinnin tarpeellisuus

Tässä alaluvussa tarkastellaan offline-validoinnin tarpeellisuutta ja vastataan tutkimuskysymykseen TK1. Alaluvun 5.1.2 tiedonsiirtoyhteyksien analyysin perusteella voidaan todeta, että online-validointi onnistuu kohtalaisen suurella luotettavuudella (> 99%). Tämän puolesta offline-validointi ei ole välttämätön, mikäli pieni lipputuloriski hyväksytään. Leimaukset käsitellään joka tapauksessa taustajärjestelmässä, ja lipputuloriski koskee vain matkustusoikeudetta matkustavia käyttäjiä, joille online-validointi epäonnistuu. Oikeudettomat nousut voidaan myös huomioida taustajärjestelmässä esimerkiksi niin, että ne merkitään käyttäjälle velaksi, ja ne tulee korvata esimerkiksi seuraavan oston yhteydessä. Systemaattisen väärinkäytön estämiseksi myös luvussa 3.2.5 esitetty estolista voi tällöin olla hyvä ratkaisu; oikeudetta matkustaneet tunnisteet lisätään estolistalle ja poistettaisiin sieltä vasta matkojen korvauksen jälkeen. Velan korvausmenettelyt eivät ole kuitenkaan mahdollisia anonyymeille kertakäyttöisille lipuille, esimerkiksi paperilipuille. Offline-validoinnilla voidaan siis ehkäistä offline-tilanteista syntyvää lipputuloriskiä, ja lisätä myös järjestelmän viansietokykyä esimerkiksi tilanteessa, jossa online-validoinnista vastaava taustajärjestelmän osa on kokonaan alhaalla.

Online-validointia käyttävien leimausten kestot saattavat heikentää leimauksen käyttökokemusta ja matkustajien sujuvaa liikkumista oleellisesti. Älykorttitekniologia yhdistys Smart Card Alliance mainitsee toimialan leimastransaktion hyväksytyt keston ylärajaksi 500 ms (2011, s. 13). Uusille lippumedioille on kuitenkin hyväksytyt pidempiäkin transaktioaikoja. Esimerkiksi Lontoon alueella operoiva Transport of London on ilmoittanut älykorttien nopeudeksi 250 ms, cEMV-pankkikorttien nopeudeksi 500 ms ja puhelimen EMV-maksun nopeudeksi 750 ms (Cheng, 2017, s. 74). Luvun 5.1.2 perusteella Pusatecin nykyisten online-validointien mediaanikesto oli 646 ms. Osa online-validoinneista kestää kuitenkin huomattavasti pidempää; vain 90 % kutsuista kestää alle 1 s. Rajapintakutsun lisäksi leimaukseen kuuluu lippumedian lukemiseen menevä aika sekä muuta leimauslaitteen prosessointia, joten leimauksen kokonaiskesto on tätä hieman suurempi. Pusatecin järjestelmässä uuden laiteohjelmaversioiden myötä validointikutsut saattavat kuitenkin olla huomattavasti nopeampia, kuten luvussa 5.1.2 mainittiin. Voidaan kuitenkin todeta, että offline-validoinnin toteutuksella voidaan parantaa leimauksen nopeutta, vaikkakin nopeuden vaikutuksesta leimaustilanteiden käyttökokemukseen ei voida työn menetelmien perusteella sanoa.

Järjestelmän käyttö halutaan mahdolliseksi myös muille toimijoille, joiden ympäristöjen rajoitteet eivät välttämättä vastaa analysoitua ympäristöä. Myös lokimateriaalin perusteella tehty analyysi käyttää suureksi osaksi vain Jyväskylän alueelta kerättyä materiaalia, eivätkä muiden alueiden tiedonsiirtoyhteydet välttämättä vastaa analyysin tuloksia. Offline-validointi päätetään toteuttaa, mikäli toteutus havaitaan mahdolliseksi ja kohtuullisen helpoksi. Toiminnallisuus otetaan mahdollisesti käyttöön, mikäli sen toiminta havaitaan riittävän luotettavaksi.

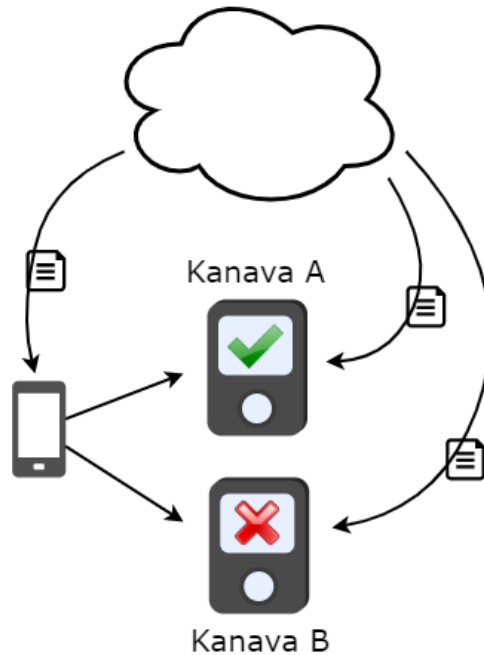
5.2.2 Offline-validointiin tarvittavat tiedot

Pusatecin järjestelmässä tietyn lipun validointi tapahtuu aina tietyn leimauskanavan avulla, kuten alaluvussa 4.1 mainittiin. Päätelaitteet käyttävät tiettyä kanavaa, ja matkustusoikeus validoidaan kanavan määrittelemien tuoteparametrien mukaisesti. Jos offline-tietoja kuljetetaan leimauslaitteille lippumedian mukana, kuten alalukujen 3.2.2 ja 3.2.3 mukaisilla malleilla, täytyy matkustusoikeus olla johdettavista tiedoista kaikille kanaville, joilla tunnistetta voidaan käyttää. Tämä vaatii todennäköisesti tuoteparametrien lataamista päätelaitteille. Jos kuitenkin tiedot jaetaan taustajärjestelmästä suoraan leimauslaitteille offline-listojen avulla, voidaan eri laitteille jakaa eri sisältöä, ja riittää, että tiedoista voidaan johtaa matkustusoikeus kyseisen laitteen kanavalle. Tällöin myös leimauslaitteilta vaadittava päättely on todennäköisesti kevyempää. Offline-tietojen kohdistaminen tietylle kanavalle vaikuttaa siis oleellisesti offline-tiedon sisältöön ja validointilaitteelta vaadittavaan päättelyyn.

Kuvassa 7 on esitetty tilanne, jossa matkustusoikeus kelpaa kanavalla A, mutta ei kanavalla B. Jos offline-validointiin tarvittavat tiedot välitetään päätelaitteille esimerkiksi mobiilisovelluksen kautta, täytyy validointi suorittaa samojen offline-tietojen perusteella kaikilla kanavilla. Jos taas offline-tiedot jaetaan suoraan päätelaitteille, voidaan eri kanavien laitteille jakaa eri sisältöä, eikä päätelaitteilla välttämättä tarvita tuoteparametreja.

Tietyissä tapauksissa matkustusoikeus riippuu vasta leimaushetkellä tiedettävistä parametreista, esimerkiksi silloin, kun matkan vyöhykkeet valitaan vasta matkan aloituksen yhteydessä. Tällöin edes kohdennetulle kanavalle ei voida muodostaa binääristä matkustusoikeuden arvoa taustajärjestelmässä, joten yksinkertaisin whitelist-tyyppinen offline-lista toteutus ei riitä. Leimauslaitteelle täytyy tällöin viedä laajempia koostetietoja, joilla validointi voidaan suorittaa täsmällisesti. Koska sisällön muodon vaatimukset riippuvat lippujärjestelmän säännöistä, eri offline-listoille saatetaan haluta lähettää erimuotoista sisältöä. Tällöin esimerkiksi järjestelmään liittyville muille toimijoille voidaan välittää mahdollisimman yksinkertaista sisältöä.

Yksi huomioitava asia offline-listojen sisällön valinnassa ovat lipputuotteet, joiden matkustusoikeus muuttuu aikaan perustuen. Esimerkiksi koululaisille suunnatun lipun, joka on voimassa vain arkipäivisin klo 7–17 välillä, matkustusoikeus muuttuu joka arkipäivä 2 kertaa. Jos offline-tietojen muoto



Kuva 7 - Offline-tietojen jakoväylät. Jos tiedot jaetaan suoraan leimauslaitteille, voidaan ne kohdentaa tietylle leimauskanavalle

pidetään yksinkertaisena, joudutaan taustajärjestelmässä seuraamaan tämänkaltaisia aikaan perustuvia matkustusosoikeuksien muutoksia, ja välittää muutokset offline-listoille aina matkustusosoikeuden muuttuessa. Tämä saattaa lisätä huomattavasti tapahtuvien muutosten määrää ja tietoliikennettä, ja suuri määrä muutoksia saattaa keskittyä samalle hetkelle, joten muutokset ovat hyvin purskeisia. Tämän vuoksi saattaa olla järkevämpää, että tämänkaltaisen validointilogiikka tapahtuu taustajärjestelmän sijaan vasta päätelaitteella, ja offline-tietoihin sisällytetään kelpaavuusaikojen huomiointiin tarvittavat tiedot.

Offline-tietoihin tarvittava sisältö riippuu siis lippujärjestelmän säännöistä ja toteutusmallista. Offline-tietojen koosta luotiin varovainen arvio, jonka perusteella offline-listojen toteutuksen käyttökelpoisuutta (engl. feasibility) voidaan tarkastella. Arvio muodostettiin luomalla joukko tyypillisiä tilien tiloja, kuten esimerkiksi yhden tai kaksi kausilippua omistavia tilejä. Näistä tileistä poimittiin offline-validointiin tarvittavat tiedot, jotka koodattiin binäärimuotoiseksi Protobuf-tekniikalla. Yhden rivin keskimääräinen koko oli suunnilleen 100 tavun ympärillä. Offline-tiedoista saattoi jäädä joitain kenttiä puuttumaan, niin käytetään yhden rivin koosta varovaisempaa 200 tavun arviota.

5.2.3 Mallin valinta

Vaatimuksessa I esitettiin tarve offline-validoinnin toimimisesta kaikilla tunnistemedioilla. Offline-tietojen kuljetus offline-listojen avulla on ainoa toteutusmalli, joka ei riipu tunnistemediasta, ja

mahdollistaa näin saman toteutuksen hyödyntämisen kaikille mediatyypeille. Lista mahdollistaa myös edellisen alaluvun mukaisesti mahdollisuuden kohdentaa offline-tiedot tietylle leimauskanavalle.

Tarkastellaan seuraavaksi, miten nykyiset tiedonsiirtoyhteydet ja päätelaitteiden resurssit riittävät offline-listojen jakelulle. Päätelaitteet ovat yhteydessä tunnistepohjaiseen taustajärjestelmään, joten listojen sisältöä jaettaisiin taustajärjestelmän jakelurajapinnasta. Listojen sisältöjen muutokset päivitetään jatkuvasti laitteiden paikallisiin listoihin, ja pitkät yhteyshäiriöt estävät muutosten hakemisen, sekä leimaus- ja myyntitapahtumien välittämisen taustajärjestelmään. Tällöin on riski, että leimaustulokset perustuvat vanhentuneeseen tilaan, ja niiden tulos on virheellinen. Näin voi tapahtua esimerkiksi silloin, jos matkustaja ostaa tunnistepohjaisen matkalipun verkkokaupasta ja pyrkii matkustamaan ennen kuin myynnistä aiheutunut offline-listan muutos on ehtinyt ajoneuvolaitteen offline-listalle. Vaatimuksessa V asetettiin uusien matkustusoikeuden päivittämisen tavoiteajaksi alle 2 minuuttia. Luvun 5.1.2 analyysin perusteella laitteilla tapahtuu kuitenkin jonkin verran yli 2 minuutin yhteyshäiriöitä, joten tähän tavoitteeseen ei kaikissa tilanteissa päästä. Yhteyshäiriöt olivat kuitenkin hyvin poikkeuksellisia, ja vaatimus salli myös pidemmät päivitysajat poikkeustilanteissa, joten voidaan todeta, että tavoitteeseen voidaan yhteyshäiriöiden osalta päästä.

Myös tiedonsiirtonopeuksien on riitettävä offline-listojen jatkuvaa jakamista varten. Koko Suomen joukkoliikenteessä tehdään alaluvussa 2.4.3 esitettyjen tietojen mukaan 1,57 miljoonaa matkaa päivässä, joka tarkoittaa noin 18 leimaustapahtumaa sekunnissa. Jos offline-listan riveistä käytetään aiemmin muodostettua 200 tavun kokoarviota, ja matkustusoikeus muuttuu jokaisella leimauksella, koko Suomen leimaustapahtumien aiheuttamien tilien muutosten välitykseen tarvitaan vain noin 29 kb/s keskimääräinen tiedonsiirtonopeus. Tämän todellista käyttöä rajusti yläarvioivan esimerkin perusteella voidaan todeta, että tiedonsiirtoyhteyksien nopeuksien ei pitäisi aiheuttaa ongelmia offline-listan jakeluun.

Listojen tulee myös mahtua päätelaitteiden muistiin. Tietueen 200 tavun kokoarviolla esimerkiksi 1 miljoonaa tiliä sisältävän offline-listan koko on 200 Mt. Modernit päätelaiteratkaisut kykenevät varmasti tämän suuruisen tietomäärän säilyttämiseen ainakin massamuistissa. Offline-listojen ylläpito tai käyttö ei myöskään vaadi erityisen suurta prosessointikykyä laitteilta. Modernien päätelaitteiden pitäisi siis pystyä listojen säilytykseen, ylläpitoon ja käyttöön vaadittuun suorituskykyyn.

Offline-listojen jakelu täyttää siis asetetut vaatimukset, ja offline-validointi päätettiin toteuttaa päätelaitteille ladattavien listojen avulla. Tämän työn aikana järjestelmästä toteutettiin offline-listojen jakelujärjestelmän prototyyppi, jonka perusteella saadaan lisää tietoa järjestelmän käytettävyydestä. Prototyyppitoteutukseen kuuluu offline-listojen muodostus ja niiden jakelurajapinta, josta muutoksia jaetaan niitä tarvitseville päätelaitteille ja mahdollisille muille tahoille.

5.3 Listojen jakelun suunnittelu

Tässä luvussa tarkastellaan offline-listojen muodostukseen ja jakeluun liittyviä haasteita ja suunnitellaan jakelujärjestelmän toteutusta. Suunnittelussa keskitytään taustajärjestelmässä tapahtuvaan listojen muodostukseen ja jakelurajapinnan tarjoamiseen listojen tilaajille. Tässä työssä ei käsitellä tarkasti päätelaitteiden offline-validoinnin teknistä toteutusta, mutta siitä annetaan korkeamman tason kuvaus, jonka perusteella toteutus voidaan tehdä.

5.3.1 Listojen muoto ja muodostus

Offline-listaa voidaan pitää kuvauksena tilien tunnisteista tileihin liittyvään offline-tietoihin. Kutsutaan yhtä listan riviä, joka sisältää siis tunnisteiden sekä offline-tiedot, listan *tietueeksi*. Taustajärjestelmässä tapahtuvat myynti- ja leimaustapahtumat muuttavat tilien tiloja ja muuttuvat offline-tiedot välitetään päätelaitteiden offline-listoille. Tietorakenteellisesti offline-lista kannattaa säilyttää hakurakenteessa, josta tietueita voidaan lukea ja kirjoittaa tilien tunnisteiden perusteella. Tunnisteet kannattaa muodostaa indeksiksi, joka voi olla tyypillinen tehokas hakurakenne, kuten hakupuu, jotta operaatiot ovat nopeita. Ensimmäisen käynnistyksen yhteydessä päätelaitteet lataavat taustajärjestelmästä koko offline-listan, ja tämän jälkeen ne lataavat pelkästään päivän mittaan tapahtuneita muutoksia, ja päivittävät omat listansa muutosten mukaisesti.

Offline-listan sisältö johdetaan matkustusoikeuteen vaikuttavista tilin tiedoista, joita ovat matkatilin liput, maksukyky, ja menneet leimaukset. Menneet leimaukset vaikuttavat matkustusoikeuteen esimerkiksi niihin liittyvän vaihto-oikeuden takia. Listojen luominen täysin tyhjästä on raskas prosessi, sillä tämä vaatii kaikkien tilien tilojen läpikäymisen ja muuttamisen offline-lista muotoon. Tämän vuoksi on järkevämpää muodostaa listat tyhjästä yhden kerran, ja tämän jälkeen ainoastaan päivittää listoja rivikohtaisilla muutoksilla.

Matkatilien tiloja hallintaan omassa kontissa ajettavassa palvelussa, joka välittää matkatilin muuttuessa Pub/Sub-viestintäkanavaa pitkin muutoksen jälkeisen tilin sisällön. Offline-listoja voidaan päivittää kuuntelemalla näitä muutosviestejä ja johtamalla niistä listoille uusia tietueita. Tietueita verrataan ensisijaisen listan senhetkiseen sisältöön, ja sisällön muuttuessa vanha tietue korvataan uudella. Taustajärjestelmään muodostettava ensisijainen offline-lista kannattaa tallentaa pysyvään säilöön, jotta listoja ei tarvitse muodostaa tyhjästä uudestaan useampaan kertaan palveluiden uudelleenkäynnistyksen yhteydessä.

Listojen ylläpitoa varten kannattaa pystyttää omassa kontissa ajettava palvelu, joka kuuntelee Pub/Subin viestejä ja kirjoittaa näiden pohjalta muutoksia ensisijaisiin listoihin. Palvelu toimii itsenäisesti omassa kontissa, eikä sen toiminta ja mahdolliset vikatilanteet vaikuta suoraan muihin taustajärjestelmän palveluihin. Yhdellä Pub/Sub-tilauksella voi olla useita käsittelijöitä, joten ylläpitopalvelusta voidaan

samanaikaisesti pitää päällä useampi itsenäinen kontti, jotka kuuntelevat ja käsittelevät saman Pub/Sub-tilauksen viestejä. Saapuvat viestit jakaantuvat konttien välillä. Pub/Sub-viestit välitetään tilaukselle ainakin yhden kerran, mutta poikkeustilanteissa sama viesti saattaa saapua samalle tilaukselle useamman kerran. Tästä syystä viestien käsittelyn tulee olla idempotenttinen operaatio, eli operaatio, jonka suorituksen toistaminen useampaan kertaan ei muuta suorituksen lopputulosta. (Google, 2021c) Offline-listan rivimuutoksen johtaminen ja tallentaminen ensisijaiselle listalle on idempotenttinen operaatio, jos esimerkiksi vanhan ja uuden tietueen sisältöä verrataan, ja muutos kirjoitetaan listalle vain tietueen muuttuessa.

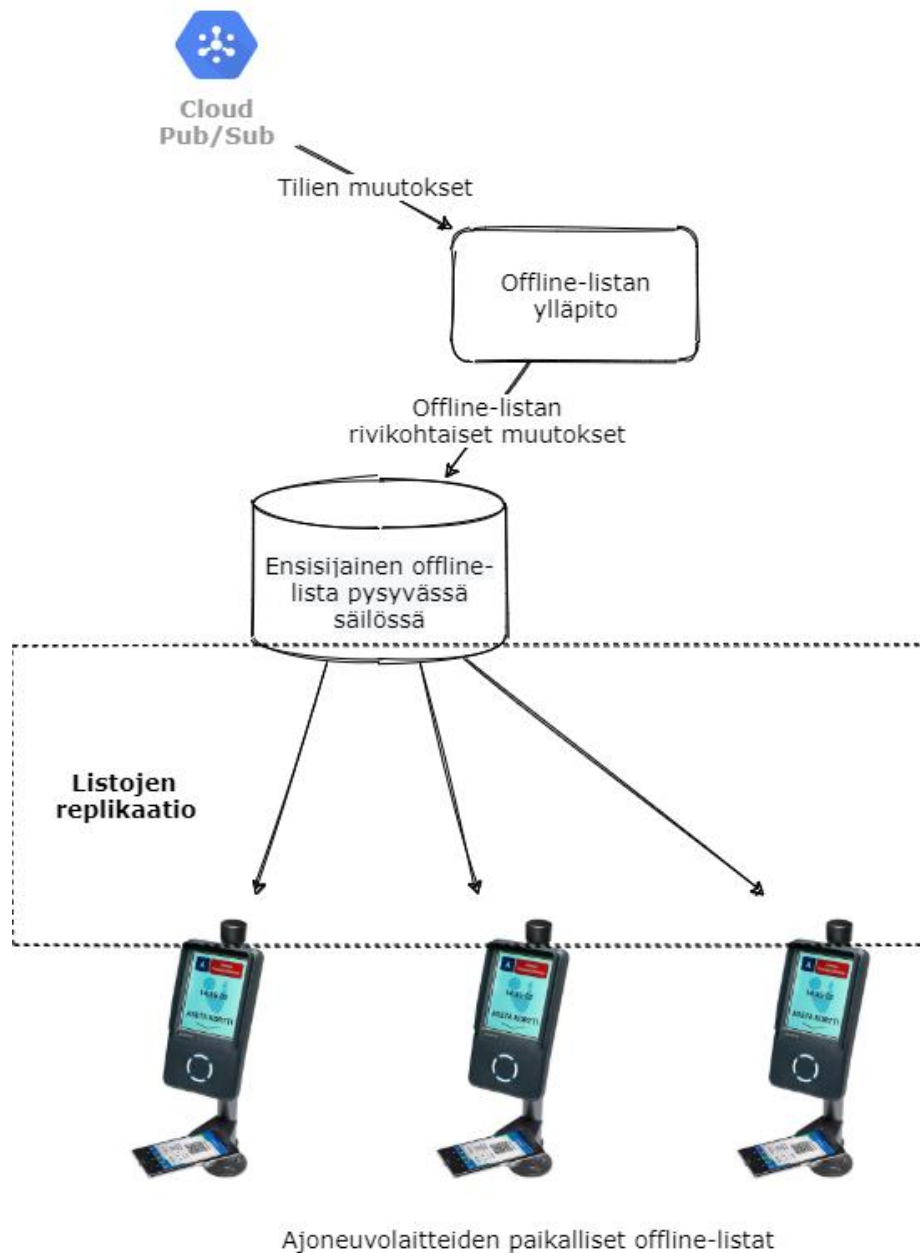
Koska offline-listalle tarvittava sisältö riippuu kyseisen lippujärjestelmän säännöistä, saatetaan eri offline-listoille tulevaisuudessa haluta eri muotoista sisältöä. Tämä huomioidaan listojen jakelujärjestelmässä, missä ei oteta kantaa listan tietueiden offline-tietojen muotoon. Prototyyppitoteutuksessa listoille voidaan viedä kaikki kyseiselle kanavalle kuuluvat tilien matkustusosoikeuteen liittyvät tiedot, kuten tilin aktiiviset liput, maksukyky ja mahdolliset vaihto-oikeudet, joista matkustusosoikeus voidaan päätellä täsmällisesti päätelaitteilla.

5.3.2 Listojen replikaatio

Offline-listojen jakelujärjestelmä on hajautettu järjestelmä, joka koostuu Pusatecin taustajärjestelmästä, useista päätelaitteista, ja mahdollisista muista listojen sisältöä tarvitsevien toimijoiden tietojärjestelmistä. Kuvassa 7 on esitetty listan muodostukseen liittyvä arkkitehtuuri. Kuvasta näkyy offline-listaa ylläpitävä palvelu, joka kuuntelee Pub/Sub-viestintäkanavalta tilien tilojen muutoksia koskevia viestejä, ja johtaa näistä ensisijaisille listoille uusia rivejä. Ensisijaiset listat replikoidaan jakelurajapinnan kautta listan tilanneille päätelaitteille.

Offline-listoja voidaan ajatella tietokantoina, joissa on yksi taulu, jonka pääavaimena toimii tilin tunniste. Listojen replikointi ensisijaisilta listoilta niiden tilaajille voi tapahtua master-slave tyylisesti. Master-slave tyyppisessä replikaatiossa yksi tietokanta toimii ensisijaisena kantana, ja kaikki tapahtuvat kirjoitusoperaatiot suoritetaan vain tähän kantaan. Offline-listojen jakelussa vain taustajärjestelmä tekee kirjoitusoperaatioita ensisijaisten listojen sisältöön; tilaajien listoihin suoritetaan vain lukuoperaatioita. Tällöin hajautettujen tietokantojen välillä ei synny kirjoituskonflikteja, vaikka replikoitujen kantojen tilat eivät säilyttäisi vahvaa yhteneväisyyttä (engl. strong consistency). (Eskelinen, 2019, s. 14)

Listojen tilaajat saattavat olla hyvinkin pitkiä aikoja poissa tiedonsiirtoyhteyksien ulottuvilta, esimerkiksi laitteiden ollessa virrattomina, jolloin ne jäävät paljon jälkeen ensisijaisesta listasta, ja haluavat käynnistyksen jälkeen päivittää itsensä ajan tasalle. Replikointi täytyy siis toteuttaa passiivisesti, eli asynkronisesti (engl. passive replication, lazy replication). Passiivisessa replikoinnissa ensisijaiseen listaan tehtyjä muutoksia ei välitetä replikoiduille tahoille välittömästi transaktion aikana, vaan vasta jälkikäteen



Kuva 7 – Ensisijaisen offline-listan muodostus pysyvään säilöön ja replikaatio ajoneuvolaitteille

(Eskelinen, 2019, s. 17). Tilaajien listojen ei tarvitse päivittyä välittömästi muutosten tapahtuessa, vaan riittää, että replikoitujen tietojen yhteneväisyys saavutetaan lopulta (engl. eventual consistency). Replikoidut listat eivät myöskään välitä toistensa tiloista, vaan ovat riippuvaisia ainoastaan ensisijaisen listan tilasta.

Kuten edellä luvussa 5.3.1 mainittiin, siirrettävien tietomäärien minimoimiseksi replikaatio kannattaa toteuttaa välittämällä päätelaitteille vain ajan mittaan tapahtuvia muutoksia. Tarvitaan siis keino muuttuneen tiedon tunnistamiseen (engl. change data capture, CDC). Menetelmiä tähän ovat

tietokantataulujen sisältöjen vertailu, rivien aikaleimat tai versionumerot, tietokannan liipaisimet (engl. database triggers), ja tietokannan lokin lukeminen (Barkaway, 2009, s. 5–6).

Mainituista menetelmistä yksinkertaisin ja sen myötä parhaiten offline-listojen replikaatioon soveltuva muutosten tunnistamismenetelmä on rivien versionumeroihin perustuva menetelmä. Tätä menetelmää varten tauluihin lisätään versionumeron sisältävä sarake, joka päivitetään aina kirjoitusoperaation yhteydessä kaikkia muita versionumeroita suurempaan arvoon. Taulusta voidaan lukea tietyn version jälkeen tapahtuneet muutokset yksinkertaisella kyselyoperaatiolla. Muutoksia voidaan hakea esimerkiksi tietynkokoisissa erissä, jotta yhden kutsun tietomäärä ei nouse kohtuuttoman suureksi. (Barkaway, 2009, s. 5–6)

Kun ensisijaiseen listaan tehdään muutoksia, täytyy rivien versionumeroa kasvattaa muiden rivien versionumeroita suuremmaksi, jotta muutosten replikointi onnistuu. Versionumeroilta voidaan vaatia ainutlaatuisuutta kaikkien listan rivien kesken, mutta ainutlaatuisen nousevan versionumeron hallittu luominen voi kuitenkin olla haastavaa hajautetussa arkkitehtuurissa, jossa samasta palvelusta voi olla useampi kontti samanaikaisesti suorituksessa. Myöskään tietokantakerros ei välttämättä tue nousevan kokonaislukuarvon automaattista luomista, mikäli kyse on esimerkiksi hajautetusta tietokantajärjestelmästä. Tällöin ainutlaatuisen nousevan versionumeron generointi tulisi tapahtua keskitetysti yhdessä prosessissa tai tietokantojen transaktioiden avulla. Versionumeroiden ainutlaatuisuudesta voidaan kuitenkin joustaa, mikäli mahdolliset päällekkäiset versionumerot huomioidaan replikoinnissa niin, ettei rivejä jää välistä. Tämä voidaan tehdä esimerkiksi sisällyttämällä seuraavaan muutoseriään uudestaan ne rivit, joiden versionumero on sama kuin muutoseriä kysyvän replikantin versionumero. Tämä johtaa kuitenkin siihen, että muutoserien välillä on jonkin verran päällekkäisiä rivejä. Muutoserän koosta ja erien hakutiheydestä riippuen tällä voi kuitenkin olla häviävän pieni vaikutus välitettäviin tietomääriin. Jos versionumeroiden päällekkäisyys sallitaan, versiona voi toimia esimerkiksi kirjoitusoperaation ajankohdan unix-aikaleima millisekunnin tarkkuudella.

Rivien versiolla toteutettua replikaatiota kutsutaan myös avainpohjaiseksi inkrementaaliseksi replikoinniksi (engl. key-based incremental replication). Yksi haaste tämänkaltaisessa toteutuksessa on, että muutosten tunnistamismenetelmä ei havaitse poistettuja rivejä. (Faraz, 2020) Tämä ongelma voidaan kuitenkin ratkaista niin, että koko rivin poistamisen sijaan riville kirjoitetaan tyhjä sisältö, joka välittyy avainpohjaisen replikointimenetelmän avulla eteenpäin. Päätelaitteet tunnistavat tyhjän sisällön merkitsevän rivin poistamista.

Avaimen perustuvassa replikaatiossa vain rivin uusin tila huomioidaan ja välitetään eteenpäin. Jos riville tapahtuu useita muutoksia muutoskyselyiden välissä, vain rivin uusin tila välitetään eteenpäin. Tämä soveltuu hyvin offline-listan replikointiin, ja saattaa vähentää välitettävää tietomäärää. Päätelaitteet ovat kiinnostuneita vain listojen uusimmasta tilasta, eikä niiden tarvitse huomioida jokaista

listoihin tehtyä operaatiota. Yksinkertaisuuden vuoksi rivien muuttuessa päätelaitteille välitetään rivin uusi sisältö kokonaisuudessaan, jotta listojen päivitysoperaatiot ovat yksinkertaisia korvausoperaatioita.

Avainpohjainen replikaatio on kohtalaisen helppo toteuttaa alustasta riippumatta sen yksinkertaisuuden vuoksi. Tiedon tallentamiseen voidaan käyttää hyvin yksinkertaisia tietorakenteita, joiden ylläpitämiseen riittää pelkästään rivien korvausoperaatiot. Jokainen offline-listan kopio tietää oman versionumeronsa, ja pystyy sen avulla hakemaan version jälkeen tapahtuneet muutokset ensisijaiselta listalta.

5.3.3 Ensisijaisten listojen säilytys

Ensisijainen lista voi olla samanlainen kuvaus tilien tunnisteista offline-listan sisällöksi kuin leimauslaitteiden listat, mutta ensisijainen lista tarvitsee tämän lisäksi edellisessä luvussa kuvattuun replikointiin tarvittavan versionumeron jokaiselle riville. Tämän rivikohtaisen versiotiedon avulla ensisijaiselta listalta voidaan hakea kaikki tietyn version jälkeen muuttuneet rivit. Ensisijaiseen listaan tehtyjen muutosten tulee aina nostaa muuttuneiden rivien versionumeroa. Ensisijaiset offline-listat voidaan säilyttää pysyvään säilöön, kuten tietokantaan, jotta niitä ei tarvitse muodostaa uudestaan tyhjästä palveluiden uudelleenkäynnistymisen yhteydessä. Tietokannan avulla myös listojen jakelupalvelut voidaan suorittaa samaa tietokantaa käyttävissä, mutta erillisissä, konteissa. Tietokannan valinnan osalta priorisoidaan hyvin skaalautuvia ja tehokkaita ratkaisuja. Koska listojen tietueiden välillä ei ole relaatioita, dokumenttietokannat soveltuvat hyvin listojen säilytykseen.

Listojen säilytyksessä ei välttämättä tarvitse ottaa kantaa rivikohtaisten tietueiden sisällön rakenteeseen tai koodaukseen, jolloin offline-tietojen muoto ja koodaus voi tarvittaessa vaihdella eri offline-listojen välillä. Listan sisältöä voidaan käsitellä ensisijaisten listojen säilytyksen aikana myös binäärimuotoisena, joten se voi sisältää mitä vain koodattuja offline-tietoja. Jokaista offline-listoille lähetettävää tiliä kohden tietokantaan tallennetaan yksi rivi, johon kuuluu offline-listan tilikohtainen sisältö binäärikoodattuna, sekä metadataa tilin tunnistusta ja listan ylläpitoa varten. Tietokantaan tehdään samanlaisia rivikohtaisia päivityksiä kuin laitteiden listoille, eli muutoksessa rivin binäärisisältö korvataan kokonaan uudella sisällöllä.

Ensisijaisista listoista luetaan tietyn version jälkeen tapahtuneita muutoksia, joten rivien versiotiedot kannattaa muodostaa indeksiksi, jotta lukuoperaatiot ovat nopeita. Tietokantaan tehtyjen muutosten yhteneväisyys (engl. consistency) täytyy huomioida niin, ettei muutosrivejä varmasti jää välistä. Mikäli esimerkiksi tietokantaan kirjoitetut rivit voivat tulla lukukyselyiden saataville eri järjestyksessä kuin mikä oli niiden kirjoitusjärjestys, on riski, että versionumeroa käyttävässä replikaatiossa rivejä jää välistä. Tämä täytyy huomioida listojen jakelurajapinnassa. Taulukko Taulukko 22 listaa tietokannan rivitietueiden kenttien tietotyypit.

Vaatimuksessa VI määritetään, että toteutus halutaan joustavasti osaksi nykyistä taustajärjestelmää, joka on rakennettu GCP:n palveluiden päälle. GCP yksi suosituimmista tietokantaratkaisuista on Cloud Firestore. Se on hyvin skaalautuva NoSQL-tietokanta, johon tieto tallennetaan itsenäisinä skeemattomina dokumentteja. Firestore soveltuu offline-listan säilyttämiseen sen skaalautuvuuden ja yksinkertaisen dokumentteihin pohjautuvien tiedon kirjoitus ja kyselyoperaatioiden takia.

Taulukko 2 - Tietokannan tietueiden kentät

TIETOKENTÄN NIMI	TIETOTYYPPI	INDEKSISSÄ
Listan tunniste	String	Kyllä
Tilin tunniste	String	Kyllä
Versionumero	64bit Integer	Kyllä
Sisältö	Binääri	Ei

5.3.4 Tiedon koodaus

Toteutuksessa halutaan vaatimuksen IV mukaisesti minimoida tietomäärät. Siirrettävän datan ei tarvitse olla esimerkiksi ihmiselle helposti luettavissa muoto, vaan priorisoidaan hyvin pakattuja ja helposti eri ohjelmointikielien purettavissa olevia formaatteja. Tämän takia tietoa voidaan jakaa binäärimuotoisena, tekstimuotoisen sisällön sijaan. Koodausta ja sen purkamista varten tiedon tulee olla jäsenneily tietyn määrityksen, eli skeeman (engl. schema), mukaisesti.

Google Protocol Buffers on ohjelmointikieli- ja alustariippumaton tiedonsiirtomuoto jäsenneilyn datan siirtämiseen. Tekniikka koodaa tiedon tehokkaasti binäärimuotoon. Siirrettävän tiedon rakenne täytyy määrittellä etukäteen skeemaan, joka tehdään tyypillisesti .proto-tiedostoissa, joiden avulla Googlen tai kolmansien osapuolten tarjoamat kirjastot osaavat koodata ja purkaa dataa. Tiedon rakenne täytyy olla sekä koodaus- että purkupuolella tiedossa. Skeemaa voidaan kuitenkin jälkikäteen muuttaa niin, että vanhaa skeemaa käyttävät toteutukset säilyvät ehjänä. Skeema tietorakenteiden lisäksi .proto-tiedostoissa voidaan määrittää rajapintakutsujen muoto, jonka kirjastot tulkita eri ohjelmointikielille sopiviksi rajapinnoiksi. Kirjastot tukevat sekä yksittäisiä kutsuja, että myös jatkuvan datavirran muodostusta, jonka sisällä voidaan välittää useita viestejä. (Google, 2020) Koska Pusatecin taustajärjestelmässä käytetään tällä hetkellä Protocol Buffers -tekniikkaa, eikä nähdä syytä tästä tekniikasta poikkeamiseen, päätetään offline-listan sisältö myös koodata Protocol Buffers -tekniikalla.

5.3.5 Jakelurajapinta

Ajoneuvolaitteita ja muita toimijoita varten taustajärjestelmä tarjoaa jakelurajapinnan, jonka avulla offline-listojen sisältöä voidaan hakea. Jakelurajapintaa varten kannattaa perustaa oma palvelu, jota ajetaan omassa kontissa. Näin rajapinta pysyy itsenäisenä muusta järjestelmästä, eikä sen toiminta ja mahdolliset ongelmat vaikuta suoraan muihin järjestelmän palveluihin.

Vaatimuksessa II mukaisesti tilin matkustusosoikeuksien muuttumisen tulee päivittyä listoille 2 minuutin kuluessa. Ei siis välttämättä ole tarvetta täysin reaaliaikaiselle yhteydelle, jossa päivitykset lähetettäisiin laitteille heti niiden saapuessa. Esimerkiksi minuutin välein välitettävät muutoserät on riittävä tahti. Yksinkertaisuuden vuoksi muutosten haku voidaan toteuttaa prototyypin toteutukseen päätelaitteiden aloittamien toisistaan irrallisten rajapintakutsujen avulla, jossa laitteille palautetaan aina erä edellisen version jälkeen tapahtuneita muutoksia. Näin jakelurajapinnan ei tarvitse ylläpitää jatkuvaa yhteyttä jokaisen listan tilanteen päätelaitteen kanssa.

Päätelaitteet voivat siis kysellä listojen sisältöä rajapinnasta. Kyselyissä laitteet kertovat rajapinnalle niiden paikallisen listakopion nykyisen version, ja rajapinta vastaa erällä kyseisen version jälkeen tapahtuneita muutoksia, sekä uudella versionnumerolla. Jos listan tilaajilla ei ole vielä mitään offline-listan sisältöä, voivat ne kysyä koko listan sisällön asettamalla kyselyn versionumeroksi 0. Rajapinnan vastauksena lähettämän muutoserän koko rajoitetaan niin, ettei se nouse kohtuuttoman suureksi. Vastauksessa rajapinta ilmoittaa myös ensisijaisen listan uusimman version, jonka avulla tilaajat tietävät, sisältääkö vastaanotettu erä uusimmat muutokset, vai kannattaako vielä hakea uusi erä muutoksia.

Offline-listan tietueiden sisältö voi siirtyä binäärimuotoisena päätelaitteille asti, jolloin tietomäärät pysyvät pieninä. Päätelaitteet voivat myös säilyttää listat binäärimuotoisena, ja purkaa koodauksen vasta leimaushetkellä, kun offline-tietoja tarvitaan. Binäärimuotoinen sisältö tarkoittaa kuitenkin, että sisällön lukeminen on replikaation aikana hankalaa, mikä saattaa vaikeuttaa vikatilanteiden selvittämistä. Mikäli muille toimijoille halutaan tarjota binäärikoodaamaton lista, voidaan binäärikoodaus purkaa myös jakelurajapinnassa ennen kutsuihin vastaamista. Tietoa voidaan näin jakaa myös muilla tavalla koodattuna, esimerkiksi JSON-muotoisena.

Tiedonsiirron yksi yleisimmistä standardeista on TCP/IP-kerros, joka vastaa tiedon oikeellisesta kuljetuksesta oikeaan kohteeseen. Offline-listojen jakelu kannattaa toteuttaa tämän päälle, jotta rivien muutoserät välittyvät varmasti oikein. TCP-kerros huolehtii muutoserien täydellisestä välityksestä, joten sen puolesta ei ole riskiä, että muutosrivejä putoaa pois tiedonsiirron aikana. Rajapintakutsut voidaan prototyypin toteutuksessa toteuttaa esimerkiksi HTTP-standardin päälle, mutta tulevaisuudessa muitakin tiedonsiirtoprotokollia voidaan tukea tarvittaessa. Liikenne kannattaa myös salata yleisellä TLS-standardilla (Transport Layer Security). Liikenteen salauksen lisäksi TLS-standardi varmistaa, että laitteet hakevat listoja oikealta taholta.

Tieto voidaan myös pakata pienempään kokoon häviöttömien pakkausalgoritmien avulla (engl. loss-less compression algorithms). HTTP-standardin viestejä on mahdollista pakata niin, että pyynnön yhteydessä asiakasohjelma ilmoittaa pyynnön otsikkotiedoissa tukemansa pakkaustekniikat, joista palvelin valitsee yhden ja ilmoittaa käytetyn algoritmin vastauksen otsikkotiedoissa. Yleisimmin käytetty pakkausalgoritmi on gzip. (Mozilla, 2021)

5.3.6 Välimuistissa olevat listat

Järjestelmän toteutuksesta halutaan vaatimuksen V mukaisesti mahdollisimman tehokas ja skaalautuva. Koska saman offline-listan haluavia leimauslaitteita voi olla useita satoja, voi olla järkevää, että jakelurajapinnan lähellä pidetään välimuistissa versiot listoissa, jotka voidaan tehokkaammin jakaa niitä tarvitseville laitteille. Välimuistilistojen avulla jokaisen rajapintaan saapuvan kyselyn ei tarvitse käydä tietokannassa asti kysymässä tarvitsemaansa sisältöä, jolloin myös tietokannan käytöstä syntyviä kustannuksia rajoitetaan. Välimuistilistat lisäävät kuitenkin jakelupalvelun muistin käyttöä. Luvussa 5.2.3 esitetystä esimerkissä offline-listan kooksi saatiin 200 Mt, mutta jakelujärjestelmä saattaa jakaa samanaikaisesti useita eri offline-listoja, ja jakelupalvelusta suoritetaan useaa konttia, joten muistinkäyttö täytyy huomioida suoritusympäristön resursseissa. Välimuistilistojen avulla kuormitusta voidaan siis siirtää tietokannasta Kubernetes-noodien resursseihin.

Välimuistilistat ovat replikaatio tietokannan ensisijaisista listoista, ja niitä käytetään puolestaan päätelaitteiden listojen replikaatiossa tiedon lähteenä. Tietueiden versiotiedon avulla toteutetun replikaation takia tämänkaltaisen replikaation välimuistin toteutus on helppoa. Muutoksia kyselevien päätelaitteiden rajapintakutsuihin vastataan välimuistilistan sisällöllä. Välimuistilistojen päivityksen tiheydestä riippuen rajapintakutsujen vastauksissa ei välttämättä ole kaikkea ensisijaisten listojen uusimpia muutoksia. Jos välimuistilistoja päivitetään kuitenkin esimerkiksi 10 sekunnin välien, ei tällä ole merkitystä. Välimuistilistojen tallennuksessa käytetty tietorakenne kannattaa valita niin, että varsinkin lukuoperaatiot ovat nopeita.

Alaluvussa 5.3.3 kuvattuja tietokannan mahdollisesta epäyhteneväisyydestä ja rivien päällekkäisistä versionumeroista johtuvat riskejä voidaan ehkäistä välimuistikerroksen avulla. Riski liittyy siis muuttuneiden rivien välistä jäämiseen niin, että kaikki kirjoitetut muutokset eivät vielä ehtineet edelliseen kyselyyn mukaan, ja seuraavassa kyselyssä käytetään jo uudempaa versionumeroa. Ongelma voidaan ehkäistä pitämällä välimuistilistojen replikaatioiden versioita esimerkiksi muutamia sekunteja listojen ylläpitopalvelua jäljessä, jolloin kannasta kysyttävät muutokset ovat varmasti saavuttaneet yhtenevän tilan. Tämä voidaan tehdä lisäämällä tietokantakyselyiden versionumeroon tietyn maksimiarvon.

5.4 Vaatimusten ratkaisut

Tässä alaluvussa kootaan yhteen, miten offline-lista järjestelmän toteutus saadaan vastaamaan luvussa 4 esitettyjä vaatimuksia. Esitetyt ratkaisut perustuvat aiemmin tässä luvussa tehtyyn analyysiin ja suunnitteluun.

I. Tunnistemediä riippumattomuus

Alaluvussa 3.2 esitettiin alan kirjallisuudesta etsittyjä offline-validoinnin toteutusmalleja. Päätelaitteille ladattavat offline-listat ovat ainoa löydetty malli, joka eri ole riippuvainen tunnistemediasta, ja tämän vuoksi offline-validointi päätettiin toteuttaa offline-listojen avulla.

II. Muuttuneiden matkustusosoikeuksien reaaliaikainen huomiointi

Luvussa 5.3.2 tehdyn tiedonsiirtoanalyysin perusteella voidaan todeta, että nykyiset tiedonsiirtoyhteydet mahdollistavat riittävän tiedonsiirtonopeuden ja kapasiteetin, jotta offline-listojen sisältöä voidaan reaaliaikaisesti välittää leimauslaitteille. Yhteyksien luotettavuus ei kuitenkaan takaa, että yli 2 minuutin yhteykskatkoja, ja niistä aiheutuneita virheellisiä leimaustuloksia, ei tapahdu. Vaatimuksessa hyväksyttiin tiedonsiirtoyhteyksien katkosta aiheutuvat ongelmat, joten asetettu vaatimus kyetään täyttämään.

III. Avoimuus muille toimijoille

Järjestelmän avoimuutta ja yhteistoiminnan helppoutta varten listojen jakelu haluttiin toteuttaa yleisillä tekniikoilla ja yksinkertaisella rajapinnalla, jonka avulla muut toimijat voivat kohtalaisen helposti toteuttaa offline-validoinnin omiin palveluihinsa ja laiterakaisuihinsa. Jakelurajapinnassa toteutukseen onkin tämän vuoksi valittu yleisiä tekniikoita alaluvun 5.3 kuvauksien mukaisesti. Myös replikaation toteutus ja rajapintakyselyiden muoto valittiin sellaiseksi, että listan tilaajien ohjelmat on helppo toteuttaa. Jos tulevaisuudessa offline-listan tietosisällölle tarvita erilaisia rakenteita eri toimijoille, myös tämä on huomioitu järjestelmän suunnittelussa. Listojen replikointijärjestelmä ei ota kantaa listojen sisällön rakenteeseen, ja samaa jakelurajapintaa voidaan käyttää eri muotoista tietoa sisältävien listojen jakeluun.

IV. Tietomäärien rajoittaminen

Koska offline-listan sisältöä välitetään suurelle määrälle laitteita, haluttiin välitettävät tietomäärät minimoida. Listojen jakelurajapinta ei ota kantaa listojen rivikohtaisen tiedon rakenteeseen, ja käsittelee listan sisältöä vain binääridatana. Tällöin eri offline-listojen rivisisältö voidaan konfiguroida erikseen ja varmistaa, että listoille ei lähetetä ylimääräistä sisältöä. Esimerkiksi eri leimauskanavien päätelaitteille jaetaan vain niille oleellista sisältöä. Pusatecin sisäisesti käyttämille listoille lippujen tiedot koodataan jakelun ajaksi kompaktiin binäärimuotoon protobuf-tekniikan avulla.

V. Tehokkuus, skaalautuvuus ja luotettavuus

Listojen replikaation toteutuksessa on huomioitu järjestelmän tehokkuusvaatimukset, joiden myötä toteutuksessa päätettiin käyttää hajautettuja ja hyvin skaalautuvia tekniikoita. Palveluiden ainoat riippuvuudet ovat Pub/Sub-viestintäkanava sekä Firestore-tietokanta, jotka molemmat ovat tehokkaita ja skaalautuvia palveluita. Tietokannan kyselymäärää vähennetään myös jakelurajapinnan pidettävien välimuistilistojen avulla. Listojen muodostus- ja jakelupalvelut jaettiin itsenäisiin kontteihin, joiden määrää ja niitä suorittavien Kubernetes-noodien resursseja voidaan tarvittaessa lisätä.

Tili- ja tapahtumamäärien kasvaessa järjestelmän rasituksen ja tietomäärien pitäisi suhteessa kasvaa kohtalaisen lineaarisesti. Luotettavuuden kannalta oleellista on listojen jakelun oikeellisuus ja se, että taustajärjestelmä ja rajapinta toimivat ilman katkoja. Taustajärjestelmää suoritetaan GCP:ssä, jossa palveluiden katkosten pitäisi olla erittäin harvinaisia.

VI. Toteutus osaksi nykyistä järjestelmää

Koska offline-listan muodostus ja jakelu toteutetaan osaksi olemassa olevaa lippujärjestelmälustaa, halutaan toteutuksessa käyttää yhteensopivia ja valmiiksi käytössä olevia tekniikoita ja ratkaisuja. Tämän luvun analyysin perusteella on havaittu, että toteutuksessa voidaan käyttää täysin samoja tekniikoita ja palveluita, kuin mitä tämänhetkinen tunnistepohjainen lippujärjestelmälusta käyttää. Valmiiksi käytössä olevat tekniikat soveltuvat hyvin myös offline-listan tarpeisiin, eikä niistä poikkeamista nähty tarpeelliseksi.

6 Toteutus

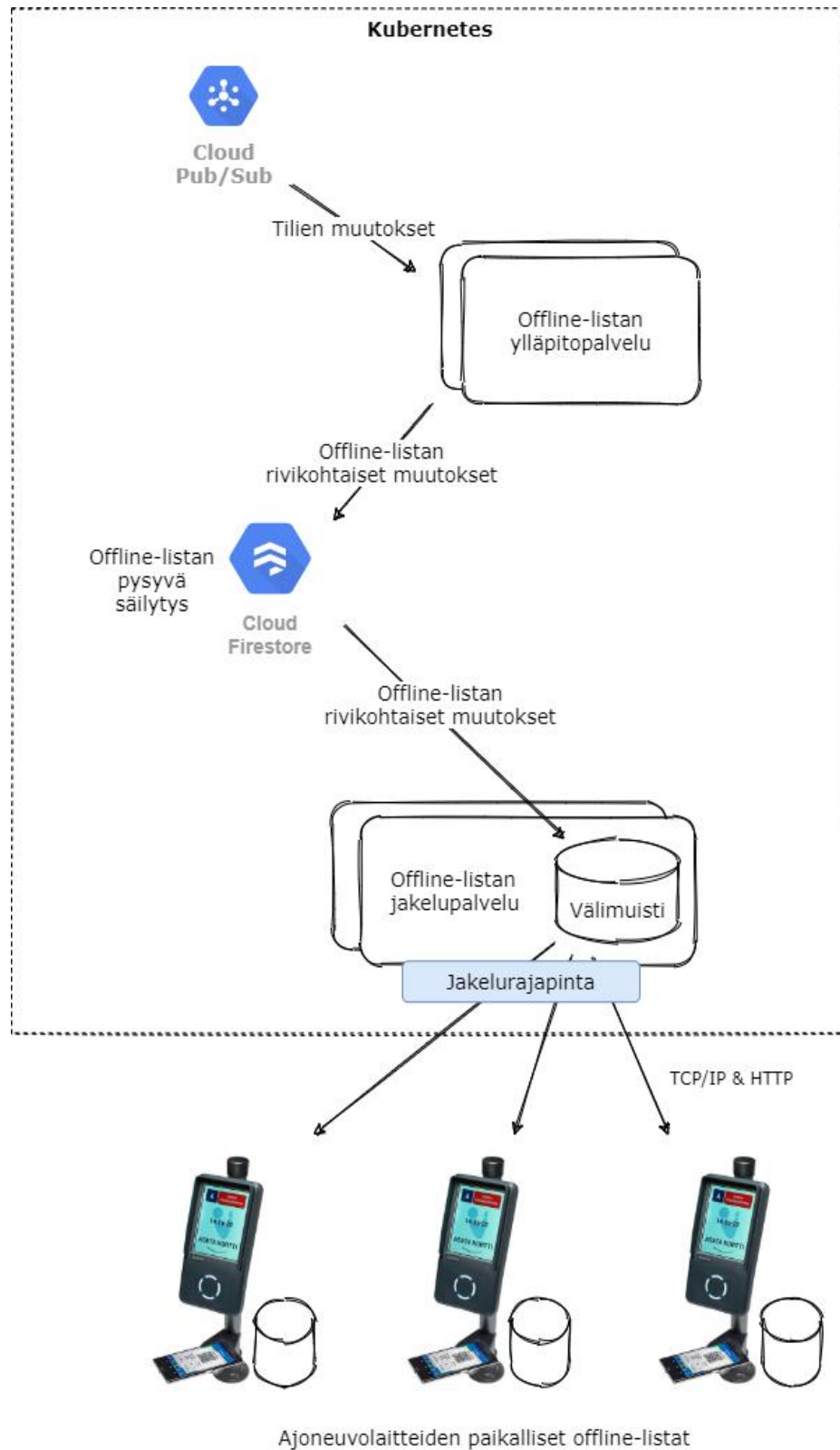
Tässä luvussa esitetään offline-listan muodostuksesta ja jakelusta vastaavan järjestelmän prototyypin toteutus, joka tehtiin osaksi Pusatec Oy:n tunnistepohjaista lippujärjestelmää. Toteutukseen sisältyy taustajärjestelmässä tapahtuva ensisijaisten offline-listojen muodostus ja säilöntä, sekä listojen jakamisesta vastaava jakelurajapinta. Leimauslaitteiden ohjelman toteutuksesta ei anneta kuvausta, mutta laitteiden offline-listaan liittyvästä toiminnallisuudesta ja listojen säilytyksestä on mainittu edellisessä luvussa 5.

Offline-listojen muodostus ja jakelurajapinta on liitetty osaksi olemassa olevaa tunnistepohjaista lippujärjestelmälustaa. Kuvassa 8 on esitetty kaavio järjestelmän arkkitehtuurista, josta näkyy offline-listojen sisältöjen muodostus ja jakelu listoja ylläpitävästä palvelusta leimauslaitteille asti. Kaaviossa näkyy offline-listojen ylläpitoa varten perustettu uusi palvelu, jota ajetaan muista palveluista erillisissä konteissa. Ylläpitopalvelu kuuntelee Pub/Sub-viestintäkanavaa pitkin tulevia tilien tiloja koskevia muutoksia ja päivittää pysyvässä säilössä olevia ensisijaisia offline-listoja muutosten mukaisesti. Kuvassa näkyy myös jakelurajapintaa varten perustettu uusi palvelu, joka jakelee ensisijaisten listojen sisältöä sitä pyytävälle tahoille. Uusien palveluiden suoritus tapahtuu samassa Kubernetes-klusterissa kuin olemassa olevien lippujärjestelmän palveluiden. Offline-listan palvelut kirjoitettiin TypeScript-ohjelmointikielellä ja niitä ajetaan Node.js-ajoympäristössä. Nämä valittiin sen vuoksi, että myös aikaisemmat palvelut on toteutettu samoilla tekniikoilla, ja ne soveltuvat hyvin palvelinohjelmien kirjoittamiseen.

6.1 Listojen konfigurointi

Jakelujärjestelmä toteutettiin niin, että offline-listojen sisällön formaatti voi vaihdella eri listojen välillä, ja sisällölle voidaan tulevaisuudessa määrittää helposti uusia formaatteja. Listan formaatti määritellään toteuttamalla rajapintaluokka, joka määrittää funktion listan rivin sisällön johtamiselle tilin tilasta. Rivin sisältö palautetaan binäärimuotoiseksi, jolloin sisältö voidaan koodata pieneen tilaan jo tässä vaiheessa. Järjestelmään voidaan helposti lisätä uusia formaatteja listojen sisällölle, kirjoittamalla rajapintaluokalle uuden toteutuksen.

Järjestelmään toteutettiin valmiiksi yksi sisältöformaatti, jossa listan sisältöön viedään kaikki käyttäjän kyseiselle leimauskanavalle kuuluvat liput, maksukyvykkyudet ja menneisiin leimauksiin liittyvät vaihto-oikeudet. Näiden tietojen avulla matkustusosoikeuden validointi voidaan toteuttaa Pusatecin leimauslaitteilla kohtalaisen yksinkertaisella validointilogiikalla.



Kuva 8 - Offline-listojen muodostuksen ja jakelun arkkitehtuuri

Offline-listojen muodostusta ohjataan offline-lista-konfiguraatietietojen avulla. Konfiguraatietiedot ovat lista muodostettavista listoista, joissa määritellään listojen tunnisteet, leimauskanava ja sisällön formaatti. Offline-listat on yhdistetty aina tiettyyn leimauskanavaan, joka määrittää

lippujen validointiin liittyvät säännöt. Tulevaisuudessa lista-konfiguraatioihin voidaan mahdollisesti lisätä erilaisia parametrejä, joilla ohjataan listojen tulevaa sisältöä. Listojen muodostuksesta vastaava ylläpitopalvelu vastaa siitä, että jokaiselle konfiguraatitiedoissa määritetyille listoille luodaan ensisijainen offline-lista tietokantaan.

6.2 Ensisijaisten listojen muodostus ja säilöntä

Offline-listojen sisältö johdetaan taustajärjestelmään tallennettujen matkatilien sisällöstä. Ensimmäisen käyttöönoton yhteydessä offline-listat muodostetaan käymällä läpi kaikki olemassa olevien tilien tilat. Koko listojen uudelleen muodostus voidaan suorittaa myös tarvittaessa uudestaan, esimerkiksi listan muodostuksen sääntöjen muuttuessa. Listojen ylläpitopalvelulle luodaan Pub/Sub-tilaus, joka kuuntelee matkatilien muutoksista syntyneitä viestejä. Viestit sisältävät uuden tilin tilan, josta johdetaan uusi sisältö jokaiselle konfiguraatiossa määritetyille offline-listalle. Rivin sisältöä verrataan ensisijaisella listalla olevaan nykyiseen riviin, ja mikäli sisältö on muuttunut, päivitetään ensisijaiselle listalle rivin uusi tila sisältö. Joissain tapauksissa johdettu sisältö voi olla tyhjä, jolloin tilistä johdettuja tietoja ei tarvita kyseiselle offline-listalle. Rivi voidaan tällöin jättää pois ensisijaiselta listalta. Mikäli ensisijaisella listalla on kuitenkin jo valmiiksi tiliä koskeva rivi, kirjoitetaan sen päälle tyhjä tietosisältö, jotta rivin sisällön poistuminen välittyy replikaation läpi päätelaitteille asti.

Ylläpitopalvelu kirjoittaa uudet rivisisällöt Googlen Datastore-dokumenttitietokantaan. Datastore on Googlen Firestorea läheisesti vastaava tietokanta, jota käytetään tällä hetkellä Pusatecin taustajärjestelmässä. Datastore-tietokanta on vanhentumassa, ja kantoja ollaan Googlen toimesta automaattisesti päivittämässä Firestore-kannoiksi vuoden 2021 aikana. Datastore ei takaa tietojen vahvaa yhtenäisyyttä, vaan kirjoitetuille tiedot luvataan tulevan lopulta yhteneväisiksi (engl. eventual consistency). Tämä otetaan huomioon jakelupalvelussa alaluvussa 5.3.6 esitetyllä tavalla. (Google, 2021e)

Listojen säilöntä ei ota kantaa offline-listalle tallennettavan tiedon sisällön rakenteelle tai koodaukselle, vaan rivien sisältöä käsitellään binääridatana. Rivin binäärisisällön lisäksi dokumentteihin tallennetaan offline-listan tunniste, tilin tunniste, sekä versiotieto. Tietokannan rivitietueiden rakenne on esitetty tarkemmin luvun 5.3.3 yhteydessä taulukossa 2. Rivien versiotietona käytetään rivin edellisen päivityksen aikaleimaa, ja kun riviin päivitetään uusi sisältö, versiokenttään asetetaan päivityshetken aikaleima millisekunnin tarkkuudella. Prototyypitoteutukseen toteutetun offline-listan sisältöformaatin tiedot koodataan binäärimuotoiseksi Protobuf-tekniikalla ennen tietokantaan tallentamista.

6.3 Listojen jakelu

Listojen jakelua varten perustettiin uusi palvelu Kubernetes-klusteriin. Palvelun vastuulla on tarjota rajapinta, jonka avulla offline-listaa tarvitsevat tahot voivat hakea ensisijaisten listojen sisältöä itselleen.

Jokaista konfiguraatioissa mainittua offline-listaa varten jakelupalvelu replikoi Datastoresta listojen sisällöt omassa muistissaan säilytettävälle välimuistilistoille. Jakelurajapintaan saapuneet kyselyt hakevat tietoa näiltä välimuistilistoilta, jolloin tietokantakyselyiden määrää saadaan karsittua. Välimuistilistoille haetaan Datastoren-kannasta uudet muutokset ensisijaisilta listoilta 10 sekunnin välein. Kyselyissä versiotiedon maksimiarvo rajoitetaan nykyistä aikaa 3 sekuntia aikaisempaa aikaleimaan, jotta tietokantaan tehdyt muutokset ovat ehtineet yhtenevään tilaan, eikä muutosrivejä jää välistä.

Välimuistilistoja säilytetään ainakin prototyyppivaiheessa Node.js-alustan Map-tietorakenteessa, jossa avaimena käytetään tilien tunnisteita. Tietyn version jälkeiset muutokset voidaan hakea tietorakenteesta yksinkertaisella suodatusoperaatiolla käymällä läpi tietorakenteen rivit ja vertaamalla niiden versionumeroita kysytyyn versionumeroon. Kirjoitusoperaatiossa tiliä koskevan rivin sisältö korvataan Map-tietorakenteeseen sen avaimena toimivan tilin tunnisteen avulla. Prototyyppivaiheen välimuistirakenteen toteutus on hyvin yksinkertainen, eikä välttämättä tehokkain ratkaisu. Välimuistilistojen säilyttämiseen käytettävä tietorakenne voidaan tulevaisuudessa toteuttaa nykyistä toteutusta tehokkaammin, mikäli rakenteen suorituskyvyssä havaitaan puutteita.

Jakelupalveluun pystytettiin HTTP-palvelin Node.js ympäristöön tarkoitetun Express-kirjaston avulla. Rajapinnasta voidaan kysyä offline-listojen sisältöä HTTP-pyyntöjen avulla. Pyyntöihin vastataan välimuistilistoilta haetuilla muutoserillä. Rajapinnan käyttäjät tunnistautuvat pyyntöihin sisälletyn autentikointi-tokenin avulla, joka mahdollistaa rajapintaa käyttävien tahojen seurannan ja käyttöoikeuksien hallinnan.

Pyyntöjen kyselyparametreissa ilmoitetaan offline-listan tunniste ja nykyinen versionumero, sekä mahdollinen maksimi eräkoko. Palvelin vastaa pyyntöön uudella versionumerolla ja erällä välimuistilistoilta luettuja rivien muutoksia, jotka sisältävät tilin tunnisteen ja offline-listan konfiguroinnissa valitun formaatin mukaisen rivisisällön. Vastauksessa sisällytetään uudestaan kysytyn versionumeron rivit, jotta mahdollisista päällekkäisistä versionumeroista ei aiheudu rivien väliin jäämistä. Vastauksessa kerrotaan myös välimuistilistojen tämänhetkinen uusin versionumero ja tieto siitä, onko listoilla kyseisen erän jälkeen vielä uudempia rivejä saatavilla, vain onko paikallinen lista nyt ajan tasalla.

Jakelurajapinta voi vastata pyyntöihin Protobuf- tai JSON-muotoisilla vastauksilla. Protobuf-muodossa koko rajapintakutsun vastaus koodataan binäärimuotoon. Tämä eroaa itse offline-listan rivien tietueiden sisällön koodauksesta, joka tehtiin jo listojen muodostuksen yhteydessä. Vastauksessa on tällöin tavallaan kahteen kertaan koodattua sisältöä. Ulompi kerros, eli koko rajapintakutsun vastauksen kerros,

voidaan purkaa välittömästi päätelaitteella niin, että tietueiden sisällön koodaus säilyy kuitenkin leimaushetkeen asti. Binäärisisältö on tarkoitettu ensisijaiseksi muodoksi jota päätelaitteet käyttävät omien listojen päivittämiseen. JSON-muotoa voidaan käyttää testaukseen ja vianetsintään, ja se mahdollistaa myös rajapinnan yksinkertaisemman käytön tarvittaessa muille toimijoille. Muoto voidaan valitaan HTTP-kyselyparametrien avulla. Binäärimuodossa vastaukseen vastataan suoraan sisällöllä, joka koodattiin jo listojen muodostusvaiheessa binäärimuotoon Protobuf-tekniikalla. HTTP-vastauksen Content-Type-otsikkotietoon asetetaan binäärisisällöstä kertova arvo "application/octet-stream". JSON-muotoisessa vastauksessa välimuistilistojen sisällön binäärikoodaus puretaan ja pyyntöön vastataan JSON-standardin mukaisella tekstisisällöllä. Listaformaatin rajapintaluokassa määritellään funktio binäärimuotoisen koodauksen purkamiselle JSON-muotoon sopivaan muotoon. Muoto on edelleen siis offline-listan formaatin hallittavissa, jolloin uusien formaattien lisääminen järjestelmään on helppoa. HTTP-vastaukset pakataan gzip tai deflate -algoritmien avulla, mikäli HTTP-kyselyn otsikkotiedoissa mainitaan tuki algoritmeille. Vastauksen otsikkotiedoissa kerrotaan, mitä pakkausalgoritmia käytettiin.

7 Toteutuksen evaluaatio

Offline-validointia ei oteta käyttöön vielä tämän työn aikana. Edellä esitetyn jakelujärjestelmän prototyypin toteutuksella pyritään arvioimaan offline-validoinnin käyttökelpoisuutta, mitä voidaan hyödyntää tulevassa päätöksenteossa. Tässä aluvussa evaluoidaan järjestelmän toteutusta ja käyttökelpoisuutta. Tämä tehdään analysoimalla prototyypin toteutusta ja simuloimalla järjestelmän toimintaa seuraten samalla toiminnan oikeellisuutta ja suorituskykyä. Testit suoritettiin kirjoittajan työtietokoneella, jolla suoritettiin molempia offline-listan palveluita samassa prosessissa, sekä Datastore ja Pub/Sub -palveluiden toimintaa emuloivia kontteja.

Testiympäristön offline-listakonfiguraatioihin määritettiin yhden offline-listan muodostus. Listojen sisältö muodostetaan tietyille testiympäristön leimauskanavalle. Tilien matkustusoikeuksien muutoksia saatiin aikaan luomalla rajapintakutsuilla uusia matkatilejä, ja suorittamalla niille lippujen myynti- ja peruutuskutsuja. Rajapintakutsut muuttavat tilien matkustusoikeuksia, ja muutosten pitäisi välittyä offline-listan muodostus- ja jakelupalveluiden läpi.

Offline-listaa hakevia laitteita simuloitiin kyseistä testaamista varten kirjoitetulla Node.js-ohjelmalla. Ohjelma luo tietyn määrän päätelaitteita simuloivia instansseja, jotka hakevat 30 sekunnin välein jakelurajapinnasta muutoksia, ja päivittävät ne omille paikallisille offline-listoille. Ohjelmalla voi simuloida useiden päätelaitteiden aiheuttamaa raskautusta samanaikaisesti.

7.1 Järjestelmän oikeellinen toiminta

Koska validointipäätökset tehdään offline-listan sisällön perusteella, on tärkeää, että jokainen tilien matkustusoikeuksien muutos välittyy offline-listoille oikein. Toimintaa testattiin järjestelmän rajapintoihin suoritettujen kutsujen avulla. Testissä yritettiin varmistua siitä, että listojen muodostus ja jakelu toimivat oikein, ja kaikki listoille kuuluva tieto välittyy rajapinnasta haettaville muutoseriille.

Testi suoritettiin tekemällä rajapintoihin useita samanaikaisia lippujen myyntikutsuja ja tehtyjen myyntien peruutuskutsuja. Nämä kutsut muuttavat taustajärjestelmän tilien matkustusoikeuksia, ja oikeuksien muutosten tulee lopulta näkyä offline-listalla. Samanaikaisesti testissä ylläpidettiin paikallista offline-listaa, hakemalla rajapinnasta muutoseriä 5 sekunnin välein. Testissä tarkastettiin, että jokainen rajapintakutsun aiheuttama muutos päättyi lopulta paikalliselle offline-listalle. Muutoseriä tarkastettiin myös, että erät sisältävät päällekkäisiä versionumeroita, jotta testissä varmistetaan, etteivät ne aiheuta rivien puuttumisia. Matkustusoikeuksia muuttavia kutsuja tehtiin 58 sekunnin aikana 400 kappaletta.

Testissä mitattiin myös järjestelmän vasteaikaa siitä, kun myynti-rajapintaan tehdään uuden matkustusoikeuksia muuttava kutsu, siihen, kun tiedot löytyvät paikalliselta offline-listalta. Kuten

vaatimuksessa II mainittiin, vasteaika on oleellinen järjestelmän toiminnan kannalta siksi, ettei tulevia validointeja tehdä vanhentuneen tilan perusteella.

Testissä järjestelmä toimi niin kuin odotettiin, ja kaikki matkustusosoikeuksien muutokset välittyivät offline-listalle. Muutosten välittyminen järjestelmän läpi kesti keskimäärin 17 sekuntia, joten testin perusteella muutosten välittymisaika saadaan ainakin taustajärjestelmän osalta pidettyä vaatimuksessa II esitettyä 2 minuutin aikarajaa pienempänä. Lopullinen nopeus riippuu kuitenkin taustajärjestelmän lisäksi päätelaitteiden toteutuksesta ja tiedonsiirtoyhteyksien toiminnasta. Päätelaitteet voivat päivittää omia listojaan esimerkiksi minuutin välein, jolloin aikatavoitteeseen todennäköisesti päästään, ellei verkkoyhteydessä ole ongelmia.

7.2 Järjestelmän tehokkuus ja skaalautuvuus

Vaatimuksessa V esitettiin tarve hyvin skaalautuvalle ja tehokkaalle järjestelmälle. Koska järjestelmää suoritetaan Googlen pilvialustan päällä, suorituskyvyn resursseja voidaan tarvittaessa saada erittäin suuria määriä, eikä niiden riittävyys ole esteenä järjestelmän toiminnalle. Järjestelmän tehokkuudessa onkin enemmän kyse siitä, onko kuormitus ja siitä syntyvät kustannukset kohtuullisia järjestelmän toimintaan ja siitä saatavaan arvoon nähden. Järjestelmän tulisi skaalautua tehokkaasti kuorman kasvaessa.

Offline-listojen muodostus- ja jakelujärjestelmä toteutettiin hyvin skaalautuvalla arkkitehtuurilla. Järjestelmä on jaettu itsenäisiin palveluihin, joista voidaan suorittaa useampaa konttia samanaikaisesti. Kuormituksen kasvaessa konttien määrää ja Kubernetes-noodien suorituskykyä voidaan tarvittaessa lisätä, ja kuorman pitäisi jakaantua tasaisesti eri konttien välille. Offline-listojen ylläpito- ja jakelupalveluiden ainoat riippuvuudet ovat jaettu tietokantakerros, eli Datastore, ja Pub/Sub-viestintäkanava. Molemmat näistä on tarkoitettu erittäin hyvin skaalautuviksi tekniikoiksi (Google, 2021b & 2021f).

Järjestelmän kuormitus muodostuu keskeisesti kahdesta muuttujasta; matkustusosoikeuksien muutosten tiheydestä ja muutoksia kysyvien tilaajien määrästä. Muutostiheys lisää listan rivien sisältöjen muodostuksia, tietokannan kirjoitusoperaatioita ja replikoitavia tietomääriä. Tilaaajien määrä vaikuttaa puolestaan muutosten jakelurajapintaan ja välimuistilistoihin kohdistuvaan kuormitukseen. Näiden lisäksi listojen rivien kokonaismäärä saattaa vaikuttaa välimuistilistoista tehtävien lukuoperaatioihin kuormitukseen.

Listojen muodostus- ja jakelujärjestelmää kohtaan suoritettiin räsistestit, jossa selvitettiin, minkälaisia kuormia järjestelmän pystyy tietyillä suorituskykyresursseilla käsittelemään. Tällä pyritään saamaan ymmärrystä järjestelmän toiminnasta syntyvistä kustannuksista ja suoritukseen tarvittavista Kubernetes-noodien resursseista. Testissä Pub/Sub ja Datastore -palveluista suoritettiin kehityskäyttöön

tarkoitettuja emuloituja versiota, joiden suorituskyky ei välttämättä vastaa todellisten GCP-ympäristön palveluiden toimintaa, eikä siitä siksi ei kannata tämän testin perusteella tehdä johtopäätöksiä.

Järjestelmän toiminnassa muutosten jakelurajapinta ja välimuistilistat on suurimman rasituksen kohteena listaa kysyvien päätelaitteiden suurien määrien johdosta, ja tämän vuoksi testi päätettiin kohdistaa rajapintaan. Offline-listalle luotiin valmiiksi 100 000 kpl rivejä, joita simuloitujen tilaajien lähtivät lataamaan omille paikallisille offline-listoilleen. Tilaajat hakivat muutoksia 10 000 muutoksen erissä ilman kutsujen välisiä viiveitä. Tämä on järjestelmään kohdistuvan rasituksen kannalta pessimistinen tilanne, jossa useat listan tilaajat aloittavat samanaikaisesti koko listojen lataamisen alusta. Todellisuudessa suurin osa tilaajista on todennäköisesti jo ajan tasalla, ja ne hakevat vain esimerkiksi minuutin välein uudet muutokset. Offline-lista-järjestelmän prototyypitoteutusta suoritettiin kirjoittajan työtietokoneella, jossa on 16 Gt keskusmuistia ja neliytiminen Intel® Core™ i7-8550U prosessori, jonka perus kellotaajuus on 1.99GHz. Tilaajien simulointi suoritettiin offline-lista järjestelmää suorittavasta tietokoneesta erillisellä tietokoneella, jotta simulointiohjelman suorituksista syntyvä rasitus ei vaikuta tarkasteltavan järjestelmän toimintaan. Testissä mitattiin, kuinka kauan offline-listan replikoimisessa kesti. Tilaajamäärää nostettiin jatkuvasti, kunnes rajapinnan toiminnassa huomattiin häiriöitä. Taulukossa 3 on esitetty testissä käytetyt tilaajien määrät ja koko listan päivitykseen kulunut viive. Yli 800 tilaajamäärällä simulointiohjelmaa suorittavasta tietokoneesta loppui muisti kesken suorituksen, joten listojen päivityksen viiveelle ei siksi saatu tulosta. 1600 tilaajan määrällä rajapinta suoriutui kuitenkin vielä oikeellisesti ja kohtalaisen tehokkaasti kuormasta. 3200 tilaajan kohdalla rajapintakutsut alkoivat heittää timeout-virhettä.

Taulukko 3 - Rasitustestissä käytetyt tilaajien määrät ja listojen päivitykseen kuluneet viiveet

TILAAJIEN MÄÄRÄ	LISTOJEN PÄIVITYKSEN VIIVE (S)
1	2,47
10	10,8
100	79,9
200	155
400	305
800	645
1600	
3200	Kutsut epäonnistuivat

Testissä havaittiin, että paikallinen yhdessä prosessissa suoritettu jakelurajapinta suoriutui tehtävästä kohtuullisen hyvin. Testi oli todella pessimistinen rajapintaan kohdistuvan käytön osalta, sillä kaikki listojen tilaajat alkoivat lataamaan samaan aikaan koko listan sisältöä alusta asti rajapinnan kautta.

Testiin käytetty tietokone suoriutui kohtalaisen suuristakin kuormista hyvin. Tietokoneen suorituskyky ei ole erityisen korkea, ja GCP:n tarjoamista vaihtoehdoista löytyy vastaavaan suoritukseen kykeneviä virtuaalikoneita kohtalaisilla kustannuksilla. Pilviympäristössä kuormaa jaetaan myös useiden konttien ja Kubernetes-noodien välille, kun testissä kaikki kuorma kohdistui yhteen prosessiin.

Ainakin rajapinnan osalta järjestelmä näyttäisi siis suoriutuvan riittävältä kuormalta. Työn aikataulun takia järjestelmälle ei suoritettu muita rasiustestejä. Listojen muodostuspuoli voidaan todeta kuitenkin myös riittävän suorituskykyiseksi edellisen alaluvun testin perusteella, jossa matkustusoikeuksien muutoksia suoritettiin 400 kappaletta sekunnissa, ja järjestelmä suoriutui tästä kuormasta. Ajan niin salliessa järjestelmään olisi kannattanut kohdistaa myös laajempi rasiustesti, jossa listojen muodostuspuolta rasiutettiin, ja tarkasteltiin matkustusoikeuksien muutosten tiheyden ja listojen kokonaisuuden vaikutusta rasiutukseen ja resurssien käyttöön. Hajautetun järjestelmäkokonaisuuden skaalautuvuutta ei toistaiseksi testattu, vaan luotetaan siihen, että hajautettu arkkitehtuuri ja GCP:n tehokkaat ja hyvin skaalautuvat palvelut tekevät järjestelmästä hyvin skaalautuvan.

7.3 Haasteet ja rajoitukset

Toteutuksen yhtenä haasteena oli hajautetun järjestelmän asettamat haasteet avainpohjaiselle replikaatiolle. Ensisijaisten listojen säilytykseen valittu tietokanta Datastore käyttää hajautettua arkkitehtuuria, eikä takaa kyselyille vahvaa yhteneväisyyttä, mikä täytyi ottaa huomioon jakelupuolella. Replikaation avaimena käytettiin aikaleimaa, jonka avulla tämä ongelma saatiin ratkaistua melko helposti niin, että jakelupuolen tietokantakyselyissä luettujen rivien versiot rajoitettiin 3 sekuntia myöhemmäksi kuin uusien rivien versionumerot. Google on korvaamassa Datastore-tietokantaa Firestore-kannalla, joka mahdollistaa kyselyiden vahvan yhteneväisyyden, joten kyseinen ongelma poistuu päivityksen jälkeen (Google, 2021e). Hajautetusta järjestelmästä seuraa myös se, että offline-listojen riveille sallitaan päällekkäiset versionumerot. Tämä täytyi ottaa huomioon replikaatiossa sisällyttämällä edellisen muutoserien viimeisimmän version uudestaan seuraavaan muutoseriin. Ratkaisut vaikuttivat toimivilta, ja suoritetuissa testeissä järjestelmä toimi täsmällisesti. Ne lisäävät kuitenkin monimutkaisuutta replikaatioon, ja varsinkin päällekkäisten rivien sisällyttäminen eri muutoseriin saattaa aiheuttaa hämmennystä rajapintaa käyttäville toimijoille. Ainutlaatuisen nousevan versionumeron luominen keskitetysti ratkaisisi päällekkäisistä versionumeroista johtuvan ongelman. Tämä kuitenkin luo uuden riippuvuuden versioiden luomisesta vastaavaan palveluun tai tietokantaan, ja saattaa aiheuttaa pullonkaulan hajautetulle järjestelmän suorituskyvylle. Vaihtoehtoinen ratkaisu voisi olla käyttää versionumerona erittäin tarkkaa aikaleimaa, ja luottaa siihen, että päällekkäisiä versionumeroiden syntymiselle on niin pieni todennäköisyys, että niiden mahdollinen vaikutus järjestelmän toimintaan on häviävän pieni. Toinen ratkaisu voisi olla, että versiotietoa käytetään rivien pääavaimena, jolloin kirjoitusvaiheessa tietokantakerros voi varmistaa, että operaatio ei

kirjoita olemassa olevan rivin ylitse. Tällöin ensisijaiselta listalta täytyy kuitenkin poistaa edelliset samaan tunnisteeseen liittyvät rivit, jotta replikoinnissa ei välitetä ylimääräistä vanhentunutta tietoa. Koko tietokantaoperaatiota täytyy tällöin myös ajaa hallitusti transaktiossa, jotta tietyn operaation epäonnistuessa ensisijaisen listan rivit säilyvät kannassa oikein.

Koska myynti- ja leimaustapahtumien aiheuttamien muutoksia päivittyminen laitteiden offline-listoille kestää hetken, validointi saattaa perustua vanhentuneeseen tilaan. Tällaisissa tilanteissa validointi ei ole täysin täsmällinen, ja varsinkin juuri ennen matkaa hankittuja matkustusoikeuksia ei välttämättä validoida oikein. Myös juuri käytetyt liput saatetaan hyväksyä vielä hetken aikaa niiden käytön jälkeen. Tämä mahdollistaa täsmällisesti kohdennetun väärinkäytön, jossa samaa tunnisteella leimataan useaan kertaan lyhyen ajan aikana. Esimerkiksi paperisia kertalippuja kopioimalla kaksi matkustajaa voi huijata järjestelmää nousemalla samanaikaisesti ajoneuvojen kyytiin.

Offline-listojen tiedoista saattaa olla pääteltävissä joitain matkatilien henkilötietoina pidettäviä tietoja, kuten tilin ikäryhmä, alennusryhmän ja asuinkunnan. Nämä tiedot ovat kuitenkin niin epätarkkoja, että tietoja on hyvin vaikea liittää tiettyyn henkilöön ilman että tiedot yhdistetään tunnisteiden kautta taustajärjestelmässä säilytettäviin henkilötietoihin. Tietojen päätyemisestä ulkopuolisten tahojen haltuun ei arvella seuraavan käyttäjien tietosuojaan liittyvää riskiä, ja tiedot voidaan siksi tallennetaan ainakin toistaiseksi salaamattomana päätelaitteiden massamuistiin.

8 Yhteenveto

Tässä työssä tarkasteltiin tunnistepohjaisten lippujärjestelmien matkustusoikeuden offline-validoinnin tarvetta ja toteutusmahdollisuuksia. Offline-validoinnissa taustajärjestelmässä säilytettävä matkustusoikeus validoidaan päätelaitteella paikallisesti ilman taustajärjestelmään suoritettavaa rajapintakutsua. Tutkimus tehtiin etsimällä tietoa alan kirjallisuudesta ja vastaavien järjestelmien nykyisistä ratkaisuista, ja analysoimalla ja ratkaisemalla ongelmaa Pusatecin tunnistepohjaisen lippujärjestelmän osalta. Työlle asetettiin kaksi tutkimuskysymystä, joihin vastataan seuraavaksi.

TK 1. Mikä on offline-validoinnin tarve ja siitä saatavat hyödyt tunnistepohjaisessa lippujärjestelmissä?

Pusatecin järjestelmän online-validointien toimivuutta analysoimalla alaluvussa havaittiin, että online-validointi onnistuu noin 99,5 % ajasta. Tätä voidaan pitää luotettavana toimintana, jonka ansiosta tiedonsiirtoyhteyksistä johtuva lipputulotappio on kohtalaisen pientä. Järjestelmän systemaattinen väärinkäyttö on vaikeaa, sillä se vaatii, että käyttäjät havaitsevat milloin päätelaitteen tiedonsiirtoyhteys on poikki, ja nousevat kyytiin tällä hetkellä. Toisaalta tarkoituksetonta väärinkäyttöä saattaa syntyä, esimerkiksi tilanteissa, joissa käyttäjä ei ole varma onko hänen lippunsa edelleen voimassa tai onko hänellä maksukykyä jäljellä, ja päättää yrittää ajoneuvoon nousua joka tapauksessa.

Analyysissä havaittu online-validointien viive on osittain alan kirjallisuudesta löydettyjä suosituksia suurempi. Tämä saattaa vaikuttaa oleellisesti lippujärjestelmän käyttökokemukseen ja toiminnan sujuvuuteen, sillä matkan aloituksen yhteydessä tapahtuvan leimauksen hitaus hidastaa matkustajien kulkeutumista liikennevälineisiin. Offline-validointi mahdollistaa nopeamman leimauksen, ja siten parantaa käyttökokemusta. Vaikutusta käyttökokemukseen on kuitenkin vaikea arvioida pelkästään tämän työn kvantitatiivisilla menetelmillä, ja kuvaavampia tuloksia varten tarvitaan lisää tutkimusta.

Tiedonsiirtoyhteyksien analysoinnissa käytetty data on peräisin Jyväskylän alueen paikallisliikenteen ajoneuvolaitteilta. Tulokset eivät välttämättä edusta muiden kaupunkiseutujen tilannetta, mutta tiedossa ei myöskään ole syitä, jotka tekisivät alueesta muihin asutuskeskuksiin verrattuna poikkeavan. Toimintaa tarkasteltiin myös taustajärjestelmän lokeista, jolloin data on peräisin useista eri Suomen joukkoliikennejärjestelmästä, mutta kuitenkin pääosin kaupunkiseutujen paikallisliikenteestä. Haja-asutusalueilla mobiiliverkkojen toiminta voi olla heikompa kuin asutuskeskusten alueilla, ja analyysin tulokset eivät välttämättä siksi päde asutuskeskusten ulkopuolella. Tuloksia ei myöskään voi yleistää esimerkiksi Suomen ulkopuolelle, jossa verkkojen toiminta voi olla hyvin erilaista. Myös päätelaitteiden

elektroniikka ja ohjelmisto vaikuttavat tiedonsiirron toimivuuteen, eivätkä muiden toimittajien laiteratkaisujen toiminta välttämättä vastaa havaittua toimintaa.

Offline-validoinnilla voidaan lisätä myös järjestelmän viansietoisuutta. Taustajärjestelmän vikatilanteissa online-validointi saattaa estyä, mikä aiheuttaa vakavan ongelmatilanteen, jossa matkustusoikeuksia ei kyetä validoimaan. Vaikutuksia matkustajille voidaan minimoida esimerkiksi niin, että päätelaitteet hyväksyvät oletuksena kaikki niille esitetyt tunnisteet, mutta tämä saattaa aiheuttaa merkittäviä lipputulotappioita. Koska esimerkiksi Pusatecin taustajärjestelmää suoritetaan Kubernetes-alustan päällä, käyttökatkoja ei pitäisi syntyä edes esimerkiksi päivitystilanteissa. Kuitenkin poikkeuksellisissa virhetilanteissa validointirajapinta ei välttämättä ole käytettävissä.

Taustajärjestelmän vikatilanteista johtuvaa käyttökatkoa voidaan rinnastaa tiedonsiirtoyhteysongelmista johtuviin online-validointikutsujen epäonnistumisiin. Molemmissa tilanteissa online-validoinnin epäonnistuminen aiheuttaa tietynsuuruisen lipputuloriskin. Lipputuloriskin hyväksyntä voi olla tietoinen päätös lippujärjestelmää operoivalta taholta, mikäli sen estämiseen tarvittavat toimet ja tekniset järjestelmät vaativat liikaa resursseja tappioiden suuruuteen nähden. Tämänkaltaisten päätösten tekeminen on kuitenkin tämän työn ulkopuolella, ja työssä tarkasteltiin lähinnä järjestelmien teknistä toimivuutta.

TK 2. Miten offline-validointi kannattaa toteuttaa, ja mitä haasteita ja ratkaisuja tähän liittyy?

Offline-validointiin löydettiin erilaisia toteutusmahdollisuuksia, joita esitettiin alaluvussa 3.2. Pusatecin kannalta parhaaksi tavaksi havaittiin päätelaitteille ladattavat offline-listat, joista matkustajien matkustusoikeus voidaan johtaa luetun tunnisteiden ja leimausparametrien avulla. Offline-listat toimivat kaikille tunnistemedioille, ja niitä voidaan pitää ajan tasalla taustajärjestelmän tilan kanssa, jolloin validointi on kohtalaisen täsmällistä. Tiedonsiirto- ja datamääräanalyysin perusteella listojen toteutus on mahdollinen ja käyttökelpoinen. Myös muodostettu prototyypitoteutus ja sen evaluaatio tukevat tätä päätelmää. Analyysissä käytetyt matkustaja- ja päätelaitemäärät vastaavat Suomen mittakaavalla tyypillistä lippujärjestelmien rasitusta, joten näiden tulosten pitäisi yleistyä Suomen muihin järjestelmiin.

Koska muutosten välittyminen offline-listoille kestää tietyn aikaa, validointi ei ole täysin täsmällistä juuri tapahtuneiden muutosten osalta. Varsinkin jos tiedonsiirtoyhteyden kanssa on ongelmaa, saattaa offline-listojen tila ja sen perusteella tehtävä validointi vanhentua. Tästä seuraa hyvin pieni lipputuloriski, jossa käyttäjät voivat esimerkiksi nousta kyytiin vielä hetken matkustusoikeuden päätyttyä, tai useampi matkustaja voi yrittää nousta kyytiin samanaikaisesti samalla kopioidulla QR-paperilipulla. Myös tilanteissa, joissa matkustaja on juuri hankkinut matkustusoikeuden, aiheuttaa virheellinen validointi käyttäjälle huonon käyttökokemuksen. Tilanne, jossa käyttäjä ostaa esimerkiksi mobiililipun juuri ennen

matkaa, on luultavasti todella yleinen. Tämän takia offline-validoinnin hylätessä tunnisteen kannattaa varmuuden vuoksi vielä suorittaa online-validointi.

Myös mobiililippuihin sisällytettävät, luvussa 3.2.3 esitetyt, offline-tiedot voivat olla hyvä ratkaisu matkustusoikeuden offline-validointiin. Varsinkin jos mobiililippujen käyttö on toteutettu tavalla, jossa käyttäjät esittävät tietyn lipun tunnisteen lukijalaitteelle QR-koodin avulla. Taustajärjestelmästä haettavan tunnisteen yhteyteen voidaan liittää muita lipun tietoja, joiden avulla lippu voidaan validoida päätelaitteella. Mobiililippujen tiedot mahdollistavat offline-validoinnin välittömästi lipun oston jälkeen. Tiedot voivat kuitenkin vanhentua, mikäli niitä ei haeta uudestaan jatkuvasti taustajärjestelmästä. Koska leimaushetkellä tiedonsiirto on vain yksisuuntaista puhelimelta päätelaitteelle, ei tietoa lipun käytöstä saada esimerkiksi mobiilitiedonsiirtoyhteydet sulkeneelle mobiililaitteelle. Lipun tietojen vanheneminen ei kuitenkaan ole riksi, mikäli liput ovat voimassa aina tietyn mittaisen ajan ja voimassaolo käynnistetään sovelluksessa ennen käyttöä, jolloin offline-tiedot voivat pysyä staattisina. Jos kuitenkin mobiililipun offline-tietojen vanhentumisesta ja siitä johtuvasta virheellisestä validoinnista seuraa merkittävä lipputuloriski, kannattaa tätä validointia käyttää varavaihtoehtona vain online-validoinnin ollessa estyneenä.

Tunnistepohjaisuuteen siirtymisen yhtenä tavoitteena on siirtää tietojen säilytys ja toimintalogiikkaa älykorteilta ja päätelaitteilta taustajärjestelmään. Offline-validoinnin toteutus täytyy kuitenkin toteuttaa osittain päätelaitteille, mikä lisää niille tarvittavaa päättelyä ja laiteresursseja. Moderneilla päätelaitteilla laiteresurssien ei pitäisi kuitenkaan muodostua ongelmaksi. Offline-validoinnin toteutus ja validointilogiikan osittainen hajauttaminen taustajärjestelmästä takaisin päätelaitteelle lisää kuitenkin järjestelmän kompleksisuutta. Myös listojen muodostus- ja jakelupalvelut lisäävät taustajärjestelmän monimutkaisuutta ja mahdollisia ongelmakohtia. Matkustusoikeuksien validointi on todella keskeinen osa koko lippujärjestelmän oikeellista toimintaa, ja mikäli listojen muodostuksessa tai jakelussa on ongelmia, jotka johtavat virheellisiin validointeihin, saattaa tästä aiheutua todella merkittäviä ongelmia ja tappioita.

Työssä lippujen validoinnin toimintaa tarkasteltiin vain teknisen toimivuuden kannalta. Työn tuloksena saatiin arvioita siitä, millä todennäköisyydellä online-validointi onnistuu ja kuinka hyvin offline-validoinnilla voidaan parantaa validointien onnistumisprosenttia ja nopeutta. Tuloksista voi kuitenkin olla vaikea tehdä johtopäätöksiä siitä, mikä on offline-validoinnin todellinen vaikutus järjestelmän käyttökokemukseen. Myöskään liiketoiminnallisista kysymyksistä, kuten siitä, minkä suuruista lipputuloriskiä ollaan valmiit hyväksymään yksinkertaisemman järjestelmän ja paremman käyttökokemuksen vuoksi, ei käsitelty tässä työssä, ja niiden selvittämiseen tarvitaan lisää tutkimusta.

Lähdeluettelo

Euroopan parlamentin ja neuvoston asetus (EY) N:o 1370/2007. Saatavilla sähköisesti <http://data.europa.eu/eli/reg/2007/1370/oj>

Joukkoliikennelaki 869/2009. Annettu Helsingissä 13 päivänä marraskuuta 2009. Saatavilla sähköisesti <https://finlex.fi/fi/laki/alkup/2009/20090869>

Laki liikenteen palveluista 24.5.2017/320. Saatavilla sähköisesti <https://www.finlex.fi/fi/laki/ajantasa/2017/20170320>

Barkaway, David (2009). "Change Data Capture and the Benefits to the Modern Enterprise Data Warehouse". SAS Institute Inc., Cary, NC. Paper 303-2009. Haettu 28.5.2021 osoitteesta <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.227.54&rep=rep1&type=pdf>

Calypso (29.06.2017). "Account Based Ticketing with Calypso". Calypso White Paper. Calypso Networks Association. Haettu 5.4.2021 osoitteesta http://www.its.be/sites/default/files/170529-CalypsoWhitePaperABT_%20v2.0P10.pdf

Campos Ferreira, Martha - Dias, Teresa Galvão - Falcão e Cunha, João (2020). "Is Bluetooth Low Energy feasible for mobile ticketing in urban passenger transport?". Transportation Research Interdisciplinary Perspectives, Volume 5. ISSN 2590-1982, <https://doi.org/10.1016/j.trip.2020.100120>

Cheng, S. K. (4/2017). "Exploring Mobile Ticketing in Public Transport - An analysis of enablers for successful adoption in The Netherlands". Analysis report. Delft University of Technology. Faculty of Industrial Design Engineering. Haettu 28.4.2021 osoitteesta https://d1rkab7tlqy5f1.cloudfront.net/IO/Onderzoek/Delft_Design_Labs/OV-chipkaart_Graduation_Lab/TUD%20-%20Mobile%20Ticketing%20-%20Analysis%20Report%20-%20Cheng.pdf

Chris Kemmerer (03.03.2015). "The SSL/TLS Handshake: an Overview". SSL.com article. Haettu 22.6.2021 osoitteesta <https://www.ssl.com/article/ssl-tls-handshake-overview/>

CPI Card Group (2016). "A Prepaid EMV® Dilemma: To Migrate or not to Migrate". CPI Card Group Whitepaper. Haettu 22.6.2021 osoitteesta https://www.fintechfutures.com/files/2016/05/CPI_EMV_Whitepaper_PrepaidEMVDilemma.pdf

Eskelinen, Juuso (2019). "Tietokannan replikaatio yksisuuntaisia ei-pysyviä yhteyksiä käyttäen". Pro gradu - tutkielma. Jyväskylän yliopisto, Informaatioteknologian tiedekunta, Informaatioteknologia, Tietojärjestelmätiede. URN:NBN:fi:jyu-201906143209. <http://urn.fi/URN:NBN:fi:jyu-201906143209>

Faraz, Muhammad (07.09.2020). "Database Replication: A Comprehensive Guide". Hevo Data. Tutorials. Haettu 9.5.2021 osoitteesta <https://hevodata.com/learn/database-replication>

Föli (2017). "Turun seudun joukkoliikenteen palvelu- ja myyntiohje". Turun seudun joukkoliikenne, Turun kaupunkiseudun joukkoliikennelautakunta. Haettu 28.5.2021 osoitteesta <https://ah.turku.fi/tksjlk/2017/0412005x/Images/1522204.pdf>

- Föli (28.2.2020). "Turun kaupunkiseudun joukkoliikennelautakunnan käsittelyyn tulevia asioita". Föli:n Uutiset, Päätöksenteko, Uutiset. Haettu 28.5.2021 osoitteesta <https://www.foli.fi/fi/turun-kaupunkiseudun-joukkoliikennelautakunnan-k%C3%A4sittelyyn-tulevia-asioita-13>
- Gooch, James (28.10.2019). "Mobile Ticketing: Why Barcode?" Masabi Blog. Haettu 28.5.2021 osoitteesta <https://www.masabi.com/2019/10/28/mobile-ticketing-why-barcode/>
- Gooch, James (14.08.2020). "Everything You Need to Know About Contactless Ticketing for Public Transport (cEMV)". Masabi Blog. Haettu 28.5.2021 osoitteesta <https://blog.masabi.com/blog/everything-you-need-to-know-about-contactless-ticketing-for-public-transport-cemv>
- Google (2021a). "Host-based card emulation overview". Android developer documentation. Haettu 5.4.2021 osoitteesta <https://developer.android.com/guide/topics/connectivity/nfc/hce>
- Google (2021b). "What Is Pub/Sub?". Google Cloud Documentation. Cloud Pub/Sub Guides. Haettu 12.5.2021 osoitteesta <https://cloud.google.com/pubsub/docs/overview>
- Google (2021c). "Subscriber overview". Google Cloud Documentation. Cloud Pub/Sub Guides. Haettu 12.5.2021 osoitteesta <https://cloud.google.com/pubsub/docs/subscriber>
- Google (2011d). "Protocol Buffers - Language Guide". Google developers. Protocol Buffers Guides. Haettu 28.5.2021 osoitteesta <https://developers.google.com/protocol-buffers/docs/overview>
- Google (2021e). "Choosing between Native mode and Datastore mode". Google Cloud Documentation. Datastore Guides. Haettu 28.5.2021 osoitteesta <https://cloud.google.com/datastore/docs/firestore-or-datastore>
- Google (2021f). "Datastore Overview". Google Cloud Documentation. Datastore Guides. Haettu 28.5.2021 osoitteesta <https://cloud.google.com/datastore/docs/concepts/overview>
- GoToTags (03.06.2019). "Apple Expands NFC on iPhone in iOS 13". GoToTags Blog. Haettu 9.4.2021 osoitteesta <https://gototags.com/blog/apple-expands-nfc-on-iphone-in-ios-13>
- HSL (26.03.2019a). "HSL:n tilinpäätös 2018". HSL:n Uutiset. Haettu 28.5.2021 osoitteesta <https://www.hsl.fi/hsl/uutiset/uutinen/2019/03/hsln-tilinpaatos-2018>
- HSL (25.02.2019b). "Tiesitkö, että väärennetyllä lipulla matkustaminen on petos". HSL:n Uutiset. Haettu 28.5.2021 osoitteesta <https://www.hsl.fi/hsl/uutiset/uutinen/2019/02/tiesitko-etta-vaarennetylla-lipulla-matkustaminen-on-petos>
- HSL (2021a). "HSL:n joukkoliikenteen yleiset matkustus- ja lippuehdot". Haettu 16.5.2021 osoitteesta <https://www.hsl.fi/hsl/kayttoehdot-uudet/hsln-joukkoliikenteen-yleiset-matkustus--ja-lippuehdot>
- HSL (07.04.2021b). "HSL ja TVV lippu- ja maksujärjestelmä Oy valitsivat Futuricen uuden lippujärjestelmän toimittajaksi". HSL:n Uutiset. Haettu 16.5.2021 osoitteesta <https://www.hsl.fi/hsl/uutiset/uutinen/2021/04/hsl-ja-tvv-lippu--ja-maksujarjestelma-oy-valitsivat-futuricen-uuden-lippujarjestelman-toimittajaksi>
- Init (2/2016). "Turku proves forerunner mentality - Finland's oldest but most modern metropolitan area went into operation with revolutionary ticketing system". INIT's customer magazine INITiative, issue 2/2016. Haettu 28.5.2021 osoitteesta <https://www.initse.com/ende/news-resources/knowledge-database/articles/2016/initiative02-turku/>

- Init (2/2018). "ID-/Account-based Ticketing - New options of providing higher service quality". Init Whitepaper. Haettu 28.5.2021 osoitteesta https://www.initse.com/fileadmin/user_upload/Content/3_Solutions/2_Ticketing/INIT_whitepaper_ID_account_based_ticketing_en.pdf
- Kanerva, Olli - Kataja, Antti - Airaksinen, Simo - Aalto, Pekka (19.2.2020). "Taustaselvitys tunnistepohjaisten maksujärjestelmien yhteentoimivuudesta". Traficom julkaisu 9/2020. ISBN 978-952-311-476-0, ISSN 2669-8757. Haettu 16.3.2021 osoitteesta https://liikkumisenrajapinnat.fi/sites/default/files/media/publication/Taustaselvitys_tunnistepohjaisten_maksujarjestelmien_yhteentoimivuudesta.pdf
- Kotakorpi, Elli (27.01.2020). "Digitalization of Public Transport in Tampere". Esitelmä. Haettu 16.5.2021 osoitteesta https://www.tampere.fi/tiedostot/s/0EXTdhVCd/Elli_Kotakorpi_Digitalization_of_public_transport_in_Tampere.pdf
- Liikenne- ja viestintäministeriö (26.04.2018). "Koko liikennejärjestelmä mukaan liikennepalvelulakiin". Liikenne- ja viestintäministeriön tiedote. Haettu 1.12.2020 osoitteesta <https://www.lvm.fi/-/koko-liikennejarjestelma-mukaan-liikennepalvelulakiin-970989>
- Liikenne- ja viestintävirasto Traficom (2/2018). "Julkisen liikenteen suoritetilasto 2018". Traficom julkaisu. Liikenne ja matkailu. ISBN 978-952-311-718-1, ISSN 2669-9087. Haettu 28.5.2021 osoitteesta https://www.traficom.fi/sites/default/files/media/file/julkisen_liikenteen_suoritetilasto_2018.pdf
- Liikennevirasto (1/2018). "Henkilöliikenteen palveluiden sanasto". Liikenneviraston oppaita 1/2018. ISSN-L 1798-6591, ISSN 1798-6605, ISBN 978-952-317-556-3. Haettu 6.1.2021 osoitteesta https://julkaisut.vayla.fi/pdf8/opas_2018-01_henkiloliikenteen_palveluiden_web.pdf
- Mozilla (2021). "Compression in HTTP". MDN Web Docs. Haettu 6.1.2021 osoitteesta <https://developer.mozilla.org/en-US/docs/Web/HTTP/Compression>
- NPX Semiconductors (10.08.2010). "System level security measures for MIFARE installations". MIFARE DESFire Documentation. AN10969. Haettu 5.4.2021 osoitteesta <https://www.nxp.com/docs/en/application-note/AN10969.pdf>
- Nysse (2020). "Nysse vuosikertomus 2019". Nysse julkaisu, vuosikertomukset. Haettu 16.5.2021 osoitteesta https://www.nysse.fi/media/julkaisut/vuosikertomukset/vuosikertomus_2019_tampereenseudunjoukkoliikenne.pdf
- OSPT Alliance (18.2.2016). "HCE Synergies with Public Transport - CIPURSE™ and HCE open up new possibilities for Transit Ticketing Systems". OSPT Alliance Whitepaper. Haettu 25.4.2021 osoitteesta https://www.osptalliance.org/assets/1/6/OSPT_WHITEPAPER_HCE_FINAL.pdf
- OSPT Alliance (ei pvm). "Implementing Account-Based Ticketing (ABT) - How open, non-proprietary standards can bridge the transport ticketing innovation gap". OSPT Alliance Documents & Whitepapers. Haettu 28.5.2021 osoitteesta https://www.osptalliance.org/assets/1/6/OSPT_ABT_eBook.pdf?258
- Pesonen, Eeva-Stiina (30.11.2017). "Whim on käynnistynyt pelottavan hyvin". Kauppalehti Uutiset. Haettu 2.5.2021 osoitteesta <https://www.kauppalehti.fi/uutiset/whim-on-kaynnistynyt-pelottavan-hyvin/a355d292-44b7-3b15-a833-2d237ca6d16d>

Salminen, Reeta (2.5.2018). "Mikä ihmeen Whim? – Uusi palvelu yhdistää bussit, taksit, kaupunkipyörät ja vuokra-autot yhden klikkauksen alle, mutta väheneekö yksityisautoilu?". Yle Uutiset. Haettu 2.5.2018 osoitteesta <https://yle.fi/uutiset/3-10171507>

Samtrafiken (2018). "BoB Token API - ID based travelling". BoB - National Ticket & Payment Standards for Public Transport - BoB Manual. Haettu 22.5.2021 osoitteesta <https://samtrafiken.atlassian.net/wiki/spaces/BOB/pages/708214823/BoB%2BToken%2BAPI%2B-%2BID%2Bbased%2Btravelling>

Samtrafiken (2016 - 2021). "BoB - National Ticket & Payment Standards for Public Transport". Samtrafiken BoB-standardin määrittelyt. Saatavilla sähköisesti <https://samtrafiken.atlassian.net/wiki/spaces/BOB/overview>. Haettu 16.3.2021.

Smart Card Alliance (11/2021). "Transit and Contactless Open Payments: An Emerging Approach for Fare Collection". A Smart Card Alliance Transportation Council White Paper. TC-11002. Haettu 28.5.2021 osoitteesta https://www.securetechalliance.org/resources/pdf/Open_Payments_WP_110811.pdf

Streeting, Mark & Howe, Douglas (2016). "Contactless Payments and Open-Loop Ticketing - What does this mean for transit authorities, and is there a business case for adoption?". L.E.K. and Mastercard Special report. Haettu 26.4.2021 osoitteesta https://www.lek.com/sites/default/files/insights/pdf-attachments/LEK_Mastercard_Contactless_Payments_Mark_Streeting.pdf

TVV lippu- ja maksujärjestelmä Oy (ei pvm). Waltin verkkosivut. Ohjeet matkustajalle. Haettu 16.5.2021 osoitteesta <https://waltti.fi/waltti-mutkaton-matkakortti>

UK Finance (12/2017). "Contactless Transit EMV Framework". UK Finance's reports and publications. Haettu 28.5.2021 osoitteesta <https://www.ukfinance.org.uk/system/files/UK-Finance-Contactless-Transit-EMV-Framework.pdf>

Valmari, Nelli (30.8.2019). "Bussien kaukoliikenteen avautuminen kilpailulle". ETLA Muistio No 82. <https://pub.etla.fi/ETLA-Muistio-Brief-82.pdf>

Whitaker, Ben (13.04.2008). "Ideas for Interoperability of Secure Barcode Tickets". Masabi Blog. Haettu 28.5.2021 osoitteesta <https://blog.masabi.com/blog/2008/04/13/ideas-for-interoperability-of-secure-barcode-tickets/>

Whitaker, Ben (22.06.2016). "The Future of Transport Ticketing: Bring Your Own Ticket (BYOT)". Masabi. [Youtube-video]. Haettu 20.4.2021 osoitteesta <https://www.youtube.com/watch?v=KESDJOjLLRw>

Zamer, Khaled (2018). "Account Based Ticketing: The Benefits and Drivers for Transit Operators". Journal of Transportation Technologies. 08. 331-342. 10.4236/jtts.2018.84018

Liite 1 - Pusatec PT10 eX ja PT11 eX -ajoneuvolaitteiden tiedot

PT10 eX	PT11 eX
<p>Ohjelma- ja datamuisti 4 Gt, haihtumaton Flash Dynaamista DDR3-muistia 1 Gt Ohjelma- ja datamuistin laajennus 16 – 256 Gt haihtumaton Flash Grafiikkakiihdytin Tuplaydinprosessori Linux-käyttöjärjestelmä Moniajo 5,7” kosketusnäyttö Graafinen näyttö asiakkaalle Näppäimistö taustavalolla Matkustajaliittymä omana moduulina laitteeseen sisäänrakennettuna Älykortinlukija, kontaktillinen Älykortinlukija, kontaktiton Nopea lämpökirjoitin max 200 mm/s Paikannus GPS 3D-magneetti- ja 3D-kiihtyvyyssanturi Yhteydet taustaan 2G ja 3G (optio 4G) Integroitu 2D-lukija/QR-koodi (optio) Liitynnät auton muihin järjestelmiin (standardi CAN open) 4 kpl turvamoduulipaikkoja</p> <p>Kattava määrä liitäntöjä ulkopuolisiin lisälaitteisiin: 4 kpl USB 2.0 -portteja HDMI-liitäntä CAN Ethernet</p>	<p>Ohjelma- ja datamuisti 4GB, haihtumaton Flash Dynaamista DDR3 muistia 1GB Ohjelma- ja datamuistin laajennus 16 – 256GB haihtumaton Flash Grafiikkakiihdytin Tuplaydinprosessori Linux-käyttöjärjestelmä Moniajo 5,7” graafinen kosketusnäyttö Älykortinlukija, kontaktiton Paikannus GPS (optio) Yhteydet taustaan 2G ja 3G (optio 4G) Integroitu 2Dlukija/QR -koodi (optio) Liitynnät auton muihin järjestelmiin (standardi CANopen) 1 kpl turvamoduulipaikka</p>

Lähde: <https://www.pusatec.fi>

