

Hubject EMP -integraatio

Eemeli Järvinen

OPINNÄYTETYÖ
Toukokuu 2022

Tietojenkäsittely
Web-palvelut

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely
Web-palvelut

EEMELI JÄRVINEN
Hsubject EMP -integraatio

Opinnäytetyö 30 sivua, joista liitteitä 0 sivua
Toukokuu 2022

Opinnäytetyön toimeksiantaja oli Plugit Finland Oy, joka on sähköautojen latauslaitteita ja latauspalveluita tarjoava yritys. Työn tarkoituksena oli toteuttaa Hsubject EMP -integraatio. Tavoitteena oli tutkia ohjelmisto- ja järjestelmäintegraatioiden toteutustapoja sekä sähköautojen latausinfraan liittyviä protokollia ja standardeja.

Työn tuloksena syntyi mahdollisuus lisätä Plugitin-sovellukseen Hsubject-palvelun kautta muiden latauspalveluita tarjoavien yritysten latauspisteitä. Hsubject on eroaming-alustaa palvelunaan tarjoava saksalainen yritys. Se on luonut Open intercharge -protokollan, joka on eRoaming-käytäntö, millä viestitään palveluiden välillä.

Integraatioista tutkittiin niihin liittyviä vaatimuksia, hyötyjä sekä haasteita. Integraation ratkaisumalleista tutustuttiin kolmeen eri vaihtoehtoon. Jokaisesta löydettiin vahvuuksia ja heikkouksia, esimerkiksi point-to-point-ratkaisumalli on toimiva ratkaisu, jos integroitavia järjestelmiä on vain kaksi, mutta muuttuu vaikeasti hallittavaksi, kun järjestelmien määrä kasvaa.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Business Information Systems
Web Services

JÄRVINEN, EEMELI:
Hsubject Emp Integration

Bachelor's thesis 30 pages, appendices 0 pages
MAY 2022

This Bachelor's thesis was commissioned by Plugit Finland Oy. The purpose of the thesis was to implement Hsubject emp integration. The main objective of the thesis was to research information about integrations, protocols and standards related to electric vehicle charging infrastructures.

The result of the thesis was the possibility to add charging points to Plugit application through Hsubject's service. Hsubject is a German company which provides eRoaming platform. They have has created and maintains an Open interchange protocol used to communicate between services.

The requirements, benefits and challenges of integrations were studied. Three of the integration technologies were studied in further detail. All of them have advantages and disadvantages for example point-to-point technology is easy to create when there are only few systems but when the number of systems grows it becomes more difficult to maintain.

Key words: integration, roaming, oicp

SISÄLLYS

1	JOHDANTO	6
2	Integraatio	7
	2.1 Mikä on integraatio?	7
	2.2 Järjestelmäintegraatio	7
	2.3 Business to business integraatio	7
	2.4 Integraation hyödyt ja tarkoitus	8
	2.5 Point-to-point	8
	2.6 Hub-and-spoke	9
	2.7 Enterprise Service Bus	10
3	Sähköautoilu	12
	3.1 Sähköautot	12
	3.2 eRoaming	12
	3.3 Protokollat	13
4	Hubject	15
	4.1 Kuvaus	15
	4.2 CPO	15
	4.3 EMP	15
	4.4 Open InterCharge Protocol	16
5	Käytetyt tekniikat	19
	5.1 Angular	19
	5.2 Akita	19
	5.3 Typescript	20
	5.4 Kubernetes ja Docker	20
	5.5 CI/CD ja GitLab	20
	5.6 MongoDB	21
6	Integraatio toteutus	22
	6.1 Projektin kuvaus	22
	6.2 Suunnittelu	22
	6.3 Toteutus	23
	6.4 Testaus	26
	6.5 Lopputuloksia	26
7	POHDINTA	30
	LÄHTEET	31

ERITYISSANASTO

API	Application programming interface eli ohjelmointirajapinta.
http	Protokolla, mitä selaimet ja www-palvelimet käyttävät tiedonsiirtoon.
npm	Node.js mukana tuleva pakettien hallintaan tarkoitettu ohjelma.
ohjelmistokehys	Ohjelmisto, joka muodostaa rungon sen päälle rakennettavalle tietokoneohjelmalle.
Paginaatio	Tiedon jakamista useammalle http-pyyntölle ja tiedon näyttämistä usealla sivulla.
rfid	Radio Frequency IDentification eli radiotaajuinen etätunnistus.
RxJs	Reactive Extensions for JavaScript, kirjasto reaktiivisen ohjelmointiin.

1 JOHDANTO

Opinnäytetyön tarkoituksena on toteuttaa Hsubject EMP -integraatio. Hsubject on saksalainen eRoaming alustaa palvelunaan tuottava yritys. Hsubject EMP on palvelu, jolla sähköajoneuvojen latauspalveluita tarjoava yritys voi ottaa käyttöönsä muiden yritysten latauspisteitä. Hsubject toimii niin sanotusti välikätenä ja tiedon standardisoijana yritysten välissä.

Opinnäytetyön toimeksiantaja on Plugit Finland Oy. Plugit on vuonna 2012 perustettu sähköautojen latausratkaisuja tarjoava yritys. Plugitilla on oman mobiili-sovellus, millä voi käyttää Plugitin latauspisteitä. Plugitilla on myös hallintatyökalu, mistä voidaan hallinnoida kaikkia latausverkolla olevia latauslaitteita.

Integraatiolla tarkoitetaan kahden tai useamman järjestelmän yhteen liittämistä ja tiedonsiirtoa niiden välillä. Integraatioita voidaan toteuttaa erilaisin ratkaisumallein. Ratkaisumallin oikea valinta perustuu järjestelmien kokoon ja niiden määrään. Integraatioita joudutaan yleensä toteuttamaan yrityksen kasvun, muuttuneiden tarpeiden tai vanhentuneiden järjestelmien takia.

2 Integraatio

2.1 Mikä on integraatio?

Integraation voi määritellä monella eri tavalla. Yleisimmin käsitteellä tarkoitetaan tapoja ja tekniikoita, joilla tuodaan yhteen pienemmät komponentit yhdeksi isoksi toimivaksi järjestelmäksi. Integraatio voi yksinkertaisimmillaan olla kahden komponentin välistä tiedonsiirtoa. Integraation avulla yritys voi tehostaa omaa toimintaansa ja lisätä sen joustavuutta. (Tähtinen 2005.)

2.2 Järjestelmäintegraatio

Järjestelmäintegraatiolla eli Enterprise Architecture Integration (EAI) tarkoitetaan yritysten sisäistä integraatiota, jossa tietoa siirretään järjestelmältä toiselle. Integraatiolla pyritään liittämään kaksi tai useampi yhteensopimaton järjestelmä toisiinsa ja luomaan automaattinen tiedonsiirto niiden välille. Tarve järjestelmäintegraatiolle tulee usein vanhasta järjestelmästä, johon uusien ominaisuuksien lisääminen tai niiden muokkaaminen olisi liian työlästä ja kallista. Järjestelmäintegraatiolla voidaan pyrkiä liittämään uusi järjestelmä vanhaan, jossa on edelleen tarvittavia ominaisuuksia ja tietoja. (Tähtinen 2005.)

2.3 Business to business integraatio

Business-to-business (B2B) on nimitys yritysten välisestä integraatiosta. Se on kahden tai useamman yrityksen välistä automaattista liiketoimintaprosessia ja kommunikointia. Sen avulla yritykset voivat työskennellä ja käydä kauppaa liikekumppaneidensa kanssa automatisoimalla keskeiset liiketoimintaprosessit. (Tähtinen 2005.)

2.4 Integraation hyödyt ja tarkoitus

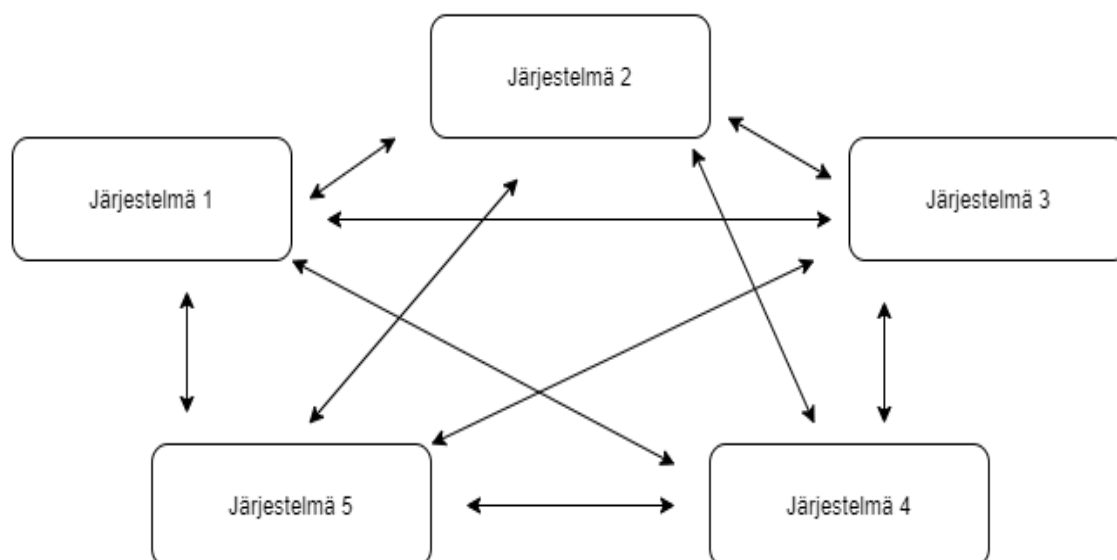
Yritykset kohtaavat uusia ja merkittäviä haasteita liiketoiminnan kasvun, asiakkaiden nopean tiedonhaun odotusten ja palvelujen toimintamallien muutosten vuoksi. Ala on siirtynyt tuotelähtöisestä mallista asiakaslähtöiseen liiketoimintamalliin. Informaatiotekniikka toimii mahdollistajana, jonka avulla yritys voi organisoida uudelleen toimintaansa ja ajatella uudelleen palvelun toimintamekanismit. Muutokset ovat välttämättömiä, jotta yritys voi toimia paremmin, vähentää kuluja ja muuttua asiakaslähtöisemmäksi. (Manouvrier & Ménard 2008.)

Integraatiota voidaan toteuttaa erilaisin ratkaisumallein. Näitä ratkaisumalleja ovat esimerkiksi Point-to-point, hub-and-spoke sekä Enterprise Service Bus. Kaikista ratkaisumalleista löytyy vahvuuksia ja heikkouksia.

2.5 Point-to-point

Point-to-point integraation ratkaisumalli on kaikista yksinkertaisin toteuttaa. Siinä nimensä mukaisesti viedään informaatio järjestelmästä järjestelmään. Point-to-point-ratkaisumallia on aiemmin käytetty paljon, kun järjestelmiä on rakennettu vain järjestelmän näkökulmasta eikä suuremman kokonaisuuden kannalta. (Manouvrier & Ménard 2008.)

Point-to-point ratkaisumallin vahvuuksia on sen kehittämisen sekä julkaisemisen yksinkertaisuus. Ratkaisumalli on täysin mukautuva, koska se luodaan vain tarpeeseen. Mahdollisuus hallita monimutkaisia informaationvälitystyypppejä sekä sääntöjä on helppoa, sillä ne on keskitetty integraatoratkaisun ytimeen. Ratkaisumalli on edullinen kehitettävä ja nopea ratkaisu vaadittuun tarpeeseen. Ratkaisumalli soveltuu hyvin suppeampien järjestelmien yhdistämiseen, jos ennalta tiedetään, että järjestelmiä ei ole tulossa enempää. (Manouvrier & Ménard 2008.)

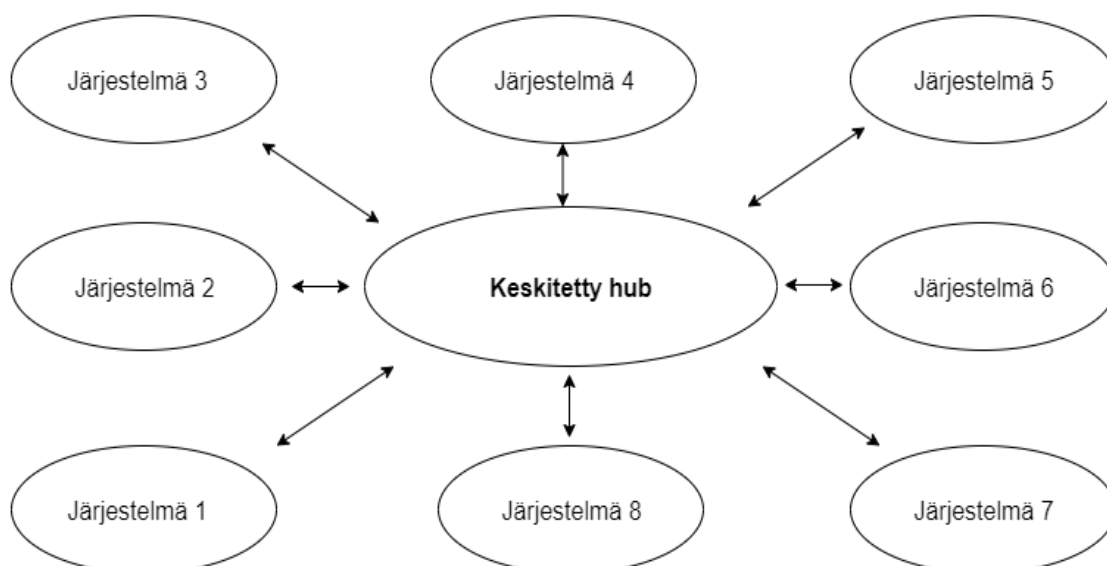


KUVIO 1. Point-to-point-ratkaisumallin järjestelmien välinen yhteyskaavio viidellä järjestelmällä toteutettuna.

Järjestelmien määrän kasvaessa, point-to-point-ratkaisumallin hallittavuus heikenee ja ylläpitäminen muuttuu kalliiksi. Yhden järjestelmän päivittyessä täytyy selvittää mahdolliset muutokset, joita päivitys voi aiheuttaa järjestelmään linkitettyihin muihin järjestelmiin. Monen eri järjestelmän kattava kokonaisuus muuttuu nopeasti vaikeasti hahmotettavaksi spagettijärjestelmäksi, jossa linkkejä järjestelmien välillä on todella paljon. On myös mahdollista, että joidenkin järjestelmien välille syntyy ruuhkautumista tiedon siirrossa, jos jotkin järjestelmät ovat suuremmissa käytössä. (Manouvrier & Ménard 2008.)

2.6 Hub-and-spoke

Hub-and-spoke-ratkaisumallissa integraatio on keskitettyä. Point-to-point-ratkaisumallin verrattaessa, missä järjestelmät ovat linkitettyjä toisiinsa, hub-and-spoke-ratkaisussa järjestelmät ovat linkitettyinä ainoastaan yhteen keskitettyyn hubiin. Hubilla tarkoitetaan integraatoratkaisun keskitettyä osaa, minkä lävitse liikkuva tieto kulkee toisille järjestelmille. (Manouvrier & Ménard 2008.)



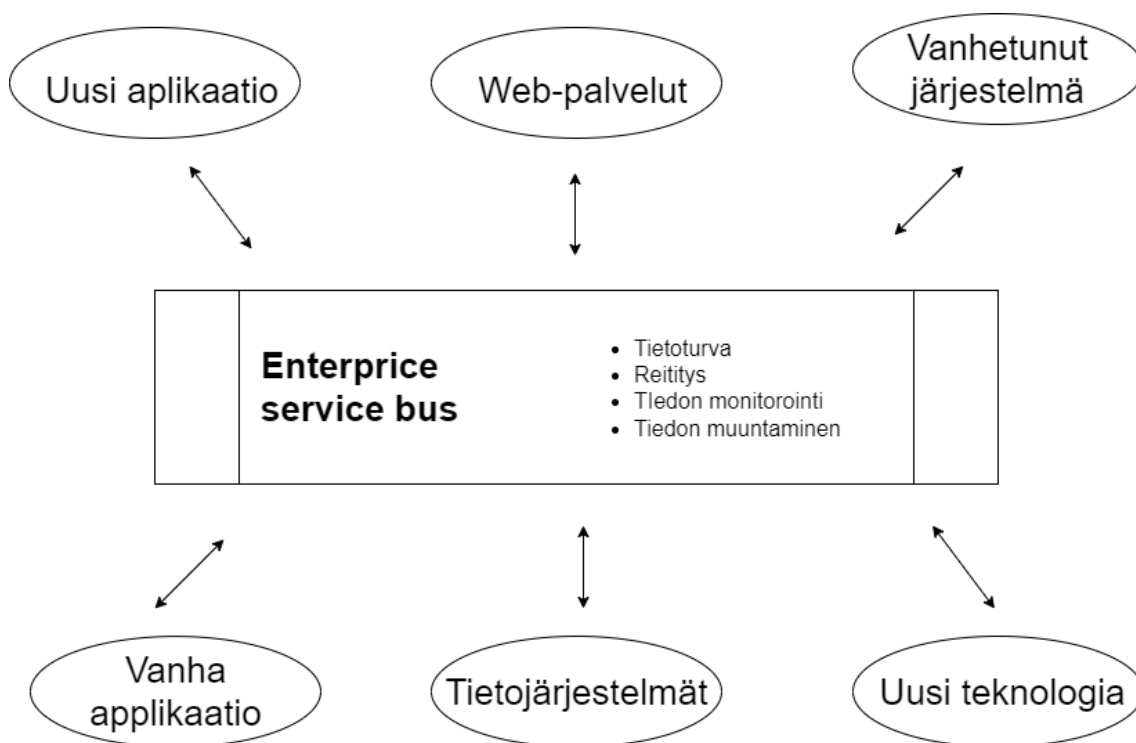
KUVIO 2. Hub-and-spoke-integraatioratkaisumalli toteutettuna kahdeksalla järjestelmällä.

Ratkaisumallin käyttö on yksinkertaista ja skaalautumien on mahdollista ja helppoa pidemmän aikaa. Keskitetyssä ratkaisussa päivittäminen onnistuu vaivattomasti, kun uuden järjestelmän luodessa sen voi liittää hubiin, miettimättä muiden järjestelmien linkkejä. Kun tieto kulkee aina keskitetysti samaa reittiä, on sen monitorointi ja kontrollointi helpompaa. (Manouvrier & Ménard 2008.)

Kun kaikki järjestelmät ovat yhden keskitetyn ratkaisun varassa, sen kaatuminen tai vioittuminen voi tarkoittaa liiketoimintaprosessien pysähtymistä. Hub-and-spoke-ratkaisumallin käyttöönotto on alussa hitaampaa ja haastavampaa kuin point-to-point-ratkaisumallissa. (Manouvrier & Ménard 2008.)

2.7 Enterprise Service Bus

Enterprise Service Bus (ESB) on ratkaisumalli, jota käytetään palvelukeskeisessä ohjelmistoarkkitehtuurissa (SOA). ESB- ja hub-and-spoke-ratkaisumalleissa on samoja piirteitä ja ominaisuuksia, mutta ESB-ratkaisumallia pidetään joustavampana järjestelmien lisäämisen kannalta. (Schorow 2008.)



KUVIO 3. ESB-integraatioratkaisumalli erilaisilla järjestelmillä.

ESB on kokoelma erilaisia teknologioita, joiden avulla SOA-palveluarkkitehtuurin viestinvälitys toteutetaan. ESB on yksinkertaisimmillaan kokonaisuus, jonka avulla eri teknologioilla rakennetut järjestelmät välittävät ohjelmisto- ja laitteisto-riippumattomia viestejä. Hub-and-spoke malliin verrattaessa, missä kaikki data on keskitetty hubiin, voidaan ESB-mallissa integraation tietoturva, reititys ja tietomuunnokset jakaa eri palveluihin. Näin kaikki toiminto ei ole keskitetty vain yhteen palveluun, vaan moniin palveluihin, jotka keskustelevat toistensa kanssa. Jaetun datan ansiosta ruuhkautumisia datan liikkeessä ei pysty syntymään niin helposti. (Schorow 2008.)

3 Sähköautoilu

3.1 Sähköautot

Viimevuosien aikana sähköajoneuvojen määrä on kasvanut todella paljon. Suomessa Sähköinen liikenne ry:n tilannekatsauksen (Q4/2021) mukaan sähköautoja on rekisteröity 99 910 vuonna 2021. Luvusta täyssähköisiä autoja on 22 921. Vuodessa sähköautokanta on kasvanut 81 prosenttia. Kaikista vuoden 2021 rekisteröityjen uusien ajoneuvojen määrästä 24 prosenttia on täyssähköautoja. (Sähköinen liikenne ry 2021.)

Tavoitteena on, että Suomessa olisi vuoteen 2030 mennessä vähintään yksi sähköauton pikalatausasema jokaista täyssähköautoa kohden. Suomen autokannan osalta tavoitteena on vuoteen 2030 mennessä, että rekisterissä olisi 700 000 täyssähköautoa ja 45 000 täyssähköistä pakettiautoa. Raskaassa liikenteessä tavoitteet ovat noin 4600 täyssähköautoa. (Fossiilittoman liikenteen tiekartta 2021.)

3.2 eRoaming

Roaming-termi tarkoittaa televiestinnässä verkkovierailua eli teleoperaattorin palveluita käytetään jonkun toisen operaattorin verkossa. eRoamingilla sähköajoneuvojen latauksista puhuttaessa tarkoitetaan suunnilleen samaa asiaa. Latauspalveluita tarjoavan yrityksen rakentamasta palvelusta voi ladata toisen palveluntarjoajan latausasemalla tai -pisteellä. Kyseessä voi olla yritys X:n lataussovellus, jossa näkyy latauspalveluita tarjoavan yritys Y:n latauspisteet ja latauksen voi aloittaa yritys X:n tarjoamasta sovelluksesta. eRoaming mahdollistaa yritykselle tavan pitää asiakkaita itsellään, koska latauspaikkojen valikoima kasvaa huomattavasti. (Ferwerda 2018.)

Sähköajoneuvot nähdään yhtenä tärkeimmistä teknologioista ympäristöystävällisen ja kestäväen liikenteen parantamiseen. Kuitenkin protokollien ja standardien puuttuminen EU:ssa vaikuttaa suuresti maan rajojen ylittävään matkailuun. Maiden yhtenäisten protokollien puuttuminen on este sähköautoilun kasvamiseen

Euroopassa. Muita syitä sähköautoilun kasvamiselle tai hyväksymiselle, ajomatkan pituuden miettimistä lukuun ottamatta, on huono informaatio latauspaikkojen sijainnista sekä saatavuudesta, latauksen hinnan selkeästä näyttämisestä ja latausmahdollisuutta tarjoavien määrästä, jolloin lataamiseen tarvitaan erilaisia sovelluksia ja lataustunnisteita. (Ferwerda 2018.)

3.3 Protokollat

Viestintäprotokolla määrittää kahden rajapinnan yhteyksissä olevan rajapinnan välisen tiedon tyyppin. Nämä protokollat ja yhteensopivuusstandardit varmistavat kahden eri järjestelmän yhteensopivuuden. Protokollien ja standardien ansiosta uusien yhteyksien lisääminen on helppoa, koska vaatimukset ovat tiedossa jo suunnitteluvaiheessa.

Open Charge Point Protocol (OCPP) on protokolla tiedonvälitykseen sähköisten ajoneuvojen ja latauspisteiden välillä. Kun käytössä on protokolla, jota laitteidenvalmistajat noudattavat, ei latauspisteen tarjoaja ole vain yhden laitteidenvalmistajan varassa. Uusien latauslaitteiden käyttöönotto on helppoa, kun kaikki toimii saman käyttöjärjestelmän avulla.

Toiminto	Viesti
Tunnistautuminen laitteelle	Authorize
Laitteen asetusten vaihtaminen	ChangeConfiguration
Yhteys tallella	HeartBeat
Sähkökulutuksen mittariarvoja	MeterValues
Latauksen aloitus	StartTransaction
Latauksen lopetus	StopTransaction

Taulukko 1. OCPP-protokollan viestiesimerkkejä.

OCPP protokolla määrittää kaiken latauslaitteiden viestinnässä. OCPP uusin julkaisu on 2.0.1 ja se on julkaistu huhtikuussa 2018. OCPP protokollan dokumentaatio pitää sisällään lähes 200 erilaista viestiä. (OCPP-dokumentaatio.)

Open Charge Point Interface (OCPI) on EVRoaming-säätiön luoma ja ylläpitämä protokolla eRoamingiin. Ero tässä opinnäytetyössä esiteltävään OICP-protokollaan on sen palveluntarjoajassa sekä tietenkin protokollat ovat sisällöltään erilaiset. OCPI-protokolla on EVRoaming-säätiön luoma ja sen käyttöönotto on ilmaista, toisin kuin OICP, joka on Hubjectin luoma ja maksullinen. (EVRoaming-säätiö kotisivut.)

EVRoaming säätiön tavoitteena on tarjota eRoaming-palveluita sähköajoneuvojen lataamiseen ja tarjota kuluttajille läpinäkyvää tietoa latauspaikoista ja hinnoista. EVRoaming nimeää tärkeimmäksi tavoitteekseen, että jokainen sähköauton kuljettaja voi ladata millä tahansa latausasemalla EU:ssa. (EVRoaming-säätiö kotisivut.)

4 Hubject

4.1 Kuvaus

Hubject on vuonna 2012 perustettu yritys, jonka tarkoituksena on helpottaa sähköisten ajoneuvojen latausta. Hubject tarjoaa palvelussaan eRoaming-alustan. Se yhdistää latauspalveluita tarjoavan yrityksen ja yrityksen, joka haluaa kyseisen yrityksen latauspalvelut käyttöönsä omassa järjestelmässä. OICP-standardin mukainen tieto kulkee Hubjectin alustan läpi, mikä tarjoaa pääsyn latausinfrastruktuuriin järjestelmästä riippumatta. (Hubject-kotisivu.)

Hubjectin käyttäminen ja käyttöönotto perustuu yritysten välisiin sopimuksiin. Sopimukset on helppo sopia Hubjectin portaalisivustolla. Sinne latauspalveluita tarjoavat yrityksen voivat laittaa kaikille avoimia tarjouksia omista latauspisteistään. Näihin muut Hubjectia käyttävät yritykset voivat niin sanotusti tehdä tilauksen, jolloin latauspisteet tulevat niille käyttöön nopeasti. (Hubject-kotisivu.)

4.2 CPO

Hubject palvelun voi ottaa käyttöön kahdella eri tavalla, CPOna tai EMPna. CPO on lyhenne sanoista Charge Point Operator, suomennettuna se tarkoittaa latauspisteen operaattoria. Operaattorin asemassa hubjectin ja eRoamingin käyttöönotosta hyötyy yleisesti asiakkaiden määrän kasvulla. (Hubject-kotisivu.)

4.3 EMP

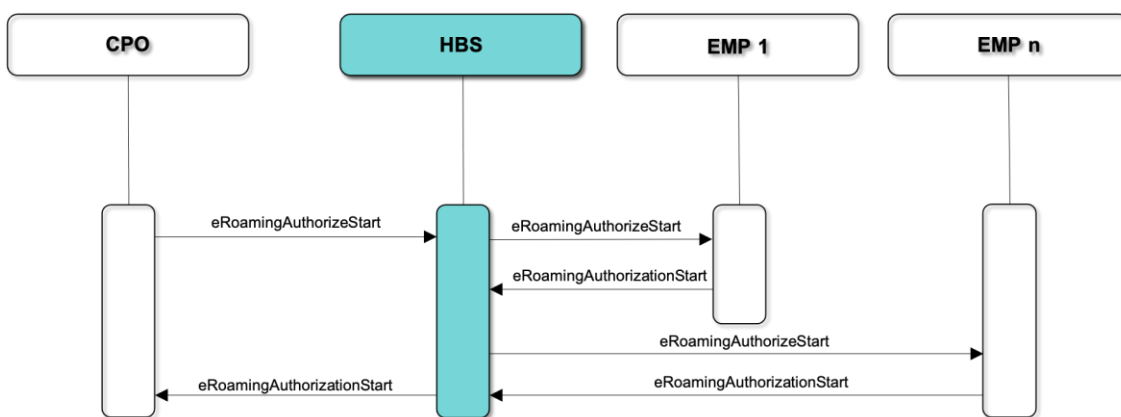
EMP tulee sanoista eMobility Service Provider, suomennettuna se tarkoittaa suurin piirtein sähköisen liikkumisen palvelun tarjoajaa. Hubjectin kautta palveluntarjoaja voi ottaa käyttöönsä omassa palvelussaan, esimerkiksi sovelluksessa muiden operaattoreiden latauspisteitä. CPO ja EMP ovat erilliset palvelut, joita Hubject tarjoaa ja on mahdollista ottaa käyttöön molemmat tai vain toinen palvelu. (Hubject-kotisivu.)

4.4 Open InterCharge Protocol

Hubject käyttää OICP-protokollaa viestinnässään. Se on Hubjectin vuonna 2012 luoma viestintäprotokolla ja se on käytössä yli 700 yrityksellä yli 40 maassa.

OICP määrittää kaikki viestityypit Hubjectin kanssa käydyssä viestinnässä. (Hubject-kotisivu.)

Kaikki viestintä Hubjectin kanssa kulkee sen rest api -rajapintojen kautta. Kun käyttäjä esimerkiksi aloittaa latauksen CPO:n tarjoamalta pisteeltä, lähtee http-pyyntö Hubjectille. Saatuaan pyynnön Hubject kysyy EMP-palvelun tarjoajilta, että kenelle kyseinen käyttäjä kuuluu. EMP-palvelun tarjoaja vastaa Hubjectin kautta CPOlle, että latauksen saa aloittaa. (OICP-protokolla dokumentaatio.)

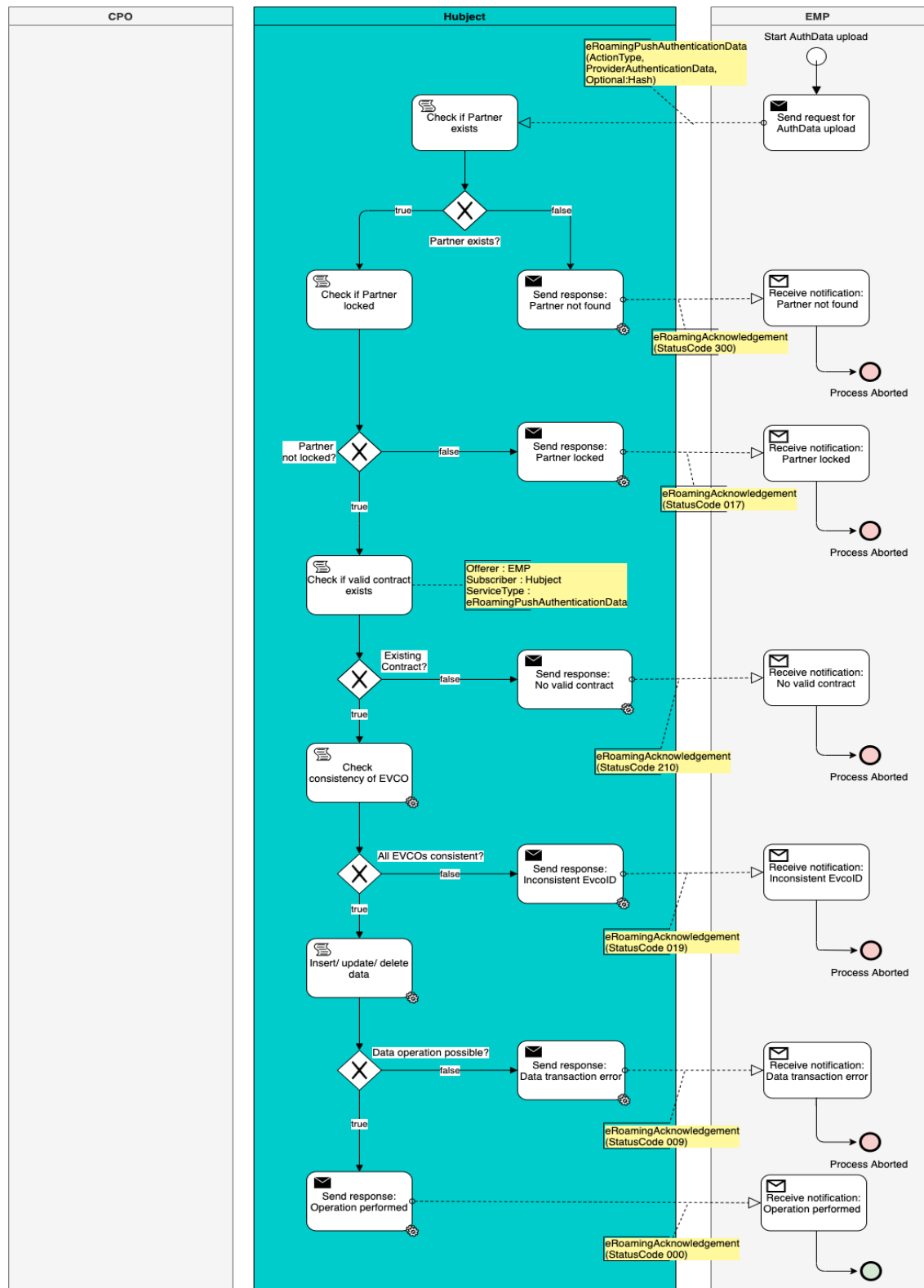


Kuva 1. Latauksen aloitus -viestin tiedonkulku. (OICP-protokolla dokumentaatio)

Latauksen voi aloittaa kahdella eri OICP-protokollan käskyllä. Latauksen aloittaminen latauslaitteella esimerkiksi rfid-tunnisteella, käytetään protokollan eRoamingAuthorizeStart-käskyä. Kun käyttäjä yrittää aloittaa latauksen latauspisteeltä, jolle käyttäjän rfid-tunniste ei kuulu ja latauksen aloittaminen epäonnistuu, CPO lähettää pyynnön Hubjectille. Hubject käsittelee pyynnön ja luo tietokantaansa ja EMP-palvelulle tarvittavat tiedot. (OICP-protokolla dokumentaatio.)

Oikean EMP -palvelun löytämiseen ja käyttäjän auktorisoimiseen Hubjectilla on käytössä kolme eri toimintatapaa. Yhteydettömässä tavassa EMP palvelun tarjoaja on lähettänyt tarvittavat tietonsa Hubjectille, mistä voidaan tarkistaa, että kuka on latauksen aloittanut. Toinen tapa, jolla Hubject voi yhdistää käyttäjän on

latauksen aloituskäskyn mukana tulleen tiedon perusteella tehtävä kysely yhdelle EMP palvelulle. Jos edellisen pyynnön vastaus on kielteinen, Hubeject lähettää kaikille CPO:n kanssa sopimuksessa olevalla EMPIlle saman viestin ja odottaa joltain näistä hyväksytyä vastausta. (OICP- protokolla dokumentaatio.)



Kuva 2. Käyttäjän tunnistaminen. (OICP- protokolla dokumentaatio.)

Kun lataus aloitetaan "etänä" esimerkiksi puhelimen sovelluksella, käytettävä käsky on eRoamingAuthorizeRemoteStart. Käskyn saatuaan Hsubject tarkistaa ensimmäisenä onko käskyn tulleeella EMPllä ja latauspisteen haltijalla CPOlla keskeinen sopimus. Jos sopimus löytyy, Hsubject välittää CPOlle käskyn, mikä sisältää tarvittavan tiedon latauksen aloittamiseen. (OICP- protokolla dokumentaatio.)

Kuten latauksen aloittamisessa, on sen lopettamisessakin kaksi eri tapaa: latauslaitteelta lopettaminen ja "etänä" lopettaminen. Lopettamiskäskyn mukana laiteaan aina latauksen aloituksessa luotu uniikki Session ID. Session IDn perusteella Hsubject osaa yhdistää lopettamiskäskyn oikeaan EMP tai CPO -palveluun. (OICP- protokolla dokumentaatio.)

CPO palvelun tarjoajan on luotava jokaisesta päättyneestä latauksesta Charge Detail Record (CDR). CDR lähetään Hsubjectille. Hsubject lähettää sen eteenpäin EMP -palvelulle ja samalla tallentaa sen omaan tietokantaansa, jos tiedon lähettämisessä tulee jotain ongelmia. CDR pitää sisällään kaikki lataukseen liittyvät tiedot kuten latauksen ajankohdan, keston ja latauksen energiakulutuksen. Se kysytään protokollan eRoamingChargeDetailRecord-käskyllä. Charge Detail Recordit voidaan myös pyytää Hsubjectilta erikseen. Tähän käskyyn täytyy aina lisätä päivämäärät siltä aikaväliltä, jolta lataukset halutaan. (OICP- protokolla dokumentaatio.)

Latauslaitteiden haku Hsubjectista toimii eRoamingPullEvseData-käskyn kautta. Käskyn saatuaan Hsubject tarkistaa onko EMP palvelulla tarvittava sopimus Hsubjectin kanssa, että tiedot voidaan luovuttaa. Käskyn mukana voidaan laittaa tieto, milloin viimeksi on latauslaitteet haettu. Tämän tiedon perusteella Hsubject osaa palauttaa vain viime käskyn jälkeen muuttuneet latauslaitteiden tiedot. Uusimman OICP-päivitykseen on tullut lisäys, että latauslaitteiden haussa on käytettävä paginaatiota. Latauslaitteiden tilatietoja, eli tietoja siitä onko latauslaite käytössä, vapaa tai vikatilassa haetaan eRoamingPullEvseStatus-käskyllä. (OICP- protokolla dokumentaatio.)

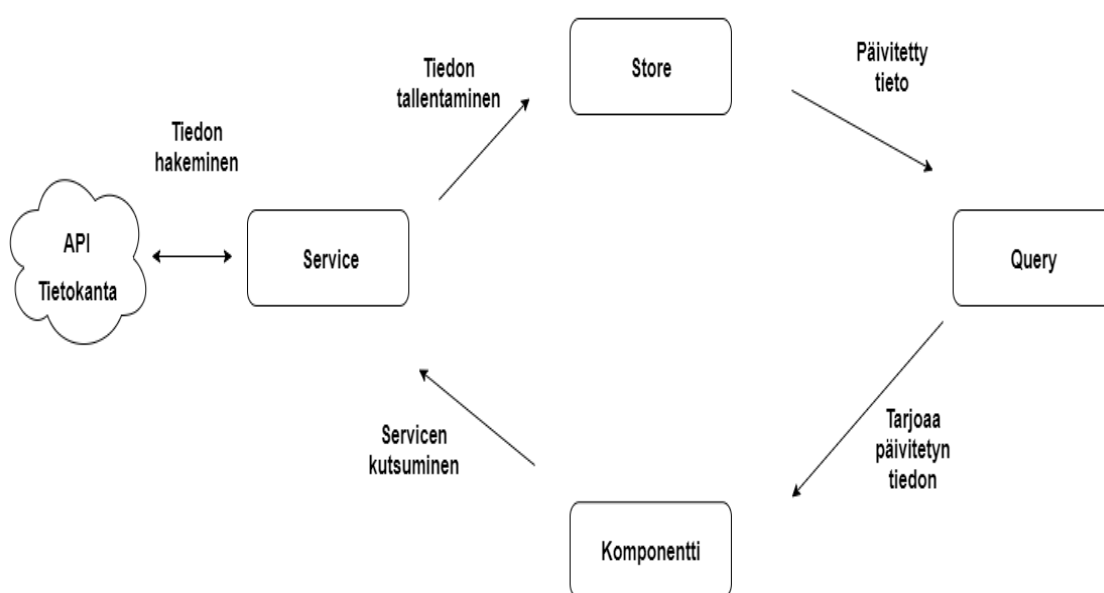
5 Käytetyt tekniikat

5.1 Angular

Angular on TypeScriptillä rakennettu ohjelmistokehys. Angularilla voidaan toteuttaa skaalautuvia verkkosivuja komponenttipohjaisesti. Komponentit ovat ”rakennuspalikoita”, jotka koostuvat TypeScript-luoksta, Hypertext Markup Language (Html) -templatesta ja tyylitiedostosta. Html-template määrittelee kuinka ja milloin komponentti renderöidään eli ajetaan. TypeScript luokan sisälle tulee komponentin logiikan hallinta. (Angular-dokumentaatio.)

5.2 Akita

Akita on Datoraman kehittämä verkkosovelluksen tilanhallintakirjasto. Tilan hallinnalla tarkoitetaan sovelluksissa ja verkkosivustoilla keskitettyä paikkaa, jonne tallennetaan ja josta haetaan tietoa sovelluksen tilasta. Akita koostuu kolmesta osasta: service, query ja store. Store määrittelee tallennuksen tietotyyppin. Servicen tehtävä on hoitaa tiedon hakeminen tietokannasta http-pyynnöillä. Queryn tehtävä on hoitaa tietojen hakeminen tallennuksesta. (Akita-dokumentaatio.)



KUVIO 4. Akita store logiikka ja tiedonkulkujärjestys.

5.3 Typescript

TypeScript on Microsoftin kehittämä vahvasti tyyplitetty ohjelmointikieli eli jokainen muuttuja voidaan tyyppittää. TypeScript käännetään JavaScriptiksi, joten kaikki JavaScript koodi on myös TypeScriptiä. Se suunniteltiin estämään ajoaikana tapahtuvia virheitä. TypeScript lisää ominaisuuksia JavaScripttiin, mutta nämä ominaisuudet ovat käytössä vain ohjelmoimissa, kuten rajapintaluokat, jotka ovat TypeScriptissä käytössä katoavat, kun koodi käännetään JavaScriptiksi. (Jansen 2016.)

5.4 Kubernetes ja Docker

Kubernetes on avoimen lähdekoodin ohjelmisto, mikä tarkoitettu esimerkiksi Docker klusterien eli konttien hallintaan. Kubernetesella voidaan toteuttaa aika-tilatutusta, työtaakan jakamista ja skaalautuvuutta. Kubernetesessa podi on pienin yksittäinen laskentaresurssi. Laskentaresurssit ovat mitattavissa olevia suureita, joita voidaan kutsua, allokoida ja kuluttaa. Podi on kokoelma yhdestä tai useammasta Docker containerista. Kubernetesista voidaan ajaa monilla pilvipalvelin ympäristöjen alustoilla kuten Google Kubernetes Enginellä, Amazon Elastic Kubernetes Servicellä, ja Azure Kubernetes Servicellä. Vaihtoehtoisesti Kubernetesista voidaan myös ajaa ja ylläpitää itse (self-hosted). (Vohra 2016.)

5.5 CI/CD ja GitLab

GitLab tarjoaa palvelunaan Git-versionhallinnan. Sen on kehittänyt ja ylläpitää GitLab Inc. GitLab on saatavilla pilvipalveluna (SaaS) tai itse ylläpidettynä. (GitLab-dokumentaatio.)

CI/CD tulee englannin kielen sanoista Continious Integration ja Continious Delivery. Suomennettuna se tarkoittaa jatkuvaa käyttöönottoa ja jatkuvaa toimintaa. CI/CD putki kulkee ohjelmoijalta tuotannossa olevaan palveluun. Ensin lisätty muutos koodissa laitetaan repositoryyn eli GitLabin tietolähteeseen. Tässä

vaiheessa suoritetaan automaattisia testejä. Näiden testien tarkoitus on varmistaa, että muutokset toimivat ja ovat organisaation määritysten mukaisia. Jos testit eivät mene läpi, pysähtyy palvelun julkaiseminen. Jos testit menevät onnistuneesti läpi, voidaan muutokset julkaista automaattisesti tai manuaalisesti. (GiLab-dokumentaatio.)

5.6 MongoDB

MongoDB on MongoDB Inc:n kehittämä tietokantaohjelmisto. Se on NoSql tietokanta. MongoDB perustuu BSON tietomuotoon. Lyhenne tulee englannin kielen sanoista Binary JavaScript Object Notation (JSON). Se on JSON-kaltainen tietomuoto. (Hows 2016.)

6 Integraatio toteutus

6.1 Projektin kuvaus

Projektin tarkoituksena oli toteuttaa Hsubject EMP -integraatio. Toimeksiantajayrityksellä Plugit Finland Oy:llä on oma sähköautojen lataukseen tarkoitettu sovellus sekä verkkosivu latauslaitteiden hallintaan ja ylläpitämiseen. Projektin toteuttamisessa molempiin palveluihin tehtiin muutoksia ja uudistuksia. Projektissa tehtiin myös uusi mikropalvelu hoitamaan tiedonsiirtoa Hsubjectin kanssa.

6.2 Suunnittelu

Hsubject EMP -integraation suunnittelussa on välttämätöntä tutustua Hsubjectin OICP-protokollaan huolella. Kaikki suunnittelu alkoi uudesta mikropalvelusta, joka tulisi hoitamaan viestinnän Hsubjectin rajapinnan kanssa. OICP-protokolla määrittää dokumentissaan kaikki pakolliset ja tarvittavat ominaisuudet.

Plugitin sovelluksen suunnittelussa täytyi miettiä, kuinka Hsubjectilta tulevat tietotyypit sopivat jo valmiiseen sovellukseen. Kun tarvittavia tietoja tallennetaan sovelluksen muistiin, täytyy tehdä päätöksiä, tallennetaanko tieto alkuperäisessä muodossa vai muodossa, joka on jo sovelluksessa käytössä.

Plugitin henkilökunnalle tarkoitetun hallintaverkkosivuun suunniteltiin uusia ominaisuuksia ja sivuja. Uusien sivujen tarkoitus oli esittää tarvittavat tiedot, jotka saadaan Hsubjectista. Tarvittavat tiedot suunniteltiin siten, että henkilökunta pystyy mahdollisissa ongelmatilanteissa auttamaan asiakasta roaming-lataamiseen liittyvissä ongelmissa. Sivujen toteutuksessa ja suunnittelussa käytettiin samankaltaisia tyylejä ja toimintoja, jotka löytyivät jo käytössä.

6.3 Toteutus

Integraation toteutus jaettiin kahteen osaan, frontendiin ja backendiin. Frontend on koodia, jota ajetaan selaimessa ja backend on koodia, jota ajetaan palvelimella. Backend-puolella luotiin kokonaan uusi mikropalvelu vastaamaan OICP-protokollan vaatimuksia. Frontend-puolella ohjelmoitiin jo valmiiseen sovellukseen valmius roaming-palvelun käyttämistä varten. Frontend-ohjelmoinnissa käytettiin Angular-ohjelmistokehystä.

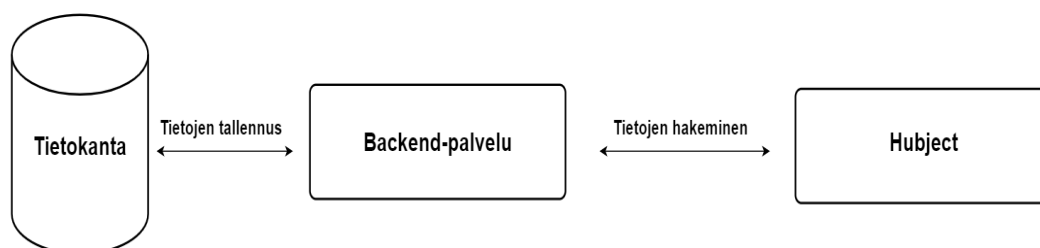
```
{
  "EvseStatuses": {
    "OperatorEvseStatus": [
      {
        "EvseStatusRecord": [
          {
            "EvseID": "DE*XYZ*ETEST1",
            "EvseStatus": "Available"
          }
        ],
        "OperatorID": "DE*ABC",
        "OperatorName": "ABC technologies"
      }
    ],
    "StatusCode": {
      "AdditionalInfo": "Success",
      "Code": "000",
      "Description": "string"
    }
  }
}
```

KUVA 3. Latauslaitteen tilatietojen http-pyynnön vastaus Hubjectista. (OICP-dokumentaatio)

Backend-palvelu ohjelmoitiin TypeScript-ohjelmointikielellä. Palvelun toteuttaminen aloitettiin kirjoittamalla kaikki tietotyypit, jotka löytyvät OICP-dokumentaatiosta. Tietotyypit ovat hyödyllisiä ohjelmoissa, sillä ohjelmointiympäristö varoittaa ja kertoo mitä tietokenttiä tietotyyppihin kuuluu ja yritetäänkö käyttää tieto-

kenttiä, joita ei ole olemassa. Kuvassa 3 esimerkki yksittäisestä roaming-latauslaitteen tilatiedosta, joiden mukaan tietotyyppejä luotiin. Tietotyyppeihin voidaan lisätä kysymysmerkillä tietokenttiä, jotka eivät ole pakollisia, mutta saattavat löytyä.

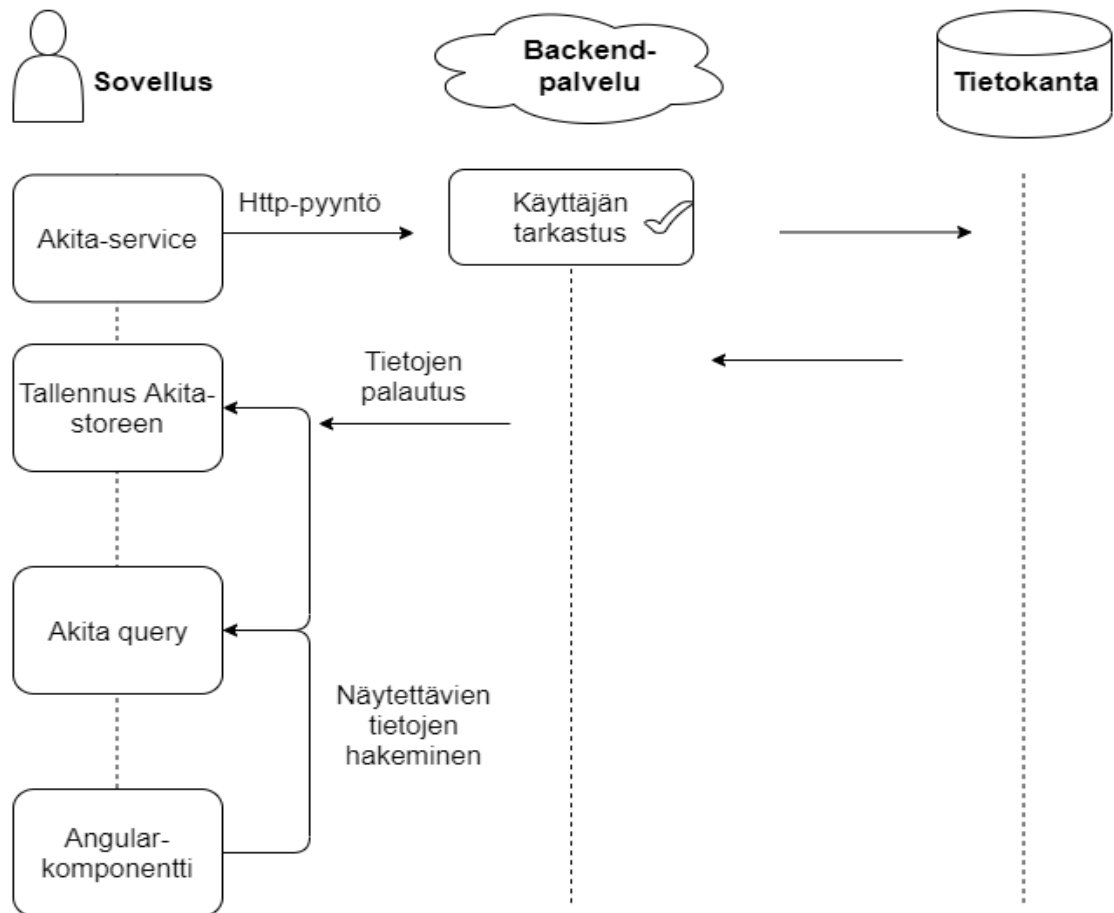
Backend-palvelu hakee latauslaitteet ja niiden tilatiedot Hubjectin rajapinnasta. Hubjectilla on vaatimus, että latauslaitteet on haettava uudelleen vähintään kerran vuorokaudessa ja tilatietoja viiden minuutin välein. Ajastetun toiminnon toteuttamiseen käytettiin Agendaa. Se on npm-paketti, joka hoitaa ajastetut tehtävät. Tehtävän suorituksen voi ajoittaa joko tietylle ajalle tai tietyn ajan välein.



KUVIO 5. Tiedon hakeminen Hubject-palveusta.

Latauslaitteet haetaan http-pyyntöllä. Vastaukseksi saadut latauslaitteet tallennetaan tietokantaan. Kun tiedot tallennetaan tietokantaan ei tarvitse aina hakea tarvittavia tietoja Hubjectin palvelimelta asti, kun palveluun tulee frontendista uusia pyyntöjä.

Sovelluksessa latauslaitteet haetaan kaikki kerralla aina kun sovellus käynnistetään. Latauslaitteiden hakemiseen käytetään Angularin http-kirjastoa, mikä on tarkoitettu http-pyyntöjen lähettämiseen. Vastaukseksi saadut latauslaitteet tallennetaan sovelluksen muistiin Akitan storeen. Sitten tarvittavat tiedot haetaan sovelluksen muistista eikä joka kerta tarvitse suorittaa raskaita http-pyyntöjä palvelimelle. Myös latauslaitteiden hintatiedot haetaan aina kerralla sovelluksen käynnistyessä, jotta ne ovat nopeasti käytettävissä.



KUVIO 6. Sovelluksesta lähtevän http-pyyntön kulku.

Latauslaitteiden tilatietoja eli onko latauslaite käytössä, vapaana tai vikatilassa on haettava viiden minuutin välein. Latauslaitteita ja niiden tilatietoja on liian suuri määrä siihen, että kaikkien latauslaitteiden tiedot haettaisiin viiden minuutin välein. Tilatiedot haetaan aina siinä vaiheessa, kun kyseinen latauslaite on näkyvillä sovelluksessa. Angularin rxjs-kirjastosta löytyy timer-ominaisuus. Se on observable, joka toteutuu sille millisekunneissa annetun ajan kuluttua. Observable on Angular-kehityksessä käytettävä toiminto, jota tarkkaillaan. Sen avulla suoritetaan asynkronisia tiedonhakuja ja voidaan välittää tietoa Angular-projektissa.

Latauksen aloitus- ja lopetuspyyntö lähtee sovelluksesta http-pyyntöllä. Aloitukseen ja lopetukseen tarvitaan käyttäjän yksilöivä tunnus ja latauslaitteen uniikki tunniste. Jos Pluginin käyttäjältä ei ole vielä tehty Hubjectin tietokantaa tunnusta, se luodaan ensimmäisen aloituspyynnön tullessa. Backend-palvelu vastaanottaa pyynnön ja tekee tarvittavat tarkistukset ja välittää tiedon eteenpäin Hubjectille. Kun vastaus Hubjectilta tulee, se tallennetaan tietokantaan ja välitetään takaisin

sovellukseen. Keskenkäynteiset lataustapahtumat haetaan aina sovelluksen käynnistyessä, jos käyttäjä on sulkenut sovelluksen kesken lataustapahtuman. Ne tallennetaan sovelluksen muistiin.

6.4 Testaus

Integraation käyttöönoton ajaksi Hsubject tarjoaa kehitysympäristössään testidataa. Testidatan kuuluisi olla OICP-dokumentaation mukaista ja sen pohjalta pitäisi pystyä kehittämään omaa palvelua. Kaikki data on myös niin sanotusti toimivaa eli testiympäristössä voi aloittaa ja lopettaa latauksia. Lataustapahtumia ja latauslaitteiden tilatietoja voidaan seurata myös Hsubjectin portaalin kautta.

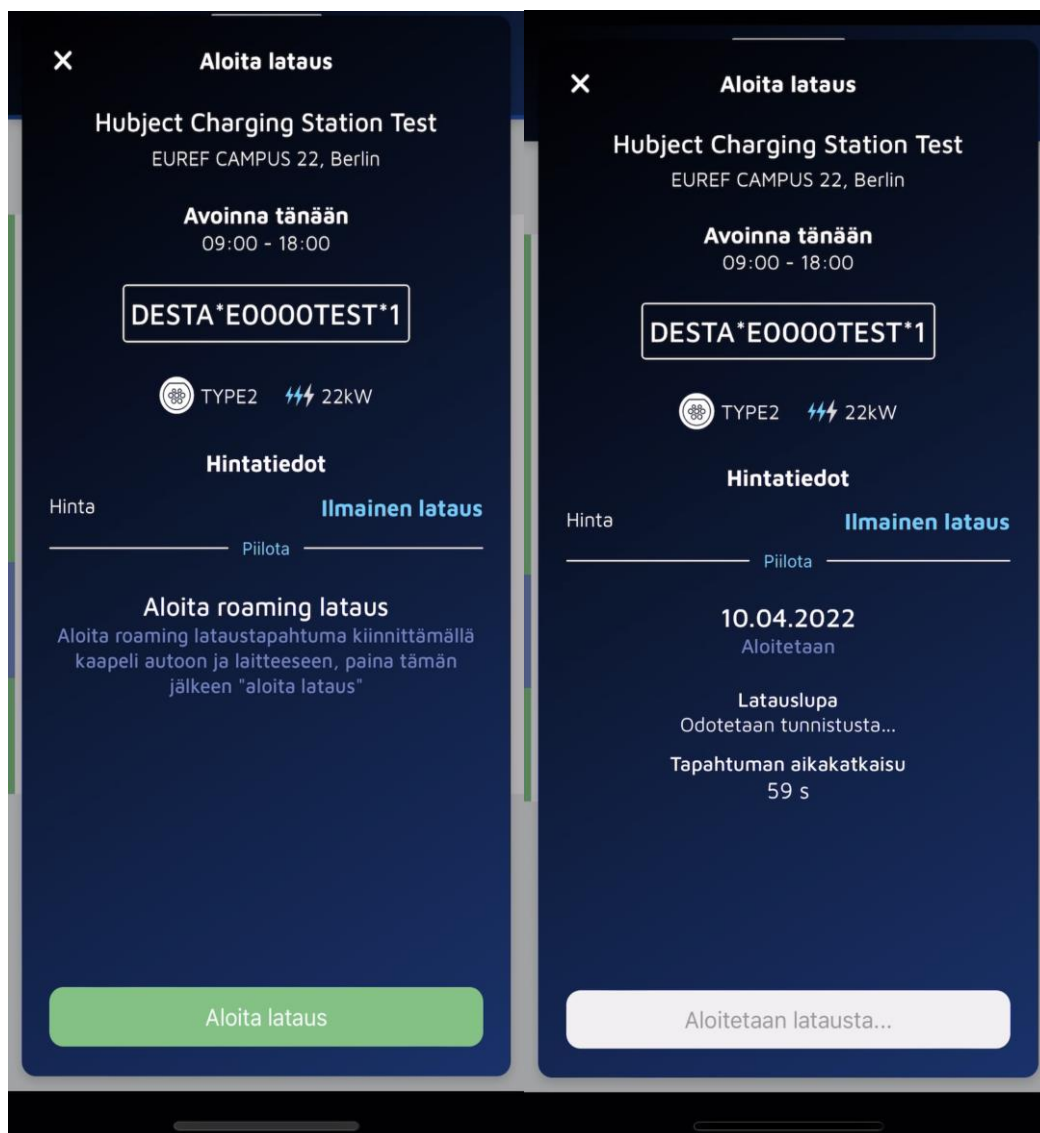
EMPn käyttöönoton aikana Hsubjectin kanssa suoritetaan kaksi testiä. Testit ovat samanlaisia, mutta toinen niistä tehdään testiympäristössä ja toinen oikeassa tuotantoympäristössä. Testien tavoitteena on selvittää, että kaikki toimii joka suuntaan oikein. Testissä aloitetaan ja lopetetaan latauksia mobiilisovelluksesta ja rfid-tunnisteella. Jokaisesta latauksesta tarkistetaan saako Hsubject tarvittavat ja oikeilla arvoilla varustetut lataustapahtumat. Testiympäristössä suoritettavien onnistuneiden testien jälkeen sama tehdään tuotantoympäristössä.

6.5 Lopputuloksia

Uuden roaming-latausmahdollisuuden takia, sovellukseen tehtiin muutamia uusia näkymiä ja muita ominaisuuksia. Näkymien tyylit tehtiin vastaamaan jo olemassa olevia tyylejä. Sovelluksen käyttäjän täytyy kuitenkin huomata, mitkä kaikki asiat sovelluksessa ovat roaming-lataamiseen liittyviä ja mitkä ovat Plugitin omia.

Kuvassa 4 näkymä roaming-latauslaitteesta. Erotettuna muista latauslaitteista kyseiseen näkymään laitettiin omat roaming-lataukseen liittyvät ohjetekstit. Myös keskellä oleva latauslaitteen uniikki tunniste, mikä esimerkkikuvassa on "DESTA*E000TEST*1", on Plugitin latauslaitteiden tunnisteisiin verrattuna erilainen. Näkymän alalaidassa oleva "Aloita lataus" -painiketta painaessa sen väriytyy

muuttuu, mikä kertoo käyttäjälle latauksen aloituksen alkaneen. Käyttäjälle myös näytetään, kuinka latauksen aloitus etenee. Aloituksessa on 60 sekunnin katkaisuaika, jos vastausta backend-palvelusta ei ole tullut siinä ajassa, käyttäjälle ilmoitetaan virheestä ja aloituksen yritys lopetetaan.



KUVA 4. Plugitin sovelluksessa latauksen aloitus ruudut Hubjectin tarjoamalla testilatauspisteellä. (Plugit-sovellus)

Sovelluksessa on mahdollisuus valita suodattimien avulla, millaisia latauspisteitä asiakas haluaa kartallaan nähdä. Valinnat valitaan vaihtokytkimiä painamalla. Sovelluksessa voidaan suodattaa latauspisteitä, esimerkiksi lataustehon, latauspistokkeen tai ilmaisten latauspisteiden mukaan. Integraation seurauksena roaming-latauspisteiden näyttäminen lisättiin suodattimien joukkoon.

Hinta suodatus

Suodata hinnan mukaan



Roaming

Näytä roaming asemat



Yksityiset latausasemat

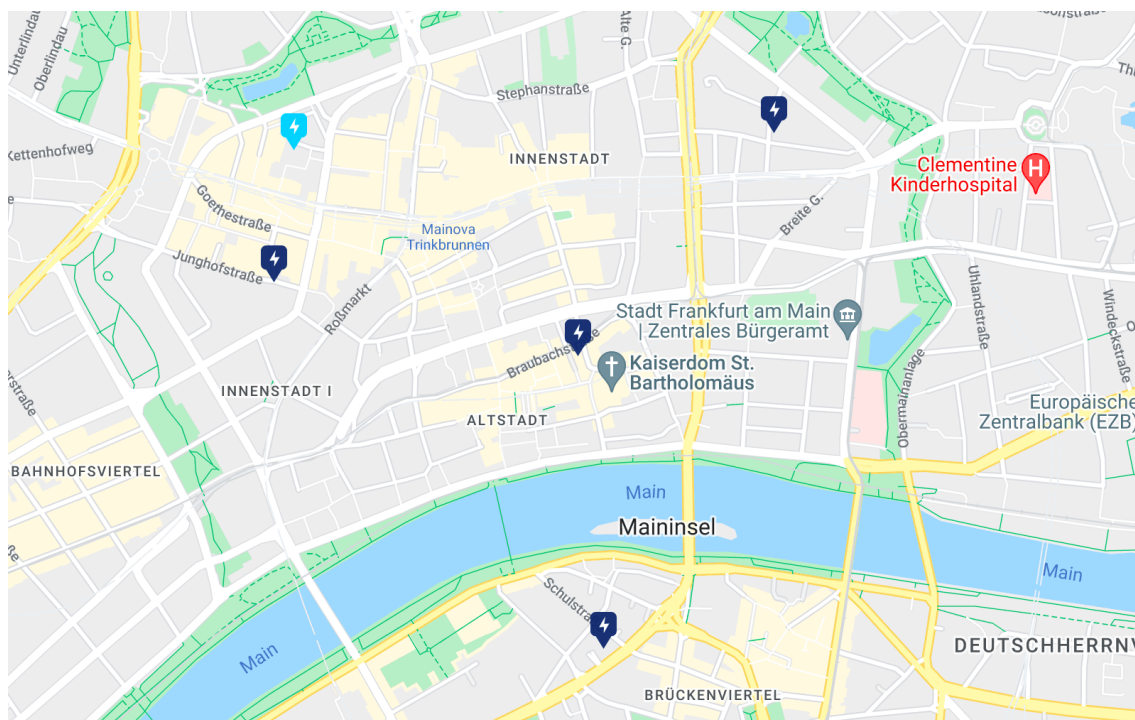
Näytä vain yksityiset pisteet



Kuva 5. Osa sovelluksen latauspisteiden suodattimien valinnoista. (Plugit-sovellus)

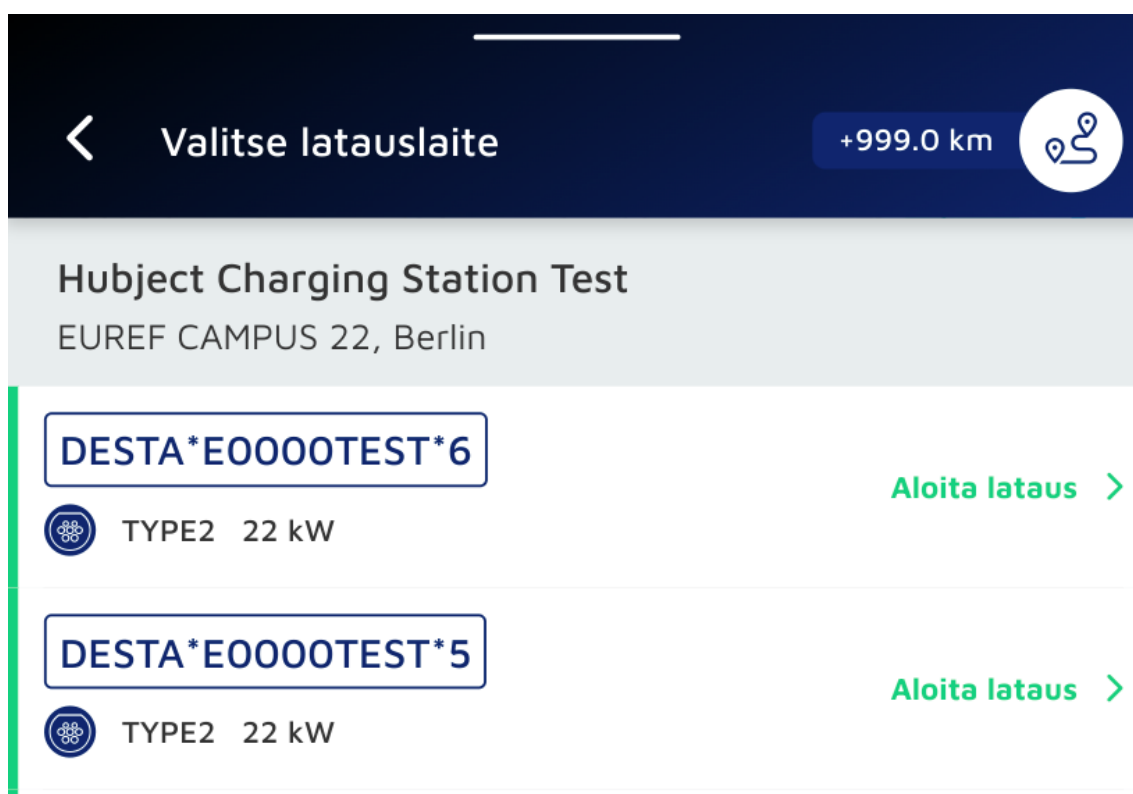
Latauspisteitä näytetään sovelluksessa kartalla. Pisteet ovat merkitty karttaan merkeillä. Plugitin omat latauspisteiden merkit ovat Plugitin omalla logolla.

Roaming-pisteet ovat merkattu erilaisella sinisellä logo, joka muuttuu turkoosiksi kun latauspisteen valitsee.



Kuva 6. Sovelluksen karttanäkymässä roaming-latauspisteitä. (Plugit-sovellus)

Latauspisteen valitsemalla avautuu latauslaitteiden listanäkymä. Näkymässä näytetään latauspisteestä osoitetiedot ja näkymästä on mahdollista avata Google Maps -navigointi kyseiselle pisteelle. Latauslaitelista koostuu yksittäisistä latauslaite-elementeistä. Latauslaitteen tilaa näytetään eri väreillä. Jos laite on vapaana, väri on vihreä ja elementin reunassa oleva teksti kertoo latauksen aloituksen olevan mahdollinen. Kun latauslaite on varattu tai vikatilassa, niin elementin värit ovat punaisella tai harmaalla. Latauslaitteen latausteho ja pistoketyyppi näytetään myös elementissä.



KUVA 7. Roaming-latauslaitteiden lista. (Plugit-sovellus)

7 POHDINTA

Sähköautojen määrät ovat kovassa kasvussa. Latauspisteiden määrän tärkeys korostuu, jotta käyttäjille voidaan tarjota latausmahdollisuuksia entistä paremmin. eRoamingin käyttöönotolla voidaan tarjota asiakkaille enemmän latauspisteitä, jolloin asiakkaat eivät tarvitsisi muita palvelun tarjoajia.

Integraation toteuttaminen on suuri projekti. Vaikka Hsubject tarjoaa erittäin kattavan dokumentaation, on integraation toteuttaminen haastavaa. Kaikista haastavinta työn tuloksen saavuttamiseksi oli Hsubjectista tulevan tiedon yhteensovittaminen jo olemassa olevaan järjestelmään.

Työ onnistui melko hyvin. Työn tarkoituksena oli toteuttaa Hsubject EMP -integraatio ja tavoitteena tutkia ja kerätä tietoa integraatioista ja niiden ratkaisumalleista. Tietoa integraatiosta ja eri ratkaisumalleista löytyi hyvin ja tieto oli luotettavaa. Sähköautoilu on vielä melko uusi käsite eikä sen takia eRoamingista löydy niin paljoa tietoa, kun olisi tarvetta. Sähköautoilu on kuitenkin koko ajan kasvamassa ja eRoaming mahdollisuutta tarjoavien yritysten määrä kasvaa. Jatkokehityksenä työlle olisi Hsubjectin käyttöönotto CPO:n roolissa. Lisäksi jatkotutkimusta eRoamingin käyttöönoton vaikutuksesta lataustapahtumien ja asiakkaiden määrään voisi toteuttaa.

LÄHTEET

Akita-dokumentaatio. <https://datorama.github.io/akita/docs/store> Luettu 01.01.2022

Angular-dokumentaatio. <https://angular.io/docs> Luettu 01.12.2021

EVRoaming-säätiö kotisivu. <https://evroaming.org/> Luettu 30.3.2022

Ferwerda, R. et al. (2018) Advancing e-roaming in Europe: towards a single 'language' for the European charging infrastructure. World electric vehicle journal. [Online] 9 (4), 50–.

Fossiilittoman liikenteen tiekartta. 2021. Valtionneuvoston periaatepäätös kotimaan liikenteen kasvihuonekaasupäästöjen vähentämisestä. <http://urn.fi/URN:ISBN:978-952-243-588-0> Liikenne- ja viestintäministeriö. Luettu 15.2.2022.

GitLab dokumentaatio. <https://docs.gitlab.com/> Luettu 31.3.2022

Hohpe, Gregor., and Bobby. Woolf. Enterprise Integration Patterns : Designing, Building, and Deploying Messaging Solutions. 1st edition. Boston: Addison-Wesley, 2003. Print.

Hows, D. et al. (2013) The Definitive Guide to MongoDB: A Complete Guide to Dealing with Big Data Using MongoDB. Berkeley, CA: Apress L. P.

Hubject-kotisivu. <https://www.hubject.com/> Luettu 01.12.2021

Jansen, R. H. et al. (2016) TypeScript. Birmingham: Packt Publishing, Limited.

Manouvrier, B. & Ménard, L. (2008) Application integration EAI, B2B, BPM and SOA. London: ISTE.

OCCP-protokolla. <https://www.openchargealliance.org/> Luettu 25.02.2022

OICP-protokolla. <https://github.com/hubject/oicp/tree/master/OICP-2.3> Luettu 10.10.2021

Schorow, D. et al. (2008) The Definitive Guide to SOA Oracle Service Bus. 2nd ed. 2008. [Online]. Berkeley, CA: Apress.

Summers, B. L. (2013) Effective methods for software and systems integration. 1st edition. [Online]. Boca Raton: CRC Press.

Sähköisen liikenteen tilannekatsaus Q4/2021. Sähköinen liikenne ry

Tähtinen, S. (2005) Järjestelmäintegraatio: tarve, vaihtoehdot, toteutus. Helsinki: Talentum.

Vohra, D. (2016) Kubernetes Microservices with Docker. 1st ed. 2016. [Online]. Berkeley, CA: Apress.