

Etäkohteiden monitorointijärjestelmän suunnittelu ja pilotointi



Ammattikorkeakoulututkinto opinnäytetyö

Valkeakosken kampus, Sähkö- ja automaatiotekniikka, insinööri (AMK)

Syksy 2020

Jaakko Saarela

TIIVISTELMÄ

Tämä opinnäytetyö on tehty opinnäytetyöntekijän omiin tarpeisiin, jolloin innoittajana on ollut omistamani erilaisten kohteiden etäseurannan tarve. Lähtökohtana oli pystyä toteuttamaan edullinen ja eri tarpeisiin mukautuva valvonta- ja ohjausjärjestelmä. Koska esineiden internet (IoT) mahdollistaa pitkäikäisiä mittalaitteita verkkoon, oli tavoitteena tehdä selvitys, miten IoT-järjestelmiä voitaisiin käyttää hyödyksi kohteiden etämonitoroinnissa sekä -kontrolloinnissa.

Opinnäytetyössä ensimmäisenä määriteltiin kohdeympäristöjen kautta vaatimukset koko järjestelmälle ja sen osille. Toisena vaiheena oli järjestelmän osien valinnat, perustuen asetettuihin vaatimuksiin. Kun järjestelmän osat oli valittu, tehtiin järjestelmäarkkitehtuurisuunnittelu, joka jakaantui HW-suunnitteluun, ohjelmistosuunnitteluun sekä itse kokonaisjärjestelmäsuunnitteluun. Lopuksi järjestelmästä tehtiin ensimmäisen vaiheen pilottiversio, jolla todennettiin konseptin toimivuus.

Kun esineiden internet on nopeasti kehittyvä eri teknologioiden osalta, selvitettävä maailma muuttui erittäin nopeasti. Opinnäytetyön valinnat ovat myös näin ollen yhden ajanjakson läpileikkaus. Loppujen lopuksi itse pilottivaiheen toteutuksen tekeminen oli nopein vaihe opinnäytetyössä. Opinnäytetyö toteutti tarpeensa ja opinnäytetyön myötä myös muita sovellusideoita syntyi opinnäytetyötä tehtäessä. Opinnäytetyö antoi varsin hyvän pohjan seuraavalle vaiheelle ja jopa tuotteistukselle nykyvalintoihin perustuen.

ABSTRACT

Inspiration of the project was derived from different facilities owned by author that required remote monitoring and control. The goal of the project was to design a remote control and monitoring system, that would be inexpensive and capable to be adopted to different needs. Because the Internet of Things is providing possibilities for create inexpensive systems with a long battery life, target of the thesis was to examine the proof of concept of remote monitoring system based on selected IoT technologies.

The first phase in the thesis project was to define the requirements for the system and its parts based on predefined remote monitoring facilities. The second phase was to make required selections for the system technologies. When the selections were made, the third step was to design system architecture, which included system design, HW design and SW design. The final part of the thesis project was to create the proof-of-concept implementation with the selected technologies.

IoT based systems are evolving really fast, therefore the research environment was also changing during the project. Because of this, this thesis is snapshot of the IoT lifecycle. In the end the thesis met the set targets and also introduced the new application ideas that it would be possible to develop with this kind of technology. This thesis gives a good basis for the next steps and even for a productization of the system with selected technologies.

Keywords Android, cross-platform development, Internet of things, iOS, remote control, remote monitoring

Pages 50 pages

Sisällys

1	Johdanto	1
2	Kehittämistyön tietoperusta	1
2.1	Etähallintajärjestelmän perusteet	2
2.2	IoT-protokollat	3
2.2.1	Low Power Wide Area Networks (LPWAN).....	4
2.2.2	Matkapuhelinverkot.....	12
2.2.3	Matkapuhelinverkkojen kattavuus DNA-verkossa.....	13
2.2.4	Wireless Personal Area Networks.....	14
2.2.5	Mesh-verkot	14
2.2.6	Verkkoteknologioiden vertailu.....	14
2.3	IoT-kehitysalustat.....	15
2.3.1	Raspberry-Pi	17
2.3.2	Arduino.....	18
2.3.3	Particle Electron	19
2.3.4	Adafruit Fona.....	21
2.3.5	LoPy4	21
2.3.6	Kehitysalustojen vertailutaulukko.....	22
3	Projektin suunnittelu ja toteutus	22
3.1	Systeemivaatimukset	23
3.1.1	Päätelaitteen vaatimukset	23
3.1.2	Pilvipalvelun vaatimukset.....	24
3.1.3	Asiakaspäätelaitesovelluksen vaatimukset.....	24
3.1.4	Kokonaisjärjestelmävaatimukset	24
3.2	Valitut teknologiat	25
3.2.1	Verkkoteknologiavalinta	25
3.2.2	Paikannusteknologiavalinta	26
3.2.3	Kehitysalustavalinta	26
3.3	Systeemiarkkitehtuuri.....	29
3.3.1	Päätelaitesensorit.....	31
3.3.2	Päätelaitesovellus.....	33
3.4	Asiakaspäättesovellus	38
3.4.1	Xamarin	39
3.4.2	iOS ja Android -asiakassovellukset.....	40

4 Johtopäätökset	41
Lähteet.....	45

Kuvat, taulukot ja kaavat

Kuva 1. LoRa-protokollapino (Nordin, n.d.).....	5
Kuva 2. LoRaWAN-tähtitopologia (LoRaWAN, 2015, s. 8).....	6
Kuva 3. LoRaWAN-verkon päätelaitekategoriat (LoRa, n.d.)	7
Kuva 4. Digitan LoRaWAN-verkon kattavuuskartta Suomessa (Digita, n.d.).....	8
Kuva 5. Sigfox-verkkotopologia (Sigfox build, n.d.-b).....	10
Kuva 6. Järjestelmäarkkitehtuuri.....	30
Kuva 7. Particle Electron:in liityntädiagrammi (Particle Docs, n.d-c, s.11).	32
Kuva 8. Päätelaitesovelluksen käynnistyminen.....	33
Kuva 9. Hätävirransäästöprofiilin vuokaavio	34
Kuva 10. Normaalivirransäästöprofiilin vuokaavio	35
Kuva 11. Hälytys- ja mittausprofiilin vuokaavio	36
Kuva 12. Normaalimittausprofiilin vuokaavio	37
Kuva 13. Xamarin-arkkitehtuuri (What is Xamarin?, 2020).....	39
Kuva 14. .Net core 5.0 (Lander, 2019)	41
Taulukko 1. Verkkoteknologioiden vertailu	15

Liitteet

Liite 1	Kehitysalustojen vertailutaulukko
Liite 2	2G ja 3G -verkon kattavuus Suomessa DNA-operaattorilla
Liite 3	4G ja 5G -verkon kattavuus Suomessa DNA-operaattorilla.
Liite 4	NB-IOT ja LTE-M -verkkojen kattavuus Suomessa DNA-operaattorilla.
Liite 5	Sigfox-verkon kattavuus Suomessa.
Liite 6	Sensoreiden liittäminen Particle Electron Asset Tracker:iin
Liite 7	Sensoritiedon luku- ja ohjauskomennon havainnollistava sekvenssikaavio

1 Johdanto

Tämä työ on syntynyt opinnäytetyön tekijän omista tarpeista. Tekijällä on useita etävalvottavia kohteita Suomessa ja työn lähtökohtana on ollut selvittää voiko edullisilla kaupallisilla IoT-kehitysalustoilla valmistaa vaatimukset täyttävän järjestelmän. Järjestelmävaatimuksina olisivat järjestelmän edullisuus, hankalia sääoloja kestävä päätelaite, joka pystyy toimimaan huonojen verkkoyhteyksien päässä. Laitteen olisi tarve mukautua erilaisiin mittaustarpeisiin, toimia pienellä energialla ja alhaisella dataliikenteellä. Merkittävänä vaatimuksena järjestelmälle on myös ollut mahdollistaa toimilaitteiden etäkontrollointi kohteessa. Tällaisia etäohjattavia toimilaitteita kohteissa voivat olla esimerkiksi valot ja lämmitysjärjestelmä, jotka pitäisi pystyä kytkemään päälle tai pois etänä. Järjestelmävaatimukset ja järjestelmän eri osien vaatimukset on listattu myöhemmin tässä opinnäytetyössä.

Vaikka etävalvonta- ja etäkontrollointijärjestelmiä on kaupallisestikin tarjolla, tämän työn tarkoituksena on luoda perusta järjestelmän myöhemmälle kaupallistamiselle tekijän oman toiminimen tuotteeksi. Opinnäytetyön tavoitteena on tehdä selvitys erilaisten teknologioiden ja tarvittavan elektroniikan osalta. Lisäksi oli tarkoitus tehdä konseptivaiheen selvitys ja ratkaisu ohjelmistojen osalta. Tuotteistamiseen liittyvät vaiheet jätetään tämän opinnäytetyön ulkopuolelle.

2 Kehittämistyön tietoperusta

Nykyään IoT (Internet of Things) alkaa olla niin verkkopuolella kuin kehitysalustojen kautta erittäin kilpailukykyinen vaihtoehto kehittää laite kulloisenkin tarpeen mukaan. IoT-kehitysalustat tarjoavat hyvin laajan tuotetarjonnan erilaisiin verkkoteknologioihin sekä mahdollistavat hyvät laajennusmahdollisuudet erilaisiin tarpeisiin.

Kaupallisina löytyy kehitysalustoja erilaisiin likiverkkoratkaisuihin kuin myös verkkoratkaisuihin, joilla voi kattaa erittäin laajoja alueita globaalisti. Likiverkkoratkaisut ovat tyypillisesti paikallaan olevien kohteiden monitorointiin ja ohaukseen tarkoitettuja verkkoratkaisuja. Likiverkkoratkaisujakin voitaisiin käyttää opinnäytetyön ratkaisussa, mutta

sensoreiden ja koko maan kattavan verkon välissä tarvittaisiin reititin. Reititin tässä tapauksessa olisi silta sensoreiden ja globaalien verkon välillä. Järjestelmään lisätyt moduulit taas lisäävät järjestelmän kompleksisuutta sekä energiahallinnan monimutkaisuutta. Tämän vuoksi tämä opinnäytetyö ei käy läpi näitä teknologioita, vaan keskittyy järjestelmäratkaisuihin, joitka kattavat koko maan ja ovat mahdollisimman kompakteja.

2.1 Etähallintajärjestelmän perusteet

Opinnäytetyön ajatuksena oli määritellä vaatimukset järjestelmälle ja tehdä teknologiset valinnat, jotka parhaiten toteuttaisivat asetetut vaatimukset. Työn tarkoituksena oli myös valintojen perusteella toteuttaa ensimmäisen vaiheen prototyyppi. Etävalvottavia ja -kontrolloitavia kohteita, jotka asettivat osan vaatimuksista, olivat asuntovaunu, huvivene sekä mökki lapissa. Kattaakseen näiden kohteiden etävalvontatarpeet ja -kontrollointitarpeet, järjestelmän on pystyttävä toimimaan hyvin erilaisissa ympäristöissä ja pystyttävä näin olleen antamaan riittävän tarkkaa tietoa kohteesta loppukäyttäjälle. Niin huvivene kuin asuntovaunu voivat liikkua erittäin huonoilla kuuluvuusalueilla maanrajojen sisällä, jolloin verkon kattavuus Suomessa on yksi perusvaatimuksia. Laitteen globaalinen tavoitettavuus otetaan toki työn osana huomioon, mutta se ei tämän työn osalta toimi valintojen kriteerinä, vaan verkon saavutettavuusvaatimus pidetään Suomen rajojen sisällä. Lisäksi kohteiden osalta paikannustiedon saaminen loppukäyttäjälle näkyviin riittävällä tarkkuudella oli merkittävä perusvaatimus. Koska asuntovaunu ja huvivene ovat liikkuvia kohteita, yksi perusvaatimus oli myös katkeamaton liikkuvuus valitussa verkossa.

Koska yksi kohteista on myös mökki lapissa, voivat laitteen huoltovälit olla erittäin pitkät. Jos mökillä käydään kerran vuodessa, laitteen pitäisi pystyä toimimaan huoltovälin yli ilman fyysistä kontrollitarvetta. Tällöin vaatimukseksi laitteelle tuli myös laitteen energiahallinta ja mahdollisuudet useaan energian syöttölähteisiin. Myös lapin olosuhteiden osalta suuret lämpötilaerot ja suuret kosteusvaihtelut, asettavat omat haasteensa mittaavalle päätelaitteelle.

Pitkien huoltovälien myötä merkittäviä vaatimuksia ovat eri energialähteiden hyödyntäminen päätelaitteessa sekä päätelaitteen mahdollisuudet virransäästöön. Varsinkin mökki lapissa on tämän suhteen haastava. Kohteessa on aurinkosähköjärjestelmä, jonka

tarjoamaan energiaa on tarkoitus hyödyntää ensisijaisesti. Energiatason vajotessa aurinkojärjestelmässä, päätelaitteen on tarkoitus hyödyntää omaa energialähdettä ja indikoida loppukäyttäjälle tilanteesta. Varsinkin talviaikana, jolloin aurinkoenergiaa ei ole tarjolla, järjestelmän energiansäästö tulee olemaan merkittävässä osassa järjestelmän toimivuutta. Kun päätelaite voi hyödyntää energiaansa useasta lähteestä ja energialähteen valintaa voitaisiin hallita etänä, mahdollistaisi tämä laitteelle pitkän toiminta-ajan ilman paikalla tehtävää huoltoa. Energiahallinnan osalta myös energian käytön optimointimahdollisuudet päätelaitteessa, olivat merkittävä osa perusvaatimuksia.

Asiakassovellukselle vaatimuksesi tuli monialustakehityksen hyödyntäminen. Tarkoitus tällöin on hyödyntää samaa ohjelmistoa eri alustoille. Alustoiksi asiakaspäätössovellukselle on määritelty iOS ja Android.

Työssä tutkittiin erilaisten verkkojen sopivuutta ja tutkittiin kaupallisten IoT-kehitysalustojen sopivuutta järjestelmälle asetettuihin perusvaatimuksiin. Valintojen pohjalta kehitettiin lopuksi pilottivaiheen toteutus, jolla todennettiin tehtyjen valintojen sopivuus.

Valintoihin perustuva tutkimus tehtiin internetistä löytyviä tietoja vertaillen. Valinnan pohjaksi järjestelmälle kirjoitettiin vaatimukset, jotka järjestelmän tulisi täyttää. Vaatimuksia on kirjattu niin koko järjestelmälle, kun sen erillisille osille. Vaatimukset on listattu tarkemmin luvussa Systemivaatimukset.

2.2 IoT-protokollat

Koska seurattavan järjestelmän kohdemaana on Suomi, niin järjestelmän käyttämän verkon tulisi kattaa koko Suomi siten, että etäseurattava ja -kontrolloitava kohde voi sijaita missä päin Suomea tahansa. Tällöin kohde pitäisi olla saavutettavissa myös niin sisävesi kuin Suomen merialueilla. Myös harvaan asutut alueet tulee huomioida valintoja tehdessä.

Erilaisia IoT-verkkoteknologioita ovat:

1. Low power Wide Area Networks (LPWAN)
2. Matkapuhelinverkko

3. Wireless Personal Area Networks (WPAN)
4. Mesh

Lähteenä käytetty sivustoa (Ray, 2020). Edellä mainitut IoT-verkkoteknologiat jakaantuvat pienempiin verkkoteknologioihin, joista kerrotaan tarkemmin seuraavissa luvuissa.

2.2.1 Low Power Wide Area Networks (LPWAN)

LPWAN sisältää teknologiat Sigfox, LoRaWAN (long range wireless area network) ja Weightless. Yleisesti kaikki nämä teknologiat ovat IoT-käyttöön optimoituja verkkoja, joten nämä olivat yksi potentiaalinen kohde opinnäytetyön selvitykselle. Seuraavissa luvuissa on avattu kutakin verkkoteknologiaa omassa luvuissaan peilaten verkon soveltuvuutta järjestelmälle asetettuihin perusvaatimuksiin: liikkuvuus verkossa, paikantaminen verkossa, verkon kattavuus Suomessa, verkon elinkaari Suomessa, verkon kommunikaation suojausmekanismit sekä mahdollisuus päätelaitesovelluksen päivittämiseen etänä (OTA update). Systemivaatimukset on listattu tarkemmin luvussa 3.1.

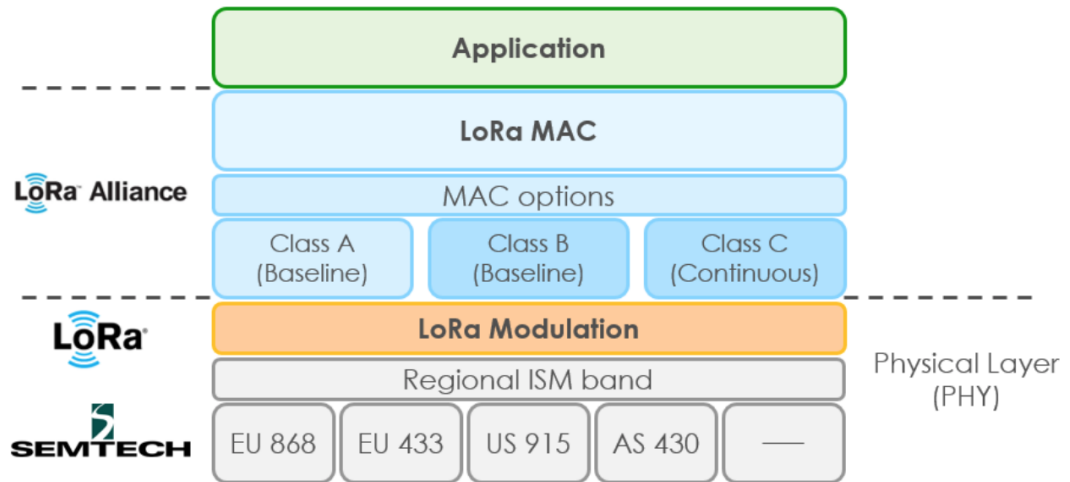
LoRa-teknologia (LoRa muodostuu sanoista Long Range) käyttää alle gigahertsin lisenssivapaita taajuuksia. Koska LoRa-teknologia määrittelee vain tiedonsiirron fyysisen kerroksen, on sen päälle määritelty LoRaWAN-protokolla, joka vastaavasti määrittelee muut tiedonsiirtoon tarvittavat ylemmät protokollakerrokset. Kuva 1 esittää LoRa-protokollapinin kerrokset.

LoRa-teknologia hyödyntää matkapuhelinverkoistakin tuttua hajaspektrimodulaatiota tiedonsiirrossa, jolla saavutetaan erittäin hyvä häiriöiden sietokyky sekä heijastumien ja diffraktion vaikutuksen minimointi siirrettävään signaaliin (Digita, 2017). LoRa-teknologian on kehittänyt ja patentoinut Amerikkalainen Semtech-yritys. LoRa-päätelaitteet ovat pääsääntöisesti suunniteltu langattomiksi ja toimimaan ilman kiinteää yhteyttä erittäin energiatehokkaasti. (LoraWAN, 2015)

LoRaWAN on vastaavasti globaali ja avoin standardi ja tätä hallinnoi LoRa Alliance-järjestö. LoRaWAN-järjestelmä muodostuu LoRa-teknologiaa hyödyntävistä päätelaitteista, reitittimistä, palvelimista sekä loppukäyttäjän sovelluksista. Koska LoRa-teknologia perustuu

lisenssivapaisiin taajuuksiin, tällöin niin yksityiset kuin kaupallisetkin toimijat voivat vapaasti asentaa ja hallita verkon päätelaitteita ja yhdyskäytäviä.

Kuva 1. LoRa-protokollapino (Nordin, n.d.).

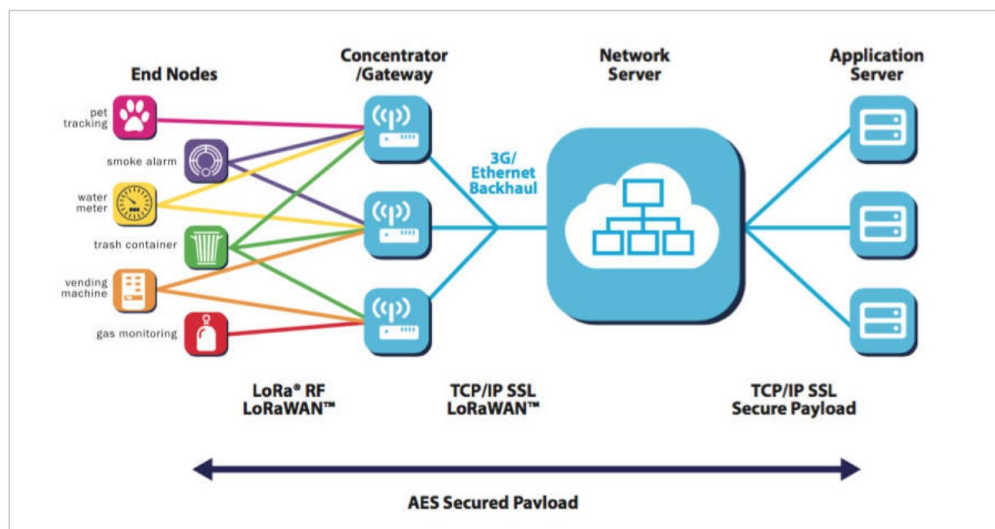


Yleisesti LoRaWAN-verkoissa käytetään star-of-stars-topologiaa (Kuva 2). Tässä järjestelmässä päätelaitteet kytkeytyvät yhdyskäytävään LoRaWAN-protokollaa hyödyntäen. Eri verkkojen yhdyskäytävät taas ovat toisiinsa ja palvelimeen yhteydessä perinteisen IP-protokollien välityksellä. LoRaWAN-päätelaitteen dataa sitten käyttää hyödyksi loppukäyttäjän sovellus. Toisin sanoen monta tähtiverkkoa voivat olla yhteydessä toisiinsa yhdyskäytävien kautta. Yksittäinen päätelaite voi olla siis yhteydessä yhteen tai useampaan yhdyskäytävään saman aikaisesti. Koska päätelaite voi olla useaan yhdyskäytävään yhteydessä yhtä aikaa, LoRaWAN tarjoaa liikkuvuuden koko verkon alueella kaiken tyyppisillä päätelaitteilla. LoRaWAN-protokolla mahdollistaa päätelaitteen ja yhdyskäytävän välillä jopa 20 kilometriä haja-asutusalueilla ja kaupunkiolosuhteissakin useita kilometrejä (Lappi, 2020, s. 23).

LoRaWAN-päätelaitteiden akkujen elinikä on huomattavan pitkä. Hyvissä olosuhteissa päätelaitteen akku voi kestää jopa 10 vuotta. Akun keston toki vaikuttaa päätelaitteen keskustelutaajuus yhdyskäytävän kanssa sekä monet muut parametrit. Muita parametreja ovat: lämpötila, kosteus, päätelaitteen virrankulutus virransäästötilassa, siirrettävän tiedon määrä, verkon kuuluvuus ja halutun mittausoperaation kuluttama teho. Edellisten lisäksi on myös muita parametreja, jotka vaikuttavat päätelaitteen energiankulutukseen. Kuitenkin

edellä mainitut ovat ehkä merkittävimmät ja ovat sellaisia, joihin voi järjestelmäsuunnittelulla helposti vaikuttaa niin asennuspaikan, kuin päätelaitteen käytön perusteella (Semtech, n.d.). Tyypillinen keskustelutaajuus päätelaitteen ja yhdyskäytävän välillä on 15 minuutista yhteen tuntiin, jolloin päätelaite on virransäästötilassa suurimman osan ajastaan (Digita, 2017).

Kuva 2. LoRaWAN-tähtitopologia (LoRaWAN, 2015, s. 8)



LoRaWAN-protokollassa siirrettävän tiedon määrä on rajattu 243 tavuun yhdessä datapaketissa. LoRaWAN on kaksisuuntainen, joten tietoa voidaan siirtää niin päätelaitteelta yhdyskäytävälle kuin yhdyskäytävältä päätelaitteelle, joka myös näin ollen mahdollistaa päätelaitteen sovelluksen päivittämisen etänä. Tiedonsiirto on hyvin salattua AES128 bittisellä salauksella päätelaitteen ja yhdyskäytävän välillä (Lappi, 2020, s. 30).

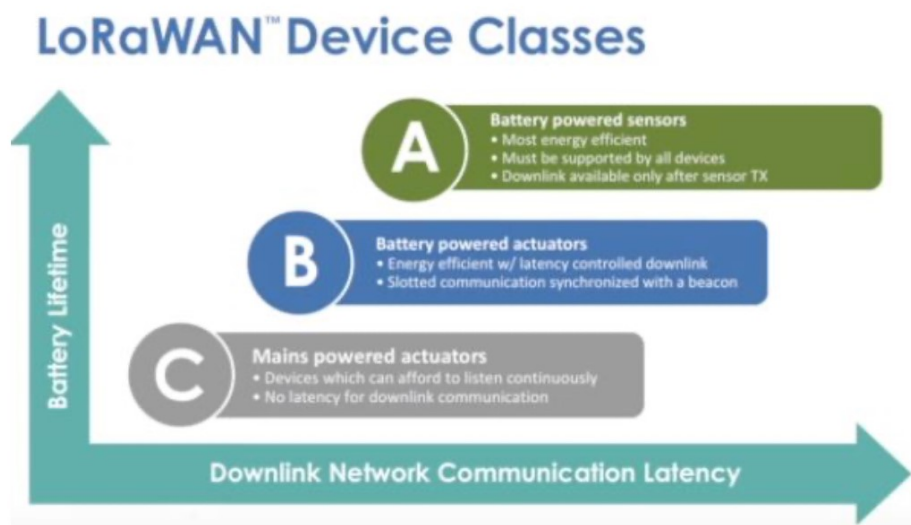
LoRaWAN-päätelaitteet jaetaan kolmeen kategoriaan, päätelaitteen energiankulutuksen ja vastaanoton viiveiden mukaan (LoRa, n.d.). C-luokan laitteet ovat jatkuvan tehonsyötön laitteita, jotka ovat valmiita vastaanottamaan lähetyksiä lähes koko ajan. Vastaanottoikkuna sulkeutuu vain siksi aikaa, kun laite itse lähettää dataa. Laitteen vastaanottoviiveet ovat näin ollen erittäin pieniä, juuri siksi että niiden vastaanottoikkuna on auki lähes koko ajan.

B-luokan laitteet taas ovat akkutehonsyötön varassa olevia laitteita, joilla on A-luokan laitteiden satunnaisten lähetyksikkunoiden lisäksi ennalta tiedossa olevat ajastetut vastaanottoikkunat. Reititin lähettää aikasynkronoidun Beacon-viestin päätelaitteelle, jonka

perusteella päätelaite avaa oman vastaanottoikkunansa. Tällöin myös palvelin tietää milloin kyseisen päätelaitteen vastaanottoikkuna on auki tiedon vastaanottamista varten.

A-luokan laitteet ovat kaikkein energiatehokkaimpia ja akkujen varassa toimivia päätelaitteita. A-luokan laitteiden datan vastaanotto synkronoituu ainoastaan niiden lähetyksen kanssa. Jokaista A-luokan laitteen lähetystä seuraa kaksi vastaanottoikkunaa, jolla laitteen tiedonsiirron kaksisuuntaisuus saadaan toteutettua. A-luokan päätelaitteen datan lähetys onkin täysin asynkronista ja täysin päätelaitteen ajastamaa. Tällöin päätelaitteelle lähetettävää dataa on tarve puskuroida verkon päässä odottamassa päätelaitteen vastaanottoikkunan avautumista. Näin ollen A-luokan laitteet ovat kaikkein vähiten tehoa kuluttavia LoRaWAN-päätelaitteita ja niiden vastaanottoviive on vastaavasti kaikkein pisin. (LoRa, n.d.) Kuva 3 esittää eri päätelaitekategoriat kommunikointiviiveiden ja akun eliniän suhteen.

Kuva 3. LoRaWAN-verkon päätelaitekategoriat (LoRa, n.d.)



LoRaWAN-verkon paikannusteknologia perustuu TDOA-menetelmään (Time Difference of Arrival), jossa vastaanotin mittaa kahden signaalin saapumisajan erotuksen (Lappi, 2020, s. 33). Tyypillinen LoRaWAN-päätelaitteen paikannustarkkuus on yhdestä kilometristä aina kahteen kilometriin asti. Paikannustarkkuus saadaan jopa sadan metrin tarkkuuteen, jos yhdyskäytäviä on saavutettavissa riittävän tiheästi. Harvaan asutuilla seuduilla tämä ei kuitenkaan tällä hetkellä Suomessa toteudu. Paikannus LoRa-verkossa on myös riippuvainen päätelaitteen aktiivisuudesta. Päätelaitteen sijainti voidaan mitata ainoastaan sen aktiivisena

aikana, joiden väli A-luokan päätelaitteilla voi olla jopa tunti tai enemmän. Myös liikkuminen verkossa on riippuvainen päätelaitteen aktiivisuudesta, sillä kytkeytyminen reitittimin tapahtuu niin ikään vain aktiivitulassa.

Tulevaisuudessa tämä teknologia voi olla hyvä korvaaja GNSS-paikannukselle sovelluksissa, joissa riittää sadan metrin paikannustarkkuus (Lappi, 2020, s. 66). Suomessa kaupallisia LoRaWAN-verkkoja tarjoavat muun muassa Digita. LoRaWAN-verkon kattavuus on Suomessa erittäin kattava, täten LoRaWAN-verkko onkin monessa IoT-sovelluksessa erittäin varteenotettava vaihtoehto. Kuva 4 esittelee LoRaWAN-verkon kattavuuden Suomessa syksyllä 2020.

Kuva 4. Digitan LoRaWAN-verkon kattavuuskartta Suomessa (Digita, n.d.)



Toinen tunnettu LPWAN-ryhmän verkko on Sigfox. Sigfox on samannimisen ranskalaisen yrityksen hallinnoima ja patentoima esineiden internetin teknologia. Sigfox käyttää lisenssivapaita alle gigahertsin radiotaajuuksia ja perustuu UNB-modulaatioon (Ultra Narrow Band). Sigfox on avoin standardi, jolloin kuka vaan voi valmistaa Sigfox-laitteita, mutta itse verkon käyttö on maksullista (Järvinen, 2017, s. 15). Sigfox-teknologian merkittävimpiä hyötynä ovat pitkät kantamat järjestelmän osien välillä. Parhaimmillaan hyvissä olosuhteissa järjestelmän osien välimatkat voivat olla jopa 50 kilometriä (Raza, Kulkarni, Sooriyabandara, 2017, s. 7).

Sigfox on myös kaksisuuntainen IoT-optimoitu verkko, mutta Sigfox-verkossa lähetettävien pakettien hyötykuorma on kuitenkin vain 0–12 tavua. Lisäksi Sigfox-verkoissa voi olla päiväkohtaisia rajoituksia, joilla rajoitetaan niin lähetettävien kuin myös vastaanotettavien pakettien määrää. Lähetettävien pakettien maksimimäärä on 140 viestiä päivässä ja vastaanotettavien viestien määrä on rajoitettu vain neljään viestiin päivässä (Järvinen, 2017, s. 16). Vastaanotettavien viestien rajoitettu määrä estää näin ollen kuittausviestien käytön Sigfox-verkossa.

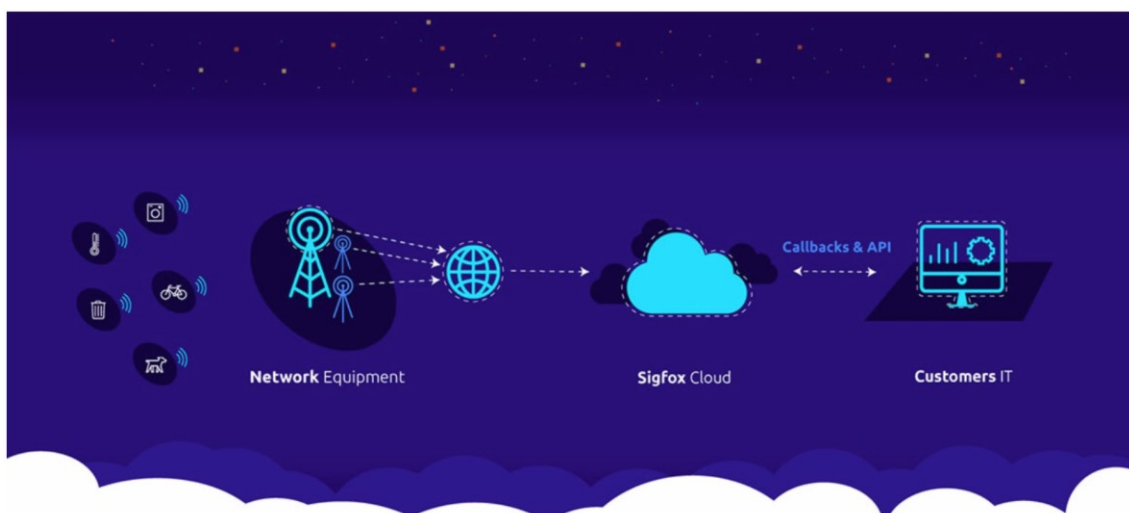
Koska Sigfox-verkon yhdyskäytävä kuuntelee kaikkia kanavia yhtä aikaa, päätelaite voi lähettää lähetyksensä Sigfox-verkossa satunnaisesti valitsemallaan taajuudella. Lähetyksen perillemenon varmistamiseksi sama lähetyksen paketti lähetetään vielä kahdesti uudelleen, mutta eri taajuuksilla (Kontu, 2020, s. 15). Tällöin ei kuittausviestejä tarvita Sigfox-teknologiassa, vaan voidaan olettaa, että viesti saavuttaa kuuntelijansa monikanava monilähetykseen perustuen. Sigfox-teknologiassa päätelaite aukaisee yhteyden, jolloin sovellusohjelmisto voi lähettää dataa päätelaitteelle vain silloin, kun päätelaite sitä pyytää. Sigfox ei oletusarvoisesti käytä salausta, mutta käyttäjä voi tarvittaessa aktivoida oman salauksensa viesteihin. Sigfox-päätelaitteet eivät ole suoraan yhteydessä internettiin, joten se luo tiettyä suojaa päätelaitteelle (Lappi, 2020, s. 21). Päätelaitteiden lähetämät viestit siirretään yhdyskäytävien kautta Sigfox-pilvipalveluun, josta ne ovat haettavissa loppukäyttäjäsovellukselle. Kuva 5 esittelee Sigfox-verkkotopologian

Sigfox-teknologia tukee paikantamista. Paikantaminen Sigfox-verkossa perustuu RSSI-menetelmään (Received Signal Strength Indication), jossa seurataan signaalinvoimakkuutta (Sigfox build, n.d.-a). RSSI-menetelmässä radiosignaalin kulkema matka voidaan laskea

perustuen menetelmään, jossa radiosignaali vaimentuu käänteisen neliölain mukaisesti. Tätä menetelmää käytetään tyypillisesti sisäpaikannusmenetelmissä esimerkiksi WLAN-verkoissa. Paikantaminen RSSI-menetelmällä on kuitenkin varsin virheherkkää erilaisten esteiden ja muiden signaaliin virhettä tuottavien ilmiöiden takia. Varsinkin pitkillä kantamilla virheiden mahdollisuus kasvaa ja täten paikantamisen tarkkuus heikkenee. Mitä useampi tukiasema on saavutettavissa, sitä parempi tarkkuus voidaan saavuttaa (Pirainen, 2018, s. 10).

OTA-päivitykset eivät ole mahdollisia Sigfox-järjestelmässä vastaanottorajoituksen myötä, joka rajoittaa päätelaitteen vastaanottoviestien maksimimääräksi neljä vastaanottoviestiä päivässä.

Kuva 5. Sigfox-verkkotopologia (Sigfox build, n.d.-b).



Sigfox-verkkoa Suomessa hallinnoi Connected Finland. Sigfox-verkko Suomessa, tätä opinnäytetyötä kirjoitettaessa, kattaa vain ruuhka-Suomen (Liite 5). Täten haja-asutus alueiden kattavuus on hyvin vajaa, joka heikentää verkossa tehtävää paikannusta haja-asutusalueilla sekä rajoittaa päätelaitteiden käyttöä harvaan asutuilla alueilla. Liite 5 esittelee Sigfox-verkon kattavuuden Suomessa syksyllä 2020.

Yksi LPWAN-ryhmän verkoista on myös Weightless. Weightless sisältää joukon erilaisia standardeja, jotka ovat Weightless SIG-etujärjestön hallinnoimia. Standardit ovat kuitenkin avoimia kaikille. Tosin Weightless-laitteiden valmistajien on kuuluttava etujärjestöön saadakseen lisenssin valmistaa laitteita ja saadakseen uusimmat standardit. Weightless-

standardit jakaantuvat tällä hetkellä kolmeen eri standardiin: Weightless-W, Weightless-N ja Weightless-P (Link Lab, 2015). Alun perin Weightless-protokollan suunnitteluperiaatteena on ollut, että laitteet voidaan valmistaa edullisesti ja joiden toiminta-aika voi olla vuosia. (Järvinen, 2017, s. 19)

Vanhin Weightless-standardeista on Weightless-W. Weightless-W hyödyntää maanpäällisten televisiolähetysten vapaita taajuuksia. Weightless-W-tukiasemat saavat eri tietokantojen kautta tiedon vapaista kanavista taajuuksilla ja hyödyntävät näitä taajuushyppely tekniikkaa apuna käyttäen päätelaitteen ja tukiaseman välillä. Lähetyksessä signaali pilkotaan osiin ja lähetetään eri taajuuskanavia pitkin halutuin aikavälein. Tällä tekniikalla, jossa lähetettävä viesti pilkkoutuu eri kanaville ja eri aikajaksoihin, parannetaan häiriönsietoa lähetyksessä. Koska eri maissa vapaat TV-kanavat ja täten myös kanavia koskevat määräykset ja säännökset vaihtelevat hyvinkin paljon, on tämä tekniikka todettu vaikeaksi toteuttaa globaalisti. Sama päätelaite ei välttämättä voi toimia eri paikoissa riippuen jo olemassa olevista taajuusalueista sekä niihin liittyvistä vaihtelevista määräyksistä. Toisin sanoen liikkuvuus Weightless-W verkossa voi olla jopa mahdotonta toteuttaa. On myös todettu, että on haastavaa suunnitella edullinen laite, joka voisi toimia samanaikaisesti sekä 400MHz ja 700MHz taajuuksilla. Paikallisesti Weightless-W-päätelaite kuitenkin voi olla hyvinkin isoja datamääriä mahdollistava järjestelmä. Weightless-W-standardi on kaksisuuntainen, näin ollen mahdollistaen myös datan siirron päätelaitteelle päin. Periaatteessa tämä mahdollistaa myös OTA-päivitykset päätelaitteelle. (Järvinen, 2017, s. 19)

Toinen Weightless-ryhmän standardi on Weightless-N. Weightless-N on niin ikään Weightless SIG-järjestön luoma, mutta vain yksisuuntaisen kommunikation mahdollistava standardi. Weightless-N hyödyntää myös lisenssivapaita alle gigahertsin ISM-taajuuksia. Weightless-N voi myös hyödyntää lisensoituja taajuuksia tarvittaessa. Järjestelmä käyttää UNB-teknologiaa (Ultra Narrow Band). Weightless-N mahdollistaa päätelaitteen liikkuvuuden verkossa tukiasemien ohjaamana, riippumatta siitä kuka verkon omistaa. Kantavuus verkossa voi olla hieman yli viisi kilometriä ja akun kestoksi arvioidaan noin kymmenen vuotta. Weightless-N-verkkoa ei kuitenkaan löydy Suomesta vielä ja kattavuus Weightless-N-verkoilla on globaalistikin hyvin rajoittunut. Koska Weightless-N on yksisuuntainen verkkoprotokolla, ei myöskään päätelaitteen sovelluksen etäpäivitykset ole mahdollisia tässä verkossa. (Järvinen, 2017, s. 20)

Uusin Weightless-standardeista on Weightless-P. Se hyödyntää, niin ikään, alle gigahertsin ISM-taajuuksia. Weightless-P on edeltäjästään poiketen kaksisuuntaiseen kommunikaatioon pohjautuva järjestelmä, mahdollistaen täten myös OTA-päivitykset päätelaiteeseen. Weightless-P myös tukee virheenkorjausta sekä viestin uudelleen lähetystä, joilla parannetaan järjestelmän viestinlähetyksen luotettavuutta. Kantama tässä järjestelmässä voi olla parikin kilometriä järjestelmän osien välillä. Akun kestoksi arvioidaan kolmesta kahdeksaan vuotta (Järvinen, 2017, s. 21). Weightless-P ei myöskään mahdollista päätelaitteen liikkuvuutta verkossaan (Dali, Mahbubur, Abusayeed, 2018).

2.2.2 Matkapuhelinverkot

Matkapuhelinverkot, kuten 2G, 3G, 4G ja 5G, mahdollistavat kattavuudeltaan ja liikkuvuudeltaan parhaat edellytykset liikkumiselle verkon alueella. Kääntöpuolena on IoT-laitteille optimoimaton verkko. Tämä voi aiheuttaa suurempaa energian kulutusta mittaavassa päätelaitteessa sekä isompaa datasiirtomäärää verkon yli. Toistaiseksi toisen sukupolven matkapuhelinverkon kattavuus on maailmanlaajuisesti vielä paras. Liite 2 esittää toisen ja kolmannen sukupolven matkapuhelinverkkojen kattavuuden DNA-verkossa Suomessa. Vastaavasti liite 3 esittää neljännen ja viidennen sukupolven matkapuhelinverkkojen kattavuuden Suomessa. Vaikkakin vanhimpia toisen sukupolven matkapuhelinverkkoja ollaan maailmanlaajuisesti ajamassa alas tulevina vuosina, niin uudempien matkapuhelinstandardien mukaisten verkkojen myötä, verkon laaja kattavuus on taattu matkapuhelinjärjestelmissä kansallisilla säännöksillä. Tällöin vanhan teknologian poistuessa, uudemmilla teknologioilla saavutetaan koko maan kattava verkon peitto.

Viime aikoina julkistetut uudet matkapuhelinverkkostandardit ovat kuitenkin tuoneet mahdollisuuden liittää matkapuhelinverkkoihinkin pientä virrankulutusta ja kevyttä dataliikennöintiä omaavia IoT-laitteita. Uusia standardeja ovat LTE-M ja NB-IOT. Molempiin verkkoihin löytyy jo kaupallisia päätelaitteita. Viime aikoina on alkanut tulla markkinoille myös laitteita, jotka pystyvät hyödyntämään molempia verkkoja edellä mainituista IoT-verkoista. Molemmat protokollat ovat tarjolla jo neljännen sukupolven matkapuhelinverkoissa (4G). (Domnique, 2020)

LTE-M (Long Term Evolution Machine Type Communication) on 3GPP:n määrittelemä standardi matkapuhelinverkkoihin liitettävälle IoT-laitteille. LTE-M tunnetaan myös lyhenteistä LTE- MTC ja LTC Cat M. LTE-M on hyvin pienien viiveiden omaava järjestelmä, joka tukee erittäin hyvin matkapuhelinverkon liikkuvuutta datayhteyden katkeamatta. LTE-M on myös globaali ja mahdollistaa verkkovierailun (roaming) verkkojen välillä (Domnique, 2020). Opinnäytetyötä ja sen vaatimuksia ajatellen, kirjoitusajankohtana LTE-M oli vielä kasvukeskuksien verkko eikä haja-asutusalueilla ollut riittävää kattavuutta. Verkonkattavuus kuvat DNA:n verkosta ovat esillä liitteen 4 sinisessä puolikkaassa.

NB-IOT on niin ikään 3GPP määrittelemä standardi paikallisille vähän virtaa kuluttaville IoT-laitteille. NB-IOT ei mahdollista liikkuvuutta verkossa. NB-IOT:n hyötyinä onkin kuuluvuus rakennusten sisältä paikoista, joista ei esimerkiksi matkapuhelin enää kuulu, sekä erittäin pieni energiankulutus päätelaitteissa. Päätelaitteen akun kesto voi olla parhaimmillaan jopa 10 vuotta. (Dominique, 2020). NB-IOT-verkonkattavuuskuvat DNA:n verkosta ovat esillä liitteen 4 oranssissa puolikkaassa.

2.2.3 Matkapuhelinverkkojen kattavuus DNA-verkossa

Suomessa toimivilla muilla mobiiliverkko-operaattoreilla, joilla on oma verkko, tilanne on hyvin saman tapainen kuin liitteissä 2–4 esitellyt DNA:n verkonkattavuuskuvat. Muita opinnäytetyön kannalta mainittavia, oman verkon omaavia mobiiliverkko-operaattoreita Suomessa ovat, Telia (Telia, n.d.) ja Elisa (Elisa, n.d.). Kaikki liitteiden 2–4 kuvat on otettu DNA:n palvelusta 11.10.2020 (DNA, n.d.).

Kuten liitteiden 2–4 kuvista voidaan helposti päätellä, perinteiset matkapuhelinverkot ovat vielä toistaiseksi kattavuudeltaan erinomaisesti Suomen alueet peittävä. Kuitenkin IoT-verkot kasvavat nopeaan tahtiin ja täten jo lähitulevaisuudessa voi IoT-verkon kattavuus saavuttaa perinteisen matkapuhelinverkon kattavuuden. IoT-verkoissa on kuitenkin verkon liikennöinti ja arkkitehtuuri optimoitu pienimääräiseen datan siirtoon, kun taas matkapuhelinverkot on optimoitu puhesignaalin ja isojen datamassojen mukaisesti.

2.2.4 Wireless Personal Area Networks

WPAN-tekniikoita (likiverkko) ovat esimerkiksi Bluetooth, Z-Wave, BLE (Bluetooth Low Energy), DASH7 ja Wifi. Nämä ovat hyvin paikallisia verkkoratkaisuja ja näin olleen ovat enemmänkin paikallisten ja paikallaan olevien sensorien ja yhdyskäytävän välisiä protokollia. Nämä teknologiat ovat parhaimmillaan paikallisissa ja paikallaan pysyvissä kohteissa. Näiden järjestelmien kuuluvuus on hyvä rakennusten sisällä ja siksi näitä käytetäänkin paljon rakennusautomaatiojärjestelmissä. Jotta näiden teknologioiden kanssa voitaisiin toteuttaa liikkuvuus koko Suomessa, tarvittaisiin yhdyskäytävä koko maan kattavaan verkkoon. Täten tämä työ ei käy näitä vaihtoehtoja sen tarkemmin läpi.

2.2.5 Mesh-verkot

IoT-Mesh teknologioihin perustuvia verkkoja ovat muun muassa Zigbee ja Wirepas (Wirepass, n.d.). Jotta näidenkin verkkojen kanssa pystyttäisiin toimimaan koko Suomen kattavasti, tarvittaisiin paikallisen mesh-verkon tueksi reititin, joka pystyy luomaan linkin järjestelmän päätelaitteen ja jonkin koko maan kattavan verkon välille. Mesh-verkot ovat niin ikään verkkoja, joita hyödynnetään paikallisissa järjestelmissä kuten rakennusautomaatiossa.

2.2.6 Verkkoteknologioiden vertailu

Verkkoteknologioiden vertailun helpottamiseksi koottiin taulukko (Taulukko 1), joka kokoaa yhteen erilaiset verkkojen perusvaatimukset. Verkkotekniikoita vertailemalla, potentiaalisimmiksi vaihtoehtoiksi opinnäytetyölle valikoituvat LoRaWAN sekä matkapuhelinverkoista 2G ja 3G. Koska yhdenkään teknologian oma paikannusmenetelmä ei mahdollista vaatimusten mukaista tarkkuutta, nämäkin teknologiat tarvitsevat lisälaitteen riittävän tarkkaan paikannukseen. Vaikka LoRaWAN-teknologia itsessään mahdollistaa paikannusta verkossa, se ei riitä vaadittuun tarkkuuteen vaan sekin tarvitsee erillisen lisälaitteen täyttämään järjestelmälle asetetut vaatimukset.

Taulukko 1. Verkkoteknologioiden vertailu

	LPWAN					Matkapuhelinverkot				Matkapuhelinverkkojen IoT-verkot	
	LoRaWan	Sigfox	Weightless-W	Weightless-N	Weightless-P	2G	3G	4G	5G	NB-IOT	LTE-M
Verkon kattavuus Suomessa	Digitan verkossa kattaa koko maan	Connected Finland verkossa, Osittainen	Ei kaupallista toteutusta	Ei kaupallista toteutusta	Ei kaupallista toteutusta	Osittainen noin 99 %	Osittainen noin 95%	Osittainen noin 85 %	Turku, Tampere, Helsinki	Osittainen noin 50 %	Osittainen noin 80 %
Mahdollistaa OTA-päivitykset	kyllä	ei	kyllä	ei	kyllä	kyllä	kyllä	kyllä	kyllä	kyllä	kyllä
Liikkuminen koko verkon alueella Suomessa	kyllä	Mahdollista, mutta vaikeaa toteuttaa	ei	kyllä	ei	kyllä	kyllä	kyllä	kyllä	ei	kyllä
Paikantaminen 10m tarkkuudella	ei ilman lisälaitetta	ei ilman lisälaitetta	ei ilman lisälaitetta	ei ilman lisälaitetta	ei ilman lisälaitetta	ei ilman lisälaitetta	ei ilman lisälaitetta	ei ilman lisälaitetta	ei ilman lisälaitetta	ei ilman lisälaitetta	ei ilman lisälaitetta
Hinta	Noin 2€/kk/anturi	Verkon käytöstä maksetaan paikalliselle operaattorille	ei tietoa	ei tietoa	ei tietoa	Operaattori-liittymä riippuvainen (HUOM: Particle-SIM Telian verkossa 3 € / kk sisältää 3 MB. Yli menevältä osalta 0.4 € / MB)	Operaattori-liittymä riippuvainen (HUOM: Particle-SIM Telian verkossa 3 € / kk sisältää 3 MB. Yli menevältä osalta 0.4 € / MB)	Operaattori-liittymä riippuvainen	Operaattori-liittymä riippuvainen	Operaattori-liittymä riippuvainen	Operaattori-liittymä riippuvainen

Järjestelmien käytön hintaluokat ovat hyvin lähellä toisiaan. Puhutaan muutamasta eurosta kuukaudessa yhteyttä kohden. Matkapuhelinverkkojen kuukausikustannukset riippuvat täysin liittymästä ja operaattorista. Halvimmat liittymät alkavat hieman alle kymmenestä eurosta. Poikkeuksena on Particle:n toimittama liittymä, jonka kuukausikustannus on kolme euroa kuukaudessa ja se mahdollistaa 3 megatavuun asti tiedon siirtoa tähän hintaan. Tämän yli menevältä osalta hinta on 0.4 euroa megatavu.

2.3 IoT-kehitysalustat

Opinnäytetyön kirjoituksen aikana ehti tapahtua paljon muutoksia IoT-kehitysalustojen osalta. Monet kehitysalustoista kokivat elinkaarensa lopun, joitain ostettiin kilpailijana pois ja joidenkin uusiutumistaajuus ei täyttänyt perusvaatimuksia. Siksi jäljelle jäivät sellaiset kehitysalustat, joiden saatavuus on ollut taattu jo vuosien ajan sekä omaavat laajan kehittäjäverkoston.

Kehitysalustoja rajasi pois myös päätös perustaa järjestelmä GNSS-paikannukseen, jolla pytyttäisiin toteuttamaan paikannusvaatimukset täyttävä järjestelmä. Myös päätös valita matkapuhelinverkkoihin perustuva järjestelmä, teki omat rajauksensa valintoihin. Yhtään kaikki merkittävät vaatimukset täyttävää kehitysalustaa ei löytynyt valmiina. Jokaisen osalta oli tarve lämpötila- ja kosteusmittaukseen, joko paikannusmahdollisuuden lisäämiseksi tai sekä paikannus- että kommunikaatio-ominaisuuden lisäämiseksi. Joidenkin kehitysalustojen osalta myös oman

akun latauksen osalta oli tarve toteuttaa se erillisellä latauspiirin sisältävällä lisälevyllä. Teknologioiden valinnasta kerrotaan tarkemmin luvussa 3.2. Liite 1 kokoaa yhteen kehitysalustojen osalta vertailuun tarvittavat parametrit.

Koska IoT-kehitysalustoja julkistetaan hyvin nopeaan tahtiin, ei tämä työ kata kaikkia erilaisia mahdollisuuksia. Edellä mainitut ovat kuitenkin jo jonkun aikaa olleet markkinoilla, joten niiden kypsyyssaste oli todettu hyväksi ja näin ollen voisivat kaikki olla turvallisia valintoja.

Työn vertailussa olivat mukana seuraavat kehitysalustat lisälevyineen:

1. Raspberry Pi 3-kehitysalusta (Raspberry, n.d.)
Adafruit FONA 808-lisälevyllä (Adafruit, n.d.-a).
2. Arduino UNO-kehitysalusta (Arduino, n.d.-a)
RAK Wireless WisLink Cellular BG96 Shield-lisälevyllä (RAK2011, n.d.).
3. ARDUINO MKR GSM 1400-kehitysalusta (Arduino, n.d.-b)
Adafruit Ultimate GPS FeatherWing-lisälevyllä (Adafruit, n.d.-b).
4. Particle Electron-kehitysalusta
Asset tracker-lisälevyllä (Ihmevekotin, n.d.).
5. Adafruit Feather 32u4 FONA-kehitysalusta (Adafruit, n.d.-c)
Adafruit Ultimate GPS FeatherWing-lisälevyllä (Adafruit, n.d.-b).
6. LoPy4-kehitysalusta (Pycom, n.d.-a)
Tracker-lisälevyllä (Pycom, n.d.-b)

Niin verkkoteknologiat kuin itse kehitysalustat saavat uusia tulokkaita sekä toimintoja aina vaan nopeammassa tahdissa. Myös kehitysalustojen tarjoamat pilvipalvelut kehittyivät ja muuttuivat opinnäytetyön tiedonetsinnän aikana hyvin paljon. Opinnäytetyön tiedonkeruuaikana (kevät 2019 – syksy 2020) joitain potentiaalisia kehitysalustoja vaihtoi omistajaa ja näin ollen pudotettiin pois alkuperäiseltä valintalistalta, koska kyseessä oli pääsääntöisesti kilpailevan teknologian ostaminen pois markkinoilta. Myös joidenkin kehitysalustojen tuki loppui opinnäytetyön tiedonetsimisjakson aikana.

2.3.1 Raspberry-Pi

Raspberry-Pi on pitkän historian omaava kehitysalusta. Raspberry-Pi:tä voi käyttää hyvin laajasti erilaisiin sovelluksiin. Raspberry-Pi:lle löytyy omia Linux-distribuutioita, esimerkiksi raspbian, joka mahdollistaa laitteen helpon käyttöönoton sekä laajan kehittäjätuen. Myös Raspberry-Pi:lle löytyy hyvin paljon esimerkkejä valmiina erilaisiin tarpeisiin. Raspberry-Pi on kuitenkin enemmänkin minitietokone kuin IoT-kehitysalusta. Raspberry-Pi:stä löytyy monta erilaista kehitysalustaversiota. Opinnäytetyön tavoitetta ajatellen ei ole merkityksellistä minkä version kehitysalustasta valitsisi, sillä lisäominaisuudet eri versioiden välillä ovat enemmänkin Raspberry-Pi:n CPU ja GPU -tehon sekä ulkoisen kommunikaation hyödyntämisessä. Täten IoT-tyyppisiin optimointeihin liittyviä lisäominaisuuksia Raspberry-Pi:stä ei niinkään löydy valmiina. Kuitenkin Raspberry-Pi:llä voi tehdä hyvin toimivan IoT-sovelluksen, mutta se voi vaatia aika paljon laitteen ja käyttöjärjestelmän optimointia itse sovelluksen lisäksi.

Raspberry-Pi:lle on erittäin laaja ja kattava kehittäjätki ja sille löytyy paljon valmiita esimerkkejä hyödynnettäväksi. Myös hyvin kattavan määrän erilaisia sensoreita voi käyttää Raspberry-Pi:n kanssa. Raspberry-Pi sisältää erittäin laajan määrän erilaisia liityntöjä, kattaen kaikki vaatimuksissakin esitellyt väyläratkaisut sekä tarvittavat tulot ja lähdöt. Raspberry-Pi:stä löytyy siis I2C-väylä sekä keskeytykselliset tulot, joita voi konfiguroida halutun reunamuutoksen mukaiseksi. (Wei-Meng & Clarence, 2020)

Myös pilvipalvelun osalta, Raspberry-Pi:n kanssa voi käyttää esimerkiksi Google-pilvipalveluita, kuten esimerkiksi Firebase Cloud Messaging-menetelmää asiakaspäätelaitteelle (Firebase, 2020). Ohjelmointikielinä voi käyttää hyvin erilaisia kieliä, koska käyttöjärjestelmä perustuu Linux:iin. Näin ollen C, C++, Python sekä shell-scriptit ovat laajasti käytettävissä.

Kuten Eystein Stenberg toteaa, OTA-päivitykset Raspberry-Pi:hin on mahdollista tehdä ulkoisen palvelun avustuksella tai sftp-yhteydellä suoraan itse laitteeseen (Stenberg, 2020). Erittäin mielenkiintoinen kehitysalusta on Raspberry Zero, joka on muotoilultaan ja suunnittelultaan minimoitu kehitysalusta (Raspberry Zero, n.d.).

Raspberry-Pi vaatii lisäksi erillisen paikannus- ja kommunikaatiolisälevyn. Tällöin kokonaishinta alkaa nousta aika korkeaksi. Lisäksi Raspberryllä ei ole valmista tehonhallintapiiriä, joka mahdollistaisi usean tehosyöttölähteen ja niiden välisen hallinnan. LiPo-akun hyödyntämiseen tarvitaan erillinen latauspiiri, joka huolehtii akun latauksesta ja estää akun yllilatautumisen. LiPo-akku on erittäin herkkä syttymään vahingoituessaan tai yllilatautuessaan. Sen vuoksi niissä järjestelmissä, missä ei ole omaa sisäänrakennettua tehon latauksen hallintapiiriä on syytä käyttää erillistä ulkoista piiriä akun ja kehityslaudan välissä.

2.3.2 Arduino

Arduino on niin ikään halpa minitietokone tyyppinen kehitysalusta. Arduino:lla on myös erittäin laaja kehittäjätki suurine määrine valmiita esimerkkejä. Arduino:n tapauksessa on kaksi mahdollisuutta valmistaa IoT-tyyppinen alusta.

Voidaan käyttää perinteistä Arduino Uno-kehitysalustaa lisälevyn kanssa, joka mahdollistaa sekä kommunikaation sekä paikannuksen. Arduino Uno:ssa on tarjolla vaatimusten mukaiset väylät, lähdöt ja tulot. Arduino Unon tehonsyötön voi järjestää usealla tavalla. Joko MicroUSB:n kautta tai erillisellä muuntimella omaan tehonsyöttöliittimeen. Omaan tehonsyöttöliittimeen voi tuoda 7–12 voltin jännitteen, josta Arduino:n oma jännitteentasaaja tasoittaa jännitteen piireille viiteen volttiin. Lisäksi levytä löytyvät tulot Vin, joihin voi syöttää 7–12 voltia tasasähköä. (Arduino, n.d.-a).

Toinen vaihtoehto on käyttää IoT-optimoitua ARDUINO MKR GSM 1400-kehitysalustaa Adafruit Ultimate GPS FeatherWing-lisälevyn kanssa. Tämä Arduino-versio itsesään sisältää mobiiliverkkoyhteysmahdollisuuden ja lisälevy tuo lisäksi kaivatun paikannusoption. Myös tässä Arduino-versiossa on vaaditut väylät, lähdöt ja tulot tarjolla. MKR 1400-versio mahdollistaa samat tehonsyöttömahdollisuudet mitä perinteisempi UNO-versiokin. (Arduino, n.d.-b).

Molempiin Arduino versioihin voi tuoda, joko viiden voltin tehonsyötön ja käyttää LiPo-akkua rinnalla. Tosin LiPo-akun käytössä kannattaa hyödyntää edelleen lisälevyä, joka huolehtii LiPo-akun latauksesta ja estää sen yllilatautumisen (OZRobotics, n.d.). OTA-päivitykset

Arduino:n tuotteisiin mahdollistuvat tallettamalla halutun päivitystiedoston SD-kortille, vaikka sftp-yhteyden yli, sekä uudelleen käynnistämällä kehitysalustan. Arduino:n ohjelmointikielenä ovat C ja C++.

Arduino mahdollistaa useankin erillisen pilvipalvelun käytön. Mahdollisuuksia ovat:

1. Blynk (Blynk, n.d.).
2. Google IoT Cloud (Google Cloud, n.d.).
3. Soracom Air IoT (Soracom, n.d.).
4. Arduino Cloud (Arduino Cloud, n.d.).

Helppokäyttöisin edellisistä on todennäköisesti Arduino:n oma pilvipalvelu, sillä Arduino:n laiteohjelma sisältää suoraan komennot sen käyttöön. Arduino sisältää myös ECC-508 crypto-piirin, jolla päätelaite ja pilvipalvelun välinen liikenne salataan.

2.3.3 Particle Electron

Particle on vuosia tehnyt erilaisia IoT-tarkoitukseen optimoituja kehitysalustoja. Particle:lta löytyy hyvin laaja tarjonta erilaisiin kommunikaatorajapintoihin olevia kehitysalustoja. Lisäksi Particle:lla on ollut usean vuoden ajan tarjolla pilvipalvelu. Pilvipalvelu mahdollistaa myös useiden laitteiden hallinnan ja sitä kautta niin sanotun laivueen hallinnan (Fleet management). (Particle Cloud, n.d)

Particle laiteohjelma (firmware) on kaikille eri Particle:n tuotteille pääosin sama ja näin ollen mahdollistaa tietyissä rajoissa saman sovelluksen siirtämisen toiseen tuoteversioon. Particle:n laiteohjelma on Particle:n hallitsema, jolloin kehitysalustan käyttäjä voi keskittyä itse sovelluksen tekoon (Particle Device OS, n.d.). Kattavan pilvipalvelun lisäksi, laiteohjelma sisältää OTA-päivitysmahdollisuuden sekä laajat virransäästöominaisuudet. Nämä virransäästöominaisuudet mahdollistavat laitteen nukuttamisen hyvin pitkäksi aikaa ja herättämisen joko ajastimella tai ulkoisen keskeytyksen perusteella (Kumar, 2016). Particle:n laiteohjelma on myös optimoitu verkonkättelyiden osalta, kun käytetään Particle:n toimittamaa SIM-korttia. Käyttämällä sovelluksessa moodia SLEEP_NETWORK_STANDBY, dataliikenne pystytään minimoimaan päätelaitteen ja verkon kättelyiden osalta. Kyseisellä

moodilla virransäätötilan maksimiajaksi voidaan asettaa jopa 23 minuuttia ilman, että tarvitaan uutta kättelyä verkon ja päätelaitteen välillä. (Particle Docs, n.d.-a)

Koska saman sovelluksen ajaminen eri Particle:n tuotteissa on tietyissä rajoissa mahdollista, Particle:n kehittäjätki on erittäin laaja ja löytyy paljon valmiita esimerkkejä erilaisiin tarpeisiin. Particle:lla on myös erittäin hyvät ja kattavat dokumentaatiot jokaisesta eri tuotteesta. Particle:n tuotteille löytyy vielä erittäin hyvin kirjastotuki esimerkiksi I2C-toimilaitteille, jotka on ensisijaisesti tehty Arduino:lle tai muille kehitysalustoille. Saman kirjaston pystyy kääntämään eri käännöslipulla Particle:n tuotteille sopiviksi. Tämä helpottaa erittäin paljon Particle:n tuoteperheen käyttämistä ja laajentamista myös muiden tuoteperheinen lisälaitteilla.

Particle Electron, Particle:n tuotevalikoimasta, on varsin kattava kehitysalusta. Electron itse sisältää mobiiliverkon kommunikaatioyhteyden ja virranhallintapiirin (Particle Electron, n.d). Particle Electron sisältää laajasti erilaisia lähtöjä ja tuloja, kattaen liki kaikki päätelaitevaatimuksissa esitetyt vaatimukset. Paikannus ei ole sisäänrakennettuna Electron:iin, mutta se voidaan toteutettua erillisellä Asset Tracker-lisälevyllä, joka tuo GNSS-vastaanottimen lisäksi kiihtyvyyssanturin toiminnot kokonaisuuteen. Kiihtyvyyssanturia voidaan käyttää yhtenä keskeytystulona itse Electron:lle mahdollistaen virrankulutuksellisesti merkittävää optimointimahdollisuutta. GNSS-vastaanottimen päivitystaajuutta voidaan muuttaa perustuen kiihtyvyyssanturin tietoon. Paikannuspiirin luku voi olla erittäin harva silloin, kun on todettu kohteen pysyvän paikallaan. Kun taas kiihtyvyyssanturi antaa keskeytystiedon kontrollerille kohteen liikkumisesta, voidaan paikannuksen päivitystaajuutta muuttaa tiheämpään päivitykseen, jolla mahdollistetaan reaaliaikainen kohteen seuranta. Kiihtyvyyssanturilla voidaan myös toteuttaa muita varsin hyödyllisiä etäseurantasovelluksia asuntovaunu ja -auto sekä huvivene käyttöön. Particle Electron sisältää myös oman BQ24195-tehohallintapiirin, jolla itse päätelaite sovelluksensa kautta hallitsee energiasyöttöjä, mutta joita voi myös etäohjata tarvittaessa. Ohjelmointikielenä Particle:n tuotteille ovat C ja C++. (Particle Docs, n.d.-b)

2.3.4 Adafruit Fona

Adafruit tarjoaa erittäin laajan tuotetarjonnan IoT-kehitykseen. Adafruit-tuotteista voi rakentaa sellaisen yhdistelmän, mikä täyttää kulloisetkin vaatimukset. Tämän opinnäytetyön vertailuun otettiin kuitenkin mukaan yhdistelmä Adafruit Fona Ultimate GPS-lisälevyllä. Tällä yhdistelmällä saadaan Adafruit-tuotteilla toteutettua pienikokoinen HW-vaatimukset täyttävä kokonaisuus.

FONA itsessään sisältää riittävästi lähtöjä ja tuloja, mukaan lukien I2C-väylän sekä toki mobiiliverkkomodeemin (Adafruit, n.d.-c). Tehohallinnan suhteen FONA sisältää useita mahdollisuuksia syöttää tehoa. FONA:ssa on oma liitin LiPo-akulle ja oma liitin MicroUSB-lataukselle. FONA sisältää myös oman latauspiirin LiPo-akulle. Yksi selkeä haaste FONA:n kanssa löytyy kuitenkin tehosyötöistä. Jotta matkapuhelinverkon modeemi jaksaisi toimia korkeimpien kuormapiikkienkin aikana, olisi syytä pitää LiPo-akku myös jatkuvasti kytkettynä. Paikannusmahdollisuus FONA:n kanssa saadaan toteutettua Adafruit Ultimate GPS-lisäpiirilevyllä. Hakujen perusteella ei löytynyt tietoa onko FONA päivitettävissä etänä (OTA). Myöskään ei ole tietoa onko valmiiksi integroitua pilvipalveluja tälle yhdistelmälle.

2.3.5 LoPy4

LoPy4 lisättynä Tracker 2.0 X-lisälevyllä on potentiaalinen LoRaWAN-verkon vaihtoehto. LoPy4 on suunniteltu suoraan IoT-kehitysalustaksi, jolloin pääsääntöinen energialähde on akku. LoPy4 sisältää erittäin kattavat ja konfiguroitavat GPIO-liitynnät. LoPy4:stä löytyy myös keskeytystulot sekä I2C-väylä. Tehohallinnan suhteen LoPy4 on hieman vajavainen, sillä se sisältää vain 3,5–5,5 voltin syötön eikä omaa muita energiansyöttömahdollisuuksia. Jos haluaisi useita energiasyöttöjä LoPy4:ään niin tarvittaisiin erillinen energianhallintapiiri, minkä kautta erilaiset energiasyöttövaihtoehdot LoPy4:lle hallittaisiin. Pycom:in tuotteet käyttävät MicroPython kieltä ohjelmointiin, mikä on hyvin läheinen kieli Pythonin kanssa. LoPy4 mahdollistaa OTA-päivitykset

Tracker 2.0 X tuo järjestelmään toivotun GNSS-paikannusmahdollisuuden. Lisäksi Tracker 2.0 X sisältää kiihtyvyyssanturin, jota voisi käyttää Particle Electron Asset Tracker:n tapaan moniin erilaisiin sovelluksiin sekä käynnistämään GNSS-paikannuksen. Pilvipalveluina LoRaWAN-

verkossa voisi käyttää esimerkiksi LoRaCloud-palvelua (LoRa Cloud, n.d.). Digitan tarjoamaa kaupallista verkkoa voisi käyttää LoRa-verkkona. Kehittäjätki on laaja Pycom:n tuotteille ja paljon erilaisia sensoreita löytyy tällekin kehitysalustalle.

2.3.6 Kehitysalustojen vertailutaulukko

Liite 1 kokoaa yhteen taulukkoon kaikki edellisissä luvuissa kerrotut kehitysalustojen parametrit. Kaikki muut kehitysalustat ovat matkapuhelinverkkoihin pohjautuvia paitsi LoPy4, joka otettiin LoRaWAN-verkon myötä vertailuksi mukaan.

LoRaWAN-verkko on puhdas IoT-verkko, joten kehitysalustassa on vain yksi energiasyöttö. Täten oletuksena IoT-pohjaisilla laitteilla on, että laite toimii yhden akun varassa ja täten laite on syvässä virransäästötilassa suurimman osan ajastaan. Tämä tekee keskeytyspohjaisten sensoreiden käytöstä vaikeaa näiden puhtaiden IoT-kehitysalustojen kanssa. Nämä sopivat parhaiten mittauseräsuorittamiseen, jolloin keskeytystulot kannattaisi hoitaa muuta teknologiaa hyödyntäen.

3 Projektin suunnittelu ja toteutus

Järjestelmäsuunnittelu alkoi vaatimusten kirjoittamisella järjestelmälle ja sen osille. Vaatimukset on jaettu neljään eri osaan; päätelaitteelle, asiakaspäätelaitesovellukselle, pilvipalvelulle ja itse kokonaisjärjestelmälle.

Vaatimusten asettamisen jälkeen selvitettiin vaatimuksia parhaiten vastaavat teknologiat. Teknologiaavalintojen jälkeen tehtiin järjestelmäsuunnittelu, jossa järjestelmän eri osat piirrettiin yhteen kuvaan (Kuva 6). Järjestelmäsuunnittelun viimeinen vaihe oli HW-suunnittelu, jolla liitettiin anturit kehitysalustaan (Liite 6).

Kun järjestelmäarkkitehtuuri oli saatu valmiiksi, tilattiin kehitysalusta sekä muut osat pilotointia varten. Toteutuksen viimeisenä vaiheena oli ohjelmistojen suunnittelu ja tekeminen niin päätelaitteeseen kuin asiakaspäätelaitteeseen.

3.1 Systemivaatimukset

Järjestelmäsuunnittelu lähti liikkeelle suunnittelemalla raja-arvot ja vaatimukset kokonaisjärjestelmälle. Vaatimuksia kasaantui kaiken kaikkiaan 23 kappaletta, jotka jakaantuivat neljään eri kategoriaan; päätelaite, pilvipalvelu-, asiakaspäätelaite- ja järjestelmävaatimuskategorioiden.

Päätelaitekategoria listasi vaatimukset itse IoT-päätelaitteelle, joka sijoitetaan kohteeseen ja toimii kontrollerina mittaustapahtumalle sekä toimilaitteiden ohjauksille.

Pilvipalvelukategoria taas kasasi vaatimukset pilvipalvelulle ja sen toiminnallisuuksille.

Vastaavasti asiakaspäätelaitekategoria listasi vaatimuksia asiakaspäätelaitesovellukselle.

Koko järjestelmää koskevat vaatimukset on listattu järjestelmävaatimuskategoriassa.

Vaatimuksille asetettiin myös prioriteetit, jotta saataisiin merkittävimmät vaatimukset, niin osa-alueiden kuin myös koko järjestelmän osalta listattua. Syntyneiden vaatimusten pohjalta voitiin alkaa tehdä rajauksia tarjolla olevista laitteista, pilvipalveluista sekä kehitysalustoista.

3.1.1 Päätelaitteen vaatimukset

Merkittävimpinä vaatimuksina päätelaitteelle nousivat:

1. IoT-laitteen useampi tehonsyöttömahdollisuus, joita voi laitteen asiakaspäätelaitteella tarvittaessa ohjata ja muuttaa energian käyttölähdettä.
2. Laitteen laiteohjelman ja applikaation päivitykset etänä (OTA - Over the Air).
3. I2C-väylä
4. Analoginen tulo ja lähtö
5. Keskeytyspohjainen konfiguroitavissa oleva tulo.
6. Tarjolla olevat eri valmistajien sensorit ja niiden saatavuus, elinikä ja hinta.
7. Kehittäjä tuki.
8. Pitkä elinkaarituki.
9. Erilaisten tulevaisuuden verkkojen tuki.
10. IoT-laitteen akunkesto

3.1.2 Pilvipalvelun vaatimukset

Pilvipalvelun merkittävimmät vaatimukset olivat:

1. Kyberturvallisuus niin päätelaitteelle kuin myös asiakaspäätelaitteelle.
2. Pilvipalvelun integraatio päätelaitteeseen.
3. Palvelun saatavuus ja tavoitettavuus.
4. Päätelaitteen dynaaminen kontrollointi ja monitorointi pilvipalvelun välityksellä.
5. Kokonaishinta.
6. Valmiin pilvipalvelun valmiusaste ja mukautettavuus.
7. Laaja ja aktiivinen kehittäjätki.
8. Pitkä elinkaarituki.

3.1.3 Asiakaspäätelaitesovelluksen vaatimukset

Asiakaspäätelaitesovellukselle merkittävimpiä vaatimuksia olivat:

1. Monialustakehitys (cross-platform development), jolloin sama business-logiikka toteutuu eri alustoille.
2. iOS-tuki
3. Android-tuki
4. Pitkä elinkaarituki.
5. Kehittäjän omat mieltymykset.

3.1.4 Kokonaisjärjestelmävaatimukset

Kokonaisjärjestelmävaatimuksista merkittävimmät olivat:

1. Koko maan, mukaan lukien vesi- ja merialueet kattava verkko.
2. Paikannus, edellisillä alueilla, vähintään 10 metrin tarkkuudella.
3. Kokonaisjärjestelmän kautta energiakulutuksen optimointi päätelaitteessa.

4. Kokonaisintegraatio järjestelmässä asiakaspäätelaitteesta pilvipalvelun kautta päätelaitteeseen.

3.2 Valitut teknologiat

Teknologioiden valinnat olivat kuitenkin varsin helpot. Koska rajaus oli koko Suomen maa- ja merialueet, se rajasi jo monta verkkoteknologiaa pois. Myös paikannuksen suhteen oli varsin selvää, mikä ratkaisu verkkoteknologian ohella olisi paras. Kehitysalustojen suhteenkin valinta päätyi varsin selvästi yhteen tuotteeseen.

Koska teknologiavalinnat myös rajasivat mahdollisuuksia muilta kategorioilta, kapeni jokaisen valinnan myötä valintamahdollisuudet usealla alueella. Kun valinnat tehtiin järjestelmätasolta kohti järjestelmän sisempiä tasoja, helpottui alakategorioiden valinnat ylempien valintojen myötä. Taulukko 1 ja liitteen 1 taulukko muodostettiin ohjaamaan eri järjestelmän osien valintoja.

3.2.1 Verkkoteknologiavalinta

Tällä hetkellä kuitenkin kattavin ja luotettavin verkko Suomessa on nykyinen matkapuhelinverkko ja varsinkin toisen sukupolven verkko (2G). Lisäksi, koska vaatimuksina oli kattaa maa- ja merialueet sekä paikannus kyseisillä alueilla halutulla tarkkuudella, matkapuhelinverkkoihin pohjautuvat järjestelmät nousivat parhaiksi mahdollisuuksiksi. Myös kansalliset lait ja asetukset pakottavat matkapuhelinverkot koko maan kattaviksi ja hitaasti muuttuviksi. Opinnäytetyön kohteitahan olivat mökki lapissa, huvivene sekä asuntovaunu, opinnäytetyön kirjoittajalla on hyväkin kokemus siitä mikä verkko kuuluu hänen omistamassaan mökissä lapissa. Tällä hetkellä ja vuosia aikaisemmin, siellä on kuulunut vain Telian 2G ja sekin erittäin heikosti matkapuhelimella tarkasteltuna.

Näin olleen työssä päädyttiin 2G-verkon hyödyntämiseen. Toisen sukupolven matkapuhelinverkon (2G) kattavuus on Suomessa, ja pitkälti koko maailman laajuisesti, vakain ja kattavin. Vaikkakin toisen sukupolven matkapuhelinverkkojen alasajo on menossa maailmanlaajuisesti, niin perustamalla järjestelmä matkapuhelinverkkoihin perustuvaksi, voidaan myöhemmässä vaiheessa radiorajapintaa muuttamalla huolehtia tulevan

järjestelmän pitkäkestoisesta tuesta. Toisaalta käyttämällä 2G-verkkoa, laitteen energiankulutuksen sekä verkon yli siirrettävän datan, kannalta päädytään hienoiseen kompromissiin. IoT-optimoidut verkot sallisivat paremman teho-optimoinnin, mutta 2G-järjestelmällä saadaan parempi liikkuvuus verkossa ja verkonkattavuus. Kuitenkin kaikissa opinnäytetyön tarkkailtavissa kohteissa on tarjolla lisäenergiaa, ja lisäenergia on ollut myös oletuksena koko järjestelmän suunnittelulle. Lisäenergiana toimisivat joko asuntovaunun tai veneen akusto tai mökin aurinkosähköjärjestelmä akustoineen. Tällöin päätelaitteen energiankulutuksen merkitys järjestelmälle oli prioriteetiltaan alempi mitä liikkuvuus verkossa ja paikannustarkkuus. Koska keskeytyspohjaiset tulot päätelaitesovelluksessa eivät salli päätelaitteen kaikkein syvimmän virransäästömodin käyttöä, se tulee edellyttämään päätelaitteelle ulkoisen energialähteen käyttöä pitkäkestoisessa toiminnassa ja täten myös päätelaitteen omaa energiahallintaa.

3.2.2 Paikannusteknologiavalinta

Koska käytetyksi verkkoteknologiaksi tuli perinteinen matkapuhelinverkko, oli helppo päätyä GNSS-paikannuksen tarjoamiin mahdollisuuksiin. Verkkojen itsensä tarjoamat paikannusteknologiat eivät kuitenkaan täyttäneet tämän opinnäytetyön tarpeita ja vaatimuksia. Lisäksi GNSS-pohjaisia lisälaitteita ja jopa suoria integraatioita oli paljon tarjolla kehitysalustojen tarpeisiin. Tällöin paikannusteknologiavalinta ei ollut rajaamassa liikaa kehitysalustavalintaa.

3.2.3 Kehitysalustavalinta

Loppujen lopuksi oli helppo päätyä Particle:n kehitysalustoihin, sillä Particle mahdollistaa juuri kriittiset vaatimukset täyttävän kehitysalustan pilvipalveluineen. Koska verkkoteknologia ja paikannusteknologia oli valittu, niin tarvitsi enää punnita kehitysalustojen muita parametreja. Näitä olivat pilvipalveluintegraatio, tehonhallinta, teho-optimointi, verkon optimointi, käyttäjätuen laajuus, dokumentaatio sekä laitetuki ja etenemissuunnitelman tulevaisuuteen.

Verkkoteknologian valinnan jälkeen Particle:n kehitysalustoista Electron oli potentiaalisin vaihtoehto.

Particle Electron mahdollistaa:

1. sekä 2G että 3G -matkapuhelinverkon käytön
2. omaa erittäin laajan kehittäjä-tuen
3. Particle:n tuotteet käyttävät C-ohjelmointi kieltä, joka mahdollistaa erittäin tehokkaan ja resursseja säästävän päätelaitesovelluksen tekemisen.
4. Laitteeseen pystyy liittämään hyvin erilaisten väylien kautta erilaisia sensoreita, jolloin käytettävä sensorituki on hyvin laaja.
5. Optimointimahdollisuudet ovat erittäin hyvät sekä verkkoliikenteen että tehonkulutuksen kannalta.
6. Tehonsyöttömahdollisuudet ovat laajat. Kun Electron vielä sisältää oman tehonhallintapiirin niin tehonsyöttömahdollisuuksia ja akun latausmahdollisuuksia voi käyttäjä optimoida päätelaitetta tarpeensa mukaan.
7. Particle tarjoaa kattavan ja päätelaitteiden laiteohjelmaan suoraan integroidun pilvipalvelun, jota pystyy käyttämään niin monitorointiin kuin ohjaukseen. Pilvipalvelu mahdollistaa jo valmiiksi niin sanotun laivueen hallinnan.
8. Particle Electron mahdollistaa OTA-päivityksen niin päätelaitteen laiteohjelma kuin myös päätelaitesovelluksen osalta.
9. Asset Tracker-lisälevy Particle Electron:iin, sisältää myös kiihtyvyyssanturin, jolloin virransäästöllisesti GNSS-piiri voidaan aktivoida vasta kun on havaittu liikettä. Integroitu kiihtyvyyssanturi mahdollistaa myös muita käyttäjälle arvokkaita käyttäjätapauksia tuotteistamismielessä, kuten esimerkiksi asuntovaunuun etävesivaaka tai veneen kallistuman seuranta.
10. Particle:n pilvipalvelun käyttöön löytyy Xamarin (monikehitysalusta) nuget-kirjasto, jota hyödyntämällä pystytään C# Xamarin-ohjelmointialustaa käyttämään asiakaspäätelaitteohjelman kehitykseen.
11. Particle tarjoaa myös hyvän kehitysympäristön niin pilvipalveluna, että suoran integraation Microsoft Code IDE:en. Täten käytettävät työkalut ovat erittäin hyvin tuetut ja mahdollistavat hyvän debuggaus-mahdollisuuden pääteohjelmistoon.
12. Particle on laajasti käytetty alusta ja näin olleen valmiita ohjelmistokirjastoja löytyy muiden kehittäjien tekeminä. Myös joitain Arduino:lle tehtyjä kirjastoja

pystyy uudelleen käyttämään, kun käänösvaiheessa huomioidaan Particle:n tarjoamat header-tiedostot ja käänösliput.

Edellistä johtuen kehitysalustaksi siis valittiin Particle Electron ja siitä vielä Asset Tracker-valmispaketti. Asset Tracker-paketti sisältää itse Particle Electron:in lisäksi GPS-piirilevyn, joka mahdollistaa paikannuksen GPS-tarkkuudella. (Ihmevekotin, n.d).

Täten Particle Electron:in valitsemalla opinnäytetyössä tarvitsee tehdä ensimmäiset kehitysversiot itse päätelaitesovelluksesta sekä asiakaspäätelaitesovelluksesta. Pilvipalvelu tulee Particle:n toimittamana, joten sen osalta niin kyberturvallisuus päätelaitteen ja pilvipalvelun välillä ja itse pilvipalvelun tekeminen itse voitiin jättää opinnäytetyöstä pois. Näin ollen voitiin keskittyä itse järjestelmäsovelluksen pilotointiin ja päätelaitesovellusten pilotin tekemiseen. (Particle Cloud, n.d)

Particle Electron sisältää itsessään 2G ja 3G -mobiiliverkkotuen. Kuten edellä on jo todettu, toisen sukupolven matkapuhelinverkoja ollaan maailmanlaajuisesti ajamassa alas. Tällöin kuitenkin uudemmilla verkkoteknologioilla tullaan saavuttamaan sama verkon kattavuus jatkossa. Tästä huolehtivat kansalliset lait ja asetukset. Particle Electron tukee sekä toisen sukupolven kuin myös kolmannen sukupolven matkapuhelinverkkoja. Jatkossa myös IOT-verkot tulevat kattamaan koko maan ja tällöin pohdittavaksi tulee muiden verkkoteknologioiden käyttö. Tässä tapauksessa myös Particle:n muut tuotteet mahdollistavat pitkän elinkaaren kehitystä samalle sovellukselle. Particle:n tuotteet pääsääntöisesti perustuvat samaan systeemiarkkitehtuuriin, jolloin olemassa olevan päätelaitesovelluksen muokkaaminen toiselle Particle:n alustalle on varsin helppoa.

Valitussa Particle Electron-kehitysalustassa on myös oma virranhallintapiiri, jolla voidaan valita kohde, josta oman LiPo-akun lisäksi voidaan energiaa ottaa. Tällöin energiastrategiana voisi olla ensisijaisesti ulkoinen energia ja sen vajotessa alas, siirrytään oman LiPo-akun käyttöön. Particle Electron pystyy ottamaan energiaa MicroUSB-portin kautta, LiPo-akusta, tai ulkoisesta 12 voltin syötöstä. Kohteen ulkoisen akun jännitetasoa taas seurataan erillisellä I2C-mittauspiirillä. Kun kohteen ulkoinen energialähde vajoaa liian alas, sovelluksen on tarkoitus lähettää varoitus pilvipalveluun sekä sitä kautta loppukäyttäjälle. Tällöin järjestelmä myös vaihtaa energialähdettään sisäiseen LiPo-akkuun.

Jotta järjestelmän matkapuhelinverkon käytön tarve ja IoT-laitteen virrankulutus saataisiin minimoitua, on yhtenä perusvaatimuksena ja järjestelmäsuunnittelun perusajatuksena ollut pilvipalvelun hyödyntäminen päätelaitteiden ja itse IoT-laitteen välissä.

Laite voi näin, asetetuin väliajoin kirjata anturitietoa pilvipalveluun, josta se on edelleen jatkojalostettavissa asiakaspäätesovellukselle. Tällöin IoT-laitteen ei tarvitse koko ajan olla yhteydessä pilvipalveluun eikä asiakaspäätelaitteeseen. Tällöin päätelaite voi siirtyä virransäästötilaan ajaksi, jolloin ei ole tarvetta anturitietojen lähettämiseksi. IoT-laite voi herätä virransäästötilasta, asetetun ajastimen herättämänä tai jokin keskeytystuloista herättämänä. Päätelaitteelle lähetettävät ohjaukset voivat toimia samalla periaatteella, jolloin asiakaspäätesovellus kirjaa halutun ohjauksen uuden tilan pilvipalveluun ja päätelaite saa ohjauksikäskyn itselleen lukiessaan pilvipalvelun muuttujat oman voimassa olevan toimintaprofiilin mukaisesti.

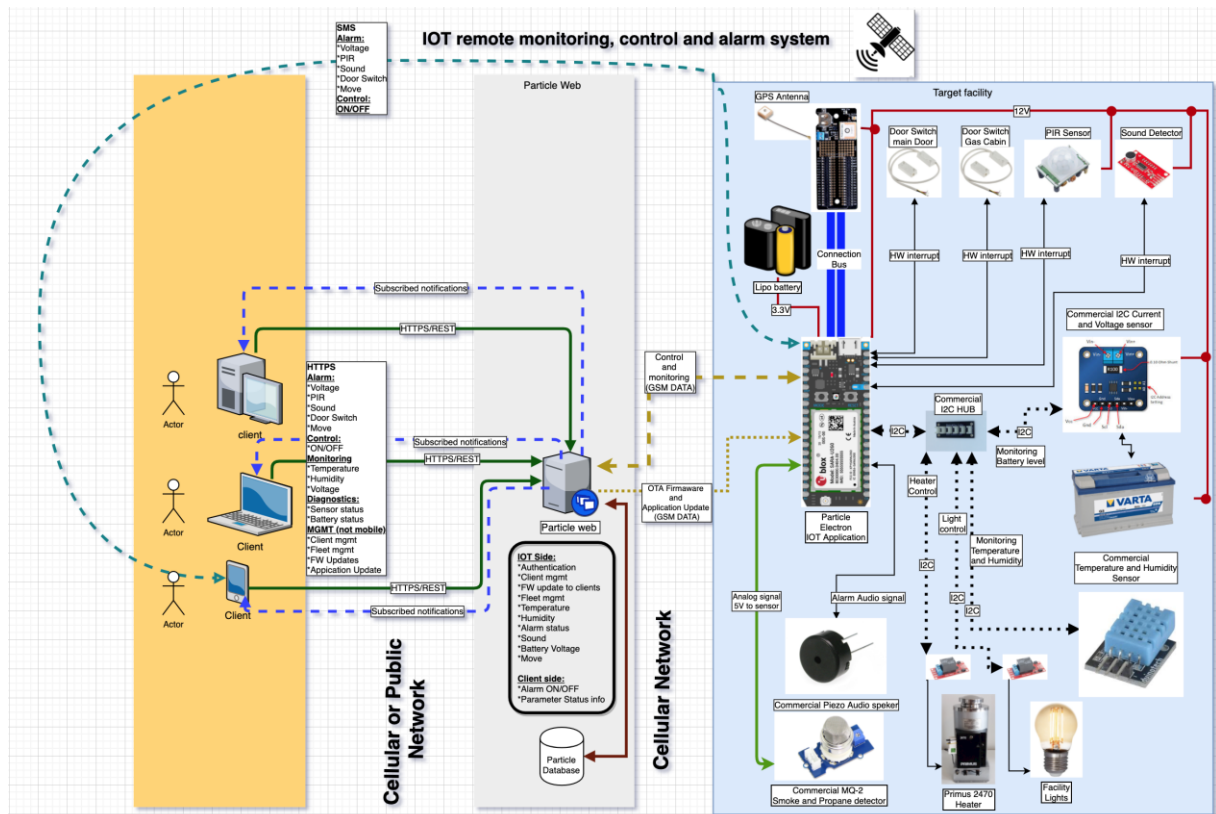
Particle:n tuotteet omaavat myös erittäin laajan kehittäjätuken. Tämä mahdollistaa estämään virheet, joita muut ovat jo tehneet. Tämä mahdollistaa myös nopean tuen ongelmissa, joita itse kohtasi. Koska Particle:n tuotteet tukevat myös suoraan kirjastoja, jotka ovat ensisijaisesti tehty muun muassa Arduino:lle, pystyi monia esimerkiksi I2C-väyläisiä Arduino:lle tehtyjä aktiivisensoreita ja niiden kirjastoja hyödyntämään myös tässä työssä. Tämä nopeutti huomattavasti pilottivaihetta.

3.3 Systemiarkkitehtuuri

Kun teknologiat oli valittu, systemiarkkitehtuurista piirrettiin yleiskattava kuva, eri rajapintoineen, eri osineen sekä sensoreineen. Systemiarkkitehtuuri on esitetty Kuva 6:ssä.

Järjestelmän anturitiedot jakaantuvat selvästi kahteen eri ryhmään. On nopeasti muuttuvaa anturitietoa ja hyvin hitaasti muuttuvaa anturitietoa. Hitaasti muuttuvia mittauskohteita ovat lämpötila, kosteus ja ulkoisen energialähteen varaustila. Nopeasti muuttuvia tietoja ovat PIR-, kaasu- ja ovianturit. Nopeita anturimuutoksia käytetäänkin hälytystiedon indikoimiseen keskeytystulojen kautta, kun taas hitaasti muuttuvia anturitietoja käytetään kohteen yleisen tilan seurantaan ja täten statistiikasta tehtäviin indikaatioihin ja varoituksiin.

Kuva 6. Järjestelmäarkkitehtuuri



Niin verkkoyhteyden datamäärän, että mittalaitteen akun säätämiseksi, pilvipalvelu toimii kokonaisjärjestelmässä välityspalveluna asiakaspäätelaitesovelluksen ja mittapäätelaitteen välillä. Eri päätelaitesovelluksen profiileissa mittaava laite käy profiiliin mukaisesti kirjoittamassa hitaan muutoksen parametritietonsa pilvipalveluun, jonka jälkeen päätelaite vaipuu takaisin energiansäästötilaan. Asiakaspäätelaitesovellukset voivat sitten pilvipalvelusta käydä lukemassa muutokset tarvittaessa. Jos asiakaspäätelaitesovellus on rekisteröinyt pilvipalvelun muuttujia seurantaansa, saa se push-ilmoituksen muuttujanarvon muuttuessa pilvipalvelussa. Kun päätelaite haluaa asettaa ohjattavia toimilaitteita päälle, asiakaspäätelaitesovellus käy kirjoittamassa pilvipalveluun tiedon kyseisen laitteen ohjaustietoihin. Mittapäätelaite päivittäessään omat parametritietonsa pilvipalveluun, lukee samalla muutokset etäohjattavien toimilaitteiden osalta ja kytkee muutokset voimaan. Kun muutos on kohteen toimilaitteelle tehty, kirjataan toimilaitteen tilamuutos pilvipalveluun indikoiden, onnistuiko kyseisen laitteen tilamuutos vai ei. Pilvipalvelu vastaavasti lähettää saman tiedon asiakaspäätelaitesovellukselle, mikäli asiakaspäätelaitesovellus on rekisteröinyt muuttuvan tiedon. Mikäli jokin ohjattavista laitteista kytkettiin päälle, mittaava laite asettaa automaattisesti itsensä normaalitoimintaprofiiliin. Tässä profiilissa mittaava

laite pystyy ottamaan komentoja vastaan kymmenen sekunnin viiveellä, jolloin ohjausviiveet saadaan riittävän lyhyeksi. Vastaavasti, kun ohjattavat laitteet kaikki kytketään pois päältä, mittaava laite vaihtuu takaisin virransäästöprofiiliin. Liite 7 havainnollistaa ohjaussekvenssin yleistä toimintaa.

3.3.1 Päätelaite-sensarit

Kokonaisjärjestelmäarkkitehtuurista poiketen pilottivaiheessa hyödynnettiin vain muutamia antureita. Ajatuksena oli varmistaa, että kyseistä tuloa ja lähtöä voidaan hyödyntää järjestelmäarkkitehtuurin edellyttämällä tavalla.

I2C-väylään kytkettiin seuraavat lisälaitteet:

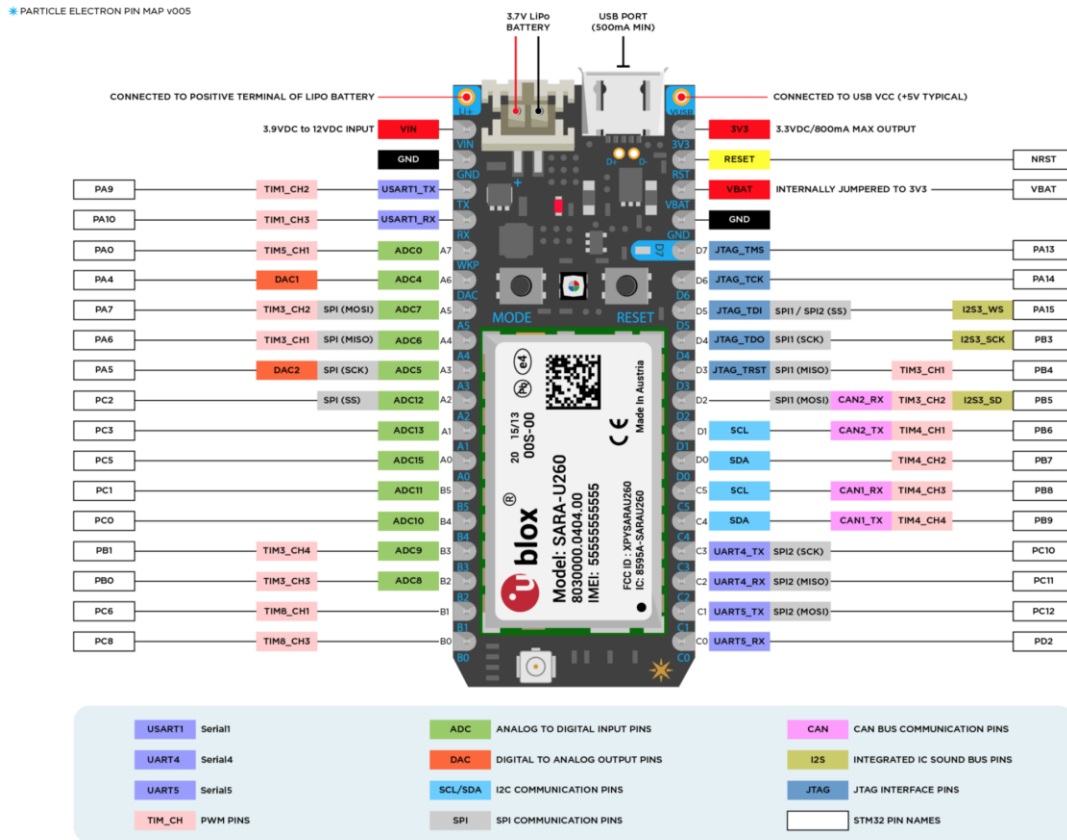
1. Ympäristösensori mittaamaan kohteen lämpötilaa, kosteutta, ilmanpainetta (Sparkfun, n.d.-a).
2. Jännite- ja virtamittauspiiri ulkoisen akun tilan seurantaan (Sparkfun, n.d.-b).
3. Kohteen toimilaitteiden ohjaukseen I2C-releitä. (Sparkfun, n.d.-c).

Pilottivaiheessa keskeytystuloina käytettiin:

1. PIR-sensari. (Sparkfun, n.d.-d).
2. Ovikytin. (SP-Electroniikka, n.d.).

Elektroniikka valittiin siten, että voitiin I2C-väylän sensorit johdottaa hyödyntäen SparcFun:in QWIIC-kaapeleita. Vastaavasti keskeytysantureiksi valittiin sensarityypit, jotka ovat normaalitilassa avoimessa tilassa. Kuva 7 esittää Particle Electron:in liityntänapat. Kun sensorin kytkin on avoimessa tilassa, järjestelmä ei tarvitse ulkoista energiaa ja järjestelmä voi olla vähän virtaa kuluttavassa valmiustilassaan. Vastaavasti reagointitilassa, sen kautta voidaan toimittaa kontrollerin keskeytystuloon käyttöjännitetaso. Particle Electron:in keskeytystulot ovat konfiguroitavissa tarpeen mukaan nousevaan reunaan, laskevaan reunaan tai muutokseen yleensä.

Kuva 7. Particle Electron:in liityntädiagrammi (Particle Docs, n.d-c, s.11).



Virransäästöllisesti keskeytystulojen kytkennässä käytetään 10kilo-ohmin alavetovastusta, joka pitää keskeytystulon maapotentiaalissa silloin kun järjestelmä on normaalitilassa ja keskeytyspiiri on avoin. Keskeytystilanteessa keskeytyspiiri sulkeutuu ja tällöin tuodaan käyttöjännitetaso keskeytystuloon. Tällöin virtapiirissä kulkee virta vain silloin, kun sensori on kytkeytyneenä ja keskeytyspiiri sulkeutuneena.

Particle Electron:in keskeytystuloina voidaan käyttää tuloja D5 ja D6 ja sovelluksessa tulon nousevaa reunaa voidaan käyttää herätteenä. I2C-sensorit liitettiin kaupallisilla QWIIC-kaapeleilla Particle Asset Tracker Grove-liitäntään. Particlen Grove-liitäntä on Particlen standardi I2C-kaapeloinnille, siinä missä QWIIC on standardi SparcFun:in tuotteiden I2C-liitynnöille. Liite 6 esittää keskeytystulojen sekä I2C-sensoreiden kytkennän Particle Electron:iin.

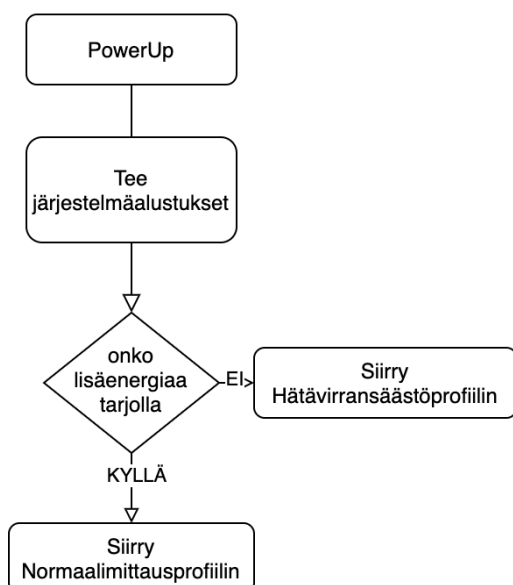
3.3.2 Päätelaitesovellus

Päätelaitesovellus on C-ohjelmointi kielellä tehty lyhyt sovellus. Käynnistyessään sovellus tekee ensin tarvittavat alustukset ja sen jälkeen alkaa suorittaa varsinaista profiilia. Jäljempänä olevat kuvat kertovat profiilien väliset suhteet sekä päätelaitesovelluksen normaalikiertokulun.

Päätelaitesovelluksella voi olla useita erilaisia toimintaprofiileja perustuen ulkoisten energialähteiden energiatasoihin tai kohteen hälytystulojen tilaan. Mikä vaan hälytystulo tai asiakaspäätelaitesovellus voi siirtää päätelaitteen hälytys- ja mittausprofiiliin. Alla on kerrottu tarkemmin eri profiilit ja niiden toimintaperiaate. Asiakaspäätelaitesovelluksella loppukäyttäjä voi halutessaan pakottaa päätelaitesovelluksen mihin tahansa profiiliin.

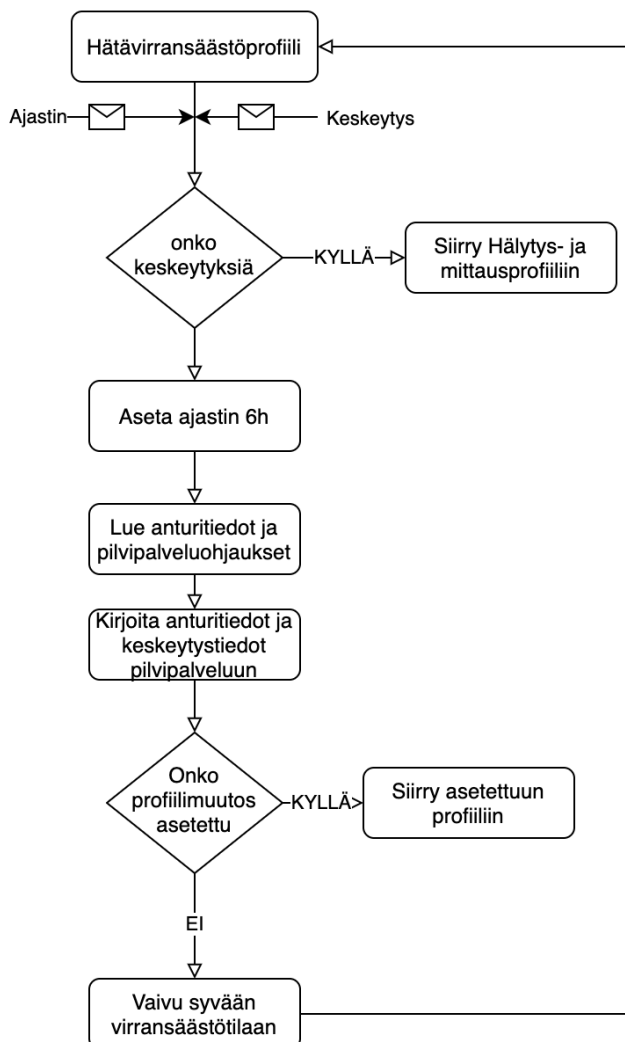
Kun päätelaitesovellus käynnistyy, niin ensin tehdään sovelluksen tarvitsemat alustukset ja sen jälkeen aletaan suorittaa varsinaista sovellusta. Ensimmäisenä tutkitaan, onko ulkoista energiaa tarjolla vai onko tarve käyttää Particle Electron:in omaa LiPo-akkua. Mikäli on ulkoista energiaa tarjolla, siirrytään normaalimittausprofiiliin. Mikäli taas on tarve käyttää Particle Electron:in omaa LiPo-akkua, niin siirrytään niin sanottuun hätävirransäästöprofiiliin. Kuva 8 näyttää loogisen kuvauksen käynnistysprofiilin tapahtumista.

Kuva 8. Päätelaitesovelluksen käynnistyminen



Kun ulkoinen syöttöenergia todetaan vajonneen liian alas, niin päätelaite ajaa itsensä hätävirransäästöprofiiliin. Tässä profiilissa päätelaite siirtyy tilaan, jossa anturiluennat sekä pilvipalveluluennat suoritetaan enää kuuden tunnin välein. Tällöin päätelaite on oman LiPo-akkunsa varassa ja kaikki ulkoiset energiaa kuluttavat tapahtumat minimoidaan. Keskeytystulojen herättäessä laitteen, kirjataan tieto välittömästi pilvipalveluun ja päätelaite siirtyy hälytys- ja mittausprofiiliin. Mobiiliverkon auki saamiseksi, kuitenkin tulee menemään aikaa verkkokäyttelyiden myötä, mutta tässä profiilissa se hyväksytään. Kuva 9 esittää loogisen vuokaavion hätävirransäästöprofiilissa.

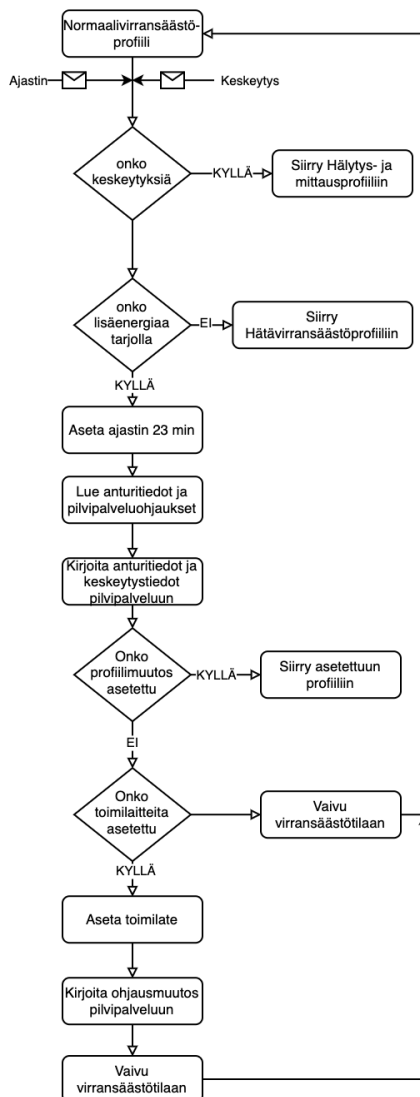
Kuva 9. Hätävirransäästöprofiilin vuokaavio



Monitoroitavan ja etähallittavan kohteen ollessa paikallaan, profiilina käytetään normaalivirransäästöprofiilia, jonka loogiset tapahtumat Kuva 10 esittää. Tällöin päätelaite voidaan nukkua tilaan, jossa kuitenkin voidaan sekä kohteen monitorointi että kohteen

toimilaitteiden etäohjaus tukea. Tässä profiilissa sensoreiden ja pilvipalveluohjausten luku suoritetaan 23 minuutin välein. Tämä aikajänne tukee myös Particle:n SIM-kortin kättelytaajuutta, jolla voidaan minimoida matkapuhelinverkon kättelyviestit päätelaitteen ja tukiaseman välillä. Tämä mahdollistaa myös matkapuhelinverkon välittömän saatavuuden, kun on tarve lukea tai kirjoittaa pilvipalveluun. Normaali virransäästöprofiilissa kaikki keskeytystulot ovat aktiivisia ja voivat ajastimen ohella herättää päätelaitteen mittausprofiiliin. Pilvipalvelun lukusyklien välisen ajan päätelaite vaipuu normaaliin virransäästöprofiiliin mukaiseen energiankulutuksen minimointiin.

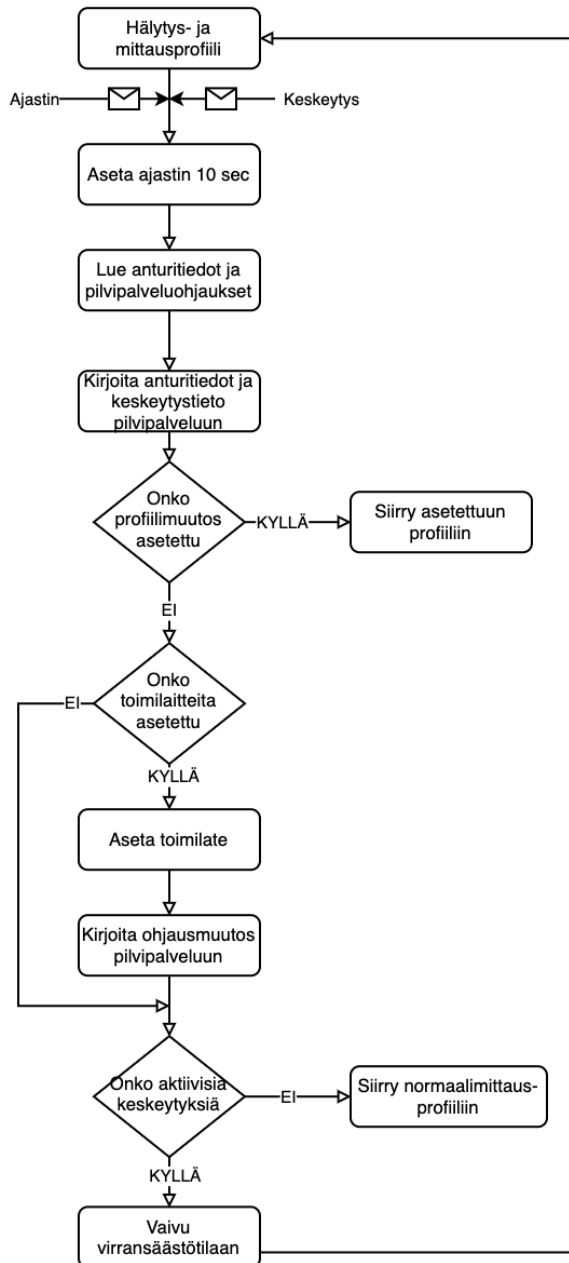
Kuva 10. Normaali virransäästöprofiilin vuokaavio



Hälytys- ja mittausprofiilissa päätelaite lukee ympäristöanturitietoja sekä paikkatietoa kymmenen sekunnin välein, kirjaten anturitiedot samalla myös pilvipalveluun. Loppukäyttäjä

voi halutessaan muuttaa profiilia pilvipalvelun kautta. Kuva 11 esittää hälytys- ja mittausprofiilin loogisen vuokaavion.

Kuva 11. Hälytys- ja mittausprofiilin vuokaavio



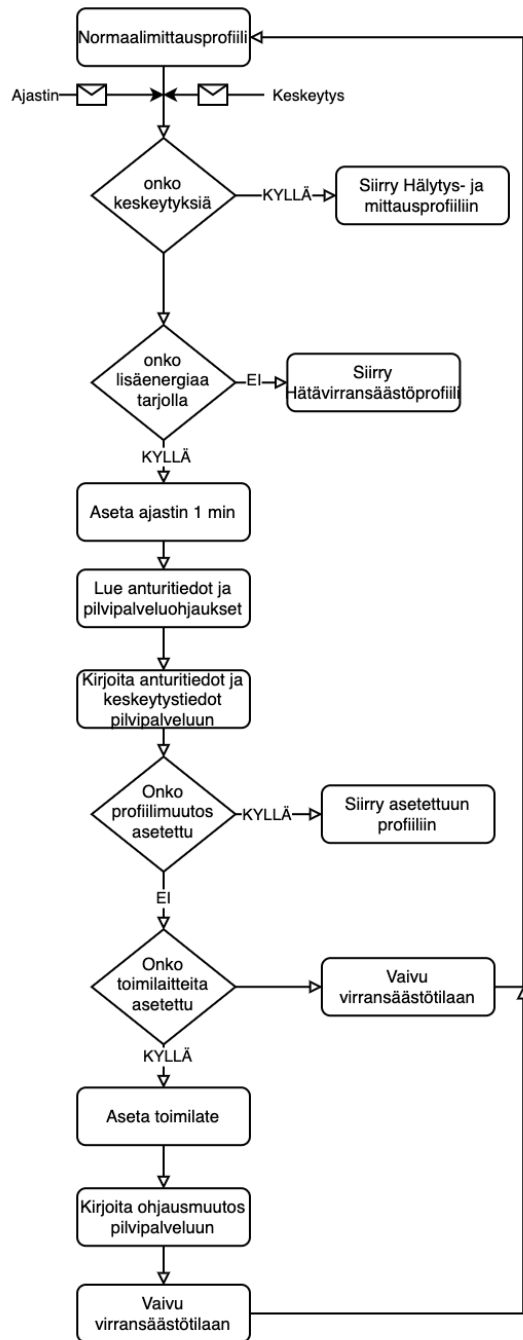
Normaalimittausprofiilissa päätelaite on herätetty asiakaspäätelaitesovelluksen kautta ja päätelaite saa energiansa ulkoisesta lähteestä. Tässä profiilissa päätelaite lukee ympäristöantureita minuutin välein ja samalla kirjaa mittausarvot pilvipalveluun.

Paikannustiedon luenta tihennetään siinä vaiheessa, kun liikeanturi ilmoittaa, että kohde

on liikkeellä. Normaali-toimintaprofiilia käytetäänkin kohteen toimilaitteiden ohjaukseen.

Kuva 12 havainnollistaa Normaali-mittausprofiilin loogisen toiminnan.

Kuva 12. Normaali-mittausprofiilin vuokaavio



Pilvipalvelun ja asiakaspäätelaitesovelluksen välillä käytetään lisäksi julkaise ja tilaa-menettelmää (publish and subscribe) reaaliaikaisen tiedon saattamiseksi pilvipalvelusta asiakaspäätteelle. Julkaise ja tilaa-menettelmässä asiakaspäätelaitesovellus ilmoittaa

serverille ne parametrit, joiden muutoksesta se on kiinnostunut. Kun parametrin arvo muuttuu niin serveri lähettää push-ilmoituksen muuttuneesta arvosta tilaajalle. Esimerkiksi edellä kerrottuun kontrolliaktivointitiedon takaisin saattamiseksi asiakaspäätesovellukselle voidaan käyttää julkaise ja tilaa-menetelmää. Kun päätelaite on saanut aktivoitua saamansa ohjaukskäskyn mukaisen toimilaitteen, päätelaite muuttaa pilvipalvelussa kyseisen parametrin arvoa ohjaukskäskyn mukaisesti. Pilvipalvelun parametrin muuttuessa pilvipalvelu lähettää välittömästi asiakaspäätelaitesovellukselle tiedon, onnistuiko ohjaukskomento vai ei. Tällöin asiakaspäätelaitesovelluksen eikä itse päätelaitteen tarvitse olla aktiivisesti seuraamassa muutosta. Indikaatiot lähetetään tiedon tilaajille vain kyseisen tiedon muuttuessa ja pilvipalvelu huolehtii muutoksen ilmoittamisesta.

Kuten edellä jo kuvattiin, niin päätelaitesovellus hyödyntää erilaisia toimintaprofiileja. Nämä perustuvat Particle Electron:in omiin sleep-moodeihin sekä erilaisiin sensoreiden lukutiheyteen sekä pilvipalvelun luku- ja kirjoitustiheyteen. Electronin käyttöjärjestelmä ja piirisarjat voidaan asettaa virransäästötilaan laiteohjelman omilla komennoilla, jolloin tehonkulutus kehitysalustassa minimoidaan. Päätelaitteen herättäminen virransäästötilasta voi tapahtua ajastimen herättämänä tai keskeytystulojen aktivoituttua.

3.4 Asiakaspäätesovellus

Microsoftin tuotteilla kehittäminen on helppoa ja omaavat myös erittäin laajan kehittäjätukien. Täten asiakaspäätelaitesovelluksen ohjelmistokehityksen alustaksi valikoitui C#-pohjainen Xamarin ja Visual Studio IDE. Näistä oli myös opinnäytetyöntekijällä kokemusta jo aikaisemmista projekteista, minkä kautta päätyminen näihin valintoihin oli helppo. Päätöstä edesauttoi se, että Particle:n pilvipalvelun hyödyntämiseen löytyi myös valmis Xamarin nuget-paketti (Particle Xamarin, 2016).

Nuget-paketti on valmiskirjasto, jonka voi helposti liittää osaksi ohjelmistoprojektia. Tosin tämä nuget-paketti ei tukenut uusimpia C#.NET-kirjastoja ja sen vuoksi kyseistä pakettia ei otettu kirjastona, vaan tuotiin aliprojektina lähdekoodeineen. Päätös oli hyvä, sillä paketin toiminnallisuuksista löytyi joitain virheitä, jotka ovat nyt korjattuna tämän opinnäytetyön ohjelmistossa. Kyseinen Xamarin-paketti sisältää kaikki tarvittavat perustoiminnot, joita asiakaspäätelaitesovellus tarvitsee käsitelläkseen Particle:n pilvipalvelua. Paketti sisältää

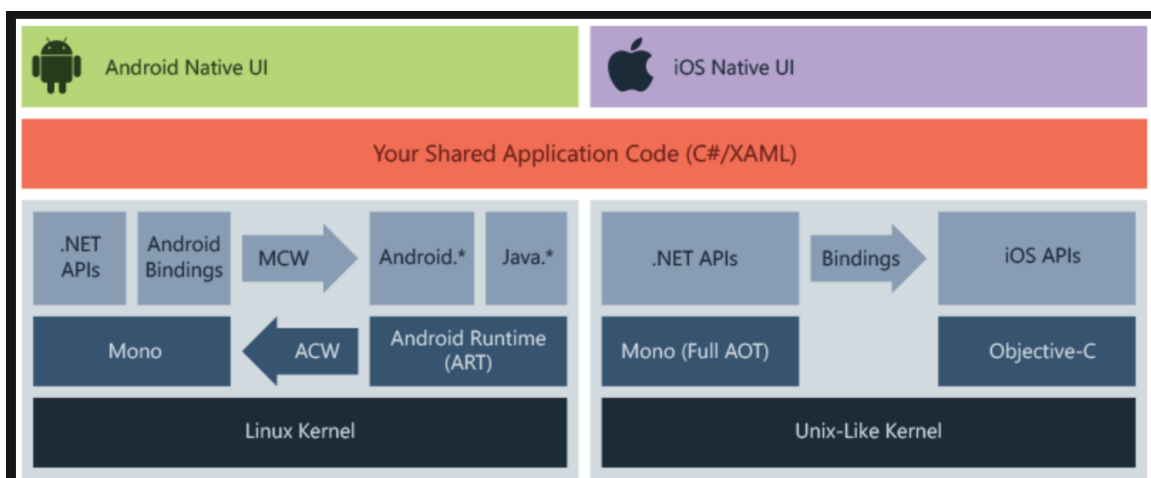
pilvipalveluun kirjautumiseen OAuth tyyllisen autentikoinnin ja siihen tarvittavat toiminnot. Myös kaikki pilvipalvelun luentaan ja kirjoittamiseen liittyvät toiminnot löytyvät kyseisestä kirjastosta. Myös julkaise ja tilaa-tyyliset muuttujien muutosten tilaukset ovat osa kyseistä nuget-pakettia. Jatkossa onkin tarkoitus viedä uusi ja korjattu versio nuget-paketista tarjolle muille kehittäjille avoimen lähdekoodin yhteisöön. (Particle Xamarin, 2016)

Muita cross-platform ohjelmointikieliä olisivat olleet Java, React, Flutter, Ionic, Appcelerator, Adobe Phone Gap, Corona, Qt, Sencha, Unity3D, 5App (Hackernoon, 2020). Päätymisen Xamarin-alustaan oli kuitenkin jokseenkin helppo nyt pilotointivaiheessa, opinnäytetyön tekijän omien mieltymysten kautta. Lopulliseen tuotteeseen tämä evaluointi kannattaa tehdä uudelleen, jotta voidaan varmistua sen hetkisestä tilanteesta. Xamarin on kuitenkin hyvää vauhtia kehittymässä kokonaisvaltaiseksi monialusta kehitysympäristöksi, joten tuki kaikille mahdollisille alustoille syntyy seuraavien .NET core 5.0 versioiden myötä. Kuva 14 kertoo Microsoft:n ajatuksen kaikkien eri alustojen yhdistämisestä .NET core 5.0-versioksi.

3.4.1 Xamarin

Xamarin on C#.Net pohjainen monialusta ohjelmistokehitysympäristö. Xamarin pohjautuu avoimeen lähdekoodiin, jolla voi tehdä sovelluksia erilaisille mobiilialustoille, kuten iOS ja Android -alustoille. Kuva 13 näyttää Xamarin-arkkitehtuurin.

Kuva 13. Xamarin-arkkitehtuuri (What is Xamarin?, 2020)



Tällä hetkellä Xamarin:in omistaa Microsoft, ostettuaan Xamarin oikeudet vuonna 2016. Microsoft on integroinut Xamarin:in osaksi työkalupalettiaan ja näin olleen myös Microsoft Visual Studiosta löytyy Xamarin-tuki. Tällä hetkellä Xamarin on käytännössä eräänlainen siltakirjasto alkuperäisten Android ja iOS -rajapintojen päällä ja käyttää täten Mono-projektin tuotosta siltakerroksena (Mono, n.d.).

3.4.2 iOS ja Android -asiakassovellukset

Opinnäytetyön tarkoituksena oli myös valmistaa pilottivaiheen päätelaitesovellus iOS ja Android -laitteille sekä edellisten lisäksi web-käyttöliittymä. Web-käyttöliittymä oli alun perin tarkoitettu kytkeytymään suoraan pilvipalveluun ja kun valinnaksi muodostui Particle:n tuotteet, päädyttiin käyttämään Particle:n oman pilvipalvelun tarjoamaa käyttöliittymää. Android ja iOS -sovellukset tehtiin Xamarin:ia hyödyntäen, jolloin saman sovelluksen business-logiikka saatiin kokonaisuudessaan hyödynnettyä molempien alustojen toteutuksiin. Opinnäytetyön osalta Android ja iOS -sovellukset saatettiin emulaattoritasolle, mikä riitti todentamaan konseptin toimivuuden. Jotta sovellukset olisi saanut matkapuhelinpätelaitteeseen asti, olisi pitänyt hankkia tunnukset sovelluskauppoihin. Tässä vaiheessa sovelluksen saattamisella matkapuheilmeen asti ei ollut konseptin toteutettavuuden kannalta arvoa, sillä emulaattorissa pystyttiin toiminnallisuudet ja suunnitelmat toteamaan toimiviksi ja käytettäväksi.

Xamarin hyödyntää iOS ja Android alkuperäisiä käyttöliittymäkirjastoja ja näin olleen vain hyvin laiteläheiset osat tarvitsevat erillisen käsittelyn Xamarin-sovelluksessa. Käyttöliittymän osalta Xamarin niin ikään hyödyntää iOS ja Android alkuperäisiä käyttöliittymäelementtejä. Näin ollen Xamarin:lla tehdyt sovellukset näyttävät kunkin alustan alkuperäisiltä sovelluksilta. Opinnäytetyön mukainen asiakaspäättesovellus sopiikin erittäin hyvin Xamarin-kehitykseen, sillä sovellus ei tarvitse mitään laiteläheisiä toimintoja, vaan komennot ja luennat menevät pilvipalveluun REST-rajapinnan kautta. Näin ollen, Xamarin pystyy hyödyntämään sovellustasolla samoja toimintoja samaan ohjelmistoon pohjautuen eri alustoilla. Esimerkiksi karttojen hyödyntäminen asiakaspäättesovelluksissa saatiin suoraan Xamarin-alustan kautta. Xamarin:lla tehty sovellus voidaan myös siirtää Mac OSX ja Windows sovellukseksi käyttämällä C#.Net core-kirjastoja. Näin olleen C#.NET on erittäin

potentiaalinen monialusta ohjelmointikieli, silloin kun sovellus toimii sovelluspinon ylimmillä tasoilla.

Tulevaisuudessa kaikki erilaiset C#.NET-alustat fuusioituvat yhdeksi .NET core 5.0-versioksi. Tämä mahdollistaa jatkossa saman sovelluksen tuottamisen kaikille tarjolla oleville eri alustoille ilman erillistä mukauttamista. Kuva 14 esittää tulevan .NET core 5.0-version rakenteen.

Kuva 14. .Net core 5.0 (Lander, 2019)

.NET – A unified platform



4 Johtopäätökset

Opinnäytetyön tekemisessä auttoi se, että tekijällä itsellään oli selkeä tavoite työn lopputulokselle. Myös tekijän pitkä historia ohjelmistojen ja järjestelmien parissa auttoi heti alussa hahmottamaan pääsuunnat. Vaatimusten kirjaaminen aikaisessa vaiheessa rajasi loppujen lopuksi jokaista osa-aluetta ja auttoi näin ollen myös rajaamaan työtä kokonaisuudessaan.

Systeemisuunnitteluvaihe oli ehkä pisin vaihe koko työssä. Ensimmäinen versio systeemiarkkitehtuurikuvasta tehtiin jo keväällä 2019. Vastaavasti viimeinen ja lopullinen kuva syntyi kesällä 2020. Systeemisuunnittelun aikana monta vaatimusta muovaantui

uusiksi. Systeemisuunnittelu onkin vaihe, johon kannattaa käyttää aikaa. Sen aikana paljastuu monta pientä asiaa ja se edesauttaa hahmottamaan kokonaisuutta.

Päätyminen Particle:n tuotteisiin oli helppo. Particle tarjoaa pilvipalvelun, jonka toiminnot löytyvät suoraan Particle:n laiteohjelmasta. Myös Xamarin nuget-paketin ja lähdekoodin löytäminen Particle:n pilvipalvelun käsittelyyn, helpotti asiakaspäätössovelluksen pilottivaiheen valmistamista.

Käytettävien verkkojen osalta päätyminen mobiiliverkkoihin oli myös helppo ratkaisu, mutta LoRa-verkon osalta on tarkoitus tehdä pilotti ennen lopullisen ratkaisun tekemistä. IoT-optimoidut verkot mahdollistavat erittäin energiatehokkaan päätelaitteen, joten työn osalta jäi hieman askarruttamaan, miten LoRa-pohjaisella laitteella voisi valmistaa vastaavat vaatimukset täyttävä laite. Particle:n Photon kontrollerilla ja Adafruit RFM96-lisälevyllä voisi rakentaa Particle:n tuotteilla LoRaWAN-tuotteen. LoRaWAN-teknologioihin löytyy myös Pycom:ilta erittäin kilpailukykyisiä kehitysalustoja (Pycom, n.d.). Tulevaisuudessa myös LTE-M myös on erittäin varteen otettava verkkovaihtoehto.

Kaiken kaikkiaan opinnäytetyössä oli tarve hyvin monenlaiselle suunnittelulle. Oli tarve osata vaatimusmäärittelyä ja systeemisuunnittelua monella tasolla. Tarvittiin myös ohjelmointia useilla kielillä sekä HW-suunnittelua. Koska työn tarkoituksena oli idean toteuttamiskelpoisuuden todentaminen, niin mitään isompia testauksia ei järjestelmälle tehty. Myöskään systeemisuunnittelun mukaisia kaikkia sensoreita ei käytetty toteutuksessa nyt, vaan todennettiin muutamalla sensorilla ja releellä I2C-väylän toimivuus niin mittauksiin kuin ohjauksiin. Myös keskeytystulojen käyttäytyminen sovelluksen osalta saatiin todennettua. Tavoitteena oli myös todentaa, että pilvipalvelua voidaan käyttää puskurina niin päätelaitesovelluksen kuin asiakaspäätössovelluksen välillä.

Asiakaspäätössovellustoteutuksella todennettiin, että niin kirjautuminen pilvipalveluun kuin tiedon lukeminen kuin myös ohjauksien toimittaminen pystytään tekemään suunnitelmien mukaisesti. Paikannustiedon saaminen päätelaitteelta pilvipalvelun kautta asiakaspäätössovellukseen ja siellä kartalle saatiin myös todennettua.

Idean toteuttamiskelpoisuusvaihetta seuraavana vaiheena on vielä edessä loppujen sensoreiden liittäminen järjestelmään, käyttöliittymäsuunnittelu sekä laajamittainen testaus.

Tämän jälkeen järjestelmä on tarkoitus asentaa valittuihin kohteisiin kenttätestausta varten. Myös tutkimus LoRaWAN-pohjaisella järjestelmällä on tarkoitus toteuttaa jatkossa.

Tämä työ myös avasi tekijälleen tarjolla olevien teknologioiden tämänhetkistä tilaa ja tulevaisuutta. IoT on jo tätä päivää ja mahdollistaa hyvin monenlaisia sovelluksia. Kilpailevia teknologioita on paljon ja lähitulevaisuus näyttää mitkä teknologioista tulevat olemaan voittavia teknologioita. Kehitys on kaikilla alueilla erittäin nopeaa ja muutamassa kuukaudessa tapahtuu paljon muutoksia. Niin kävi tämänkin työn selvittelyvaiheen aikana. Se minkä mukaan kehitysalustojen osalta pohjat tehtiin keväällä 2019, oli tilanne muuttunut täysin keväälle 2020.

Tulevaisuudessa IoT-laitteita tullaan näkemään myös teollisuudessa prosessien mittalaitteina. Teollinen Internet on tuomassa hyvää vauhtia langattomia ja pitkän elinkaaren omaavia antureita teollisuuden tarpeisiin. Langattomuudella voidaan saavuttaa kustannussäästöjä ja sensorin asennukseen joustavuutta kaapeloinnin osalta, mutta toisaalta se luo avoimen riskin järjestelmän haavoittuvuudelle. Monet IoT-teknologiat sisältävät kuitenkin sisäänrakennettuna salauksen liikennöinnille sekä muita kyberturvallisuutta tukevia rakenteita.

Vaikka tämä työ toteutettiin kehitysalustoihin pohjautuen, niin löytyy myös kaupallisia IoT-antureita monelta toimittajalta. Myös monet kaupalliset toimijat myyvät valmiita kokonaispalveluita, joilla voi halunsa mukaan mitata melkein mitä vaan ja toimittaa mittaustiedon pilvipalveluun analysoitavaksi. Esimerkiksi, Wapice (IoT-Ticket[®], n.d.), Etteplan (More, n.d.), Digita (Digita, n.d.), Tinksi (TiNKSI[™] n.d.) myyvät palveluita, joissa voi valita haluamansa anturit ja kokonaispalvelun anturitietojen seuraamiseen. Monet noista palveluista ovat pääsääntöisesti kohdistettu yrityksille, joten hinta kiipeää aika korkeaksi. Yhtenä vaihtoehtona olisi ostaa LoRaWAN anturit erillisiin mittaustarpeisiin ja liittää ne Digitan koko maan kattavaan kaupalliseen verkkoon. LoRaWAN-sensoreita löytyy erittäin paljon erilaisiin mittaustarpeisiin (The Things Network, n.d.). Hinnat kuitenkin ovat muutama kymmenen euroa anturia kohden sekä muutama euro kuukaudessa kaupallisen verkon käytöstä jokaista anturia kohden. Näiden lisäksi pitäisi vielä olla pilvipalvelupalvelin, joka kerää kaiken halutun anturitiedon yhteen palveluun. Tähän tarkoitukseen voisi esimerkiksi käyttää LoRaCloud-palvelua (LoRaCloud, n.d.).

Loppujen lopuksi opinnäytetyö täytti tavoitteensa ja lisäksi herätti tekijässään taas innovatiivisuuden ja ratkaisuhakuisuuden markkinoilta löytyvien IoT-elektroniikkaelementtien kanssa. Internet on täynnä erilaisia valmiita ja kaupallisia elektroniikkakortteja, joiden hinnat ovat muutamasta eurosta muutamaankymmeneen euroon. Kaupallisilla korteilla ja niiden erilaisilla yhdistelmillä voi mitata melkein mitä vaan. IoT-palvelut ja -järjestelmät ovat jo tätä päivää ja siksi IoT-maailmaan perehtyminen luo pohjaa monelle alalle ja mahdollistaa näin ollen monen perinteisen teknologian uudelleen ajattelemisen mahdollisuuden. IoT myös mahdollistaa isojen datamassojen keräämisen ja sen hyödyntämisen tekoälyanalyysiin.

Tulevaisuutta ajatellen, myös viidennen sukupolven verkot luovat täysin uuden mahdollisuuden hyödyntää ja kerätä dataa. Nopeiden verkkojen myötä, liki reaaliaikainen laskenta mittaustietoihin perustuen, voidaan tulevaisuudessa tehdä pilvipalvelussa. Tällöin sensoreiden älykkyys ei enää ole niin kriittistä. Tässä tapauksessa puhtaat IoT-sensorit vain tuottavat mittausdataa pilvipalveluun ja kaikki erilaisiin mittauksiin pohjautuva logiikka ja laskenta tapahtuukin täysin pilvipalvelussa. Tämä myös lisää osaltaan turvallisuutta järjestelmään datavalidointien ja järjestelmän osien kahdennuksen kautta.

Lähteet

3GLTEinfo (n.d.). *LoRa Architecture*. Haettu 12.9.2020 osoitteesta

<http://www.3glteinfo.com/lora/lora-architecture/>

Adafruit, (n.d.-a). *Adafruit FONA 808 - Mini Cellular GSM + GPS Breakout*. Haettu 11.10.2020 osoitteesta <https://www.adafruit.com/product/2542>

Adafruit, (n.d.-b). *Adafruit Ultimate GPS FeatherWing*. Haettu 11.10.2020 osoitteesta <https://www.adafruit.com/product/3133>

Adafruit, (n.d.-c). *Adafruit Feather 32u4 FONA*. Haettu 11.10.2020 osoitteesta <https://www.adafruit.com/product/3027>

Arduino, (n.d.-a). Haettu 11.10.2020 osoitteesta <https://www.arduino.cc>

Arduino, (n.d.-b). *ARDUINO MKR GSM 1400*. Haettu 11.10.2020 osoitteesta <https://store.arduino.cc/arduino-mkr-gsm-1400-1415>

Arduino Cloud, (n.d.). *Arduino IoT Cloud*. Haettu 11.10.2020 osoitteesta <https://www.arduino.cc/en/IoT/HomePage>

Blynk, (n.d.). *Build your first IoT app in five minutes*. Haettu 11.10.2020 osoitteesta <https://blynk.io/en/getting-started>

Dali, I., Mahbubur, R., Abusayeed, S. (2018). *Low-power wide-area networks: opportunities, challenges, and directions*. Haettu 13.9.2020 osoitteesta https://www.researchgate.net/publication/323986231_Low-power_wide-area_networks_opportunities_challenges_and_directions

Digita (1.11.2017). *Mikä on LoRaWAN?* Haettu 17.10.2020 osoitteesta <https://www.digita.fi/asiakastarinat/mika-on-lorawan/>

Digita, (n.d.). *Palvelut yrityksille*. Haettu 28.11.2020 osoitteesta <https://www.digita.fi/etusivu/palvelut-yrityksille/>

Digita, (n.d.). *IoT:n kartta*. Haettu 30.8.2020 osoitteesta <https://www.digita.fi/iotn-kartta/>

- DNA, (n.d.). *Peittokartta*. Haettu 11.10.2020 osoitteesta <https://www.dna.fi/yrityksille/dna-verkko/peittokartta>
- Dominique, A. (2020). *LTE-M ja NB-IoT – taistelkaa!* Blogi julkaistu 25.6.2020. Haettu 13.9.2020 osoitteesta <https://www.tivi.fi/kumppaniblogit/dna/lte-m-ja-nb-iot-taistelkaa/2b6de30e-7a3c-4b27-972c-28badfc8ace4>
- Elisa, (n.d.). *Elisa kuulumuus*. Haettu 11.10.2020 osoitteesta <https://elisa.fi/kuulumuus/>
- Firebase, (27.10.2020). *Firebase Cloud Messaging*. Haettu 8.11.2020 osoitteesta <https://firebase.google.com/docs/cloud-messaging/>
- Google Cloud, (n.d.). *Solve more with Google Cloud*. Haettu 11.10.2020 osoitteesta <https://cloud.google.com>
- Hackernoon, (2020). *11 Popular Cross-Platform Tools for App Development in 2020*. Haettu 10.10.2020 osoitteesta <https://hackernoon.com/9-popular-cross-platform-tools-for-app-development-in-2019-53765004761b>
- Ihmevekotin, (n.d.). *Particle Electron Asset Tracker 2G kit Global*. Haettu 11.10.2020 osoitteesta https://ihmevekotin.fi/product/788_particle-electron-asset-tracker-2g-kit-global
- IoT-Ticket®, (n.d.). *Wapice IoT-Ticket®*, Haettu 28.11.2020 osoitteesta <https://www.wapice.com/fi/tuotteet/iot-ticket>
- Järvinen, J. (2017). *IoT-verkkoteknologioiden vertailu*. Insinööriyö. Tietotekniikan koulutusohjelma. Metropolia Ammattikorkeakoulu. Haettu 12.9.2020 Osoiteesta https://www.theseus.fi/bitstream/handle/10024/128159/Jarvinen_Joonas.pdf?sequence=1
- Kontu, V. (2020). *Sigfox, LoRa ja NB-IoT -verkkojen vertailu*. Kandidaattityö. Konetekniikan tutkinto-ohjelma. Teknillinen tiedekunta. Oulun Yliopisto. Haettu 31.10.2020 osoitteesta <http://jultika.oulu.fi/files/nbnfioulu-202004181472.pdf>

- Kumar, S., (2016). *Particle Electron Power Consumption Analysis*. Haettu 11.10.2020 osoitteesta <https://blog.voltaicsystems.com/particle-electron-power-consumption-analysis/>
- Lander, R., (6.5.2019). *Introducing .NET 5*. Haettu 1.11.2020 osoitteesta <https://devblogs.microsoft.com/dotnet/introducing-net-5/>
- Lappi, T. (2020). *Paikantaminen Lora Verkossa*. Pro gradu -tutkielma. Tietotekniikka. Informaatioteknologian tiedekunta. Jyväskylän Yliopisto. Haettu 13.9.2020 osoitteesta <https://jyx.jyu.fi/bitstream/handle/123456789/68228/1/URN%3ANBN%3Afi%3Aju-202003182452.pdf>
- Link Labs, (2015). *What Is Weightless*. Haettu 13.9.2020 osoitteesta <https://www.link-labs.com/blog/what-is-weightless>
- Lora, (n.d.). LoRa Device classes. Haettu 12.9.2020 osoitteesta <https://lorawan.r-us.co.uk>
- LoRaCloud, (n.d.). *Overview*. Haettu 28.11.2020 osoitteesta https://www.loracloud.com/documentation/join_service?url=intro.html
- LoRaWAN, (2015). *What is it?* Haettu 12.9.2020 osoiteesta <https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf>
- Mono, (n.d.). *Mono Project*. Haettu 10.10.2020 osoitteesta <https://www.mono-project.com>
- More, (n.d.). *Etteplan More*. Haettu 28.11.2020 osoitteesta https://www.etteplanmore.com/palvelut/tiedolla-johtaminen?gclid=EAlaIqobChMIroG0tJOI7QIVLmyCh2e8wZMEAAAYASAAEgLzPfd_BwE
- Nordin, L. (n.d.). *What is LoRa and LoRaWAN*. Haettu 12.9.2020 osoitteesta <https://zakelijforum.kpn.com/lora-forum-16/what-is-lora-and-lorawan-8314>
- OZRobotics (n.d.). *Lipo Power Shield*. Haettu 11.10.2020 osoitteesta <https://ozrobotics.com/shop/lipo-power-shield/>
- Particle Cloud, (n.d.). *Device Cloud*. Haettu 10.10.2020 osoitteesta <https://www.particle.io/device-cloud/>

Particle Device OS, (n.d.). Haettu 10.10.2020 osoitteesta

<https://docs.particle.io/tutorials/device-os/device-os/>

Particle Docs, (n.d.-a). *Particle Device OS API Network System sleep configuration*, Haettu 30.8.2020 osoitteesta

<https://docs.particle.io/reference/device-os/firmware/electron/-network-systemsleepconfiguration->

Particle Docs, (n.d.-b). *Electron Datasheet*. Haettu 30.8.2020 osoitteesta

<https://docs.particle.io/datasheets/electron/electron-datasheet/#functional-description>

Particle Docs, (n.d.-c). *Electron datasheet*. Haettu 18.10.2020 osoitteesta

<https://docs.particle.io/assets/pdfs/datasheets/electron-datasheet.pdf>

Particle Electron, (n.d.). *Electron 2G/3G (EMEA) Starter Kit*. Haettu 10.10.2020 osoitteesta

<https://store.particle.io/collections/cellular/products/electron-3g-kit-intl>

Particle Xamarin, (2016). *Particle Xamarin Cloud SDK*. Haettu 17.10.2020 osoitteesta

<https://github.com/michael-watson/particle-xamarin>

Piirainen, L. (2018). *Sisätilapaikannus*. Insinööriyö. Tieto- ja viestintäteknikka. Lahden Ammattikorkeakoulu. Haettu 25.10.2020 osoitteesta

https://www.theseus.fi/bitstream/handle/10024/155695/Piirainen_Lauri.pdf?sequence=1&isAllowed=y

Pycom, (n.d.). *Pycom*. Haettu 28.11.2020 osoitteesta <https://pycom.io>

Pycom, (n.d.-a). *LoPy4*. Haettu 28.11.2020 osoitteesta <https://pycom.io/product/lopy4/>

Pycom, (n.d.-b). *Pytrack 2.0 X*. Haettu 28.11.2020 osoitteesta

<https://pycom.io/product/pytrack-2-0-x/>

RAK2011, (n.d.) Haettu 11.10.2020 osoitteesta

<https://www.rakwireless.com/en-us/products/nb-iot-boards/rak2011>

Raspberry Zero, (n.d.). Haettu 2.11.2020 osoitteesta

<https://www.raspberrypi.org/products/raspberry-pi-zero/?resellerType=home>

- Raspberry, (n.d.). Haettu 11.10.2020 osoitteesta <https://www.raspberrypi.org>
- Ray, B. (2020). *Four Types Of IoT Wireless Networks*. Blogi julkaisu 24.3.2020. Haettu 30.8.2020. <https://www.iotacommunications.com/blog/types-of-iot-networks/>
- Raza, U. Kulkarni, P., Sooriyabandara, P., (11.1.2017). *Low Power Wide Area Networks: An Overview*. Haettu 13.9.2020 osoitteesta <https://arxiv.org/pdf/1606.07360.pdf>
- Semtech, (n.d.). *Calculate battery life*. Haettu 12.9.2020 osoitteesta <https://loradevelopers.semtech.com/library/tech-papers-and-guides/the-book/calculate-battery-life/>
- Sigfox, (n.d.). *Sigfox coverage*. Haettu 17.10.2020 osoitteesta <https://www.sigfox.com/en/coverage>
- Sigfox build, (n.d.-a). *Sigfox Network location*. Haettu 3.10.2020 osoitteesta <https://build.sigfox.com/geolocation-technologies#sigfox-network-location>
- Sigfox build, (n.d.-b). *Network overview*. Haettu 25.10.2020 osoitteesta <https://build.sigfox.com/sigfox#network-overview>
- Soracom, (n.d.). *Global cellular connectivity designed for IoT and M2M*. Haettu 11.10.2020 osoitteesta <https://www.soracom.io>
- SP-Electroniikka, (n.d.). *Magneettikytkin ovikytkin karmipuoli ja ovipuoli MAKY2 NO/NC*. Haettu 18.10.2020 osoitteesta <https://www.spelektroniikka.fi/p6872-magneettikytkin-ovikytkin-karmipuoli-ja-ovipuoli-maky2-no-nc-fi.html>
- Sparkfun, (n.d.-a). *SparkFun Environmental Combo Breakout - CCS811/BME280 (Qwiic)*. Haettu 18.10.2020 osoitteesta <https://www.sparkfun.com/products/14348>
- Sparkfun, (n.d.-b). *Zio Current and Voltage Sensor - INA219 (Qwiic)*. Haettu 10.10.2020 osoitteesta <https://www.sparkfun.com/products/15176>
- Sparkfun, (n.d.-c). *SparkFun Qwiic Single Relay*. Haettu 18.10.2020 osoitteesta <https://www.sparkfun.com/products/15093>

Sparkfun, (n.d.-d). *SparkFun OpenPIR*. Haettu 18.10.2020 osoitteesta

<https://www.sparkfun.com/products/13968>

Stenberg, E., (2020), *Mender 2.3 released: Easily update both system and applications*.

Haettu 11.10.2020 osoitteesta

<https://mender.io/blog/mender-2-3-released-easily-update-both-system-and-applications>

Stevens, J., Berkvens, R., Looockx, W., Weyn, M. (2013). *Robot Localization With DASH7*

Technology. Haettu 25.10.2020 osoitteesta <https://anet.uantwerpen.be/desktop/irua>

Telia, (n.d.). *Verkkokartta*. Haettu 11.10.2020 osoitteesta

<https://www.telia.fi/asiakastuki/verkko/verkko/verkkokartta>

The Things Network, (n.d.). *Devices*. Haettu 28.11.2020 osoitteesta

<https://www.thethingsnetwork.org/marketplace/products/devices>

TiNKSI™, (n.d). *TiNKSI kauppa*. Haettu 28.11.2020 osoitteesta <https://tinksi.fi/kauppa/>

Wei-Meng, L. , Clarence, C. , (n.d.). *Introduction to IoT Using the Raspberry Pi*. Haettu

11.10.2020 osoitteesta <https://www.codemag.com/article/1607071/Introduction-to-IoT-Using-the-Raspberry-Pi>

Wirepass, (n.d). Haettu 30.8.2020 osoitteesta <https://wirepas.com>

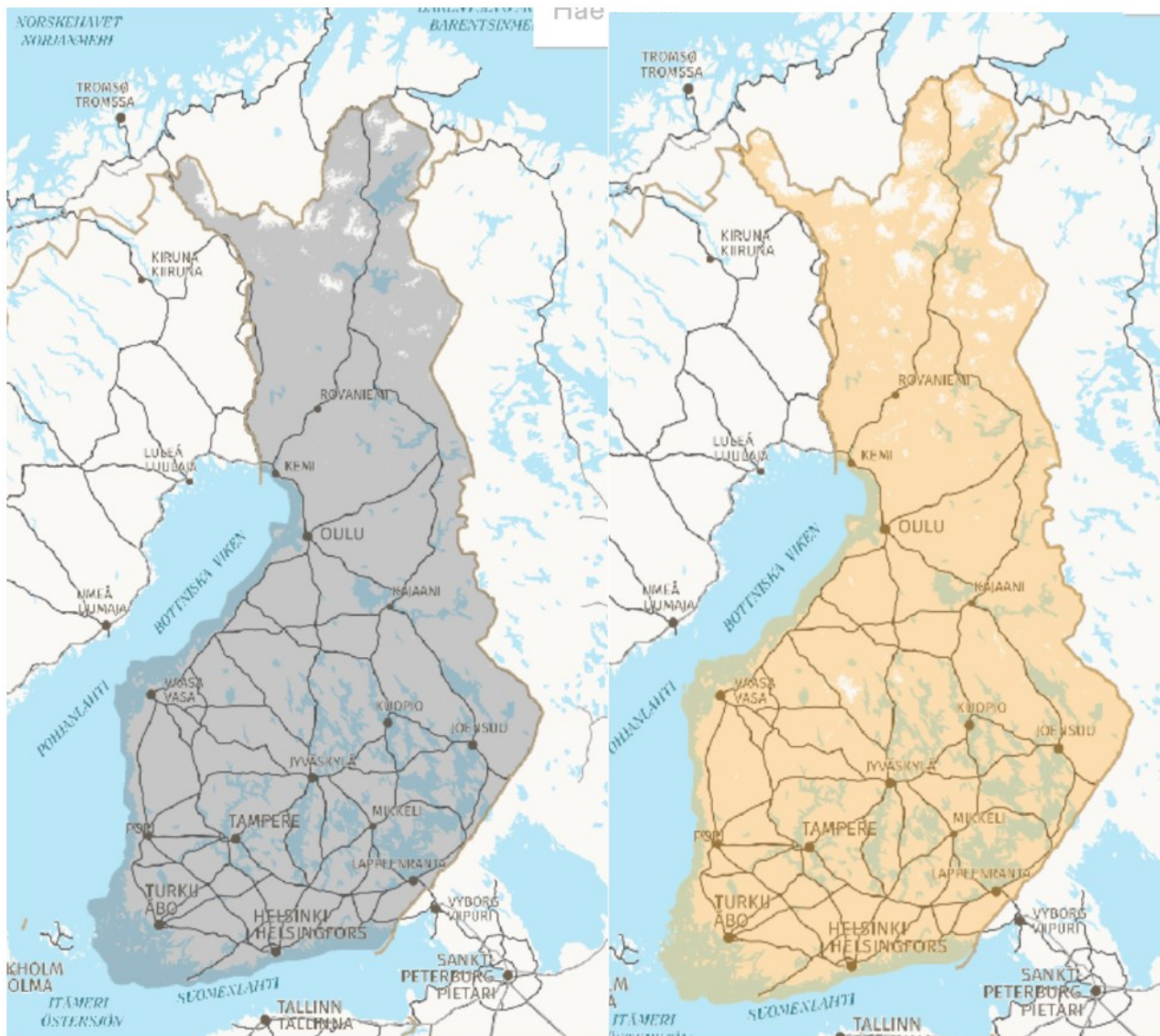
What is Xamarin?, (28.5.2020). Haettu 10.10.2020 osoitteesta

<https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>

Liite 1: Kehitysalustojen vertailutaulukko

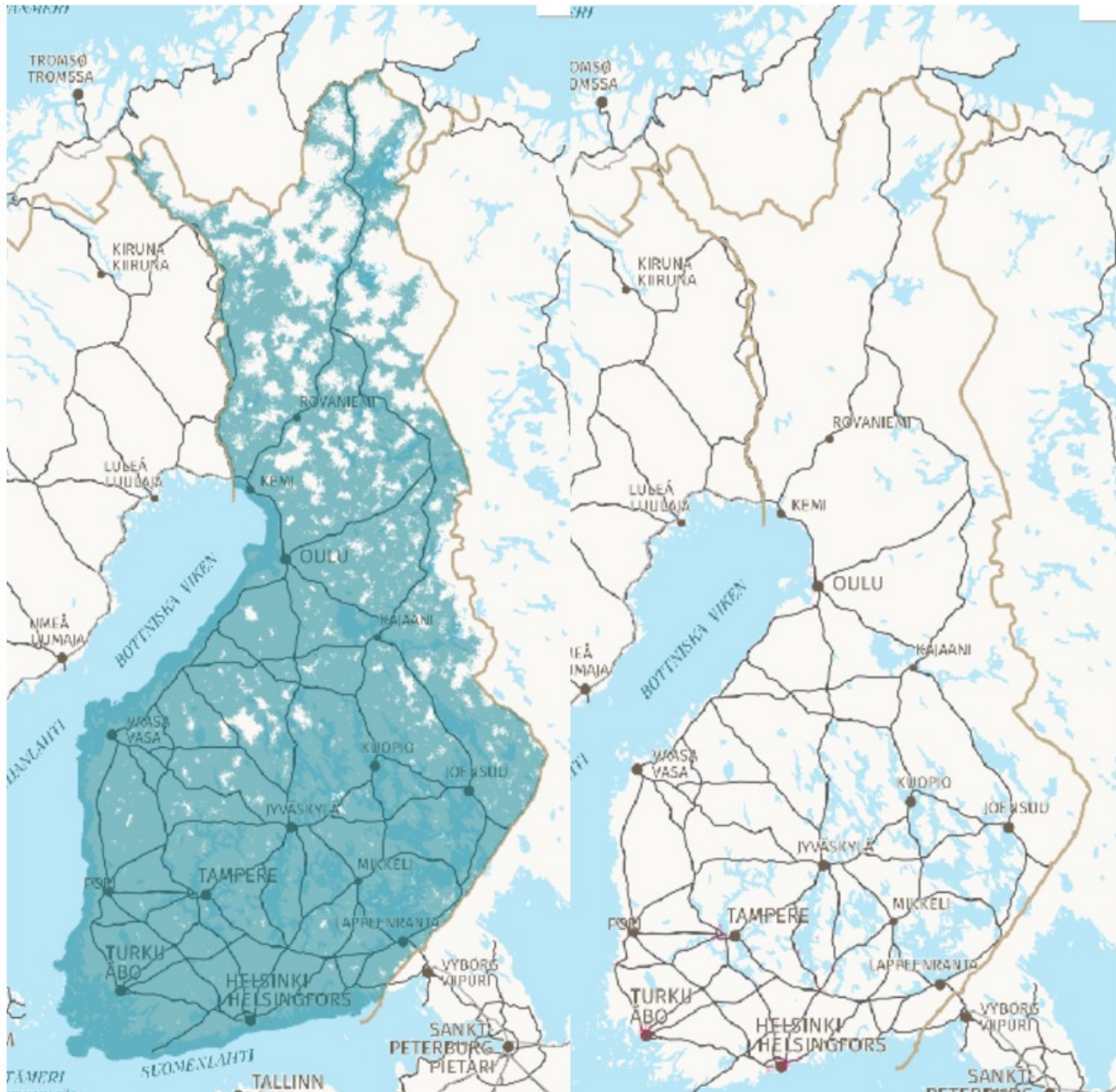
	Raspberry PI + Adafruit FONA 808	Arduino UNO + RAK Wireless WisLink Cellular BG96 Shield	Arduino MKR GSM 1400 + Adafruit Ultimate GPS FeatherWing	Particle Electron + Asset Tracker shield	Adafruit Feather 32u4 FONA + Adafruit Ultimate GPS FeatherWing	PyCom LoPy4 + Pytrack 2.0 X
Emolevyn Data Sheet	https://www.raspberrypi.org/documentation/usage/gpio/README.md	https://store.arduino.cc/arduino-uno-rev3	https://store.arduino.cc/arduino-mkr-gsm-1400-1415	https://docs.particle.io/datasheets/electron/electron-datasheet/	https://cdn-learn.adafruit.com/downloads/pdf/adafruit-feather-32u4-fona.pdf	https://docs.pycom.io/datasheets/development/lopy4/
I/O tuki keskeytystuloille	Kyllä	kyllä 2kpl	kyllä	kyllä	kyllä	kyllä
I2C-väylä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
Analogiatuloja	26 GPIO	6	7	12	10	36 GPIO
Analogialähtöjä	26 GPIO	6 PWM	1	13 PWM	7 PWM	36 GPIO
Digitaalituloja	26 GPIO	14	8	30 GPIO	20 GPIO	36 GPIO
Digitaalilähtöjä	26 GPIO	14	8	30 GPIO	20 GPIO	36 GPIO
Energiansyöttö- mahdollisuuksia	USB, Jos käyttää LiPo- akku on syytä käyttää erillistä LiPo- akulle tarkoitettua lisälevyä.	USB, VIN 12V	USB, VIN 12V	Oma LiPo-akku, Ulkoinen 12V ja USB	USB, 12V LiPoly Akku	Vain yksi 3.5 - 5.5 V
Energiasyötön hallintapiiri	ei	ei	ei	Kyllä, BQ24195	Akun lataukseen kyllä, 500 mA Lipoly charger	ei
Hintaluokka koko päätelaitteelle	Rpi 3 B+ 30€ Adafruit FONA 808 69,90€	Arduino UNO 29,90€ RAK Wireless WisLink Cellular BG96 Shield noin 67€	Arduino MKR GSM 1400 59,90€ Adafruit Ultimate GPS FeatherWing 37,13€	140 €	Adafruit Feather 32u4 FONA 38,74€ Adafruit Ultimate GPS FeatherWing 37,13€	LoPy4 34,95€ Tracker 36,95€
Integroitu pilvipalvelu	ei	kyllä	kyllä	Kyllä	ei	ei
Käyttöjärjestelmä	Linux	Open Source FW kirjastot	Open Source FW kirjastot	Open Source FW kirjastot	Open Source FW kirjastot	MicroPython
käyttöjärjestelmä ja firmawaretuki pilvipalvelulle	ei	kyllä, löytyy valmis kirjasto https://github.com/arduino-libraries/ArduinoIoTCloud	kyllä, löytyy valmis kirjasto https://github.com/arduino-libraries/ArduinoIoTCloud	Kyllä	Kyllä	ei
OTA-päivitykset sovellukseen	kyllä erillisen palvelun kautta	Kyllä SD kortin kautta	Kyllä SD kortin kautta	kyllä	Ei tietoa	kyllä
OTA-päivitykset firmwareen ja käyttöjärjestelmään	kyllä erillisen palvelun kautta	Kyllä SD kortin kautta	Kyllä SD kortin kautta	kyllä	Ei tietoa	kyllä
Tietoturva päätelaitteen ja pilvipalvelun välillä	Vaatii sovelluskehittäjältä erillistä hallintaa.	Arduino Cloud integroitu Firmwareen	Arduino Cloud integroitu Firmwareen.	Particle Cloud integroitu Firmwareen	Eri pilvipalveluihin löytyy eriaisia kirjastoja	Ei tietoa
Ohjelmointikielät	C,C++, Python, Java	C ja C++	C ja C++	C ja C++	C ja C++	MicroPython ja Python
Tuki monen valmistajan sensoreille	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
Kehittäjätki	Laaja	Laaja	Laaja	Laaja	Laaja	Laaja
Laitteen elinkaari	Pitkä	Pitkä	Pitkä	2G lyhyt, 3G pitkä	Pitkä	Pitkä
Tulevaisuuden tuki	Pitkä	Pitkä	Pitkä	Erittäin hyvä.	Pitkä	Pitkä

Liite 2: 2G ja 3G -verkon kattavuus Suomessa DNA-operaattorilla



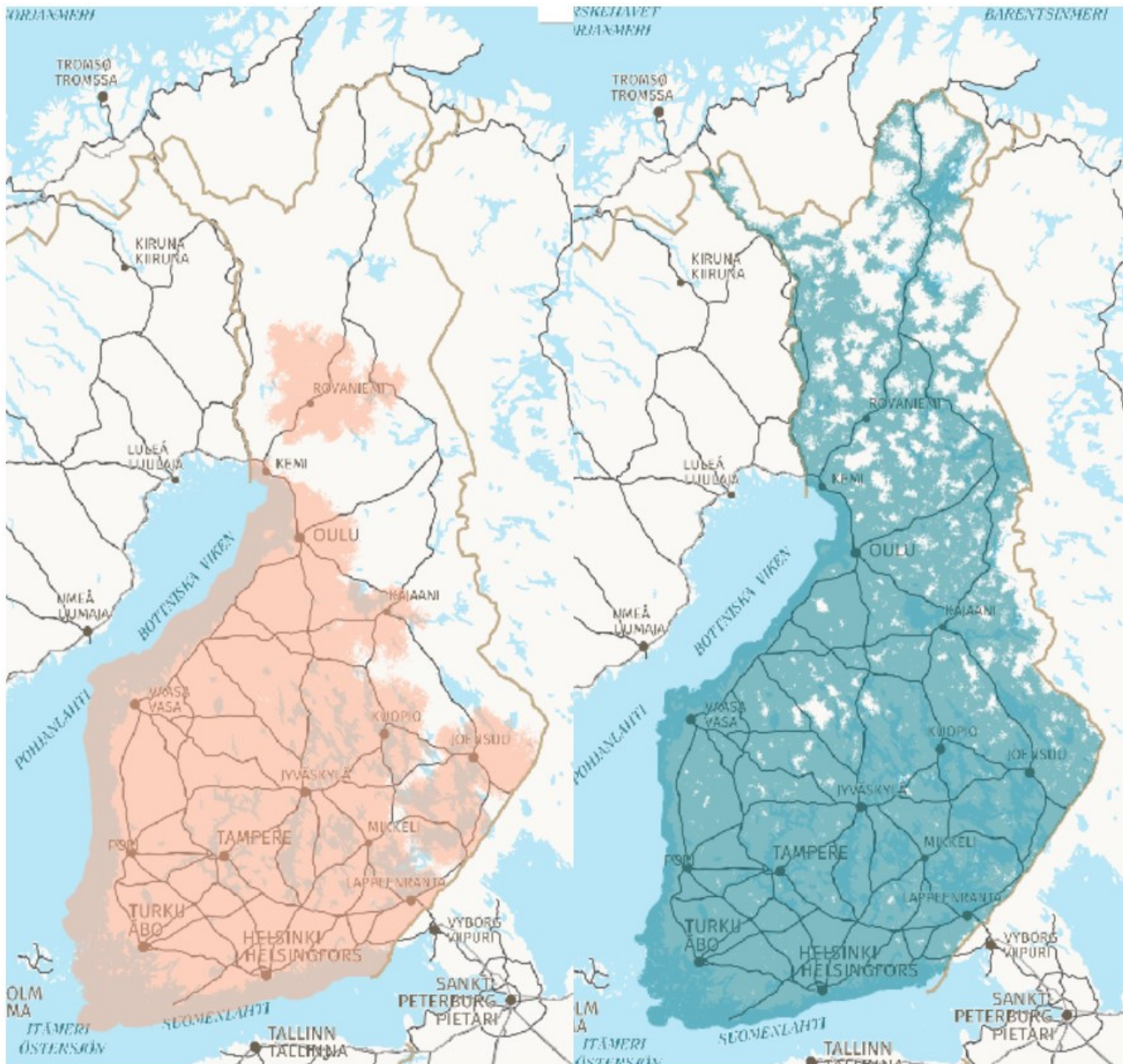
Kuvassa vasemmalla esitetty 2G-verkon kattavuus (harmaa) ja oikealla 3G-verkon kattavuus (keltainen). Kattavuuskuvat otettu DNA:n palvelusta 11.10.2020 (DNA, n.d.)

Liite 3: 4G ja 5G -verkon kattavuus Suomessa DNA-operaattorilla.



Kuvassa vasemmalla esitetty 4G-verkon kattavuus (sininen) ja oikealla 5G-verkon kattavuus (punainen). Kattavuuskuvat otettu DNA:n palvelusta 11.10.2020 (DNA, n.d.)

Liite 4: NB-IOT ja LTE-M -verkkojen kattavuus Suomessa DNA-operattorilla.



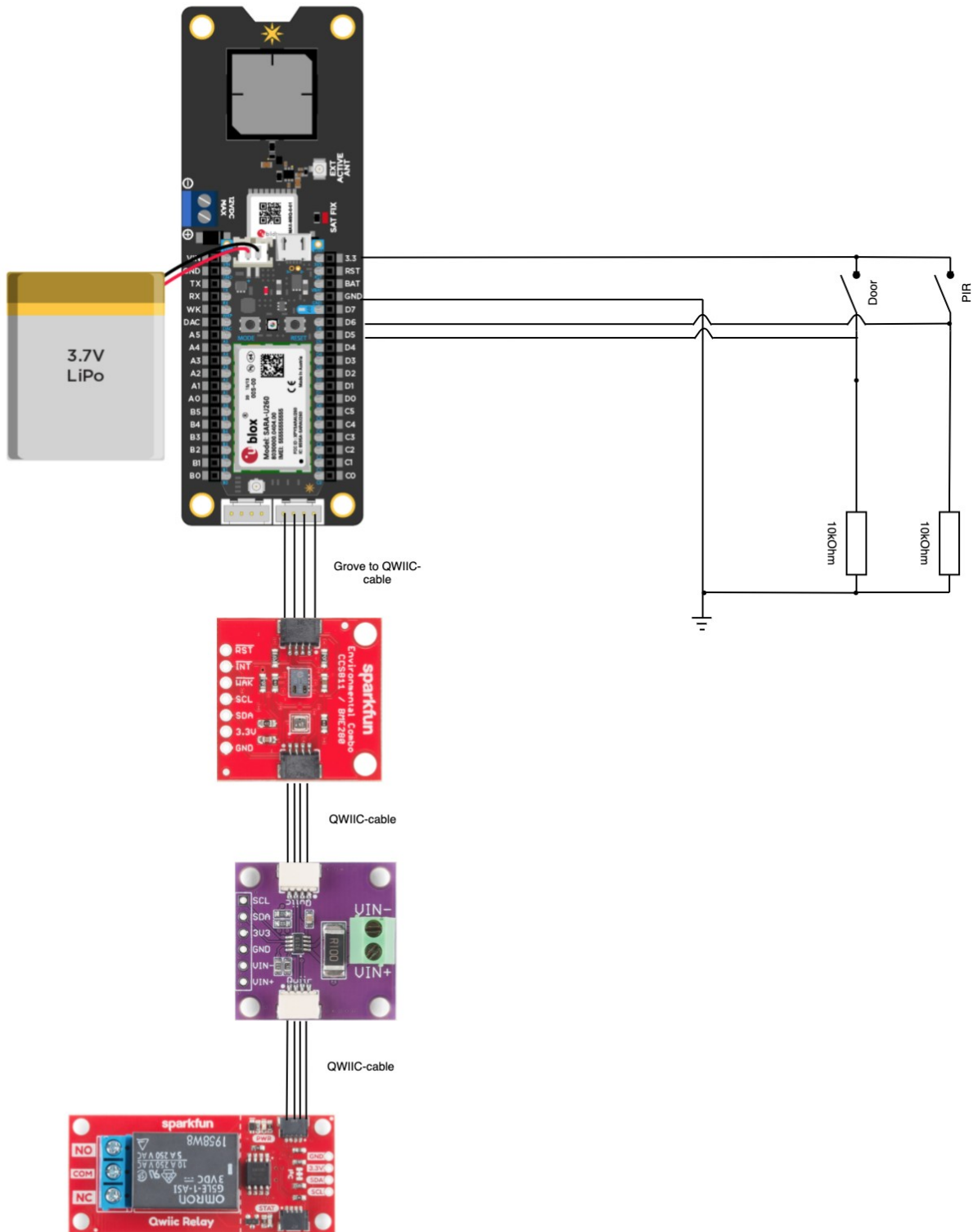
Kuvassa vasemmalla esitetty NB-IOT-verkon kattavuus (oranssi) ja oikealla LTE-M-verkon kattavuus (Sininen). Kattavuuskuvat otettu DNA:n palvelusta 11.10.2020 (DNA, n.d.)

Liite 5: Sigfox-verkon kattavuus Suomessa.



Kattavuuskuva otettu Sigfox palvelusta (Sigfox, n.d.) 17.10.2020.

Liite 6: Sensoreiden liittäminen Particle Electron Asset Tracker:iin



Liite 7: Sensoritiedon luku- ja ohjauskomennon havainnollistava sekvenssikaavio

