

Antero Raudas

Pilvialustat IoT havainnointidatan käsittelyssä

Tietotekniikan
pro gradu -tutkielma
27. huhtikuuta 2022

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Kokkolan yliopistokeskus Chydenius

Tekijä: Antero Raudas

Yhteystiedot: anteror@iki.fi

Puhelinnumero: 040-833 5247

Ohjaaja: Ismo Hakala

Työn nimi: Pilvialustat IoT havainnointidatan käsittelyssä

Title in English: Cloud platforms as IoT data processors

Työ: Tietotekniikan pro gradu -tutkielma

Sivumäärä: 57+0

Tiivistelmä: Tässä työssä tutustutaan pilvipalveluiden ja IoT mittalaitteiden yhteistoimintaan. Työssä luodaan katsaus kolmeen tunnettuun mittalaitteiden käyttämään verkkoteknologiaan, ja seitsemään suosittuun kaupalliseen ja avoimen lähdekoodin IoT alustaan. Työssä esitellään alustoille asetettuja vaatimuksia ja valintakriteereitä ja tutustutaan IoT alustojen referenssimalleihin, joita on kehitetty helpottamaan sovellusten kokonaisuuden hallintaa. Työssä toteutetaan Mökin olosuhdetietojen mittaussovellus, joka kerää olosuhteisiin ja säätilaan liittyvää havaintodataa ympäristöstään, ja hyödyntää Microsoft Azuren pilvipalveluita datan prosessointiin, tallennukseen ja esittämiseen.

Avainsanat: IoT, pilvipalvelut, vaatimuksia ja valintakriteereitä, referenssimalli, olosuhteiden mittaussovellus

Abstract: This thesis introduces the co-operation between cloud services and IoT measuring devices. The work provides an overview of three well-known network technologies used by metering devices, and a seven IoT cloud platforms for which both commercial and open source platforms have been selected. The work presents the requirements and selection criteria for them and introduces well-known reference models of IoT platforms. The work implements the Cottage's condition data measurement application, which collects observational data related to conditions and weather from its environment, and utilizes Microsoft Azure's cloud services for data processing, storage and presentation.

Keywords: IoT, cloud platforms, demands and selection criteria, reference model, environment measuring solution

Copyright © 2022 Antero Raudas

All rights reserved.

Sisältö

Sanasto	1
1 Johdanto	2
2 Low Power Wide Area Network teknologia	4
2.1 LPWAN	4
2.2 LoRa ja LoRaWAN	5
2.3 Sigfox	6
2.4 NB-IoT	7
2.5 Yhteenveto esitetyistä LPWA-teknologioista	8
3 Referenssimalleja ja vaatimuksia IoT alustoille	10
3.1 IoT alustan referenssimalleja	10
3.2 IoT alustoille asetettuja vaatimuksia ja valintakriteereitä	13
3.2.1 Toiminnalliset palveluvaatimukset	15
3.2.2 Ei-toiminnalliset palveluvaatimukset	17
3.2.3 Arkkitehtuurivaatimukset	18
3.2.4 Valintakriteereitä	19
4 Katsaus IoT pilvialustoihin	22
4.1 Avoimen lähdekoodin IoT alustat	22
4.1.1 Kaa	23
4.1.2 Temboo	24
4.1.3 ThingSpeak	25
4.2 Kaupalliset IoT alustat	27
4.2.1 AWS IoT Platform	27
4.2.2 Microsoft Azure IoT Platform	28
4.2.3 Google Cloud Platform	29
4.2.4 IBM Watson IoT Platform	29

5	Microsoft Azure IoT	32
5.1	Azure IoT palveluita	32
5.2	IoT datan rakenteen mukainen luokittelu	37
5.3	Azuren tietovarastot	37
5.4	Azure sovelluskehitys	39
6	IoT sovelluksen toteutus pilvialustalle	41
6.1	Mökin olosuhdetietojen mittaussovellus	41
6.1.1	Olosuhdemittaus	43
6.1.2	Ilmatieteen laitoksen avoin data	43
6.1.3	LoRa verkkopalvelu	44
6.1.4	Microsoft Azure	44
6.1.5	Power BI raportointityökalu	46
6.2	Kokemuksia	46
7	Johtopäätökset ja pohdintaa	50
	Lähteet	53

Sanasto

Big data	Massadata, erittäin suurten tietotietomassojen keräämistä, analysointia jakamista ja säilyttämistä
CoAP	Kommunikaatioprotokolla
Gateway	Yhdyskäytävä
HTTP/- HTTPS	Tietoliikenneprotokollia
IaaS	Infrastructure as a Service. Pilvipalvelumalli
IoT	Internet of Things
JSON	JavaScript Object Notation. Yleinen dataformaatti
LoRa	Long Range Wide Area
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
M2M	Machine to machine
MQTT	Kommunikaatioprotokolla
NB-IoT	Solupohjainen LTE-verkko
NoSQL	Relaatiomallista poikkeava tietokanta
PaaS	Platform as a Service. Pilvipalvelumalli
QoS	Quality of Service. Palvelun laatu
SaaS	Software as a Service. Pilvipalvelumalli
Sigfox	LPWAN verkko-operaattori
SQL	Structured Query Language. Relaatiotietokantojen standardoitu kyselykieli

1 Johdanto

Käsitteen IoT (Internet of Things, Esineiden internet) esitti ensimmäisenä Kevin Ashton [8] vuonna 1999. IoTin perusajatuksena on kytkeä fyysiset laitteet internetiin, ja tästä syystä syntyi siis nimi esineiden internet. Verkossa olevien laitteiden on tarkoitus toimia mahdollisimman itsenäisesti, ilman että ihmisen tarvitsee puuttua niiden toimintaan. Laitteet voivat havainnoida ympäristöään ja jakaa ja vastaanottaa tietoa toisiltaan. Ne voivat toimia itsenäisinä tiedonkeruupisteinä tai osana suurempaa järjestelmää.

Madakam et al. [27] määrittelevät IoT:n näin: *Avoim ja kattava älykkäiden objektien verkko, joka pystyy automaattisesti järjestämään, jakamaan informaatiota, dataa ja resursseja, ja reagoimaan ja toimimaan tilanteiden ja ympäristön muutosten edessä.* Langatonta teknologiaa hyödyntävien verkkosovellusten ja IoT-laitteiden ja -sovellusten määrä on kasvanut valtavasti viime vuosina, ja IoT sovellukset ovat nykyisin yhä enemmän iteknologian valtavirtaa. Pienempien ja halvempien laitekomponenttien esiinmarssi on tuonut komponentit yhä useampien ulottuville, ja tämä on osaltaan johtanut siihen, että erilaisia sovelluksia erilaisiin käyttötarkoituksiin ja eri sovellusaloille on voitu kehittää entistä enemmän.

IoT-teknologiaa ja tiedonvälitykseen tarvittavia verkkoja mittalaitteineen käytetään siis useilla eri aloilla. Teollisuus hyödyntää niitä muun muassa kokoonpanolinjoiltaan, kuljetusala seuraa kulkuneuvojen sijaintitietoja, ja maatalous hyödyntää teknologiaa esimerkiksi peltojen kastelussa. Älykodeissa valaistus ja lämmitys säätyvät sensoreista kerättävän informaation mukaan, älykaupungeissa liikennevaloja ohjataan havaintojen perusteella. Erilaiset mittalaitteet mittaavat esimerkiksi ilmanlaatua ja patojen vedenkorkeutta. Terveysthuollossa kehoon kiinnitettävät anturit tuottavat tärkeää tietoa potilaan terveydentilasta. Kerättävää tietoa voidaan käsitellä ja kerätyn tiedon perusteella voidaan suorittaa toimintoja tosiaikaisesti, tai kertynyttä dataa voidaan analysoida ja käsitellä eri tavoin. Sovelluksia ja mahdollisuuksia on lukemattomasti.

IoT-laitteiden räjähdysmäinen kasvu tarkoittaa luonnollisesti myös kasvanutta datamäärää. Laitteet keräävät ja lähettävät tietoa taustajärjestelmiin varastoitavaksi ja analysoitavaksi, ja erilaisten sovellusten käyttöön. Pilvipalvelut, joista on yhtä lailla tullut jo it-teknologioiden valtavirtaa, tarjoavat koko joukon IoT alustoja jotka varastoivat, välittävät, laskevat ja analysoivat sensoreiden keräämää dataa.

IoT-sovellus voidaan kuvata koostuvan Microsoft Azure IoT arkkitehtuurimallin [31] mukaisista komponenteista. Laitteella (*Device*) voidaan rekisteröityä pilveen turvautusti ja lähettää telemetriatietoja. Laitteessa voi olla asetuksia joita määritellään pilvestä käsin. Vaihtoehtoisesti useampi laite voi kommunikoida paikallisesti reunalaitteen (*Edge Gateway*) kanssa. Reunalaite voi suorittaa datalle prosessointia ja välittää datan eteenpäin pilveen. Pilven yhdyskäytäväpalvelu (*Hub*) tarjoaa palvelut saapuvan datan tietoturvalliseen lukemiseen ja laitehallinnan kyvykkyyksien tarjoamiseen. Stream-prosessorit (*Stream processors*) käyttävät samaansa telemetriadataa, yhdistävät sen liiketoimintaprosesseihin ja tallentavat datan tietovarastoon (*Data storage*). Käyttöliittymä (*User interface*) visualisoi datan käyttäjälle ja helpottaa laitehallintaa.

Tässä tutkielmassa luodaan katsaus siihen kokonaisuuteen, jossa sensoreista saatu havaintodata matkaa mittalaitteelta radioteitse yhdyskäytävälle, sieltä edelleen pilveen varastoitavaksi, analysoitavaksi ja sovellusten jalostettavaksi, ja esitettäväksi loppukäyttäjille. Tämän johdantoluvun jälkeen toisessa luvussa perehdytään LPWAN (Low Power, Wide Area Networking) -teknologiaan, ja LoRa (Long Range) - ja LoRaWAN (Long Range Wide Area Network), Sigfox- ja NB-IoT -protokolliin. Luvussa kolme perehdytään pilviratkaisujen valinnan tueksi kehitettyihin referenssimalleihin, ja käydään läpi vaatimuksia ja valintakriteereitä, joita voidaan asettaa ratkaisuille. Neljännessä luvussa luodaan katsaus työhön valittuihin pilvipalvelualueisiin ja tutkitaan niiden ominaisuuksia. Koska sopivin alusta määräytyy aina tapauskohtaisesti riippuen ratkaisuun tarvittavista ominaisuuksista, alustoja ei yritetä asettaa minkäänlaiseen paremmuusjärjestykseen. Viides luku kertoo meille tarkemmin Microsoft Azure IoT arkkitehtuurista ja sen tarjoamista palveluista. Kuudennessa luvussa tutustutaan työssä toteutettuun Mökin olosuhdetietojen mittaussovellukseen. Luvussa katsotaan mistä komponenteista sovellus koostuu ja käydään läpi työn kulku, sen vaiheet ja esiin nousseita haasteita. Seitsemännessä luvussa tehdään yhteenveto työstä, ja luku sisältää pohdintaa suoritetusta kokonaisuudesta.

2 Low Power Wide Area Network teknologia

Kasvava ja koko ajan kehittyvä IoT -sovellusmaailma on asettanut uusia vaatimuksia myös verkkoteknologioille. Perinteiset verkot eivät kaikilta osin vastaa uusien IoT-sovellusten vaatimuksiin. Näitä IoT-sovellusten vaatimuksia ovat skaalautuvuus, energiatehokkuus ja kyky kommunikoida pitkienkin välimatkojen päähän. Välimatkan merkitys korostuu varsinkin sovelluksissa, joiden mittalaitteet havainnoivat ympäristöä laajalta alueelta [13]. Edellä mainitut vaatimukset ovat vauhdittaneet Low Power Wide Area (LPWA) -teknologioiden kehitystä. Nämä teknologiat mahdollistavat uusien verkkosovellusten kehityksen ja julkaisemisen.

2.1 LPWAN

Low Power Wide Area Networks (LPWAN) on yleisnimitys langattomille verkkoteknologioille, jotka mahdollistavat pitkän kantaman langattoman kommunikoinnin alhaisemmilla kustannuksilla ja pienemmällä virrankäytöllä. LPWAN verkkojen ominaispiirteitä ovat energiatehokkuus, lähetettävän datan pieni määrä, skaalautuvuus ja verkon pitkä kantama. Teknologioilla saavutetaan kaupunkioiloissa 2-5 kilometrin ja harvaan asutuilla seuduilla jopa 10-40 kilometrin kantama [23]. Laitteet kuluttavat vähän virtaa, ja virtalähteenä käytettävä paristo voi kestää jopa yli 10 vuotta. Lisäksi laitteissa käytetty radiotekniikka ja laitteiden operointikulut ovat edullisia.

LPWAN-verkot on standardoitu, ja ne on luokiteltu kahdentyyppiseen tekniikkaan ja protokollaan. Toinen niistä operoi lisensoimattomilla 2.4 GHz:n ja alle 1 GHz:n radiokanavilla, joiden yleisnimitys on ISM-kanavat (Industrial, Scientific and Medical). Nämä kanavat ovat yleisesti "Short Range"laitteiden käytössä, ja protokollat kuten esimerkiksi Zwave, 802.11.ah, 802.15.4g ja 802.15.4 käyttävät myös näitä samoja taajuuksia. Paikallisilla viranomaisilla on usein määräyksiä ja rajoituksia ISM-kanavien käytölle. Moduloinnissa käytetään tyypillisesti joko Ultra Narrow Band, kapeakaistaista modulointia tai hajaspektritekniikkaa (Spread Spectrum). LoRa ja Sigfox ovat tämän tyyppin teknologioita.

Toisessa tyypissä kommunikointi tapahtuu matkaviestintäverkoissa. Verkot ovat 3GPP standardointiorganisaation standardoimia, ja ne käyttävät matkapuhelinoperaattorien lisensoituja LTE- ja GSM-taajuuksia viestintään. Suomessa LTE-verkon taajuusalueet ovat 450 MHz, 700 MHz, 800 MHz, 1800 MHz ja 2.6 GHz, GSM-verkon vastaavasti 900 MHz ja 1800 MHz, ja 5G taajuudet 3.5 GHz, 700 MHz ja 30 GHz [47]. NB-IoT on esimerkki tämän tyyppin teknologiasta.

2.2 LoRa ja LoRaWAN

LoRa on Semtechin kehittämä ja omistama OSI-mallin mukainen fyysisen kerroksen verkkoteknologia. Se soveltuu hyvin pienten datamäärien nopeaan ja vähätehoiseen lähettämiseen ja vastaanottamiseen. LoRa verkko muodostetaan mittalaitteista jotka ovat yhteydessä LoRaWAN-protokollaa käyttäen yhdyskäytäviin (gateway), jotka puolestaan välittävät datan eteenpäin taustajärjestelmiin esim. Ethernetiä tai matkapuhelinverkkoa pitkin.

LoRa pohjautuu käytännössä Chirp Spread Spectrum (CSS), viserryshajaspektritekniikkaan, johon on integroitu Forward Error Correction (FEC) virheenkorjaustekniikka. CSS-tekniikalla signaali levitetään koko taajuusalueelle. Näin saavutetaan tietoturvalisempää viestintää ja suurempi sietokyky radiokohinaa ja häirintää vastaan. LoRa-laitteet kommunikoivat keskenään lisenssivapaalla ISM-taajuusalueella käyttäen alle 1 GHz:n matalia taajuuksia. Käytetyt taajuuskaistat vaihtelevat alueittain. Euroopassa käytössä ovat 868 MHz ja 433 MHz, USAssa 915 MHz, ja Aasiassa 433 MHz taajuuskaistat.

LoRan omistaa Semtech, mutta OSI-mallin ylemmät kerrokset ovat vapaita kehitykselle. LoRaWAN on hyvin tuettu avoin protokolla, jota hallinnoi yleishyödyllinen yhdistys LoRa Alliance [13]. Lora Alliance (<https://lora-alliance.org/>) on voittoa tavoittelematon yhdistys, jonka tavoitteena on edistää ja kehittää LoRaWAN-teknologiaa ja IoT-ekosysteemiä [26]. LoRaWAN on operaattorivapaa ja lisenssimaksuton malli, mikä tekee infrastruktuurin hallinnasta yksinkertaista ja halvempää. LoRaWAN verkon käyttöönottoon tarvitaan LoRa Alliancen myöntämä NetID, tai vaihtoehtoisesti LoRaWAN verkon käytöstä maksetaan verkon tarjoajan kautta.

LoRaWAN tukee turvallista, mobiilia kaksisuuntaista kommunikointia maksimi-hyötykuorman (payload) ollessa 242 tavua [25]. LoRawanin pakettikohtainen otsikkotietojen koko on 13 tavua [25]. LoRan kantama ja siirtokapasiteetti riippuu linkibudjetista. Sitä voidaan vaihdella muuttamalla kaistanleveyttä, koodausjärjestelmää, lähetystehoa, kantoaaltotaajuutta ja spreading factory (SF)-arvoa. SF-arvoja voi olla 6 välillä SF7-SF12. Mitä suurempi SF-arvo, sitä pienempi on siirtokapasiteetti ja kantama suurempi. LoRan kantamaksi on testeissä saavutettu kaupunkioloissa 5 km, jolloin on onnistuneesti vastaanotettu 85% paketeista, ja maaseudulla 8 km, ja paketit on tällöin vastaanotettu 100-prosenttisesti [13].

LoRaWAN jakaa päätelaitteet kolmeen luokkaan A, B ja C [28]. Jako tehdään downlinkin latenssin ja virrankulutuksen mukaan. Kaikki luokat sallivat kaksisuuntaisen kommunikoinnin. Luokkien ero tulee siitä, kuinka noodi vastaanottaa sanomia gatewaylta.

Luokan A laite lähettää sanoman tukiasemalle, jonka jälkeen sillä on kaksi aikaikkunaa jolloin se voi vastaanottaa kuittaussanoman. Muina aikoina noodi on unitilassa. Luokka A on vähiten virtaa kuluttava.

Luokan B laitteella on samanlaiset aikaikkunat vastaanottoon, mutta niiden lisäksi sillä on ylimääräinen aikajakso, jolloin se kuuntelee tulevia sanomia. Tukiaseman ja päätelaitteen synkronointiin käytetään ns. beacon-viestejä.

Luokan C laite vastaanottaa jatkuvasti sanomia tukiasemilta, pois lukien lähetysaika. Se on luotettavin, mutta kuluttaa samalla myös eniten virtaa. C-luokan laitteita ovat mm. aktuaattorit, jotka ovat jatkuvasti verkkovirrassa.

2.3 Sigfox

Sigfox on LPWAN verkko-operaattori, joka hallinnoi omaa patentoituun teknologiaansa perustuvaa IoT-verkkoa. Paikallinen verkko-operaattori tarjoaa Sigfox-verkon ja siihen sopivat päätelaitteet ja tukiasemat, jotka välittävät dataa ip-verkon kautta pilvipalveluihin. Loppukäyttäjät saavat mittalaitteiden tuottaman datan omille sovellusaloilleen pilvestä. Verkko on suunniteltu välittämään pieniä datamääriä, ja sen kantamaksi on mitattu kaupunkioloissa 3-10 kilometriä ja harvaan asutuilla

seuduilla 20-50 kilometriä [13]. Verkko toimii alle 1 GHz:n ISM-kanavilla, ja päätelaitteet kommunikoivat tukiasemien kanssa käyttäen BPSK-modulaatiota.

Kommunikointi oli alkujaan yksisuuntaista päätelaitteilta tukiasemille, mutta nykyisin voidaan lähettää myös downlink-viestejä pääteasemilta sensoreille. Downlink-viestin lähetystä täytyy tosin edeltää uplink-viesti mittalaitteelta tukiasemalle, jonka jälkeen tukiasema voi välittömästi lähettää downlink-viestin. Viestien määrä on rajoitettu. Uplink-viestejä voidaan lähettää päivässä 140 kpl, ja downlink-viestien päivittäinen määrä on 4 kpl. Tämä johtaa siihen, että ACK-kuittausviestit eivät onnistu joka viestille. Viestien perillemeno varmistetaan lähettämällä sama viesti kolmeen kertaan satunnaisesti valittujen alikanavien kautta. Käytetty kanava (esim. Euroopassa välillä 868.180 MHz - 868.220 MHz) jaetaan 400:n 100 Hz:n ultra-narrowband -alikanavaan (joista 40 on varattu ja 360 käytettävissä), ja näistä valitaan satunnaisesti kolme alikanavaa viestin lähettämiseen. Tukiasemat osaa- vat kuunnella kaikkia näitä kanavia ja poimia käsittelyyn yhden näistä kolmesta samasta viestistä. Viestien hyötykuorma on 12 tavua uplink-viesteissä ja 8 tavua downlink-viesteissä.

2.4 NB-IoT

NB-IoT (Narrowband IoT) on solupohjainen LTE-verkko (Long Term Evolution). Se on pyritty pitämään mahdollisimman yksinkertaisena, jotta laitteiden kulut ja paristo- jen virrankulutus pystyttäisiin pitämään mahdollisimman alhaisina. Tästä syys- tä siitä on karsittu useita LTE:n ominaisuuksia, jotka eivät ole välttämättömiä IoT- sovelluksissa. Tällaisia ovat muun muassa laadun mittaukseen liittyviä monitoroin- tiominaisuuksia ja kanavan vaihtomahdollisuus. NB-IoT toimii LTE:n käyttämällä 700, 800 ja 900 MHz taajuuksilla ja käyttää 200 KHz kaistanleveyttä. Modulointime- netelmänä hyödynnetään QPSK:ta [42].

NB-IoT:ssa on kolme erilaista toimintaoperaatiota: *stand-alone*, *guard-band* ja *in-band operation*. Stand-alone -toiminnassa operoidaan käytössä olevien taajuuksien seassa, guard-band:ssa hyödynnetään taajuuksien välissä olevia suojakaistoja, ja in-band toiminnassa hyödynnetään vapaita resurssilohkoja LTE-kantoaallossa.

NB-IoT on luotettava tekniikka. Sen siirtonopeudet, 200 kb/s downlink ja 20 kb/s

uplink-suuntaan, viestin maksimikoko 1600 tavua ja pieni latenssi ovat LoRaa ja Sigfoxia paremmat. Toisaalta sen käyttö on huomattavasti kalliimpaa, koska tarvittavan NB-IoT -liittymän joutuu ostamaan alueellisilta teleoperaattoreilta.

NB-IoT vaatii LTE-verkon. Verkkoja löytyy parhaiten tiheästi asutuilla taajama- ja kaupunkialueilta, joten se ei ole kovin käyttökelpoinen harvaan asutuilla seuduilla. NB-IoTin kantamaksi on mitattu kaupunkialueiden yhdestä kilometristä maaseudun 10 kilometriin [28]. NB-IoT-tekniikkaa hallinnoi ja kehittää 3rd Generation Partnership Project (3GPP) ja GSMA, kaksi televiestintäalan standardisoimisyhdistystä, joilla on yhteinen tavoite edistää matkaviestinverkkojen ja -laitteiden kehitystä.

2.5 Yhteenveto esitetyistä LPWA-teknologioista

LoRaWAN, NB-IoT ja Sigfox -teknologioissa on jokaisessa ominaispiirteitä joilla on merkitystä valittavaan teknologiaan. Merkittäviä tekijöitä ovat muun muassa viestien kantama, viestin hyötykuorman maksimikoko, lähetettävien ja vastaanotettavien viestien määrän mahdolliset rajoitukset sekä teknologian aiheuttamat kustannukset.

Taulukkoon 2.1 on koostettu LoRaWANin, Sigfoxin ja NB-IoTin keskeisiä piirteitä. Taulukosta voidaan nähdä, että Sigfoxin kantama on verrattavista paras sekä kaupunkialueilla että harvaanasutuilla seuduilla. LoRaWANin kantama on melko hyvä, kun taas NB-IoT ei yllä yhtä pitkiin kantamiin muihin verrattuna. NB-IoTin viestien maksimikoko on ylivoimainen muihin verrattuna, ja LoRaWANin suurempi kuin Sigfoxin. Sigfoxin huonona ominaisuutena on rajoitettu viestien päivittäinen lukumäärä, joskin myös LoRaWANilla on viranomaisten määräyksiin perustuvia rajoituksia viestien määrässä. Sigfoxissa downlink-viestejä voidaan lähettää paljon vähemmän kuin uplink-viestejä (140 UL/4 DL), mikä johtaa siihen, että vain harvoihin uplink-viesteihin voidaan lähettää luotettavan viestinnän aikaansaamiseksi kuittausviesti mittalaitteelta. NB-IoT on kustannuksiltaan huomattavasti muita kalliimpi. Kaiken kaikkiaan, ja luonnollisesti käyttötarkoituksen huomioiden, LoRaWANin voidaan ajatella olevan ominaisuuksiltaan vähintäänkin kohtuullinen vaihtoehto teknologiaa valittaessa.

Taulukko 2.1: LoRaWAN, NB-IoT ja Sigfox-protokollien keskeisiä ominaisuuksia (taulukko mukailen Mekki et al. esitystä [28])

	LoRaWAN	NB-IoT	Sigfox
Modulaatio	CSS	QPSK	BPSK
Taajuus	Lisensioimattomat ISM-taajuudet (868 MHz Euroopassa, 915 MHz Pohjois-Amerikassa ja 433 MHz Aasiassa)	Lisensoidut LTE-taajuudet	Lisensioimattomat ISM-taajuudet (868 MHz Euroopassa, 915 MHz Pohjois-Amerikassa ja 433 MHz Aasiassa)
Kaistanleveys	250 kHz ja 125 kHz	200 kHz	100 Hz
Maksimi datanopeus	50 kbps	200 kbps	100 bps
Viestien määrä/päivä	Maakohtaisia viranomaisrajoituksia	rajoittamaton	140 (UL), 4 (DL)
Maksimi hyötykuorman pituus	243 tavua	1600 tavua	12 tavua (UL), 8 tavua (DL)
Kantavuus	5 km (kaupunki), 20 km (maaseutu)	1 km (kaupunki), 10 km (maaseutu)	10 km (kaupunki), 40 km (maaseutu)
Todennus ja salaus	Kyllä (AES 128b)	Kyllä (LTE-salaus)	Ei tuettu
Standardointi	LoRa Alliance	3GPP	Sigfox-yritys tekee yhteistyötä ETSI:n kanssa Sigfox-pohjaisen verkon standardoinnissa

3 Referenssimalleja ja vaatimuksia IoT alustoille

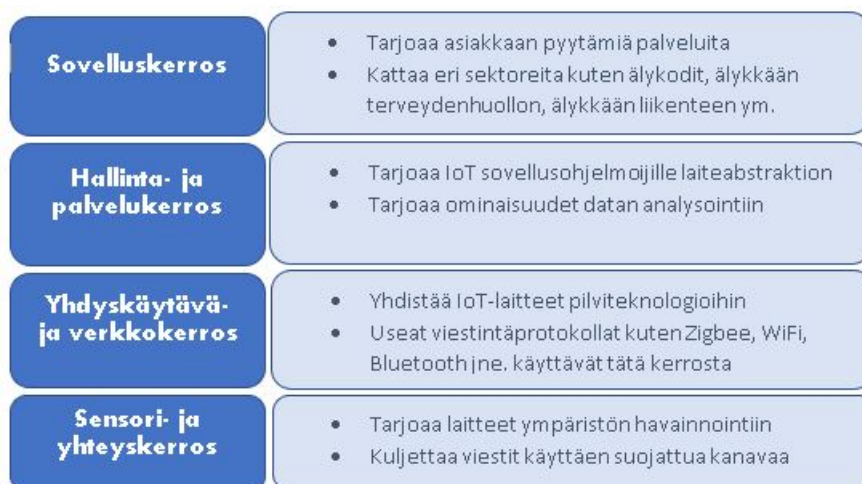
Mittalaitteet ovat vuosi vuodelta kehittyneet, niistä on tullut edullisia, ja IoT-ratkaisujen kehittäminen on tullut entistä helpommaksi sekä vähemmän erityisosaamista vaativaksi. Ratkaisujen yleistyessä myös havainnointidatan määrä on kasvanut valtavasti. Tätä kerättyä dataa tulee kyetä käsittelemään nopeasti ja sen on oltava helposti saatavilla. Esimerkiksi itseohjautuvien autojen tulee pystyä käsittelemään mittalaitteista saatu havaintotieto välittömästi, jotta auton kontrollointi olisi mahdollista. Dataa tulee kyetä myös varastoimaan suuria määriä, sillä havainnointidatata tuottavien laitteiden määrä on valtavassa kasvussa. Pilvialustat pystyvät vastaamaan näihin IoT-ympäristöjen tuomiin haasteisiin. Ne ovat nopeita, helposti skaalautuvia ja niihin saa tallennettua suuria määriä dataa.

Hyödyntämällä valmiita pilvialustoja yritys voi keskittyä omaa liiketoimintaansa tukevien järjestelmien ylläpitoon ja kehittämiseen sen sijaan, että sen tarvitsisi huolehtia tämän lisäksi pilvi-infrastruktuurin rakentamisesta ja ylläpitämisestä. Markkinoilla on tarjolla lukuisia erilaisia pilvipalveluita, joilla on jokaisella omat erityispiirteensä. Saattaa olla haasteellista valita juuri se omaan ympäristöön sopivin palveluratkaisu.

Valintaa tukemaan on tehty useita vertailuja ja tutkimuksia, joista muutamia tarkastellaan seuraavassa. Ensin luodaan katsaus referenssimalleihin, joita on luotu selkeyttämään pilvipalveluita. Sen jälkeen käsitellään alustoille asetettuja vaatimuksia. Vaatimuksista tarkastellaan erikseen toiminnallisia ja ei-toiminnallisia palveluvaatimuksia sekä arkkitehtuurivaatimuksia. Lopuksi käsitellään muutamia valintakriteereitä, joita voidaan hyödyntää IoT-alustan valinnassa.

3.1 IoT alustan referenssimalleja

Pilvipalveluiden valintaa selkeyttämään on kehitetty useita erilaisia referenssimalleja. Mallit helpottavat eri palveluntarjoajien alustojen vertailua, ja ovat tukena valitessa omiin tarpeisiin sopivaa ratkaisua. Referenssimalleissa pyritään jakamaan IoT-



Kuva 3.1: Kuva mukailen Agarwal et al. referenssiarkkitehtuuria [2]

kokonaisuus pienempiin loogisiin komponentteihin ja saamaan kokonaisuus paremmin hallittaviin osiin. Referenssimalleja ovat kehittäneet mm. Agarwal et al. [2], Hejazi et al. [19], Guth et al. [17] ja Fremantle et al. [14].

IoT-ratkaisut ovat pohjimmiltaan samankaltaisia: ne yhdistävät laitteita, keräävät, prosessoivat ja varastoivat laitteiden tuottamaa dataa, ja kykenevät tekemään päätöksiä prosessoidusta datasta saatujen tulosten perusteella. Ne tarjoavat lisäksi visuaalisen näkymän loppukäyttäjille. Tämän johdosta myös referenssimallit ovat hyvin samankaltaisia. Jako neljään kerrokseen (mittalaittekerros, verkkokerros, palvelukerros ja sovelluskerros) on yleinen ja selkeä jakotapa, jota Agarwal et al. referenssimalli noudattaa. Agarwal et al. mallia tullaan hyödyntämään tässä työssä.

Agarwal et al. referenssimalli koostuu seuraavista kerroksista: sensori- ja yhteyskerros (*Sensor and Connectivity Layer*), yhdyskätävä- ja verkkokerros (*Gateway and Network Layer*), hallinta- ja palvelukerros (*Management and Service Layer*), ja ylimpänä sovelluskerros (*Application Layer*) (kuva 3.1).

Sensori- ja yhteyskerros koostuu mittalaitteista datan havaittajoina ympäristöstä, ja aktuaattoreista kerrokselle välitettyjen toimintojen toteuttajana. Mitattavat suureet voivat olla esimerkiksi lämpötila, liike, valoisuus tai ilmankosteus. Mittalaitteet toimittavat mitatun datan eteenpäin digitaalisessa muodossa yhdyskätävä- ja verk-

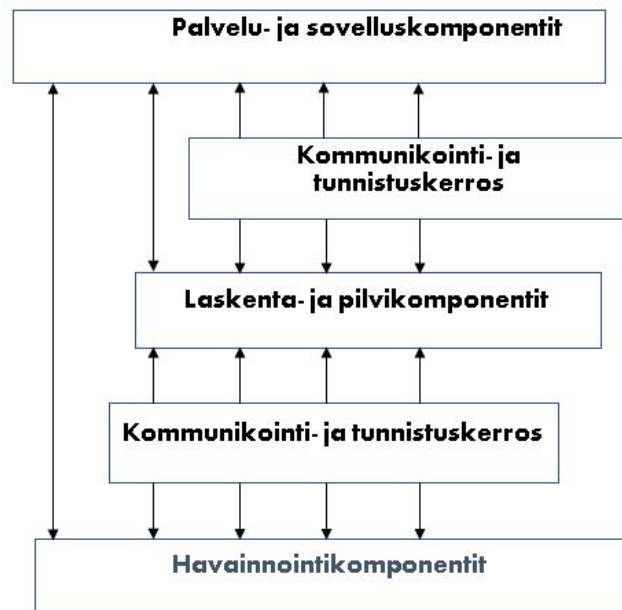
kokerrokselle. Mittalaitteet voivat mitata esimerkiksi huoneen valaistusta, ja aktuaattorit voivat saada mitatun tiedon perusteella käskyn joko himmentää tai kirkastaa huoneen valaistusta.

Suurin osa datan ja resurssien hallinnasta tapahtuu yhdyskäytävä- ja verkkokerroksessa. Kerros huolehtii lisäksi tiedon varastoinnista. Tehokkaaseen resurssienhallintaan löytyy useita algoritmeja [5]. Big dataa varten tehtyjä tiedon varastointipalveluita löytyy pilvipalveluissa useita [24]. Varastoitava data voi olla muodoltaan joko rakenteista, puolirakenteista tai ei-rakenteista. Kaikkia rakennetyyppejä voidaan käsitellä yhtäläisesti [4].

Hallinta- ja palvelukerros yhdistää palvelun pyytäjänsä kanssa. Tämä kerros tarjoaa ominaisuuksia, joiden avulla IoT -sovellusohjelmoijat voivat työskennellä mittalaitteiden parissa saumattomasti, huolehtimatta alemmista kerroksista. Kerros myös prosessoi vastaanotetun datan, ratkaisee sen jatkokäsittelyn, ja ratkaisujen perusteella toimittaa dataa eteenpäin sovelluskerrokselle. Tässä kerroksessa hyödynnetään erilaisia analyyttisiä ratkaisuja älykkäiden päätösten tekemiseksi.

Sovelluskerros sisältää loppukäyttäjän palvelut ja sovellukset. Palvelu voi olla esimerkiksi potilaan sykemittaustietojen välittämistä terveydenhuollon palvelujen tarjoajalle. Sovelluskerros tarjoaa palveluita varsinkin monille aloille, kuten esimerkiksi ratkaisuja älykkääseen kotiin, älykkääseen terveydenhuoltoon, älykkääseen liikenteeseen, teolliseen automaatioon ja vaikkapa älykkääseen energianhallintaan. Sovelluksia on tarjolla lähes kaikille datan esittämiseen kykeneville, verkkoon liitettäville laitteille.

Verrattaessa **Hejazi et al.** (kuva 3.2) referenssimallia Agarval et al. malliin, nähdään että kummassakin ovat alin ja ylin kerros samankaltaiset. Alin kerros pitää molemmissa sisällään mittalaitteet, ylin palvelut ja sovellukset. Siinä missä Agarval et al. välikerrokset ovat yhdyskäytävä- ja verkkokerros sekä hallinta- ja palvelukerros, Hejazi tunnistaa mallissaan mittalaitteiden ja sovellusten välissä laskenta- ja pilvikomponenttikerroksen ja kommunikointi- ja tunnistuskerroskerroksen. Viimeksi mainittu on mallissa sekä havainnointikomponenttien ja laskenta- ja pilvikomponenttikerroksen että laskenta- ja pilvikomponenttikerroksen ja palvelu- ja sovelluskomponenttikerroksen välissä. **Guth et al.** (kuva 3.3) referenssimallissa havainnoin-

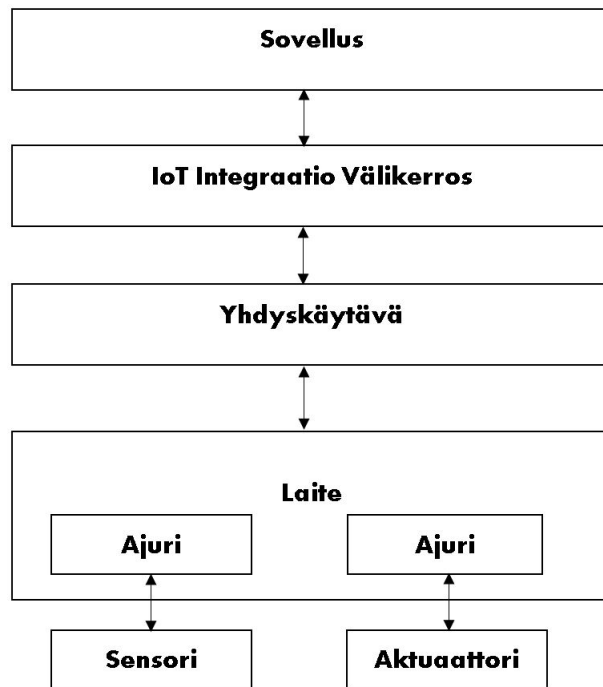


Kuva 3.2: Kuva mukaillen Hejazi et al. referenssiarkkitehtuuria [19]

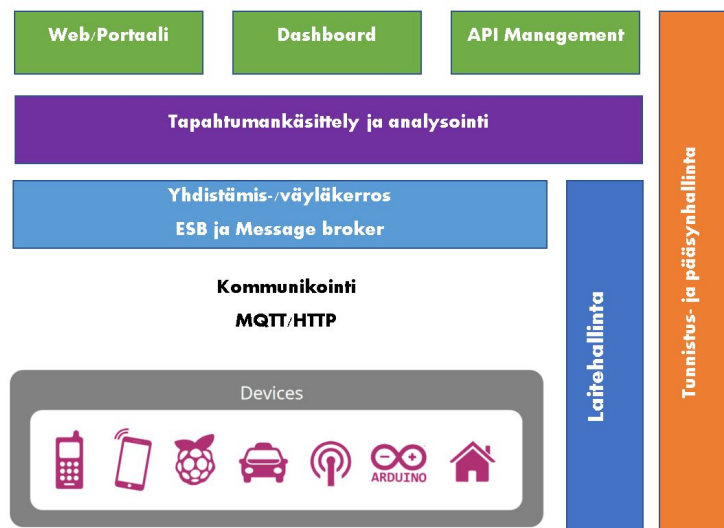
tilalaitteet (mittalaitteet ja aktuaattorit) voivat olla omia komponenttejaan, ja yhdistettynä laitteeseen ajureilla. Sekä Hejazi et al. että Guth et al. referenssimalleista voidaan helposti tunnistaa samat elementit kuin Agarwal et al. referenssimallissa, kun taas **Fremantle et al.** referenssimallin (kuva 3.4) lähestymistapa poikkeaa hieman edellä esitetyistä. Se ei ole niin abstrakti kuin muut, ja siinä esimerkiksi nimetään joitakin mallissa käytettyjä teknologioita.

3.2 IoT alustoille asetettuja vaatimuksia ja valintakriteereitä

Koska markkinoilla on huomattava määrä erilaisia IoT-alustoja, oikean pilvialustan valinta voi olla vaikeaa. Mitä seikkoja pitää ottaa huomioon valintaa tehtäessä? Agarwal et al. [1] jakavat IoT-alustoihin kohdistuvat vaatimukset toiminnallisiin ja ei-toiminnallisiin palveluvaatimuksiin sekä arkkitehtuurivaatimuksiin. Valintakriteereiksi artikkelissa nostetaan ominaisuudet kuten vakaus, saatavuus, jakelutapa, hinnoittelumalli, tuki tarvittavalle laitteistolle, tietoturvaan liittyvät vaatimukset, tuki kommunikaatioprotokollille, tiedon tallennustekniikat ja tuettu analytiikka. Hejazi et al. [19] puolestaan nostavat edellä mainittujen lisäksi esille joustavuuden.



Kuva 3.3: Kuva mukaillen Guth et al. referenssiarkkitehtuuria[18]



Kuva 3.4: Kuva mukaillen Fremantle et al. referenssiarkkitehtuuria [14]

IoT alustoille asetetut palveluvaatimukset voidaan jakaa toiminnallisiin, ei-toiminnallisiin ja arkkitehtuurivaatimuksiin. Ei-toiminnalliset vaatimukset liittyvät lähinnä palvelun laadun varmistamiseen (Quality of Service, QoS) [10][38]. Arkkitehtuurivaatimukset tukevat sovellusten kehittämistä. Täyttämällä vaatimukset helpotetaan useita ohjelmointitehtäviä kuten esimerkiksi laitteistototeutusten ohjelmoinnin abstrahointia, sisäänrakennettujen rajapintojen tekoa ja kirjastojen luontia [2].

3.2.1 Toiminnalliset palveluvaatimukset

Agarwal et al. jakavat toiminnalliset palveluvaatimukset seuraaviin eri vaatimuksiin:

- resurssien näkyvyys (*Resource Discovery*)
- resurssien hallinta (*Resource Management*)
- tiedonhallinta (*Data Management*)
- tiedon keruu (*Data Acquisition*)
- tiedon käsittely (*Data Processing*)
- tiedon tallennus (*Data Storing*)
- tiedon visualisointi (*Data Visualization*)
- tiedon siivous (*Data Cleaning*)
- kyselyjen teko (*Query Processing*)
- tapahtumankäsittely (*Event Processing*)
- ohjelmakoodin hallinta (*Code Management*)

IoT yhdistää heterogeenisiä laitteita dynaamiseen ympäristöön. Täytyy löytyä jokin automatisoitu tapa julkaista, löytää ja tilata resursseja keskitetysti [2]. Tämän hoitamiseksi IoT alusta ylläpitää rekisterikomponenttia, johon mittalaitteet rekisteröityvät käyttäen siihen soveltuvaa rajapintaa (*Resource Discovery*). Rekisteröinnin jälkeen laitteet näkyvät verkossa ja rekisteröitynyt laite voi julkaista metatietoa palveluistaan. Muut laitteet voivat lisäksi hakea laiterekisteristä tarvitsemiaan palveluita ja resursseja. Palvelun laadun varmistamiseksi IoT-alustan tulee suorittaa resurssien hallintaa (*Resource Management*). Resursseja on seurattava, niiden jakamista suunniteltava ja niitä on kohdennettava oikeudenmukaisella ja konfliktittomalla tavalla alustalla. Tämän tehtävän suorittamiseksi alustojen on ylläpidettävä tietoja laitteiden akun kestosta, muistin käytöstä, prosessointitehosta ja tehokkaan resurs-

sienhallinnan kannalta muista olennaisista tiedoista. Tehokas resurssien hallinta takaa sen, että mittalaite, joka jo suorittaa useata tehtävää tai jonka virtalähde alkaa olla loppuillaan, saa suoritettavakseen vähemmän uusia pyyntöjä [10].

Tiedonhallinnalla (*Data Management*) on aina tärkeä rooli sovelluskehityksessä. IoT-maailmassa tiedolla tarkoitetaan sitä tietoa, jonka mittalaitteet havaitsevat. IoT-alusta tarjoaa tiedonhallintapalveluja, jotka sisältävät tietojen hankintaa, tietojenkäsittelyä, kyselyitä ja visualisointia. IoT-alustat keräävät tietoja (*Data Acquisition*) erilaisilta mittalaitteilta. Havainnoinnissa tieto on yleisesti ottaen muodoltaan ei-rakenteista tai puolirakenteista [9]. Datan lähteet ovat moninaisia, riippuen käyttötarkoituksesta. Tiedonhankinnan suunnitteluun liittyviä haasteita ovat tiedon kohdistaminen, skaalautuvuus, kokoonpano, turvallisuus, yksityisyys ja yhteensovittaminen [7]. IoT-alustojen vahvuutena on osata analysoida nopeasti ja tehokkaasti kerättyä tietoa ja kyky tehdä sen perusteella mielekkäitä johtopäätöksiä. Tiedon käsittely (*Data Processing*) ja analysointi tapahtuu yleensä kahdessa tilassa, reaaliaikaisesti tai striimattuna, ja eräkäsittelynä tai historiallisena analyysinä. Reaaliaikaisessa analyysissä oikea-aikaista, nopeasti vaihtuvaa tietoa tulee kyetä analysoimaan tarpeeksi tehokkaasti. Historiallinen analyysi käsittelee eräaikaisesti tallennettua historiallista dataa, ja tekee siitä päätelmiä.

Mittalaitteet tuottavat valtavan määrän eri muotoista tietoa jota IoT-alustan tulee tallettaa (*Data Storing*). Tiedon tallettamiseen käytetään tietotyypistä riippuen pääasiassa relaatio- ja NoSQL -tietokantoja. Relaatiotietokannat ovat tarkoitettu rakenteisen, transaktionaalisen tiedon ja NoSQL -tietokannat dynaamisesti vaihtuvamuotoisen tiedon tallentamiseen. Suosittuja NoSQL -tietokantoja ovat mm. Hadoop, CouchDB, HBases ja MongoDB. Mittalaitteilta saatu data voi olla epätäydellistä, epäjohdonmukaista, vääristynyttä, tai se voi olla merkityksetöntä. Järjestelmän luotettavuuden kannalta on tarpeen puhdistaa ja esikäsitellä tällaista dataa jatkoa ajatellen (*Data Cleaning*). Poikkeamien havainnointiin [9], normalisointiin, suodattamiseen ja semanttisen johdonmukaisuuden säilyttämiseen on kehitetty erilaisia algoritmeja ja proseduureja [12][35]. Jotkin alustat käyttävät strategioita, joissa haluttu data kerätään käyttämällä suodattimia, jotka suodattavat epäolennaisen datan pois [43]. Datan puhdistaminen myös vähentää tallennettavan datan määrää. Data visualisointitekniikat *Data Visualization* tarjoavat graafisen näkymän kyselyiden hakeaman tiedon analysoinnin tuottamista lopputuloksista. Saadut tulokset voidaan esit-

tää monella eri tapaa, kuten karttoina, raportteina tai esimerkiksi kojelaudan muodossa.

IoT-alustalle tallennettuun tietoon suoritetaan kyselyitä (*Query Processing*). Kyselyt voivat olla käyttötarkoituksesta riippuen yksinkertaisia tai monimutkaisia, useista tietokannoista tuloksia hakevia kyselyitä. Kyselyt voivat myös hyödyntää publish/-subscribe mekanisme [9].

IoT-alustojen tehtävänä on prosessoida tapahtumia sujuvasti (*Event Processing*). Tapahtuma on yksinkertaisimmillaan muutos ympäristössä jonka mittalaite on havainnut, ja joka vaatii tiettyjä toimenpiteitä IoT-alustalta. Usein tapahtuman käsittely tulee suorittaa reaaliaikaisesti. IoT-alustan tulisi antaa käyttäjälle mahdollisuus ohjelmoida omia ohjelmiaan erilaisille tapahtumille. Tapahtumankäsittelyn suurena haasteena on kerätä tietoa monilta sensoreilta samanaikaisesti eri muotoisena, yhdistää ne ja toteuttaa haluttu ratkaisu näiden tietojen pohjalta. Tapahtumat esitetään yleensä käytössä olevien ontologioiden muodossa metatiedoilla niiden oikeasta tulokinnasta. IoT-alusta on tärkeässä roolissa, kun ohjelmakoodia hallitaan ja levitetään IoT-sovelluksille (*Code Management*). Alusta tarjoaa erilaisia migraatiopalveluita tähän tarkoitukseen [10]. Koodin allokoinnilla huolehditaan siitä että oikea koodi suoritetaan oikealla mittalaitteella. Koodin migraatio huolehtii koodin siirtoon tarvittavien siirrettävien palveluiden tarjoamisesta.

3.2.2 Ei-toiminnalliset palveluvaatimukset

Agarwal et al. [1] jakavat ei-toiminnalliset palveluvaatimukset seuraaviin vaatimuksiin:

- skaalautuvuus (*Scalability*)
- oikea-aikainen toteutus (*Timeliness*)
- saatavuus (*Availability*)
- suojaaminen ja yksityisyyden turvaaminen (*Security and privacy*)
- jakelun helppous (*Ease of deployment*)

IoT-alustaan tulee voida liittää suuri määrä laitteita. Alustan on tarjottava ominaisuuksia, joiden avulla voidaan lisätä tai poistaa mikä tahansa määrä laitteita (*Scalability*). Laitemäärien suurenkaan vaihtelun ei tule heikentää palvelun laatua [10]. Tapahtumankäsittely riippuu pääasiassa tietojen oikea-aikaisesta toteutuksesta (*Timeliness*). Suurin osa tapahtumista edellyttää tosiaikaista käsittelyä. Tosiaikaisuus tarkoittaa tiedon käsittelyä nopeasti ilman viiveitä. Käsittelyyn tarvittava aika on kriittinen ja sen määrittävät ne sovellukset, jotka käsittelevät tapahtumia. Kriittisille sovelluksille IoT-alustan saatavuus tulee olla saatavilla kaikkina aikoina 24/7 periaatteella (*Availability*). Alustan sisäiset häiriötilanteet eivät saa vaikuttaa sen toimivuuteen, ja häiriöistä toipumiseen kuluvan ajan tulee olla mahdollisimman lyhyt, eikä se saa vaikuttaa systeemin suorituskykyyn. Sekä luotettavuus että saatavuus ovat tärkeitä tekijöitä puhuttaessa IoT-alustan vikasietoisuudesta.

Eräs tärkeä vaatimus IoT-alustalle on käyttäjätietojen suojaaminen ja yksityisyyden turvaaminen (*Security and privacy*). Monissa sovelluksissa käsitellään käyttäjien henkilökohtaisia tietoja kuten sijaintitietoja, salasanoja ym., ja näiden tietojen tulee olla suojattuna hyökkäyksiltä sekä tietoa siirrettäessä että sitä tallennettaessa. Riittävät menetelmät yksityisyyden ja turvallisuuden takaamiseksi tarvitaan sekä toiminnallisille että ei-toiminnallisille vaatimuksille [3]. IoT-alustoja käytetään loppukäyttäjien omien sovellusten levittämiseen, joita sovelluskehittäjät integroivat omille laitteilleen. Alustojen tulee siis olla käyttäjäystävällisiä, niiden tulee olla helposti asennettavia kokemattomienkin käyttäjien toimesta (*Ease of deployment*)

3.2.3 Arkkitehtuurivaatimukset

Agarwal et al [1] mukaan arkkitehtuurivaatimukset ovat seuraavat:

- ohjelmoinnin abstrahointi (*Programming Abstraction*)
- yhteistoiminta (*Interoperable*)
- palvelupohjaisuus (*Service Based*)
- adaptiivisuus, (*Adaptivity*)
- kontekstietoisuus (*Context Aware*)
- autonomisuus (*Autonomous*)

- hajautuneisuus (*Distributed*)

IoT-alustan sovelluskehityksen ohjelmointirajapinnan on pystyttävä piilottamaan järjestelmän sisäinen toiminta (*Programming Abstraction*). Käyttäjien erilainen kokemustaso on myös hyvä huomioida tarjottavassa rajapinnassa. Kokeneemmille käyttäjille tulee olla tarjolla enemmän mahdollisuuksia ohjelmointiympäristön muokkaamiseksi. IoT-alustojen tulee voida toimia yhteistyössä erilaisten laitteiden, teknologioiden ja sovellusten kanssa (*Interoperable*). Laitteiden tulee kommunikoida keskenään, vaihtaa dataa ja tehdä saumatonta yhteistyötä saavuttaakseen halutun lopputuloksen. Yhteentoimivuus voidaan saavuttaa verkkotasolla, semanttisella ja syntaktisella tasolla [11]. IoT-alustoille tulee olla mahdollista asentaa uusia palveluliittymiä erilaisiin sovelluksiin ilman että se vaikuttaa taustalla oleviin laitteistorajapintoihin (*Service Based*). Palvelupohjainen kehysarkkitehtuuri tarjoaa joustavasti kehityksen jonka päälle rakentaa eri tyyppisiä sovelluksia.

IoT-alustat toimivat yleensä dynaamisessa ympäristössä, ja niiden tulee mukautua muutoksiin ympäristössään. Tämä pitää huomioida niiden arkkitehtuuriratkaisuissa (*Adaptivity*). Kontekstitietoisuus antaa lisäarvoa datalle, jonka mittalaitteet havaitsevat ympäristöstään. Mittalaitteen tulee pystyä tavoittamaan käyttäjä- ja laitetietoja, jotta se osaa tehdä tarkoituksenmukaisia johtopäätöksiä datasta (*Context Aware*). Kaikkien IoT-ympäristön laitteiden täytyy hallinnoida itseään (*Autonomous*). Niiden tulee toimia autonomisesti muiden laitteiden kanssa, ilman että tarvitaan ihmisen väliintuloa [16][48]. Automaattisuus voidaan tarjota sulautetun älykkyyden, analytiikan ja itsenäisten agenttien kautta. Tukeakseen erilaisia jakelusovelluksia, kuten kuljetus- ja liikennesovelluksia, IoT -alustan on kyettävä tarjoamaan hajautettua, ei-keskitettyä käsittelyä (*Distributed*). Alustan on tuettava toimintoja, jotka voidaan suorittaa fyysisesti hajautetussa infrastruktuuriympäristössä.

3.2.4 Valintakriteereitä

Hejazi et al. [19] kiinnittävät IoT-alustan valinnassa huomiota seuraaviin piirteisiin: alustan elinkaaren jatkuvuus, joustavuus ja skaalautuvuus sekä hinnoittelu. Agarwal et al. [1] puolestaan ottavat seuraavat piirteet huomioon: vakaus ja saatavuus, jakelutapa, hinnoittelumalli, tarvittavien laitteistojen tuki, tietoturva vaatimukset, tuki eri kommunikaatioprotokollille, tiedon tallennustekniikat ja tuki datan analysointiin.

Markkinoilla on lukuisia pilvialustoja, ja on todennäköistä, että jotkut niistä kuihtuvat pois. On tärkeää, että valittu alusta on olemassa vielä vuosienkin päästä, ettei siihen investoitu panos mene hukkaan. Alustan tulee mukautua tulevaisuuden tarpeisiin. Aluksi saattavat riittää pienemmätkin resurssit, mutta liiketoiminnan kasvaessa resurssejakin tulee pystyä kasvattamaan tarpeen mukaan. Alustaan tulisi myös pystyä implementoimaan kivuttomasti uusimmat teknologiat. Sen pitäisi myös olla teknologiariippumaton eikä lukittautua vain tiettyyn, esimerkiksi verkkoteknologiaan. Hinnoittelumalliin tulee perehtyä tarkkaan. Aiemmin kustannukset olivat helpommin ennustettavia niiden muodostuessa suurelta osin kertaluontoisista pääomakustannuksista (capital expences). Nykyisin toimintakulut (operational expenses) ovat tärkeässä roolissa, ja palveluiden tarjoajilla on erilaisia hinnoittelumalleja [19]. Tapa hinnoitella IoT-alustapalvelut vaihtelevat eri palveluntarjoajien kesken. Hinnoittelu voi perustua esimerkiksi käyttöön, tietovaraston kokoon, yhdistettyjen laitteiden määrään, tai se voi olla esimerkiksi tilauspohjaista. Joissain pienen mittakaavan sovelluksissa jotkut toimittajat voivat tarjota tietyn määrän vapaata tallennustilaa. Jälleen paras vaihtoehto on se, joka soveltuu omiin käyttötarkoituksiin parhaiten.

Alustan vakaus ja saatavuus ovat tärkeitä kriteereitä, varsinkin kun puhutaan kriittisistä järjestelmistä joiden pitää olla käytettävissä jatkuvasti 24/7. Esimerkiksi potilaan tilaa monitoroivat sairaalasovellukset ovat tällaisia. Avoimen lähdekoodin alustat mahdollistavat sovelluskehittäjille alustojen muokkaamisen omien tarpeiden mukaan, kun taas kaupallisten IoT-alustojen hallinta tapahtuu pääsääntöisesti palveluntarjoajan toimesta. Riippuu paljon kehitettävästä sovelluksesta, haluaako kehittäjä mahdollisuuden muokata alustaa mieleisekseen vai antaako palveluntarjoajan huolehtia IoT-alustan perustana olevista tekniikoista. Useimmat väliohjelmistoalustat käyttävät pilvitekniikkaa tallennustarkoituksiin. Resurssitarpeen mukaan kehittäjä voi valita eri pilvipalvelumalleista - Infrastructure as a Service (IaaS), Platform as a Service (PaaS) tai Software as a Service (SaaS). Markkinoilla on tarjolla lukuisia erilaisia IoT-kehitysalustoja, ja jokaisella niistä on omia ominaispiirteitään. IoT-kehitysalustat eroavat toisistaan tyypillisesti prosessorien, GPU:n, kelloaajuuksien koon ja RAM muistin suhteen. IoT-pilvialustan tulee tukea käytössä olevia laitekomponentteja. Vaadittava tietoturvan ja suojausten taso vaihtelee eri sovelluksissa. Esimerkiksi pankkisovellukset vaativat korkeampaa tietoturvaa kuin vaikka-

pa älykkään maatalouden sovellukset. Eri alustat tarjoavat erilaista tietoturvan tasoa. Jotkut alustat sallivat sovelluskehittäjien toteuttaa omia suojausalgoritmejaan. IoT-alustan tarjoamalla tietoturvan tasolla voi olla suuri merkitys valittaessa omiin tarkoituksiin sopivaa alustaa.

Tarjottavat kommunikaatioprotokollat vaihtelevat eri IoT-alustoilla. Jotkut ovat kevyempiä, toiset tarjoavat enemmän turvallisuutta. CoAP on samankaltainen kuin HTTP mutta kevyempi, ja soveltuu näin paremmin mobiilisovelluksille. CoAP soveltuu hyvin multicast ja broadcast viestintään. MQTT on samoin kevyt ratkaisu, joka tukee välittäjä-konseptia tehden siitä soveltuvan sovelluksille, joilla on käytävissä rajoitettu kaistanleveys kommunikointiin. IoT-alustapalvelujen tarjoajat käyttävät pääasiassa pilveä tallennustilanaan. Eri pilvipohjaiset tallennus- ja käsittelytekniikat tukevat erityyppisiä analytiikkoja. Tietojen käsittelyvaatimusten mukaan voidaan valita IoT-alusta sopivalla datan tallennustekniikalla. IoT-sovellukset tuottavat yleensä tosiaikaista/striimattua tai historiadataa. Sovelluksen tuottaman datan analysointiin tarvitaan oikeanlaista teknologiaa, ja eri alustat eroavat tarjoamissaan teknologioissa toisistaan.

4 Katsaus IoT pilvialustoihin

Erilaisia pilvialustoja alkaa olla markkinoilla jo runsain määrin. Alustoja ei ole syytä yrittää laittaa paremmuusjärjestykseen, koska tietyn IoT-alustan sopivuus omiin tarpeisiin on aina se valinnan ratkaiseva tekijä. Vertailua voi kuitenkin suorittaa tiettyjen kriteerien perusteella. Olemassa olevat vertailut helpottavat omaan käyttöön parhaiten sopivan IoT-palvelualustan valinnassa.

Tässä työssä tehdään katsaus muutamaanki mielenkiintoiseen IoT-alustaan. Agarwal et al. [1] jakavat alustat nelikenttämalliin, jonka kenttinä ovat *kaupalliset IoT-alustat*, *avoimen lähdekoodin IoT-alustat*, *sovelluskehittäjäystävälliset välikerroksen alustat* ja *päästä päähän yhteydellä varustetut alustat*. Yleisemmin ja pelkistetyemmin mallit jaetaan kaupallisiin ja avoimen lähdekoodin IoT-alustoihin. Tähän katsaukseen on valittu mukaan avoimen lähdekoodin alustoista *Kaa*, *Temboo*, *Thingspeak*, ja kaupallisista alustoista *AWS IoT Platform*, *Google Cloud Platform*, *IBM Watson IoT Platform* ja *Microsoft Azure IoT*. Katsauksessa kiinnitetään huomiota mm. seuraaviin Aragwal et alin tunnistamiin vertailukriteereihin: tiedon hallinta, tiedon tallennus- ja saantitapa, tuettu dataformaatti, ratkaisun skaalautuvuus, tuetut ohjelmointikielet [2], saatavuus (onko jatkuva 24/7) ja hinnoittelumalli [1].

4.1 Avoimen lähdekoodin IoT alustat

Avoimen lähdekoodien IoT pilvialustoja on olemassa lukuisia. Ratkaisuista löytyy sekä kokonaan maksuttomia että kaupallisia sovelluksia. Yhdistävänä tekijänä on mahdollisuus tehdä avoimeen koodiin omia komponentteja, joko omaan käyttöön tai julkaistavaksi yleiseen käyttöön. Avoimen lähdekoodin IoT pilvialustojen palvelumalli on yleensä PaaS. Avoimen lähdekoodin alustoista valikoitui lyhyesti esiteltäviksi *Kaa*, *Temboo* ja *ThingSpeak*. Taulukkoon 4.1 on koottu muutamia avoimen lähdekoodin alustojen keskeisiä ominaisuuksia.

4.1.1 Kaa

KAA (<https://www.kaaiot.com/>) on monikäyttöinen pilvipohjainen IoT-alusta älykkäiden, end-to-end IoT-ratkaisujen toteutukseen. Kaa on mikropalveluarkkitehtuuriin pohjautuva IoT-alusta, jossa jokainen Kaan mikropalvelu on itsenäinen kokonaisuus. Sovellukset rakennetaan hyödyntäen tarvittavia mikropalveluita. Kaan hinnoittelu perustuu pilvipalvelussa käytössä olevien laitteiden määrään, ja pilvipalvelumallina on PaaS [21].

Kaan ydinominaisuuksia ovat laitehallinta (*Device management*), kommunikointi (*Communication*), tiedon keruu (*Data collection*), konfiguraatiohallinta (*Configuration management*), komentokutsut (*Command invocation*), ohjelmistopäivitykset (*Software updates*), visualisointi (*Visualization*), infrastruktuuri (*Infrastructure*) ja muut lisätoiminnot (*Miscellaneous*) [22].

Laitehallinnan avulla pidetään kirjaa laitteista, hallinnoidaan laitteiden käyttöoikeustietoja, hallinnoidaan laitteiden metatietoja ja ryhmitellään laitteita metatietojen mukaan. Kommunikaation päätoiminnallisuudet ovat tiedonsiirron toteuttaminen laitteiden ja Kaa-alustan komponenttien välillä, yhdistettävien laitteiden (asiakkaat ja päätelaitteet) todentaminen, laitteiden ominaisuuksien tunnistaminen, tiedonsiirron turvaaminen ja laitteen yhteystilan (laite yhdistetty/yhteys pois) hallinta.

Kaan tiedonkeruu- ja tallennusominaisuus on suunniteltu keräämään dataa yhdistetyiltä päätelaitteilta ja lähettämään ne erilaisiin datan tallennus- ja käsittelyjärjestelmiin. Tyypilliset tietotarpeet ovat aikasarjadataa, telemetriatietoja, laiteohjelmistoa ja ohjelmistodataa sekä lokitietoja ja hälytyksiä. Kaan konfiguraatiohallinta ominaisuus auttaa hallitsemaan yhdistettyjen laitteiden toimintaa ylläpitämällä konfigurointitietoja.

Kaan komentojen kutsutoiminto auttaa kontrolloimaan laitteita etänä. Ominaisuuden avulla voidaan päätelaitteille lähettää lyhytikäisiä viestejä, saada laitteet käyttäytymään halutusti ja lähettämään tarvittaessa vastausviestejä. Kaa-ohjelmistopäivitysominaisuus auttaa jakamaan ohjelmistopäivityksiä verkon yli langattomasti yhdistettyihin laitteisiin. Kaa Web Dashboard -mikropalvelu toteuttaa verkkokäyttöliittymän mm. kojelaudan ja tietojen visualisointiin, laitehallintaan, päätelaitteiden määrittysten hallintaan, komentojen suorittamiseen ja Kaa IoT-alustan hallin-

taan.

Kaa hyödyntää useita kolmannen osapuolen infrastruktuurikomponentteja alustan toiminnan mahdollistamiseksi ja helpottamiseksi. Nämä sisältävät mm. NATS-viestivälittäjän, InfluxDB- ja MongoDB-tietokannat, Kubernetes-konttiorkestrointijärjestelmän, Helmin, NGINX:n, Prometheusin, Fluentdin ja Grafanan. Näiden kolmannen osapuolen komponenttien lisäksi Kaa sisältää omat infrastruktuurikomponentit, jotka yksinkertaistavat Kaa-pohjaisten klustereiden toimintaa ja tarjoavat Kaan ratkaisukohtaisia hallintatoimintoja. Kaan tukemia protokollia ovat mm. HTTP, MQTT ja CoAP. Kaa-alustassa on lisäksi mikropalveluita, jotka toteuttavat erilaisia lisätoimintoja. Yksi tällainen mikropalvelu on Time Series Extension (TSX), jonka avulla päätelaitteet voivat pyytää aikaleimaa Kaa-palvelimelta synkronoidakseen sisäisen kellonsa.

4.1.2 Temboo

Temboo (<http://temboo.com>) on avoimen lähdekoodin yksityinen pilvipohjainen ohjelmakehitysalusta. Temboossa on tarjolla valmiita ratkaisuja ympäristön havainnointiin, ja ratkaisut ovat helposti otettavissa käyttöön. Temboo soveltuu paremmin vähän resursseja vaativien sovellusten IoT-alustaksi [37]. Temboo autogeneroi tarvittavat ohjelmakoodit valmiiksi kehittäjille, tehden Temboosta käyttäjystävällisen pilvialustan [44]. Laitteita ja niiden etähallintaa, työnkulkuja, prosesseja ja datan suodatusta voidaan konfiguroida visuaalisesti Web portaalissa. IoT-ratkaisut ovat nopeasti kehitettäviä ja kaupallistettavia.

Temboon hinnoittelu perustuu kolmeen hinnoittelumalliin ja lisäksi asiakaskohtaiseen räätälöityyn hinnoitteluun [44]. Saatavilla on myös ilmaisversio, joka tarjoaa rajoitetun määrän Temboon palveluita. Pilvipalvelumallina on PaaS.

Temboossa on yli 90 sisäänrakennettua Choreos kirjastoa, jotka tarjoavat helppokäyttöiset rajapinnat kolmansien osapuolten palveluihin. Näitä palveluita ovat esimerkiksi Yahoo weather, Amazon cloud, Flickr photo management, Facebook Graph API, Google analytics ja Twitter micro blogging [37]. Temboossa tuetut tietomuodot ovat Excel, CSV, XML ja JSON, ja ohjelmointikielinä on Android, C, Java, Python ja Javascript. Temboon kanssa yhteensopivia laitteita ovat Texas Instrumentsin, Arduinin ja Samsung Artikin tuotteet, ja käytettyjä teknologioita ovat Microsoft Power BI

ja Google BigQuery.

Käytettävät protokollat datan keräämiseen ovat MQTT ja CoAP, ja integraatorajapintana on REST API. Temboossa ei itsessään ole datalle tietovarastoa, mutta datan varastointiin on useita ulkoisia vaihtoehtoja, mm. Choreos rajapinnat Google BigQueryyn ja NoSQL:ään [44]. Temboo ei myöskään sisällä ratkaisuja datan visualisointiin, mutta se generoi visualisointiin tarvittavan ohjelmakoodin kolmansien osapuolten ratkaisuille. Tähän sillä on SDK:t (Software Development Kit) useille ohjelmointikielille.

4.1.3 ThingSpeak

ThingSpeak (<https://thingspeak.com>) on julkiseen pilviteknologiaan pohjautuva avoimen lähdekoodin IoT-alusta. ThingSpeak mahdollistaa tosiaikaisen tiedon keräämisen, analysoinnin ja käytön avoimen rajapinnan välityksellä. Käyttäjätietoa tallennetaan, visualisoidaan, monitoroidaan ja integroidaan kolmansien osapuolten alustojen kanssa ThingSpeakin sovellusten ja laajennosten avulla. Kolmansien osapuolten ratkaisuja ovat mm. ioBridge, Arduino, Twilio, Twitter, ThingHTTP ja MATLAB [37]. ThingSpeak on sopiva pilvialusta ennenkaikkea pieniin sovelluksiin, joissa laitekomponentteja on rajoitetusti, eikä niiden ylläpito tai sovelluksen skaalautuvuus ole niin merkitsevässä roolissa. ThingSpeak-käyttäjällä olisi hyvä olla ainakin jonkin verran ohjelmointikokemusta [34].

ThingSpeakin hinnoittelu perustuu lisensseihin. Alusta on ilmainen pienille, ei-kaupallisille sovelluksille, ja kaupalliseen käyttöön on valittavissa neljä lisenssimallia. Pilvipalvelumallina on PaaS [46]. Käytettyjä protokollia ovat sovellusten REST-rajapinnat ja HTTP ja MQTT laitteiden väliseen kommunikointiin. Mittalaitteen tuottama data, joka voi olla minkä tyyppistä tahansa, kerätään ThingSpeakissa kanaviin, joiden tallennusjärjestelmä on aikasarjatietokanta. Kanavien sisältöä voidaan käyttää haluttuun tarkoitukseen (laskenta, analysointi, esittäminen) eri sovellusten avulla.

Tuetut tietomuodot ovat ThingSpeak API, JSON ja XML, ja ohjelmointialustana on MATLAB. Yhteensopivia laitteita löytyy valmistajilta Arduino, Particle photon, ESP8266 wifi ja Raspberry Pi.

Taulukko 4.1: IoT-alustojen ominaisuuksia

Avoimen lähdekoodin alustat			
Alusta	Kaa	Temboo	ThingSpeak
Saatavuus (24/7)	-	-	-
Pilvipalvelumalli	Paas	Paas	Paas
Hinnoittelu	Laitteiden määrän mukaan	Eri tasoisia lisenssejä, räätälöity malli	Ilmaisversio ja eri tasoisia lisenssejä
Skaalautuvuus	Heikko	Heikko	Heikko
Protokollat	HTTP, MQTT, CoAP	HTTP, MQTT, CoAP	HTTP, MQTT
Tuetut ohjelmointikielet	Java, C, C++	C, Java, Python, Android, Javascript	Matlab, Ruby, Python, Node.js
Tietoturva	Salaus, auditointi	-	Salaus
Tiedon tallennus	NoSQL, Cassandra, Hadoop, MongoDB	Microsoft Power BI, Google BigQuery	Matlab, dashboard
Tietomuodot	REST API, JSON	Excel, CSV, XML, JSON	ThingSpeak API, JSON, XML
Tiedon esitys	Web Dashboard (WD)	SDK:t, Kosmos IoT System	MATLAB

Taulukkoon 4.1 on koottu avoimen lähdekoodin IoT-alustojen ominaisuuksia. Taulukosta voidaan esimerkiksi todeta, että avoimen lähdekoodin alustojen pilvipalvelumalli on kaikilla vertailuilla PaaS, skaalautuvuus heikohko ja tuetut protokollat suunnilleen samoja. Eroja syntyy esimerkiksi tiedon esitysratkaisuissa, tiedon tallennusvälineissä ja tuetuissa ohjelmointikielissä.

4.2 Kaupalliset IoT alustat

Kaupallisten IoT pilvialustojen palvelutarjonta on runsasta, koska niitä tarjoavat yritykset ovat alan suurimpia. Ratkaisujen kustannukset kohoavat luonnollisesti sitä mukaa kun palveluja ja käytettyjä resursseja lisätään, mutta useimmilla kaupallisilla alustoilla on myös ilmaisia palveluita rajoitetussa muodossa. Tässä käydään läpi neljä isoa vaihtoehtoa, AWS IoT Platform, Microsoft Azure IoT Platform, Google Cloud Platform ja IBM Watson IoT Platform. Taulukkoon 4.2 on koottu kaupallisten alustojen keskeisiä ominaisuuksia.

4.2.1 AWS IoT Platform

AWS IoT Platform (<https://aws.amazon.com/iot/>) on Amazon Web Servicen kehittämä kaupallinen Iot-alusta [6]. AWS-ratkaisuun voidaan liittää lähes rajattomasti IoT laitteita, ja yhdistäminen toteutetaan tietoturvallisen yhdyskäytävän läpi. AWS IoT-alustaan on integroitu useita yhdessä toimivia välikerroksen teknologioita. Lukuisat AWS:n sisäänrakennetut teknologiat tukevat heterogeenisten laitteiden yhdistämistä, datan taltiointia ja tietoturvallista tiedonsiirtoa pilvessä. AWS:ssä on laajat tietoturvaominaisuudet ja runsaasti analytiikkatyökaluja. Myös monia kolmansien osapuolten sovelluksia varten on tuki AWS:ssä. AWS:ssä on valmiita ja helposti käyttöönotettavia sovellusmalleja eri toimialoille, ja näitä malleja voidaan muokata omaan käyttöön vähällä vaivalla.

AWS IoT Platformin hinnoittelumalli perustuu pay-as-you-go -malliin, jossa palvelut hinnoitellaan käytettyjen resurssien mukaan. Pilvipalvelumalleina ovat IaaS ja PaaS. AWS tukee muun muassa protokollia MQTT, HTTP ja Websockets. Datan tallennustapoja on AWS:n aikasarjatietokannat, ja dataformaatti JSON. Datan tallennusratkaisuja ovat Dynamo DB ja NoSQL -tietokannat

AWS IoT Platform tukee tunnettuja ohjelmointikieliä kuten Java, Javascript, NodeJS, C, Python. Yhteensopivia laitteita valmistaa muun muassa Broadcom, Marwell, Renesas, Texas Instruments ja Microchip Intel. Tiedon visualisointiin on muun muassa AWS IoT Analytics console ja Amazon QuickSight. AWS:n saatavuus on (24/7). AWS on yksi johtavista palveluntarjoajista, ja sillä on erittäin laaja palveluvalikoima. Toisaalta se voi olla myös melko kallis vaihtoehto.

4.2.2 Microsoft Azure IoT Platform

Microsoft Azure IoT on Microsoftin kehittämä kaupallinen IoT-alusta, jonka markkinaosuus on yksi suurimmista [30]. Sen katsotaan olevan ainoa hybridi pilvipalveluratkaisu, ja sen tekoälyominaisuudet ovat pitkälle kehittyneitä. Azure IoTissa on lukuisia palveluita, työkaluja ja teknologioita kerätyn datan turvaamiseen, varastointiin ja prosessointiin, ja se tukee esikonfiguroituja ratkaisuja [30].

Azure IoT tarjoaa pilvipohjaiset taustajärjestelmät, luotettavan tietoturvan ja laitteiden välisen kommunikaation. Keskeisistä palveluista *Azure IoT Hub* on yhteensopiva *Azure IoT Edgen* ja *Azure Stackin* kanssa, joka mahdollistaa hybridisovellusten toteuttamisen, ja täysi integrointi *Azure Event Gridin* ja palvelimettoman laskennan kanssa yksinkertaistaa IoT -sovellusten kehittämistä. Azure IoT Hub käyttää laitekohtaista autentikointia. Device Update for IoT Hub mahdollistaa päivitysten langattoman siirtämisen ja asennuksen IoT-laitteisiin, jolloin IoT-laitteet pysyvät ajan tasalla ja suojattuina.

Azure IoT Hub tukee HTTP, AMQP, AMQP over WebSockets, MQTT ja MQTT over Websockets protokollia. Jos mikään edellä mainituista protokollista ei sovellu, voidaan protokollaa kustomoida *Microsoft Azure IoT Protocol Gatewayn* avulla tai Azure IoT Edge laajennuksella. Tuettu tietomuoto on JSON, ja Azure IoT kanssa tuettuja ohjelmointikieliä ovat C, Embedded C, C#, Java, Python, Node.js ja UWP. Tietovarastoratkaisuja ovat muun muassa Azure Cosmos DB, Azure SQL DB ja Azure Tables.

Yhteensopivia laitteita toimittavat mm. Intel, Raspberry, FreeScale ja Texas Instrument. Tietoa voidaan visualisoida esimerkiksi Microsoft Power BI raportointityökalulla. Azure IoT Platformin hinnoittelu on yhdistelmä palvelutasoista, joissa on kasvava laitteiden ja viestien määrä, lisäpalveluista ja pay-as-you-go -mallista, jossa kustannukset perustuvat käytettyjen resurssien määrään [30]. Pilvipalvelumalleja ovat IaaS, PaaS, aPaaS ja SaaS. Microsoft lupaa Azure IoT Platformille 99,9%:n palvelutasosopimuksen (SLA, Service Level Agreement). Tämän tason SLA takaa sen, että IoT alusta on toiminnassa 99,9% ajasta.

4.2.3 Google Cloud Platform

Google Cloud Platform (GCP) (<https://cloud.google.com/solutions/iot/>) on Googlen kehittämä kaupallinen IoT-alusta [15]. Se kykenee yhdistämään erityyppisiä laitteita ja tukee myös kevytrakenteisempia sovelluksia. Google Cloud Platformin katsotaan olevan palveliton (serverless) arkkitehtuuri, jossa taustasovelluksen tarvitsema määrä palvelimia otetaan käyttöön aina tarpeen mukaan. Tämä mahdollistaa sovellusten sujuvan skaalautuvuuden. Google Cloud Platformissa on tuki useille analytiikkakirjastoille, joista mainittakoon TensorFlow, ohjelmistokirjasto koneoppimiseen ja tekoälyyn. Google Cloud Platform käyttää publish-subscribe mekanismia laitteiden tunnistamiseen ja ylläpitää yhdistetyistä laitteista laiterekisteriä. Alustassa on tuki mobiilisovelluksille [2].

Google Cloud Platformin hinnoittelumalli perustuu pay-as-you-go malliin, jossa maksut määräytyvät käytettyjen tietovarastojen mukaan. Pilvipalvelumallina on IaaS. Datan keräämisen protokollat ovat HTTP ja MQTT. Käytettyjä datateknologioita ovat Googlen BigData tool, Riptide IO, BigQuery, Firebase ja PubSub. Google Cloud Platformin tietomuoto on JSON. Data tallennetaan relaatiotietokantoihin kuten MySQL, PostgreSQL tai SQL Server, tai BigQuery Data warehouse-tietovarastoon.

Tuettuja ohjelmointikieliä on Go, Java, .NET, Node.js, php, Python ja Ruby. Yhteensopivia laitteita valmistavat Raspberry Pi. Tiedon visualisointiin on tarjolla mm. Googlen Data Studio. Google Cloud Platformin saatavuus on (24/7).

4.2.4 IBM Watson IoT Platform

IBM Watson IoT Platform (<https://internetofthings.ibmcloud.com>) on IBM:n kehittämä kaupallinen IoT-alusta, joka tunnettiin aiemmin nimellä IBM Bluemix [20]. Helppokäyttöiseen ja tietoturvalliseen IoT-alustaan voidaan kytkeä laitteita yhdestä aina lukuisia mittalaitteita sisältäviin laajoihin ja kompleksisiin teollisiin ratkaisuihin. Laitteet kuten ARM mbed, Arduino, Intel Galileo, yhdistetään pilveen käyttämällä kevyttä MQTT protokollaa [37]. Jokaiselle laitteelle annetaan yksilöivä id-tunniste. Datan tallennukseen käytetään IBM:n hajautettua tietokantapalvelua Cloudant NoSQL.

IBM Watson IoT Platformin hinnoittelu perustuu laitteiden ja tietoliikenteen mää-

riin ja käytetyn tietokannan kokoon. IoT alusta pilvipalvelumallit ovat PaaS ja IaaS, ja tuettu protokolla on MQTT. Datan tallennus tapahtuu Cloudant NOSQL DB - tietokantaan. Tuetut tietomuodot ovat JSON ja CSV, ja ohjelmointikielina on C#, C, Python, Java ja Node.js.

Yhteensopivia laitteita valmistaa Broadcom, Marvell, Renesas, Texas Instruments ja Microchip Intel. Tiedon visualisointiin on tarjolla muun muassa IBM Watson Studio. IBM Watson IoT Platform saatavuus on (24/7), mikä on yleistä kaupallisille alustoille.

Taulukkoon 4.2 on koottu kaupallisten IoT-alustojen ominaisuuksia. Taulukosta nähdään että kaupallisten alustojen ominaisuudet ja kyvykkyydet ovat korkealla tasolla, eivätkä ne niissä poikkea juurinkaan toisistaan. Suurimmat erot syntyvät esimerkiksi tiedon tallennukseen ja visualisointiin käytetyistä työkaluista, joissa jokaisella tarjoajalla on omat ratkaisunsa.

Taulukko 4.2: IoT-alustojen ominaisuuksia

Kaupalliset alustat				
Alusta	AWS IoT Platform	Azure IoT Platform	Google Cloud	IBM Watson IoT Hub
Saatavuus (24/7)	K	K	K	K
Pilvipalvelumalli	PaaS, IaaS	IaaS, PaaS, aPaaS, SaaS	PaaS, IaaS	PaaS, IaaS
Hinnoittelu	Käyttöön perustuva	Lisenssit	Käyttöön perustuva	Käyttöön perustuva
Skaalautuvuus	Hyvä	Hyvä	Hyvä	Hyvä
Protokollat	MQTT, HTTP, Websockets	AMQP, HTTP, MQTT	MQTT, HTTP	HTTP, MQTT
Tuetut ohjelmointikiel	Java, Javascript, NodeJS, C, Python, Arduino, iOS	C, Embedded C, C#, Java, Python, Node.js	Go, Java, .NET, Node.js, php, Python ja Ruby	C#, C, Python, Java, Node.js
Tietoturva	Salaus, autentikointi, auktorisointi, auditointi	Salaus, autentikointi, auktorisointi, auditointi	Salaus, autentikointi, auktorisointi, auditointi	Salaus, autentikointi, auktorisointi, auditointi
Tiedon tallennus	Amazon Dynamo DB	Azure CosmosDB, Azure Tables, SQL database	Googlen BigData tool, BigQuery	Cloudant NOSQL DB
Tietotyypit	JSON	JSON	JSON	JSON, CSV
Tiedon esitys	AWS IoT Analytics console, Amazon QuickSight	Power BI	Data studio	IBM Watson Studio

5 Microsoft Azure IoT

Tämän työn empiirisen osuuden toteutusratkaisuksi valittiin Microsoft Azure IoT, kaupallinen IoT-alusta, jota käsitellään tarkemmin tässä luvussa. Luvun pääasiallisena lähteenä on käytetty Azuren IoT-dokumentaatiota [31], ellei lähde ole erikseen mainittu.

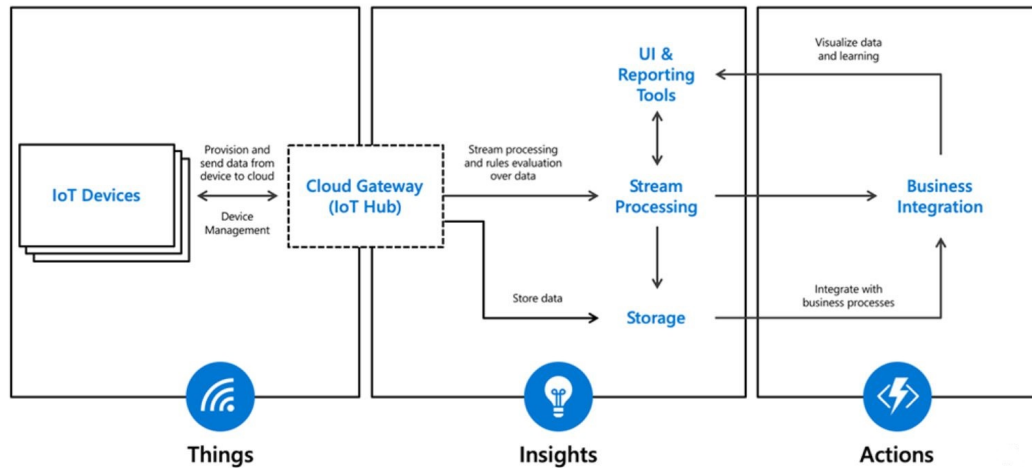
Microsoft Azure IoT on Microsoftin kehittämä kaupallinen IoT-alusta, jonka markkinaosuus on yksi suurimmista. Sen katsotaan olevan ainoa hybridi pilvipalveluratkaisu, jonka tekoälyominaisuudet ovat pitkälle kehittyneitä. Microsoft Azure IoT sisältää runsaan määrän Azure-palveluita (kuva 5.2), joita voidaan hyödyntää IoT-ratkaisujen arkkitehtuurissa. Azure IoT Platform tukee laajaa valikoimaa laitteita, muiden muassa teollisuuslaitteet, mikro-ohjaimet ja mittalaitteet eri tarkoituksiin. Azure IoT ratkaisuarkkitehtuuri (kuva 5.1) soveltuu useille eri liiketoiminta-aloille.

5.1 Azure IoT palveluita

Azuressa on lukuisia IoT palveluita, joista saadaan koottua haluttu IoT-kokonaisuus (kuva 5.2). Seuraavassa kuvataan muutamia keskeisiä niistä.

Kaksi ylätasoa palvelua ovat IoT Hub ja IoT Central. Kaikissa IoT-ratkaisuissa on oltava jompikumpi näistä palveluista [31]. Azuren IoT tietoturvapalveluja ovat Azure Sphere, Azure Device Provisioning Service (DPS) ja Azure Security Center for IoT. Azure IoT Edge on palvelu reunalaitteille, ja Azuren data-analytiikkapalveluita ovat Azure Time Series Insights, Azure Maps, Azure Digital Twins ja Azure Stream Analytics. Työnkulkua voidaan tehdä *Azure Logic Apps*-palvelulla. Azure tarjoaa kokeneemmille sovelluskehittäjille useita SDK:ta (Software Development Kit). Esimerkiksi Azure IoT Hubin SDK:t ovat Azure IoT Device SDK, Azure IoT Service SDK, Provisioning Device SDK ja Provisioning Service SDK.

Azure IoT Hub on pilvessä isännöity hallittu palvelu, joka yhdistää, seuraa ja hallitsee miljoonia IoT-resursseja. Azure IoT Hub toimii keskuksena kahdensuuntaisessa



Kuva 5.1: Microsoft Azure IoT arkkitehtuuri [31]

viestinnässä IoT-sovelluksen ja sen hallinnoimien laitteiden välillä. Azure IoT Hub tarjoaa luotettavan ja suojatun viestinvälityksen miljoonien IoT-laitteiden ja pilvipalvelun taustaratkaisujen välillä. Azure IoT -keskittimet on suunniteltu käsittelemään miljardeja telemetriaviestejä miljoonilta laitteilta. IoT Hub ylläpitää mittareita, laitteita monitorointiin ja sen avulla voidaan määrittää hälytyksiä, joita voidaan käyttää toimintojen käynnistämiseen. Nämä toiminnot vaihtelevat esimerkiksi varoitussähköpostin lähettämisestä automaattisen Azure-toiminnon käynnistämiseen. IoT Hubiin voidaan liittää monia muita erikoisominaisuuksia, kuten reititys, tallennus, automaattinen laitteiden käytön esivalmistelu, aikasarjatiedot ja koneoppiminen.

Azure IoT Central on Azuren isännöity IoT sovellusalusta joka on turvallinen, skaalautuva ja integroitavissa olemassa oleviin järjestelmiin. Azure IoT Centralin avulla sovelluksia voidaan asentaa vähemmällä vaivalla, se vähentää hallintataakkaa ja toimintakustannuksia ja yleiskustannuksia.

Azure IoT Hub on PaaS-pohjainen palvelu, joka antaa laajemmat mahdollisuudet hallinnoida siihen kiinnitettyjä laitteita ja muita palveluita. Azure IoT Central on taas SaaS-pohjainen palvelu, jonka avulla ratkaisun toteuttaminen on yksinkertaisempaa, mutta jonka laitteiden hallintaominaisuudet ovat vähäisempiä. Azure IoT Hub antaa siis enemmän työkaluja hallita IoT-kokonaisuutta, kun taas IoT Central

madaltaa kynnystä toteutuksen aloittamiseen.

Azure Stream Analytics on analytiikka ja tapahtumankäsittelymoottori, joka on suunniteltu prosessoimaan ja analysoimaan suuria datamääriä useista samanaikaisista lähteistä. Datan lähteet ovat moninaisia, mittalaitteista sosiaalisen median tietovirtoihin ja erilaisiin sovelluksiin. Saatua tietoa voidaan analysoida, siitä voidaan hakea erilaisia malleja ja käyttäytymiskuvioita, ja tiedon perusteella voidaan mm. käynnistää työkulkuja. Datasta voidaan koostaa raportteja, ja sitä voidaan tallentaa eri tietovälineille myöhempää käyttöä varten. Stream Analytics tukee myös Edge computingia, tiedon prosessointia reunalaitteissa.

Azure Logic Apps on pilvipohjainen palvelu, jolla luodaan ja suoritetaan automatisoituja työkulkuja, jotka integroituvat sovelluksiin, dataan ja muihin palveluihin ja järjestelmiin. Azure Logic Appsilla voidaan esimerkiksi ajoittaa ja lähettää sähköposti-ilmoituksia Office 365:llä, kun tietty tapahtuma tapahtuu, esimerkiksi uusi tiedosto ladataan. Sillä voidaan reitittää ja käsitellä asiakkaiden tilauksia paikallisissa tilausjärjestelmissä ja pilvipalveluissa, siirtää tiedostoja SFTP- tai FTP-palvelimelta Azure Storageen ja esimerkiksi hakea dataa ulkoisilta nettisivuilta järjestelmän hyödynnettäväksi, kuten tämän työn empiirisessä osassa tullaan tekemään. Azure Logic Apps työkulku voidaan käynnistää esimerkiksi HTTP-kutsulla tai käynnistää se ajastetusti.

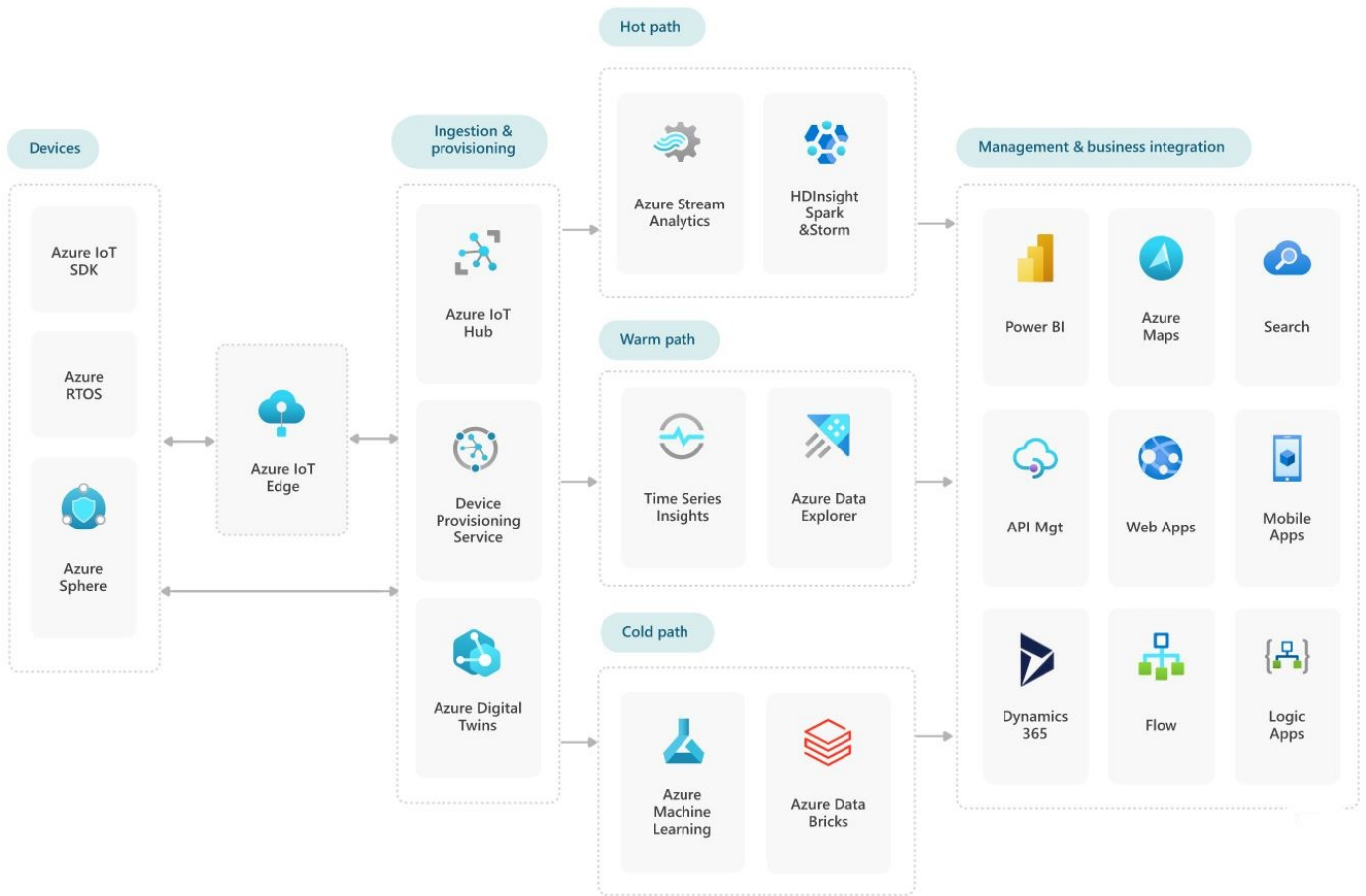
Azure Sphere sertifioidut mikrokontrollerit varmistavat, että viestintä laitteiden ja palvelinten välillä on suojattua. Azure Spheren nojalla laitevalmistajat voivat valmistaa turvallisia laitteita, joita voidaan päivittää, ohjata, valvoa ja ylläpitää myös etänä. Laitteita voidaan käyttää suojatun sovellusympäristön luomiseen. *Azure Device Provisioning Service (DPS)* provisioi (autentikoi) laitteet automaattisesti, ilman että ihmisen tarvitsee puuttua provisiointiin. Tämä tekee provisioinnista vähemmän viriheherkkää. Laitteet saavat maailmanlaajuisen yksilöivän tunnisteiden. *Azure Security Center for IoT* tarjoaa kokonaisvaltaisen, yhdestä portaalista hallinnoidun näkyvän IoT-ratkaisun tietoturva-aiheisiin.

Reunalaskennalla (*Edge Computing*) [41] datan käsittelyä voidaan jakaa ja siirtää pilvestä verkon reunalle. Reunalaitteena voi olla esim. yhdyskäytävä, joka kerää sensoreilta dataa ja suorittaa niille mahdollisesti laskentaa ennen kun välittää prosessoi-

dun datan IoT-keskittimelle. *Azure IoT Edge* tarjoaa tämän palvelun Azuressa. Palvelu on tarkoitettu asiakkaille, jotka haluavat esimerkiksi analysoida suurimman osan laitetiedoista paikallisesti pilven sijaan. Reunalaskennalla saadaan myös vähennettyä verkon yli lähetettävän datan määrää.

Azure IoT Hubissa ja Azure Event Hubissa voi olla miljardeja tapahtumia, joista kerätyn tiedon analysointiin Azurella on tarjolla useita palveluita. *Azure Time Series Insights* tarjoaa kyvykkyyden tarkastella tapahtumien poikkeavuuksia, joita voi muuten olla vaikeaa havaita visuaalisesti. Poikkeavuuksien havainnointi ja trendien löytäminen aikasarjatiedoista auttaa esim. oppimisalgoritmien kehittämisessä. *Azure Maps* sisältää joukon ominaisuuksia geospaatialisen tiedon käsittelemiseen. IoT-sovelluksissa hyödyllisiä ominaisuuksia ovat mm. liikenne, reititys, vallitseva säätila ja haku. Azure Mapsin avulla voidaan suunnitella esimerkiksi reitti paikasta A paikkaan B ottaen huomioon erilaiset kulkuvaihtoehdot: ajoneuvotyyppi, vaarallinen lasti, ajoneuvon mitat, välipysäkit, liikenne ja niin edelleen.

Azure Digital Twins on IoT-palvelu, jolla luodaan kattavia ohjelmistomalleja fyysiseen ympäristöön. Palvelulla mallinnetusta avaruudesta voidaan suorittaa kyselyitä sen sijaan että kysely kohdistuisi useisiin eri antureihin. *IoT Hub SDK:t* tukevat useita yleisiä ohjelmakieliä, mm. .NET, C, Java, Node.js, Python, ja iOS. *Azure IoT Device SDK:t* sisältävät koodia, joka helpottaa Azure IoT Hub -palveluihin yhdistävien ja niiden hallinnoimien sovellusten luomista. Se tarjoaa kustomointivaihtoehtoja eri laitetyypeille. *Azure IoT Service SDK:t* sisältävät koodia, joka helpottaa sellaisten sovellusten rakentamista, jotka ovat suoraan vuorovaikutuksessa IoT Hubin kanssa laitteiden ja suojauksen hallintaa varten. SDK:n avulla voidaan lähettää viestejä, ajoittaa töitä, kutsua suoria menetelmiä ja lähettää haluttuja ominaisuuspäivityksiä IoT-laitteille ja -moduuleille. *Provisioning Device SDK:t* mahdollistavat IoT-laitteilla toimivien sovellusten rakentamisen ja yhteydenpidon Device Provisioning Servicen kanssa. SDK:t sisältävät useita kirjastoja, jotka tarjoavat ratkaisuja moniin IoT-toteutusten keskeisiin ongelmiin ja tarpeisiin. *Provisioning Service SDK:t* mahdollistavat taustasovellusten rakentamisen, joilla voi hallita Device Provisioning Serviceen rekisteröityjä laitteita.



Kuva 5.2: Microsoft Azure IoT palvelukomponentteja [31]

5.2 IoT datan rakenteen mukainen luokittelu

Mittalaitteiden tuottama IoT-data on tyypillisesti kerätty monenlaisista päätelaitteista, ja sitä voi olla hyvinkin suuria määriä. Data voi olla muodoltaan rakenteista, osittain rakenteista tai ei-rakenteista. Pilvipalvelut tarjoavat IoT-datalle erilaisia niille hyvin soveltuvia tallennusvaihtoehtoja.

Rakenteinen data on ennalta määritellyn muotoista, ja tiedon rakenne pysyy koko ajan samana. Sitä voidaan tallettaa tietokannan tauluihin ja sarakkeisiin. Datalla on taulussa yksilöivä avaintieto, ja taulut voivat muodostaa relaatioita toisiin tauluihin avaintiedon perusteella. Dataa kutsutaan yleisesti relationaaliseksi. SQL (Structured Query Language) on yleinen tiedon käsittelykieli. Excel-taulukko data on esimerkki rakenteisesta datasta.

Osittain rakenteinen data ei ole niin määrämuotoista kuin rakenteinen data, mutta se on jossain määrin hierarkkista. Data tunnistetaan avainten ja tagien perusteella. Osittain rakenteista dataa kutsutaan yleisesti ei-relationaaliseksi dataksi tai NoSQL-dataksi. JSON-muotoinen data on esimerkki osittain rakenteisesta datasta.

Ei-rakenteinen data ei noudata mitään formaattia. Pohjimmiltaan kaikki data mikä ei ole rakenteista tai osittain rakenteista kuuluu tähän luokkaan. Ei-rakenteista dataa voi olla esimerkiksi video- tai audiotiedosto, tai vaikkapa PDF-dokumentti.

Azuressa on datalle myös tasot *hot*, *cold* ja *archive* [30]. Hot dataa tarvitaan jatkuvasti, ja järjestelmän välitön toiminta perustuu datan analysoinnin tulokseen. Esimerkiksi auton automaattiohjaukseen liittyvä data on tällaista. Cold data puolestaan on enemmän historiallista dataa, jota ei haeta kovinkaan säännöllisesti, ja jota talletetaan vähintään 30 päiväksi. Tällaista dataa on esimerkiksi lämpötilamittausten tilastot. Archive dataa haetaan harvoin ja sitä talletetaan vähintään 180 päiväksi. Haulla ei ole latenssivaatimuksia, toisin kuin hot datalla.

5.3 Azuren tietovarastot

Azure IoT Hubissa käytettyjä datan tallennusvarastoja ovat:

- Azure SQL database

- Azure Cosmos DB
- Azure Blob Storage
- Azure Data Lake Storage
- Azure Files
- Azure Disc Storage
- Azure Queue

Käytettävä varastointivaihtoehto riippuu muun muassa tiedon rakenteesta ja tallennettavan datan määrästä. Microsoft hajauttaa tietovarastonsa ympäri maailman maantieteellisiin aluekeskuksiinsa, ja käyttäjän datan jako aluekeskukseen lähelle käyttäjää nopeuttaa huomattavasti sovelluskehitystä missä päin maailmaa tahansa. Hajautuksella varmistetaan myös tiedon varastoinnin luotettavuutta.

Azure SQL database on täysin manageroitu Platform as a Service (PaaS) tietokantakone, joka hallitsee suurimman osan tietokannan käsittelyfunktioista, kuten päivitykset, varmuuskopioinnin ja monitoroinnin ilman että käyttäjän tarvitsee huolehtia niistä. Azure SQL Database on aina viimeisin vakaa käytössä oleva versio, jonka saatavuudeksi luvataan 99,99%. Azure SQL Database PaaS-kyvykkyydet vapauttavat käyttäjien resurssit keskittymään liiketoiminnan kannalta kriittisiin toimialuekohtaisiin tietokantojen hallinto- ja optimointitoimiin [31].

Azure Cosmos DB on täysin manageroitu NoSQL tietokanta moderniin sovelluskehitykseen. Millisekuntien vastausaika, ja automaattinen ja välitön skaalautuvuus takaavat riittävän nopeuden joka mittaluokassa. Käyttäjien liiketoiminnan häiriöttömyys on taattu SLA:han pohjautuvalla saatavuudella ja yritystason tietoturvaratkaisuilla [31]. Kuten Azure SQL Databasessa, Azure Cosmos DB:n versio on aina viimeisin, ja täysin hallinnoituna se huolehtii päivityksistä, varmuuskopioinneista ja monitoroinnista.

Azure Blob Storage on Microsoftin pilvessä tarjoama objektien varastointiratkaisu. Blob Storage on optimoitu tallentamaan massiivisia määriä ei-rakenteista dataa.

Azure Data Lake Storage (ADLS) on täysin manageroitu elastinen, skaalautuva ja

turvallinen tietojen tallennus- ja analyysipalvelu. Se tukee Hadoop Distributed File Systemiä (HDFS) ja Cosmos semantisia. ADLS on erityisesti suunniteltu ja optimoitu laaja -alaiselle Big Data -analytiikalle, joka on riippuvainen erittäin suurista rinnakkaisista luku ja -kirjoitusoperaatioista ja datan rinnakkaisesta käsittelystä suurilla kaistanleveyksillä ja matalalla latenssilla. ADLS yhdistää Microsoftin ja HDFS:n sisäisten asiakkaiden käyttämän Microsoft Cosmos-tiedostojärjestelmän keskeiset komponentit ja ominaisuudet, ja se on yhtenäinen tiedostojen tallennusratkaisu Azuren analytiikkaan. ADLS:n piirteitä ovat sen suunnittelu useiden tallennuskerrosten käsittelyyn, exabyte-asteikko ja kattavat suojaus- ja tiedonjako -ominaisuudet [36].

Azure Files tarjoaa täysin manageroidun levyjakopalvelun pilvessä. Levyt ovat saatavilla Server Message Block (SMB) protokollan tai Network File System (NFS) protokollan kautta. Azure Files levyjaot voidaan asentaa samanaikaisesti pilveen ja on-premises ympäristöön. SMB Azure Files levyjaot ovat saatavilla Windows, Linux, ja macOS clienteleillä. Lisäksi Azure Files levyjakoihin päästään Windows Servereiltä Azure File Syncin avulla käyttäjää lähinnä olevasta palvelukeskuksesta.

Azure Disc Storage (managed disks) ovat Azuren virtuaalikoneiden käyttämiä block-tason Azuren hallinnoimia tallennusvolyymeja. Käsite Block tarkoittaa suuria tietoryhmiä jotka ovat tallennettu yhtenä kokonaisuutena johonkin Microsoftin maantieteellisistä aluepalvelukeskuksista.

Azure Queue Storage on palvelu suurten viestimäärien tallentamiseen. Viestejä voidaan vastaanottaa mistä vain autentikoiduilla HTTP ja HTTPS -kutsuilla. Yhden Azure Queuen viestin koko voi olla suurimmillaan 64 kb. Queueissa voi olla miljoonia viestejä Azure Queue Storagen kokonaiskapasiteettitilaan saakka. Queueja käytetään yleisesti luomaan asynkronisesti backlogeja työnkuluista [31].

5.4 Azure sovelluskehitys

Azure IoT koostuu erilaisista hallituista palveluista, joiden avulla yhdistetään, valvotaan ja ohjataan IoT-laitteita. IoT sovelluskehitykseen ja sovellusten hallintaan on useita vaihtoehtoja. Azure Portal tarjoaa visuaalisen web-pohjaisen käyttöliittymän, ja Azure CLI (Command-Line Interface) ja Azure Cloud Shell komentorivipohjaisen

käyttöliittymän Azure resursseihin. Azure Portal antanee varsinkin tottumattomalle käyttäjälle helpomman, visuaalisen lähestymistavan, kun taas komentorivipohjaisilla liittymillä ja niillä tallennetuilla komentoskripteillä voidaan tehokkaammin hallita isompia kokonaisuuksia, esimerkiksi suorittaa kerralla useiden samankaltaisten IoT-laitteiden konfigurointeja.

Internetistä löytyy lukemattomia ohjesivustoja IoT-sovelluskehitykseen, sekä maksullisia että maksuttomia. Tämä pätee myös Azure IoT-kehitykseen. Microsoftin oma dokumentaatio on luonteva paikka aloittaa aiheeseen tutustuminen. Työn empiirisessä osassa (6) kuvataan IoT-ratkaisun toteuttaminen Azuren IoT ympäristössä.

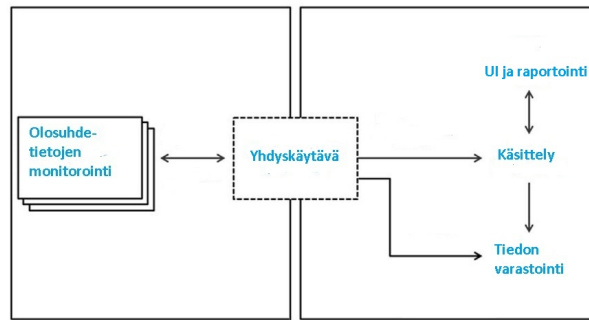
6 IoT sovelluksen toteutus pilvialustalle

Loma-asunnon omistajan olisi hyödyllistä ja mielenkiintoista kyetä seuraamaan loma-asuntonsa olosuhde- ja säätilatietoja etänä. Yksi tai useampi mittalaite sijoitettuna mökissä eri paikkoihin, ja internetistä vapaasti saatavilla oleva säätilatieto voisivat yhdessä kertoa esimerkiksi lämmityksen tehokkuudesta tai vaikkapa ulkoseinän lämmöneristävyydestä eri sääolosuhteissa. Tässä työssä toteutettiin edellä kuvattuun tarkoitukseen Microsoft Azuren IoT arkkitehtuuria mukailleen (kuva 6.1) IoT-ratkaisu. Mökin ilmanlaatu ja säätilasovellukseen tarvittiin ilmanlaatu- ja säätietoja tuottavat laitteet, monitorointitiedon välittämisen mahdollistava langaton tiedon siirtoratkaisu, joukko pilvipalveluita ja raportointityökalu. Lisäksi sovelluksessa haettiin hyödyntää Ilmatieteen laitoksen avointa säähavaintodataa.

6.1 Mökin olosuhdetietojen mittaussovellus

Tässä työssä toteutettu palvelu kerää mittaustietoa mittalaitteesta, siirtää mitatun datan pilvialustalle ja tallettaa, prosessoi ja esittää mitatun havaintodatan raportin muodossa. Havainnot rikastetaan sää tiedoilla, jotka saadaan haettua Ilmatieteen laitoksen avoimesta rajapinnasta. Mittalaite lähettää LoRa-verkon välityksellä Azuren pilvipalveluihin dataa, sitä muokataan ja se tallennetaan tietokantaan Azure-pilvessä. Havainnot rikastetaan hakemalla Ilmatieteen laitoksen avoimesta rajapinnasta säätietoja ulkoilmasta, ja lopputulokset esitetään Power BI raportin muodossa loppukäyttäjille.

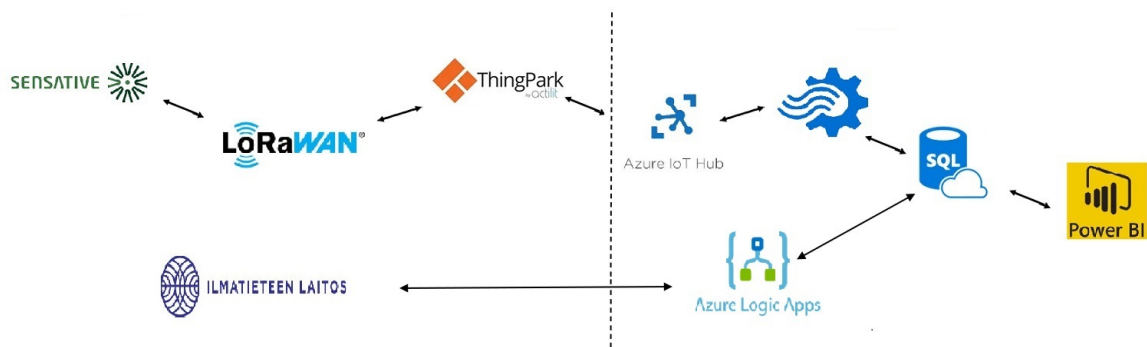
Työssä käytettiin kuvassa 6.2 esitettyjä komponentteja. Mittalaitteena oli Sensative MS +Comfort [39] sisäilman havainnointiin, ja LoRa-verkkopalveluna ThingPark Actility [45], jonne mittalaite lähettää dataa. Pilvipalveluista otettiin käyttöön Azure IoT Hub, pilvipalveluiden solmu, joka vastaanottaa datan LoRa-verkosta, Azure Stream Analyticsin analytiikkapalvelut datan käsittelyyn, Azure Logic Apps työnkulut Ilmatieteen laitoksen havaintojen käsittelyyn, Azure SQL Database datan varastointiin ja Power BI raportointityökalu datan esittämiseen.



Kuva 6.1: Työssä seurattu IoT arkkitehtuuri mukaillee Microsoftin Azure IoT arkkitehtuuria [31]

Mittalaite konfiguroitiin mittaamaan ympäristöstään lämpötila, ilmakosteus ja valoisuus kerran tunnissa. Mitattu data lähetettiin LoRa-verkon välityksellä Azuren IoT Hubiin, ja sieltä Azure Stream Analyticin kautta Azure SQL tietokantaan. Azure Logic Apps palvelu haki kerran kahdessa tunnissa Ilmatieteen laitoksen säähavainnot ja tallensi ne Azure SQL tietokantaan. Tietokantaan luotiin näkymä, jossa yhdistettiin mittalaitteen ja Ilmatieteen laitoksen hakema data yhdeksi kokonaisuudeksi. Power BI raportointiväline haki tietokantaan tallennetut tiedot tästä näkymästä ja muodosti niistä raportin.

Verrattaessa ratkaisua luvussa 3 esitettyyn Agarwal et al. referenssimalliin (3.1), komponenttien jako voidaan tehdä hyvin suoraviivaisesti. Mittalaite asettuu *sensori- ja yhteyskerrokselle*, tiedonsiirtoratkaisu *yhdyskäytävä- ja verkkokerrokselle*, pilvipalvelut *hallinta- ja palvelinkerrokselle* ja raportointityökalu sijoitetaan *sovelluskerrokseen*.



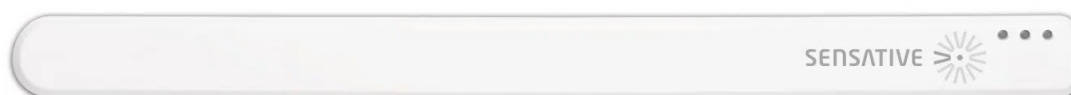
Kuva 6.2: Mökin olosuhdetietojen mittaussovellus. Havaintodataa saadaan sisätilan mittalaitteesta ja Ilmatieteen laitoksen avoimesta rajapinnasta.

6.1.1 Olosuhdemittaus

Työn mittalaitteena oli *Sensitive MS +Comfort* [39] (kuva 6.3). Sensitive (<http://sensitive.com>) on ruotsalainen IoT ratkaisuja tarjoava yritys, jolla on laaja valikoima erilaisia IoT-ratkaisuja, mukaan lukien mittalaitteita. MS +Comfort mittaa ympäristön lämpötilaa, ilmankosteutta ja valoisuutta, ja käyttää LoRa-viestejä datan lähettämiseen. Laitteessa on myös magneetilla toimiva anturi, joka voidaan asentaa esimerkiksi oveen ilmaisemaan onko ovi auki vai kiinni. Mittalaitteeseen voidaan asettaa haluttu konfiguraatio HTTP-kutsulla. Sensitive tarjoaa useita valmiita konfiguraatioprofiileja web-sivuillaan, ja niitä voidaan myös luoda itse [40]. Profiilissa määritellään, mitä mittatietoa kerätään ja millä aikasyklillä sitä halutaan lähettää. Profiili on heksademaalimuotoinen merkkijono, joka lähetetään HTTP-kutsulla mittalaitteeseen. Vastaavasti mittalaitteen lähettämä mittadata on myös heksadesimaalimuotoista.

6.1.2 Ilmatieteen laitoksen avoin data

Ilmatieteen laitos tarjoaa avoimen OGC Web Feature Service 2.0 (WFS) rajapinnan (<https://www.ilmatieteenlaitos.fi/avoin-data>) kautta koneluetta- via säähavaintoja, joita käytettiin tässä työssä. Rajapinnasta haetaan kerran kahdes- sa tunnissa mittauspisteen ulkolämpötila, tuulen voimakkuus ja tuulen suuntatiedot. Tiedot haettiin Azure Logic Apps työnjolla [32] ja talletettiin Azure SQL tietoi-



Kuva 6.3: Sensative Multi-sensor +Comfort [39]

kantaan.








6.1.3 LoRa verkkopalvelu

Yliopistokeskuksen informaatioteknologian yksikkö hallinnoi keskitetysti ThingPark Actility LoRa-verkkopalvelua ja siihen liitettyjä mittalaitteita. Yksikkö tarjosi mittalaitteen, ja heidän tuellaan mittalaite liitettiin ThingPark Actilityn verkkopalveluun. Mittalaite konfiguroitiin välittämään dataa Azure IoT Hubiin (5.1), jolla yhdistetään, seurataan ja hallinnoidaan IoT-resursseja.

6.1.4 Microsoft Azure

Työn toteuttamisessa käytettiin Microsoftin opiskelijoille tarjoamaa ilmaista Azure ympäristöä [29]. Microsoft Azure pilvipalveluista valittiin seuraavat palvelut (kuva 6.4): Azure Account, Azure Subscription ja Resource Group jotka ovat hallinnointipalveluita palveluiden organisointiin ja muun muassa laskutukseen. Datan varastointia ja analytiikkaa varten luodut palvelut olivat Azure IoT Hub, Azure Logic Apps, Azure Stream Analytics, Azure SQL Database ja Azure SQL Server.

Palveluita on kuvattu tarkemmin edellisessä luvussa (5.1). Hieman yksinkertaistettuna kokonaisuuteen tarvitaan Azure Account, käyttäjätili jonka alle luodaan Azure Subscription. Sen alle luodaan Resource Group, ja tämän alle käytetyt resurssit, Azure IoT Hub ym. Isommissa organisaatioissa ja ratkaisuissa Accounteja, Subscriptioneita ja Resource Groupeja voi luonnollisesti olla useampia, ja niiden järkevällä hallinnoinnilla on suuri merkitys onnistuneen pilviratkaisun toteuttamiseen.

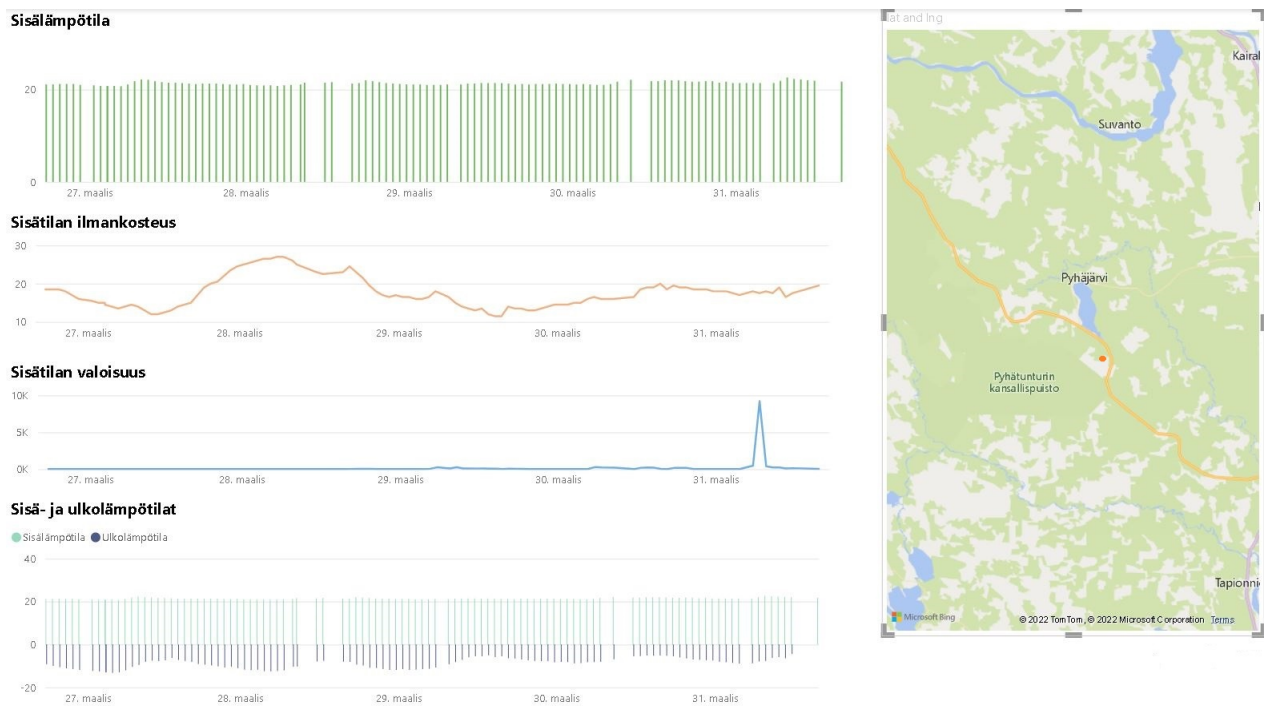
 iot-lorawan-rg	Resource group
 iot-lorawan-db	SQL database
 iot-lorawan-lap	Logic app
 iot-lorawan-sa	Stream Analytics job
 iot-lorawan-hb	IoT Hub
 iot-lorawan-srv	SQL server
 Azure for Students	Subscription

Kuva 6.4: Azure-palvelut Mökin olosuhdetietojen mittaussovellukseen.

Tables	Views
<ul style="list-style-type: none"> dbo.IoTdata <ul style="list-style-type: none"> ID (PK, bigint, not null) SensorID (varchar, not null) data (varchar, null) date (datetime2, null) temperature (decimal, null) humidity (decimal, null) lux (decimal, null) rsi (float, null) snr (float, null) lat (float, null) lng (float, null) 	<ul style="list-style-type: none"> dbo.FMIdata <ul style="list-style-type: none"> ID (PK, bigint, not null) LocationID (nvarchar, not null) timeID (nvarchar, null) date (datetime2, null) temperature (decimal, null) windspeed (decimal, null) winddirection (decimal, null) dbo.WeatherReport <ul style="list-style-type: none"> ID (bigint, not null) SensorID (varchar, not null) data (varchar, null) date (datetime2, null) insideTemp (decimal, null) humidity (decimal, null) lux (decimal, null) rsi (float, null) snr (float, null) lat (float, null) lng (float, null) outsideTemp (decimal, null) windspeed (decimal, null) winddirection (decimal, null) LocationID (nvarchar, null)

Kuva 6.5: Mökin olosuhdetietojen mittaussovelluksen tietokannan rakenne. Tietokanta oli hyvin yksinkertainen. Se koostui kahdesta taulusta ja yhdestä näkymästä.

IoTdata-tauluun talletettiin mittalaitteen lähettämästä datasta tapahtuman sisäinen ID, laitteen ID, aikaleima, data sellaisena kuin se tuli, lämpötila-, ilmankosteus- ja valaistusarvot, laitteen lähetystehoon liittyviä tietoja sekä mittalaitteen paikkatieto karttakoordinaatiston pituus- ja leveysarvoina. FMIdata-tauluun tallennettiin Ilmatieteen laitoksen dataa jalostaen sisäinen ID, LocationID havaintopaikkaa varten,



Kuva 6.6: Power BI raporttisivu. Aikajanalla sisä- ja ulkolämpötilat, valoisuus ja ilmankosteus pylväs- ja janadiagrammeina esitettyinä. Lisäksi mittauspiste karttanäkymässä

TimeID Logic Appin sisäiseen käyttöön, päivämäärä- lämpötila, tuulen voimakkuus ja suuntatiedot. Taulujen merkitsevät kentät koottiin WeatherReport-näkymään, josta raportointityökalu Power BI haki tiedot kootusti raportille.

6.1.5 Power BI raportointityökalu

Raportointivälineeksi valikoitui Microsoft Power BI raportointityökalu [33]. Raportointityökalusta voidaan asentaa työasemalle ilmainen Power BI Desktop versio. Kuvissa 6.6 ja 6.7 on esitetty raportin tuottamat sisä- ja ulkolämpötilat, kosteus- ja valoisuusarvot sekä tuulen voimakkuus ja suuntatiedot ajan suhteen.

6.2 Kokemuksia

Työn lopputuloksena saatiin Proof of concept -tyyppinen ratkaisu, joka kerää ympäristönsä olosuhteista dataa, lähettää sen LoRa-verkon välityksellä Microsoft Azure



Pvm	tuulen suunta	voimakkuus m/s
30.3.2022 14:36:27	322,00	11,20
30.3.2022 15:36:29	330,00	13,60
30.3.2022 16:36:30	335,00	12,30
30.3.2022 17:36:26	332,00	12,20
30.3.2022 18:36:27	333,00	13,20
30.3.2022 19:36:26	335,00	11,90
30.3.2022 20:36:27	335,00	10,70
30.3.2022 21:36:28	341,00	10,10
30.3.2022 22:36:25	344,00	9,70
30.3.2022 23:36:28	343,00	8,50
31.3.2022 0:36:27	331,00	7,90
31.3.2022 1:36:26	332,00	8,10
31.3.2022 2:36:21	330,00	8,60
31.3.2022 3:36:28	321,00	7,90
31.3.2022 5:36:29	328,00	7,40
31.3.2022 6:36:28	333,00	7,00
31.3.2022 7:36:23	340,00	7,00
31.3.2022 8:36:26	337,00	6,40
31.3.2022 9:36:22	327,00	5,20
31.3.2022 10:36:24	349,00	7,40
31.3.2022 11:36:20	342,00	8,10

Kuva 6.7: Power BI raporttisivu. Raportilla tuulen voimakkuus pylväsdiagrammina ja tuulen voimakkuus ja suunta taulukkona.

pilvipalveluihin, tallettaa datan ja esittää sen Power BI -raportilla. Mittalaitteen tuottaman datan lisäksi haetaan säätietoja Ilmatieteen laitoksen avoimesta rajapinnasta. Työn toteutus oli kohtalaisen vaivatonta. Azure oli tekijälle entuudestaan tuttu, joskin IoT Hubiin ja Stream Analyticsiin tutustuttiin ensi kertaa. Internetistä löytyy runsaasti ohjeita toteutuksesta, eikä vähiten Microsoftin omilta sivuilta.

Sensativellä on internetissä vapaasti hyödynnettävä git-projekti, jota voidaan hyödyntää sovelluskehityksessä. Sensativen konfigurointisivustolla on myös dekoodeeri, jolla voidaan purkaa auki mittadatan merkkijono. Missään ei kuitenkaan varsinaisesti avata sitä, miten merkkijono muodostuu. Tämä tuotti työhön hieman haastetta. Mittausdata saatiin onnistuneesti tallennettua tietokantaan koko Azure palveluputken läpi, mutta aluksi käsillä oli vain heksamuotoinen merkkijono vailla mitään merkitystä. Ratkaisuna oli käyttää Sensativen tarjoamaa dekooderia. Dekoodeerille syötettiin erilaisia merkkijonoja, ja yrityksen ja erehdyksen kautta saatiin selville mistä kohtaa merkkijonoa eri mittausarvot, lämpötila, ilmankosteus ja valoisuus, löytyivät.

Pilvipalveluiden kehitystyö tehtiin Azure Portalissa (5.4). Työtä varten luotiin Azure for Students, Microsoftin opiskelijoille tarjoama käyttäjätili. Tilille on talletettu valmiiksi 100\$ krediittiä, joilla voidaan maksaa käytettyjä palveluita. Tili antaa opiskelijoiden käyttöön myös suuren joukon kokonaan ilmaisia palveluita. Laskutettavat palvelut ovat ilmaisia tilin alkusaldon 100 dollariin asti, jonka jälkeen tili toimii pay-as-you-go periaatteella. Tällöin käytetyistä resursseista ruvetaan laskuttamaan.

Kun käyttäjätili oli luotu, oli seuraava vaihe Subscriptionin luonti, joka kuten muutkin palvelut, luotiin Microsoftin yksityiskohtaisten ohjesivujen avulla. Tämän jälkeen luotiin Resource Group vastaavalla tavalla ohjeita seuraten. Resource Groupin alle luotiin työn käyttämät varsinaiset resurssit. Ensinnä tarvittiin Azure IoT Hub, joka ottaa vastaan mittalaitteen lähettämää mittadataa. IoT Hubin luonnissa syntyy Connection String, joka toimii IoT Hubin yksilöivänä osoiteavaimena. Connection String toimitettiin Yliopistokeskuksen informaatioteknologian yksikölle, ja mittalaite ohjattiin lähettämään dataa luotuun IoT Hubiin. IoT Hubin lähettämän datan jatkokäsittely tapahtuu Azure Stream Analytics palvelussa. Luodulle palvelulle määritettiin *Input*iksi IoT Hub, joka siis syötti dataa, ja *Output*iksi Azure SQL DB, tietokanta jonne data tallennettiin. Datan muokkaamiseksi Stream Analyticsiin teh-

tiin *Query*. Query on kysely, joka haki datan IoT Hubilta, muokkasi datan ja tallensi sen SQL tietokantaan. Heksadesimaaliarvojen parserointiin tehtiin Stream Analytics funktio, jota käytettiin kyselyssä.

Ilmatieteen laitoksen avoin data haettiin ja käsiteltiin Logic Apps työjonolla. Työjonon työkuluissa data parseroitiin oikeaan muotoon ja tallennettiin tietokantaan. Datan tallennusta varten luotiin Azure SQL DB relaatiotietokanta. Tietokanta oli yksinkertainen, sisältäen vain kaksi tietokantataulua ja yhden näkymän. Toiseen tauluun talletettiin mittalaitteen lähettämä data ja toiseen tauluun ilmatieteen laitoksen data. Tiedot yhdistettiin näkymään, josta ne tarjottiin Power BI:n käytettäväksi. (kuva 6.5).

Microsoft Azure pilvimaailma oli entuudestaan tuttua, joten navigointi Azure Portalissa, missä komponentit toteutettiin, onnistui ongelmitta. Azure Subscriptionin ja Resource Groupin luonti on varsin suoraviivaista ja käy nopeasti. Azure IoT Hub ja Azure Stream Analytics olivat entuudestaan tuntemattomia palveluita, mutta niidenkin toteutus sujui vaikeuksista. Samoin Azure SQL database, relaatiotietokanta luotiin sujuvasti. Tietokannan rakennetta ei kerättävän datan yksinkertaisuuden takia joutunut juurikaan miettimään.

Ilmatieteen laitoksen latauspalvelusta voi ladata dataa koneluettavassa muodossa. latauspalvelu on toteutettu OGC Web Feature Service 2.0 (WFS) rajapinnan avulla. Data tuotetaan XML-muotoisena. Datan hakua ja käsittelyä varten tehtiin Azure Logic Apps, jonka työkuluilla data noudettiin rajapinnasta ja tallennettiin käsittelyn jälkeen tietokantaan. Tämä työvaihe vaati jonkun verran ponnisteluja. Dataa jouduttiin melko lailla parseroimaan oikeaan muotoon ja miettimään parhaat tavat, kuinka haluttu tieto saadaan kaivettua esiin rajapinnan tarjoamasta datasta. Logic Appsista löytyi lopulta oikeat komponentit datan muokkaamiseksi, ja tiedot säätilasta saatiin tuotettua sovellukselle. Aiemmasta Logic Apps kokemuksesta oli varsin paljon hyötyä tässä työvaiheessa.

Power BI ei ollut entuudestaan tuttu, ja tämän johdosta työssä päädyttiin toteuttamaan hyvin yksinkertaiset raporttinäkymät (kuvat 6.6, 6.7). Power BI:ssä on erittäin monipuoliset mahdollisuudet datan visualisointiin, joten raportin ulkoasua voidaan kehittää jatkossa näyttävämmäksi.

7 Johtopäätökset ja pohdintaa

Työn tavoitteena oli luoda katsaus pilvipalveluihin IoT-datan hallinnan näkökulmasta. Työssä käytiin läpi teoreettisia referenssimalleja, esitettiin IoT-alustoille asetettuja vaatimuksia ja käytiin läpi valintakriteereitä, joita voidaan hyödyntää valittaessa palveluratkaisua. Työhön valittiin yleisiä ja suosittuja pilvialustoja, joita käytiin yleisellä tasolla läpi. Läpikäytäviin alustoihin haluttiin poimia sekä kaupallisia että avoimen lähdekoodin ratkaisuja.

Työssä esiteltyt LPWAN-teknologiat muodostavat tiedonsiirtoratkaisun mittalaitteilta yhdyskäytävälle. Silmiinpistävin ero teknologioissa oli käytetyissä taajuuksissa. NB-IoT toimii niistä ainoana LTE-taajuuksilla, ja on tästä syystä kallein ratkaisu. LoRaWAN ja Sigfox toimivat kumpainkin edullisemmilla ISM-taajuuksilla.

Referenssimallit, joita on kehitetty selkeyttämään arkkitehtuurin kuvausta, ovat pitkälti samankaltaisia. Tämä on toki luonnollista, sillä IoT-ratkaisuissa käytettävät komponentit ovat suurimmaksi osaksi samoja. Tarvitaan mittalaitteita havainnoimaan ympäristöä, jokin tiedonsiirtoratkaisu, välineet tiedon käsittelyyn ja tapa jolla analysoitu tieto esitetään loppukäyttäjille. Referenssimallit helpottavat hallitsemaan kokonaisuutta jakamalla komponentit pienempiin kokonaisuuksiin.

IoT pilvialustojen vertailussa ei haettu niinkään eroja alustojen välillä, mutta haluttiin luoda katsaus muutamaa tunnettuun ja suosittuun pilvialustaan. Mukaan valittiin sekä kaupallisia että avoimen lähdekoodin alustoja. Ei ollut yllätys, että suurten kaupallisten palveluntarjoajien pilvialustat tarjosivat monipuolisempia ja laajempia kokonaisuuksia. Tämä näkyi myös kaupallisten ratkaisujen korkeampina kustannuksina.

Työn empiirisessä osuudessa haluttiin testata käytännössä, mitä IoT-ratkaisun toteutus vaatii. Osuudessa toteutettiin Mökin olosuhdetietojen mittaussovellus, jossa kaupallinen mittalaite mittaa ympäristönsä olosuhdetietoja, tässä tapauksessa mökin sisälämpötilaa, valaistusta ja ilmankosteutta. Mittalaite lähettää havainnot

LoRa-viestinä pilvipalveluille, jossa dataa ja jalostetaan ja se tallennetaan tietovarastoon. Mittalaitteen mittaaman havaintodatan lisäksi Ilmatieteen laitokselta haetaan tietokantaan avoimen rajapinnan kautta havaintodataa mittauspaikan ulkopuolisesta säätilasta. Nämäkin tiedot tallennetaan tietokantaan, ja lopuksi kaikki tietokantaan tallennettu data noudetaan esitettäväksi raportointivälineellä.

Koska toteutettu IoT-ratkaisu oli vaatimattoman kokoinen sisältäen vain yhden mittalaitteen, ei kaikkia luvussa (3.2) esitettyjä vaatimuskriteerejä saatu todennettua. Muutamia vaatimuksia voidaan kuitenkin työstä saadun kokemuksen perusteella tarkastella.

Toiminnallisista vaatimuksista poimittuina voidaan yhteisesti todeta, että vaatimuksiin *tiedonhallinnasta, tiedon keruusta, käsittelystä ja tallennuksesta* Azure vastaa vähintäänkin hyvällä tasolla. *Tiedon visualisointiin* löytyy kattavat työkalut, esimerkkinä näistä tässä työssä käytetty Power BI runsaine visualisointiominaisuuksineen. Azure Stream Analyticsin kyselyt toteutetaan Stream Analytics Query Languagella, joka perustuu T-SQL standardiin. Standardin tunteminen entuudestaan auttaa toki kyselyjen teossa, mutta syntaksi on myös kokemattoman käyttäjän helppo omaksua. Vaatimukseen *kyselyjen teosta* siis vastataan.

Skaalautuvuutta ei yhden mittalaitteen sovelluksessa kyetä todentamaan, mutta Azuren komentorivipohjaiset käyttöliittymät mahdollistavat monikäyttöisten skriptien toteutuksen. Skripteillä voidaan esimerkiksi lisätä mittalaitteita toteutuksiin massa-ajoilla. Sovellus ei ole niin kriittinen tiedon tosiaikaisuudesta, joten *oikea-aikaisen toteutuksen* merkitys ei ollut kovin suuri. Vaatimuksen *saatavuus* merkitys ei myöskään ollut merkittävä, ei haitannut vaikka dataa ei olisi saatu jatkuvasti. Azuressa on erittäin kehittyneet tietoturvaratkaisut, joten vaatimus *tiedon suojaamisesta ja yksityisyyden turvaamisesta* voidaan katsoa toteutuneen.

Azure on puhtaasti palvelupohjainen ratkaisu. Palveluiden tarjonta on varsin runsasta, ja ratkaisut voidaan toteuttaa valitsemalla tarvittavat palvelut useista eri vaihtoehtoista. Esimerkiksi tietokantaratkaisuja on useita, ja työssä toteutettu Logic Apps -palvelu olisi hyvin voitu toteuttaa myös esimerkiksi käyttämällä Azure Function -palvelua. Vaatimus *palvelupohjaisuudesta* toteutuu Azuressa.

Azuressa palveluiden sijainti voidaan valita maakohtaisesti ja alueellisesti. Valitusta palvelutasosta riippuen palvelut voidaan myös *hajauttaa* maantieteellisesti ympäri maailman Microsoftin aluekeskuksiin. Tämä tapa turvaa palveluiden saatavuuden siinäkin tapauksessa, että jokin aluekeskuksista olisi syystä tai toisesta alhaalla. Kustannukset laajempaa hajautusta käyttäen tosin luonnollisesti kasvavat. Kaikki resurssit on syytä luoda samalle maantieteelliselle alueelle (tässä työssä North Europe) lisäkustannusten välttämiseksi.

Kaiken kaikkiaan voidaan todeta, että IoT toteutuksen kehittäminen pilvipalveluilla on melko helppoa ja onnistuu ilman sen suurempaa aikaisempaa kokemusta pilvialustoista. Internetistä löytyvät ohjeet ja esimerkit antavat hyvät valmiudet omien ratkaisujen toteuttamiseen. Pilvipalveluiden tarjonta on runsasta, ja käyttäjien on nykyisin helppo toteuttaa IoT sensoreihin perustuva sovellus pilvipalveluilla. Alustoissa löytyy runsaasti valinnanvaraa halusipa sitten käyttää avoimen lähdekoodin tai kaupallista ratkaisua. Käytön helppous ja ohjeistuksen ja esimerkkien runsaus madaltavat kynnyksiä tarttua toimeen.

Palveluiden valinnassa kannattaa olla tarkkana ja suunnitella tarkoin tulevan ratkaisun arkkitehtuuri ja resurssitarpeet. Palvelutasojen valinnalla voidaan saavuttaa huomattavia kustannussäästöjä, sillä useiden palveluntarjoajien hinnoittelumalli perustuu käytettyihin resursseihin. Esimerkkinä mainittakoon työn empiirisen osan tietokanta. Tietokannan palvelutaso ja sille varatut resurssit määriteltiin aluksi turhan suuriksi, jolloin käytössä ollut Microsoftin opiskelijoille tarjoama 100 dollarin krediitti rupesi hupenemaan huomattavaa tahtia. Kannan uudelleen luonti vastaamaan pienempää resurssitarvetta alensi heti kustannuksia.

IoT ratkaisujen tulevaisuusnäkymät ovat valoisat. IoT maailma ja pilvipalvelut tarjoavat lupaavat kaupalliset näkymät laitevalmistajille, internetpalveluiden tarjoajille ja sovellusten toteuttajille. Tässä työssä toteutettu Mökin olosuhdetietojen mittaussovellus olkoon hyvänä esimerkkinä käytännön ratkaisusta, jollaisia mittalaitteilla ja pilvipalveluilla voidaan toteuttaa. Sovelluksen jatkokehitystä voisi toteuttaa esimerkiksi lisäämällä mökkiin useampi mittalaite ja parantamalla raportin visuaalista ilmettä. Kehitystyötä on hyvä lähteä jatkamaan toteutetun ratkaisun pohjalta.

Lähteet

- [1] AGARWAL, P., JA ALAM, M. Investigating iot middleware platforms for smart application development. Kirjassa *Smart Cities - Opportunities and Challenges*. Springer, 2020, ss. 231–244.
- [2] AGARWAL, P., JA ALAM, M. Open service platforms for iot. Kirjassa *Internet of Things (IoT)*. Springer, 2020, ss. 43–59.
- [3] AL-FUQAHA, A., GUIZANI, M., MOHAMMADI, M., ALEDHARI, M., JA AYYASH, M. Internet of things: A survey on enabling technologies, protocols and applications. *IEEE communications surveys & tutorials* 17, 4 (2015), 2347–2376.
- [4] ALAM, M., JA SHAKIL, K. A. Big data analytics in cloud environment using hadoop. *CoRR abs/1610.04572* (2016).
- [5] ALI, S. A., AFFAN, M., JA ALAM, M. A study of efficient energy management techniques for cloud computing environment. Julkaisusarjassa *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (2019), IEEE, 13–18.
- [6] AMAZON. Aws iot. URL <https://aws.amazon.com/iot/>, viitattu 17.8.2021.
- [7] APOLINARSKI, W., IQBAL, U., JA PARREIRA, J. X. The gambas middleware and sdk for smart city applications. Julkaisusarjassa *2014 IEEE International conference on pervasive computing and communication workshops (PERCOM WORKSHOPS)* (2014), IEEE, 117–122.
- [8] ASHTON, K., ET AL. That ‘internet of things’ thing. *RFID journal* 22, 7 (2009), 97–114.
- [9] CHENG, B., LONGO, S., CIRILLO, F., BAUER, M., JA KOVACS, E. Building a big data platform for smart cities: Experience and lessons from santander. Julkaisusarjassa *2015 IEEE International Congress on Big Data* (2015), IEEE, 592–599.

- [10] DA CRUZ, M. A., RODRIGUES, J. J. P., AL-MUHTADI, J., KOROTAEV, V. V., JA DE ALBUQUERQUE, V. H. C. A reference model for internet of things middleware. *IEEE Internet of Things Journal* 5, 2 (2018), 871.
- [11] FAHMIDEH, M., JA ZOWGHI, D. An exploration of iot platform development. *Information Systems* 87 (2020), 101409.
- [12] FILIPPONI, L., VITALETTI, A., LANDI, G., MEMEO, V., LAURA, G., JA PUCCI, P. Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors. *Julkaisusarjassa 2010 Fourth International Conference on Sensor Technologies and Applications* (2010), IEEE, 281–286.
- [13] FINNEGAN, J., JA BROWN, S. A comparative survey of LPWA networking. *CoRR abs/1802.04222* (2018).
- [14] FREMANTLE, P. A reference architecture for the internet of things. *WSO2 White paper* (2015), 02–04.
- [15] GOOGLE. Google cloud iot solutions. URL <https://cloud.google.com/solutions/iot/>, viitattu 17.8.2021.
- [16] GUBBI, J., BUYYA, R., MARUSIC, S., JA PALANISWAMI, M. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems* 29, 7 (2013), 1645–1660.
- [17] GUTH, J., BREITENBÜCHER, U., FALKENTHAL, M., FREMANTLE, P., KOPP, O., LEYMAN, F., JA REINFURT, L. A detailed analysis of iot platform architectures: concepts, similarities, and differences. *Kirjassa Internet of everything*. Springer, 2018, ss. 81–101.
- [18] GUTH, J., BREITENBÜCHER, U., FALKENTHAL, M., LEYMAN, F., JA REINFURT, L. Comparison of iot platform architectures: A field study based on a reference architecture. *Julkaisusarjassa 2016 Cloudification of the Internet of Things (CIoT)* (Nov 2016), IEEE, 1–6.
- [19] HEJAZI, H., RAJAB, H., CINKLER, T., JA LENGYEL, L. Survey of platforms for massive iot. *Julkaisusarjassa 2018 IEEE International Conference on Future IoT Technologies (Future IoT)* (2018), IEEE, 1–8.
- [20] IBM. Internet of things on ibm cloud. URL <https://www.ibm.com/internet-of-things>, viitattu 17.8.2021.

- [21] KAA. Elevate your iot experience with business-ready iot dashboards. URL <https://www.kaaiot.com/>, viitattu 25.9.2021.
- [22] KAA. Kaa documentation. URL <https://docs.kaaiot.io/KAA/docs/v1.3.0/Welcome/>, viitattu 24.10.2021.
- [23] KALYAN, M., REDDY, V., JITESH, K., ASHIF, S., ET AL. Lpwan technologies for iot deployment. *International Journal of Electrical Engineering and Technology* 11, 3 (2020).
- [24] KHAN, S., SHAKIL, K. A., JA ALAM, M. Cloud-based big data analytics—a survey of current research and future directions. *Big data analytics* (2018), 595–604.
- [25] LEENDERS, G., CALLEBAUT, G., OTTOY, G., VAN DER PERRE, L., JA DE STRYCKER, L. Multi-rat for iot: The potential in combining lorawan and nb-iot. *IEEE Communications Magazine* 59, 12 (2021), 98–104.
- [26] LORA ALLIANCE. Lora alliance. URL <https://lora-alliance.org/>, viitattu 23.10.2021.
- [27] MADAKAM, S., RAMASWAMY, R., JA TRIPATHI, S. Internet of things (iot): A literature. *Journal of Computer and Communications* 3 (2015), 164–173.
- [28] MEKKI, K., BAJIC, E., CHAXEL, F., JA MEYER, F. A comparative study of lpwan technologies for large-scale iot deployment. *ICT express* 5, 1 (2019), 1–7.
- [29] MICROSOFT. Azure for students. URL <https://azure.microsoft.com/en-us/free/students/>, viitattu 2.4.2022.
- [30] MICROSOFT. Azure iot hub. URL <https://azure.microsoft.com/en-in/services/iot-hub/>, viitattu 17.8.2021.
- [31] MICROSOFT. Azure iot hub documentation. URL <https://docs.microsoft.com/en-in/azure/iot-hub>, viitattu 7.10.2021.
- [32] MICROSOFT. What is azure logic apps. URL <https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview>, viitattu 2.4.2022.
- [33] MICROSOFT POWER BI. Power bi. URL <https://powerbi.microsoft.com/en-us/>, viitattu 26.2.2022.

- [34] PASHA, S. Thingspeak based sensing and monitoring system for iot with matlab analysis. *International Journal of New Technology and Research (IJNTR)* 2, 6 (2016), 19–23.
- [35] PETROLO, R., LOSCRI, V., JA MITTON, N. Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms. *Transactions on Emerging Telecommunications Technologies* 28, 1 (2017), e2931.
- [36] RAMAKRISHNAN, R., SRIDHARAN, B., DOUCEUR, J. R., KASTURI, P., KRISHNAMACHARI-SAMPATH, B., KRISHNAMOORTHY, K., LI, P., MANU, M., MICHAYLOV, S., RAMOS, R., SHARMAN, N., XU, Z., BARAKAT, Y., DOUGLAS, C., DRAVES, R., NAIDU, S. S., SHASTRY, S., SIKARIA, A., SUN, S., JA VENKATESAN, R. Azure data lake store: A hyperscale distributed file service for big data analytics. *Julkaisusarjassa Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017* (2017), ACM, 51–63.
- [37] RAY, P. P. A survey of iot cloud platforms. *Future Computing and Informatics Journal* 1, 1-2 (2016), 35–46.
- [38] RAZZAQUE, M. A., MILOJEVIC-JEVRIC, M., PALADE, A., JA CLARKE, S. Middleware for internet of things: A survey. *IEEE Internet of Things Journal* 3, 1 (2016).
- [39] SENSATIVE. Sensative. URL <https://sensative.com/>, viitattu 2.1.2022.
- [40] SENSATIVE. Sensative strips configuration application. URL <https://strips-lora-config-app.service.sensative.net/profiles>, viitattu 26.2.2022.
- [41] SHI, W., CAO, J., ZHANG, Q., LI, Y., JA XU, L. Edge computing: Vision and challenges. *IEEE Internet of Things Journal* 3, 5 (2016), 637–646.
- [42] SINHA, R. S., WEI, Y., JA HWANG, S.-H. A survey on lpwa technology: Lora and nb-iot. *Ict Express* 3, 1 (2017), 14–21.
- [43] SOLDATOS, J., KEFALAKIS, N., HAUSWIRTH, M., SERRANO, M., CALBIMONTE, J.-P., RIAHI, M., ABERER, K., JAYARAMAN, P. P., JA ZASLAVSKY, A. Openiot: Open source internet-of-things in the cloud. *Interoperability and open-source solutions for the internet of things* (2015), 13–25.
- [44] TEMBOO. Temboo. URL <https://temboo.com>, viitattu 25.9.2021.

- [45] THINGPARK ACTILITY. Thingpark enterprise platform for dedicated enterprise iot lorawan networks. URL <https://www.actility.com/enterprise-iot-connectivity-solutions/>, viitattu 2.4.2022.
- [46] THINGSPEAK. Thingspeak for iot projects. URL <https://www.thingspeak.com>, viitattu 25.9.2021.
- [47] TRAFICOM. Matkaviestinverkkojen taajuudet ja luvanhaltijat. URL <https://www.traficom.fi/fi/viestinta/viestintaverkot/matkaviestinverkkojen-taajuudet-ja-luvanhaltijat>, viitattu 2.4.2022.
- [48] WANG, H., ZHOU, X., ZHOU, X., LIU, W., LI, W., JA BOUGUETTAYA, A. Adaptive service composition based on reinforcement learning. *Service-Oriented Computing* (2010), 92–107.