

Jan Terttula

Ratkaisut IoT-päätelaitteiden etähallintaan

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinööriytyö

30.11.2018

Tekijä Otsikko	Jan Terttula Ratkaisut IoT-päätelaitteiden etähallintaan
Sivumäärä Aika	48 sivua 30.11.2018
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	tieto- ja viestintäteknikka
Ammatillinen pääaine	Tietoverkot ja IoT
Ohjaajat	erikoisasiantuntija Olli Aaltonen lehtori Jukka Louhelainen
<p>Opinnäytetyön tarkoituksena oli tutkia IoT-laitteiden etähallintaan liittyviä ratkaisuja. Työssä keskityttiin sähköasemilla reunalaitteina toimivien IoT-yhdyskäytävien etähallintaan sekä mittauksia suorittavien anturijärjestelmien etähallintaan.</p> <p>Aluksi selvitettiin, mitä etähallinnalla tarkoitetaan ja mitä hyötyjä ja haasteita sen käyttöönotto aiheuttaa. Lisäksi perehdyttiin siihen, mitä Internet of Things tarkoittaa ja mitä Fingrid Oyj pyrkii IoT-järjestelmällä saavuttamaan.</p> <p>Työssä esiteltiin IoT-yhdyskäytävä ja sen toiminta sekä kaksi järjestelmänhallintatyökalua, joilla yhdyskäytävää voidaan hallita. Anturijärjestelmien hallintaa varten esiteltiin Microsoft Azure -pilvipalvelualusta ja sen IoT-ratkaisun rakentamiseen tarjoamat palvelut.</p> <p>Insinööriyön käytännön osuudessa rakennettiin testiratkaisuksi IoT-järjestelmä, joka hyödynsi Ubuntu Landscape -järjestelmänhallintatyökalua sekä Microsoft Azurea eri palveluihin. Lopputuloksena analysoitiin rakennetun järjestelmän toimintaa ja sitä, soveltuisiko testiratkaisun kaltainen toteutus Fingridin sähköasemille tuotantoon.</p> <p>Landscape todettiin erittäin toimivaksi, mutta liian kalliiksi työkaluksi reunalaitteen hallintatarpeisiin. Azure ja sen palvelut osoittautuivat hyvin kyvykkäiksi ja kustannustehokkaiksi vaihtoehdoiksi IoT-järjestelmän toteutuksessa. Fingrid tulee mitä luultavimmin jatkamaan yhteistyötä Azuren kanssa IoT-järjestelmän kehityksessä.</p>	
Avainsanat	Internet of Things, Etähallinta, Azure, Ubuntu, Landscape, Docker, Anturi

Author Title	Jan Terttula Solutions for Remote Management of IoT Terminal Units.
Number of Pages Date	48 pages 30 November 2018
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Data Networks and IoT
Instructors	Olli Aaltonen, Special Adviser Jukka Louhelainen, Senior Lecturer
<p>The purpose of this thesis was to examine the remote management solutions of IoT devices. The thesis consists of two parts: the remote management of the IoT Gateway device and the remote management of the IoT sensor system.</p> <p>First, the meaning and the pros and cons of remote management were investigated. Secondly, the concept Internet of Things and the possibilities it offers were looked into.</p> <p>The IoT Gateway and its purpose along with two system management tools were introduced in this thesis. Microsoft Azure cloud computing service and its tools were also introduced. Azure was meant to be used in managing the IoT sensor systems located in electrical substations.</p> <p>In the thesis' practical part an IoT system was built as a test. The system was built around Ubuntu Landscape system management tool and Microsoft Azure and its multiple services. Eventually the test system was analyzed to see if it matches Fingrid's IoT needs.</p> <p>In conclusion it was determined that Landscape is a very capable but too expensive tool to be used in Fingrid's IoT case. Azure was confirmed to be very capable and cost-efficient platform and will most likely be used in the future IoT projects at Fingrid.</p>	
Keywords	Internet of Things, Remote Management, Azure, Ubuntu, Landscape, Docker, Sensor

Sisällys

Lyhenteet

1	Johdanto	1
2	Kohdeyritys	2
2.1	Fingrid Oyj	2
2.2	Digitaalinen sähköasema	4
2.3	IoT-ratkaisut kunnonhallinnan hyötyyn	4
3	Etähallinta	5
3.1	Etähallinta yleisesti	5
3.2	Etähallintavaatimukset	5
3.3	Etähallinnan hyödyt ja haasteet	6
4	IoT	7
4.1	IoT yleisesti	7
4.2	IoT Fingridillä	9
4.3	IoT-protokollat	10
4.3.1	OSI-malli	10
4.3.2	MQTT	11
4.3.3	AMQP	11
4.4	Konttitekologia	13
5	IoT-Gateway	13
5.1	Reunalaite	13
5.2	Ubuntu Landscape	15
5.3	VMware Pulse	16
6	Microsoft Azure	17
6.1	Azure Portal ja Azure CLI	18
6.2	Azure IoT Hub	19
6.3	Azure IoT solution accelerators	20
6.4	Azure IoT Central	20
6.5	Azure IoT Edge	21
6.6	Azure Container Registry	23

6.7	Azure Stream Analytics	24
6.8	Azure Blob Storage	24
7	Testiratkaisu	25
7.1	Fyysiset reunalaitteet	25
7.2	Ubuntu-hallinta	26
7.2.1	Landscape SaaS	26
7.2.2	Landscape On-premises	27
7.2.3	VMware Pulse	28
7.2.4	Landscapen toiminta	29
7.2.5	Vaihtoehtojen vertailu	31
7.3	Telemetry	34
7.3.1	IoT Edge testissä	35
7.3.2	Datan varastointi	36
7.3.3	Anturimoduulien hallinta	38
7.3.4	Toimintaperiaate	40
7.3.5	Visualisointi	42
8	Yhteenveto	43
	Lähteet	46

Lyhenteet

ACR	Azure Container Registry. Microsoftin palvelu, jolla voidaan hallita monipuolisesti Docker-konttiohjelmistolla tuotettuja kontteja.
AMQP	Advanced Message Queuing Protocol. Luotettava ja alustasta riippumaton avoimen standardin viestintäprotokolla.
ASA	Azure Stream Analytics. Palvelu, joka kykenee analysoimaan suuria määriä reaaliaikaista datavirtaa.
BIOS	Basic Input-Output System. Ohjelma, joka hallitsee tietokoneen käyttöjärjestelmän toimintaa.
CLI	Command Line Interface. Komentorivi.
DIOT	IoT-ratkaisut kunnonhallinnan hyötyyn. Fingridin tutkimus- ja kehityshanke.
E2E	End-to-end. Periaate, jonka mukaan yhteys kulkee alkupisteestä loppupisteeseen välittämättä yhteyden varrella olevista solmukohtista.
HTTP	Hypertext Transfer Protocol. WWW-palvelimien ja selainten käyttämä tiedonsiirtoprotokolla.
HTTPS	Hypertext Transfer Protocol Secure. HTTP-protokollan ja TLS/SSL-salausprotokollan yhdistelmä, jolla saavutetaan turvallisempi tiedonsiirto verkossa.
IoT	Internet of Things. IoT, suomeksi esineiden internet tarkoittaa internetverkon leviämistä yhä laajemmalle erilaisiin laitteisiin, joita voidaan ohjata internetverkon yli.
JSON	JavaScript Object Notation. Tiedonvälityksessä käytettävä yksinkertainen tiedostomuoto.
Kontti	Docker-container. Docker-ohjelmistolla luotu virtuaalinen paketti, johon voidaan pakata sovellus sekä sen taustajärjestelmä.

MASE	Microsoft Azure Storage Explorer. Azureen luotujen datavarastojen tarkasteluun kehitetty erillinen sovellus.
MQTT	Message Queuing Telemetry Transport. Yksinkertainen viestintäprotokolla, joka on kehitetty käytettäväksi hitaassa tai epävakaassa ympäristössä.
OSI-malli	Open Systems Interconnection Reference Model. Kuvaa tiedonsiirtoprotokollien yhdistelmän seitsemässä kerroksessa.
PaaS	Platform as a Service. Palveluna tarjottava sovellusalausta, joka mahdollistaa käyttäjälle monipuoliset työkalut palvelun kehitykseen.
Pilvi	Pilvipalvelu. Termi, jota käytetään, kun tiedostot ja sovellukset sijaitsevat ja pyörivät pilvipalvelua tarjoavan yrityksen palvelimilla.
RBAC	Role-based Access Control. Rooleihin pohjautuva käyttöoikeusmalli, joka estää yhteyden niistä lähteistä, joita ei ole konfiguroitu sallituiksi rooleiksi.
SaaS	Software as a Service. Palvelu, joka tarjotaan asiakkaalle ohjelmistona.
SASL	Simple Authentication and Security Layer. Salausmenetelmä, jolla voidaan suojata dataa internetliikenteessä.
SQL	Structured Query Language. Kyselykieli, jolla voidaan tehdä hakuja ja muutoksia relaatiotietokantaan.
TLS	Transport Layer Protocol. IP-verkkojen yli kulkevan tietoliikenteen salaukseen kehitetty protokolla. TLS:n edeltäjä on Secure Sockets Layer (SSL).

1 Johdanto

Yhteiskunta digitalisoituu hurjalla vauhdilla ja Internet of Things on saamassa huomattavasti jalansijaa digitalisaation kehittyessä. Internet of Things (IoT) eli esineiden internet tarkoittaa internetverkon leviämistä kaikenlaisiin monitoroitaviin ja hallittaviin laitteisiin. IoT on yleistymässä kaikkialla, yksityisasiakkaille suunnatuissa palveluissa sekä yhtiöiden sisäisissä toimissa.

Tämä insinöörityö tehtiin Fingrid Oyj:lle. Osana Fingridin strategista ”Digitaalinen sähköasema” -hanketta sähköasemille ollaan ottamassa käyttöön IoT-tekniikalla toimivia antureita, joilla voidaan mitata erilaisia arvoja sähköasemien komponenteissa. Mittaus tuloksien perusteella voidaan paikantaa vikaantuneet komponentit ja tilata niille tarvittavat huoltotoimenpiteet. Näin vältetään mahdollisen vian aiheuttamilta vaara- ja ongelmatilanteilta.

Insinöörityön tarkoituksena oli tutkia IoT-laitteiden etähallintaan liittyviä ratkaisuja. Työssä esitellään etähallinta sekä sen hyöty- ja haittapuolet. Työssä esitellään myös IoT käsitteenä sekä IoT:n merkitys yhteiskunnalle ja Fingridille.

Sähköasemien IoT-etähallinta jakautuu kahteen osaan: IoT-yhdyskäytävän hallintaan ja anturijärjestelmän hallintaan. Yhdyskäytävän hallinnan osalta perehdyttiin kahteen järjestelmänhallintatyökaluun: Ubuntu Landscapeen ja VMware Pulseen. Lisäksi selvitettiin, soveltuuko Microsoft Azure -pilvipalvelualusta ja sen tarjoamat alapalvelut anturijärjestelmien hallintaan.

Työn lopussa rakennettiin testiratkaisuna IoT-järjestelmä käyttäen hyödyksi Landscape työkalua ja Microsoft Azure -alustaa. Järjestelmän rakennusvaiheet dokumentoitiin. IoT-järjestelmän toimintaan perehdyttiin ja lopputuloksena syntyi analyysi siitä, vastaako testiratkaisun kaltainen toteutus Fingridin IoT-tarpeisiin ja voisiko toteutusta hyödyntää sähköasemien IoT-järjestelmää suunniteltaessa.

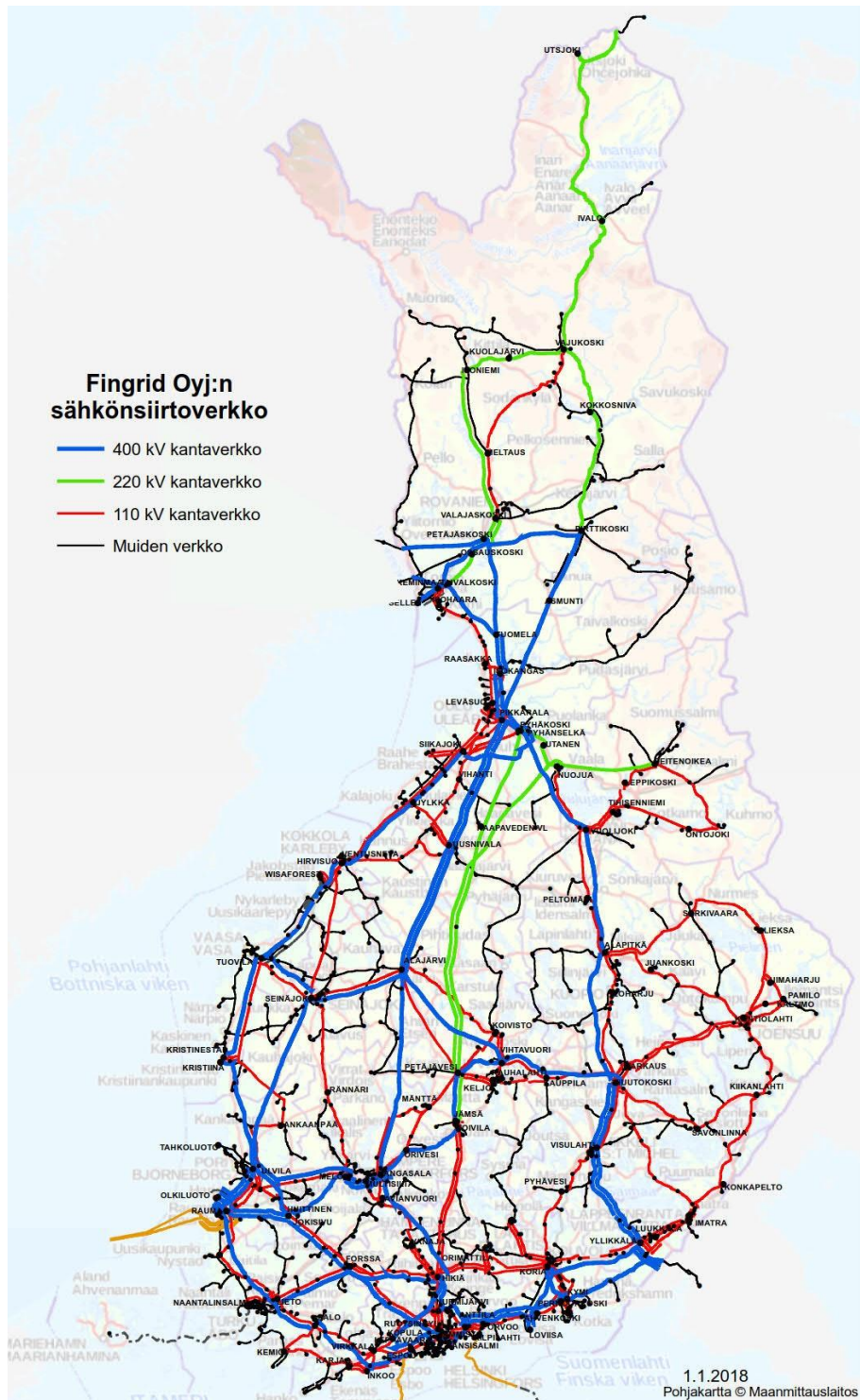
2 Kohdeyritys

2.1 Fingrid Oyj

Fingrid Oyj on suomalainen julkinen osakeyhtiö, jonka tehtävänä on vastata Suomen kantaverkon sähkönsiirrosta. Maanlaajuinen kantaverkko on keskeinen osa Suomen sähköjärjestelmää. (Fingridin esittely 2018.)

Fingrid on perustettu vuonna 1996, mutta operatiivisen toimintansa yhtiö aloitti vasta seuraavana vuonna. Nykyään Fingrid työllistää noin 350 ihmistä ympäri Suomea, joista suurin osa työskentelee Helsingin pääkonttorilla. Tämän lisäksi toimipisteitä löytyy viideltä muultakin paikkakunnalta. (Fingridin esittely 2018.)

Suomen kantaverkko käsittää yli 14000 kilometriä 400, 220 ja 110 kilovoltin voimajohtoja sekä yli sata sähköasemaa. Lisäksi kantaverkkoon ovat liittyneet suuret tehtaot ja voimalaitokset sekä alueelliset jakeluverkot. Fingridin tehtäviin kuuluu kantaverkon käytön suunnittelu ja valvonta sekä verkon ylläpito ja kehittäminen. Kaikesta Suomessa käytetystä sähköstä noin kolme neljäsosaa kulkee Fingridin verkon kautta. Suomen kantaverkko on osa yhteispohjoismaista sähköjärjestelmää, joka on kytketty Keski-Euroopan järjestelmään tasavirtayhteyksin. Suomesta on myös tasasähköyhteydet Venäjälle ja Vieroon. (Fingridin esittely 2018.)



Kuva 1. Fingrid Oyj:n sähkösiirtoverkko 1.1.2018 (Fingridin esittely 2018.)

Verkko on rakennettu rengasverkoksi siten, että yksittäisen johdon häiriötilanteessa sähkönsiirto ei keskeydy, vaan kiertää toista kautta kohteeseen. Kantaverkkoon on liitettyä varavoimalaitoksia, jotka saadaan tarvittaessa nopeasti käyntiin, jos verkossa tai voimalaitoksissa ilmenee suurempia häiriötilanteita. (Fingridin esittely 2018.)

2.2 Digitaalinen sähköasema

”Digitaalinen sähköasema” on Fingrid Oyj:n strateginen kehityshanke, jonka tavoitteena on tutkia ja kartoittaa laaja-alaisesti, mitä täysin digitaaliseen sähköasemaan siirtyminen vaatisi rakentamisen, suojauksen toimivuuden, asema-automaation, konfiguraatioiden hallinnan, tietoturvan, osaamisen, primäärlaitteiden, sovitussyksiköiden sekä kunnonhallinnan näkökulmasta. Hankkeessa tehdään ensin selvityksiä nykYTEknologian kypsyydestä ja mahdollisuuksista sekä edetään varsinaiseen pilottihankkeeseen sopivaksi katsotun yhteistyökumppaniyhtiön kanssa sopivassa kohteessa. (Strategiset hankkeet 2017, 17.)

Digitaalinen sähköasema -hanke on laajuutensa takia pilkottu pienempiin osiin, joista jokainen osa on oma hankkeensa.

2.3 IoT-ratkaisut kunnonhallinnan hyötyyn

Yhtenä osana strategista Digitaalinen sähköasema -hanketta on tutkimus- ja kehityshanke ”IoT-ratkaisut kunnonhallinnan hyötyyn” (DIOT). Hankkeen tavoitteena on kehittää kunnonvalvontamenetelmiä, joilla pystyttäisiin varmistamaan verkon korkea käyttövarmuus, tehostamaan verkon käytettävyyttä ja havaitsemaan kohteet, joissa kunnossapitotöitä tarvitaan. (Laitinen 2018, 3.)

DIOT-hanke edellyttää Internet of Things (IoT) -anturiratkaisujen kehittämistä, langattomien tiedonsiirtomenetelmien hyödyntämistä sekä nykyaikaisten pilvialustojen ja tehokaiden visualisointityökalujen käyttöönottoa. Hankkeessa laaditaan Fingridille IoT-konsepti mittauskohteiden, IoT-arkkitehtuurin, tietoturvan, analytiikan sekä visualisoinnin ja muutoksen hallinnan osalta. (Laitinen 2018, 3.)

3 Etähallinta

3.1 Etähallinta yleisesti

Yksinkertaisuudessaan etähallinnalla tarkoitetaan sitä, että laitetta pystytään hallitsemaan muualtakin kuin laitteen fyysisestä sijaintipaikasta. Monesti ajatellaan tämän tarkoittavan sitä, että laitteen hallinta suoritetaan esimerkiksi Fingridin tapauksessa sähköaseman sijaan toimistolta. Nykyaikana tekniikka on jo niin kehittynyttä, että etähallintayhteys voidaan muodostaa mistä tahansa, kunhan vain internet-yhteys on olemassa.

Jotta laitetta pystytään etähallitsemaan, tarvitaan etähallintapalvelin sekä hallittavalle laitteelle asennettava etähallintasovellus. Palvelin ja sovellus viestivät verkon välityksellä keskenään keräten laitteen järjestelmä- ja sovellustietoja. Palvelimen kautta hallittavaa laitetta on mahdollista monitoroida ja tarvittaessa päivittää. (Remote Monitoring & Management (RMM) Defined.)

3.2 Etähallintavaatimukset

Fingridillä suunnitteilla olevassa ratkaisussa etähallinnan voidaan katsoa jakautuvan kahteen osa-alueeseen: IoT-yhdyskäytävänä toimivan reunalaitteen hallintaan sekä IoT-anturijärjestelmän eli telemetriadataa tuottavan järjestelmän hallintaan. Telemetrialla tarkoitetaan laitteilta kerättyjen mittausarvojen ja laitetietojen lähettämistä toisaalla sijaitsevaan tietojärjestelmään (Techopedia, Telemetry). Sähköasemalla sijaitsevaan reunalaitteeseen tulee pystyä etähallinnalla asentamaan ohjelmisto- ja tietoturvapäivityksiä sekä laitteen tilaa on pystyttävä valvomaan. Päivittäminen on avaintekijä laitteen tietoturvan kannalta. Lisäksi laite on pystyttävä etähallinnan avulla sekä sammuttamaan että käynnistämään uudelleen.

Pilvessä sekä osittain myös reunalaitteen sisällä toimivaa anturijärjestelmää on myös pystyttävä etähallitsemaan. Pilvellä eli pilvipalvelulla tarkoitetaan sitä, kun tiedostot ja sovellukset sijaitsevat ja pyörivät oikeasti palvelua tarjoavan yrityksen palvelinkeskuk-sissa eivätkä oman tietokoneen kovalevyllä (Eronen 2016). Etähallinnalla on voitava määritellä järjestelmään liitettävät anturit sekä niiden toimintatapa. Toimintatapa pitää sisällään anturien datan keräykseen ja datan lähetykseen liittyviä seikkoja. Kuinka usein ja millaiselta aikaväliltä dataa kerätään? Missä formaatissa kerätty data lähetetään

eteenpäin ja minne? Anturijärjestelmään on myös tärkeää voida lisätä automaattisesti suoritettavia toimintoja, mikäli jokin anturilta vastaanotettu parametri ei vastaa odotettua arvoa.

3.3 Etähallinnan hyödyt ja haasteet

Etähallintaominaisuus tarjoaa käyttäjälle lukemattomien hyötyjen lisäksi myös joitakin riskejä. Hallintajärjestelmän tavoite on maksimoida saatavilla oleva hyöty minimoiden järjestelmästä aiheutuvat haitat. Tavoitteen saavuttaminen vaatii usein runsaasti resursseja ja monien eri tahojen välistä yhteistyötä.

Laitteiden etähallinta tarjoaa parhaimmillaan hyvinkin paljon lisäarvoa erilaisille järjestelmäkokonaisuuksille. Itsestään selvin asia on tietenkin se, että laitetta voidaan hallita menettä itse laitteen luokse. Näin säästetään sekä aikaa että esimerkiksi polttoaineisiin kuluva rahaa, koska Suomessa sähköasemien väliset etäisyydet ovat pitkiä. Etähallintajärjestelmään liitetyt visualisointityökalut mahdollistavat lukemattomien laitteiden samanaikaisen monitoroimisen ja toimintojen suorittamisen. Monissa tapauksissa järjestelmä pystyy havainnoimaan mahdollisia häiriöitä ja ilmoittamaan virhetilanteesta automaattisesti eteenpäin, jolloin tarvittaviin korjaustoimiin voidaan heti ryhtyä. (Remote Monitoring & Management (RMM) Defined.)

Suurimpana riskitekijänä etähallintaratkaisuisissa pidetään yleensä järjestelmien tietoturvaa. Etähallinnassa käytetään paljon langattomia tiedonsiirtotekniikoita, joten suojauksen on tällöin oltava kunnossa, ettei kukaan ulkopuolinen pääse tietoon tai laitteistoon käsiksi. Hallittavan laitteen luona paikan päällä on luultavimmin aina turvalliseksi rakennettu verkkoympäristö, josta hallintayhteyden muodostaminen on turvallista. Etähallintayhteyden voi sen sijaan avata myös sellaisista verkoista ja sellaisten tukiasemien kautta, jotka saattavat olla tietoturvaominaisuuksiltaan puutteellisia. Tällöin haittaohjelmatartunta tai tietojen vakoilu saattaa olla mahdollista. (Remote Monitoring & Management (RMM) Defined.)

Etähallintayhteyden vikaantuminen tai erilaiset ongelmatilanteet ovat myös mahdollisia haittoja. Jos yhteys yhdelle kriittiselle hallittavalle laitteelle syystä tai toisesta menetetään, voi se johtaa kokonaisten suurten järjestelmien käytettävyysongelmiin. Etähallintakomennoissa voi myös ilmetä virheitä, joiden vuoksi komento lähtee hallittavalle laitteelle

erilaisena kuin oli alun perin tarkoitettu. Tällöinkin voi koitua ongelmia järjestelmän käytettävydessä tai toiminnassa. (Remote Monitoring & Management (RMM) Defined.)

4 IoT

4.1 IoT yleisesti

Internet of Things (IoT), suomennettuna esineiden internet on seuraava tietoteknologian valtava kehitysaskel, joka mahdollistaa uusia mullistavia tapoja hyödyntää tiedonsiirrossa käytettäviä verkkoratkaisuja. Termi "Internet of Things" muodostuu esineistä (things) ja internetistä. Nämä esineet keskustelevat keskenään internetverkon välityksellä. IoT tuo myös mukanaan paljon riskejä, jonka vuoksi on ensiarvoisen tärkeää, että tietoturvaratkaisut pysyvät muun kehityksen mukana. Ajatuksen tasolla IoT on ollut olemassa jo kaksi vuosikymmentä, mutta nyt sitä ollaan tuomassa entistä enemmän osaksi jokapäiväistä elämää. (Rouse 2016.)

Karkeasti ilmaistuna IoT tarkoittaa internetyhteyksien leviämistä myös sellaisiin laitteisiin, joilla ei perinteisesti ole ollut mahdollista päästä internetiin. Olemme tottuneet saamaan nopeasti ja vaivattomasti muodostettua verkkoyhteyden tietokoneilla, puhelimilla, tableteilla ja älytelevisioilla. Tulevaisuudessa onkin odotettavissa, että lähes jokainen sähköä kuluttava laite olisi jollain tasolla liitettyä internetverkkoon, jolloin laitteen hallinta ja monitoroiminen ovat mahdollista internetyhteyksien yli.

IoT:n mahdollistamiin palveluihin mukaan pääseminen voi olla aikaa vievää ja monimutkaista sekä tietoturvan osalta epävarmaa. Hyvin suunniteltu ja laadukkaasti loppuun asti saatettu IoT-järjestelmä säästää kuitenkin sekä aikaa että rahaa. Se tekee työnteosta tehokkaampaa ja vähemmän riskialtista. IoT:n ansiosta voidaan toteuttaa monipuolisempia ja tehokkaampia työkaluja asiakkaiden tarpeisiin. (Rouse 2016.)

Pienen pienistä yksinkertaisia mittauksia suorittavista sähkökomponenteista lähtien esimerkiksi ajoneuvot saattavat IoT-tekniikan kehittyessä siirtyä käyttämään ajotoimintojensa ohjaamiseen internetverkkoa. Autot voivat mahdollisesti viestiä keskenään ajoneuvoista ja -reiteistä, jolloin liikenneonnettomuuksia ei pääse tapahtumaan. Tämä tosin vaatisi vahvan tietoturvasuojauksen, ettei yksikään kyberrikollinen onnistu saamaan ajo-

neuvoa omaan hallintaansa ja pääse tekemään sillä tuhoa. On myös tärkeää, ettei kriittisissä automatisoiduissa IoT-toteutuksissa pääse tapahtumaan ohjelmistovirheitä. Esimerkiksi ajoneuvon ohjausta säätelevän komponentin on noudatettava nopeusrajoituksia, eikä se saa ohjelmistovirheen vuoksi nostaa auton nopeutta maantienopeuteen taa-jama-alueella. (Huomo, Vähä-Heikkilä & Halunen 2018, 13.)

Koska IoT laajenee koko ajan huomattavasti, tarjoaa se tietoturvarikollisille laajan toimintakentän hyökkäyksille ja tietomurroille. IoT-järjestelmä voi koostua valtavasta määrästä laitteita, joita saatetaan päivittää epäsäännöllisesti tai ei ollenkaan. Kun laitteita ei päivitetä, ne ovat haavoittuvaisia kyberhyökkäyksille. Laitteet voivat olla niin tiiviisti yhdistetty toisiinsa, että rikollisen tarvitsee murtautua vain yhdestä tietoturva-aukosta sisään, josta koko IoT-järjestelmä voidaan lamauttaa käyttökelttomaksi. On siis tärkeää, että tietoturva-asiat otetaan hyvin huomioon IoT-ratkaisua suunniteltaessa, rakentaessa ja ylläpidettäessä. (Rouse 2016.)

Työmarkkinoilla IoT:n mahdollistama automaatio näyttölee suurta uhkaa joitakin aloja kohtaan. Automatisoinnin kehittyessä jotkin monotoniset työtehtävät kuten liukuhihna- ja varastotyö pystytään tulevaisuudessa korvaamaan yhä tehokkaammin koneilla. Siirtymävaihe on hintava, mutta se luultavasti tuottaa yhtiölle ajan mittaan taloudellista hyötyä, kun voidaan korvata viisikymmentä varastotyöntekijää kymmenellä robotilla ja muutamalla laiteinsinöörillä. (Huomo ym. 2018, 13.)

Elokuussa 2018 IoT-Analytics -verkkosivusto julkaisi artikkelin, jossa analysoitiin IoT-laitteiden määrää ja sen kasvun ennustetta. Artikkelin mukaan vuonna 2018 IoT-laitteita oli asennettuna noin 18 miljardia, joista vasta 7 miljardia oli käytössä. Vuoden 2019 aikana ennusteiden mukaan käytössä olevien IoT-laitteiden määrä ohittaisi maailman väkiluvun. Vuoteen 2025 mennessä laitteita olisi asennettu 34 miljardia ja käytössä olevia laitteita olisi 21,5 miljardia. Eri organisaatioiden ennusteissa on paljonkin eroja, mutta kaikki ennusteet viittaavat siihen, että IoT-laitteiden määrä tulee vähintään kolminkertaistumaan seuraavan kymmenen vuoden aikana. (State of the IoT 2018.)

4.2 IoT Fingridillä

Fingrid on implementoimassa sähköasemilleen erilaisia anturijärjestelmiä. Järjestelmien tavoitteena on saada aikaan parempi näkyvyys laitekantaan ja omaisuuden kuntoon, jolloin mahdollisten vikojen kehittymistä voidaan ennakoida hallitummin. Samalla varmennetaan sähköaseman palvelujen laatua. Anturijärjestelmillä voidaan myös vähentää käyttökeskeytyksiä menetelmäkehityksellä. Kustannustehokkuus saadaan paremmaksi, kun kunnossapitotöitä voidaan kohdentaa tarveperusteisesti. Joillekin sähköasemille on jo testeissä rakennettu etähallittavia anturijärjestelmiä, mutta lopullista toteutusratkaisua ei ole vielä valittu.

Olkiluodon sähköasemalla sattui vaarallinen tilanne kesällä 2018, kun virtamuuntajassa sattui räjähdys, joka aiheutti tulipalon. Tästä johtuen Olkiluodon ydinvoimalan yksi reaktoreista irtosi varotoimenpiteenä verkosta. Viankorjauksen aikana toinenkin reaktori irtosi tuntemattomasta syystä verkosta. Tämän vuoksi sähkö oli vaarassa loppua Suomesta, sillä sähkönkulutus ylitti sähköntuotannon sekä -tuonnin ulkomailta. Muun muassa Meilahden sairaala-alueella koettiin noin tunnin mittainen sähkökatko Olkiluodon tapauksen aikaan. Kuitenkaan varmuutta sille, liittyikö sairaala-alueen sähkökatko Olkiluodon muuntajaräjähdykseen, ei pystytty todentamaan. (Olkiluoto 1 kytkettiin takaisin kantaverkkoon 2018.)

IoT-etähallittavilla anturijärjestelmillä pyritään toteuttamaan sähköasemien kriittisiin komponentteihin jatkuva valvonta. Olkiluodon muuntajaräjähdyksen kaltaiset tapahtumat voidaan tulevaisuudessa mahdollisesti ehkäistä, kun asennetaan kunnonvalvontajärjestelmiä kaikkiin sellaista vaativiin kohteisiin. Päämuuntajissa on ollut kunnonvalvonta jo käytössä. IoT-järjestelmillä kunnonvalvonta ollaan toteuttamassa virtamuuntajiin ja lisäksi myös muihin komponentteihin. Valvonta toteutetaan erilaisilla antureilla, jotka mittaavat muun muassa lämpötilaa, kosteutta, ilmanpainetta, jännitettä ja värinää. Antureilta data lähtee tarvittavien verkkolaitteiden ja sovellusten kautta pilveen. Jos joltain anturien mittaamaa tavallisesta poikkeavaa dataa havaitaan, pystytään heti tekemään tarvittavat toimenpiteet ilman, että suuria vahinkoja pääsee syntymään.

Anturijärjestelmiin liittyen Fingridillä on käynnissä useita erilaisia testiratkaisuja eri sähköasemilla. Ratkaisujen toimivuutta seurataan ja tulevaisuudessa parhaaksi tai parhaiksi valitut toteutukset rakennetaan mahdollisuuksien mukaan kaikille Fingridin sähköasemille sekä muihin valvontaa vaativiin paikkoihin. Tämän opinnäytetyön tarkoituksena on

tutkia ja etsiä sopiva ratkaisu IoT-laitteiden etähallintaan. Ensiksi perehdytään sähkö-asemille sijoitettavien gateway-laitteiden etähallintaan. Vahvimpina ehdokkaina tähän ovat Ubuntu Landscape sekä VMware Pulse. Lisäksi tutkitaan Microsoft Azuren soveltuvuutta anturijärjestelmien etähallinnan toteutuksessa. Gateway-laitteiden käyttöjärjestelmien osalta testejä jatketaan Ubuntuun kanssa.

4.3 IoT-protokollat

4.3.1 OSI-malli

Open Systems Interconnection Reference Model eli tuttavallisemmin OSI-malli havainnollistaa tiedonsiirrossa käytettävät protokollat seitsemään kerrokseen. Kukin kerroksesta käyttää yhtä kerrosta alemman palveluja ja tarjoaa omia palvelujaan yhtä kerrosta ylemmäs. OSI-malli on kehitetty tietokonejärjestelmien suunnittelun helpottamiseksi. OSI-mallin kerrokset ovat alhaalta päin lueteltuna fyysinen kerros, siirtokerros, verkkokerros, kuljetuskerros, istuntokerros, esitystapakerros ja sovelluskerros. (Shaw 2018.)



Kuva 2. OSI-mallin kerrokset ja protokollat (Wikipedia, OSI-malli).

Fyysinen kerros määrittelee tiedonsiirron fyysisen menetelmän. Siirtokerros hoitaa paikallisen liikennöinnin lähiverkon laitteiden välillä. Verkkokerroksen tehtävä on löytää ja reitittää liikenne globaalisti perille internetissä. Kuljetuskerros huolehtii pakettien toimit-

tamisesta perille oikeassa järjestyksessä. Istuntokerros hallinnoi useiden samassa yhteydessä kulkevien istuntojen kanavointia. Esitystapakerros muuttaa datan oikeaan muotoon käyttäjän tulkittavaksi. Sovelluskerrokseen sijoittuvat käyttäjälle näkyvät tiedonsiirtoa tarvitsevat sovellukset. (Shaw 2018.)

4.3.2 MQTT

Message Queuing Telemetry Transport (MQTT) on erittäin kevyt ja yksinkertainen protokolla, joka on suunniteltu käytettäväksi machine-to-machine-liikenteessä siirtonopeudeltaan hitaassa tai epävakaassa ympäristössä. Machine-to-machine tarkoittaa sitä, että laitteet ovat yhteydessä keskenään ilman ihmisvuorovaikutusta. MQTT-protokolla toimii OSI-mallin sovelluskerroksella. Protokolla kehitettiin vuonna 1999 ja sen kehitystä jatkoi IBM vuonna 2010. Samana vuonna IBM julkaisi sen avoimena protokollana. Tällä hetkellä MQTT-protokolla on yksi IoT-kehityksen tärkeimpiä protokollia, jonka vuoksi useimmista IoT-pilvipalveluista löytyy protokollalle tuki. (Ojala 2017, 31.)

MQTT-viesti koostuu kiinteämittaisesta otsakkeesta ja tarpeen vaatiessa jatko-otsakkeesta, jonka pituus voi vaihdella. Viestiin sisältyy lisäksi aihe- sekä datakenttä. Suurimmillaan yhden MQTT-viestin koko voi olla 256 megatavua. IP-protokollaan sitoutumisen kanssa MQTT-protokollassa on havaittu ongelmia, jonka vuoksi protokollasta on kehitetty uusia versioita. MQTT-SN eli MQTT Sensor Networks -protokolla mahdollistaa jatkuvan viestityksen toiminnan pienen energialähteen kanssa. Tällöin laite voi olla niin sanotussa nukkumatilassa, mutta silti varastoida vastaanotettuja viestejä. Herätessään se välittää saadut viestit eteenpäin ja palaa takaisin nukkumatilaan. Tämä menetelmä säästää pitkällä tähtäimellä huomattavasti energiaa. (Ojala 2017, 32-33.)

4.3.3 AMQP

Advanced Message Queuing Protocol (AMQP) on OSI-mallin sovelluskerroksella toimiva avoimen standardin viestiprotokolla, joka on suunniteltu toimimaan väliohjelmistona useiden erilaisten prosessien, sovellusten ja järjestelmien viestinvaihdossa. AMQP kehitettiin, koska haluttiin viestintäprotokolla, joka ei ole sidoksissa standardeihin eikä normeihin vaan toimii eri sovellusten välillä alustasta riippumatta. Sen takia AMQP tukeekin useita eri viestintäsovelluksia ja kommunikaatiomalleja. (Tezer 2013.)

Vahvuuksinaan AMQP tarjoaa laadukkaan ja luotettavan toimituksen viesteilleen sovelusten ja prosessien välillä. Lisäksi AMQP lupaa toimittaa viestit nopeasti ja takaa ilmoituksen toimitetuille viesteille, kun viestit vastaanotetaan onnistuneesti. AMQP soveltuu monitoroimiseen ja päivitysten jakamiseen maailmanlaajuisesti. Se soveltuu myös eri järjestelmien keskustelun yhdistämiseen, viestin jakamiseen useille vastaanottajille, epäaktiivisten asiakasohjelmien myöhempään tiedonhakuun, järjestelmien täysin asynkronisten toimintojen esittelyyn sekä sovellusten käyttöönoton luotettavuuden ja käytettävyyden lisäämiseen. Lisäksi AMQP mahdollistaa palvelimen vastata välittömästi sille osoitettuihin pyyntöihin ja siirtää aikaa kuluttavat tehtävät myöhempää käsittelyä varten. (Tezer 2013.)

AMQP-protokollan terminologia koostuu viestin välittäjästä, viestistä, kuluttajasta, tuottajasta, vaihdosta, jonosta ja säännöistä. Välittäjä on se sovellus, joka ottaa AMQP-mallin käyttöön ja joka hyväksyy yhteyden asiakkaaseen viestin reititystä tai jonotusta varten. Viesti on se data, joka lähetetään tai reititetään. Kuluttaja on sovellus, joka vastaanottaa viestin. Tuottaja on sovellus, joka laittaa viestin jonoon vaihdon välityksellä. Vaihto on osa välittäjää ja se vastaanottaa viestit ja reitittää ne jonoon. Jono on yhteisö, johon viestit siirtyvät ja josta kuluttajat ne vastaanottavat. Erinäiset säännöt määrittelevät, miten viestit jakautuvat vaihdoista jonoihin. (Tezer 2013.)

AMQP perustuu viestien jonorakenteeseen, josta viestit lähtevät käyttäen erityyppisiä virtausohjattuja menetelmiä. Viestin välittäjä toimii AMQP:ssa siten, että se kääntää sovelluksen, joka vastaanottaa alkuperäiset viestit ja reitittää ne sopiville vastaanottajille eli kuluttajille. Vastaanotetut viestit prosessoidaan vaihdossa ja reititetään yhteen tai useampaan jonoon. Reititystapa riippuu vaihtotavasta, joita on useita erilaisia. Suora vaihto (Direct Exchange) käyttää apunaan reititysavaimia. Tyypillinen käyttötapa suoralle vaihdolle on työntekijöiden väliset kuormanjakotehtävät. Fanout-vaihto (Fanout Exchange) ei välitä avaimista, vaan lähettää kaikki viestit kaikkiin siihen liitettyihin jonoihin. Fanout-vaihtoa käytetään esimerkiksi chat-palvelujen viestienjaossa. Aiheenvaihtotapauksissa (Topic Exchange) käytetään reititysavainta jonon sääntöjen kanssa viestin sovittamiseen ja lähettämiseen. Otsikkovaihto (Headers Exchange) käyttää reititysavaimien sijaan ylimääräisiä ylätunnisteita viestien yhdistämiseen ja reitittämiseen jonoihin. (Tezer 2013.)

AMQP-liikenteen salaukseen käytetään Transport Layer Security (TLS)- tai Simple Authentication and Security Layer (SASL) -menetelmää tai niitä molempia samanaikaisesti. Molemmat menetelmät ovat internetliikenteen salaukseen tarkoitettuja salausprotokollia.

4.4 Konttitekнологia

Ennen vanhaan kuorma-auto ajoi tehtaalle, jossa tuotteet lastattiin auton lavalle. Auto ajoi satamaan, jossa tuotteet siirrettiin lavalta laivaan. Laiva seilasi toiselle mantereelle, jossa tuotteet siirrettiin laivasta taas kuorma-auton lavalle. Auto ajoi varaston pihaan, jossa kuorma purettiin varastohallin hyllyille. Sitten keksittiin rahtikontti. Kontti lastattiin täyteen tuotteita tehtaalla, jonka jälkeen koko kontti kuljetettiin rekoilla ja laivalla samaa reittiä pitkin varastolle, jossa se purettiin. Ylimääräisiltä purkutöiltä satamassa vältyttiin, kun valmiiksi täytetty kontti pystyttiin viemään sellaisenaan alusta loppuun. (Kotilainen 2017.)

Rahtikontteihin perustuva teknologia on siirtynyt myös tietotekniikan pariin. Container eli kontti on keino paketoita yhteen sovellus ja kaikki sen tarvitsemat ohjelmakirjastot ja taustajärjestelmät. Yhteisen alustan päällä ajettavat itsenäiset kontit toimivat eristettyinä alustasta ja muista konteista. Eristämisen vuoksi konteissa toimivat sovellukset eivät sekoita toisiaan eikä yhteensopivuusongelmia synny. Sovellusten käyttö konttien avulla on nopeaa ja tehokasta. Kontit ovat myös helposti siirrettävissä. Tietokoneelle paikallisesti asennettu sovellus tai virtuaalisoitu sovellus ovat huomattavasti raskaampia kokonaisuuksia, koska sovelluksen lisäksi koneissa pyörii muitakin ohjelmia, jotka syövät resursseja. Sen sijaan kontissa on vain se yksi haluttu sovellus, eikä mitään ylimääräistä. (Kotilainen 2017.)

Tällä hetkellä konttitekнологian tunnetuin toimija on Docker. Docker-ohjelmisto on työkalu, jolla kontteja voidaan muokata. Monista pilvialustoista, kuten Microsoft Azuresta löytyy omia työkaluja Docker-konttien hallintaan. Vaikka konttitekнологia ei olekaan yksinomaan IoT:hen liittyvä teknologia, voidaan sitä hyödyntää tehokkaasti erilaisissa IoT-ratkaisuissa. (Kotilainen 2017.)

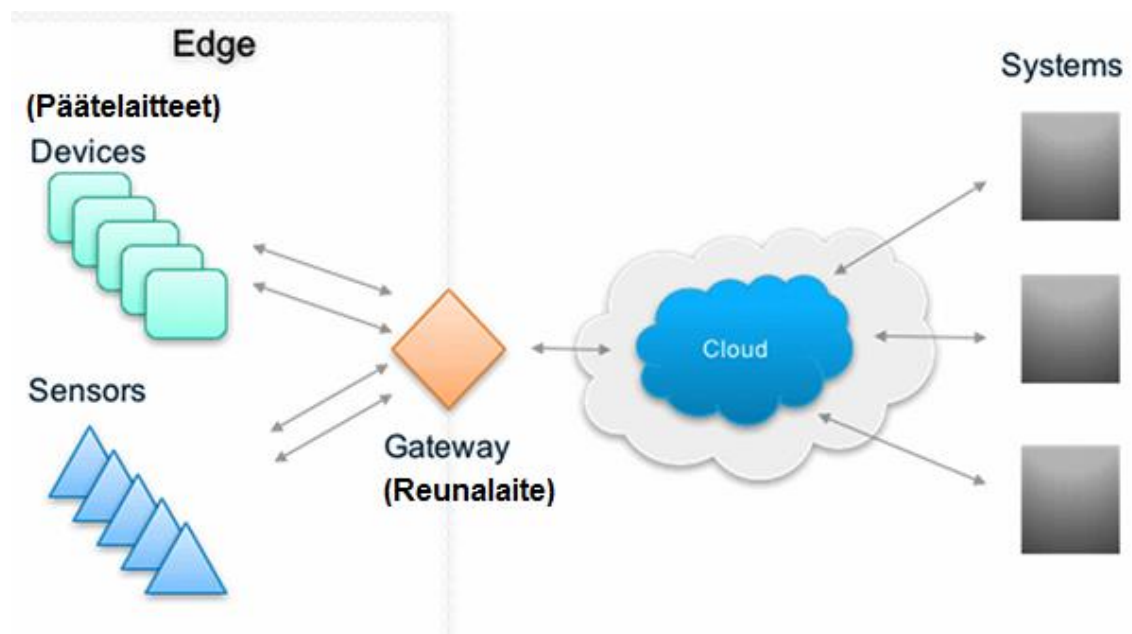
5 IoT-Gateway

5.1 Reunalaite

IoT-gateway eli IoT-yhdyskäytävä on verkkolaite, joka hoitaa tiedonsiirron kentällä olevien IoT-päätelaitteiden ja pilvipalvelun välillä. Monesti puhutaan myös reunalaitteesta.

Reunalaite sijaitsee nimensä mukaisesti järjestelmän ”reunalla” ja kerää kokoon päätelaitteiden, kuten anturien datan, muuntaa datan protokollat sopiviksi ja prosessoi anturidatan ennen lähettämistä. Reunalaitteen sisällä tapahtuva datan prosessointi tunnetaan termillä ”edge computing”, kömpelösti suomennettuna reunaprosessointi. Reunaprosessin ansiosta IoT-järjestelmän skaalautuvuus on parempi, koska IoT-päätelaitteita ei tarvitse hallita yksitellen. Reunalaitteen avulla voidaan hallita isoakin IoT-laitemassaa. Päätelaitteiden ei tarvitse olla suorituskyvyltään tehokkaita, koska reunalaite prosessoi ja lähettää datan päätelaitteiden puolesta. Näin saavutetaan taloudellisia säästöjä, kun anturilaitteet voivat olla yksinkertaisia ja edullisia. (Desai 2016.)

IoT-laitteiden ja -sovellusten räjähdysmäinen kehitys on johtanut laajojen IoT-järjestelmien yleistymiseen. Reunalaitteiden implementoiminen helpottaa monimutkaisten järjestelmien hallittavuutta ja tuo kokonaisuuteen yhden tietoturvakerroksen lisää. Anturien ja pilven yhdistämisen lisäksi reunalaite tarjoaa tallennustilaa sekä paikallista suoritus-
tehoa. Osa datasta voidaan prosessoida jo reunalaitteessa, jolloin reunalaitteen vasteaika ja reagoivuus paranevat. Reunalaite voi siis suodattaa rutiinomaiset ilmoitukset pois ja keskittyä välittämään vain sellaiset viestit, jotka vaativat tarkempaa huomiota. (Desai 2016.)



Kuva 3. IoT-yhdyskäytävän sijainti IoT-järjestelmässä (Desai 2016).

IoT-yhdyskäytävälaitteiden käyttöjärjestelmäksi on Fingridillä valittu Canonical Ltd:n kehittämä avoimen lähdekoodin Ubuntu. Se pohjautuu Linux-jakeluun kuuluvaan Debianiin, joka on Unix-tyyppinen käyttöjärjestelmä. Fingridin tapauksessa sähköasemalle sijoitetut reunalaitteet keräävät niihin yhdistettyjen anturien keräämän datan kokoon ja lähettävät sen edelleen pilveen.

Fingridillä on yli sata sähköasemaa ja monilla asemista on useampia valvontaa vaativia kohteita, kuten laitetiloja, muuntajabunkkereita ja kytkinkenttiä. Näin ollen yhtä sähköasemaa kohti tarvitaan aseman suuruudesta ja valvontaa vaativien kohteiden määrästä riippuen jopa useita kymmeniä reunalaitteita. Näitä laitteita on pystyttävä hallitsemaan etänä. Reunalaitteelle on voitava asentaa etätoiminnolla järjestelmä-, ohjelmisto- sekä tietoturvapäivityksiä. Lisäksi laitteiden statusta sekä mahdollisia virheilmoituksia on pystyttävä seuraamaan. Tällaisia sovelluksia löytyy runsaasti, mutta Ubuntu-käyttöjärjestelmälle sopivia löytyy vain kourallinen. Tutkittaviksi valittiin Ubuntu Landscape sekä VMware Pulse.

5.2 Ubuntu Landscape

Landscape on Ubuntu:n kehittäjän Canonical Ltd:n kehittämä järjestelmänhallintatyökalu. Landscapella pystytään hallitsemaan työasemia, palvelimia sekä pilvisijoituksia selainpohjaisen käyttöliittymän kautta. Yhdellä Landscape-istunnolla voidaan hallita jopa 40 000 Ubuntu-konetta. Työkalulla tehdyt komennot asettuvat Landscape-palvelimella jonoon, josta hallittavat laitteet eli Landscape-agentit käyvät poimimassa ne suoritettaviksi. (Schnober 2014.)

Kustomoitujen laitetyyppi-profiilien, käyttäjien ja käyttäjäryhmien luominen on Landscape:ssä mahdollista. Yhdestä näkymästä voidaan asentaa päivityksiä, lisätä tai poistaa sovelluksia ja toteuttaa monia muitakin järjestelmäylläpitäjän toimintoja. Myös koneita, jotka on liitetty Landscape-hallintaan, mutta ovat pois päältä, on mahdollista päivittää. Tällöin päivitys suoritetaan heti koneen käynnistyttyä. Landscape mahdollistaa myös erilaisten käytäntöjen määrittämisen automaattisten järjestelmä- ja tietoturvapäivitysten asentamista varten. (Schnober 2014.)

Agentteihin voidaan ajaa Landscapesta ohjelmakoodeja eli skriptejä, jos skriptien suorittaminen sallittiin agentin rekisteröintivaiheessa. Samat käskyt, jotka syötetään laitteen

paikalliseen komentoriiviin, voidaan lähettää Landscapen kautta skriptien avulla. Kerta-käyttöisten skriptien lisäksi työkalusta löytyy varasto, johon skriptejä voidaan tallentaa myöhempää käyttöä varten. (Schnober 2014.)

Landscapen avulla voidaan nähdä palvelimeen liitettyjen Ubuntu-koneiden tekniset ominaisuudet. Työkalu tarjoaa myös visuaalisen näkymän, josta voidaan seurata graafisesti laitteiden lämpötilaa, levyn- ja muistinkäyttöä sekä prosessorin kuormitusta. Virhetilanteita, kuten liian korkeaa lämpötilaa tai puuttuvaa tietoturvapäivitystä varten käyttäjä pystyy luomaan hälytyksiä, jotka kulkeutuvat järjestelmän ylläpitäjälle. Ennen kaikkea Landscapella pystytään automatisoimaan yksinkertaisia päivittäisiä töitä. (Schnober 2014.)

Tietoturvan kannalta on oleellista, että Landscape-palvelimeen saa otettua yhteyden vain Landscape-agentilta eli siltä Ubuntu-koneelta, jota hallitaan. Tässä hyödynnetään Role-based Access Control (RBAC) -suojausta. RBAC-suojaus perustuu rooleihin ja pääsy dataan sallitaan vain niistä lähteistä, jotka on vahvistettu sallituiksi rooleiksi. Agentin saa yhdistettyä palvelimeen syöttämällä Landscape-palvelimen käyttäjänimen ja palvelimelle annetun rekisteröintiavaimen. Tämän jälkeen agentin pyytämä yhteys on sallittava palvelimen päässä. Agentilta viestit lähtevät palvelimelle suojaamattomassa Hypertext Transfer Protocol (HTTP) -muodossa. HTTP on verkkoselainten ja palvelinten käytämä tiedonsiirtoprotokolla. Palvelin taas puolestaan lähettää agentille viestejä suojaustussa Hypertext Transfer Protocol Secure (HTTPS) -tiedonsiirtoprotokollaformaattissa. (Schnober 2014.)

Canonical tarjoaa Landscape-palveluaan moneen eri tarpeeseen. Ilmaiseksi sovellus on saatavilla On-premises-versiona kymmeneen laitteeseen. Tämän lisäksi erihintaisia palvelupaketteja on saatavilla järjestelmän tyypistä ja käyttötarpeesta riippuen. On kuitenkin tärkeää muistaa, että Landscape soveltuu vain ja ainoastaan Ubuntu-koneiden hallintaan eikä työkalu tue mitään muita käyttöjärjestelmiä. (Schnober 2014.)

5.3 VMware Pulse

VMware Pulse IoT Center on virtualisointiohjelmistoihin keskittyvän VMwaren kehittämä IoT-laitteiden hallintatyökalu. Ubuntu Landscapen tavoin VMware Pulsella pystytään monitoroimaan liitettyjen laitteiden tilaa ja suorittamaan erilaisia päivityksiä etänä. VMware

Pulse on silti Landscapea huomattavasti kompleksisempi, sillä Pulse mahdollistaa reunalaitteiden hallinnan lisäksi myös niihin liitettyjen IoT-päätelaitteiden valvonnan ja hallinnan. Koska tässä opinnäytetyössä syvennytään Microsoft Azuren soveltuvuuteen an-turienhallinnan osalta, Pulsen toiminta keskitetään vain reunalaitteen hallintamahdollisuuteen. (Rubens 2017.)

Huomattavan monipuolisesta VMware Pulsen käyttöliittymästä löytyy kattava valikoima toiminteita, joilla pystytään suorittamaan etähallinnan kannalta oleellisia tehtäviä. Liitettyjen laitteiden päivitystiedot ovat tarkastettavissa siten, että Pulse osaa tarvittaessa ilmoittaa puuttuvista päivityspaketeista. Paketit voidaan tällöin asentaa laitteeseen etätoiminnolla. Pulseen pystytään määrittelemään sääntöjä, jotka lähettävät ylläpitäjälle sähköposti-ilmoituksen, mikäli laitteessa havaitaan jokin ongelma, joka vaatii korjaustoimenpiteitä tai silmälläpitoa. Kuten Landscapea, myös Pulsessa laitteiden lämpötila-, muisti- ja prosessoritietoja voidaan tarkkailla. (Rubens 2017.)

Monipuolisuuden lisäksi Pulse on toiminnaltaan turvallinen, sillä jokaista Pulseen liitettyä laitetta kohtaan generoidaan oma yksilöity käyttäjänimi ja salasana. Turvatoiminaan Pulse käyttää myös Access Control List -pääsyoikeuslistoja ja Hash-based Message Authentication Code -todennusta, jotka takaavat sen, että laitteeseen muodostettu yhteys tulee sovitusta lähteestä. Jos järjestelmä kuitenkin altistuu tietomurrolle, Pulsesta löytyy etätoiminto, jolla kaikki laitteen sisältämä data voidaan hävittää, ettei se joudu väärin käsiin. Kehitteillä on myös metodi, jolla laite voidaan tarvittaessa asettaa karanteenitilaan odottamaan tulevia toimenpiteitä varten. (Rubens 2017.)

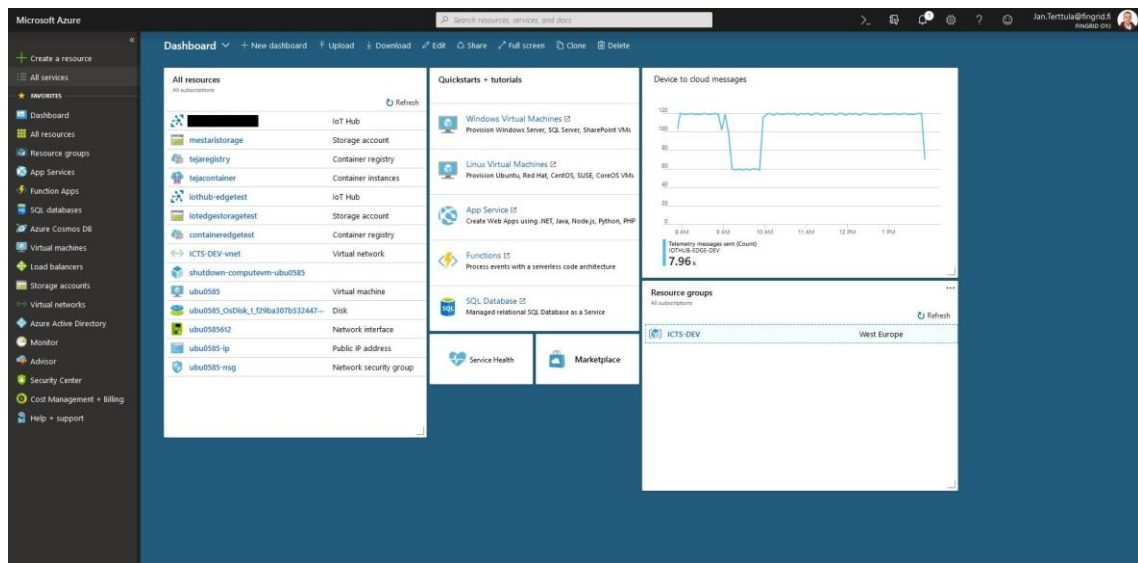
6 Microsoft Azure

Pilvipalvelujen rintamalla Microsoftin kehittämä Azure on yksi markkinoiden johtavista ja tunnetuimmista julkisista pilvipalvelualustoista. Azure julkaistiin helmikuussa 2010 nimellä Windows Azure, mutta tuotteen nimi vaihtui maaliskuussa 2014 Microsoft Azureksi. Azurea voidaan käyttää sekä virtuaalipalvelinten alustana että sovellusalustoja tarjoavana palveluna. Valmiita komponentteja ja palveluja Azuresta löytyy tietotekniikan moniin eri osa-alueisiin liittyen, kuten tekoälyyn, suorituskyvykkääseen tietojenkäsittelyyn, digitaaliseen markkinointiin, peliteknologiaan sekä esineiden internettiin liittyen. (What is Azure?)

Azure tarjoaa käyttäjilleen IoT-palveluja aina yksikertaisista ja suoraviivaisista pienille yrityksille tarkoitetuista ratkaisuista monipuolisiin ja korkeatasoisiin suuria resursseja vaativiin menetelmiin. Tietoturvallisuus ja alustan monipuolisuus ovat pääpiirteitä, joilla Azuren kotisivuilla tuotetta kuvaillaan ja mainostetaan. Vaikka Azure onkin Microsoftin tuote, se on silti yhteensopiva myös monien muiden valmistajien sovellusten ja käyttöjärjestelmien kanssa. (What is Azure?)

6.1 Azure Portal ja Azure CLI

Microsoft Azuren käyttäjille tarkoitettu Azure Portal -työkalu on internetiselaimella käytettävä palvelu. Azure Portalin kautta käyttäjä voi tarkastella ja hallita visuaalisesti kaikkia Azuresta löytyviä resursseja kuten verkkosovelluksia, virtuaalikoneita ja -verkkoja, tietokantoja sekä tallennustiloja. Portaalin oletusnäkyminen on käyttäjän itse muokattavissa sen mukaan, mitä halutaan saada näkyviin ja missä järjestyksessä. (Microsoft Azure Portal.)



Kuva 4. Kuvankaappaus Azure Portal -käyttöliittymästä.

Azure Portalin mukana tulee myös Azuren oma komentorivi, Azure CLI. Komentorivin asetuksista valittavissa on joko Unix-käyttöjärjestelmistä tuttu Bash-ympäristö tai Microsoftin oma PowerShell-ympäristö. Molemmista ympäristöistä löytyy samat toiminnallisuudet, mutta komentojen syntaksi on erilainen. Azure CLI -komentorivin kautta pystytään hallinnoimaan samoja Azure-resursseja kuin Azure Portalin kautta, mutta komentorivillä kaikki data on tekstimuodossa. (Azure, Cloud Shell.)

Azure CLI -komentoriviä voidaan käyttää suoraan Azure Portalin kautta, jolloin siitä käytetään usein nimitystä Azure Cloud Shell. Azure CLI pystytään myös asentamaan paikallisesti tietokoneelle tai ajamaan Docker-kontissa. (Azure, Cloud Shell.)

6.2 Azure IoT Hub

Azure IoT Hub on yksittäinen Azuren palvelu, jolla voidaan yhdistää, monitoroida ja hallita lukuisia, jopa miljoonia IoT-laitteita ja niiden lähettämää telemetriadataa. Palvelu mahdollistaa luotettavan ja tietoturvallisen kaksisuuntaisen yhteyden IoT-laitteeseen. Azure IoT Hub ei yksinään tarjoa End-to-end (E2E) -ratkaisua eli yhteyttä, joka kulkee koko verkon läpi alkupisteestä loppupisteeseen. Sitä voidaan toki käyttää alkupisteenä mille tahansa Azuren IoT-ratkaisulle. (What is Azure IoT Hub? 2018.)

Device twin eli digitaalinen kaksonen on IoT Hubiin tallennettu JavaScript Object Notation (JSON)-tiedosto, joka sisältää liitetyn laitteen tietoja, kuten metadataa, konfiguraatio- sekä tilatietoja. JSON on yksinkertainen ja kevyt tiedonvälityksessä käytettävä helposti ihmisen ja laitteen tulkittavissa oleva tiedostomuoto (Introducing JSON). Koska laitteita voi IoT Hubiin olla kytkettynä lukuisa määrä, jokaista laitetta kohtaan on IoT Hubissa oma digitaalinen kaksoensa. Digitaalinen kaksonen mahdollistaa laitekohtaisen metadatan varastoimisen pilveen, laitteen sovelluksen käytettävissä olevista ominaisuuksista raportoinnin sekä sovelluksen näkymän ja sovelluksen taustajärjestelmän välisten työtoimintojen synkronoinnin. Lisäksi digitaalisen kaksosen avulla laitteelta voidaan tiedustella metadataa, kuntoa ja konfiguraatietietoja. (Understand and use device twins in IoT Hub 2018.)

IoT Hubiin kytkettyjen laitteiden sisälle luotujen moduulien tiedot on varastoitu module twineihin. Module twin on digitaalisen kaksosen tapaan JSON-tiedosto, josta löytyy moduulin metadataa, konfiguraatio- ja tilatietoja. Jokaista IoT Hubiin liitettyä moduulia vastaa oma module twin. Module twin varastoi moduuliin sidonnaista tietoa, jonka IoT Hub saa synkronoitua laitteen ja IoT Hubin välillä. IoT-ratkaisun taustajärjestelmä voi tiedustella myös module twiniltä käynnissä olevia toimintoja. Käytännössä module twin mahdollistaa aivan samat asiat kuin digitaalinen kaksonen, mutta module twin tekee asiat askeleen pienemmällä portaalla, IoT Hubiin kytketyn laitteiden sisällä toimivien moduulien tasolla. (Understand and use module twins in IoT Hub 2018.)

Muita Azuren palveluja yhdistämällä IoT Hubiin voidaan rakentaa tarkasti asiakkaan tarpeita vastaava kustomoitu IoT-kokonaisuus.

6.3 Azure IoT solution accelerators

Azure IoT solution accelerators on Azuren tarjoama Platform as a Service (PaaS) -ratkaisu. PaaS tarkoittaa palveluna tarjottavaa sovellusalustaa, ja se mahdollistaa käyttäjälle monipuoliset työkalut, joilla lopullista IoT-ratkaisua pystytään kehittämään (Eronen 2016). Azure IoT solution accelerators sisältää valmiiksi konfiguroituja toimintoja tavallisiin IoT-tarpeisiin. Nämä toiminnot ovat monipuolisesti muokattavissa vastaamaan asiakkaan tarpeita. PaaS soveltuu yrityksille, joilla on suuri määrä laitteita tai laitetyppejä ja jotka haluavat korkeatasoisen hallinnan IoT-ratkaisuihinsa. (Azure IoT technologies and solutions: PaaS and SaaS 2018.)

Data ja analytiikka sisältyvät Azuren PaaS-pakettiin. Paketti tuo pilviällyn päätelaitteeseen ja varastoi IoT-laitedatan kustannustehokkaasti. Suurien datamäärien visualisointiin on kehitetty Azure Time Series Insight -palvelu, johon data pystytään integroimaan ja joka kuuluu PaaS-pakettiin. (Azure IoT technologies and solutions: PaaS and SaaS 2018.)

6.4 Azure IoT Central

Eryisesti pienemmille yrityksille Azure suosittelee Azure IoT Central -ratkaisua. Azure IoT Central on Software as a Service (SaaS) -palvelu, mikä tarkoittaa palvelua, joka tarjotaan asiakkaalle ohjelmistona. Esimerkiksi monet internetselaimella toimivat sähköpostipalvelut ovat tyypillisesti SaaS-palveluja (Eronen 2016). Azuren SaaS-ratkaisua suositellaan sellaisille yrityksille, joilla ei ole valmiuksia tai resursseja omaan IoT-ratkaisuun tai joilla on vähemmän laitteita tai helpommin ennustettavia tapahtumaketjuja. (Azure IoT technologies and solutions: PaaS and SaaS 2018.)

Taulukko 1. Azuren PaaS- ja SaaS-pakettien vertailutaulukko (Azure IoT technologies and solutions: PaaS and SaaS 2018).

	Azure IoT solution accelerators	Azure IoT Central
Primary usage	To accelerate development of a custom IoT solution that needs maximum flexibility.	To accelerate time to market for straightforward IoT solutions that don't require deep service customization.
Access to underlying PaaS services	You have access to the underlying Azure services to manage them, or replace them as needed.	SaaS. Fully managed solution, the underlying services aren't exposed.
Flexibility	High. The code for the microservices is open source and you can modify it in any way you see fit. Additionally, you can customize the deployment infrastructure.	Medium. You can use the built-in browser-based user experience to customize the solution model and aspects of the UI. The infrastructure is not customizable because the different components are not exposed.
Skill level	Medium-High. You need Java or .NET skills to customize the solution back end. You need JavaScript skills to customize the visualization.	Low. You need modeling skills to customize the solution. No coding skills are required.
Get started experience	Solution accelerators implement common IoT scenarios. Can be deployed in minutes.	Application templates and device templates provide pre-built models. Can be deployed in minutes.
Pricing	You can fine-tune the services to control the cost.	Simple, predictable pricing structure.

Azure IoT Centralin mukana ei tule mitään oheisohjelmia, jotka kuuluvat Azure IoT solution accelerators -ratkaisuun. SaaS-paketin saa otettua käyttöön vaivattomasti, mutta se ei ole yhtä monipuolisesti muokattavissa kuin Azuren tarjoama PaaS-ratkaisu. (Azure IoT technologies and solutions: PaaS and SaaS 2018.)

6.5 Azure IoT Edge

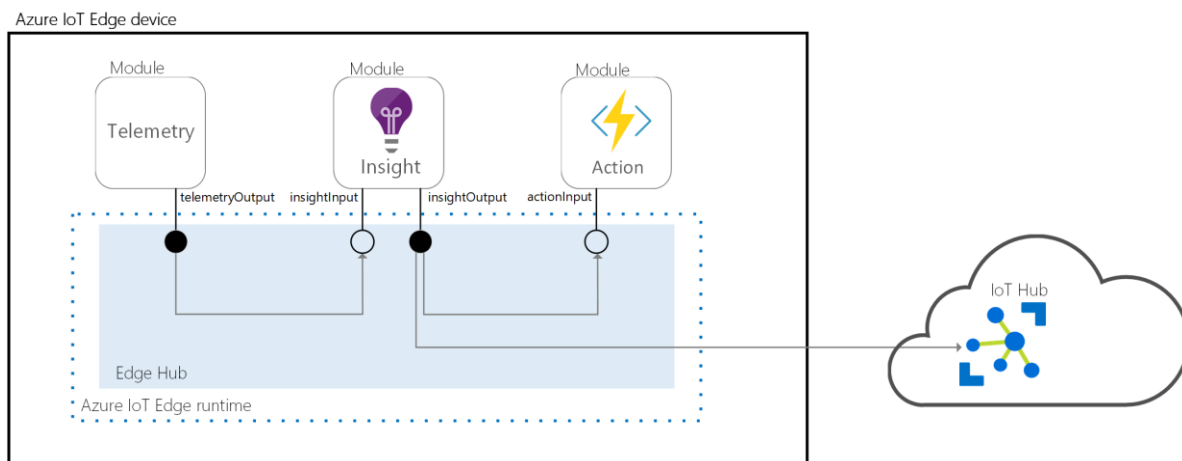
Azure IoT Edge on IoT-palvelu, joka rakentuu Azure IoT Hubin päälle. Palvelu toimittaa pilvi- ja tekoälyn sekä Azure-palvelut suoraan paikallisesti IoT-laitteelle. IoT Edge on tarkoitettu asiakkaille, jotka haluavat analysoida dataa reunalaitteilla eivätkä pilvessä. Datankäsittely reunalaitteella on tehokasta, koska laitteen ei tarvitse käyttää aikaa tai resursseja datan lähettämiseen pilveen. Tällöin laitteelle jää enemmän aikaa reagoida järjestelmässä tapahtuviin muutoksiin. Edge-laite voidaan ohjelmoida toimimaan itsenäisesti tilanteessa, jossa telemetrian lähetys pilveen on estynyt. (What is Azure IoT Edge? 2018.)

IoT Edge -palvelu vaatii järjestelmältä reunalaitteen käyttöä. Fingridin tapauksessa sähköasemalle sijoitettavaksi reunalaitteeksi soveltuu tietotekninen laite, johon voidaan asentaa käyttöjärjestelmä ja johon saa verkkoyhteyden. IoT Edge runtime on kokoelma ohjelmia, joiden on oltava asennettuna reunalaitteelle, jotta reunalaitte voisi toimia IoT

Edge-laitteena. IoT Edge runtimen komponentit mahdollistavat reunalaitteen kyvyn vastaanottaa ajettavaa ohjelmistokoodia sekä kyvyn kommunikoida koodin tuloksista. Runtime vastaa monista tärkeistä ominaisuuksista, kuten laitteen työmäärän päivittämisestä, turvallisuusstandardeista ja pilviyhteyden eheydestä sekä IoT Edge -moduulien keskinäisestä kommunikaatiosta. IoT Edge runtimen tehtävät jakautuvat kahteen kategoriaan, joista runtimen pääkomponentit vastaavat. IoT Edge Hub on vastuussa kommunikoinnista ja IoT Edge Agent hoitaa moduulien hallinnan ja monitoroinnin. Sekä Edge Hub että Edge Agent ovat itsekin IoT Edge -laitteen moduuleja. (Understand the Azure IoT Edge runtime and its architecture 2018.)

Paikallisena välityspalvelimena IoT Hubiin toimiva IoT Edge Hub on toinen IoT Edge runtimen moduuleista ja vastaa kommunikaatiosta. Edge Hubista löytyy tuki AMQP- ja MQTT-protokollille, mutta ei HTTP-protokollalle. Joitakin viestejä Edge Hub delegoi eteenpäin IoT Hubille. Esimerkiksi ensimmäinen Edge Hubille tuleva autentikointipyyntö lähtee IoT Hubille, joka vastaa pyyntöön. Vasta tämän jälkeen kyseisen yhteyden turvallisuustieto varastoidaan paikallisesti myös Edge Hubille tulevia autentikointeja varten. Edge Hub seuraa muodostettujen yhteyksien tilaa ja optimoi liikenteen keräämällä loogiset yhteydet yhteen fyysiseen yhteyteen, joka suuntautuu pilveen. Jos Edge Hub menettää yhteyden IoT Hubiin, se varastoi kerätyn datan ja lähettää sen IoT Hubille yhteyden palatessa. (Understand the Azure IoT Edge runtime and its architecture 2018.)

IoT Edge Hub johtaa myös moduulien välistä viestintää. Moduulit säilyvät erillisinä toisista moduuleista, koska kaikki tiedonvaihto kulkee Edge Hubin välityksellä. Moduuleille tarvitsee vain määritellä sisään-tulo, josta ne vastaanottavat viestejä sekä ulostulo, jonka kautta ne kirjoittavat viestejä. Sisään- ja ulostulopisteet voidaan kytkeä yhteen, jolloin muodostuu ketju, joka tuottaa dataa halutussa järjestyksessä. Ylläpitäjän on myös määriteltävä säännöt, joita Edge Hub noudattaa moduulien välisten viestien siirtelyssä. Reititysviestit määritellään pilvessä ja ohjataan sieltä Edge Hubille. (Understand the Azure IoT Edge runtime and its architecture 2018.)



Kuva 5. Moduuleille määriteltyjen sisään- ja ulostulojen muodostama ketju. Kaikki data kulkee IoT Edge Hubin kautta. (Understand the Azure IoT Edge runtime and its architecture 2018.)

IoT Edge runtimen toinen moduuli, IoT Edge Agent, huolehtii kaikesta moduulien toiminnasta, paitsi niiden välisestä kommunikaatiosta. Agentti huolehtii, että moduuli pysyy aktiivisena ja raportoi moduulin tilaa IoT Hubille. Käynnistyessään Agent käy IoT Hubilta hakemassa luettelon moduuleista, jotka pitää käynnistää. Luettelosta Edge Agent selvittää myös moduulien tarkempia tietoja, joita noudattamalla moduuleja ohjataan. Luettelosta käy ilmi muun muassa moduulin käynnistysvaiheessa tarvittavat kirjautumistiedot konttirekisteriin, josta kerrotaan tarkemmin seuraavassa kappaleessa. Edge Agentilla on tärkeä rooli IoT Edge -laitteen tietoturva-asioiden parissa. (Understand the Azure IoT Edge runtime and its architecture 2018.)

6.6 Azure Container Registry

Sovellus sekä sen taustajärjestelmä voidaan pakata virtuaaliseen pakettiin eli containeriin, mutta suomalaisittain puhutaan kontista. Microsoftin kehittämä Azure Container Registry (ACR) on palvelu, jolla voidaan varastoida ja hallita kontteja. Käyttäjä luo konttirekisterin avulla uusia tai muokkaa jo olemassa olevia kontteja. IoT-järjestelmän IoT Edge Agent voidaan asettaa hakemaan automaattisesti konttirekisteristä uudet tai muutuneet kontit ja ajamaan niiden sisältämät komennot. Azure IoT Edge -laitteelle asennetaan konttien suorittamiseen tarkoitettu konttityökalu, joka hakee ACR:stä kontteja. On suositeltavaa käyttää Moby-projektiin pohjautuvia työkaluja ja kirjastoja. Moby on kontti-kehitysympäristö, joka toimii Docker-ohjelmiston päällä tarjoten tuen useimmille eri

Docker-versioille ja -alustoille. Moby on ainoa työkalu, joka on virallisesti tuettu IoT Edgessä. Dockerin CE/EE -konttilevykuvat sopivat ajettavaksi Mobyn kanssa. (Introduction to private Docker container registries in Azure 2018.)

Esimerkkitapauksessa konttirekisterissä on kontti, joka sisältää toiminnallisuuden, jolla IoT-reunalaite kerää anturilla mitattua ilman kosteuspitoisuusdataa minuutin välein ja lähettää datan Azuren pilveen. Järjestelmän ylläpitäjän mielestä riittäisi, että kosteuspitoisuutta mitattaisiin minuutin sijasta viiden minuutin välein. Ylläpitäjä muuttaa konttirekisterissä olevan kontin koodia tarpeidensa mukaan. Reunalaite havaitsee rekisterissä kontin, johon on tehty muutoksia ja osaa hakea tämän kontin ajettavaksi.

6.7 Azure Stream Analytics

Azure Stream Analytics (ASA) on tapahtumaprosessointimoottorina toimiva palvelu, joka pystyy analysoimaan suuria määriä reaaliaikaista datavirtaa. Sisään tuleva data voi tulla laitteilta, antureilta, verkkosivuilta, sosiaalisesta mediasta tai sovelluksilta. ASA tukee datavirtojen purkamista, kaavojen tunnistamista ja yhteyksiä. Datavirran arvojen perusteella voidaan luoda toiminteita kuten hälytyksiä, jotka lähetetään raportointityökalulle tai varastoidaan myöhempää käyttöä varten. ASA:n avulla voidaan suodattaa laitteelta pilveen lähetettävä niin sanottu turha data pois, jolloin jäljelle lähetettäväksi jää vain tarpeellinen tärkeä data. ASA siis toimii eräänlaisena edge computing -työkaluna. (What is Stream Analytics? 2018.)

ASA on täysin hallinnoitu palvelimeton PaaS-ratkaisu, jolla voidaan toteuttaa datan analytiikkaa. Sen käyttö ei vaadi erillistä laitteistoa tai klusteria, vaan se toimii itsenäisesti Azuren pilvessä. Kielenään ASA käyttää SQL:n kaltaista kyselykieltä, jolla suodatukset, lajittelut, ryhmittämiset ja liittämiset suoritetaan. ASA kykenee tulkitsemaan miljoonia tapahtumia sekunnissa ja lupaa analysoimilleen tapahtumille 99,9 prosentin toimivuuden. (What is Stream Analytics? 2018.)

6.8 Azure Blob Storage

Anturien järjestelmästä keräämän datan varastointiin Azuressa on tarjolla useampia vaihtoehtoja aina tietokannoista erilaisiin datavarastoihin. Azure Blob Storage on raken-

teeltaan epäsäännöllisen datan varastointiin tarkoitettu palvelu. Laajalti skaalautuvaan Blob Storageen on mahdollista tallettaa jopa miljardeja tiedostoja tyypistä riippumatta; kuvia, videoita, äänitiedostoja, dokumentteja, tekstitiedostoja tai ihan mitä tahansa. Talletuskerroksia on useita riippuen siitä, kuinka usein talletettuun dataan on tarvetta päästä käsiksi. Blob Storage on erityisen kustannustehokas, kun varastoidaan massiivisia määriä dataa. (Introduction to object storage in Azure 2018.)

7 Testiratkaisu

Tässä luvussa esitellään Fingridille Azuren avulla testikäyttöön toteutettu IoT-anturijärjestelmä aina laitetasolta erinäisten palvelujen ja sovellusten kautta visualisoinnin tasolle. Luvussa myös esitellään järjestelmän toimintaperiaatteet ja vertaillaan tarvittaessa eri ratkaisuvaihtoehtoja toisiinsa.

7.1 Fyysiset reunalaitteet

Testissä käytettiin IoT-reunalaitteina kolmea erilaista rugged-computeria. Rugged-computer on tietokone, joka on suunniteltu toimimaan luotettavasti erilaisissa, usein jopa haastavissa olosuhteissa. Lisäksi käytössä oli yksi Raspberry Pi 3, joka on yhden piirilevyn tietokone. Kaikki laitteet sijaitsivat testin aikana Fingridin toimistorakennuksessa. Verkkoyhteys reunalaitteille saatiin toimistoverkosta erillään olevan WLAN-reitittimen kautta.

Ensimmäinen laite oli Hewlett Packard Enterprisen kehittämä HPE Edgeline EL20. EL20:n mitat ovat noin 24 x 13 x 7 cm. Laitteessa on Intelin kaksisytiminen i5 4300 1.90Ghz -suoritin ja näytönohjaimena Intelin Haswell-ULT Integrated Graphics Controller. EL20 sisältää 8 gigatavua (DDR3 1600 MHz) keskusmuistia ja 64 gigatavun nopea-toimisen Solid-state drive (SSD) -levyn. Laitteessa on muun muassa useita USB-portteja, VGA- ja HDMI-portit sekä kaksi Ethernet-porttia. Lisäksi laitteessa on langattoman internet-yhteyden mahdollistavat antennit.

Toinen testissä käytetty reunalaite oli Dellin valmistama Embedded Box PC 3000. Dellin laite on lähes samaa koko- ja painoluokkaa HPE:n laitteen kanssa. Suorittimena Dellin laitteessa on Intelin kaksisytiminen Atom E3825 1.33GHz. Näytönohjain on Intelin Atom Processor Z36xxx/Z37xxx Series Graphics & Display. Laitteessa on 4 gigatavua (DDR3

1600 MHz) keskusmuistia. Kovalevytilaa Dellin laitteessa on 500 gigatavua. Testiratkaisun aikana käytettiin USB-, VGA- ja Ethernet-portteja, mutta näiden lisäksi laitteessa on myös muitakin portteja. Laite kytkettiin verkkoon Ethernet-kaapelilla.

Kolmas laitteista oli aiempia huomattavasti pienempi HPE Edgeline EL10. Siinä on Intelin Atom E3826 1.46GHz -suoritin, Intelin Atom Processor Z36xxx/Z37xxx Series Graphics & Display -näytönohjain, 32 gigatavun SSD-levy, 4 gigatavua keskusmuistia sekä WLAN-antennit. EL10:ssä on VGA-, HDMI- ja Ethernet-portit sekä kaksi USB-porttia. Viimeinen ja kaikkein pienin laitteista oli Raspberry Pi 3. Raspberryssä on neliytiminen BCM2837 1.2Ghz -suoritin, Broadcom VideoCore IV -näytönohjain, yksi gigatavu keskusmuistia sekä WLAN-mahdollisuus. Raspberryssä on neljä USB-porttia, HDMI-portti sekä Ethernet-portti. Raspberryssä käytettiin 16 gigatavun microSD-muistikorttia. Molemmat näistä laitteista yhdistettiin langattomasti testikäytössä olleeseen reitittimeen.

Kaikille yllämainituille laitteille asennettiin Ubuntu 16.04 LTS Server -käyttöjärjestelmä. Asennuksissa käytettiin keskenään tismalleen samoja asetuksia. Käyttöjärjestelmä ei sisältänyt graafista käyttöliittymää, joten tämä oli ladattava ja asennettava erikseen. Asennuksen jälkeen vältettiin asentamasta mitään järjestelmä- ja tietoturvapäivityksiä, koska niiden osalta haluttiin testata Landscape-järjestelmänhallintatyökalua.

7.2 Ubuntu-hallinta

7.2.1 Landscape SaaS

Kun Ubuntu oli asentunut graafista käyttöliittymää myöten, kaikille laitteille asennettiin Landscape-client -ohjelma. Tämän jälkeen Landscapen verkkosivuilla luotiin käyttäjätili. Landscapen selainpohjaisessa käyttöliittymässä tilille konfiguroitiin rekisteröintiavain. Tilille satunnaisgeneroitua tunnusta ja rekisteröintiavainta käyttämällä reunalaitteet saatiin niiden komentoriveille syötetyllä komennolla yhdistettyä juuri luotuun Landscape-tiliin. Komento sisälsi yhdistettävän laitteen nimen, Landscape-palvelimen käyttäjätunnuksen sekä rekisteröintiavaimen. Hetken aikaa reunalaitteet keräsivät tietoja itsestään ja asennetuista paketeista, jonka jälkeen Landscapen käyttöliittymään ilmaantui ilmoitus laitteista, jotka halusivat muodostaa yhteyden. Yhteydenotot hyväksyttiin ja laitteet ilmaantuivat järjestelmänhallintatyökalun näkymään. Samalla yhdistetyt laitteet eli agentit kertoivat keräämänsä tiedot palvelimelle. Landscapessa luotiin vielä ryhmät fyysisille ja virtuaalisille laitteille. Rekisteröidyt laitteet siirrettiin fyysisten laitteiden ryhmään.

Pian laitteiden rekisteröinnin jälkeen Landscapen käyttöliittymä huomautti asentamattomista päivityksistä. Päivitysten sisältämien pakettien tiedot olivat nähtävillä, joten voitiin helposti päättää, mitkä päivityksistä olivat tarpeellisia. Kaikille agenteille asennettiin Landscapen kautta kaikki saatavilla olleet päivitykset.

Microsoft Azuren käyttöliittymässä luotiin vielä kaksi virtuaalista 16.04 LTS Ubuntu-palvelinta. Näillekin palvelimille asennettiin Landscape-client-ohjelma ja ne yhdistettiin samaan Landscapen SaaS-tiliin kuin fyysiset reunalaitteet. Virtuaalipalvelimet sijoitettiin virtuaalisten laitteiden ryhmään.

7.2.2 Landscape On-premises

Toinen vartenotettava vaihtoehto reunalaitteiden hallitsemiseen on Canonicalin Landscape On-premises (On-prem). Ratkaisu perustuu siihen, että omalle Ubuntu-laitteelle, yrityskäytössä omaan konesaliin, asennetaan Landscape-palvelinsovellus, johon liitetään Landscape-agenteina toimivia reunalaitteita. Tämä ratkaisu on vaativampi toteuttaa ja vaatii käyttäjältä palvelimen päivittämistä aina silloin, kun uusi versio julkaistaan. Palvelimen päivittäminen aina uusimpaan versioon parantaa sovelluksen tietoturvaa sekä toimintavarmuutta.

On-premin käyttöönotto aloitettiin lataamalla komentorivin kautta yhdelle reunalaitteista Landscape-palvelimen asennuksessa tarvittava tiedostopaketti. Testissä käytettiin 17.03 version Landscape-palvelinta, koska se oli yhteensopiva reunalaitteiden Ubuntu Server 16.04 -käyttöjärjestelmän kanssa. Komentoriviltä reunalaitteelle asennettiin Landscape-server-quickstart-sovellus. Tämän jälkeen laitteen verkkoselaimen, Mozilla Firefoxin asetuksista sallittiin selaimen käyttää Landscape-palvelimen SSL-sertifikaattia. Sertifikaatti tarvittiin, jotta laite sai muodostettua suojatun verkkoyhteyden selaimen kautta Landscapen käyttöliittymään. Laite täytyi käynnistää uudelleen, jotta sertifikaattimuutos astui voimaan.

Reunalaitteen käynnistyttyä voitiin Firefoxilla kirjautua Landscapen käyttöliittymään osoitteessa [https://\[hostname\]](https://[hostname]). Testissä Landscape-server-quickstart asennettiin laitteelle nimeltä ubuntuHP, joten käyttöliittymä avattiin selaimella osoitteesta <https://ubuntuHP>. Ensiksi piti luoda käyttäjätunnus ja tällä tunnuksella kirjaututtiin Landscape-palvelimelle sisään. Kirjautumisen onnistuttua palvelimelle asetettiin rekisteröintiavain, jonka avulla palvelimeen saataisiin liitettyä hallittavia Ubuntu-laitteita.

Kaksi Landscape SaaS:iin yhdistettyä laitetta haluttiin siirtää Landscape On-premin hallittavaksi. Näille agenteille oli jo asennettu Landscape-client -ohjelma. Saman sertifiointin, jota käytettiin aiemmin palvelimen verkkoyhteyden salaamiseen, vastakappaleena toimiva SSL-avaintiedosto oli kopioitava palvelimelta agenteille. Agenttilaitteilla avaintiedosto tallennettiin Landscapen asennuskansioon. Lisäksi agentin Landscapen konfiguraatitiedostoon oli tehtävä lisäys, joka määrittelee käytettävän SSL-avaimen. Agenttilaitteen hosts-listalle oli määriteltävä käytettävän hallintapalvelimen paikallinen host-name ja sen IP-osoite, jotta yhteys palvelimelle saatiin muodostettua. Hosts-tiedoston muokkaamisen vuoksi testiratkaisussa ei tarvinnut lainkaan suunnitella nimipalvelinasetuksia.

Asetusten jälkeen agentin komentoriville syötetyllä komennolla voitiin lähettää yhdistyspyyntö On-prem-palvelimelle. Komento sisälsi hallittavan laitteen nimen, On-prem-käyttäjätunnuksen, rekisteröintiavaimen, viestijärjestelmän verkko-osoitteen ja ping-osoitteen. Komennon syöttämisen jälkeen palvelin havaitsi pyynnön, ja se hyväksyttiin käyttöliittymästä, jolloin yksi laite ilmaantui käyttöliittymän nähtäville.

Myös On-prem-ratkaisussa luotiin kaksi laiteryhmää ja hallittavat kaksi laitetta sijoitettiin eri ryhmiin. Kaikki saatavilla olleet Landscapen ehdottamat päivitykset asennettiin hallittaville laitteille.

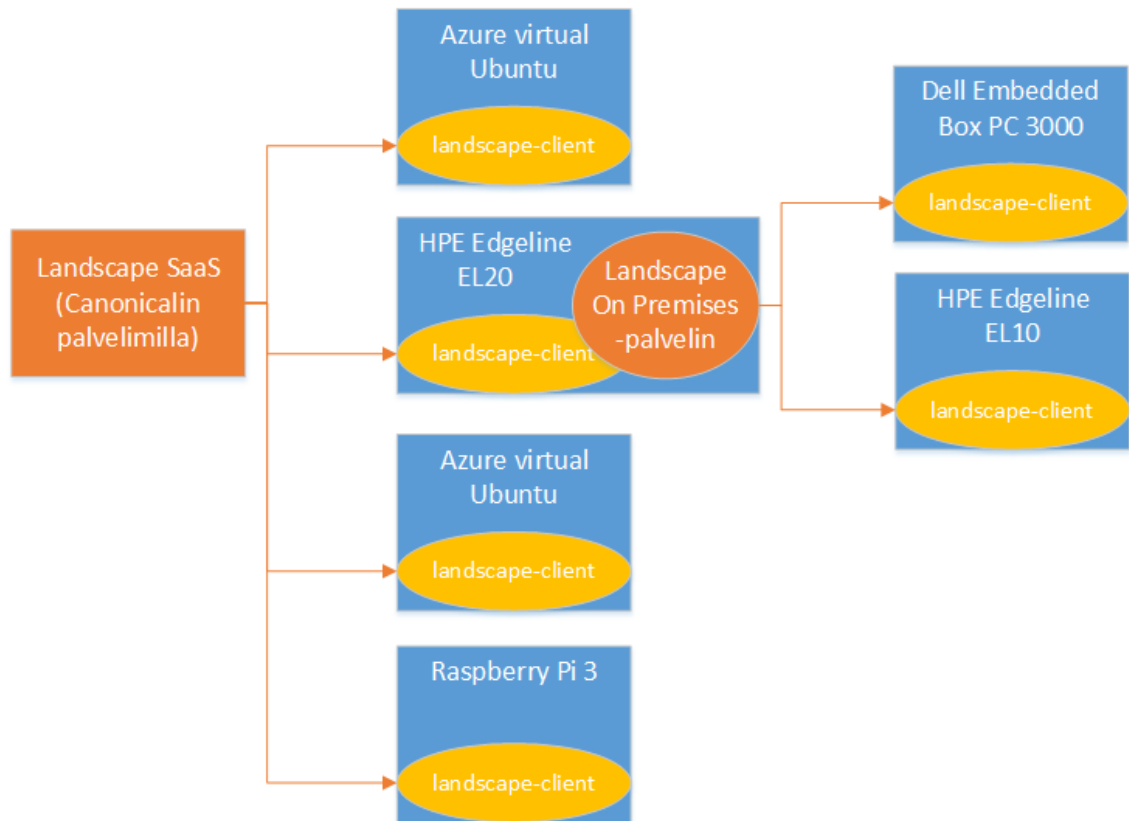
Havaittiin, että osa järjestelmän toiminnallisuuksista päättyi toistuvasti erilaisiin virheilmoituksiin. Vianetsinnän yhteydessä huomattiin, että On-prem-palvelimen asennusvaiheessa käytetty nimi *ubuntuHP* ei isojen kirjaimiensa vuoksi ollut yhteensopiva Apache-HTTP-palvelinohjelman kanssa. Apachen konfiguraatitiedostoa oli muokattava siten, että kaikki laitteen hostnimestä esiintyvät isot kirjaimet oli korvattava pienillä kirjaimilla. Tämän korjauksen jälkeen virheiden ilmaantuminen lakkasi.

7.2.3 VMware Pulse

VMware Pulse osoittautui haasteellisesti käyttöönotettavaksi hallintatyökaluksi. Pulse on kaikin puolin monimutkaisempi järjestelmä eikä siitä löydy ilmaisjakelussa riittävästi sopivaa materiaalia, jota olisi voitu hyödyntää testiratkaisun rakentamisessa. VMware Pulse IoT Centerin voidaan myös olettaa olevan Landscapea kalliimpi, koska Pulse on ominaisuuksiltaan kyvykkäämpi. Tarkkaa hintatietoa ei kuitenkaan selvitetty.

7.2.4 Landscapen toiminta

Landscape SaaS- ja Landscape On-premises-järjestelmänhallintaratkaisujen yhdistäminen johti lopulta siihen, että SaaS-palvelimeen oli liitetty hallittaviksi agenteiksi kaksi fyysistä ja kaksi virtuaalista laitetta. Toinen fyysisistä laitteista toimi On-premises-palvelimena ja siihen oli liitetty kaksi fyysistä laitetta.



Kuva 6. Landscape-järjestelmänhallintatyökalulla rakennettu Ubuntu-laitteiden testiympäristö.

Neljää Canonicalin SaaS-palvelimeen liitettyä laitetta pystyttiin hallitsemaan verkkoselaimella miltä vain tietokoneelta, jolla oli yhteys internettiin. Kahta On-premises-palvelimeen liitettyä laitetta voitiin sen sijaan hallita ainoastaan siltä tietokoneelta, jolle kyseinen On-premises-palvelin oli asennettu.

Landscape-agentit lähettävät säännöllisesti tiedusteluviestejä palvelimelle. Jos reuna-laitte kadottaa verkkoyhteyden tai jokin vika estää laitetta lähettämästä Landscapeen tiedusteluviestejä, Landscapen käyttöliittymä huomauttaa viiden minuutin kuluttua, että yhteys johonkin laitteeseen on menetetty. Kun yhteys muodostetaan takaisin, kestää yleensä viidestä kymmeneen minuuttia, että huomautus häviää.

Päivitysten osalta Landscape toimi tehokkaasti. Kun Landscapesta annettiin käsky päivityspakettien asennukselle, paketti asetettiin palvelimella työjonoon. Jonossa paketti viipy yleensä vain minuutin tai kaksi, kunnes agentti nouti paketin, joka koostaan riippuen asentui yleensä muutamassa minuutissa. Joidenkin päivitysten jälkeen laite oli käynnistettävä uudelleen. Uudelleenkäynnistyskomento asetettiin myös jonoon, josta noin minuutin kuluessa agentti haki komennon suoritettavaksi. Laitteen käynnistys kesti viidestä kuuteen minuuttia. Työjonon odotusajan oletusarvo on enimmillään viisi minuuttia, mutta harva paketti viipy jonossa niin kauaa (Dimotakis 2018). Lähes kaikki paketit noudettiin asennettaviksi parissa minuutissa. Landscape myös ilmoitti käyttäjälle, onnistuiko vai epäonnistuiko päivitysten asennus tai uudelleenkäynnistys. Niissä harvoissa tapauksissa, jolloin päivityksen asentaminen oli epäonnistunut, sen asentaminen onnistui aina toisella yrityksellä.

Päivitysten asentumisen onnistuminen varmistettiin tarkastelemalla paikallisesti laitteiden dpkg-lokitiedostoja. Lokitiedostoista pystyttiin näkemään onnistuneesti asentuneiden päivityspakettien nimi, versio, sisältö ja aikaleima. Lokitiedoista nähtiin myös virheilmoitukset, mikäli päivitysten asentaminen oli jostain syystä epäonnistunut. Dpkg-loki saatiin myös luettua Landscapesta ajatun skriptin avulla. Molemmissa ratkaisuisa skriptit toimivat muilta osin hyvin, mutta root-komentoja ei pystytty kummassakaan ajamaan. Oletusasetuksilla super user -komentoja eikä pääkäyttäjänä ajettavia skriptejä pystytä suorittamaan. Niiden suorittaminen on kuitenkin mahdollista, mutta se vaatisi asiantuntijan opastusta ja luultavasti maksullisen Landscape-tilauksen, jota ei tässä insinööri-työssä lähdetty tekemään. (Dimotakis 2018.)

Landscape mahdollistaa päivitysprofiilien luomisen, joilla voidaan ajastaa automaattinen päivitysten asentaminen. Molemmissa Landscape-ratkaisuissa luotiin päivitysprofiili, jonka määriteltiin asentamaan kaikki saatavilla olevat päivitykset aina lauantai-iltaisain sekä kaikki tietoturvapäivitykset keskiviikkoiltaisain. Päivitysten automatisointi toimi jokseenkin, kuten profiilissa oli määritelty. Asennuksen alkaminen käynnistyi profiiliin asetetusta kellonajasta muutaman minuutin päästä ja oli suoritettu onnistuneesti noin viidessä minuutissa.

Hallittavia laitteita varten Landscapessa pystytään luomaan uusia ryhmiä, joihin käyttäjä voi sijoittaa haluamansa laitteet. Tiliin on myös mahdollista lisätä uusia käyttäjiä. Testissä laitteet ryhmiteltiin fyysisten ja virtuaalisten laitteiden ryhmiin. Ryhmittely voidaan toteuttaa myös esimerkiksi laitevalmistajan tai maantieteellisen sijainnin mukaan. Käyttäjätiliin

oikeudet voidaan rajata tiettyyn ryhmään, esimerkkitapauksessa Tampereella työskentelevä käyttäjä olisi oikeutettu hallitsemaan vain Pirkanmaan laiteryhmässä olevia laitteita. Ryhmien lisäksi agenteille voidaan lisätä eräänlaisia leimoja eli tägejä. Tägien avulla voidaan luoda eri ryhmien agenteista sekalaisia ryhmiä. Sekä käyttäjä- että laiteryhmien sekä tægien todettiin toimivan hyvin.

Agenttilaitteiden paikallisia käyttäjiä voidaan Landscapen kautta lukita tai vastavuoroisesti avata. Myös uusien käyttäjien luominen ja vanhojen poistaminen ovat Landscapessa mahdollista ja ne todettiin testeissä toimiviksi.

7.2.5 Vaihtoehtojen vertailu

Landscapen SaaS-ratkaisussa järjestelmän hallintapalvelimen sijaitessa Canonicalin palvelimella käyttäjän ei tarvitse huolehtia sen päivittämisestä, sillä Canonical hoitaa palvelimen päivityksen. Fingridillä on osittain negatiivisia kokemuksia joistakin SaaS-palveluista muiden projektien osalta. Huomattiin, että osa palvelun toiminnoista oli kokonaan poistunut sovelluspäivityksen myötä ja täten sovellus ei täyttänyt enää vaatimuksiaan. On-prem-ratkaisussa Landscape-palvelin ei pyöri Canonicalin palvelimella vaan oman laitteen sisällä, jolloin käyttäjä on itse vastuussa Landscapen päivittämisestä. On-prem-ratkaisussa hallintayhteys käyttää omia laitteita ja omaa verkkoa, eikä kulje ulkopuolisia väyliä pitkin. Kun Landscape On-prem otetaan käyttöön tietoturvalisessä ympäristössä, se on SaaS-ratkaisua turvallisempi vaihtoehto.

Landscape SaaS oli erittäin yksinkertainen ottaa käyttöön. Toiminnaltaan se oli vakaa ja luotettava. Vain satunnaisia virheitä päivityspakettien asennuksessa pääsi tapahtumaan. Landscape ilmoitti virheestä ja päivityksen asentamisen uusintayritys onnistui ilman virheitä.

On-premises oli huomattavasti vaikeammin käyttöönotettava ratkaisu. SaaS-ratkaisuun verrattuna On-prem oli yleistasoltaan hitaampi päivityspakettien ja muiden käskyjen toimitamisessa. Lisäksi On-premillä asennettavien päivitysten kohdalla sattui enemmän virheitä, joiden vuoksi päivitykset oli asennettava uudelleen. Uusinta-asennus onnistui kuitenkin kaikilla kerroilla. On-premin käyttöönotto vaatii enemmän resursseja toimiakseen, eikä tällöinkään toiminut yhtä tehokkaasti kuin SaaS-ratkaisu. SaaS saatiin toimimaan kymmenissä minuuteissa, mutta On-premin käyttöönotto venyi monien haasteiden vuoksi usean päivän mittaiseksi työksi.

Jotta vaihtoehtojen kustannuksia voitaisiin havainnollistaa paremmin, kuvitellaan sähkö-
aseman kytkinkentälle 50 reunalaitetta, jotka täytyy ottaa hallintaan. Reunalaitteet ovat
fyysisiä Ubuntu-palvelimia. Landscape SaaS -vaihtoehdolla maksu perustuu siihen,
kuinka monta tuntia laite on aktiivisena ja hallinnassa. Hallinta maksaa 0,01 Yhdysvaltain
dollaria tunnissa yhtä laitetta kohtaan. Tällöin vuosikustannukset kaikkia reunalaitteita
kohtaan ovat 4380 Yhdysvaltain dollaria eli noin 3860 euroa, jos laitteet ovat jatkuvasti
käynnissä. Jos vastaavanlaisia kytkinasemia tulee lisää ja tarvitaan yhteensä 500 reu-
nalaitetta, nousee kaikkien reunalaitteiden hallinta SaaS-ratkaisulla hintaan 38600 euroa
vuodessa. Valuutanmuunnoksissa käytettiin TransferWisen dollarikurssia 9.11.2018
11:44.

Landscape On-premises -vaihtoehto on ilmainen kymmenelle laitteelle, mutta sen jäl-
keen vaatii Ubuntu Advantage -paketin tilauksen. Tällöin hinta kohoaa huomattavasti
korkeammaksi kuin SaaS-ratkaisussa. Ubuntu Advantagen mukana tulee Landscape-
hallintatyökalun lisäksi Kernel Livepatch -päivitystyökalu, Extended Security Mainte-
nance -turvallisuusratkaisu sekä Knowledge Base, joka on verkkopohjainen tietokirjasto
Ubuntu Server -kehitykseen ja -testaukseen. Tällöin lisenssi 50 reunalaitteelle maksaa
vuodessa 11250 dollaria, mikä tekee noin 9910 euroa vuodessa. Tämän lisäksi tarvitaan
erillinen 2000 dollarin (1760 €) lisenssi On-premises-palvelimelle. 500 reunalaitteelle
hintaa kohoaisi reunalaitteiden osalta vuosittaiseen 99100 euroon, jonka lisäksi olisi han-
kittava tarvittava määrä lisenssejä palvelimille. Valuutanmuunnoksissa käytettiin Trans-
ferWisen dollarikurssia 9.11.2018 11:44. (Halpenny 2018.)

Taulukko 2. Landscape-vaihtoehtojen vertailutaulukko.

	SaaS	On-premises
Landscape	kyllä	kyllä
Livepatch	ei	kyllä
ESM	ei	kyllä
Knowledge Base	ei	kyllä
50 laitetta	3860 €/v	9910 €/v
500 laitetta	38600 €/v	99100 €/v
lisä-kustannukset	ei	1760 €/On-premises-palvelin

Testissä parhaaksi järjestelmähallintatyökaluksi valikoitui Landscape SaaS, koska se on varta vasten Ubuntu-pohjaisille järjestelmille suunniteltu työkalu. Sen käyttö oli helppoa ja yksinkertaista. Työkalu täytti kaikki ennalta asetetut vaatimukset ja hintansa puolesta se päihitti On-premises -vaihtoehdon. Muutamaa poikkeusta lukuun ottamatta kaikki sen kautta tehdyt hallintatoiminnot kulkeutuivat aina onnistuneesti perille hallittaville laitteille.

Vaikka Landscapen SaaS-vaihtoehto täyttikin kaikki vaatimukset ja tuntui kaikin puolin tehokkaalta hallintatyökalulta, sen ei katsottu tarjoavan riittävän suurta hyötyä hintaansa nähden. Yksittäisen reunalaitteen hankintahinnan ollessa noin 100 euroa, on noin 75 euron vuosimaksu yhden laitteen hallinnasta yksinkertaisesti liikaa. Fingridin IoT-projektien edetessä hallittavien laitteiden määrä kasvaa tulevien vuosien aikana paljon ja laitteiden hallintaan kuluvat kustannukset kasvaisivat valtaviksi. Landscapea voidaan vielä harkita, jos Canonicalin kanssa päästään hinnasta sopuun. Hinnan voidaan olettaa laskevan, jos Fingrid tarjoutuu ostamaan yhdellä kertaa suuren määrän SaaS-lisenssejä.

Yksi mahdollisuus olisi yhdistää molemmat Landscapen ratkaisut yhdeksi hybridiratkaisuksi. Jokaista sähköasemaa kohtaan olisi SaaS-hallinnassa muutama päälaitte, jotka toimivat On-premises-palvelimina. Sähköaseman muut reunalaitteet yhdistetään On-premises-palvelimiin, jolloin saadaan hyödynnettyä kymmenen ilmaista lisenssiä kutakin

palvelinta kohti. Esimerkiksi asema, jossa on 30 reunalaitetta tarvitsisi kolme On-premises-palvelimena toimivaa laitetta lisää. Toteutus ei olisi toiminnaltaan yhtä tehokas kuin puhdas SaaS- tai On-premises-ratkaisu, mutta ajaisi silti tehtävänsä ollen huomattavasti edullisempi.

Ubuntu Server -käyttöjärjestelmää käytetään yleensä suuremman mittakaavan palvelintietokoneissa, jonka vuoksi hinnoittelu on myös korkeampi. Fingridin tapauksessa Ubuntuä käytetään hyvinkin pienissä laitteissa, joita sijoitellaan kohdeasemille määrällisesti paljon. Canonicalilta ei löydy valmiina kevyisiin palvelintietokoneisiin tarkoitettua hallintatyökalua. Jos Fingrid vaihtaisi reunalaitteiden käyttöjärjestelmän Ubuntusta johonkin toiseen, voitaisiin tutkia useampia vaihtoehtoja, koska Ubuntulle hallintatyökaluja on hyvin vähän. Käyttöjärjestelmän vaihto aiheuttaisi kuitenkin koko IoT-konseptin muuttumisen ja tuottaisi turhan paljon ylimääräistä työtä.

Järjestelmänhallintatyökalun rinnalla lähdetään kokeilemaan eräänlaista lokienhallintaratkaisua. Reunalaitteille annetaan komento, joka käskee laitetta lataamaan automaattisesti saatavilla olevat päivitykset ja asentamaan ne. Lokienhallintatyökalulla reunalaitteelta kysytään lokitiedostoja, joista nähdään, ovatko päivitykset asentuneet onnistuneesti. Tämä on järjestelmänhallintatyökalua kevyempi ratkaisu ja se mahdollistaa laitteiden päivitys- ja tilatietojen monitoroimisen, mutta muita järjestelmänhallintatyökalun toimintoja ei tällöin saavuteta. Lokienhallintaratkaisua ei dokumentoitu, koska se ei kuulunut opinnäytetyön aiheeseen.

7.3 Telemetry

Microsoft Azuren tarjoamista IoT-vaihtoehtoista päädyttiin kokeilemaan ratkaisua, joka muodostaa pilvipalvelun ja anturijärjestelmän välisen yhteyden käyttäen Azure IoT Hub -palvelua. Samalla testattiin Azure IoT Edge -palvelua, joka mahdollistaa reunalaitteen käyttämisen IoT-yhdyskäytävänä. IoT-järjestelmätestien myötä myös muita Azuren palveluja yhdistettiin IoT Hubin ja IoT Edgen ympärille.

Azure IoT solutions accelerators- sekä Azure IoT Central -ratkaisut sivuutettiin testien osalta kokonaan. Vaikka ne tarjoavat valmiita helposti käyttöönotettavia ja muokattavissa olevia toteutus pohjia IoT-tarpeisiin, tahdottiin anturijärjestelmien IoT-osuus rakentaa pala kerrallaan vastaamaan tarpeita.

7.3.1 IoT Edge testissä

Azurea varten tehtiin verkkoselaimella käyttäjätili Azuren palveluportaaliin. Fingridillä oli valmiina testikäyttöä varten Azuren Standard-tilaus, resurssiryhmä ja IoT Hub. Resurssiryhmän tarkoitus on toimia varastotilana kaikille Azureen luoduille resursseille. Luodulle tilille sallittiin oikeudet testiratkaisun rakentamisen kannalta oleellisiin resursseihin.

IoT Hubin sisälle rekisteröitiin IoT Edge -toiminnallisuus. Rekisteröinti voitiin suorittaa Azure Portal -näköymästä tai Azuren CLI-komentoriviltä. Testissä rekisteröinti suoritettiin komentoriviltä. Syötetyn komennon parametreista tuli ilmetä, että luotiin uusi laite, jonka tuli olla edge-toiminen. Lisäksi komento määritteli, minkä nimiseen IoT Hubiin laite sijoitettiin ja mikä nimi uudelle laitteelle annettiin. Luodun IoT Edge -laitteen satunnais-generoitu liityntäkoodi, jolla laite saadaan yhdistettyä fyysiseen reunalaitteeseen, oli selvitettävä uudella komennolla.

Azure IoT Edgen käyttöönottoa jatkettiin HPE EL20 -laitteen paikalliselta komentoriviltä. Aluksi oli ladattava asennuksessa tarvittavat tiedostopakettit, asetukset ja julkinen avain. Paketit päivitettiin, jonka jälkeen komentorivikomennolla asennettiin Docker-konttien ajamiseen tarkoitettun Moby-työkalun moottori. Moby CLI eli työkalun komentorivi oli asennettava erillisellä komennolla. Tiedostopakettit päivitettiin uudelleen, jolloin reunalaitteella oli konttityökalu, mutta ei vielä yhtäkään aktiivista konttia.

Varsinainen IoT Edge -toiminnallisuus asennettiin Mobyin jälkeen uudella komennolla. IoT Edgen mukana laitteelle asentui IoT Edge runtime, joka sisälsi Edge Hub- ja Edge Agent -moduulit. Nämä kaksi moduulia siirtyivät automaattisesti Mobyin Docker-kontteihin ajettaviksi. Jotta IoT Edge -laite saisi yhteyden IoT Hubiin, oli muokattava IoT Edgen konfiguraatitiedostoa. Tiedostoon oli lisättävä aiemmin talteen otettu liityntäkoodi. Koodin lisäämisen jälkeen IoT Edge käynnistettiin uudelleen, jotta koodinmuutos tuli voimaan. Käynnistyksen jälkeen IoT Hubin ja reunalaitteella pyörivän IoT Edgen välinen yhteys oli muodostettu. Jos liityntäkoodin epäillään joutuneen väärin käsiin, voidaan koodi generoida uudelleen, jonka jälkeen uusi koodi on lisättävä konfiguraatitiedostoon.

IoT Edge -laitteelle haluttiin sijoittaa uusi moduuli. Tämä tehtiin Azure Portalissa. Testissä laitteeseen luotiin Microsoftin tarjoama valmis tempSensor-moduuli, joka generoi simuloitua lämpötila-, ilmanpaine- ja ilmankosteusdataa. Muiden moduulien tapaan tämäkin moduuli siirtyi IoT Edge -laitteessa Docker-konttiin. Oletusarvoilla tempSensor

tuottaa data-arvoja viiden sekunnin välein. Erilliset mittausarvot kuvastavat kuviteltua mitattavaa komponenttia ja sen lähiympäristöä. Lähiympäristön lämpötila ja kosteus pysyvät jokseenkin samoissa arvoissa, mutta komponentin lämpötila kohoaa hiljalleen 20 Celsius-asteesta noin 100 asteeseen.

Koska reunalaitteita oli kaksi, kaikki työvaiheet tehtiin myös Dellin reunalaitteelle. Tuloksena saatiin IoT Hub, joka oli yhdistetty kahteen fyysiseen reunalaitteeseen, joissa kummassakin pyöri IoT Edge -toiminnallisuus. Kummassakin laitteessa oli sisällä kolme moduulia. Kaksi moduuleista operoi komponenttien ja moduulien välisiä yhteyksiä. Yksi moduuli, tempSensor, tuotti jatkuvasti simuloitua lämpötila- ja ilmankosteusdataa.

7.3.2 Datan varastointi

Oletuksena tempSensor-moduuli varastoi tuottamansa datan laitteen sisäiseen puskuri-varastoon. Siellä data varastoituu lokiin, jonka selaaminen voi olla hyvin haastavaa. Pitkään aktiivisena olleen laitteen lokissa voi olla jopa kymmeniä, ellei satoja tuhansia mitaustuloksia. Jotta dataa voisi tarkastella huomattavasti helpommin, oli luotava datan säilytystä varten varasto. Testissä päädyttiin Azure Blob Storage -ratkaisuun, sillä se on edullinen ja kyvykäs varastoimaan suuria datamääriä. Vaihtoehtoisesti anturidata voitaisiin viedä esimerkiksi Structured Query Language (SQL)-tietokantaan tai Azure Cosmos DB -tietokantaan.

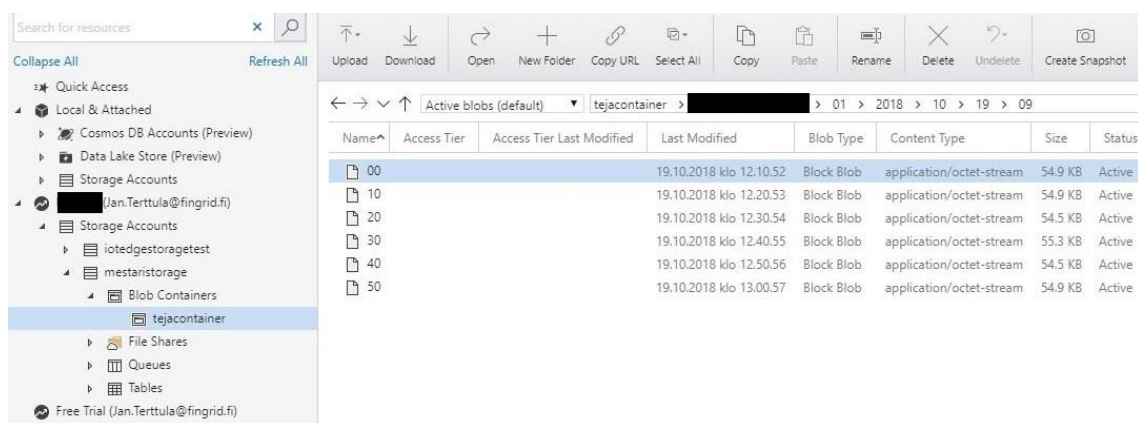
Azure Portalin kautta luotiin Microsoft-varastotili samaan resurssiryhmään, jossa IoT Hub oli. Varastotilille määriteltiin nimi, sijainti, tyyppi sekä varaston taso. Tasolla viitataan siihen, tarvitseeko dataa hakea varastosta usein (hot) vai harvoin (cold). Testissä luotiin hot-tyypin varasto. Varastotilin sisään luotiin Blob-säiliö, jolle piti määritellä vain nimi ja käyttöasteen asetus. Käyttöasteella tarkoitetaan, minkälaiset oikeudet säiliön sisällön tarkasteluun tuntemattomilla käyttäjillä on. Testissä valittiin asetus, joka estää pääsyn kaikista tuntemattomista lähteistä.

Blob-säiliö oli liitettävä IoT Hubiin, jotta data kulkeutuisi tempSensor-moduulilta Blob-varastoon. Azure Portalissa IoT Hubin asetuksissa määriteltiin kohde, johon IoT Hub lähettää dataa. Kohteeksi valittiin aiemmin luodun varastotilin sisälle luotu Blob-säiliö. Samalla oli myös mahdollista muokata datapakettien lähetyksessä käytettävät elinaika-asetukset. Asetukset määrittävät, kuinka monta kertaa dataa yritetään lähettää varastoon ja

kuinka kauan datan lähetyksestä syntyvä ilmoitus on näkyvässä. Testissä pitäydettiin oletusasetuksissa.

Kun IoT Hubille oli määritelty datavarasto, oli luotava reitti, jota pitkin data kulkeutuu varastoon. Reitti luotiin Portal-käyttöliittymässä IoT Hubin asetuksista. Reitille oli annettava nimi. Lisäksi piti täsmentää datan muoto sekä tarvittaessa säännöt, joita datan oli noudatettava päästäkseen käyttämään reittiä. Datan muodoksi valittiin telemetriaviestit ja ylimääräisiä sääntöjä ei lisätty. Reitille oli vielä valittava päätepiste, johon data kuljeteaan. Päätepisteeksi valittiin aiemmin luotu Blob-säiliö. Samalla voitiin valita arvot, jotka määrittävät lähetetyn datan suurimman sallitun koon sekä taajuuden, kuinka usein data lähetetään Blob-säiliöön. Datapaketin kooksi valittiin 10 MB ja taajuudeksi 600 sekuntia. Näillä asetuksilla kaikki saapuva telemetriadata kulkee reitin läpi aiemmin luotuun Blob-varastoon. Dataa saapuu 600 sekunnin eli kymmenen minuutin välein ja kukin varastoon saapuva datapaketti on kooltaan enimmillään 10 MB.

Varastoituja telemetriatapaketteja pystyttiin tarkastelemaan Azure Portalissa, kun navigoitiin käyttöliittymässä omaan varastotiliin ja sieltä Blob-säiliöön. Vaihtoehtoisesti varastoa voi seurata erillisellä tietokoneelle asennettavalla ohjelmalla, Microsoft Azure Storage Explorerilla (MASE). Se voidaan liittää Azure-tiliin, jolloin kaikki sinne luodut varastot tulevat MASE:n nähtäville. Varasto pohjautuu kansiorakenteeseen, johon datatiedostot järjestyvät automaattisesti päivämäärän ja kellonajan mukaan. Kansiorakenne koostuu IoT Hub -, vuosi-, kuukausi-, päivä- sekä tuntikansioista. (Azure Storage Explorer 2017.)



Kuva 7. Kuvankaappaus Microsoft Azure Storage Explorerista. Kuvasta näkyy yhdeltä IoT Edge-laitteelta 19.10.2018 kello 12:n ja 13:n välillä Blob-varastoon lähetetyt telemetriatapaketit.

Varaston tiedostot ovat ladattavissa ja niiden sisältöä voidaan tarkastella esimerkiksi tavallisella Windowsin tekstieditorilla, mutta suositeltavampaa on käyttää esimerkiksi Notepad++-tekstieditoria, joka näyttää tiedoston rakenteen selkeämmin.

7.3.3 Anturimoduulien hallinta

ASA

Simuloitua dataa tuottavan tempSensor-moduulin telemetriaviestien prosessointia varten otettiin molemmissa IoT Edge -laitteissa käyttöön Azure Stream Analytics -palvelu. IoT Edge ja ASA on integroitu siten, että Azure Portalin kautta luotu ASA-työ voidaan suoraan sijoittaa IoT Edgeen moduuliksi.

Azure Portalissa luotiin uusi Stream Analytics Job eli ASA-työ. Työlle annettiin nimi ja se luotiin samaan resurssiryhmään kuin aikaisemmatkin resurssit. Työn ympäristöksi valittiin Edge eli työ määritettiin toimimaan reunalaitteella.

Portal-käyttöliittymässä navigoitiin ASA-työn asetuksiin, jossa oli asetettava työn sisään- ja ulostulo sekä tiedustelukoodi. Sisääntuloksi asetettiin Edge Hub, nimeksi temperature ja formaatiksi JSON. Muut asetukset jätettiin oletusarvoisiksi. Ulostuloksi asetettiin niin ikään Edge Hub, nimeksi alert ja formaatiksi JSON. Tiedustelukoodi määrittää, mikä on ASA-työn tehtävä. Koodiksi kirjoitettiin lyhyt SQL-koodinpätkä. Koodinpätkä määrittä sen, että mitattavan moduulin lämpötilan kohotessa yli 30 sekunniksi yli 70 asteeseen, lähetettiin telemetriaviestien seassa hälytys, ja mitattava moduuli oli käynnistettävä uudelleen.

ASA-työn asetusten määrittämisen jälkeen oli työ siirrettävä IoT Edgelle toimintaan. Azure Portalissa navigoitiin halutun IoT Edge -laitteen asetuksiin, josta laitteelle lisättiin uusi moduuli tempSensor-moduulin rinnalle. Uudeksi moduuliksi valittiin juuri luotu ASA-työ. Moduulien välisen reitin määrittävää JSON-koodia oli muokattava siten, että sekä tempSensor että ASA-työ toimivat yhtä aikaa keskenään. Koodista kävi ilmi, että ASA-moduulin haluttiin analysoivan tempSensorin tuottamaa datavirtaa ja resetoivan tempSensor-moduulin, kun lämpötila karkaa liian korkealle. Molempien moduulien viestit tulisi lähettää IoT Hubille, joka lähettää ne edelleen Blob-varastoon.

Moduuliasetukset tallennettiin, jolloin uusi moduuli siirtyi automaattisesti hetken kuluttua reunalaitteen IoT Edgeen Docker-kontin sisälle. ASA-moduuli aloitti saman tien tempSensor-moduulin datavirran tutkimisen. ASA-moduuli todettiin toimivaksi tarkkailemalla reunalaitteen moduulilokia tai Storage Explorerilla Blob-varaston sisältöä. Molemmista nähtiin, että tempSensor-moduulin generoidessa liian korkeaa lämpötilaa ASA-moduuli lähetti datavirran joukkoon hälytyksen. Sen jälkeen tempSensor käynnistyi uudelleen aloittaen lämpötila-arvojen generoimisen alhaisista lukemista.

ACR

Järjestelmässä toimivien moduulien käytön helpottamista varten haluttiin ottaa käyttöön Azure Container Registry. Tällä konttirekisterillä voidaan etätoimisesti muokata IoT Edgen Docker-konteissa ajettavia moduuleja. Menetelmä on erittäin käytännöllinen, koska moduulien pyöriessä itsenäisesti Docker-konteissa ne eivät ole reunalaitteen Ubuntu-käyttöjärjestelmän kanssa missään tekemisissä. Ubuntu voi toimia järjestelmän pohjalla, ja moduulit toimivat Ubuntusta täysin eristettyinä konttien sisässä. Tällöin toimintojen välille ei pääse syntymään minkäänlaisia yhteensopivuusongelmia, jotka haittaisivat järjestelmän toimintaa.

Konttirekisteri luotiin Azure Portalissa samaan resurssiryhmään, johon muutkin resurssit oli luotu. Rekisterille oli annettava ainoastaan nimi. Luomisen jälkeen konttirekisterin kirjautumistiedot kirjattiin ylös myöhempää käyttöä varten.

IoT Edge -laitteena toimivalle Ubuntu-koneelle oli jo testin aikaisemmassa vaiheessa asennettu Dockeriin pohjautuva Moby-työkalu, joka pystyttiin yhdistämään Azureen luotuun konttirekisteriin. Yhdistäminen suoritettiin Ubuntu komentoriviltä komennolla, johon oli syötettävä rekisterin kirjautumistiedot. Tiedot sisälsivät konttirekisterin kirjautumispalvelimen osoitteen, käyttäjänimen sekä satunnaisgeneroidun salasanan. Kun ilmoitus onnistuneesta kirjautumisesta vastaanotettiin, voitiin reunalaitteella toimiva moduuli työntää konttirekisteriin. Testissä rekisteriin vietiin lämpötilaa simuloiva tempSensor-moduuli. Azure Portal -käyttöliittymässä konttirekisterin säilössä voitiin nähdä juuri sinne siirretty moduuli.

Täysin Ubuntu-laitteista erillään olevalle Windows-tietokoneelle ladattiin ja asennettiin Docker for Windows -sovellus. Tietokoneen virtualisointitekniikka piti aktivoida Basic Input-Output Systemistä (BIOS). BIOS on pohjatason ohjelma, joka huolehtii käyttöjär-

jestelmän lataamisesta tietokoneen keskusmuistiin ja käyttöjärjestelmän käynnistämisestä tietokoneen käynnistyessä. Lisäksi tietokoneen hallinta-asetuksissa oli sallittava Dockerin käyttö lisäämällä haluttu käyttäjä Dockerin käyttäjäryhmään. Dockeria käytettiin Windows-koneen omalta komentoriviltä. Dockerin oli muodostettava yhteys aiemmin luotuun konttirekisteriin samalla komennolla, jolla Ubuntu oli yhdistetty.

ACR:n toiminta haluttiin testata mahdollisimman yksinkertaisella menetelmällä. Ensiksi Windows-laitteen komentoriville syötetyllä komennolla konttirekisteristä haettiin sinne työnnetty tempSensor-moduuli. Haettu moduuli tunnistettiin ja merkittiin eli ”tägättiin” uudella komennolla, jolloin moduulille asetettiin myös uusi nimi. Tämä tismalleen sama, mutta uudenniminen moduuli työnnettiin takaisin konttirekisteriin. Azure Portalista nähtiin konttirekisterin sisältävän kaksi erinimistä, joskin muuten identtistä moduulia.

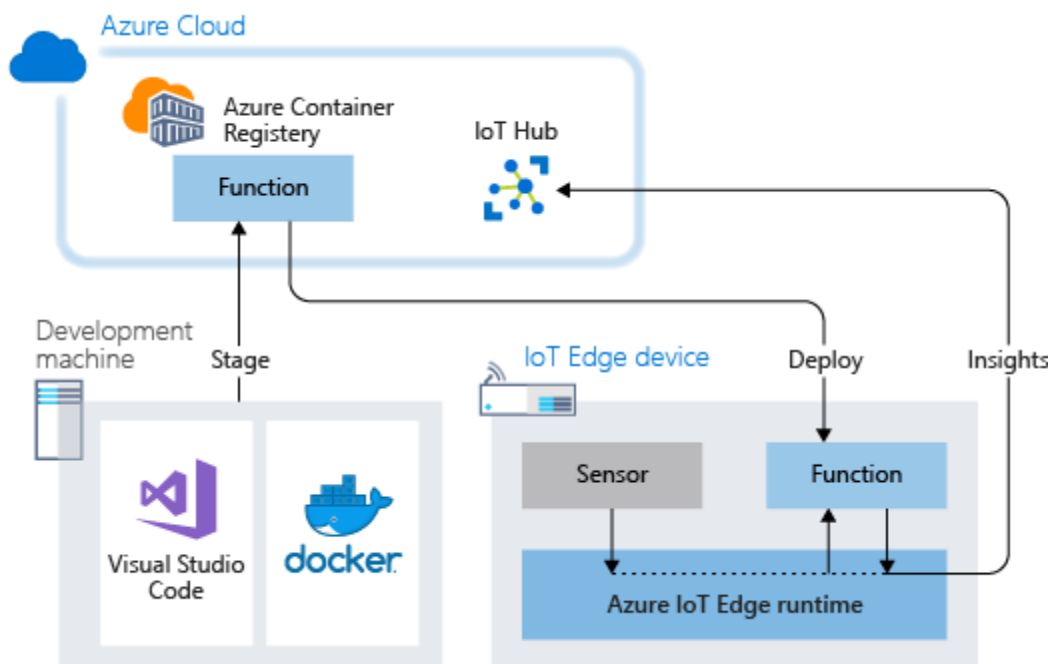
Jotta uudelleennimetyn moduulin saisi siirrettyä IoT Edgeen Docker-konttiin ajettavaksi, oli määriteltävä Azure Portalissa halutulla reunalaitteella käytettävä konttirekisteri. Tämän jälkeen reunalaitteelle luotiin uusi moduuli, jolle syötettiin polku, jolla IoT Edge löytää oikean moduulin konttirekisteristä. Moduuli siirtyi konttirekisteristä IoT Edge -laitteelle Docker-konttiin toimintaan.

7.3.4 Toimintaperiaate

Telemetriadataa tuottavan järjestelmän etähallinta keskittyy Azuren IoT Edgeen. Sen ympärille rakentuu joukko muitakin palveluja, mutta IoT Edge on juuri se sovellus, jolla etähallintayhteys saavutetaan sähköasemalla sijaitsevaan IoT-yhdyskäytävään, laitteeseen, johon anturit ovat liitetty. Pilvialustana toimii Microsoft Azure. IoT Hub toimii yhteydenmuodostajana IoT Edgen ja pilven välillä.

Jokaisessa IoT Edge -laitteessa on ainakin kaksi moduulia: edgeAgent ja edgeHub. Näiden lisäksi laitteessa voi toimia useita muitakin moduuleja, jotka suorittavat tehtäviään. Kun moduulit sijoitetaan laitteelle, tarvitsee samalla lähettää sijoitusluettelo (deployment manifest). Luettelo on JSON-tiedosto, ja se määrittelee, mitä moduuleja laitteeseen sijoitetaan ja miten ne toimivat keskenään. Lisäksi luettelo kertoo, ovatko moduulit peräisin esimerkiksi Microsoftin moduuliarkistosta vai käyttäjän luomasta konttirekisteristä. Kaikki IoT Edge -laitteet tarvitsevat toimiakseen sijoitusluettelon. Luettelo voidaan generoida automaattisesti Azure Portalin kautta, mutta se on myös mahdollista kirjoittaa ja käyttää manuaalisesti itse. (Learn how to deploy modules and establish routes in IoT Edge 2018.)

Azureen luotu ACR-konttirekisteri on liitetty reunalaitteeseen. Ohjelmointityökaluilla, kuten Visual Studio Codella, voidaan luoda uusia konttikuvia. Nämä konttikuvat voidaan tallettaa konttirekisteriin miltä vain tietokoneelta, jossa on Docker asennettuna. Azure Portalin kautta rekisterin konttikuvia voidaan sijoittaa halutuille IoT Edge -laitteille kontteihin ajettaviksi. Laitteen voi myös konfiguroida kyselemään konttirekisteriltä uusista tai muokatuista konteista, jolloin laite automaattisesti hakee tarvittavat kontit itselleen suoraan rekisteristä.



Kuva 8. Moduulien kulkeutuminen Dockerilta Azuren konttirekisterin kautta reunalaitteella ajettavaan konttiin. (Tutorial: Deploy Azure functions as IoT Edge modules (preview) 2018.)

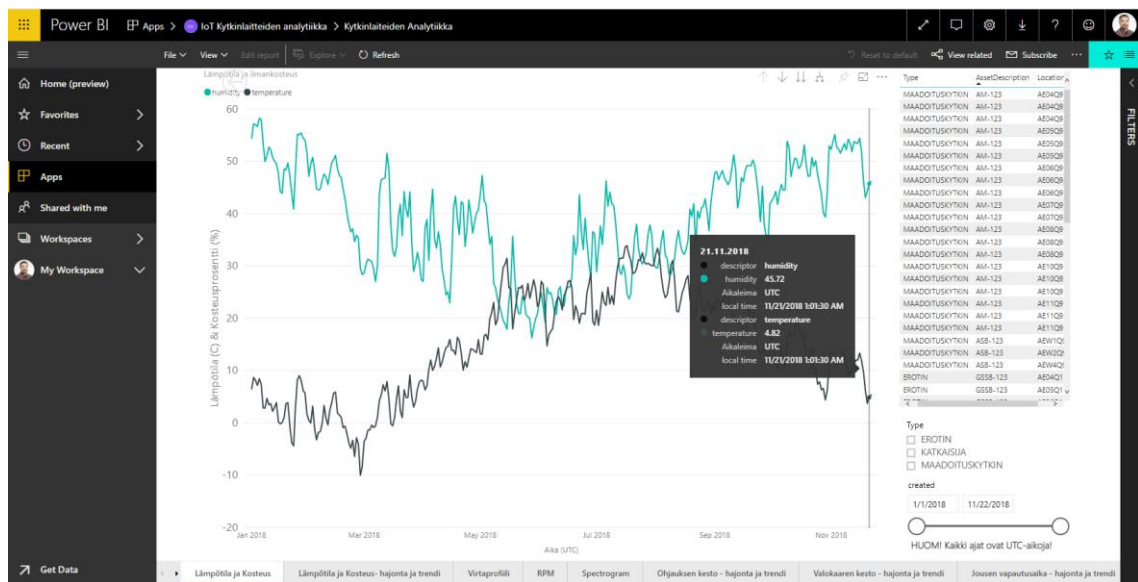
Reunalaitteelle konttiin sijoitettu Azure Stream Analytics -moduuli analysoi anturin mittaamaa datavirtaa. ASA voidaan määrittää suorittamaan erilaisia toimenpiteitä, jos datavirrassa esiintyy järjestelmän toiminnan kannalta huonoja arvoja. Jos esimerkiksi tärinää mittaava anturi mittaa komponentissa huomattavan suuria lukemia, ASA-moduuli havaitsee ongelman, tekee tilanteesta hälytyksen ja voi mahdollisesti lähettää tärisevälle komponentille sammutuskäskyn.

Mitattu data kulkeutuu anturilta IoT Edgen ja IoT Hubin kautta Azuren pilveen ja siellä luotuu Blob-datavarastoon. Testissä IoT Hub käytti viestinvaihdossa AMQP-protokollaa. Sekä IoT Hub että IoT Edge tukevat myös MQTT:tä, mutta testissä päätettiin pitäytyä AMQP:ssä, sillä se on viestinnällisiltä ominaisuuksiltaan kyvykkäämpi ja turvallisempi.

Rakennettu järjestelmä oli toiminnaltaan kyvykäs ja varma. Azuren komponenteilla voidaan toteuttaa hyvin monenlaisia monipuolisia ratkaisuja. Testiratkaisun kaltainen järjestelmä voitaisiin toteuttaa Fingridin sähköasemille. Tietysti valmis tempSensor-moduuli pitäisi korvata oikeisiin antureihin yhteensopivilla moduuleilla, mutta muilta osin järjestelmän kokoonpano pysyisi hyvin pitkälti samana.

7.3.5 Visualisointi

Koska telemetriadatan seuraaminen lokitiedostoista on hankalaa, on parempi käyttää visualisointityökalua. Soveltuvia työkaluja löytyy runsaasti, mutta Fingrid on päätenyt käyttämään Microsoftin kehittämää Power BI -sovellusta. Power BI:llä voidaan muodostaa yhteys erillisiin tietolähteisiin aina Excel-taulukoista pilvipohjaisiin hybriditietovarastoihin. Myös Azure-tili on mahdollista liittää suoraan Power BI:hin, jolloin tilin haluttu data on vaivattomasti tuotavissa varastosta sovellukseen. Power BI:stä löytyvät monipuoliset työkalut, joilla erilaisten tietokantojen ja datavarastojen sisältö voidaan tuoda vaikeaselkoisen taulukkomaisen rakenteen sijasta ihmissilmää miellyttävämpään visuaaliseen muotoon. Data voidaan saattaa esimerkiksi diagrammeihin. (What is Power BI Desktop? 2018.)



Kuva 9. Erään Fingridin sähköaseman toiminnassa olevan IoT-järjestelmän vuoden 2018 aikana mittaamia ilman lämpötila- ja kosteusprosenttiarvoja visualisoituna PowerBI-näkymään.

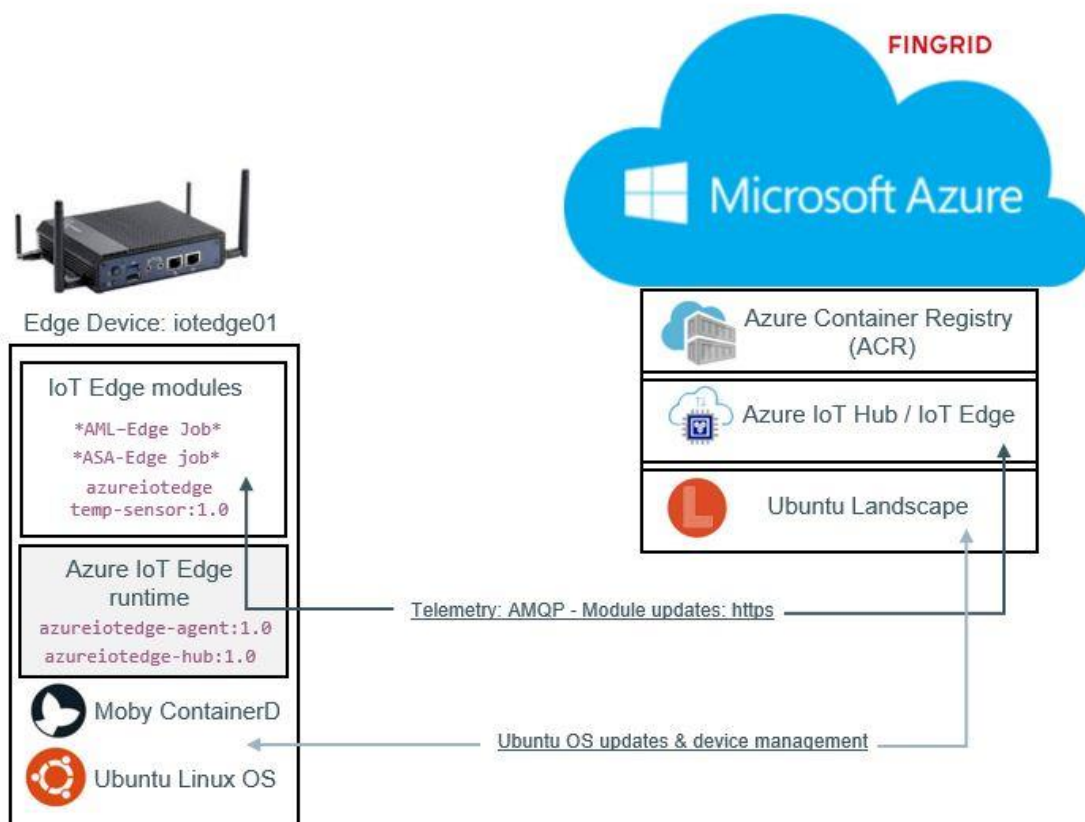
Power BI:stä löytyy useampia palveluja erilaisiin käyttötarpeisiin. Tärkeimmät palvelut useimmissa tapauksissa kuitenkin lienevät Power BI Desktop ja Power BI Service. Lisäksi Power BI:stä löytyy mobiilisovellukset eri käyttöjärjestelmille, raporttipalvelinratkaisu sekä kehittäjätyökaluja. Työpöytäsovelluksella, Power BI Desktopilla, toteutetaan itse datan muokkaus halutun näköiseksi. Se on nimestänsäkin päätellen ladattava paikallisesti työasemalle. Muokkauksen jälkeen visuaalinen näkymä voidaan jakaa muiden nähtäväksi Power BI Servicellä. (What is Power BI Desktop? 2018.)

Perusteellisesti rakennettu visualisointijärjestelmä toimii siten, että Power BI on linkitetty datavarastoon, josta Power BI hakee arvot ja piirtää niiden perusteella näkymän kaaviot. Kun uusia arvoja virtaa antureilta varastoon, visuaalinen näkymä muuttuu samalla.

8 Yhteenveto

Opinnäytetyössä perehdyttiin digitalisaation mukanaan tuomiin mahdollisuuksiin. Digitalisaation suuri teknologiamullistus, Internet of Things, tulee vaikuttamaan merkittävästi tulevaisuuden sähkökomponenttien seuraamiseen ja hallintaan. Työssä käytiin läpi IoT:n ja etähallintakonseptin keskeisimmät asiat. Samalla esiteltiin, mitä hyötyä Fingrid Oyj saisi sähköasemilleen käyttöönotettavasta IoT-järjestelmästä. Työn pääasiana selvitettiin ja analysoitiin IoT-laitteiden etähallintaan liittyviä ratkaisuja ja käytäntöjä.

Kokonaisuuden hahmottamiseksi työn käytännön osuudessa rakennettiin useista tietokoneista ja palveluista koostuva IoT-järjestelmä. Järjestelmän hallinnassa käytettiin järjestelmänhallintatyökalua ja pilvipalvelualustaa. Myös useat pilvialustan IoT-palvelut olivat käytössä työn edetessä. Rakennusvaihe dokumentoitiin. Rakennettua IoT-järjestelmää ja sen palvelujen toimintaa tutkittiin. Myös tutkimuksissa selvinneet asiat dokumentoitiin ja eri ratkaisuvaihtoehtojen toimintatapaa ja -varmuutta vertailtiin.



Kuva 10. Kaaviokuva työssä rakennetun IoT-järjestelmän palveluista.

IoT-yhdyskäytävän etähallintaa varten tutkittiin kahta järjestelmänhallintatyökalua. Ominaisuuksiltaan parhaaksi työkaluksi valikoitui Ubuntu Landscape, mutta nykyisen hintansa puolesta Landscape ei osoittautunut sopivaksi vaihtoehdoksi. Laitteen hankintahinta on verrattain pieni Landscape-hallinnan kustannuksiin verrattuna. Sopivan yhdyskäytävän etähallintaratkaisun löytäminen jäi vielä tekemättä. Fingridillä tutkitaan yhdyskäytävien osalta myös lokienhallintaratkaisua, jolla voitaisiin järjestelmänhallintatyökalua kevyemmin toteuttaa joitain laitevalvonnan kannalta tärkeitä ominaisuuksia.

Pilvialustana Microsoft Azure osoittautui erittäin kyvykkääksi vaihtoehdoksi anturien etähallintaan. Azure ja sen tarjoamat lukuisat IoT-palvelut olivat hyvin toimintavarmoja ja monipuolisia. Tuntui jopa siltä, että vain mielikuvitus on rajana sille, mitä Azurella voidaan toteuttaa. Rajoittavana tekijänä oli ainoastaan pilvialustan monipuolisuudesta johtuva käyttötaitojen puute. Välillä oli hankala ymmärtää järjestelmän monimutkaisia toimintatapoja. Onneksi Microsoftin verkkosivujen opetusmateriaali oli helppolukuista ja ymmärrettävää ja siitä sai tarvittaessa lisää ymmärrystä.

Fingrid tulee seuraavana vuonna rakentamaan etähallittavia IoT-anturijärjestelmiä ainakin kahdeksalle sähköasemalle. Näille asemille IoT-yhdyskäytäviä on mahdollisesti tulossa jopa yli 500. Tavoitteena on käyttöönottaa IoT-järjestelmä kaikille Fingridin sähköasemille tulevina vuosina, jolloin reunalaitteiden ja anturien määrä skaalautuisi useisiin tuhansiin. Näin suureen kokonaisuuteen on tärkeää valita toimintavarma ja kustannustehokas ratkaisu. Pilvipalvelualustana tulee hyvin todennäköisesti olemaan Microsoft Azure.

Opinnäytetyön aikana opittiin laajalti IoT:n mahdollisuuksista teknologian kehityksessä ja IoT:n vaikutuksista tulevaisuuden tekniikkaan. Linuxiin pohjautuvat käyttöjärjestelmät ovat keskenään samankaltaisia ja IoT-yhdyskäytävän käyttöjärjestelmänä oli Linuxiin pohjautuva Ubuntu, joka tuli työn aikana tutuksi. Työn aihe oli mielenkiintoinen ja on kiinnostava nähdä, mihin suuntaan maailmamme kehitty teknologian ja varsinkin IoT:n ponnistaessa kunnolla vauhtiin.

Lähteet

Fingrid, Esittely. 2018. Luettu 22.8.2018. <https://www.fingrid.fi/sivut/yhtio/esittely/>.

Laitinen, T. 2018. DIOT projektisuunnitelma, Luettu 27.8.2018.

Fingrid, Strategiset hankkeet – Hankekortit. 2017. Luettu 28.8.2018.

Olkiluoto 1 kytkettiin takaisin kantaverkkoon, sähköpulan uhka on ohi – ”Poikkeuksellinen tapaus”. 2018. Luettu 28.8.2018. <https://yle.fi/uutiset/3-10312198>.

What is Azure? N.d. Luettu 10.9.2018. <https://azure.microsoft.com/en-gb/overview/what-is-azure/>.

What is Azure IoT Hub? 2018. Luettu 10.9.2018. <https://docs.microsoft.com/en-us/azure/iot-hub/about-iot-hub>.

Eronen, H. 2016. IaaS, PaaS, SaaS? Mikä pilvipalvelu sopii yrityksellesi. Luettu 10.9.2018. <https://blog.planeetta.net/iaas-paas-saas>.

Azure IoT technologies and solutions: PaaS and SaaS. 2018. Luettu 12.9.2018. <https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-services-and-technologies>.

What is Azure IoT Edge? 2018. Luettu 12.9.2018. <https://docs.microsoft.com/en-us/azure/iot-edge/about-iot-edge>.

Ojala, R. 2017. MQTT IoT-protokolla. Opinnäytetyö, AMK, Jyväskylän ammattikorkeakoulu, tekniikan ja liikenteen ala, tieto- ja viestintätekniikan koulutusohjelma. Luettu 14.9.2018. <http://www.theseus.fi/handle/10024/139798>.

Introduction to object storage in Azure. 2018. Luettu 19.9.2018. <https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction>.

Azure Storage Explorer. 2017. Luettu 19.9.2018. <https://azure.microsoft.com/en-us/features/storage-explorer/>.

Microsoft Azure Portal. N.d. Luettu 20.9.2018. <https://azure.microsoft.com/en-us/features/azure-portal/>.

Azure, Cloud Shell. N.d. Luettu 20.9.2018. <https://azure.microsoft.com/en-us/features/cloud-shell/>.

Introduction to private Docker container registries in Azure. 2018. Luettu 20.9.2018. <https://docs.microsoft.com/en-us/azure/container-registry/container-registry-intro>.

Wikipedia, OSI-malli. N.d. Luettu 20.9.2018. <https://fi.wikipedia.org/wiki/OSI-malli>.

Tezer, O.S. 2013. An Advanced Message Queuing Protocol (AMQP) Walkthrough. Luettu 20.9.2018. <https://www.digitalocean.com/community/tutorials/an-advanced-message-queuing-protocol-amqp-walkthrough>.

State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating. 2018. Luettu 21.9.2018. <https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>.

Desai, N. 2016. What is an IoT Gateway and How Do I Keep It Secure? Luettu 24.9.2018. <https://www.globalsign.com/en/blog/what-is-an-iot-gateway-device/>.

Understand the Azure IoT Edge runtime and its architecture. 2018. Luettu 27.9.2018. <https://docs.microsoft.com/en-us/azure/iot-edge/iot-edge-runtime>.

Introducing JSON. N.d. Luettu 1.10.2018. <https://www.json.org/>.

Understand and use device twins in IoT Hub. 2018. Luettu 1.10.2018. <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-device-twins>.

Understand and use module twins in IoT Hub. 2018. Luettu 2.10.2018. <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-module-twins>.

What is Power BI Desktop? 2018. Luettu 2.10.2018. <https://docs.microsoft.com/en-us/power-bi/desktop-what-is-desktop>.

Schnober, C. 2014. Canonical's Landscape tool maintains Ubuntu environments. Luettu 5.10.2018. <http://www.admin-magazine.com/Archive/2014/20/Landscape>.

Rubens, P. 2017. VMware's Pulse Perfectly Timed for an Increasingly Internet of Things World. Luettu 5.10.2018. <https://www.serverwatch.com/server-trends/vmwares-pulse-debut-perfectly-timed-for-an-increasingly-internet-of-things-world.html>.

What is Stream Analytics? 2018. Luettu 22.10.2018. <https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction>.

Remote Monitoring & Management (RMM) Defined. N.d. Luettu 26.10.2018. <https://www.continuum.net/resources/mspedia/remote-monitoring-management-rmm-explained>.

Rouse, M. 2016. internet of things (IoT). Luettu 29.10.2018 <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>.

Tutorial: Deploy Azure functions as IoT Edge modules (preview). 2018. Luettu 2.11.2018. <https://docs.microsoft.com/en-us/azure/iot-edge/tutorial-deploy-function>.

Dimotakis, N. 2018. Puhelinpalaveri Landscapeen liittyen. Canonical. Haastattelu 2.11.2018.

Learn how to deploy modules and establish routes in IoT Edge. 2018. Luettu 7.11.2018. <https://docs.microsoft.com/en-us/azure/iot-edge/module-composition>.

Shaw, K. 2018. The Osi model explained: How to understand (and remember) the 7 layer network model. Luettu 7.11.2018. <https://www.networkworld.com/article/3239677/lan-wan/the-osi-model-explained-how-to-understand-and-remember-the-7-layer-network-model.html>.

Halpenny, L. 2018. Sähköpostikeskustelu Landscapen hinnastoon liittyen. Canonical. Haastattelu 9.11.2018.

Kotilainen, S. 2017. Koodi sujahtaa konttiin – sovellusten kehittäminen mullistuu. Luettu 12.11.2018. https://www.tivi.fi/Kaikki_uutiset/koodi-sujahtaa-konttiin-sovellusten-kehittaminen-mullistuu-6674085.

Huomo, T., Vähä-Heikkilä, T., Halunen, K., 2018. Tietoturva koskettaa koko yhteiskuntaa – VTT vastaa haasteisiin. ITPro-lehti, 11, 13. Luettu 14.11.2018.

Techopedia, Telemetry. N.d. Luettu 22.11.2018. <https://www.techopedia.com/definition/14853/telemetry>.