VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Niina Niemelä

IOT-PALVELUN LUOMINEN THINGWORX-ALUSTALLA

Liiketalous
2018

VAASAN AMMATTIKORKEAKOULU
Tietojenkäsittely

TIIVISTELMÄ

Tekijä	Niina Niemelä
Opinnäytetyön nimi	IoT-palvelun luominen ThingWorx-alustalla
Vuosi	2018
Kieli	suomi
Sivumäärä	60+10
Ohjaaja	Kenneth Norrgård

Opinnäytetyön tavoitteena oli saada peruskäsitys esineiden internetistä sekä Big datasta. Tavoitteena oli myös toteuttaa pienimuotoinen IoT-palvelu projektimuodossa, joka hyödyntää ThingWorx-alustaa sekä Raspberry Pi 3 B-mallia. Työn tarkoituksena oli ymmärtää keskeisimmät osa-alueet, joista nämä teknologiat koostuvat sekä ymmärtää miten näitä voidaan käyttää niin yritysmaailmassa kuin myös kuluttajamarkkinoilla. Esineiden internetin ja Big datan käyttömahdollisuudet ovat lähes loputtomat ja kaikkia käyttötapauksia ei ole vielä ajateltukaan. Nämä teknologiat ovat hyvin ajankohtaisia ja varmasti myös tulevaisuudessa vielä enemmän käytössä. Opinnäytetyön osatavoitteena oli tutustua Raspberry Pi 3 B-mallin, erilaisten anturien sekä ThingWorx-alustan käyttöön ja mahdollisesti auttaa myös muita aiheesta kiinnostuneita ymmärtämään paremmin näiden teknologioiden käyttöä.

Teknologiapinon avulla eriytettiin esineiden internetin palvelu pienempiin osa-alueisiin, joiden avulla voidaan paremmin ymmärtää mitä kyseiseen teknologiaan kuuluu. Big data on oleellinen osa esineiden internetiä, sillä sen avulla voidaan käsitellä isoja määriä anturien luomaa tietovirtaa sekä tuottaa lisäarvoa tiedon analysoinnilla. Linkitetyssä ja avoimessa datassa on mahdollisuuksia luoda uusia näkökulmia ja monipuolisempia tiedon analysointeja suurempiin projekteihin. Ongelmia ja haasteita näistäkin teknologioista löytyy ja opinnäytetyössä käsitellään sekä pohditaan myös näiden teknologioiden vajavaisuuksia kriittisestä näkökulmasta.

Teoriassa esitellyn teknologiapinon käyttäminen oli helppoa suunnittelussa sekä toteutuksessa. Suunnittelussa käytettiin erilaisia suunnitteludokumentteja, muun muassa prosessikaaviota ja käyttäjätarinoita. Toteutusvaiheessa kaikkia suunniteltuja antureita ei käytetty, mutta palvelun toteutus oli silti onnistunut. ThingWorx-alusta ja Raspberry Pi 3 B-malli olivat helppokäyttöisiä. Ongelmaksi kehkeytyi etäisyysanturin asentaminen, ohjelmointi sekä testaus. Opinnäytetyön avulla opin käyttämään Raspberry Pi-mikrotietokonetta, ThingWorx-alustaa sekä ymmärtämään IoT:iä ja Big dataa. Opinnäytetyön jälkeen aion jatkaa saman aiheen parissa omalla ajalla.

Avainsanat Esineiden internet, Big data, anturit, IoT alusta, IoT suunnittelu

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Niina Niemelä
Title	Creating an IoT service with ThingWorx platform
Year	2018
Language	Finnish
Pages	60+10
Name of Supervisor	Kenneth Norrgård

The main aims for this thesis were to establish a basic understanding of Internet of Things and Big Data and to execute a small IoT project using the Raspberry Pi 3 Model B and the ThingWorx platform. The main objective of the thesis was to comprehend the crucial details of IoT and Big Data and to understand more how these technologies could be utilized in the corporate as well as in the consumer market. The possibilities of these technologies are nearly limitless and it is certain that there are other uses that have not been yet thought of or innovated as of now. Overall, these technologies are very topical and will be discussed even more in the future. The secondary objective for this thesis was to become familiar with Raspberry Pi 3 Model B-microcomputer, different sensors used with it and the ThingWorx platform. This thesis can also be used by others who are interested in the relating topics and technologies as a reference.

The technology stack introduced in the theoretical section of this thesis is an excellent model for the IoT service design and for the actual implementation of the service as it breaks the IoT concept to more comprehensible chunks. Big Data is a crucial element of successful IoT services, because of its capabilities to handle and analyse vast amounts of data in data flows in effective ways and producing added value to an IoT service. Linked and open data enables new aspects and ways of analysing data in larger projects. As every technology has its challenges and deficiencies, the problems of IoT and Big Data were also addressed and contemplated from a critical point of view in this thesis.

In the design phase, different design documents were used, e.g. a process flowchart and user stories. In the implementation phase, all of the planned sensors were not used, nevertheless, the project was successful. Utilisation of the ThingWorx platform and the Raspberry Pi 3 Model B were effortless. Installation of the distance sensor, coding and testing turned out to be the most problematic aspects of the project. This thesis taught me how to use Raspberry Pi-microcomputer, the ThingWorx platform and increased my understanding of IoT and Big Data. In the future I will continue to learn and experiment with these technologies.

Keywords Internet of Things, Big Data, sensors, IoT dashboard, IoT design

SISÄLLYS

KUVIO- JA TAULUKKOLUETTELO

LIITELUETTELO

KÄSITTEET

AES	Advanced Encryption Standard, lohkosalausmenetelmä
API	Application Programming Interface, ohjelmointirajapinta
IEC	International Electrotechnical Comission, kansainvälinen sähköalan standardointiorganisaatio
IEEE	Institute of Electrical and Electronics Engineers, kansainvälinen tekniikan alan järjestö
IIoT	Industrial Internet of Things, teollinen internet
ISO	International Organization of Standardization, kansainvälinen standardisoimisjärjestö
JSON	JavaScript Object Notation, avoimen standardin yksinkertainen tiedostomuoto tiedonvälitykseen
MEMS	Micro-electromechanical systems, mikrosysteemit
M2M	Machine to Machine, koneiden ja muiden laitteiden välinen tietoliikenne
PCB	Printed Circuit Board, piirilevy
PIR	Passive Infrared, passiivinen infrapuna
P2P	Peer to Peer, vertaisverkko
RAM	Random Access Memory, keskusmuisti/käyttömuisti
REST	Representational state transfer, ohjelmointirajapintojen toteuttamiseen tarkoitettu HTTP-protokollaan perustuva arkkitehtuurimalli
SaaS	Software as a Service, ohjelmiston hankkiminen palveluna
TCP	Transmission Control Protocol, tietoliikenneprotokolla

TLS/SSL	Transport Layer Security/Secure Sockets Layer, salausprotokolla
UDP	User Datagram Protocol, yhteydetön protokolla

1 JOHDANTO

Opinnäytetyön aiheena on esineiden internet, sillä kyseessä oleva tekniikka on ajankohtainen ja monet yritykset ovat ottamassa sitä käyttöön eri prosesseissa. Uskon, että esineiden internet ja varsinkin kerättävän tiedon hyödyntäminen liiketoiminnassa tulevat olemaan arvokkaita tekniikoita IT-alalla. Olen kiinnostunut kyseessä olevasta aiheesta ja siitä, miten tiedon keräämisellä sekä analysoinnilla voidaan luoda uutta arvoa.

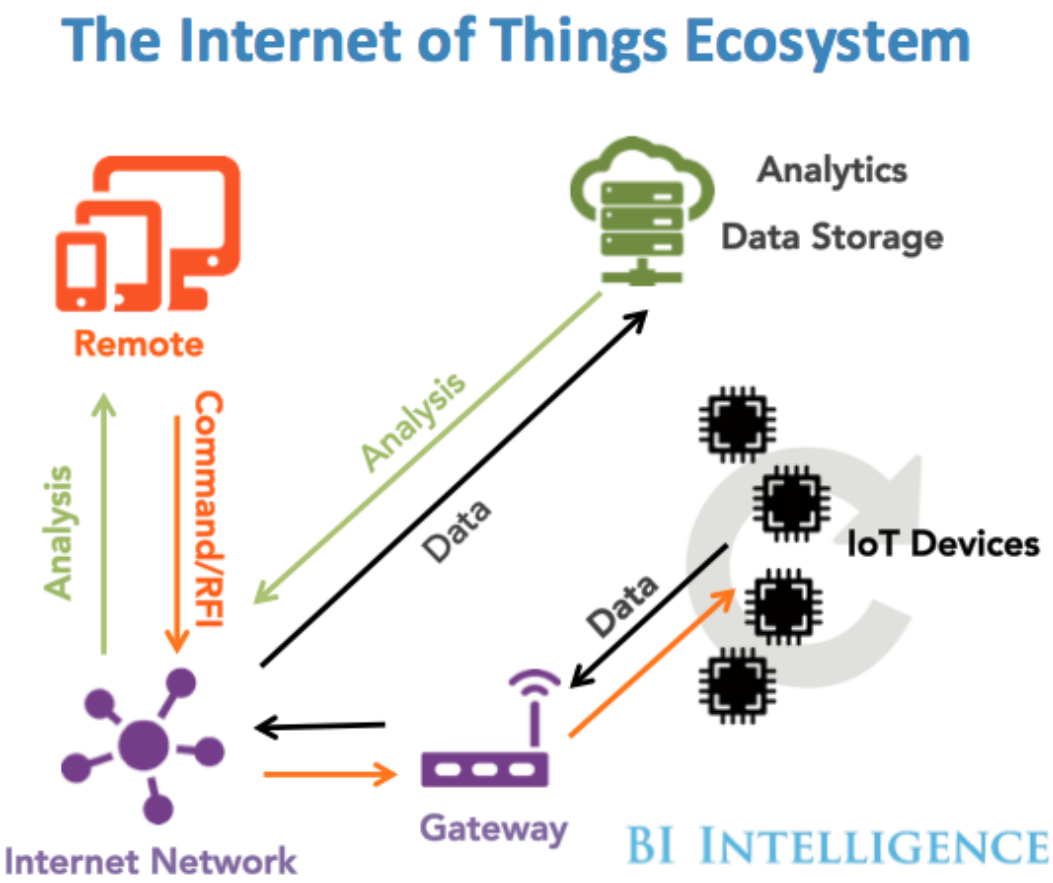
Opinnäytetyön päällimmäisenä tavoitteena on tutustua esineiden internetin perusteisiin sekä Big datan mahdollisuuksiin. Mitä kaikkea sisältyy kyseessä oleviin tekniikkoihin, miten luodaan esineiden internetin palvelu, mistä kannattaa lähteä liikkeelle. Big datasta haluan ymmärtää, mitä se konkreettisesti on ja tutustua Hadoop-tekniikkaan teoriassa. Opinnäytetyö on projektimuotoinen työ, jossa suunnittelen ja toteutan yksinkertaisen esineiden internetin palvelun. Palvelussa käytän samanlaista ThingWorx-alustaa kuin Vaasan ammattikorkeakoulun hankeprojekti (Technobothnia IoT Testing and Learning Hub) ja omaa Raspberry Pi 3 B-mallia sekä kolmea erilaista anturia.

Tämän opinnäytetyön tavoitteena on myös ymmärtää kokonaisuutena esineiden internetin käytännön toteutuksen, erilaiset toteutusmuodot sekä mitä palvelusta voidaan hyötyä. Tarkoituksena on myös ajatella kriittisesti esineiden internetin vajavaisuuksia, yleisiä ja tulevaisuuden ongelmakohtia. Tavoitteena on opetella ThingWorx-alustan käyttämistä sekä mahdollisesti auttaa Vaasan ammattikorkeakoulun hankeprojektin muita jäseniä käyttämään ThingWorx-alustaa helpommin. Opinnäytetyön avulla haluan myös jatkaa tämän aiheen tutustumisessa ja mahdollisesti myös tehdä muita projekteja näihin teknologioihin liittyen.

2 ESINEIDEN INTERNET

Internet on mahdollistanut monia uusia innovaatioita ja tekniikoita, yksi näistä on esineiden internet. Esineiden internet kostuu periaatteessa mistä tahansa laitteesta, anturista tai muusta objektista, joka on yhteydessä internetiin. Esineiden internet-tekniikalla on monia eri nimiä, jotka kaikki tarkoittavat peruskäsitykseltään samaa asiaa, objektin yhteyttä internetiin. Vuonna 2014 arvioitiin yhteydessä olevien esineiden määräksi noin 6 miljardia, vuoteen 2020 mennessä esineiden määrän on arvioitu olevan noin 20–30 miljardia sekä vuoteen 2025 mennessä liiketuotoksi arvioidaan noin 3 biljoonaa dollaria. (Management Today 2014; Vesa 2016.)

Kevin Ashtonin, englantilaisen teknologian edelläkävijän, joka työskenteli MIT:llä (Massachusetts Institute of Technology), sanotaan keksineen termin esineiden internet (The Internet of Things) jo vuonna 1990. Ensimmäisiä oikeita esineiden internetin hyvin mainostettuja tuotteita oli jääkaappi, joka pystyi kertomaan käyttäjälle, mikä tuote on lopussa. (Management Today 2014.) Laitteeseen tai esineeseen kiinnitetty anturi kerää reaaliaikaisesti erilaisia tietoja, kuten esimerkiksi lämpötilaa, kosteutta ja liikettä. Kerätty tieto lähetetään edelleen internetyhteyden avulla joko pilvipalveluun tai muuhun tiedon tallentamispalveluun. Tietoa voidaan sitten analysoida, visualisoida tai käyttää muissa tarkoituksissa. Käyttäjälle voidaan lähettää viesti, jos lämpötila nousee tietyn rajan yli. Esineiden internetiä voidaan hyödyntää jokaisella alalla esimerkiksi parempien asiakaskokemusten luomiseen, tekniikan mahdollistamien etujen parempaan käyttöön tai prosessien laadun muokkauksessa. (Kompella 2015; McEwen & Cassimally 2014, 10–11.) (Kuvio 1)



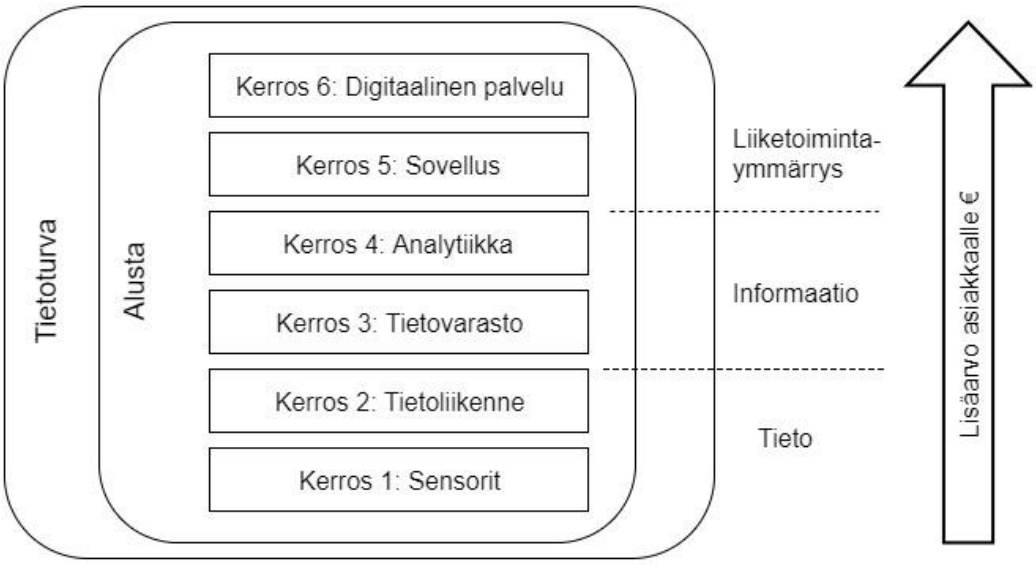
Kuvio 1. Esineiden internetin ekosysteemi (Greenough 2016)

Esineiden internet on myös osakseen liiketoimintamallien murtaja, jonka edellytyksenä on yritysten johdon ja organisoitumisen muutosten hyväksyminen. Osakseen uutta tekniikkaa pidetään uhkana kuin toiset pitävät sitä hyvänä mahdollisuutena ja pystyvät siten yksinkertaisten projektien avulla keräämään kokemuksia sekä hyödyntämään esineiden internetin tuomia hyötyjä omissa prosesseissa. Monet työtehtävät muuttuvat automatisaation takia sekä työturvallisuus kehittyy paremmaksi teollisuudessa. (Collin & Saarelainen 2016, 18–19; Vesa 2016.)

2.1 Teknologiapino

Esineiden internetiä voidaan kuvata selkeästi teknologiapinon avulla (Kuvio 2). Se voi koostua eri tavoin joko neljästä–kuudesta kerroksesta. Tietoliikennekerros koostuu liikkuvasta informaatiosta, joka voi liikkua niin ylös kuin alas. Tietoliikenne kerros sisältää myös käytettävät protokollat ja standardit. Kun halutaan esineiden internetin ratkaisun toimivan mahdollisimman hyvin, pitäisi teknologiapino

hallita hyvin. Hyvä alusta, joka toimii kaikkien kerrosten kanssa, mahdollistaa toimivan ratkaisun luomisen. Unohtamatta tietoturvaa, joka vaikuttaa jokaiseen kerrokseen. (Collin & Saarelainen 2016, 142–144.)



Kuvio 2. Teknologiapino

Antureiden avulla voidaan mitata reaalimaailman tapahtumia ja muuttaa ei-sähköistä informaatiota sähköiseksi informaatioksi. MEMS-anturit ovat todella yleisiä, edullisia, pienikokoisia ja kuluttavat vähän virtaa. Anturit koostuvat prosessoivasta keskusyksiköstä, mikroprosessorista sekä muista komponenteista. Anturityyppejä on monia, ja niillä voidaan mitata esimerkiksi lämpötilaa, ilmanpainetta ja -kosteutta, valoisuutta, biometrisia asioita, ääntä ja liikettä. Anturi voidaan kiinnittää esineeseen tai sijoittaa jonnekin, toimiakseen se kuitenkin tarvitsee jonkinlaisen virransyötön. Anturien virransyöttö voi olla toteutettu muun muassa paristolla, ympäristöstä kerätyllä energialla (energy harvesting) tai PoE-teknologialla (Power-over-Ethernet). (Collin & Saarelainen 2016, 152–162.)

Tietoliikennekerros yksinkertaisesti välittää tietoa eteenpäin, jotta mitattua informaatiota voidaan käyttää edelleen seuraavissa kerroksissa. Tietoliikenne koostuu käytettävistä verkkoratkaisuista, protokollista sekä standardeista, joista kerrotaan tarkemmin kappaleessa 2.2. Seuraava kerros on tietovarasto, joka tarkoittaa anturien informaation tallentamista jonkinlaiseen tietovarastoon. Tietovarasto voi olla

tietokanta, pilvipalvelu tai jokin muu ratkaisu. Jos käytetään tietokantaa, voidaan valita rakenteellinen (SQL) tai ei-rakenteellinen (NoSQL) tietokanta. Rakenteellinen eli strukturoitu tietokanta on perinteinen tietokantatyyppi ja tarkoittaa, että tieto tallentuu ennalta määritellyn rakenteen mukaan. Strukturoitu rakenne sisältää tauluja, sarakkeita ja rivejä. Ei-rakenteellinen tietokanta taas on joustavampi ja skaalautuvampi tietokantatyyppi, jonka tallennettavan tiedon ei tarvitse olla tietyssä muodossa luokiteltuna tai pilkottuna. Ei-rakenteellinen tietokanta voi pitää sisällään sekalaista informaatiota, periaatteessa missä muodossa tahansa, strukturoitua tai ei-strukturoitua tietoa. Ei-rakenteelliseen tietokantaan tarvitaan kuitenkin jonkinlainen rakenne, jonka avulla voidaan päästä käsiksi yksittäiseen tietueeseen. (Collin & Saarelainen 2016, 196–197.) Molemmilla tietokantamalleilla on omat hyvät ja huonot puolensa, jonka perusteella voidaan valita paras vaihtoehto omalle käyttötapaukselle. Tietovarastointiin liittyy todella tiiviisti myös Big data, jota käsitellään erikseen.

Tiedonhallintaan tarvitaan tietokannan lisäksi käytettävä arkkitehtuuri, jolla voidaan toteuttaa tiedon liikennettä ja hallintaa useaan suuntaan. Voidaan valita joko keskitetty tai hajautettu arkkitehtuuri. Keskitetyssä arkkitehtuurissa käytetään yhtä taustajärjestelmää, arkkitehtuuri kerää informaatiota, toteuttaa päätepisteiden (esimerkiksi anturien) hallinnoinnin ja rakentaa keskitettyjä palveluita. Esineiden internetissä keskitetty arkkitehtuuri on yleinen, jonka ominaisuuksiin kuuluvat muun muassa tapahtumien prosessointi ja reaaliaikainen analytiikka. Kehittyneemmissä esineiden internetin ratkaisuissa hajautettu arkkitehtuuri on vaadittu, jolloin keskitetty taustajärjestelmä on käytössä sekä päätepisteet voivat olla yhteydessä toisiinsa suoraan. Hajautetun arkkitehtuurin avulla voidaan saada prosessointiaika ja siitä johtuva viive paremmaksi ja nopeammaksi. Hajautetun arkkitehtuurin myötä voidaan myös toteuttaa P2P-vertaisviestintää, hajautettua auditointia sekä informaation jakamista, joka tarvitsee toimiakseen hyvän, mahdollisimman automatisoidun ja luotettavan yhteyksien muodostuksen (Collin &Saarelainen 2016, 201–202).

Anturien tuottama raakainformaatio ei kerro käyttäjälle mitään. Jotta voitaisiin käyttää mitattuja tietoja hyödyksi, voidaan lisätä analytiikkaa informaation käsitte-

lyyn. Analytiikalla saadaan luotua kerätylle tiedolle arvoa esimerkiksi yrityksen liiketoiminnan analysointityökalussa. Analytiikkaa voidaan käyttää verkon rajalla, informaation kuljetuksen aikana tai sitten kun informaatio on tallennettu tietovarastoon. Analytiikalla voidaan esimerkiksi mahdollistaa poikkeamien reaaliaikaisen tunnistamisen ja ennakoinnin sekä liiketoiminnan kriittisten päätöksien tukemisen. Jotta saataisiin hyödyllisiä tuloksia esineiden internetin informaatiosta, tulee tehdä oikeanlaisia kysymyksiä tai etsiä oikeanlaista informaatiota tietomassasta sekä tarvitaan liiketoiminnan prosessien hallitsemista. Jos informaation kontekstia, käytettäviä järjestelmiä tai päätelaitteita ei tunneta tarpeeksi hyvin, voidaan tehdä vääriä johtopäätöksiä tai tulkita informaatiota virheellisenä korrelaationa tai kausaliteettina. Tällä menetelmällä voidaan testata erilaisia hypoteeseja ja mahdollisesti havaita uusia liiketoimintamahdollisuuksia tai kerätä uusia tietoja tuotteista, prosesseista tai teknologiasta. Analytiikka koostuu algoritmista, joka on tehokas ja automaattinen matemaattiseen kaavaan pohjautuva ohjelmakoodi. Algoritmeja voidaan luoda itse erilaisilla ohjelmointikielillä, avoimen lähdekoodin ohjelmistokirjastojen tai sitten koneoppimiseen tarkoitettujen valmiiden työkalujen avulla. (Collin & Saarelainen 2016, 206–210.)

Sovellus ja digitaalisen palvelun kerrokset ovat yhteydessä toisiinsa. Tarvitaan jonkinlainen sovellus, jonka avulla voidaan esittää informaatiota visuaalisessa muodossa. Tarvitaan myös jonkinlainen palvelu, jossa voidaan käyttää esineiden internetin sovellusta ja yritykset voivat hyötyä tästä. Sovelluksen ohjelmointiin sovelluskehittäjät tarvitsevat ohjelmointirajapinnan eli APIn, kuten esimerkiksi REST-rajapinnan. Suurin osa esineiden internetin sovelluksista ovat SaaS-pohjaisia ratkaisuja eli sovellus on selainpohjainen pilvipalvelu, joka toimii skaalautuvasti tietokoneella sekä mobiililaitteilla. Teknologiapinon ylin kerros, digitaalinen palvelu, on esineiden internetin myytävä ja markkinoitava palvelu. Digitaalinen palvelu koostuu reaaliaikaisuudesta, mobiliteetista, ennakoitavuudesta sekä automaatiosta. (Collin & Saarelainen 2016, 218–224.)

Teknologiapinon alusta toimii perustana kaikille muille kerroksille, sisältäen myös tietoturvan, jonka avulla voidaan koota eri kerrokset yhteen. Alustalla voidaan päästä käsiksi anturiverkkoon ja ratkaista yhteysongelmia sekä testata muidenkin

kerroksien toimivuutta. Alustoille ei ole olemassa tiettyä standardia, joten alustat saattavat olla hajanaisia. Alusta voi olla valmis ratkaisu tai itse tehty avoimen lähdekoodin ratkaisu. Alustojen tarjonta on kattava ja markkinoilla on paljon avoimia sekä kaupallisia alustoja. (Collin & Saarelainen 2016, 228.)

Hyvälle alustalle on kahdeksan vaatimusta, joiden perusteella voidaan päättää omalle ratkaisulle paras vaihtoehto. Anturien yhdistäminen sekä rekisteröinti ovat tietoliikennekerroksen tasolta hyvin tärkeä ominaisuus alustalle, kuin myös protokollien ja dataformaattien tuominen alustalle. Laitehallinta ja tietokanta ovat oleellisia osia alustassa. Laitehallinnalla voidaan mahdollistaa antureiden etähallinta ja päivitys. Käytettävä tietokanta on melko yleisesti pilvipohjainen ratkaisu, sillä tietokannan tulee käydä läpi tietomassoja ja tiedon muoto voi olla hyvin tulkinnanvaraista. Tietokannan tuleekin selviytyä rakenteellisen ja rakenteettoman tiedon läpikäymisestä. Prosessointi, toimintojen hallinta sekä analytiikka muokkaavat tietomassaa selkeämmäksi. Prosessoinnissa ja toimintojen hallinnassa on kyse sääntöjen ja ehtojen asettamisesta, esimerkiksi laukaisin, joka aktivoidaan silloin kun ennalta määrätty poikkeama tapahtuu. Hyvässä alustassa on myös oma analytiikkamoottori, joka mahdollistaa tietomassojen edistyneemmän analysoinnin algoritmien avulla. Pelkät säännöt ja ehdot eivät riitä silloin kun esineiden internetin ratkaisun käyttötapaukset ovat vaativampia. Analytiikassa käytetään hyväksi myös dynaamisia koneoppimiseen pohjautuvia algoritmeja, jolloin analytiikasta saadaan automatisoituneempaa sekä älykkäämpää. Datan visualisointi, kehitys-, hallinta- ja raportointityökalut ovat tiedon visualisoimisen tärkeitä ominaisuuksia. Voidaan käyttää erilaisia mittaristoja, graafisia yhteenvetoja ja kolmiulotteisia kuvia. Alustan kehitystyökalulla voidaan muokkailla visuaalista näkymää helposti sovelluseditorin avulla sekä tehdä prototyyppejä. Hallintatyökalu on osa tietoturvaa eli voidaan antaa eri käyttäjille oikeuksia nähdä tietoja tai muita oikeuksia. Voidaan helposti määritellä, kenellä on oikeus tietoihin sekä antureihin. Raportointityökalulla voidaan tehdä kyselyjä tietomassasta sekä viedä tietoja eri tiedostomuotoihin, esimerkiksi yhteenvetona. Viimeisenä vaatimuksena hyvälle alustalle on tehokkaat ja hyvin määritellyt ulkoiset rajapinnat, useammin halutaan integroida muita ohjelmia alustaan tai alustan tietoja muihin ohjelmiin. Muita seikkoja, joita kannattaa ottaa huomioon alustan valitsemisessa ovat alustan tukemat standardit ja protokollat, miten ja mihin tieto

tallennetaan sekä miten alusta skaalautuu ja vastaa tulevaisuuden tarpeisiin. (Collin & Saarelainen 2016, 230–234.)

Teknologiapinossa tietoturva on kaiken pohjalla, sen tulisi olla osa kaikkia teknologiapinon kerroksia. Yksi riskitekijä on se, että tietoturvaan ei aina panosteta tai varata tarpeeksi resursseja. Joissain perinteisissä teollisuuden hallintajärjestelmissä tietoturvaratkaisu voi olla täysin olematon tai puutteellinen, jolloin hyökkääjien on helppo murtautua ja aiheuttaa vahinkoja erilaisiin järjestelmiin. Yhteys internetiin voi olla epäsuora, esimerkiksi järjestelmä voi olla liitettynä epäsuorasti toiminnanohjausjärjestelmään. Erilaisten järjestelmien haavoittuvuudet ja konfigurointivirheet mahdollistavat järjestelmiin tunkeutumisen. (Collin & Saarelainen 2016, 243–245.)

Ennen esineiden internetin ratkaisun käyttöönottoa tulee hoitaa tietoturva kuntoon. Yrityksessä voidaan laatia verkkotopologiamalli ja kartoittaa kaikki yrityksen käyttämät laitteistot sekä järjestelmät. Voidaan myös tehdä verkkokaavioita tai datavirtojen malleja. Kartoittamisen avulla voidaan havaita tietoturvaan liittyviä epäkohtia sekä riskejä. Verkkoratkaisun tulee olla luotettava, valvottava ja ylläpidettävä. Hyvä esineiden internetin ratkaisu tarvitsee palomuureja, kerroksien suojauksen, tietoturvapolitiikkaa ja oikeuksienhallintaa. Verkkoyhteys tulee myös dokumentoida hyvin, päättää onko verkkoyhteys kaksisuuntainen vai ei, mahdollisimman harvan portin auki pitäminen sekä yhteyspyynnöt tulee olla ennalta määrätyt tietyistä osoitteista. Modernit standardit, protokollat sekä verkkoarkkitehtuuri auttavat luomaan turvallisia ja kestäviä verkkoratkaisuja. (Collin & Saarelainen 2016, 245–247.)

2.2 Protokollat ja standardit

Esineiden internetin toteutettavan palvelun ja käyttäjämäärän mukaan valitaan sopivimmat ratkaisut. Verkkojen kattavuuksia on erilaisia ja verkoista on langallisia ja langattomia muotoja. PAN-verkko (Personal Area Network) on niin sanottu henkilökohtainen verkko, josta tavallisin esimerkki on älypuhelin. Tämän verkkomuodon kantama on lyhyt, metristä kymmeniin metreihin. PAN-verkossa käytetään hyväksi yleisesti Bluetooth-standardia, joka mahdollistaa verkon langattomuuden.

LAN-verkko (Local Area Network) voidaan tehdä langalliseksi, langattomaksi tai näiden yhdistelmäksi. LAN-verkko on yleisin verkkoratkaisu, jota käytetään tyypillisesti kotiverkkojen ratkaisuissa. LAN-verkon kantama on noin 100 metriä. Langaton MAN-verkko (Metropolitan Area Network) voi olla esimerkiksi yhden kaupungin verkkoratkaisu tai verkkoratkaisu, joka koostuu useammasta LAN-verkosta. MAN-verkon kantama on LAN- ja WAN-verkkojen välissä. WAN-verkko (Wide Area Network) on kantamaltaan laajin, sen kantama voi olla jopa kaupungista kaupunkiin tai toisesta maasta toiseen maahan. WAN-verkkoa voidaan käyttää myös lyhyempiinkin kantamiin, kuitenkin ideana on saada eri verkot kommunikoimaan keskenään. Näistä teknologioista on myös langattomia ratkaisuja, joista taulukossa 1 käytettävistä standardeja, tekniikoita, kantamia sekä tiedonsiirronnopeuksia voidaan verrata keskenään. (Isotalo 2017, 11–12; Mata 2015, 6–7; Mitchell 2017.; Romunen & Tikkanen 2010, 12–16)

Taulukko 1. Langattomat verkkotekniikat

	WLAN	*WMAN*	*WPAN*	*WWAN*
Standardit/tekniikat	WiFi, IEEE 802.11	WiMAX, IEEE 802.16	Bluetooth, Zigbee, IEEE 802.15	GSM, 3G, 4G/LTE, 5G
Kantama (km)	0,1 km	< 30 km	< 0,015 km	> 30 km
Tiedonsiirtonopeus	54 Mbps	100 Kbps – 150 Gbps	2 Mbps	56–170 Kbps

Internet-protokollia, IPv4 (Internet Protocol version 4) ja IPv6 (Internet Protocol version 6), tarvitaan esineiden liittämiseen internetiin. Jokainen esine tarvitsee oman IP-osoitteen. Jos esineiden internetin ratkaisujen liittämiseen internetiin käytetään vain IPv4-protokollaa, vastaan tulee ongelmia. Ongelmana on se, että IPv4-protokollaa käytetään eniten (yli 90%) ohjaamaan internetin tietoliikennettä sekä IPv4-protokollan osoitteita on rajallinen määrä. 1990-luvulla IETF (Internet Engineering Task Force) kehitti IPv6-protokollan, joka sopii pienitehoisten anturien

käyttöön sekä ratkaisee IPv4-protokollan IP-osoitteiden määrän ongelman lähitulevaisuudessa. (Lewis 2015; McEwen & Cassimally 2014, 23.)

Tiedonsiirto esineestä eteenpäin internetin kautta tarvitsee protokollia ja standardeja, joiden avulla voidaan nopeasti ja tehokkaasti liittää uusia laitteita järjestelmiin. Protokollia on kehitetty useisiin erilaisiin käyttötarkoituksiin, jolloin protokollan valinnassa tulee ottaa huomioon protokollan kehitys, tietoturva sekä suorituskyky. Useiden erilaisten laitteiden liittäminen yhteen voi olla hankalaa, jos laitteet käyttävät toisistaan erilaisia tai vanhempia protokollia ja standardeja. Tiedonsiirron onnistumiseksi voidaan joutua muuttamaan informaatiota protokollamuuntimilla. Tämä voi vaikuttaa käsitellyn informaation eheyteen sekä luotettavuuteen. Esineiden internetin ongelmaksi on koitunut standardoinnin vähäisyys, hajanaisuus sekä suljetut ratkaisut. Ongelmien ehkäisemiseksi on hankkeita, jotka puuttuvat näihin ongelmiin. Yksi merkittävä esineiden internetiin liittyvä standardointihanke IEEE P2413 on toteutettu IEEE-järjestön toimesta. Hankkeen perimmäistarkoituksena on yhtenäistää arkkitehtuurikehystä yhteensopivuuden ja informaation jakamisen erilaisten järjestelmien, laitteiden ja sovellusten kannalta sekä luoda referenssiarkkitehtuureja eri teknologioille ja toimialoille. Tämän hankkeen yhtenä tavoitteena on saada järjestelmäarkkitehtuurille näkyvyyttä sekä muun muassa saada tukea suorituskyvyn mittaamiselle, yhtenäisen tiedonvälityskerroksen kehittämiselle ja turvallisuuden parantamiselle. (Collin & Saarelainen 2016, 181–184; Mata 2015, 29.)

2.2.1 CoAP ja MQTT

Yleisimpiä tiedonsiirron protokollia, joita käytetään esineiden internetissä, ovat CoAP, MQTT, AMQP, XMPP ja DDS. CoAP (Constrained Application Protocol) on joustava M2M-ratkaisuihin tarkoitettu sovelluskerroksen protokolla, joka käyttää asiakas/palvelin -mallia. CoAP-protokolla on etupäässä yksinkertaisten elektronisten laitteiden hallintaan ja valvontaan internetin kautta. CoAP-protokollassa viestintä ei perustu aikaisemmin luotuun yhteyteen sekä on asynkronista eli ei-reaaliaikaista, joka sisältää HTTPn tutut metodit (GET, PUT, POST ja DELETE). Tietojenvälitys tapahtuu UDP-kuljetuskerroksessa. CoAP-protokollan suurena

etuna on se, että protokollaan on sisäänrakennettu DTLS-pohjainen (Datagram Transport Layer Security) tietoturva. (Collin & Saarelainen 2016, 186–187.)

MQTT (Message Queuing Telemetry Transport) on suosituimpia esineiden internetin protokollia, joka perustuu julkaisija/tilaaja -malliin ja se on kehitetty telemetriadatan keräykseen, etämittaukseen ja -valvontaan soveltuvaksi protokollaksi. Julkaisija/tilaaja -mallissa kummankin osapuolen tulee olla niin sanotusti hereillä, jotta viesti voidaan lähettää ja vastaanottaa. Protokolla on kevyt, yksinkertainen ja tehokas, ja sitä voidaan käyttää monien etälaitteiden tietojen lähettämiseen keskitettyyn IT-palveluun, kuten esimerkiksi pilvipalveluun. MQTT-protokollasta tuli vuonna 2016 ISO/IEC-standardi. MQTT-protokollan puutteita ovat vajavainen virheenkorjaus ja suhteellinen hitaus. MQTT-protokollalla ei voi lähettää viestejä samanaikaisesti eri vastaanottajille ja yhteys puuttuu eri päätelaitteiden välillä. Tietoturvan kannalta MQTT-protokollasta puuttuu täysin salaus, joka kuitenkin voidaan kiertää käyttämällä tietoliikenteessä VPN-tunnelia (Virtual Private Network) ja koska liikennöinti tapahtuu TCP-protokollalla, voidaan vaihtoehtoisesti käyttää vahvaa TLS/SSL-salausta. (Collin & Saarelainen 2016, 187.)

2.2.2 AMQP, XMPP ja DDS

AMQP (Advanced Message Queing Protocol) on vuonna 2003 kehitelty binaarinen protokolla, josta tuli vuonna 2014 kansainvälinen ISO/IEC-standardi. AMQP-protokolla perustuu julkaisija/tilaaja -malliin sekä viestijonoihin. Protokolla on tehokas suurien määrien pienten viestien välittämiseen. Muita etuja kyseisessä protokollassa ovat yksinkertaisuus, varmuus ja äärimmäinen luotettavuus, tietoturva sekä yhteentoimivuus. (Collin & Saarelainen 2016, 188.)

DDS (Data Distribution Service) on todella tehokas M2M-standardi, jonka tavoitteena on yhdistää sellaiset laitteet yhteen, jotka käyttävät reaaliaikaisesti vastaanotettuja viestejä toimintansa ohjaukseen. DDS-standardin etuja ovat yhteyksien laadunhallinta, luotettavuus, nopea monilähetys, monipuolinen tietojen suodattaminen sekä tietojen kohteiden määritteleminen. DDS-standardi mahdollistaa valtavien määrien viestien lähettämisen yhtäaikaisesti monille vastaanottajille. (Collin & Saarelainen 2016, 189.)

XMPP (Extensible Messaging and Presence Protocol) on vuonna 1999 julkaistu protokolla (alkuperäinen nimi Jabber), joka toimii TCP-tietoliikenneprotokollan päällä sekä protokolla on XML-pohjainen (Extensible Markup Language). XMPP-protokollan viestintä perustuu rakenteellisten viestien lähettämiseen kahden tai useamman verkkoyksikön välillä ja siihen, että se käyttää viestien kuljettamiseen toisiinsa liittymättömiä pisteitä. XMPP-protokollan etuja ovat sen yksinkertaisuus (osoiterakenne on kuin sähköpostiosoite, esimerkiksi vastaanottaja@domain.com), tietoturvallisuus sekä skaalautuvuus. Protokolla on hidas ja sitä ei suositella käytettäväksi sulautetuissa järjestelmissä. XMPP-protokollan ongelmana on myös lopullisen viestin suuri tavu määrä. (Collin & Saarelainen 2016, 188–189.)

2.2.3 Wi-Fi, Bluetooth ja ZigBee

Yleisimpiin langattomiin verkkoratkaisuihin kuuluvat Bluetooth, WLAN ja ZigBee. Bluetooth Low Energy eli BLE on vähäkulutteinen Bluetooth-teknologia, jonka mahdollistavat teknologian käyttämät kolikon kokoiset BLE-piirit, esimerkiksi nappipariston kesto BLE-teknologialla on arvioitu helposti kestävän parikin vuotta. Muina etuina voidaan nähdä suora yhteensopivuus mobiililaitteiden tekniikan kanssa, hyvä tietoturva (voidaan käyttää 128-bittistä AES-salausta) sekä sovelluskehityksen helppous. Lyhyen kantaman lisäksi muita puutteita BLE-teknologiassa on sen verkkomalli, joka mahdollistaa tietoliikenteen keskittämisen vain yhteen pisteeseen. Vuonna 2016 Bluetooth-tekniikka sai mesh-ominaisuuden, joka mahdollistaa eri piirien linkittämisen yhteen jopa kolmenkymmenen piirin ketjuun. Piirien linkittäminen voi pidentää Bluetooth-teknologian kantamaa jopa 1,5 kilometriin. (Collin & Saarelainen 2016, 172–173.)

WLAN eli Wi-Fi on nopea sekä langaton standardi, joka käyttää ethernet-verkon topologiaa sekä standardeja. Uusi 802.11ah-standardi on kehitetty esineiden internetin ja M2M käyttöön. 802.11ah-standardi julkaistiin vuonna 2016. WLAN-standardin etuja ovat pitkä kantama, rakenteiden hyvä läpäisykyky, vähäinen virrankulutus, laitteiden välinen viestintä sekä useiden eri laitteiden liitettävyys samaan verkkoon. Toinen uusi standardi 802.11ad (WiGig) on suurilta osin suunnattu enemmänkin kuluttajien käyttöön. Uuden standardin etuna on useiden gigabittien

datanvälitys sekunneissa, mutta toisaalta tämän standardin kantama on vain muutama kymmentä metriä. (Collin & Saarelainen 2016, 171–172.)

ZigBee on vähäkulutteinen sekä edullinen teknologia, joka soveltuu todella hyvin muun muassa sulautettuihin järjestelmiin, teolliseen hallintaan sekä automaatioon. Tämä kommunikaatiostandardi on ZigBee-allianssin luoma, jonka luvataan olevan helppokäyttöisempi sekä edullisempi kuin Bluetooth tai WLAN. ZigBee toimii IEEE 802.15.4 standardilla ja WPAN-verkon tyylisesti. Tämä standardi mahdollistaa useiden laitteiden käyttämisen alhaisella tiedonsiirrolla. ZigBee-standardin kantama on 10–100 metriä. ZigBee-standardin avulla voidaan luoda laajennettuja ZigBee-verkkoja sekä se tukee useita verkkotopologioita. Muita etuja ZigBee-standardissa ovat skaalautuvuus sekä reliabiliteetti. (Agarwal 2018.)

2.2.4 Z-Wawe ja 6LoWPAN

Z-Wawe-standardi on kehitetty vuonna 2005 Z-Wawe allianssin toimesta, standardi käyttää radiotaajuusteknologiaa ja mesh-verkkokonfigurointia. Z-Wawe on avoin standardi ja sen tarkoituksena on vakiintua langattomien kotilaitteiden standardiksi. Radiotaajuusteknologian avulla laitteet pystyvät kommunikoimaan keskenään verkossa sekä laitteiden lisääminen tai poistaminen verkosta on helppoa. Z-Wawe mahdollistaa myös internet portin, VERA:n, käytön, jolla voidaan luoda ja personoida käyttäjän mukaan haluamia erilaisia komentoja, aikaan sidonnaisia asetuksia ja muita toimintoja. Z-Wawe koostuu kahdesta eri tyypistä, hallintalaitteesta (controller) sekä niin sanotusta orjalaitteesta (slave). Orjalaite vastaanottaa ja suorittaa erilaisia komentoja, jotka tulevat hallintalaitteelta. Tämän tyyppisillä laitteilla voi olla jonkinlaista informaatiota eri reiteistä muille laitteille. Hallintalaitteita voi olla yksi tai useampi, kuitenkin vain yhdellä hallintalaitteella voi olla luotettava tieto käytetystä verkon topologiasta. HAN-verkossa (Home Area Network) tulee olla yksi päähallintalaite, jonka avulla orjalaitteet pystytään yhdistämään tähän päähallintalaitteeseen sen oman tunnisteen avulla sekä jokaiselle orjalaitteelle on oma tunniste, jota pystytään hallinnoimaan päähallintalaitteesta. Tämän teknologian etuuksina ovat sen luotettavuus, helppokäyttöisyys sekä yhteentoimivuus. (Mata 2015, 20; Yassein, Mardini & Khalil 2016, 2, 6.)

6LoWPAN on verkkostandardi, joka on ensimmäisiä esineiden internetin tarkoitukseen kehitettyjä standardeja. 6LoWPAN-standardi tukee ainoastaan IPv6-protokollaa, joka mahdollistaa muun muassa osoiteavaruuden laajuuden sekä verkon asennuksen tuen. Standardi on pienitehoinen, langaton ja lyhyenkantaman teknologia. Tätä verkkostandardia käyttävät laitteet, jotka ovat eri verkoissa voivat kommunikoida keskenään. Tämän standardin ongelmana on eri valmistajien käyttämät protokollapinot, jotka eroavat toisten valmistajien ratkaisuista, jolloin yhteensopivuus kärsii. (Collin & Saarelainen 2016, 174–175.)

2.2.5 Mobiiliverkkojen standardit

4G/LTE (Long Term Evolution) on langattoman verkon standardi, joka mahdollistaa esineiden internetin toimivuuden. Vuonna 2016 markkinoille tuli LTE-M tai toisin sanottuna LTE-MTC (Long Term Evolution Machine-Type Communications) vähäkulutteinen ja edullinen standardi M2M-viestintään ja muihin sovelluksiin. Tämän standardin kantaman luvataan olevaan tarpeisiin vastaava. Yhtenä LTE-M standardin etuna on se, että se on yhteensopiva 4G/LTE-verkkoon. Myös vuonna 2016 kehitetty NB-IOT-standardi (NarrowBand IoT) on todella pienikulutteinen standardi. Se toimii erityisesti sellaisten ratkaisujen kanssa, jotka vaativat standardilta kapeaa kaistanleveyttä. Tämän standardin tavoitteina ovat hyvä akunkesto, kuuluvuuden paraneminen sisätiloissa sekä yksittäisen verkkosolun NB-IOT-laitteiden määrä voi olla jopa 50 000. NB-IOT-standardi ei kuitenkaan ole osa 4G/LTE-standardia, jolloin liikennöinti tapahtuukin suojataajuuskaistalla (LTE-operaattorin vierekkäisten taajuuskaistojen välisessä taajuuskaistalla). (Collin & Saarelainen 2016, 176–177.)

Vuoteen 2020 mennessä ennustetaan 5G-teknologian olevan kaupallisessa käytössä. 5G-teknologian tavoitteena on tuoda huomattavia etuja esineiden internetiin yleisesti. 5G-teknologian avulla mahdollistetaan informaation välittämisen nopeammin, vähintään 10 vuoden akunkeston sekä yhden millisekunnin latenssin. Kuitenkin osa mainituista tavoitteista ovat toisensa pois sulkevia ominaisuuksia, sillä samaan aikaan ei voida saada nopeaa ja suurta tiedonvälityskykyä, hyvää akunkestoa sekä pientä latenssia. Tämän takia 5G-teknologia tulee varmimmin jakautumaan

eri versioihin, sen mukaan minkälainen käyttötarkoitus sillä on. (Collin & Saarelainen 2016, 177–178.)

2.2.6 LPWAN

LPWAN (Low Power Wide Area Network) radiostandardille on kehitetty kolme vähävirtaista ja kilpailevaa standardia, jotka ovat LoRa, SigFox ja Weightless. Nämä kaikki standardit ovat juuri esineiden internetiä varten kehitettyjä standardeja, joissa on otettu huomioon todella monia tärkeitä osa-alueita, kuten esimerkiksi tietoturvallisuus, luotettavuus, kantama, edullisuus akunkestoa ja niin edespäin. Ongelmaksi LPWAN-teknologioille kehittyy se, että ne käyttävät lisensoimattomia radiotaajuuksia tietoliikennöintiin, jolloin laitteiden määrän kasvaessa taajuuksilla ruuhkauttaa liikennöinnin. Toisaalta tätä ongelmaa ehkäistään taajuuksien harvemmalla käytöllä ja kun taajuuksia käytetään, informaatiota lähetetään pieninä piikkeinä. Toisena ongelmana on standardien yhteensopimattomuus, jota voidaan korjata LoRa- ja Sigfox-tuen omaavilla yhdistelmämodeemeilla. (Collin & Saarelainen 2016, 178.)

LoRa (tai LoRaWAN, Low Range Wide Area Network) on vähävirtainen radioteknologia, jonka kuuluvuus ja kantama erilaisten rakenteiden läpi on hyvä. Tämän teknologian avulla voidaan toteuttaa yksityisiä verkkoja esimerkiksi tehdasalueille. Parhaiten LoRa-teknologia soveltuu pienten informaatiomäärien välitykseen. Etuja tälle teknologialle on pitkä akunkesto, 15 kilometrin kantama sekä se, että se on kehitetty varta vasten esineiden internetin ratkaisuille. Sigfox-radioteknologia on verrattavissa LoRa-teknologiaan, jonka akun keston sanotaan olevan kahden AA-patterin voimin jopa 20 vuotta. Samoin kuin LoRa, tämäkin teknologia käyttää lisenssivapaita taajuuksia. Tämän teknologian etuja ovat piirien edullisuus, hyvä kantama ja läpäisykyky sekä helppokäyttöisyys. Puutteeksi ilmenee viestin enimmäiskoko (12 tavua) sekä 140 viestin enimmäismäärää päivässä. Suomessa ensimmäinen lisensoitu Sigfox-verkko avattiin vuonna 2016 ja ranskalainen Sigfox-yhtiö operoi ja toimii useissa maissa luoden globaalin verkon. Weightless on vuonna 2012 Weightless SIG-etujärjestön julkaisema tukiaseman ja useiden laitteiden välille tarkoitettu M2M-viestinnän avoin LPWAN-standardi. Tukiasema yhdistetään

internetiin, jonka kantamaksi on ilmoitettu noin 10 kilometriä. Tästä standardista on kaksi versiota, N- ja W-versio. N-versio käyttää yhden gigahertsin ISM-radiotaajuuksia ja W-versio käyttää kapeita televisiolähetysten taajuusalueita. (Collin & Saarelainen 2016, 178–180.)

2.3 Mahdollisuudet ja haasteet

Esineiden internet tuo erilaisia mahdollisuuksia, esimerkkinä teollisen internetin uusien palveluliiketoiminnan ulottuvuuksien kehittäminen. Teollisessa internetissä voidaan luoda uusia palveluita, jotka ovat niin asiakkaalle kuin itse yritykselleki lisäarvoa tuottavia ratkaisuja, kuten esimerkiksi ennakoiva huolto, etäpäivitys, -optimointi ja -käytönvalvonta. (Collin & Saarelainen 2016, 60–63.)

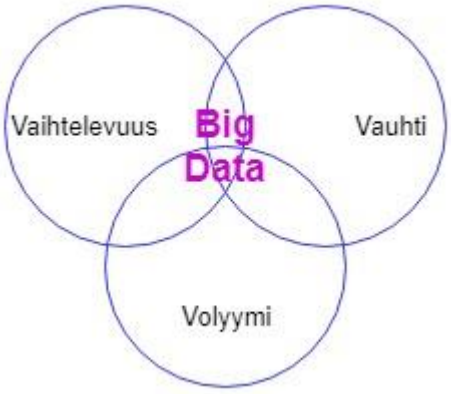
Kuten kaikissa tekniikoissa, myös esineiden internetissä on omat haasteensa ja ongelmansa. Suurimmaksi haasteeksi on noussut pinnalle turvallisuus. Turvallisuudella tarkoitetaan monien asioiden suojaamista, halutaan pitää yritykset, ihmiset ja ihmisten tiedot turvassa. On aina olemassa mahdollisuus, että hakkeri saa käsiinsä yksityistietoja ja näitä voidaan väärinkäyttää. Tähän liittyen kuluttajamarkkinoilla olevien ratkaisujen ja tuotteiden yksityisyys nousee pinnalle. Missä kulkee yksityisyyden raja? Yksityistietojen suojaukseen liittyen uusi EU-asetus tulee voimaan 2018 toukokuussa, joka vaikuttaa henkilötietojen suojaukseen ja yleissääntelyyn. Asetuksen mukaan määritetään henkilötietolaissa eri asioita, kuten esimerkiksi erityisten henkilötietojen määrittelystä ja mikä ylipäätään on henkilötietoa, millä perusteilla ja säännöillä henkilötietoja voidaan käsitellä, minkälaisia oikeuksia henkilöllä on, jonka tietoja käsitellään. Tämä osakseen rajaa sitä, mitä tietoja ihmisistä voidaan kerätä ja se suojaa myös henkilöiden tietoja väärinkäytöltä. Kaikki yritykset tulevat varmasti miettimään, miten tämä uusi asetus tulee vaikuttamaan liiketoimintaan ja muihin prosesseihin. (Elinkeinoelämän keskusliitto 2016; Kompella 2015.)

Myös palveluiden käyttötapausten määritteleminen on ongelmallista, koska yksi palvelu ei sovi kaikkiin yrityksiin tai käyttötapauksiin. Ei voida olettaa, että yksi käyttötapaus tuo heti etuja yritykselle. Käyttötapauksia tulee suunnitella, testata ja sitten vasta laittaa käyttötapaukset oikeaan toimintaan. Tämä tuo kilpailua alalle ja

uusia ideoita, pystytään kehittämään yksilöllisiä ratkaisuja erilaisiin käyttötapauksiin. Kolmantena ongelmana on standardien puute, riippuen alasta joko standardeja on paljon tai sitten niitä ei ole. Vaikka esineiden internet on ollut olemassa jo jonkin aikaa, se ei silti ole lähtenyt nousuun vasta kuin muutama vuosi sitten. Myös kuluttajille on tullut erilaisia palveluita, joita voidaan käyttää mobiililaitteilla tai erillisessä kannettavassa tai puettavassa laitteessa, mutta näitäkään ei ole paljon tai ne eivät suoranaisesti vielä tuo tarpeeksi lisäarvoa käyttäjille. Langattomien yhteyksien hinnat ovat tulleet vuosi vuodelta alemmas sekä niiden saatavuus, toimivuus ja suurien tietomassojen käsitteleminen ovat nykyään helpompaa sekä tehokkaampaa toteuttaa. Tämä mahdollistaa ubiikin internetin hyödyntämisen tehokkaammin kuin aikaisemmin ja myös muiden teknologioiden käyttämisen yhdessä esineiden internetin kanssa. (Kompella 2015; Niemann-Ross 2017; McEwen & Cassimally 2014, 10, 13–14.)

3 BIG DATA

Big data käsite nousi pintaan vuonna 2005 ja vuonna 2011 se nousi suosioon. Big data tarkoittaa informaatiota, jonka määrä kasvaa paljon sekä nopeasti. Big dataan liittyy kolme eri asiaa: volyymi, vaihtelevuus sekä vauhti (Kuvio 3). Volyymillä tarkoitetaan informaation suurta määrää, vaikka Big datan suora suomennos tarkoittaakin informaation paljoutta, siihen kuuluu paljon muutakin kuin vain sen kvantitatiivinen ominaisuus. Tulevaisuudessa digitaalisen informaation määrä maailmassa tulee olemaan valtava. On arvioitu, että vuoteen 2030 mennessä informaation määrä tulee olemaan noin jottatavun (YB) verran eli tuhat tsettatavua (ZB) (vrt. vuonna 2014 informaation määrän oli noin 5 tsettatavua). Vaihtelevuus tarkoittaa informaation laatua eli informaatio voi olla strukturoitua, puolistrukturoitua tai strukturoimatonta. Strukturoitua ja strukturoimatonta informaatiota käsiteltiin jo aikaisemmin kappaleessa 2.1 ja tarkoittaa tässäkin samaa. Puolistrukturoitua informaatiota voi olla esimerkiksi jonkinlainen metadatatieto äänianturin lähettämästä äänitallenteesta (äänitiedoston tunniste, aika, anturin tunniste, tiedoston kokonaiskoko jne.). Vauhti yksinkertaisesti viittaa siihen, kuinka nopeasti informaatiota tulee tai siihen miten nopeasti uuteen informaatioon ja sen luomaan paineeseen reagoidaan. Vauhdilla voi olla suurikin merkitys siihen, miten nopeasti saadaan tehtyä erilaisia ratkaisuja tai seulottua sekä käsiteltyä informaatiota läpi. Jos tarkoituksena on reagoida nopeasti, informaatiota ei kannata tallentaa ensin vaan se kannattaa käsitellä ns. raakainformaationa ja tämän jälkeen vasta tallentaa se tietovarastoon. Big dataa voidaan määritellä näillä kolmella termillä ja mitä paremmin kerättävä informaatio täyttää nämä termit, sen helpommin voidaan sanoa kerättävää informaatiota Big dataksi. (Salo 2014, 26–28.)



Kuvio 3. Big datan ominaisuudet

Informaatiota, voidaan myös jaotella sen mukaan, miten se mahdollisesti liikkuu. Tyypillisimmät käytetyt termit ovat vertauskuvainnollisesti luonnontieteestä meri tai järvi sekä joki. Merenä tai järvenä kuvaava informaatio on jonkinlaisessa tietovarastossa oleva informaatio. Termi kuvastaa sitä, että informaatio on paremmin hallittavissa oleva sekä sitä voidaan niin sanotusti louhia palasiksi. Liikkuvaa informaatiota voidaan kuvailla jokena, joka soljuu eteenpäin omalla jatkuvalla virrallaan, esimerkkinä anturien tuottama jatkuva informaatio. (Salo 2014, 28.)

Silti Big data käsite on hieman abstrakti, vaikka käsitettä voidaankin avata ja selventää aikaisempien esimerkkien avulla. Big dataa voidaan selventää sillä, että se on pilvipalveluihin verrattava kattokäsite informaation havainto sekä ratkaisumalli. Big datan yleisimpiä selvennyksiä ovat informaation kasvuvauhti sekä sen paljous, jota kerääntyy koko ajan isolla vauhdilla. Ja toinen selvennys käsitteelle on se, että Big data ratkaisuna tai ratkaisumallina pyrkii vastaamaan siihen, miten voidaan järkevästi käsitellä ja reagoida valtaviin määriin informaatiota. Ratkaisu tai ratkaisumalli voi olla esimerkiksi konkreettinen palvelu, tuote, tekniikka tai prosessinmuutos. Tämän hetken haasteena on se, että informaatiota kerääntyy jo nyt valtavat määrät. Nykyhetken ratkaisut eivät pysty vastaamaan Big datan luomiin haasteisiin ja eivät ole optimoitu tällaisia varten. Koska informaatiota tulee koko ajan, kaikkea ei voida tallentaa, saati sitten edes analysoida. Big datan yhtenä keskeisenä osana on tilastomatematiikan hyödyntäminen ja ymmärtäminen, jolloin voidaan ideaalisesti saada automaattisesti ennusteita, päätöksiä ja todellista arvoa valtavalle määrälle informaatiota sekä kasvattaa ymmärrystä. (Salo 2014, 30–31.)

3.1 Hadoop

Hadoop on vuonna 2007 aloitettu avoimen lähdekoodin ohjelmistoprojekti, jonka kehittäjänä toimi Doug Cutting. Hadoopin inspiraationa toimi Googlen GFS-tiedostojärjestelmä (Google File System) sekä MapReduce-tekniikka, jolla analysoidaan hajautuneesti varastoituja tietoja. Hadoop toimii omalla HDFS (Hadoop Distributed File System) tiedostojärjestelmällä sekä MapReducella. Hadoop on keskeinen tekniikka Big datassa ja siihen liittyen on luotu paljon uutta liiketoimintaa. Hadoopilla voidaan luoda palvelinklustereita eli voidaan luoda loogisia kokonaisuuksia virtualisoiduista tai fyysisistä palvelimista. Tällaisen klusterin etuina ovat kannattavuus, toimintavarmuus sekä skaalautuvuus. Kannattavuudella tarkoitetaan tässä tapauksessa sitä, että Hadoopissa ei ole lisenssimaksuja sekä se on riippumaton laitetoimittajan ratkaisuista. Hadoopin toimintavarmuus on toteutettu klusterin toimintalogiikalla. Käytetty toimintalogiikka pohjautuu hajauttamiseen, joka mahdollistaa klusterin vikasietoisuuden. Vaikka jokin laite vikaantuisikin, niin se ei vaikuta klusterin toimivuuteen tai johda tiedonmenetykseen. Skaalautuvuus Hadoopissa on hyvin tärkeä ominaisuus, joka mahdollistaa helpon klusterin koon laajentamisen tarpeita vaativaksi sekä mahdollisesti erityyppisten ratkaisujen käyttämisen, esimerkiksi Hadoopin asentaminen kannettavan tietokoneen virtuaalikoneelle. Tulee kuitenkin ottaa huomioon, että Hadoop ei ota kantaa tiedon laatuun tai missä muodossa tiedot ovat tiedostojärjestelmässä. (Salo 2014, 72–74.)

Hadoop toimiikin parhaiten suurten tietomäärien kanssa. Klusterin hyödyt tulevat sitä paremmin esille, mitä suuremmasta tietomäärästä on kyse. Tulee kuitenkin ymmärtää, että myös Hadoopissa tulee raja vastaan, kuinka suuria tietomääriä se pystyy käsittelemään. Hadoop on kehittynyt paljon vuosien varrella ja on hyvin tunnettu osa Big data ratkaisuja. Hadoopille vaihtoehtoisia tekniikoita ovat Spark ja Tez. (Salo 2014, 83, 87–88.)

3.2 Linkitetty ja avoin data

Avoin data on osa Big dataa, se liittyy myös linkitettyyn dataan. Voidaan käyttää BOLD lyhennettä, joka koostuu sanoista Big and Open Linked Data. Linkitetty data liittyy strukturoidun sekä koneluettavan tiedon yhdistämiseen, jota voidaan siten

etsiä semanttisesti. BOLD käsitteellä viitataan siihen, että ainoastaan Big data, joka tuotetaan esimerkiksi organisaation sisällä, ei tuota informaatiolle monipuolista lisäarvoa vaan tämän toteutumiseksi tarvitaan myös muita hyödyllisiä informaation lähteitä. Avointa dataa voidaan käyttää erilaisiin käyttötarkoituksiin vapaasti. Avoimessa datassa siis avataan erilaisia rajapintoja esimerkiksi julkishallinnon tietovarastoihin ja myös muiden toimijoiden tietovarastoihin. Tämä käytännössä tapahtuu siten, että voidaan tehdä niin sanottua vaihtokauppaa tietovarastoilla tai avata rajapintoja datamarkkinapaikkojen välityksellä. Datamarkkinapaikat tarjoavat käytettäviä tietovarastoja pilvessä joko vastikkeellisesti tai vastikkeetta, jota voidaan laskuttaa esimerkiksi transaktioiden tai tietomäärän mukaan. (Saxena 2017, 10–11; Salo 2014, 43.)

Avoimen julkisen tiedon ominaisuus on tiedon helppo saatavuus, joka voidaan toteuttaa portaalilla, jossa listataan avatut tietovarastot. Sujuvuuden vuoksi tulisi myös asettaa standardoituja formaatteja, rajapintoja sekä käyttöehtoja. Toinen ominaisuus on informaation aktiivinen keräys tietovarastoihin. Suurimman osan tiedoista keräävät yksityiset toimijat, mutta kaikkea eivät hekään kerää. Verratessa kustannuksia ja aikaa mitä tietovaraston käyttöönottoon sekä taloudellisen hyödyn kertymiseen kuluu, tiedon kerääminen ei välttämättä ole kannattavaa yrityksille, vaikka tiedolla olisikin lisäarvoa. Kun tietovarastot ovat yhteiskunnan tuottamia, yritykset pystyvät hyödyntämään näitä paremmin luodakseen uusia oivalluksia ja lisätäkseen ymmärrystä. Yhteiskunnan tuottamien tietovarastojen esimerkkejä ovat väestölliset tilastot, ilmasto- ja säätiedot sekä tieteellisten tutkimusten tuottamat tiedot. Uusia sovellusalueita saadaan aikaiseksi avoimen datan innovatiivisilla käyttötarkoituksilla, jolloin huomataan aivan uusia mahdollisuuksia joita ei ennen tiedonkeräämistä voitu edes kuvitella. Parhaiten saadaan uusia liiketoiminnan mahdollisuuksia, kun tietoja kerätään paljon, erilaisissa muodoissa sekä liittyen reaalimaailman erilaisiin ilmiöihin. Big datan, avoimen datan sekä linkitetyn datan todellinen hyödyntäminen on uusi ja nouseva ilmiö. (Saxena 2017, 10–12; Salo 2014, 43–45.)

Avoimen datan luoma lisäarvo globaalisti on todella suuri, sen on arvioitu tuottavan lisäarvoa noin 3000–5000 miljardia dollaria. Tämän arvion mukaan voidaan sanoa,

että avoin ja Big data on arvokasta. Avoimen datan huolenaiheina ja ongelmina ovat tietovarastojen hajanaisuus, visioiden ja koordinaation puute sekä tietoturva- ja yksityisyysongelmat. Yhtenä huolena on myös julkishallinnon puolella kerättyjen tietojen jakaminen ilmaiseksi, koska siihen on käytetty julkishallinnon omia varoja. (Salo 2014, 168–170.)

3.3 Big datan ja pilvipalveluiden huolet sekä tietoturva

Big dataan liittyvän tietoturvan alle lukeutuvat juridiset oikeudet ja velvoitteet liittyen tietojen keräämiseen, tietojen käyttöoikeuksiin, palveluiden luotettavuuteen sekä jatkuvuuteen. Yleisimmät tietoturvaan liittyvät tapaukset liittyvät yksilön tietojen prosessointiin, oli se sitten profilointi-, markkinointi- tai seurantatarkoituksessa ja kaikille tämän kaltaiset tapaukset ovatkin helpoiten ymmärrettävissä. Kuitenkin Big datan tietoturva ei ainoastaan koostu näistä aiheista. Tietomurrot, urkinta sekä tietojen väärinkäyttö ovat yleisimpiä huolenaiheita, vaikka raakainformaatio ei välttämättä ole minkään arvoista, yritys tai palveluntarjoaja voi silti kärsiä tästä johtuvasta imago- tai liiketoimintatappioista. Kun tietoja on prosessoitu, analysoitu ja yhdistelty, tämän tuottaman arvon tai käytettyjen algoritmien menettäminen voi olla hyvinkin kallista. Tällaisten tilanteiden ennakointiin kuuluu tietojen suojaaminen koko sen käyttöelinkaaren sekä käytön jälkeisenä aikana. Tietojen saatavuus- ja pysyvyyshuolet ovat mahdollisia, jos pilvipalveluun tai muuhun tietovarastoon ei päästä käsiksi. Tällöin mahdollisen katkon aikana liiketaloudelliset tappiot perustuvat täysin tiedon tärkeysasteeseen. Tähän ongelmaan ratkaisuna on tiedon hajauttaminen useampaan paikkaan. Tietojen väärentäminen on myös ongelma ja se muokkaa analyysin tuloksia mahdollisesti järjestelmään murtautuneen tahon haluamaan suuntaan. Kun käytettävä tieto koostuu useammasta eri lähteestä ja yrityksen ulkoisesta tietolähteestä, hyökkäyksien tai väärennöksien havaitseminen muuttuu vaikeammaksi sekä kompleksisemmaksi. Big datan avulla voidaan havaita tähän liittyen trendien äkillisiä muutoksia tai poikkeamia nopeasti. (Salo 2014, 50–52, 105–107, 109.)

Fyysisten tietoturvakomponenttien käyttäminen voi tuoda mukanaan haasteita luku-, kirjoitus- tai laskentatehtäviin. Järjestelmissä joissa suorituskyky on optimoitu, useimmiten käytettävä tieto ei ole mitenkään salattu järjestelmän levyillä, vasteaikojen vaatimusten mukaan salaus ei ole mahdollista tai yksittäisiä tietoja ei voida kategorisoida sisällön perusteella erilaisiin tietoturvatasoihin. Tietoturvallisuutta voidaan parantaa näissä tapauksissa reaaliaikaisella tunnistamisella, koneoppimisella sekä sallittujen toimien asettamisella, jolloin poikkeukset voidaan havaita nopeasti sekä mahdollisesti myös reagoida poikkeamiin automaattisesti, jolloin käyttäjälle jää tehtäväksi raporttien sekä tehtyjen toimenpiteiden seuranta. Pilvipalveluita käytettäessä voidaan vaikuttaa tietoturvatasoon siirtymällä yksityisiin palveluihin tai täysin omassa hallinnassa oleviin laitteisiin. Voidaan myös kartoittaa miten kriittistä ja minkälaisia riskejä kuuluu siihen, jos tiedot joutuvat tietoturvauhan alle. (Salo 2014, 52–53, 105.)

Myös Big datan parissa yksityisyyteen liittyvät huolet ovat pinnalla, sillä pelkän päätelaitteen avulla voidaan luoda käyttöprofiili ja kun tämä profiili linkittyy reaalimaailmassa yksityishenkilöön, saadaan luotua edelleen erilaisia mahdollisuuksia tietojen käytölle sekä analysoinnille. Käyttäjä ei mahdollisesti edes tiedä miten laajasti tietoja kerätään ja miten yksityiskohtainen käyttöprofiili saattaa olla. Myös verkossa tapahtuva käyttäytyminen luo käyttöprofiilia ja profiiliin liittyvät tiedot saattavat vuotaa lisäpalveluiden tarjoajille, yleisemmin sosiaalisen median palveluille. (Salo 2014, 54.)

4 THINGWORX

ThingWorx on esineiden internetin yksi monista alusta teknologioista. ThingWorx kuuluu PTC yrityksen tuotteisiin. PTC on vuonna 1985 perustettu maailmanlaajuinen ohjelmistoyritys, joka toimii 30 maassa. PTC tarjoaa erilaisia ratkaisuja esineiden internetin lisäksi esimerkiksi tietokoneavusteisen suunnittelun (CAD), tuotteen elinkaaren hallinnoinnin (PLM), palvelun elinkaaren hallinnoinnin (SLM), lisätyn todellisuuden (AR) ja älykkään teollisuuden (Industrie 4.0) ratkaisuja. PTCn esineiden internetin ratkaisun lupauksena on mahdollistaa teollisen internetin (IIoT) ohjelmien luomisen tehokkaasti, jonka avulla saadaan uutta arvoa digitaalisesta sekä fyysisestä maailmasta. (PTC 2017a.)

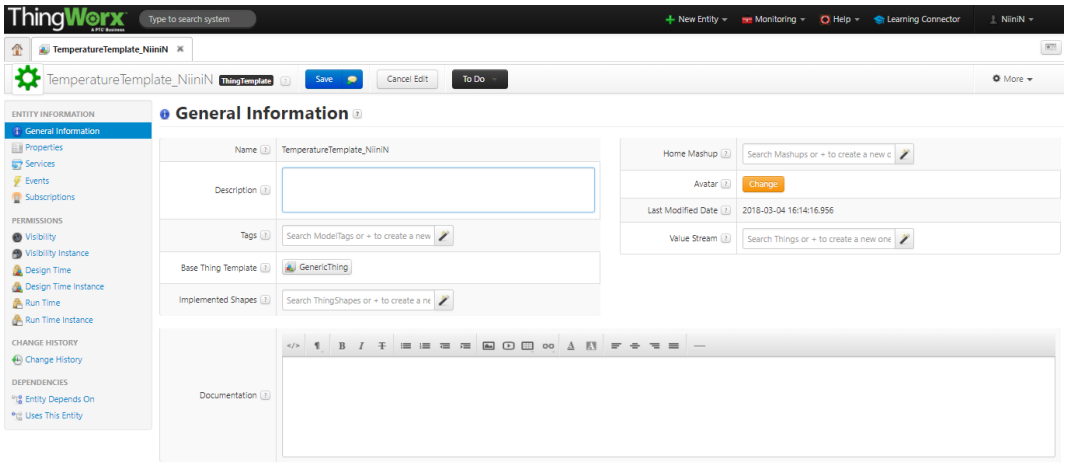
Vuonna 2014 PTC osti ThingWorx-alustan ja mahdollistaa tehokkaan korkea-arvoisten IoT-sovellusten luomisen, joka tukee asiakkaiden toimintasuunnitelmissa sekä palveluiden tarjoamisessa. IoT-sovelluksella voidaan toteuttaa esimerkiksi ennakoivaa kunnossapitoa tai järjestelmän valvontaa. PTCn portfoliopalveluita (PLM ja SLM) voidaan hyödyntää erilaisissa ratkaisuissa useilla toimialoilla, jotka haluavat pohjimmiltaan hyödyntää IoT-teknologian mahdollisuuksia. PTC tarjoaa asiakkailleen tavan luoda yhteyden turvallisesti älykkäisiin tuotteisiin ja alustan jolla voidaan helposti hallita ja ylläpitää näitä sekä luoda parempaa ymmärrystä tuotteiden toiminnasta, käytöstä sekä luotettavuudesta. Vuonna 2015 Elisa otti käyttöön ThingWorx-alustateknologian, jonka tarkoituksena on Elisa IoT-palvelun tuominen markkinoille Suomeen sekä Viroon. ThingWorx-teknologian avulla voidaan tuoda helppoja ratkaisumalleja yritysten liiketoimintaprosessien innovatiivisille kokeiluille kustannustehokkaammin sekä vähemmällä riskillä. ThingWorx on myös oivallinen lähtökohta yrityksille, jotka haluavat ymmärtää älykkyyden ja yhteyden lisäämisen arvon erilaisiin tuotteisiin ja muihin palveluihin. (Professional Services Close – Up 2014a, 2015b.)

4.1 Integraatio ja tiedon käsittely ThingWorx Composerissa

ThingWorx-alustalle voidaan yhdistää esineitä monella eri tapaa esimerkiksi laitepilven (device cloud), APIn, EMSn (Edge MicroServer) ja SDK -pakettien (Soft-

ware Development Kit) avulla. Laitepilven kautta voidaan yhdistää laitteita alustalle Azure IoT Hub tai AWS (Amazon Web Services) IoT-palveluista. API-rajapintojen avulla voidaan päästä käsiksi eri toimintoihin, esineisiin ja ominaisuuksiin. API-rajapintojen käyttäminen on helppoa ja nopeaa. SDK-paketteja on olemassa C, .Net, Java, iOS ja Android laitteille ja ohjelmistoille. SDK ja EMS käyttävät ThingWorxin omaa AlwaysOn-protokollaa. ThingWorxilla on myös erillinen ratkaisu teollisen internetin laitteiden yhdistämiseen alustaan. (PTC 2017b.)

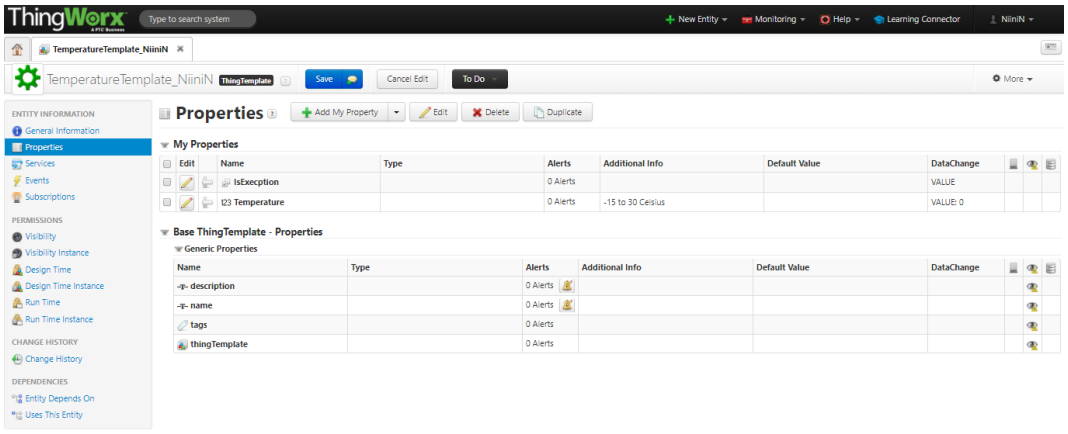
Mallipohjan avulla voidaan luoda useita samantyyppisiä esineitä tehokkaasti, jotka perivät samat ominaisuudet mallipohjasta (Kuvio 4). Mallipohjien avulla voidaan määritellä perusominaisuuksia, joita halutaan esineillä olevan, kuten esimerkiksi lämpötila-anturien ominaisuutena voisi olla lämpötila (Kuvio 5). Mallipohja itse tulee periytyä yleiseen esineeseen (GenericThing), jotta se saa muut olennaiset ThingWorx esineen ominaisuudet. (IoT University 2018.)



Kuvio 4. Esineen mallipohjan perustiedot

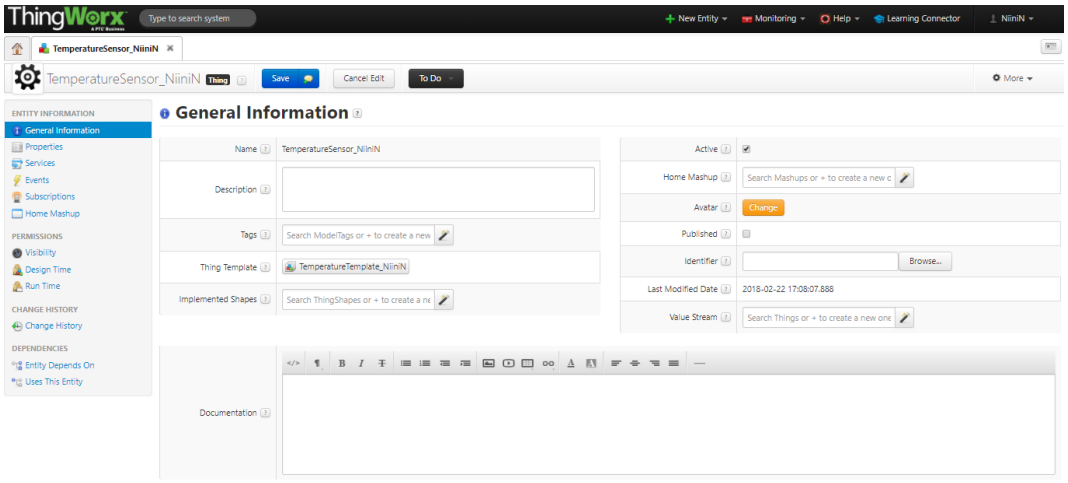
Mallipohjassa voidaan lisätä myös muita ominaisuuksia ja tapahtumia. Ominaisuuksille voidaan antaa tietotyyppi esimerkiksi string, datetime, long, boolean, blob jne., yksikkö, minimi- ja maksimiarvot sekä alkuarvo. Mallipohjiin voidaan lisätä metodien tapaisia tapahtumia JavaScript-ohjelmointikielellä. Ohjelmointi on helppoa myös sellaiselle, joka ei ole aikaisemmin ohjelmoinut, koska tapahtumia voidaan helposti luoda valmiiden pienten koodikokonaisuuksien avulla. ThingWorx

esineellä voi olla erilaisia tapahtumia, jotka voivat toteuttaa jonkun toiminnon. Tilaukset ovat myös toimintoja, jotka laukaistaan tietyn tapahtuman yhteydessä. (IoT University 2018.)



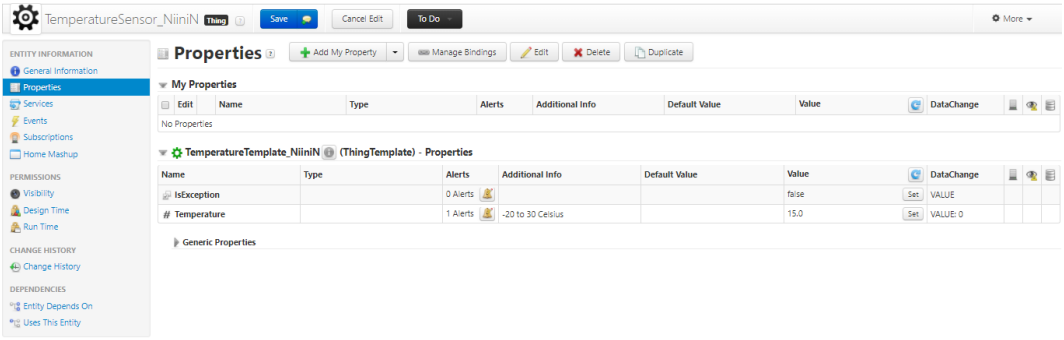
Kuvio 5. Esineen mallipohjan ominaisuudet

Mallipohjien käyttämisellä säästytään turhalta ja toistuvalta työltä, vaikka näiden luominen ei olekaan pakollista. Kun mallipohja on liitetty tiettyyn esineeseen, sitä ei voida poistaa, vaan joudutaan ensin poistamaan kaikki esineet, jotka ovat perineet mallipohjan ominaisuudet. Mallipohjaan voidaan myös lisätä tilauksien tapaisia hälytyksiä, joita voidaan asettaa myös muilla tasoilla. Hälytyksen tyyppejä ovat muun muassa yhtä kuin ja eri suuri, tämän lisäksi hälytykselle annetaan nimi ja arvo. (IoT University 2018.)



Kuvio 6. Esineen perustiedot

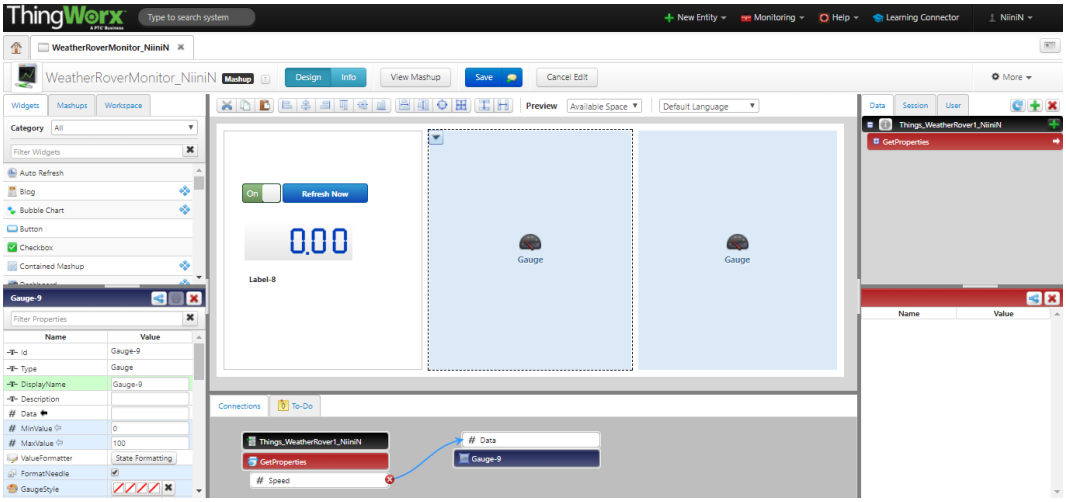
Esine tässä tapauksessa tarkoittaa digitaalista esitystä fyysisestä asiasta. Mallipohjassa esiteltyjä ominaisuuksia, tapahtumia ja tilauksia voidaan myös määritellä esineiden tasolla (Kuvio 6). Tilausten avulla voidaan analysoida esineitä ja ominaisuuksia, voidaan esimerkiksi havaita, milloin lämpötila nousee yli asetetun rajan ja näyttää käyttäjälle varoitusmerkki. Analysointi voi olla kuvailevaa eli yksinkertaisuudessaan annetaan numeeriselle arvolle tarkoitus tai paljon monimutkaisempaa erilaisten tapahtumien ja tilauksien ansiosta. Tilausten luomisessa valitaan, minkälainen tapahtuma on kyseessä, mihin ominaisuuteen tilauksen laukaiseminen kohdistuu, esimerkiksi lämpötila ja esine johon tilaus liittyy. Myös esineen tasolla voidaan luoda lisää sille kuuluvia ominaisuuksia ja perustietoja, kuten esimerkiksi nimikkeitä ja kuvake (Kuvio 7). (IoT University 2018.)



Kuvio 7. Esineen ominaisuudet

4.2 Tiedon visualisointi ThingWorx Composerissa

ThingWorx Composer-alustalla voidaan luoda esineiden tietoja näyttäviä visuaalisia esityksiä. Visuaalisia esityksiä on helppo toteuttaa ThingWorx Composer-alustalta löytyvän editorin avulla, jossa voidaan vetää erilaisia elementtejä sivulle ja lisätä näihin älykkyyttä eri tapahtumien ja ominaisuuksien avulla. Esitystä luodessa voidaan valita, onko esitys responsiivinen vai staattinen sekä minkälaista esitystä luodaan. (IoT University 2018.)



Kuvio 8. ThingWorx Composer-alustan visualisointi editori

ThingWorx Composerin editorissa on kaksi erillistä näkymää, design ja info. Design on itse visualisoinnin editorinäkymä ja info sisältää visualisoinnin perustiedot. Kuviossa 8 ThingWorx Composer on design-näkymässä ja näkymästä on valittu keskimmäinen mittari, jonka alapuolella voidaan nähdä, minkälaisia yhteyksiä kyseiseen elementtiin kuuluu ja mihin esineeseen sekä ominaisuuteen elementti on

liitetty. Oikealla yläkulmasta olevasta ikkunasta voidaan valita erilaisia metodeja, joilla voidaan esimerkiksi hakea ominaisuuksien arvot esineistä. Vasemmassa alanurkassa on valitun elementin ominaisuudet, joita voidaan muokata vapaasti. Vasemmassa yläkulmasta löytyy Widget-välilehti, jossa on kaikki editorissa käytettävät elementit. Elementtityyppejä ovat tietoa näyttäviä, visualisaation rakenteeseen ja käyttäjän vuorovaikutukseen vaikuttavia elementtejä. Mashups-välilehdeltä löytyy kaikki olemassa olevat ThingWorx Composeriin tehdyt visuaaliset esitykset ja Workspace-välilehdelle kootaan kaikki sillä hetkellä avatun esityksen käytetyt elementit puurakenteeseen. Visualisointia voidaan testata erillisessä selainvälilehdessä. (IoT University 2018.)

5 PALVELUN SUUNNITTELU

Projektin toteutuksen helpottamiseksi tein erilaisia suunnitelmia ja suunnitteluun liittyviä dokumentteja, kuten käyttäjätarinoita, prosessikaavio sekä tietokannan relaatiokaavio. Ajattelin käyttää ketterän projektimallin erilaisia dokumentteja sekä tapoja palvelun suunnittelussa sekä käytännön toteutuksessa. Toteutuksessa kaikki suunnitellut asiat eivät välttämättä tule toteutumaan, kuten esimerkiksi kaikkia suunniteltuja antureita ei välttämättä tulla ottamaan käyttöön.

Käyttäjätarinoita käytetään yleisesti ketterissä projekteissa, ja ne ovatkin informatiivisia, koska saadaan selville kuka käyttää kyseistä palvelua, mitä halutaan tehdä ja minkälaista lisäarvoa käyttäjä haluaa palvelusta (Nazzaro & Suscheck 2010). Tämän palvelun käyttäjänä voi toimia melkeinpä kuka tahansa, joka on kiinnostunut tietämään käyttäjätarinoissa mainittavista asioista reaaliaikaisesti visuaalista alustaa käyttäen, kuten seurata tietoja lämpötilasta, liikkeestä sekä valoisuudesta ja halutessaan saada viestin puhelimeen tai sähköpostiin näiden arvojen mennessä asetettujen rajojen yli (Kuvio 9). Suunnitellussa palvelussa on viisi erilaista arvoa, josta voidaan saada sekä kerätä tietoja. Ja käytettäviä antureita olisi tällöin neljä, koska lämpötilaa ja ilmankosteutta voidaan mitata yhdellä anturilla. Kuten käyttäjätarinoissakin mainitaan, että anturit ja muut toteutukseen liittyvät elementit tulisi asettaa sellaiselle paikalle, jossa olisi mahdollisimman vähän häiriötekijöitä, joten arvoja voidaan mitata luotettavasti.

Lämpötilan tarkkailu

Käyttäjänä haluan tietää (huoneen) sisälämpötilan, koska haluan tarkkailla huoneen lämpötilaa ja havaita mahdollisesti erilaisia poikkeamia.

- Sensorin tulee sijoittaa sellaiseen paikkaan, jossa voidaan mitata lämpötilaa ilman häiriötekijöitä.
- Sensorin tuottamia arvoja lähetetään edelleen alustalle, joka näyttää käyttäjälle arvot numeerisena omassa elementissään.

Liikkeen tunnistus

Käyttäjänä haluan saada tiedon jos huoneessa liikkuu joku (kun en ole itse kotona), koska haluan turvata kotini tunkeilijoilta ja tietää esim. lemmikkieni liikkeestä sensorien lähellä.

- Tunnistin tulee asettaa sillä tavalla, että saadaan tietää luotettavasti liikkeestä.
- Tunnistimia voi mahdollisesti olla useampiakin.
- Kun tunniste huomaa liikettä, arvo näytetään alustalla sille elementille tarkoitetulla tavalla.
- Tulee tietää pitääkö arvon muuttuessa ilmoittaa käyttäjälle.

Reaaliaikainen tilanteen visuaalinen seuranta

Käyttäjänä haluan analysoida tilannetta reaaliaikaisesti ja erilaisten visuaalisten elementtien avulla, koska haluan ymmärtää helposti mitä kotonani tapahtuu juuri sillä hetkellä.

- Alustalla täytyy olla eri sensoreille omat visuaaliset elementit sekä näihin liittyvät koodit esim. muutoksien varalta tai rajojen ylittämiselle.
- Helppokäyttöinen liittymä asiakkaan käyttöön.
- Erilaisia merkkejä erilaisille tilanteille.
- Sensorien lähettämien tietojen saatavuus ja luotettavuus.
- Tietojen analysointi tehdään ennen visualisointia

Etäisyystunnistus

Käyttäjänä haluan tietää lähestyykö joku sensoria, koska haluan tietää kuinka lähelle joku pääsee sensoreita.

- Sensori tulee kohdistaa oikeaan suuntaan.
- Sensorin tulee tunnistaa kuinka lähellä kohde on, tieto lähetetään eteenpäin ja tieto esitetään alustalla sekä käyttäjän asettama ilmoitusmuoto tulee olla asetettu ja yhteystiedot oikein.

Ilmankosteuden tarkkailu

Käyttäjänä haluan saada tietoja (huoneen) ilmankosteuden arvoista, koska haluan tietää miten ilmankosteus vaihtelee huoneessa.

- Sensori tulee sijoittaa sellaiselle paikalle, jossa ei ole liikaa häiriötekijöitä.
- Arvot lähetetään edelleen alustalle ja arvot näytetään numeerisena käyttäjälle omassa elementissään.
- Arvot näytetään prosentteina.

Asetettujen lämpötila- ja ilmankosteusarvojen muuttuminen yli rajojen

Käyttäjänä haluan saada huomautuksen lämpötilan ja ilmankosteuden muuttuessa paljon, nopeasti tai yli tietyn rajan, koska haluan tietää jos jotain tapahtuu tai arvot menevät yli.

- Lämpötilaa ja ilmanpainetta mittaavien sensorien tulee olla yhtä aikaa toiminnassa.
- Arvot näytetään alustassa, ja arvojen viereen ilmestyy elementtiin liittyvä varoitusmerkki.
- Käyttäjälle tulee lähettää jolloin ilmoitusmuodolla rajan ylitys.

Muiden arvojen muuttuminen ja yleinen ylläpito

Käyttäjänä haluan saada ilmoituksia jos rajat ylittyvät tai havaitaan jotain muuta, koska haluan pysyä ajan tasalla tilanteesta, missä tahansa ja milloin tahansa.

- Sensorien lähettämät tiedot tulee olla reaaliaikaisia, luotettavia sekä oikeassa muodossa.
- Käyttäjän tiedot tulee olla oikein ja alustalla tulee olla myös tiedot kaikista rajoista, joista tulee ilmoittaa.

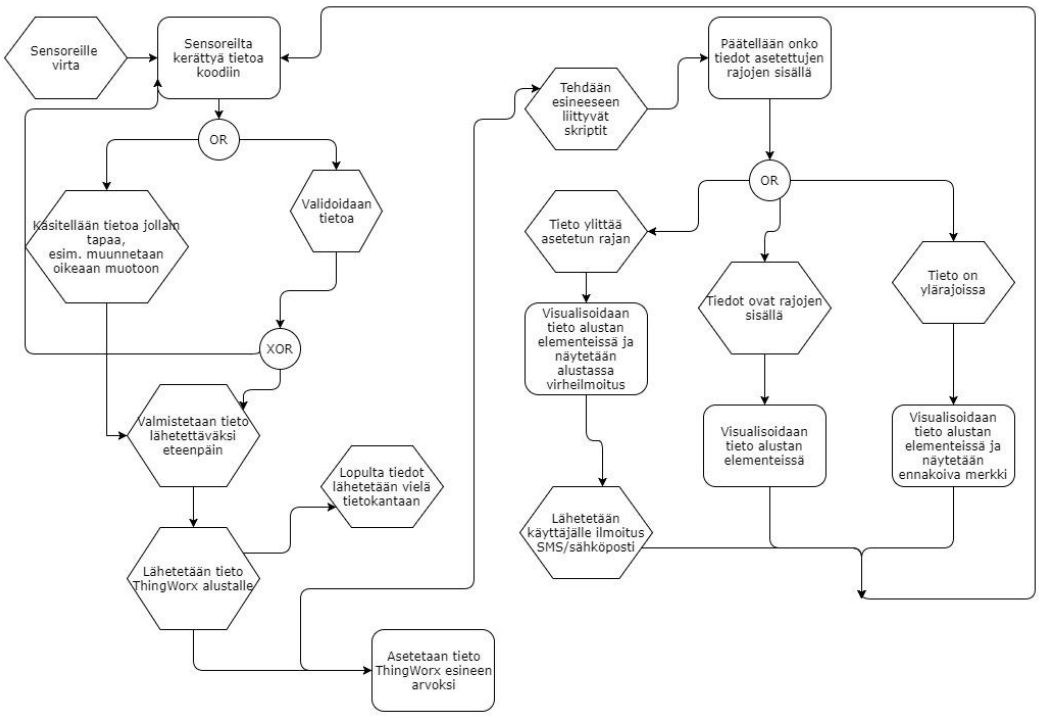
Valoisuuden ilmoittaminen

Käyttäjänä haluan saada tietää valoisuudeen arvoja sensorilla, esim. onko valot päällä vai ei, koska voin tietää milloin valot ovat päällä tai tarkistaa jäivätkö valot jostain syystä päälle.

- Valosensori tulee kohdistaa oikein.
- Tiedot tulee lähettää edelleen ja tiedoille tulee olla oma elementti alustassa.
- Pitää tietää tuleeko muutoksesta lähettää jossain vaiheessa käyttäjälle ilmoitus.

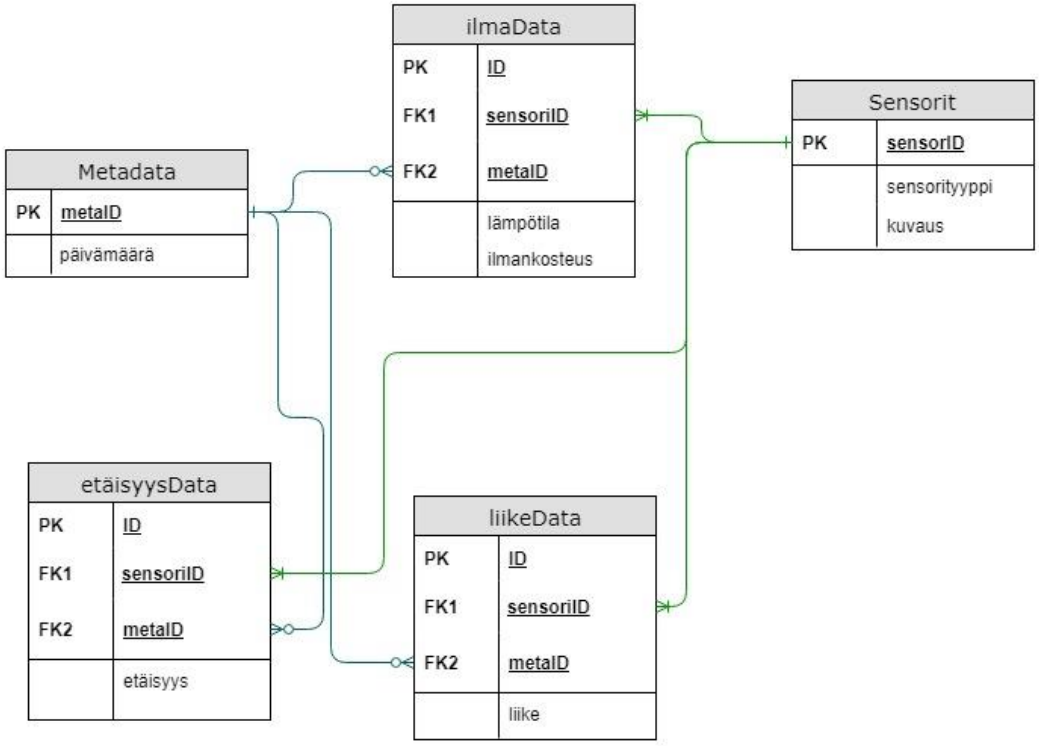
Kuvio 9. Palvelun käyttäjätarinat

Palvelun prosessikaaviosta selviää paremmin toimintalogiikka alusta loppuun, jota palvelun toteutuksessa aiotaan hyödyntää (Kuvio 10). Ensimmäisenä varmistetaan, että anturit ovat oikein asennettu ja saavat virtaa. Ohjelmakoodi kerää antureilta tietoa, joka sitten validoidaan, mahdollisesti muutetaan tietoa toiseen tietotyyppiin tai käytetään anturien arvoja laskutehtäviin. Kuitenkin kaikki lähetettävät tiedot valmistellaan edelleen lähetykseen. Jos validoitava tieto ei ole hyväksyttävää, se huomioidaan ja lähetetään edelleen alustalle sekä tietokantaan. Tällaista tietoa voi olla häiriöt anturissa tai liian pienet arvot. Tiedot lähetetään edelleen kohti ThingWorx-alustaa käyttäen REST API-komentoa. ThingWorx vastaanottaa tiedot ja ne asetetaan niille kuuluvien esineiden ominaisuuksiin. Sitten tieto pääsee analysointivaiheeseen, joka tässä tapauksessa on asetettujen rajojen käyttäminen tapahtumissa sekä tilauksissa. Jos tieto ylittää rajan, silloin esineeseen liittyvä visualisointielementti näyttää arvon lisäksi virheilmoituksen visualisointialustalla sekä mahdollisesti lähettää käyttäjälle virheilmoituksen. Jos tieto on ylärajoilla, tieto visualisoidaan esineeseen liittyvässä elementissä sekä käyttäjälle näytetään keltainen huomiomerkki. Jos taas tieto on rajojen sisällä, se vain visualisoidaan alustalla. Ja lopulta anturin tiedot lähetetään tallennettavaksi tietokantaan, jonka jälkeen pääsemme takaisin anturikerrokselle.



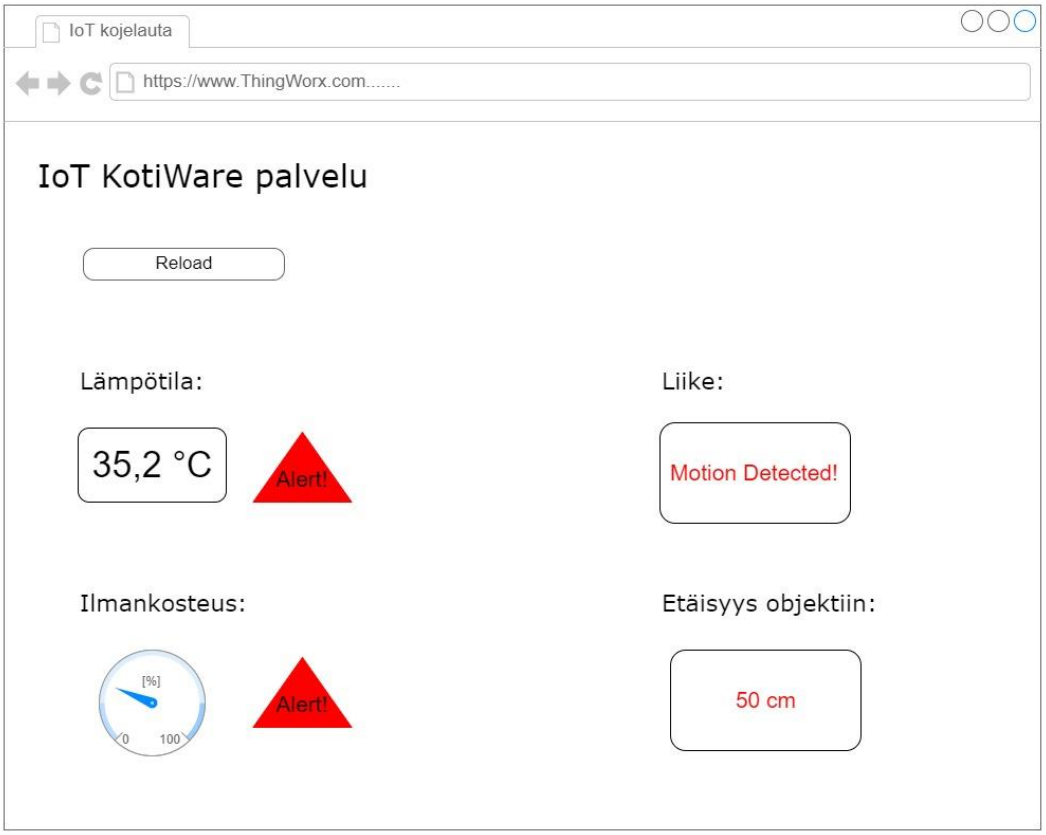
Kuvio 10. Palvelun prosessikaavio

ER-kaaviolla (Entity-Relationship) suunnittelin tietokannan tauluja ja minkälaisilla suhteilla tietokannan taulut yhdistyvät toisiinsa (Kuvio 11). Suunnitellussa tietokannassa on antureille ja metadatalle omat taulut. Sensorit-taulu sisältää anturityypin (esimerkiksi DHT11) ja vapaan kuvauksen esimerkiksi minkälaista tietoa kyseinen anturi kerää (lämpötila ja ilmankosteus) sekä yksikkö (˚C ja %). Metadata-taulussa on päivämäärä, jonka avulla voidaan analysoida tietyn aikavälin aikana kerättyjä tietoja. Päädyin suunnittelussa laittamaan anturien arvot omiin tauluihinsa, vaikka tietokanta voitaisiinkin luoda vähemmillä tauluilla. Antureiden tauluihin liittyvät sensorit- ja metadata-taulujen viiteavaimet (sensoriID ja metaID), joilla voidaan yhdistää tietyt tietueet yhteen. Kaikki muut tiedot anturien tauluissa, paitsi liike-tietue (tyypiltään totuusarvomuuttuja), ovat liukulukuja.



Kuvio 11. Tietokannan ER-kaavio

Tein myös todella alkeellisen rautalankamallin, johon sijoitin elementtejä ja otsikoja (Kuvio 12). Lämpötila- ja ilmankosteustiedot olisi järkevää asettaa sivulle lähekkäin. Lämpötilaa voisi näyttää LED-näyttöelementillä ja ilmankosteutta voitaisiin esittää mittarielementillä. Näille molemmille haluaisin myös luoda 3 erilaista varoitusta, punainen kolmio kun arvo menee ylitse, keltainen kolmio kun arvo on rajamailla ja sininen kolmio kun arvo on liian alhaalla. Liike- ja etäisyystietoa voisi esittää samantyylisillä elementeillä, kuten esimerkiksi tekstilaatikolla tai muulla yksinkertaisella elementillä. Elementin fontin väri voisi olla punainen silloin, kun liikettä havaitaan ja etäisyys on vain 20 cm. Palvelulle voisi antaa myös jonkinlaisen otsikon ja päivityspainikkeen sekä automaattisen päivityksen n. 8 sekunnin välein. Näihin elementteihin tulee tietysti liittää itse esineet, joita tulee olemaan yhteensä 3. Jokaisella esineellä on omat ominaisuudet ja mallipohjana tulee olemaan yleinen esine, koska esineitä ei ole kuin yksi jokaista. Mallipohjan voitaisiin luoda silloin, jos käytössä olisi useampia samanlaisia antureja.



Kuvio 12. Rautalankamalli visuaalisesta esityksestä

PTCllä on oma IoT university-portaali, jossa on erilaisia kursseja liittyen muun muassa ThingWorxin käyttöön ja palveluiden suunnitteluun. Ennen oman palvelun suunnittelua, tein yhden kurssin IoT university-portaalista (Fundamentals of IoT Development with ThingWorx), jossa tehtiin alusta loppuun pienimuotoinen esineiden internetin projekti ja käytiin lävitse esineiden internetiin liittyviä osa-alueita yksitellen. Tämän kurssin avulla sain peruskäsityksen, miten esineitä ja näihin liittyviä muita elementtejä luodaan alustalle, miten analysointia voidaan käyttää esineiden tietoihin ja miten luodaan visualisointi alusta. Ja käytännön toteutuksessa käytän samaista kurssimateriaalia mallina omalle palvelulle.

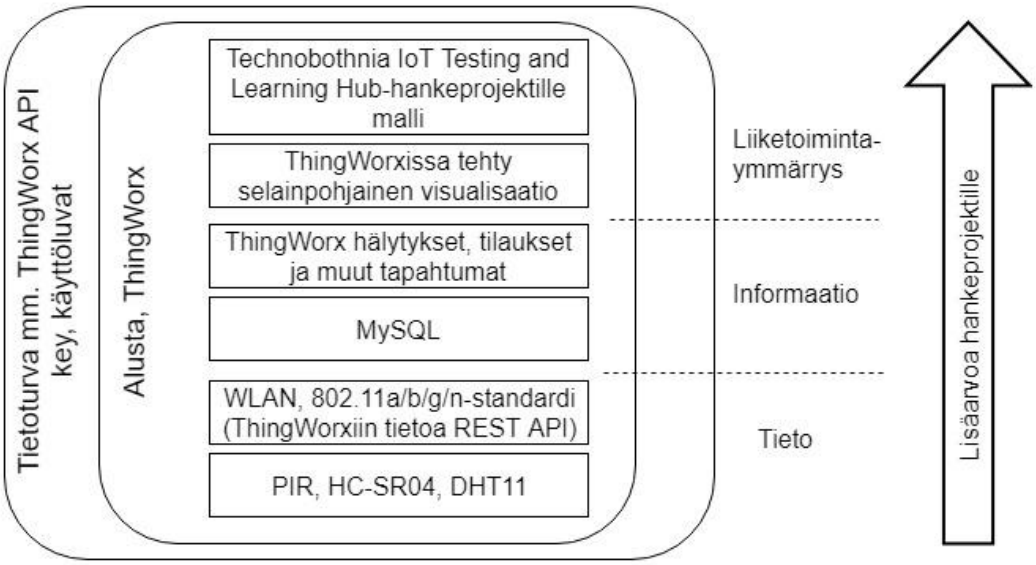
6 PALVELUN TOTEUTUS

Palvelun toteutus alkoi Raspberry Pi -mikrotietokoneen ja sopivan anturipaketin tilauksella. Raspberry Pi 3 B-malli on mikrotietokone, joka soveltuu kaikenlaisiin projekteihin, sillä sen käyttöönotto on nopeaa ja se on helppokäyttöinen sekä edullinen vaihtoehto. Raspberry Pi 3 B-mallissa on 40-pinnin GPIO (General Purpose Input/Output) portti ja Broadcom BCM43438-piiri, joka tukee Bluetooth 4.1 Classic-, BLE-, (802.11n) WiFi-teknologiaa. Tässä mallissa on myös 10/100 Ethernet, 1GB RAM, 4 ARM Cortex-A53-prosessoria sekä VideoCore IV-grafiikkasuoritin. (The MagPi Magazine 2015.) Raspberry Pin käyttöjärjestelmäksi valitsin ilmaisen Debian-pohjaisen Raspbian-käyttöjärjestelmän. Raspberry Pin kanssa toimivia käyttöjärjestelmiä on monia ja käyttäjä voi itse valita sellaisen käyttöjärjestelmän, joka sopii parhaiten omaan käyttöön. Toteutuksessa käytin komentoriviä tarvittavien päivitysten, pakettien ja muiden komponenttien lataamiseen, ohjelmakoodin ajamiseen sekä testaukseen. Käytin erillistä tietokonetta tiedon etsintään, koodin kirjoittamiseen Notepad++-ohjelmalla sekä FileZilla-tiedonsiirto-ohjelmaa ottaakseni yhteyttä ja siirtääkseni ohjelmakoodin Raspberry Pihin. Anturipaketin löysin Amazonista (The Internet of Things-anturipaketti Kookyelta), jossa mukana tuli koekytkentälevy, useita antureja, naaras- ja uroskaapeleita, vastuksia, LED-valoja ja muita osia. Pienissä projekteissa koekytkentälevyn käyttäminen on helpointa, sillä anturien asentamiseen ei tarvita juottamista, vaan anturit vain painetaan koekytkentälevyn koloihin ja sähköpiirien luominen on todella helppoa (McEwen, Cassimally 2014, 88, 91–92). Paketti soveltui omaan projektiin todella hyvin, ja hintakin oli huokea. Valitsin tämän paketin, koska paketissa oli monenlaisia osia ja sitä voisi käyttää myös muissakin projekteissa. Etäisyysanturin tilasin erikseen Amazonista, koska sitä ei ollut anturipaketissa. Alustana toimii ThingWorx Composerin 8.2 version kuukauden kokeilutili.

6.1 Anturit

Käytännön toteutuksessa sovelsin teoriassakin käsiteltyä teknologiapinoa (Kuvio 13). Ensimmäiset kaksi kerrosta koostuvat tiedosta, joka toteutetaan antureilla ja

tietoliikenteellä. Tietoliikenteenratkaisuna toimi oman kodin WLAN, vaikka antureille kannattaisikin tehdä oma erillinen verkko tietoturvan takia. WLAN-verkko käyttää IEEE 802.11a/b/g/n-standardia. 802.11a/b/g/n-standardi on tälle projektille ihan hyvä, osakseen siksi että Raspberry Pi 3 B-malli ei tue kuin 802.11n-standardiin asti. Anturit, jotka asensin, ovat lämpötilan ja ilmankosteuden, etäisyyden sekä liikkeentunnistuksen anturit. YouTubesta ja internetistä löytyi useita lähteitä, joiden avulla sai peruskäsityksen siitä, miten antureita asetetaan koekytkentälevyyn, saadaan virtaa sekä miten ne toimivat. Useissa lähteissä oli myös ohjelmakoodi, miten anturien tietoja voidaan saada digitaaliseen muotoon Python-koodin avulla ja tulostamaan päätelaitteelle arvoja. Näiden avulla pystyin aloittamaan jostain ja saada lisää tietoa useiden anturien lisäämisestä yhteen koekytkentälevyyn. Etäisyysanturi oli vaikein ottaa käyttöön, koska kytkentään tarvittiin vastuksia ja en ole aikaisemmin tehnyt tällaisia sähköpiirejä. Jokaisessa anturin pinnissä on yleensä erilliset PCB-merkinnät, mikä helpotti asennusta.



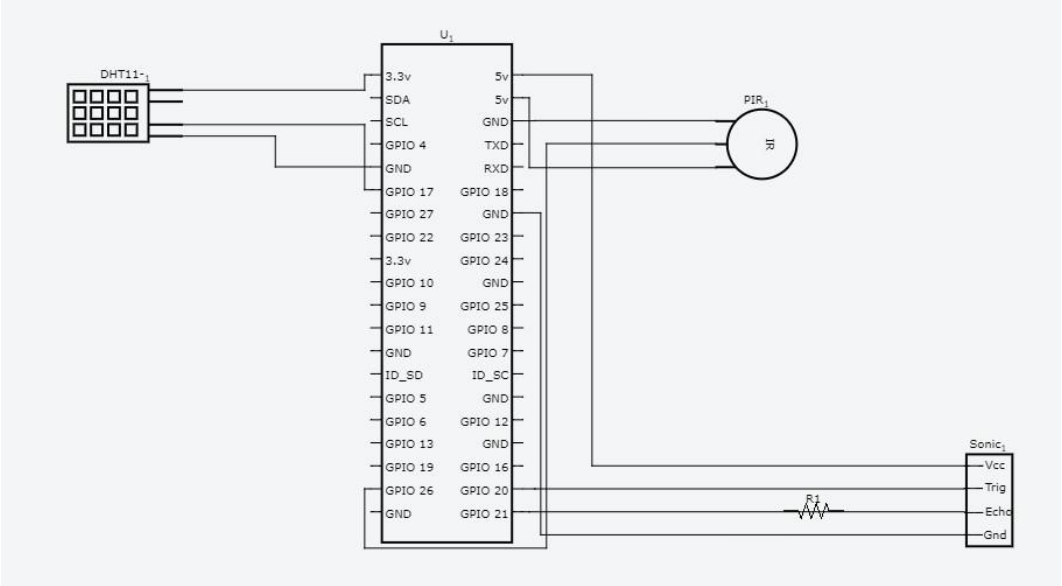
Kuvio 13. Toteutuneen palvelun teknologiapino

HC-SR501 eli PIR-liiketunnistin kostuu kolmesta pinnistä (virta-, maa- ja tietopinnistä), anturissa on myös viive- ja herkkyyssäädin. PIR-liiketunnistin on helppokäyttöinen, pienitehoinen, linssin peittoalue on laaja sekä se on edullinen. Anturin toiminta perustuu pyrosähköiseen anturiin, joka havaitsee infrapunasäteilyä ympäristöstä. Pyrosähköinen anturi on itseasiassa puolitettu ja nämä ovat asetettu niin,

että puolikkaat kumoavat toisensa. Se etsii liikkeen muutoksia, ei vain infrapunasäteilyä ja sen havaintoalue on 110*70 astetta sekä noin 6 metriä. (Adafruit 2017a.)

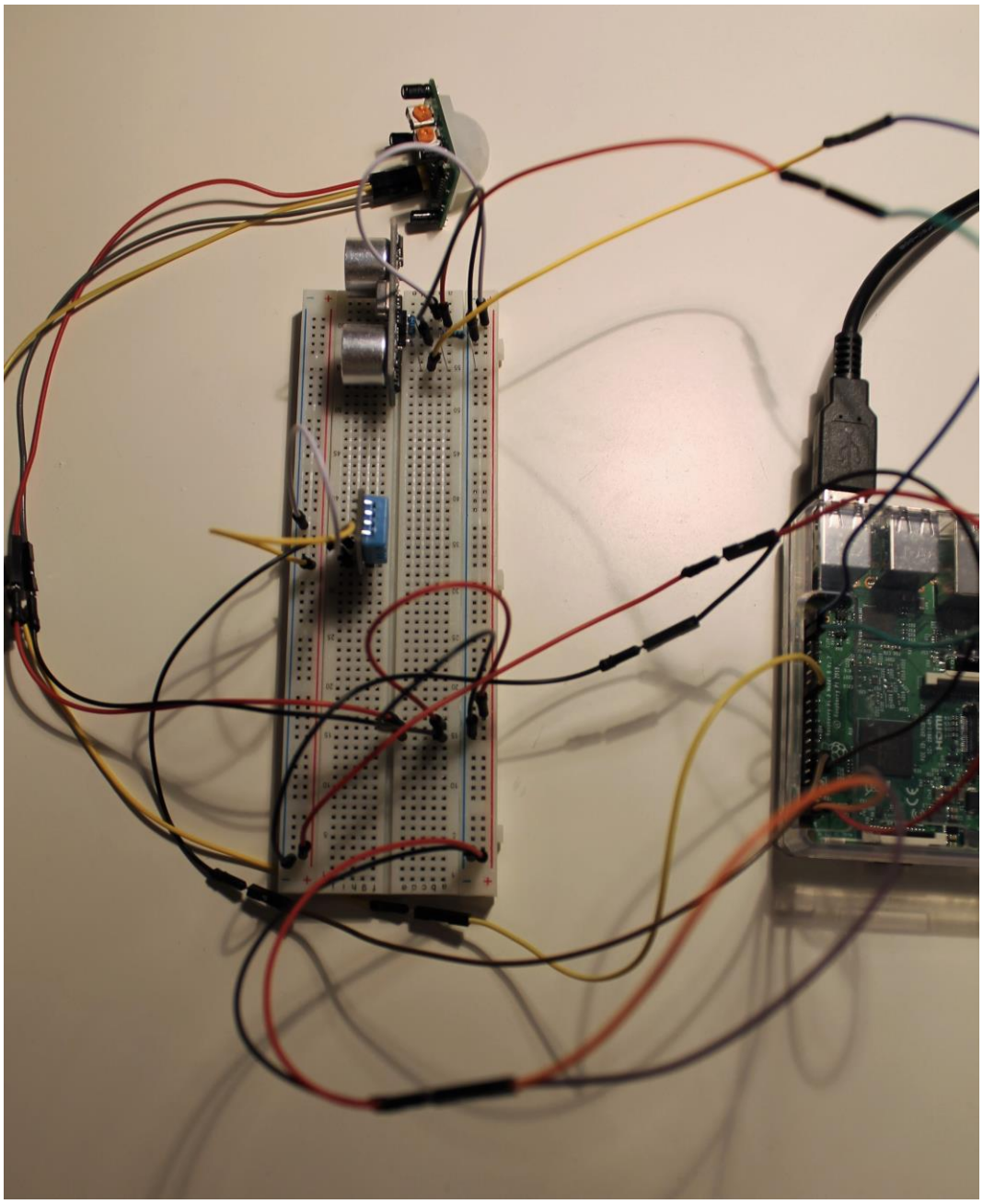
DHT11-anturilla saadaan mitattua lämpötilaa sekä ilmankosteutta yhtä aikaa, joka käyttää 3,3–5V virtapinniä, maapinniä sekä tietopinniä. Lämpötilaa voidaan tarkastella 0–50 astetta, jonka tarkkuus on ±2 astetta. Ilmankosteutta voidaan mitata 20–80%, jonka tarkkuus on ±5%. DHT11-anturi on hyvin edullinen, vaikka ei olekaan yhtä hyvä kuin DHT22-anturi, jolla voidaan mitata myös pakkasasteita ja sen tarkkuus on paljon parempi. DHT11-anturi on kuitenkin hyvä pieniin projekteihin. Tiedonkeräystä suositellaan enintään kerran 2 sekunnissa, jos tietoja kerätään tätä useammin tiedot saattavat vääristyä. (Adafruit 2017b.) DHT11-anturin käyttöönotto oli todella helppoa ja ohjeita kyseisen anturin ohjelmointiin sekä asentamiseen löytyy paljon.

HC-SR04 on ultraääneen perustuva etäisyysanturi, jolla voidaan mitata etäisyyttä anturin alueella olevaan objektiin. Anturi koostuu yhdestä lähettimestä, vastaanottimesta sekä ohjauspiiristä. Anturin toiminta perustuu lähettimen korkeataajuisiin ultraääniaaltoihin, jotka osa heijastuvat ja kimpoavat takaisin osuessaan objektiin. Kun vastaanotin havaitsee takaisin kimpoavan ultraääniaallon, voidaan laskea ultraääniaallon lähetykseen ja vastaanottoon kulunut aika ja laskea objektin etäisyys melko tarkasti. Anturi koostuu virtapinnistä, maapinnistä, kaikupinnistä sekä liipaisupinnistä. Liipaisupinnin avulla lähetetään ultraääniaalto ja kun takaisin tuleva ultraääniaalto huomataan, lähetetään kaikupinnille 5V signaali. Signaalin ajaksi asetetaan kaikupinnin arvo korkeaksi, josta lasketaan välimatka osuneeseen objektiin. Yksi tärkeä ja huomioitava asia on se, että Raspberry Pin GPIO-pinnit eivät kestä kaikupinnin 5V virtaa, vaan se pitää madaltaa vastuksien avulla GPIO-pinnien kestävään enintään 3,3V virtaan. (ModMyPi 2014.)



Kuvio 14. Elementtien piirikaavio

Piirikaaviossa näkyy, miten anturit käytännössä kytkettäisiin Raspberry Pihin. Piirikaavio toimii myös helposti ymmärrettävänä mallina, kun antureita asetetaan koekytkentälevylle (Kuvio 14). Vaikka koekytkentälevyä ei piirikaaviossa olekaan piirretty, sen käyttäminen on helppoa. Joissain tapauksissa se ei edes ole välttämätöntä, sillä anturit voidaan kytkeä suoraan Raspberry Pin GPIO-pinneihin. R1-vastus etäisyysanturissa on 200 Ω ja itseasiassa monessa tutoriaalissa oli laitettu vastus menemään myös maan kautta. Vastuksilla alennetaan sähkövirtaa sekä maan kautta menevä 200 Ω vastus vaikuttaa signaalin vahvuuteen. Käytetyssä DHT11-anturissa on vain 3 pinniä, koska tarvittava vastus tiedon lähettämiseen on sisäänrakennettuna tässä mallissa. PIR-anturin viive- ja herkkyyssäätimissä käytettiin oletusarvoja. Kuviossa 15 nähdään, miten anturit on asennettu koekytkentälevylle.



Kuvio 15. Anturit koekytkentälevyssä

6.2 Ohjelmointi, tietovarastointi ja analysointi

Anturien ohjelmoinnissa käytettiin Pythonia (2.7), jota en ole aikaisemmin käyttänyt. Pythonin käyttäminen ei kuitenkaan ollut vaikeaa, koska peruslogiikka minkä tahansa ohjelmointikielen taustalla pysyy samana, vain syntaksi muuttuu ja Pythonin sanotaan olevan maailman helpoin ohjelmointikieli. Tietojen lähettämiseen alustalle käytetään REST APIa sekä PUT-komentoja. Komennot alustettiin JSON-

muotoon Postman-ohjelmalla, jota käytettiin myös PTC:n IoT university-portaalin kurssilla. Postman-ohjelmalla voidaan myös testata manuaalisesti ThingWorx esineiden ja tilauksien toimivuutta. Ensimmäisenä tulee lisätä eri ohjelmakirjastot, joita käytetään muissa kohdissa koodia esimerkiksi Adafruitin valmis GitHub DHT-anturien ohjelmakirjasto. Asetetaan myös valmiiksi GPIO-pinnit, joita käytetään Raspberry Pissä antureiden tietojen siirtämisessä. Myös Python-pohjaisen py-MySQL-tietokantayhteyden käyttämiseen ja tietojen lisäykseen luodaan siihen käytettävät metodit (Liite 7).

```
# Prepare SQL query to INSERT a records into the database.
sqlTempHumi = """INSERT INTO ilmaData(ilmaID,
   sensoriID, metaID, lampotila, ilmankosteus)
   VALUES (NULL, %s, %s, %s, %s)"""
sqlMotion = """INSERT INTO liikeData(liikeID,
   sensoriID, metaID, liike)
   VALUES (NULL, %s, %s, %s)"""
sqlDistance = """INSERT INTO etaisyysData(etaisyysID,
   sensoriID, metaID, etaisyys)
   VALUES (NULL, %s, %s, %s)"""
sqlMeta = """INSERT INTO metadata(metaID, paiva)
   VALUES (NULL, NULL)"""

#thingWorx rest
urlMotion = "https://devportal.ptc.io/Thingworx/Things/MotionSensor/Properties/Motion"
urlTemperature = "https://devportal.ptc.io/Thingworx/Things/TemperatureAndHumiditySensor/Properties/Temperature"
urlHumidity = "https://devportal.ptc.io/Thingworx/Things/TemperatureAndHumiditySensor/Properties/Humidity"
urlDistance = "https://devportal.ptc.io/Thingworx/Things/DistanceSensor/Properties/Distance"
```

Kuvio 16. SQL-lauseiden ja REST ”otsikoiden” alustaminen

SQL-lauseet tietojen lisäykseen luotiin valmiiksi erillisiin muuttujiin (Kuvio 16). Myös REST API-komennoissa käytetyt parametrit, kuten esimerkiksi sovellusavain, alustettiin valmiiksi ja vähentämään turhaa toistoa. Lisäsin erillisen metodin etäisyysanturia varten, koska useimmissa mallitutoriaaleissa oli laitettu etäisyyden laskenta erilliseen metodiin (Liite 8). Itse anturien tietojen lukeminen, käsittely ja lähetys eteenpäin tehdään while-silmukan sisällä (Kuvio 17). Ensimmäisenä kokeillaan DHT11-anturin lukemista, jonka jälkeen on laitettu ehtolausekkeet eri antureiden arvoille. Ensimmäinen ehtolauseke on liiketunnistimelle, valmistellaan tiedot lähetettäväksi ja ensin lähetetään tiedot REST PUT-komennolla ThingWorxin esineen ominaisuuteen. Koska testailin ja korjailin koodia erillisen näytön kautta komentorivillä, oli helpompaa myös tulostaa arvo näkyviin. PUT-komennon jälkeen otetaan yhteyttä tietokantaan, luodaan ensin metatiedot ja lisäyslauseen luoma uusi ID-tieto otetaan talteen erilliseen muuttujaan. Lisätään anturin lukema tieto tietokantaan ja suljetaan yhteys. Ja viimeisenä on muutamien sekuntien viive, jotta saadaan luotettavaa tietoa antureilta.

```
while True:
    #read the DHT11 sensor data
    humidity, temperature = Adafruit_DHT.read_retry(11, gpio)
    if GPIO.input(pir): #Check whether pir is HIGH
        motion = True
        #REST PUT
        payload = json.dumps({"Motion":True})
        response = requests.request("PUT", urlMotion, data=payload, headers=headersMotion)
        print "Motion Detected!"
        #connect
        conn, cur = connect()
        #metadata insert
        metaID = executeMeta(conn, cur, sqlMeta)
        sensoriID = 2
        #data insert
        data_word = (sensoriID, metaID, motion)
        executeSQL(conn, cur, sqlMotion, data_word)
        #close connection
        disConnect(conn)
        time.sleep(2) #D1- Delay to avoid multiple detection
        time.sleep(0.1) #While loop delay should be less than detection(hardware) delay
```

Kuvio 17. Liiketietojen käsittely if-lauseessa

Ehtolauseet ovat logiikaltaan samanlaiset myös muissa antureissa, vain anturien mittaamat tulokset, käytettävät REST-parametrit ja SQL-lauseet ovat erilaisia (Liite 9–10). Jos anturi ei saa lukemaa tai se on virheellinen, tiedot täytyy myös silloin lähettää tietokantaan sekä alustalle. Tämä toteutetaan else-lauseessa, jolloin poikkeamat havaitaan ja voidaan edelleen analysoida tilannetta (Kuvio 18).

```
    else:
        print "No motion.."
        motion = False
        #Motion REST
        payload = json.dumps({"Motion":False})
        response = requests.request("PUT", urlMotion, data=payload, headers=headersMotion)
        #connect
        conn, cur = connect()
        #metadata insert
        metaID = executeMeta(conn, cur, sqlMeta)
        sensoriID = 2
        #data insert
        data_word = (sensoriID, metaID, motion)
        executeSQL(conn, cur, sqlMotion, data_word)
        #close connection
        disConnect(conn)
        time.sleep(2)
        time.sleep(0.1)
```

Kuvio 18. Liiketunnistimen else-lause

Tietovarastona toimii MySQL-tietokanta, vaikka ThingWorxissä voidaan luoda myös tietovaraston tyyppisiä esineitä ja kokonaisuuksia, kuitenkin erillisen relaatiotietokannan luominen oli suositeltavaa (Liite 4–6). Tietokanta luotiin aiemman suunnitelman mukaisesti. Analytiikkana toimivat esimerkiksi tilaukset ja hälytykset ThingWorxin esineiden tasolla (Taulukko 2).

Taulukko 2. Esineiden tiedot, hälytykset, tilaukset ja tapahtumat.

Esine (esineen pohjamalli)	*Ominaisuudet (tietotyyppi, yksikkö)*	*Hälytykset*	*Tilaukset*
TemperatureAndHumiditySensor (yleinen esine)	Temperature (number, Celsius) IsTooHighTemp (boolean) Humidity (number, %) IsTooHighHumi (boolean)	High Temperature High Humidity	Lähetä tekstiviesti käyttäjälle, kun lämpötila tai ilmankosteus hälytys laukaistaan. Muuta IsTooHighTemp ja IsTooHighHumi ominaisuuksien arvot, kun temperature ja humidity ominaisuuksien arvot muuttuvat.
MotionSensor (yleinen esine)	Motion (boolean) Detected (string)		Raakainformaatio käyttäjälle ymmärrettävään muotoon.
DistanceSensor (yleinen esine)	Distance (number, cm) IsTooClose (boolean)	TooClose	Arvon muuttuessa arvioidaan, onko objekti liian lähellä (IsTooClose) ja vaihdetaan totuusarvomuuttuja sen mukaan.
smsToPhone (Twilio)			Twilion palvelut: GetNotificationHandlers SensSMSMessage SendVoiceMessage

Kuten aikaisemminkin mainitsin, esineitä oli niin vähän, että en luonut mallipohjia esineille. Mallipohjia on hyvä käyttää periytymiseen, jos esineillä on useita samanlaisia ominaisuuksia tai niitä muodostetaan enemmän kuin yksi. Jokainen esine perii kuitenkin yleisen esineen ominaisuudet, jotka ovat esineen toiminnalle oleellisia. Lämpötilan ja ilmankosteuden esineen ominaisuudet ovat numeerisia arvoja, joten paras tietotyyppi oli liukuluku. Hälytyksiä tein tälle esineelle, kun ominaisuuksien arvot nousevat asetettujen rajojen yli (lämpötila 30 ˚C ja ilmankosteus 60 %). Nämä raja-arvot valitsin palvelun käyttötapauksen takia sekä anturei den mittauskapasiteettien mukaan. Liike-esineellä on kaksi erilaista ominaisuutta, liike ja liikkeen havaitseminen. Liikeominaisuus on tietotyypiltään totuusarvomuuttuja ja liikkeen havaitsemisen ominaisuus on tekstityyppi. Liike-esineellä on yksi tilaus, joka muuttaa totuusarvomuuttujan käyttäjälle ymmärrettävään muotoon (”Motion Detected!”, ”No Motion..” ja ”Unknown”, ks. Liite 2). Tilaus on JavaScipt-ohjelmointikielellä tehty moniosainen ehtolause. Viimeisenä on etäisyysesine, jolla on etäisyysominaisuus, joka ilmoitetaan liukulukuna. Esineessä on myös totuusarvomuuttuja IsTooClose-ominaisuus, joka ilmoittaa, kun objekti on liian lähellä. Totuusarvomuuttuja vaihtuu tilauksen avulla joka kerta kun etäisyyden arvo muuttuu. SmsToPhone-esine on nimensä mukainen esine, jonka avulla voidaan lähettää käyttäjän puhelimeen viestejä ja ääniviestejä.

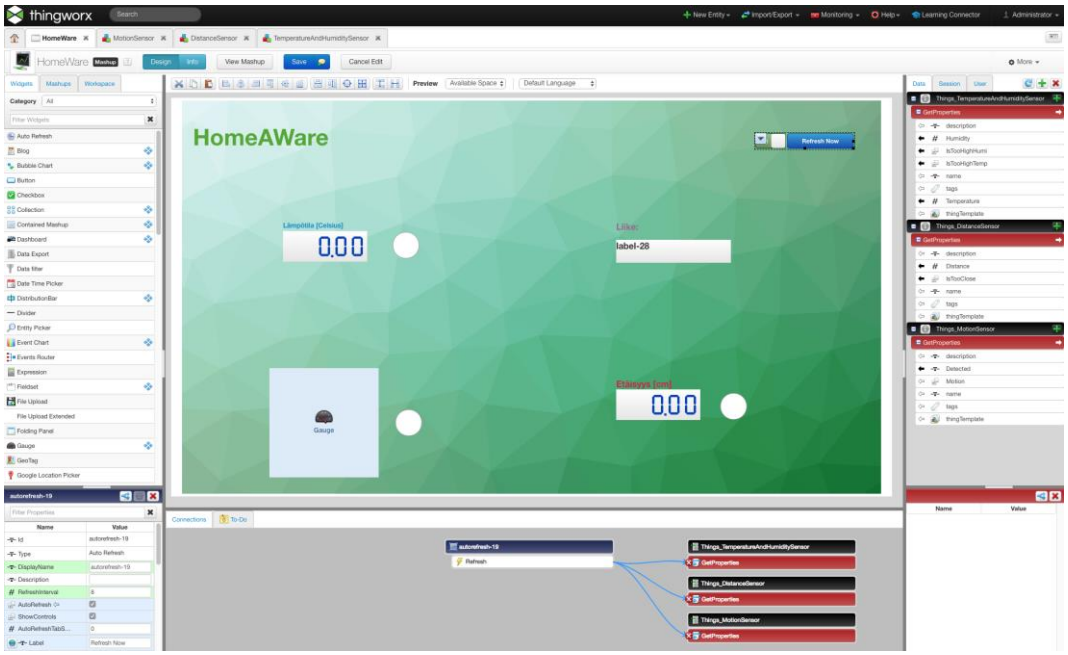
6.3 Tiedon visualisointi

Visuaalisen esityksen eli mashupin luominen oli helppoa, koska ohjelmointia ei tarvita ja elementit ovat helppo yhdistää esineiden ominaisuuksiin tai esimerkiksi hälytyksiin. Halusin tehdä yksinkertaisen esityksen, jossa voin käyttää erilaisia elementtejä. Tein sivusta responsiivisen ja yhdistin sivulle valmiin ThingWorx bannerin sivun ylätunnisteeseen. Kun lisäsin otsikon sivulle, editori lisäsi automaattisesti myös paneelin sivulle, koska otsikkoelementti ei voi olla ilman paneelia. Paneelielementti muutti sivun responsiivisuutta siten, että paneelin alue ei ole responsiivinen, mutta sen ulkopuolella oleva alue eli banneri on responsiivinen. Laitoin paneeliin myös Creative Commons-haun avulla taustakuvan, joka antaa sivulle hieman lisää väriä (Pixabay 2017). Koska paneelielementti ei kuitenkaan ole responsiivinen, tietyn kokoisissa päätelaitteissa elementit eivät mahdu taustakuvan päälle.

Mielestäni tämä ei ole kovin hyvin toteutettu, jos halutaan kaikkien elementtien olevan responsiivisia. Kuvia voidaan helposti lisätä ThingWorxiin, ne ladataan etukäteen valmiiksi mediakokonaisuuksiin.

Ensimmäiset elementit tiedonnäyttämiseen ovat LED-näyttö lämpötilalle sekä mittari ilmankosteudelle. Lisäsin näille elementeille omat otsikot, elementin ominaisuusikkunasta sekä vaihdoin otsikontyyliä. Seuraavaksi lisäsin LED-näytön etäisyydelle sekä otsikkoelementin liikkeelle. Aluksi ajattelin tehdä kuvaelementit liikkeelle, esimerkiksi kuvan, kun liikettä on havaittu, jossa on punainen varoituskolmio. Kuitenkin olin tehnyt liike-esineelle tilauksen, joka muuttaa digitaalisen arvon käyttäjälle ymmärrettävään muotoon ja halusin käyttää tätä hyväksi. Lisäsin vain uuden tyhjän otsikkoelementin ja lisäsin sille taustavärin erottuakseen sivulta.

Data-välilehdeltä lisäsin kaikkien esineiden ominaisuudet, ThingWorxin valmiilla GetProperties-metodin avulla ja koska halutaan, että esineiden ominaisuudet latautuvat sivun avautuessa rastitaan myös ”mashup loaded”-valintaruutu. Kun kaikkien esineiden ominaisuudet on haettu, data-välilehdeltä löytyy kaikki esineet, käytetty metodi sekä ominaisuudet. Aloitin lämpötilan ja ilmankosteuden esineiden ominaisuuksien yhdistämisen editorin elementteihin. Valitsin data-välilehdeltä lämpötila-ominaisuuden ja vedin sen LED-näytön päälle. Tämä toiminto laukaisee pienen ponnahdusikkunan, josta voidaan valita, minkä tyyppiseen tietomuotoon halutaan kyseinen ominaisuus liittää, valitsin data-tyypin. Tein samalla tavalla myös ilmankosteudelle sekä etäisyydelle. Liike-elementin yhdistäminen ominaisuuteen oli hieman erilainen, sillä halutaan käyttää tilaukseen liittyvää ominaisuutta tiedon näyttämisessä. Tilauksessa käytettiin detected-nimistä ominaisuutta näyttämään käyttäjälle suunnattua tekstiä, joten valitsin sen ominaisuuden ja pudotin sen tyhjän otsikkoelementin päälle. Tällä kertaa halutaan näyttää ominaisuuden tieto tekstityyppisenä, joten valitsin ponnahdusikkunasta text-tyypin (ks. Liite 3).



Kuvio 19. Visualisointi editorin päänäkymä

Halusin myös lisätä eri esineille statusta esittävät elementit sivuille, jolloin voidaan näyttää käyttäjälle, milloin esimerkiksi lämpötila menee asetetun rajan yli. Lisäsin lämpötilan, ilmankosteuden sekä etäisyyden elementtien viereen muotoelementit, jotka kaikki ovat ympyröitä. Sen lisäksi halusin, että muotojen väri vaihtuu tilamäärittelyn mukaan, kun hälytys laukaistaan. Tein tätä varten lämpötilalle, ilmankosteudelle ja etäisyydelle omat tilaukset, joka laukaistaan aina kun arvo muuttuu ja vaihtaa tähän liittyvän ominaisuuden arvon totuusarvon mukaan (true tai false) esimerkiksi lämpötilaan liittyvän IsTooHighTemp-ominaisuuden mukaan. Lisäsin myös Twilio-laajennuksen ThingWorx Composeriin, joka on pilvipohjainen (PaaS, Platform as a Service) viestintäpalvelu tekstiviestien, sähköpostien ja puheluiden lähettämiseen ja vastaanottamiseen. Twilio-laajennuksen esineellä on valmiita tilauksia, kuten esimerkiksi SendSMSMessage-tilauksella voin lähettää käyttäjälle tekstiviestin, joka ilmoittaa, kun lämpötila ja ilmankosteus nousevat liian korkeiksi (ks. Liite 1). Twilio-laajennuksen lisääminen oli todella helppoa ja yksinkertaista, tein oman kokeilutilin Twilio-palveluun ja sain sieltä konfigurointitiedot, joita tarvitsin ThingWorxin esineessä (SmsToPhone-esine).

Kuvio 20. Palvelun valmis visualisaatio

Viimeisenä lisäsin sivulle automaattisen päivityksen ja painikkeen manuaaliselle päivitykselle. Päivitys tehdään 8 sekunnin välein, aikaa voidaan muokata painikkeen ominaisuuksista. Jotta päivitys toimisi, tulee elementin oma Refresh-tapahtuma vetää data-välilehdellä olevien esineiden GetProperties-tapahtumien päälle, jolloin painike liitetään tähän tapahtumaan ja ominaisuudet haetaan tietyin väliajoin (Kuvio 19). Visualisaatioon voidaan myös lisätä dynaamisia tapahtumia, joiden avulla voidaan muokata esineiden ominaisuuksia manuaalisesti esimerkiksi painikkeiden sekä tekstikenttien avulla. Omaan visualisaatioon en lisännyt dynaamisia elementtejä. Valmiin visualisaation testaus tapahtui erillisessä selaimessa, jossa havaittiin eri elementtien toimivuus suunnitellusti ja tiedon saatavuus (Kuvio 20).

7 PÄÄTELMÄT JA JATKOKEHITYS

Mielestäni projektin toteutus ja oman opinnäytetyön tavoitteet onnistuivat ja toteutuivat hyvin, vaikka etäisyysanturin käyttöönotto olikin hieman vaikeampaa. Tämän työn jälkeen ymmärrän enemmän IoT- sekä Big data-teknologioiden kokonaisuuksia. Molemmat teknologiat ovat hyvin laajoja, jatkuvasti kehittyviä sekä näitä voisi tutkia ja lukea loputtomiin. Kerätystä informaatiosta voidaan johtaa erilaisia analyysejä, joita voidaan käyttää edelleen tuomaan lisäarvoa asiakkaalle tai yritykselle. Monipuolisiin IoT-palveluihin voidaan lisätä erilaisia teknologioita, kuten esimerkiksi koneoppimista, lisättyä todellisuutta tai vaikkapa todennäköisyyksien laskentaa. Haluaisin jatkaa Big datan ja tiedon analysoinnin parissa omalla ajalla. Yleisesti tällaisen pienen IoT-projektin toteutus on hyvin helppoa ja tällaisesta projektista saa hyvän kokonaiskäsityksen kyseisestä tekniikasta. Teknologiapinon käyttäminen projektissa auttoi hahmottamaan omaa IoT-palvelua ja sen rakennetta. Oli mielenkiintoista käyttää Raspberry Pi-mikrotietokonetta sekä ohjelmoida Python-ohjelmointikielellä. Jatkokehityksenä projektille voisi olla toteutuksesta jääneiden anturien liittäminen palveluun ja mahdollisesti jonkin toisen tietoliikenneratkaisun kokeileminen, kuten esimerkiksi Raspberry Pin yhdistäminen alustalle LoRaWAN-tekniikalla tai ThingWorxin omalla EMS-ratkaisulla. Myös oman alustan ohjelmoiminen voisi olla hyvin mielenkiintoista.

Visuaalisen esityksen työstämiseen olisi voinut laittaa enemmän aikaa ja lisätä monimutkaisempia analysointeja. Projektin jatkokehityksenä voisi olla myös monipuolisempien elementtien ja ominaisuuksien lisääminen visualisaatioon. Kuitenkin tämän opinnäytetyön päätavoitteena oli tutustua ThingWorx-alustaan ja sen toimintoihin. Esineiden internetin alustoja on todella paljon ja eri ominaisuudet vaihtelevat alustoittain, kuitenkin periaatteet varmasti pysyvät samoina. ThingWorx Composer-alusta on helppokäyttöinen ja pienimuotoisten prototyyppien luominen on nopeaa, kuten PTC lupaakin sivuillaan.

Teoriaan liittyen, jos vuoteen 2020 mennessä esineiden määrä tulee olemaan noin 20–30 miljardia, tämä yhteydessä olevien esineiden ja asioiden määrä on huimaava! Tämä tuo varmasti lisää mahdollisuuksia, haasteita ja ongelmia esineiden internetin

kehityksessä. Esineiden internetiä voidaan käyttää hyväksi melkein missä vain, voidaan parantaa liiketoiminnan ja teollisuuden prosesseja, parantaa asiakkaiden käyttökokemuksia ja tuotteiden menestyvyyttä sekä markkinointia, luoda uusia innovaatioita, lisätä älykkyyttä kodin elektroniikkaan ja mahdollisesti myös tulevaisuudessa pelastaa ihmishenkiä. VTT:n mukaan suomalaiset yritykset ovat aloittamassa erilaisia esineiden internet hankkeita (Vesa 2016). Olen myös itse huomannut, että yritykset ovat yleisestikin alkaneet panostaa enemmän digitalisaatioon, mutta myös esineiden internetiin. Mielestäni esineiden internet on tässä vaiheessa vielä suunnattu paremminkin yritysten käyttöön kuin kuluttajille. Tähän vaikuttaa se, että kyseessä olevassa teknologiassa on parannettavaa ja tietoturvan pitäisi olla parempi, jotta saataisiin kuluttajille laadukas, hyödyllinen ja edullinen palvelu. Voisiko EU:n uusi tietosuoja-asetus olla hyödyksi kuluttajamarkkinointiin liittyvien ongelmien suhteen? Toisaalta juridiset ongelmat, kuten yksityisyyden suoja, ja esineiden internetin vähäinen hyöty kuluttajamarkkinoilla ovat varmasti yksi syy siihen, miksi esineiden internetin palveluita ei ole niin paljon kuluttajamarkkinoilla. Toinen syy on se, että osaamista näiden teknologioiden suhteen ei ole tarpeeksi. On todella mielenkiintoista seurata esineiden internetin ja Big datan kehitystä sekä todennäköisesti tulevaisuudessa olla myös itse mukana näihin teknologioihin liittyvissä projekteissa.

LÄHTEET

Adafruit. 2017a. PIR Motion Sensor. Viitattu: 16.2.2018. https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/overview

Adafruit. 2017b. DHT11, DHT22 and AM2302 sensors. Viitattu: 16.2.2018. https://learn.adafruit.com/dht/overview

Agarwal, T. 2018. ZigBee Wireless Technology Architecture and Applications. ElProCus. Viitattu: 9.1.2018. https://www.elprocus.com/what-is-zigbee-technology-architecture-and-its-applications/

Collin, J. & Saarelainen, A. 2016. Teollinen Internet. Helsinki. Talentum.

Elinkeinoelämän keskusliitto. 2016. EU-asetus henkilötietojen suojasta julkaistu lopullisessa muodossaan. Viitattu: 2.9.2017. https://ek.fi/ajankohtaista/uutiset/2016/05/12/hyotytietoa-yrityksille-eu-asetus-henkilotietojen-suojasta-julkaistu-lopullisessa-muodossaan/

Greenough, J. 2016. How the 'Internet of Things' will impact consumers, businesses, and governments in 2016 and beyond. Nordic Business Insider. Viitattu: 8.4.2018. http://nordic.businessinsider.com/how-the-internet-of-things-market-will-grow-2014-10?r=US&IR=T

IoT University. 2018. Fundamentals of IoT Development with ThingWorx. Viitattu: 5.3.2018. https://www.iotu.com/enrollment/student/fundamentals-of-iot-development-with-thingworx

Isotalo, A. 2017. WLAN-verkon suunnittelu ja testaus. Savonia-ammattikorkeakoulu. Viitattu: 6.4.2018. http://urn.fi/URN:NBN:fi:amk-201701191492

Kompella, K. 2015. A Guide to The Internet of Things. EContent. 38, 3, 30–31. Viitattu: 2.9.2017. http://www.econtentmag.com/Articles/Editorial/Feature/A-Guide-to-The-Internet-of-Things-102679.htm

Lewis, D. 2015. Is The Internet of Things IPv6 Ready? Forbes. Viitattu: 11.12.2017. https://www.forbes.com/sites/davelewis/2015/01/28/is-the-internet-of-things-ipv6-ready/#37285058546a

Management Today. 2014. Internet of Things – The Internet of Everything. Viitattu: 5.9.2017. http://www.managementtoday.co.uk/internet-everything/article/1291215

Mata, T. 2015. Esineiden internet standardit ja protokollat. Haaga-Helia ammattikorkeakoulu Oy. Viitattu 28.11.2017 http://urn.fi/URN:NBN:fi:amk-201505198874

McEwen, A. & Cassimally, H. 2014. Designing the Internet of Things. United Kingdom. Wiley.

Mitchell, B. 2017. What Is a Wide Area Network (WAN)? Lifewire. Viitattu: 11.12.2017. https://www.lifewire.com/wide-area-network-816383

ModMyPi. 2014. HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi. Viitattu 16.2.2018. https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi

Nazzaro, W. & Suscheck, C. 2010. New to User Stories? Scrum Alliance. Viitattu: 19.1.2018. https://www.scrumalliance.org/community/articles/2010/april/new-to-user-stories

Niemann-Ross, M. 2017. IoT is More Than Just Internet + Things. LinkedIn Pulse. https://www.linkedin.com/pulse/iot-more-than-just-internet-things-mark-niemann-ross

Pixabay. 2017. Viitattu: 12.3.2018. https://pixabay.com/fi/taustaa-vihre%C3%A4-kuvio-2551941/

Professional Services Close – Up. 2014. PTC purchases ThingWorx. Close Up Media, Inc. Viitattu: 11.12.2017. https://search-proquest-com.ezproxy.puv.fi/docview/1476526931?accountid=27304

Professional Services Close – Up. 2015. Elisa to deploy ThingWorx platform. Close Up Media, Inc. Viitattu: 11.12.2017. https://search-proquest-com.ezproxy.puv.fi/docview/1647874177?accountid=27304

PTC. 2017a. About PTC. Viitattu: 12.12.2017. https://www.ptc.com/en/about

PTC. 2017b. Connect. Viitattu 2.3.2018. https://developer.thingworx.com/capabilities/connect

Romunen, E. & Tikkanen, K. 2010. Langattomat lähiverkot. Case: WPK. Tampereen ammattikorkeakoulu. Viitattu: 6.4.2018. http://urn.fi/URN:NBN:fi:amk-2010121718567

Salo, I. 2014. Big data & pilvipalvelut. Jyväskylä. Docendo.

Saxena, S. 2017. BOLD (Big and Open Linked Data): What's Next? Library Hi Tech News. 34, 5, 10–13. http://www.emeraldinsight.com.ezproxy.puv.fi/doi/full/10.1108/LHTN-04-2017-0020

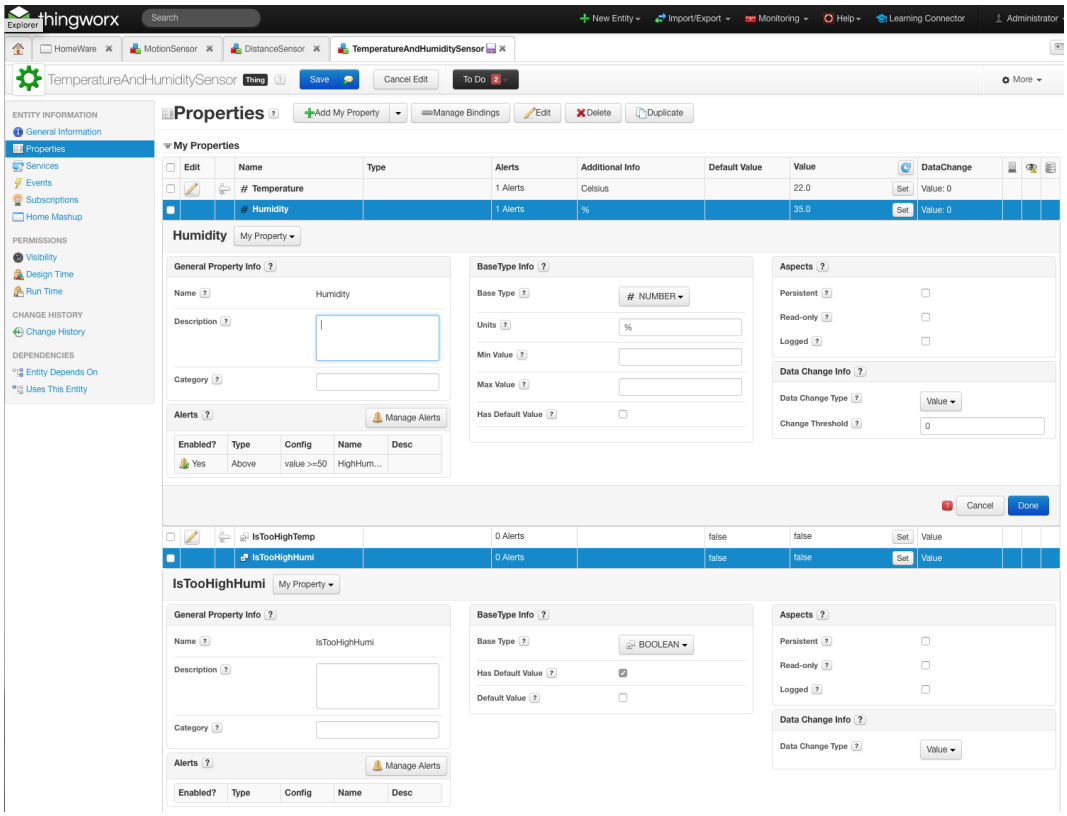
The MagPi Magazine. 2015. Raspberry Pi 3 Is Out Now! Specs, Benchmarks & More. Viitattu: 2.3.2018. https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/

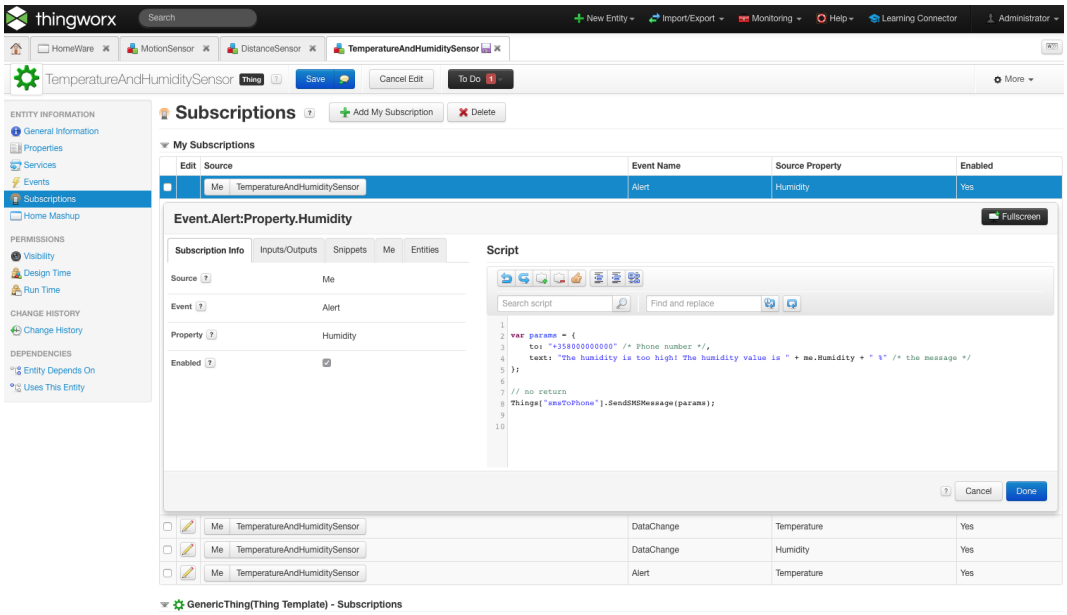
Vesa, J. 2016. Onko esineiden internet pelkkä kupla? Tekniikka & Talous. Metallitekniikka. 7–8. Viitattu 9.9.2017. http://www.tekniikkatalous.fi/tekniikka/metalli/onko-esineiden-internet-pelkka-kupla-6591827

Yassein, M. B., Mardini, W. & Khalil, A. 2016. Smart homes automation using Z-wave protocol. Engineering & MIS (ICEMIS), International Conference. 1–6. http://ieeexplore.ieee.org.ezproxy.puv.fi/document/7745306/

LIITE 1

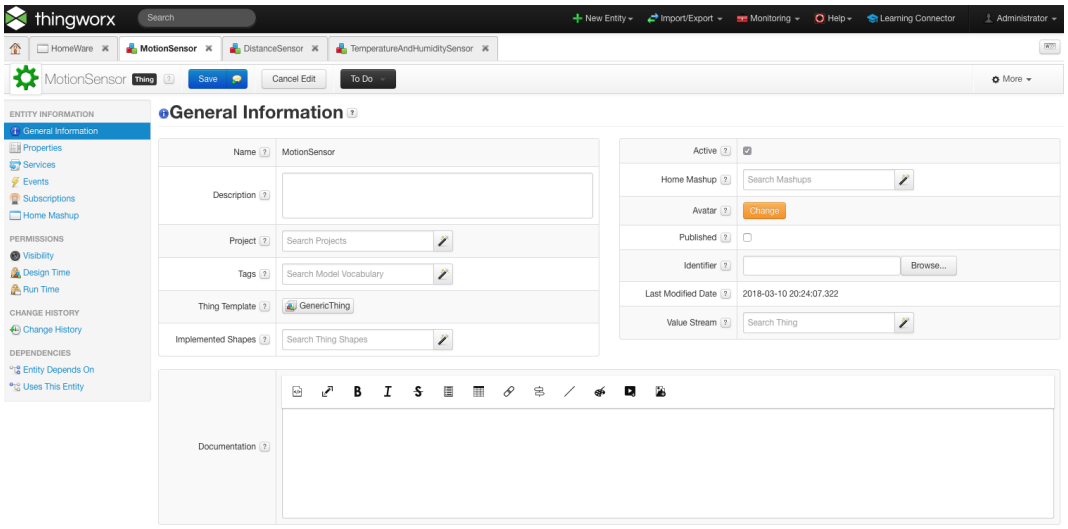
Ilmankosteus- ja lämpötila-esineen tiedot

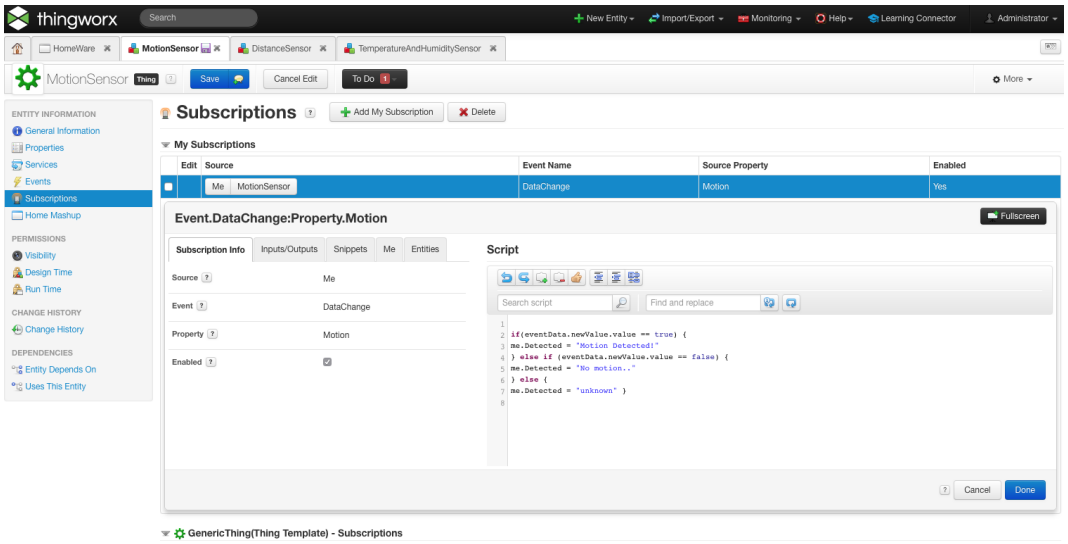




LIITE 2

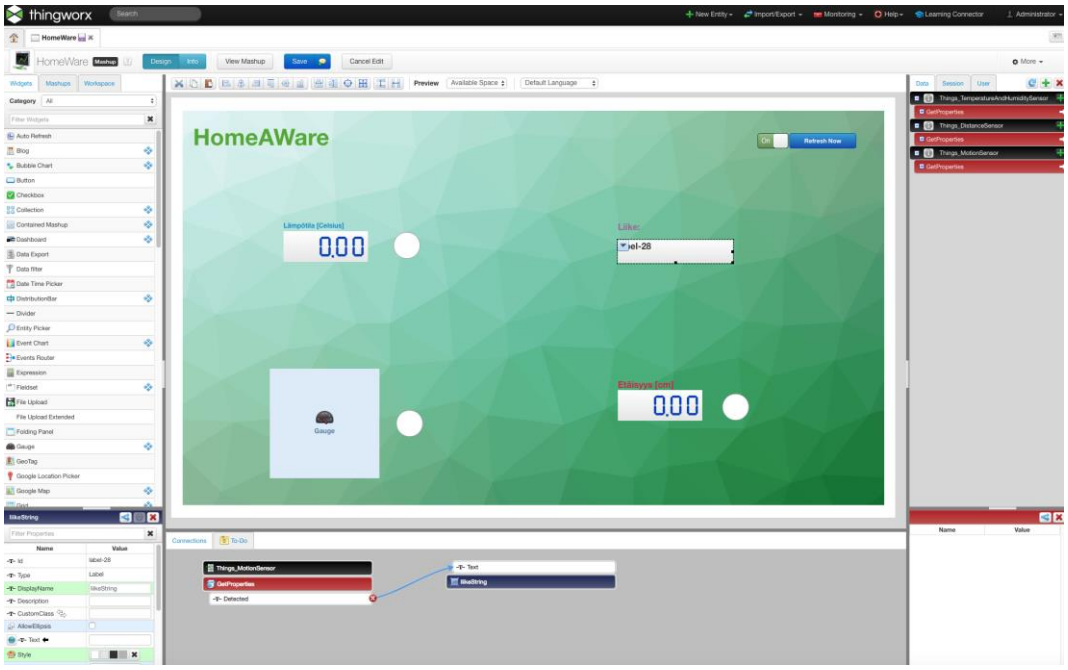
Liike-esineen tiedot

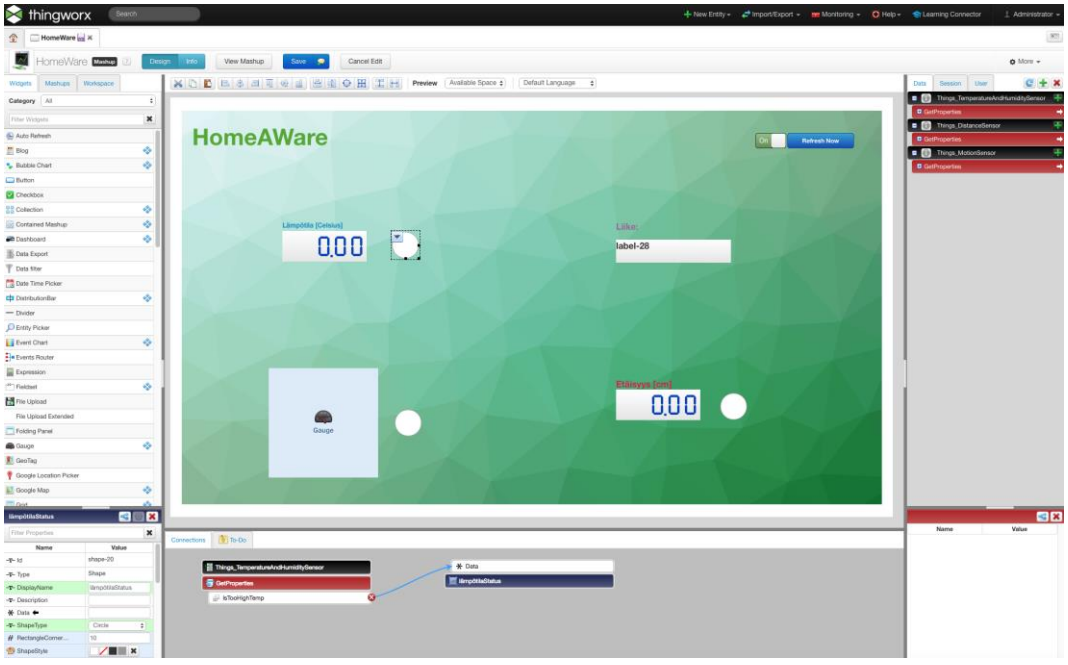




LIITE 3

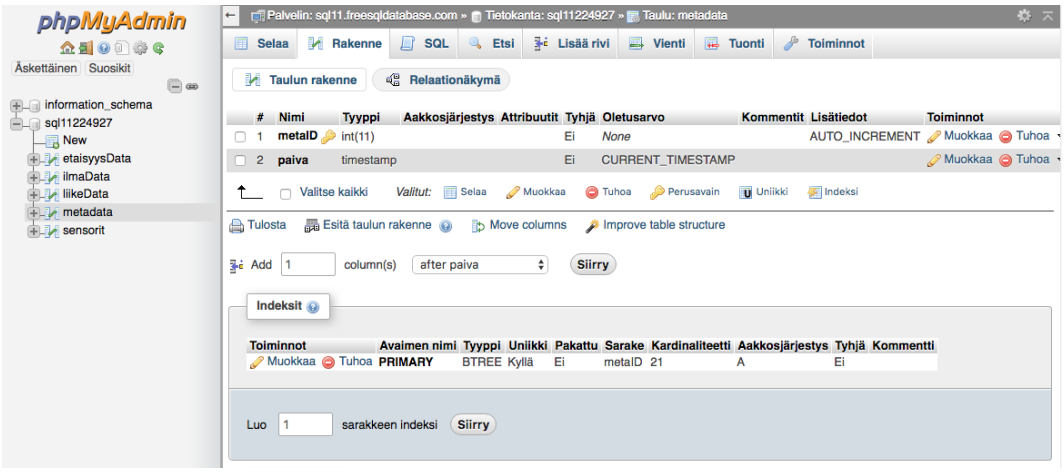
Visualisaation kuvakaappauksia

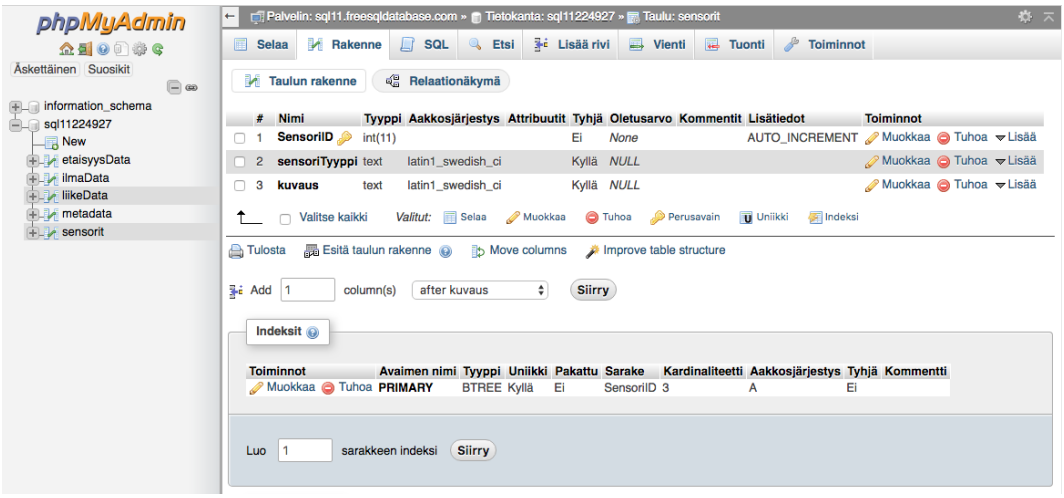




LIITE 4

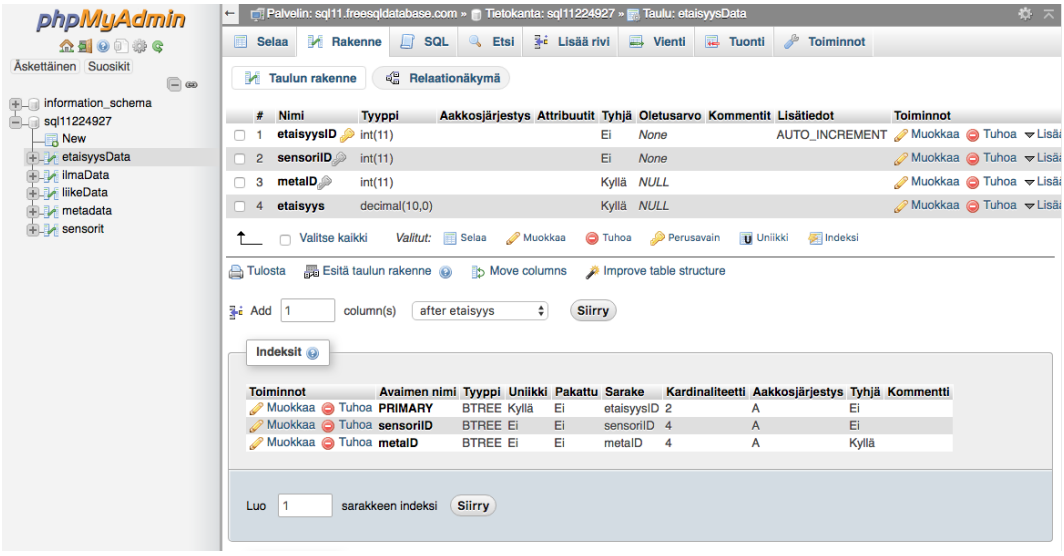
Tietokannan toteutetut metatiedot- ja sensorit-taulut

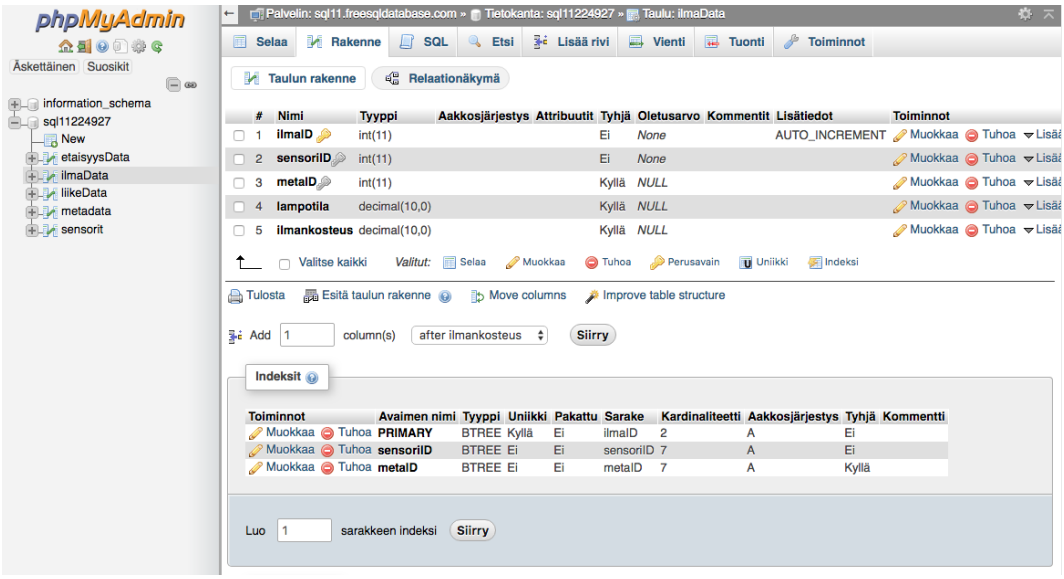




LIITE 5

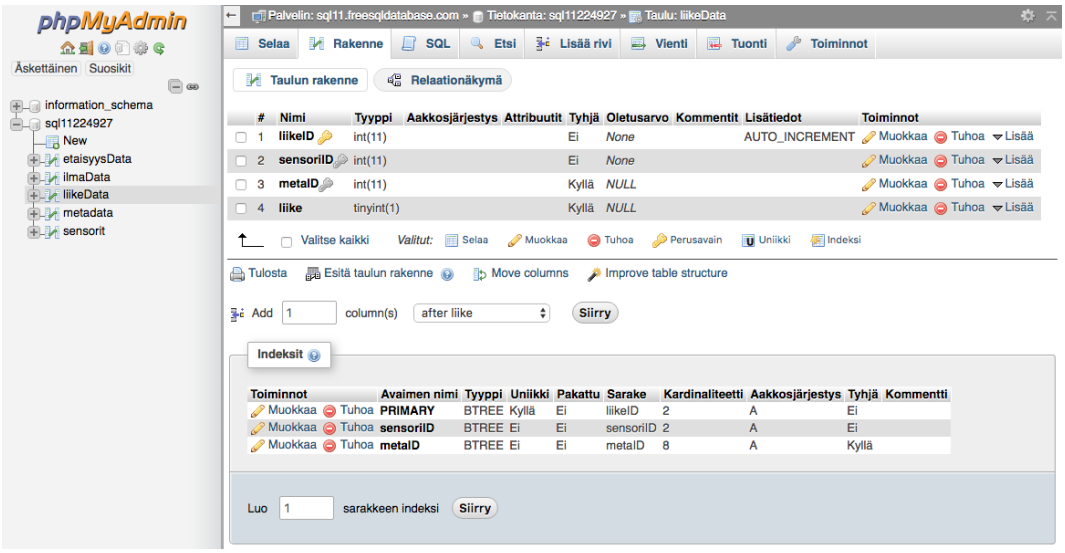
Tietokannan toteutetut etäisyys- sekä ilmankosteus- ja lämpötila-taulut





LIITE 6

Tietokannan toteutettu liike-taulu



LIITE 7

Anturien koodi; ohjelmakirjastot ja tietokannan metodit

```
import sys
import requests
import json
import pprint
import Adafruit_DHT
import time
from time import sleep
import RPi.GPIO as GPIO #Import GPIO library
import pymysql.cursors
import pymysql
import sys

#set GPIO Pins for Ultrasonic sensor
GPIO_TRIGGER = 38
GPIO_ECHO = 40
# Set sensor type : DHT11
sensor=Adafruit_DHT.DHT11
# Set GPIO DHT11 sensor is connected to
gpio=17
#PIR
GPIO.setmode(GPIO.BOARD) #Set GPIO pin numbering
pir = 26 #Associate pin 26 to pir sensor
GPIO.setup(pir, GPIO.IN) #Set pin as GPIO in
#US
GPIO.setup(GPIO_ECHO, GPIO.IN)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
```

```
def connect():
    # Connect to the database
    connection = pymysql.connect(host='localhost',
                                 user='user',
                                 password='pass',
                                 db='db_name',
                                 charset='utf8mb4',
                                 cursorclass=pymysql.cursors.DictCursor)
    cursor = connection.cursor()
    return connection, cursor

def executeMeta(connection, cursor, sql):
    cursor.execute(sql)
    connection.commit()
    meta = cursor.lastrowid
    return meta

def executeSQL(connection, cursor, sql, data):
    cursor.execute(sql, data)
    connection.commit()

def disConnect(connection):
    connection.close()
```

LIITE 8

Anturien koodi; REST JSON ns. “otsikot” ja etäisyyden laskenta

```
headersTemp = {
                'content-type': "application/json",
                'appkey': "appKey",
                'cache-control': "no-cache",
                'postman-token': "token"
              }
headersMotion = {
                 'content-type': "application/json",
                 'appkey': "appKey",
                 'cache-control': "no-cache",
                 'postman-token': "token"
               }
headersHumi = {
                'content-type': "application/json",
                'appkey': "appKey",
                'cache-control': "no-cache",
                'postman-token': "token"
              }
headersDistance = {
                   'content-type': "application/json",
                   'appkey': "appKey",
                   'cache-control': "no-cache",
                   'postman-token': "token"
                  }

```

```
def distance():
    # set Trigger to HIGH
    GPIO.output(GPIO_TRIGGER, True)

    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

    StartTime = time.time()
    StopTime = time.time()

    # save StartTime
    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()

    # save time of arrival
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()

    # time difference between start and arrival
    TimeElapsed = StopTime - StartTime
    # multiply with the sonic speed (34300 cm/s)
    # and divide by 2, because there and back
    distance = TimeElapsed * 17150

    return distance


print "Waiting for sensor to settle"
time.sleep(2) #Waiting 2 seconds for the sensor to initiate print "Detecting motion"

```

LIITE 9

Anturien koodi; lämpötila- ja ilmankosteusarvojen käsittely

```
    if humidity is not None and temperature is not None:
        #send the data to ThingWorx with REST PUT
        #Temperature
        dataTemp = {"Temperature":temperature}
        payload = json.dumps(dataTemp)
        response = requests.request("PUT", urlTemperature, data=payload, headers=headersTemp)
        #Humidity
        dataHumi = {"Humidity":humidity}
        payload = json.dumps(dataHumi)
        response = requests.request("PUT", urlHumidity, data=payload, headers=headersHumi)
        #print if possible
        print('Temp={0:0.1f}*C  Humidity={1:0.1f}%'.format(temperature, humidity))
        #and then lastly send data to also a database
        #connect
        conn, cur = connect()
        #metadata insert
        metaID = executeMeta(conn, cur, sqlMeta)
        sensoriID = 1
        # data insert
        data_word = (sensoriID, metaID, temperature, humidity)
        executeSQL(conn, cur, sqlTempHumi, data_word)
        #close connection
        disConnect(conn)
        time.sleep(2)
        time.sleep(0.1)
```

```
    else:
        #print if possible
        print('Failed to get temper and humidity reading. Try again!')
        temp = 0
        humi = 0
        #REST Put
        #Temperature
        dataTemp = {"Temperature":temp}
        payload = json.dumps(dataTemp)
        response = requests.request("PUT", urlTemperature, data=payload, headers=headersTemp)
        #Humidity
        dataHumi = {"Humidity":humi}
        payload = json.dumps(dataHumi)
        response = requests.request("PUT", urlHumidity, data=payload, headers=headersHumi)
        #connect
        conn, cur = connect()
        #metadata insert
        metaID = executeMeta(conn, cur, sqlMeta)
        sensoriID = 1
        data_word = (sensoriID, metaID, temp, humi)
        executeSQL(conn, cur, sqlTempHumi, data_word)
        #close connection
        disConnect(conn)
        time.sleep(2)
        time.sleep(0.1)
```

LIITE 10

Anturien koodi; etäisyysarvon käsittely

```
219     #then lastly read if there's something in front of the distance sensor
220     dist = distance()
221     if dist is not None:
222         print ("Measured Distance = %.1f cm" % dist)
223         dataDist = {"Distance":dist}
224         payload = json.dumps(dataDist)
225         response = requests.request("PUT", urlDistance, data=payload, headers=headersDistance)
226         conn, cur = connect()
227         #metadata insert
228         metaID = executeMeta(conn, cur, sqlMeta)
229         sensoriID = 3
230         #data insert
231         data_word = (sensoriID, metaID, dist)
232         executeSQL(conn, cur, sqlDistance, data_word)
233         #close connection
234         disConnect(conn)
235         time.sleep(1)
236     else:
237         distanceCalc = 0
238         dataDist = {"Distance":0}
239         payload = json.dumps(dataDist)
240         response = requests.request("PUT", urlDistance, data=payload, headers=headersDistance)
241         #connect
242         conn, cur = connect()
243         #metadata insert
244         metaID = executeMeta(conn, cur, sqlMeta)
245         sensoriID = 3
246         #data insert
247         data_word = (sensoriID, metaID, distanceCalc)
248         executeSQL(conn, cur, sqlDistance, data_word)
249         #close connection
250         disConnect(conn)
251         time.sleep(2)
```