

Sensorien liittäminen ja niiden toiminta Elisa IoT -palvelussa

Jesse Ala-Lahti

Opinnäytetyö

Marraskuu 2017

Tekniikan ja liikenteen ala

Insinööri (AMK), Tieto- ja viestintätekniikka

Tietoverkkotekniikka

Tekijä(t) Ala-Lahti, Jesse	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä marraskuu 2017
	Sivumäärä	Julkaisun kieli Suomi
	52	Verkojulkaisulupa Kyllä
Työn nimi Sensorien liittäminen ja niiden toiminta Elisa IoT -palvelussa		
Tutkinto-ohjelma Tieto ja viestintäteknikka, Tietoverkkotekniikka		
Työn ohjaaja(t) Kokkonen Tero, Rantonen Mika		
Toimeksiantaja(t) Väänänen Olli (Jyväskylän Ammattikorkeakoulu JAMK)		
<p>Tiivistelmä</p> <p>Työssä haluttiin kartuttaa Thingworx-palvelun tarjoamia mahdollisuuksia ja palveluita. Haluttiin myös avata sensorien datan matkaa aina visuaalisesti esitettäväksi asti. Tämä polku esitetään mahdollisimman yksinkertaisesti ja selkeästi, jotta seuraavissa tutkimuksissa olisi mahdollisimman paljon taustatietoa kyseisestä palvelusta, sillä suomenkielisiä ohjeita palvelun käyttöön ei ollut.</p> <p>Datan matkaa avataan mahdollisimman paljon ja sen visualisointiin ja visualisoinnin tueksi tehtyjen palveluiden läpikäynti on tärkeää työlle. Työssä käyt MQTT-protokollaa, jonka toimintaperiaatteita on myös avattu työssä.</p> <p>Työssä saatiin selville sensorien tiedon lähettämiseen eri tarpeet ja niiden toiminta. Eri sensorialustoja myös vertailtiin työssä ja näiden eri rajoituksia kartutettiin. Myös niitä palveluita, joita ei käytetty työssä, mutta joita Thingworx tarjoaa, käytiin työssä. Nämä palvelut olivat kuitenkin tärkeitä käydä muitten tulevien projektien kannalta, jonka takia niitä nähtiin tarpeelliseksi esitellä ja opastaa niiden toimintaan.</p> <p>Työn valmistuttua sen tarkoitus on muodostaa pohja tuleville opinnäytetöille, jotka käyttävät samaa palvelua. Tarjolla olevat palvelut on tutkittu ja niistä on tarjolla tarpeellista informaatiota työn ansiosta. Voidaan myös todeta Thingworxin tukevan hyvin MQTT-protokollaa lisäosien lataamisen jälkeen ja antaa hyvät mahdollisuudet isompientkin projektien läpivientiin aina dokumentoinnista ja wikistä aina pohjapiirustuksien ja dataa analysoivien osien muodossa.</p>		
<p>Avainsanat (asiasanat) IoT, Internet of Things, Thingworx, QoS, ESP8266, DHT11, MQTT</p>		
Muut tiedot		

Author(s) Ala-Lahti, Jesse	Type of publication Bachelor's thesis	Date November 2017 Language of publication: Finnish
	Number of pages 52	Permission for web publication: Yes
Title of publication Connecting sensors and operating them in Elisa IoT service.		
Degree programme Information and Communication Technology, Data Network Technology		
Supervisor(s) Kokkonen Tero, Rantonen Mika		
Assigned by Väänänen Olli (JAMK University of Applied Sciences)		
Abstract <p>In this thesis, data was collected about services offered by Thingswork. The whole road from data provided by the sensor, is visualized and shown in Thingworx. The thesis attempts to show the road as simply and clearly as possible; thus, it is easy to use it as a guide to Thingworx for later users.</p> <p>Most of the services offered by Thingworx are demonstrated in the thesis. The thesis focuses more on basic services needed for basic projects. Some services helping to organize data and project or users are presented.</p> <p>Some sensors have to meet special needs and requirements in order to work correctly. The information about these sensors and some boards for MQTT projects are shown and commented on. Besides hardware, few software solutions about MQTT data transmitting are documented and commented. These hardware and software solutions seemed important to document for future projects with Thingworx since there is limited information available for projects in Finnish.</p> <p>A deal of information about Thingworx was gathered: from adding pictures to documenting and from making wiki pages to analyzing data. This information provides a basic understanding of the services, and how-to information about Thingworx, and how MQTT protocol works with different services.</p>		
Keywords/tags (subjects) IoT, Internet of Things, Thingworx, QoS, ESP8266, DHT11, MQTT		
Miscellaneous		

Sisältö

Lyhenteet	5
1 Johdanto	6
2 Tutkimusasetelma	6
3 Työn tekniset osa-alueet.....	8
3.1 Projektin osat ja niiden toiminta.....	8
3.1.1 MQTT.FX	9
3.1.2 MyMQTT	9
3.1.3 Arduino Uno.....	12
3.1.4 DHT11	13
3.1.5 ESP8266	13
3.1.6 NodeMCU	14
3.2 MQTT	15
3.2.1 Message Queue Telemetry Transport.....	15
3.2.2 MQTT-verkon osat	15
3.2.3 Yhteyden muodostus.....	16
3.2.4 MQTT ja tietoturva	17
3.2.5 Käyttäjän ID	18
3.2.6 Käyttäjätunnus ja salasana	18
3.2.7 X509-sertifikaatti	18
3.2.8 Sisällön salaus	19
3.3 Topic ja sen rakenne työssä	19
3.4 MQTT ja QoS.....	20
3.4.1 QoS 0.....	20
3.4.2 QoS 1.....	21
3.4.3 QoS 2.....	21
4 Elisa IoT.....	22

	2
4.1 Yleistä	22
4.2 ThingWorx- käyttöliittymä	23
4.3 Materiaalin luomisen vaihtoehdot.....	23
4.3.1 Thing	24
4.3.2 Thing Template	25
4.3.3 Project.....	26
4.3.4 Model Tag	26
4.3.5 Mashup	27
4.3.6 Gadget	28
4.3.7 Media	28
4.3.8 Value Stream	29
4.3.9 Application key	30
4.3.10 New User	31
4.3.11 User Group.....	32
4.3.12 Wiki	32
4.3.13 Style Definition	34
5 Projektin toteutus	34
5.1.1 MQTT lisäosan lataaminen	34
5.1.2 Projektin osien luonti.....	35
5.1.3 MQTT- Broker	37
5.1.4 Esineen arvojen luominen	37
5.1.5 Mashupin luonti.....	42
5.1.6 Datan lähetys palveluun	43
5.1.7 Datan liittäminen mashuppiin	45
6 Testaus ja tulokset.....	46
7 Pohdinta.....	47
Lähteet	49

Liitteet..... 51

Liite 1. Antureiden datan emulimiseen käytetty skripti

Kuviot

Kuvio 1. Projektitalon pohjapiirustus numeroituna	7
Kuvio 2. MQTT.FX-sovellus lähettää sensoritietoja brokerille	9
Kuvio 3. MyMQTT-sovelluksen asetukset yhdistämistä varten	10
Kuvio 4. MyMQTT saa lähetetyn viestin myös vastauksena, sillä on tilannut kyseisen kanavan	11
Kuvio 5. MyMQTT on vastaanottanut MQTT.FX sovelluksesta lähetetyn viestin.....	11
Kuvio 6. Arduino Uno ilman lisäosia.....	12
Kuvio 7. DHT11-lämpötilasensori.....	13
Kuvio 8. ESP8266-01 mahdollistaa yhdistämisen verkkoon	14
Kuvio 9. NodeMCU on yksinkertainen ja tehokas alusta IoT- projekteille.....	15
Kuvio 10. Projektin rakenne	16
Kuvio 11. Yhteyden muodostus MQTT- protokollan mukaan.....	17
Kuvio 12. Opinnäytetyön kanavien topologia	20
Kuvio 13. QoS 0:n toiminta.....	21
Kuvio 14. QoS 1:n toiminta.....	21
Kuvio 15. QoS 2 toiminta.....	22
Kuvio 16. Thingworx-sovelluksen aloitusruutu	23
Kuvio 17. Mahdollisuudet projektin osien luontiin	24
Kuvio 18. Esineen luonnin asetukset.....	25
Kuvio 19. Templaten luonnin asetukset.....	26
Kuvio 20. Asetukset projektin luonnissa	26
Kuvio 21. Mahdollisuudet tagien luontiin	27
Kuvio 22. Tyhjä mashup-alusta	28
Kuvio 23. Esineen luonti mashupia varten	29
Kuvio 24. Tietojen tallennuspaikan asetusten täyttäminen	30
Kuvio 25. Asetukset avaimen luontiin.....	31

Kuvio 26. Käyttäjän luonti	31
Kuvio 27. Ryhmän luonti	32
Kuvio 28. Wikin asetukset	33
Kuvio 29. Wikin visuaalinen luonti asetusten asettamisen jälkeen.....	33
Kuvio 30. Graafisen ilmeen suunnittelu	34
Kuvio 31. MQTT-lisäosan tiedostojen tuonti.....	35
Kuvio 32. Projektin luonti ja tietojen asetus	36
Kuvio 33. Projektiin liitetyt tiedostot	36
Kuvio 34. Brokerin yhteystiedot ja tiedot	37
Kuvio 35. MQTT-osien asetusten asettaminen	38
Kuvio 36. Tietojen hakemisen asetukset.....	39
Kuvio 37. Tietomuodot luotaville tiedoille.....	40
Kuvio 38. Tarkemmat polut tietojen hakemiselle	40
Kuvio 39. Brokerin asetukset thingworxissa	41
Kuvio 40. Thingworx on onnistuneesti yhdistänyt brokeriin	41
Kuvio 41. Tietojen liittäminen kuvaan mashupissa.....	42
Kuvio 42. MQTT.FX:n asetukset	43
Kuvio 43. Saunan mittari mashupissa	44
Kuvio 44. MQTT.FX:n toiminta komentorivillä.....	45
Kuvio 45. Mashuppiin lisätyt tiedot ja niiden nimet	46
Kuvio 46. Mashup toiminnassa	47

Lyhenteet

IoT	Internet of Things
MQTT	Message Queue Telemetry Transport
QoS	Quality of Service
TCP/IP	Transmission Control Protocol / Internet Protocol
USB	Universal Serial Bus
Wi-Fi	Wireless local area network

1 Johdanto

Työssä tutkittiin lämpötila- ja kosteusanturien toimintaa ja niiden tiedon siirtämistä verkossa toimivalle palvelimelle. Työn suunnittelu aloitettiin jo vuonna 2016, kun paikkakunnassa, jossa projektirakennus sijaitsee, löydettiin monia rakennuksia, joissa epäiltiin ja todettiin olevan hometta. Tällä projektilla haluttiin ennaltaehkäistä mahdollisia kosteusvaurioita rakenteissa käyttämällä kosteusanturista saatua dataa. Myös lämmityskustannuksia haluttiin tiputtaa optimoimalla lämpötila oikeaksi rakennuksen osissa. Tähän liitettynä oma mielenkiintoni Internet of Things (IoT) -ratkaisuihin sai aikaan työn suunnittelun ja aloittamisen.

IoT on Message Queue Telemetry Transport (MQTT)-protokollaa käyttävä termi, jolla tarkoitetaan esineiden internetiä. MQTT mahdollistaa nopean ja vähän sähköä kuluttavan tavan lähettää tietoa. Tämä on tuonut mahdollisuuksia lisätä eri palveluita verkkoon ja parantaa näin laitteen hallittavuutta etänä verkosta käsin. Internettiin voidaan lisätä esimerkiksi jääkaappi, joka lähettää lämpötilansa ja sisällön tiedot suoraan omistajan puhelimeen.

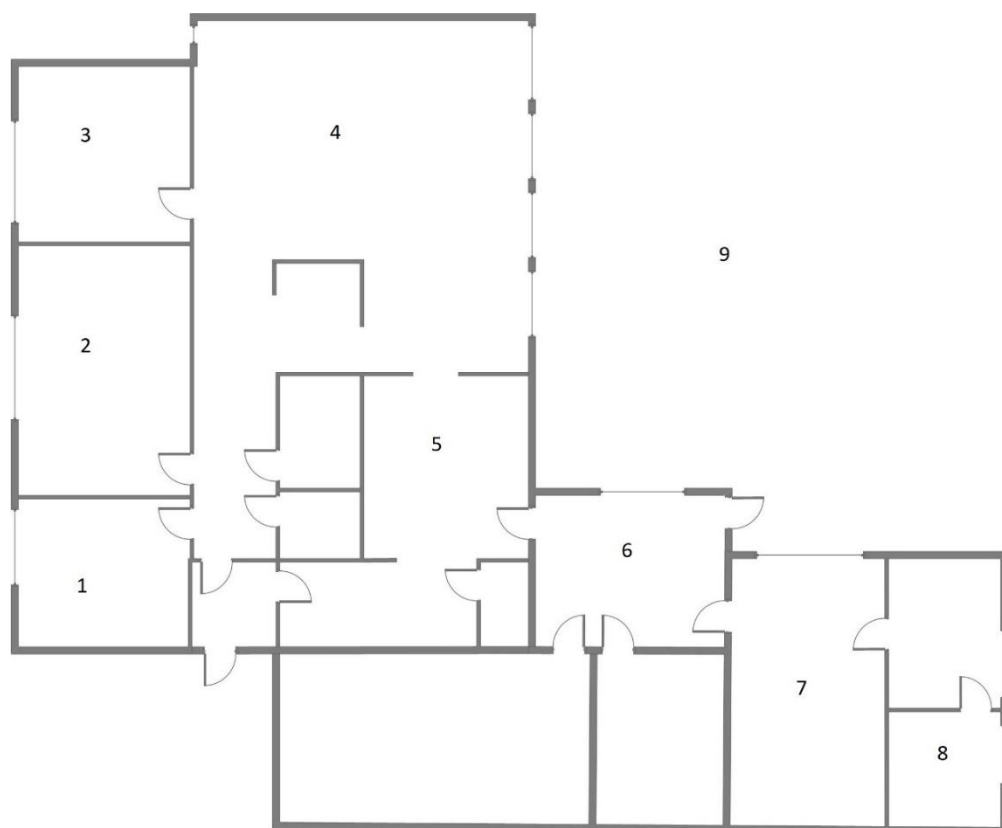
Työn alustaksi valittiin Thingworx ratkaisu, joka tarjosi hyvät resurssit projektin tekemiseen. Toinen vaihtoehto työn pohjaksi olisi ollut Bluemix, joka on IBM:n tarjoama palvelu. Tähän ei kuitenkaan päädytty, sillä Jyväskylän ammattikorkeakoululla oli käytössä Elisan tarjoama Thingworx alusta, josta haluttiin saada tietoa tämän opinnäytetyön avulla.

2 Tutkimusasetelma

Työn kohteena toimi vuonna 1974 rakennettu omakotitalo. Sisätilaa oli yhteensä 210 m². Tältä alueelta mitattiin lämpötilaa ja kosteutta yhteensä yhdeksästä eri kohdasta.

Mittauskohdat valittiin siten, että mahdollisimman hyödyllistä dataa saatiin. Mitattavat kohteet olivat seuraavat ja ne on merkitty myös kuvioon 1.

1. Huone 1
2. Huone 2
3. Huone 3
4. Olohuone
5. Keittiö
6. Välihuone
7. Takkahuone
8. Välikatto saunan yläpuolella
9. Ulkolämpötila



Kuvio 1. Projektitalon pohjapiirustus numeroituna

Tutkimuksen laatu on toimintatutkimus, jossa haluttiin käydä läpi koko toiminnan kaari. Selvitys tämän opinnäytetyön toimintaperiaatteista ja niiden dokumentoimisen on tarkoitus luoda hyvä pohja tähän tutkimuksen pohjautuville jatkotutkimuksille. (Koivula, U. Suihko, K. & Tyrväinen, J. 2002, 18-22)

Opinnäytetyön lopuksi kappaleessa 7 tuloksia arvioidaan ja pyritään vetämään johtopäätöksiä siitä, päästiinkö asetettuihin tavoitteisiin. Johtopäätösten vetäminen on tärkeää, sillä muuten opinnäytetyön tuloksista ei saada mitään suoranaista helposti omaksuttavaa tietoa, vaan työn tulosten vetäminen yhteen jää tyystin pois. (Hirsijärvi, S. Remes, P. Sajavaara, P. 2007, 224-225)

Mittaamisen toteuttamiseen oli monta vaihtoehtoa. Alkuperäinen suunnitelma oli toteuttaa mittaus Arduino Uno -mikrokontrollerilla. Tähän oltaisiin yhdistetty ESP8266, joka mahdollistaisi yhdistämisen langattoman verkkoon. Sensorina toimisi DHT11, joka on halvempi, mutta hiukan epätarkempi kuin DHT22. Toisena vaihtoehtona toteutukselle oli NodeMCU, jossa yhdistyy Arduinon ohjelmointi ja ESP8266:den uudempi versio. Tämä vähentäisi erillisten kytkentöjen tekemistä ja pienentäisi näin virheiden mahdollisuutta. Viimeisenä vaihtoehtona oli käytössä MQTT.FX, jolla voidaan lähettää tietoja sensorin tapaan eteenpäin. Kaikki sensorit ja ohjelmat käydään läpi tarkemmin luvussa 3.

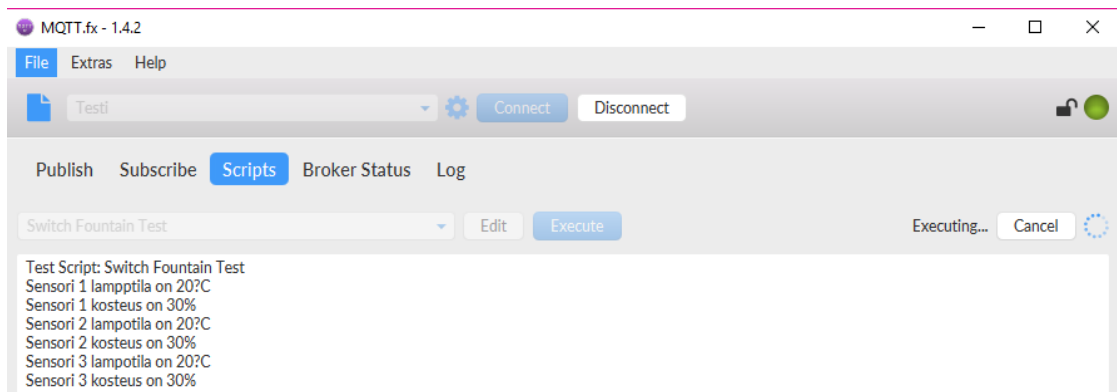
3 Työn tekniset osa-alueet

3.1 Projektin osat ja niiden toiminta

Opinnäytetyössä käytetään monia työkaluja, jotka eivät välttämättä ole kaikille tuttuja. Teknisten osien käsittelyssä käydään läpi, millä apuvälineillä lähetetään dataa eteenpäin Thingworx- palveluun. Osa lähetyksen mahdollistavista tavoista on fyysisiä ja osa lähetyksistä voidaan hoitaa joko tietokoneella tai kännykällä olevalla ohjelmalla.

3.1.1 MQTT.FX

MQTT.FX on sovellus, jolla voidaan lähettää viestejä MQTT- protokollaa käyttäen. Ohjelmassa on graafinen käyttöliittymä ja sillä voidaan lähettää yksittäisiä viestejä, mutta myös Javascript- skriptejä, joilla pitemmät viestit voidaan ajaa ajoitetusti ja tiettyssä järjestyksessä. Graafisen käyttöliittymän voi nähdä kuviossa 2. Viestejä voidaan myös vastaanottaa ohjelmaan käyttämällä Subscribe- vaihtoehtoa. Ohjelman on suunnitellut Jens Deters ja se on ladattavissa osoitteesta <http://mqttfx.iensd.de/>. Ohjelmaa käytettiin demoamaan viestien lähetystä fyysisten sensorien sijaan. (Hilitski, F. 2016.)

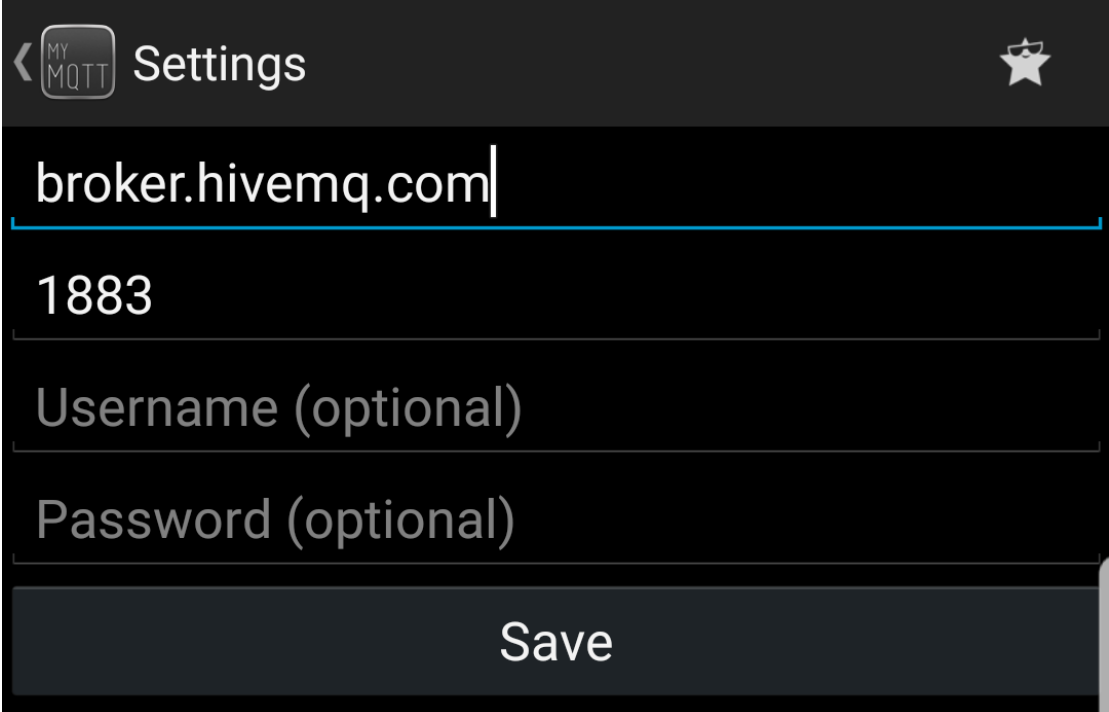


Kuvio 2. MQTT.FX-sovellus lähettää sensoritietoja brokerille

3.1.2 MyMQTT

MyMQTT on Androidille tarkoitettu sovellus, jolla voidaan lähettää ja vastaanottaa MQTT- viestejä. Työssä käytetään palvelua lähettämään numero 1, jos sauna on

päällä, ja numeron 0, jos sauna on pois päältä. Kuviossa 3 syötetään brokerin tiedot MyMQTT-sovellukseen, jotta saamme lähetettyä ja vastaanotettua tietoja brokerilta.



Settings

broker.hivemq.com

1883

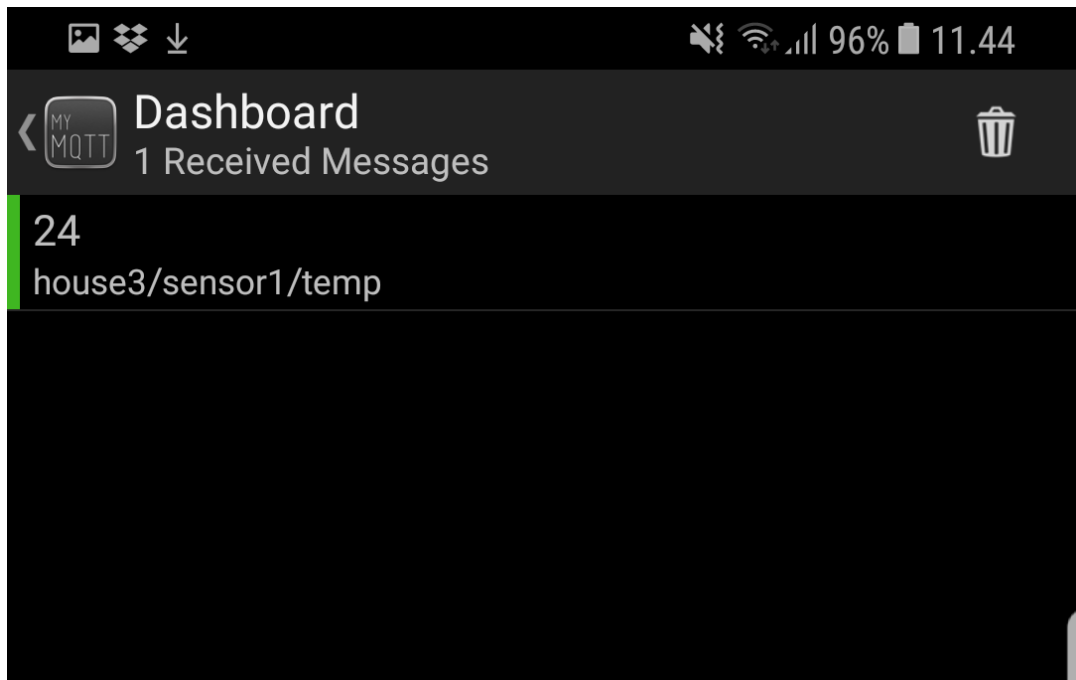
Username (optional)

Password (optional)

Save

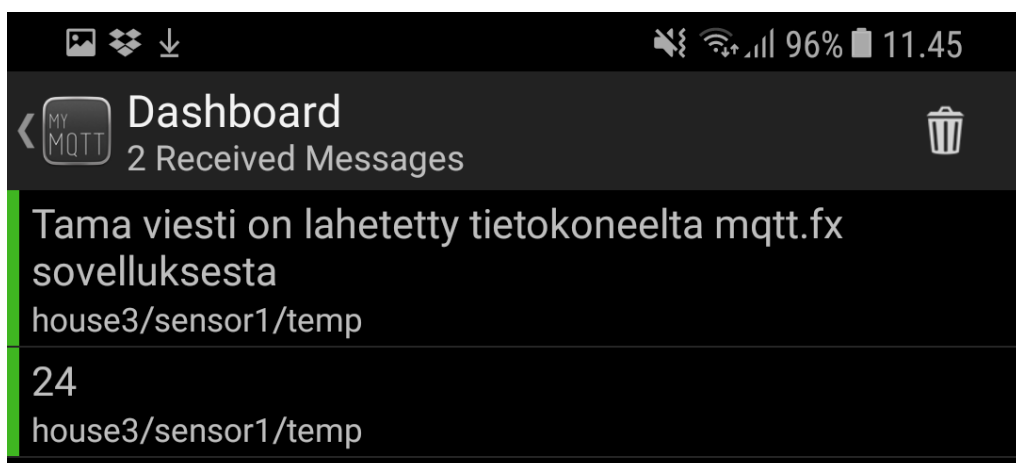
Kuvio 3. MyMQTT-sovelluksen asetukset yhdistämistä varten

Yhdistämisen jälkeen kirjoitetaan haluttu viesti ja lähetään sen house3/sensor1/temp topickiin, sillä palvelun toimivuutta vain testataan. Viestinä toimii numero 24, jolla haluttu esittää huoneen lämpötilaa, joka näkyy kuviossa 4.



Kuvio 4. MyMQTT saa lähetetyn viestin myös vastauksena, sillä on tilannut kyseisen kanavan

Saadaan myös testattua palvelun toimivuutta, kun tilataan house3/sensor1/temp topicin. On yhdistetty myös tietokoneella toimivan mqtt.fx- palvelun brokeriin ja lähetetään viesti. Koska vastaanotto onnistuu myös puhelimen palvelulla, niin kuin nähtiin myös kuviossa 5, voidaan olla varmoja sen toiminnasta.



Kuvio 5. MyMQTT on vastaanottanut MQTT.FX sovelluksesta lähetetyn viestin (MyMQTT 1.0. 2017.)

3.1.3 Arduino Uno

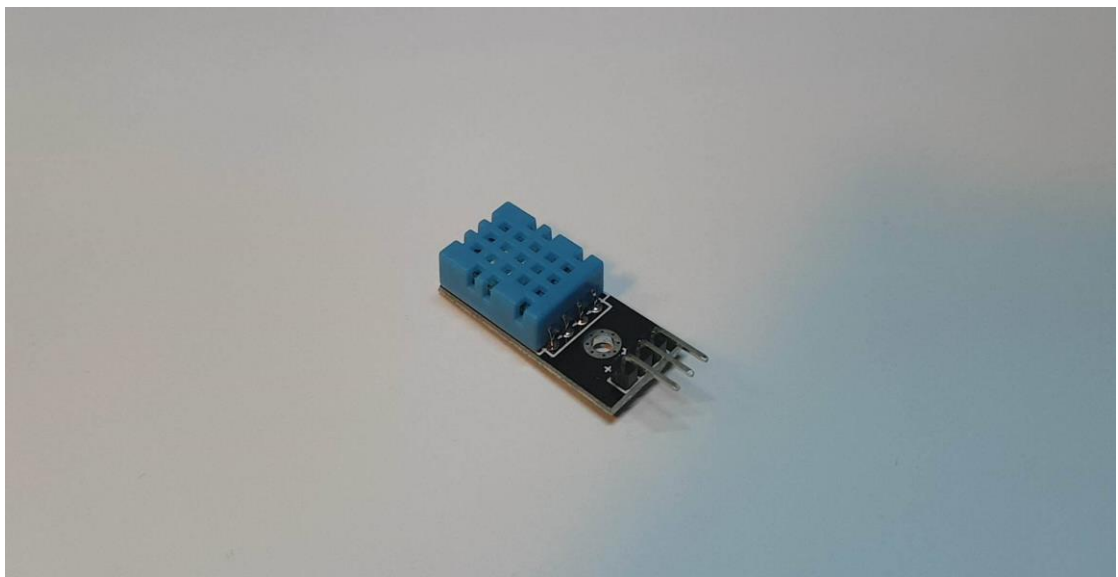
Arduino on ohjelmoitava mikrokontrollerialusta. Arduinoja on monta eri mallia ja tässä työssä käytettiin mallia UNO. Tämä valittiin työhön hyvien liittämismahdollisuuksien ja hyvien verkossa olevien ohjeiden perusteella. Koska UNO on pienempiin versioihin verrattaen tehokas, haluttiin sillä varmistaa myös tulevaisuudessa liitettävien komponenttien liitännämahdollisuus ja mikrokontrollerin tehokkuuden riittävyys. Arduino UNO oli myös nopeasti saatavilla varastosta ja suhteessa edullinen. Arduino UNOon pystyttiin liittämään maksimissaan 13 digitaalista sisääntuloa, mikä mahdollistaa lisäsensorien liittäminen tulevaisuudessa. Myös virransaantiin on erillinen ulkoisen virransaannin mahdollistava portti. Virran voi myös syöttää USB B -porttiin, joka näkyy kuviossa 6. (Arduino Uno REV3. 2017.)



Kuvio 6. Arduino Uno ilman lisäosia

3.1.4 DHT11

DHT11 on pieni kosteus- ja lämpötilasensori, joka on kooltaan noin pikkurillin pään kokoinen. Sensori toimii 3.3v – 5.5v jännitteellä ja antaa tuloksen 16- bitin resoluutiolla ja 5% virhemarginaalilla digitaalisessa muodossa. Komponentissa on 4 pinniä, joita käytetään komponentin kanssa kommunikointiin ja virransyöttöön. Koska neljästä käytössä olevasta pinnistä vain kolme oli käytössä, käytettiin työssä erillistä piirilevyä, johon sensorin oli kiinnitetty. Piirilevyssä pinnien määrä on muutettu kolmeen, joten vain tarpeellisia pinnejä käytettiin. DHT11- moduuli valittiin sen helpon saatavuuden ja hintaansa nähden hyvän tarkkuuden takia. Vaihtoehtoina olisi ollut DHT22, joka on pikkuveljeään DHT11:sta tarkempi, mutta yli kaksi kertaa kalliimpi, joten päädyttiin halvempaan DHT11- versioon. Kuviossa 7 on DHT11 sensorin. (Lady, A 2017.)

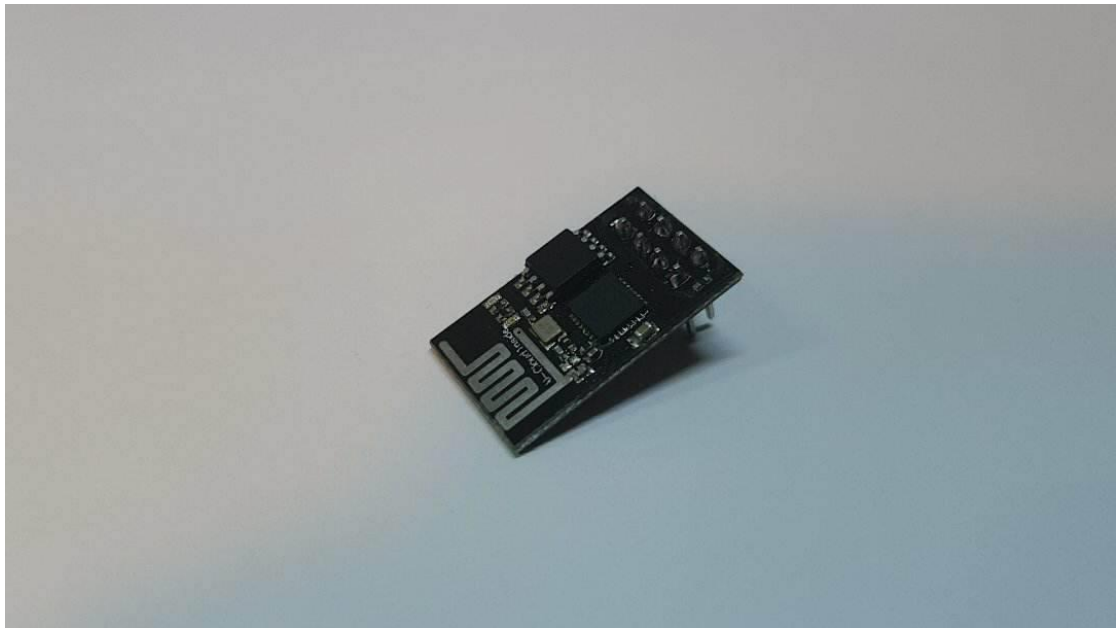


Kuvio 7. DHT11-lämpötilasensori

3.1.5 ESP8266

ESP8266 on kustannustehokas Wi-Fi moduuli, joka tukee TCP/IP:tä. ESP:stä on yli 14 versiota, ja jokainen malli on kooltaan ja rakenteeltaan hiukan erilainen. Työssä käytettiin isointa ja ensimmäistä versiota eli ESP-01:stä. Tässä mallissa on kiinnitettyä

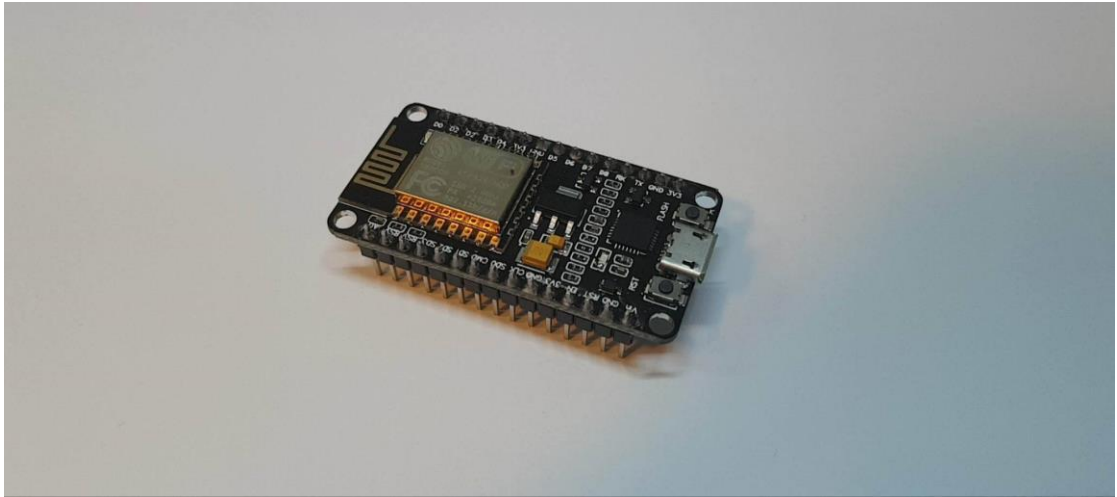
valmiiksi 8 kpl pinnejä, jotka ovat kuviossa 8 moduulin alla. Näistä pystytään ohjaamaan moduulia. Pinnien sijoittelu on kuitenkin todella huono, eikä moduulia pysty tällaisenaan käytetään testausalustassa. ESP on täydellinen pari Arduinoon, koska vuonna 2004 tulleen päivityksen mukaan moduulia voidaan ohjelmoida suoraan toisen ohjelman kautta, joten sen ohjelmointi käy Arduinin kanssa ja poistaa erillisten ohjelmien tarpeen. (Benchoff 2014.)



Kuvio 8. ESP8266-01 mahdollistaa yhdistämisen verkkoon

3.1.6 NodeMCU

NodeMCU yhdistää samaan pakettiin ESP8266-12- sirun, joka on paranneltu versio ensimmäisen version ESP8266:sta. Tämä vähentää ylimääräisten piuhojen vetämisen, kun yhdistetään monta eri komponenttia yhteen. Liittäminen tietokoneeseen on myös helpompaa micro-usb-kaapelin avulla. Koska verkkoyhteys onkin mahdollistettu jo suoraan laitteessa, on yhteensopivuusongelmien ilmeneminen vähäistä. Kuviossa 9 on NodeMCU. (Benchoff 2015.)



Kuvio 9. NodeMCU on yksinkertainen ja tehokas alusta IoT- projekteille

3.2 MQTT

3.2.1 Message Queue Telemetry Transport

Message Queue Telemetry Transport (MQTT) on kevyt viestintäprotokolla, joka toimii TCP/IP protokollan päällä. MQTT suunniteltiin erillään olevan kohteen yhdistämiseen. Koska lähetettävä paketti on pieni, soveltuu se hyvin kohteisiin, joissa lähetyskaista on rajallinen. Ensimmäinen versio protokollasta julkaistiin vuonna 1999. Viimeisin versio kirjoitushetkellä on MQTT versio 3.1.1, joka on julkaistu 29.10.2014. (OASIS Open 2014.)

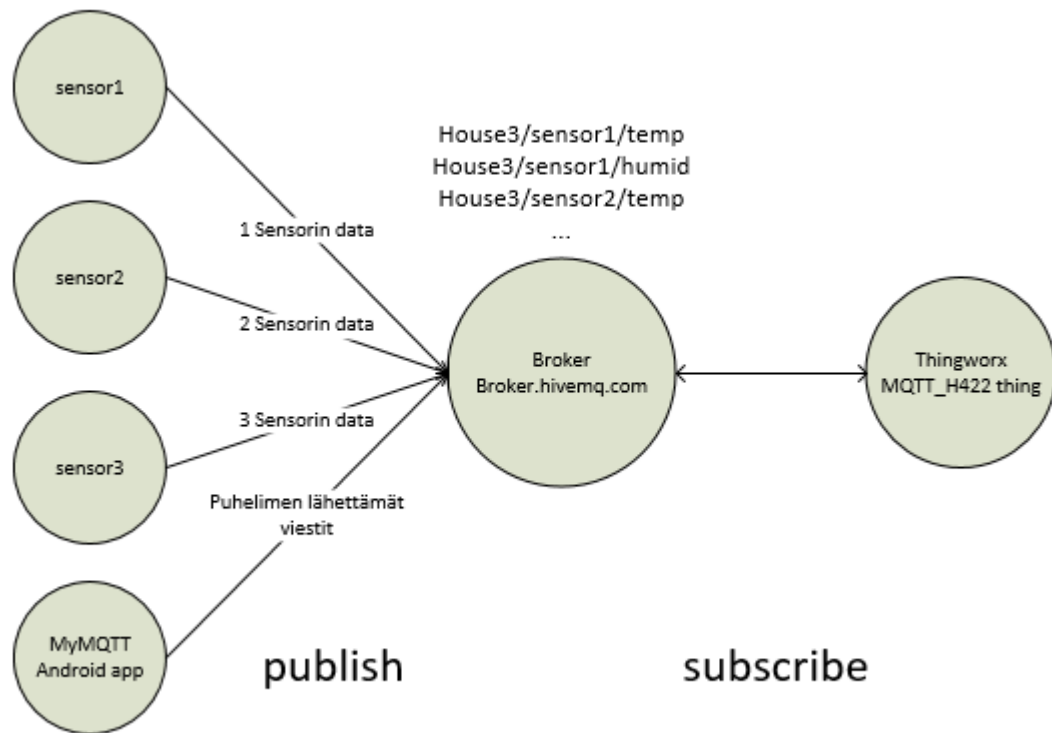
3.2.2 MQTT-verkon osat

MQTT-verkkoon kuuluu aina kolme osaa:

1. Tiedon lähettäjä eli publisher, joka julkaisee tiedon halutulle palvelimelle.
2. Rikkoja eli broker. Tämän tehtävä on vastaanottaa publisherin lähettämät viestit, säilöä ne ja jakaa ne niitä tilaaville tahoille.
3. Tilaaja eli describer voi olla ohjelma tai laite, joka haluaa kyseisen tiedon Brokerilta. Aina, kun broker saa tiedon sen julkaisijalta, se jakaa tiedon automaattisesti sen tilaajien kanssa. (MQTT Essentials Part 3. 2017.)

Kuviossa 10 sensorit sensor1, sensor2, sensor3 ja MyMQTT jakavat tietoja halutulle brokerille, joka on tässä tapauksessa broker.hivemq.com. Tiedot tallennetaan ennaltamäärättyihin topiceihin. Tämän jälkeen MQTT_H4222 thing saa tiedot brokerilta

ennalta tilatuista topicista. Tiedon voi saada kuka tahansa, joka on tilannut saman topicin.



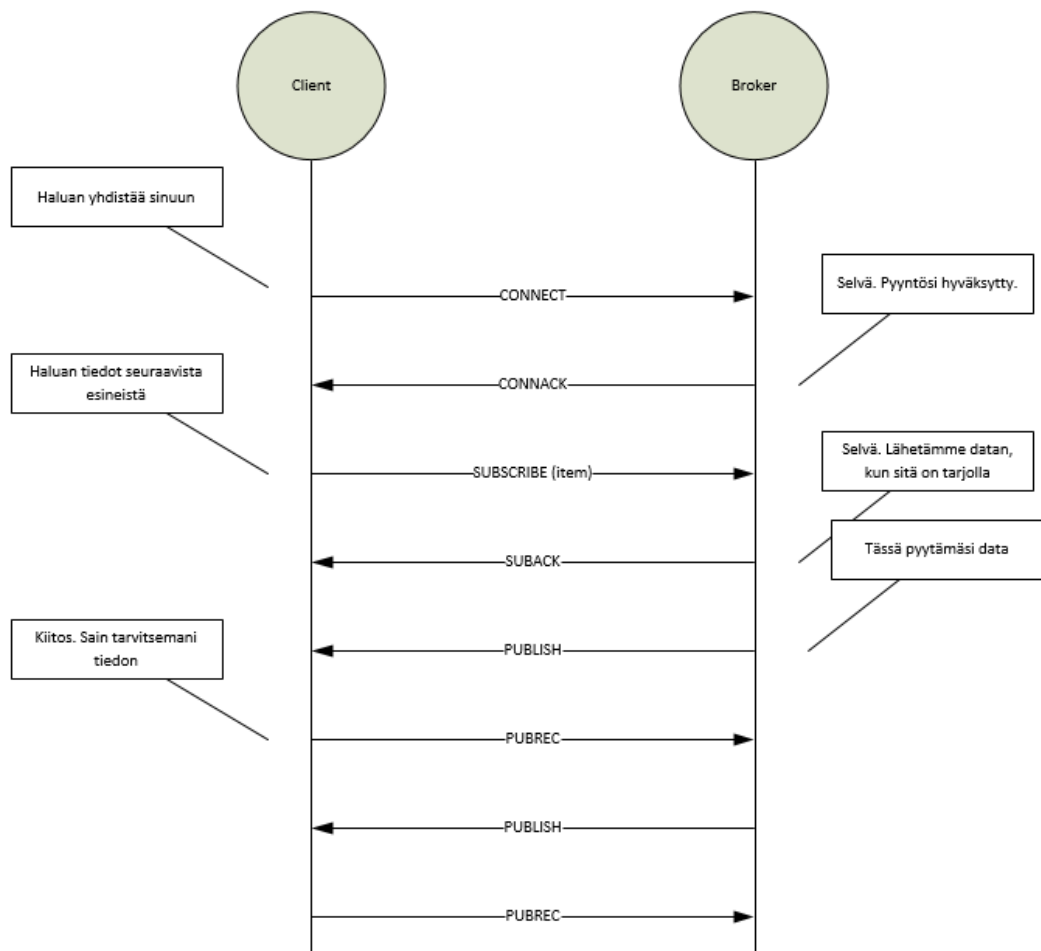
Kuvio 10. Projektin rakenne

3.2.3 Yhteyden muodostus

Asiakas, joka haluaa tiedon haluamastaan esineestä, on yhdistettävä brokeriin, joka omistaa halutut tiedot. Yhdistämisen jälkeen kommunikointi näitten välillä voi alkaa ja haluttu tieto saadaan asiakkaalle aina uuden tiedon saapuessa lähettäjältä. Yhdistämisen eri vaiheet löydetään kuvioista 11. Käydään läpi eri kohtien toiminta.

1. Asiakas haluaa yhdistää brokeriin ja lähettää pyynnön.
2. Broker hyväksyy pyynnön ja lähettää tästä vahvistuksen asiakkaalle
3. Asiakas tilaa Brokerilta halutut tiedot lähettämällä Subscribe- viestin Brokerille
4. Broker lähettää paluuviestinä hyväksynnän, jolla lupaa lähettää tiedon, kun se on saatavilla.
5. Kun Broker saa viimeisimmän tiedon, se lähettää (Publish) tiedot sitä pyytäneille.

6. Asiakas ilmoittaa saaneensa tiedot.
7. Tämä jatkuu, kunnes asiakas ei enää halua vastaanottaa tietoja.



Kuvio 11. Yhteyden muodostus MQTT-protokollan mukaan

3.2.4 MQTT ja tietoturva

Internetiin yhdistettyjen laitteiden määrä on kasvanut nopeasti, kun useampiin laitteisiin tulee mahdollisuus yhdistää ne verkkoon. Statista.comin mukaan verkkoon yhdistettyjä laitteita oli 15,41 biljoonaa vuonna 2015, kun taas jo vuonna 2017 laitteita on 20,35 biljoonaa. Vuonna 2025 yhdistettyjä laitteita arvioidaan olevan jopa 75,44 biljoonaa. (Statista 2017.)

Laitteiden räjähdysmäinen kasvu antaa myös mahdollisuudet uusille tietoturva-uhkille. Usein uudet laitteet tai ohjelmat, jotka käyttävät MQTT-protokollaa ei ole suojattu oikein, tai loppukäyttäjä ei ole osannut/pystynyt tekemään tarvittavia salasana-muutoksia hyökkäyksien estämiseksi. Tästä hyvä esimerkki on laitteet, joissa oli alkuperäinen käyttäjätunnus ja salasana. Ne oli mahdollista kaapata ja käytettyä osana botnet-verkkoa. Tietoturvamielessä pystytään kuitenkin suojaamaan oman verkon laitteet suhteellisen helposti. Suojauksia voi tehdä aina dataa lähettävään laitteeseen, lähetettävään dataan ja viestin vastaanottavaan brokeriin asti. Käydään tässä kappaleessa läpi muutaman hyvän tavan suojata tieto. (Woolf, N 2016.)

3.2.5 Käyttäjän ID

Jokaisella laitteella on oma ID eli tunnistenumero. Kun käyttäjä yhdistää aiheeseen, sen tunnistenumero tulee näkyviin topickiin. Tunnuksella voidaan seurata käyttäjän toimintaa. Riippuen asetuksista voidaan broker asettaa hyväksymään vain tietyn alueen laitteita. Tällä saadaan hiukan tietoturvaa, mutta mikäli hyökkääjä tietää sallitun alueen, pääsee yhdistämään laitteelle tästä huolimatta.

3.2.6 Käyttäjätunnus ja salasana

Käytettävä broker voidaan asettaa kysymään laitteelta käyttäjätunnusta ja salasanaa. Mikäli nämä tiedot ovat oikein, pääsee käyttäjä yhdistämään, muuten yhteys katkaistaan. Vaikka käyttäjätiedot voidaan salata brokerissa, ei tunnusten lähetystä tietoverkossa salata automaattisesti millään salauksella. Turvallisuuden takaamiseksi kannattaa suojata myös liikenne, jossa kirjautumistiedot kulkevat.

3.2.7 X509-sertifikaatti

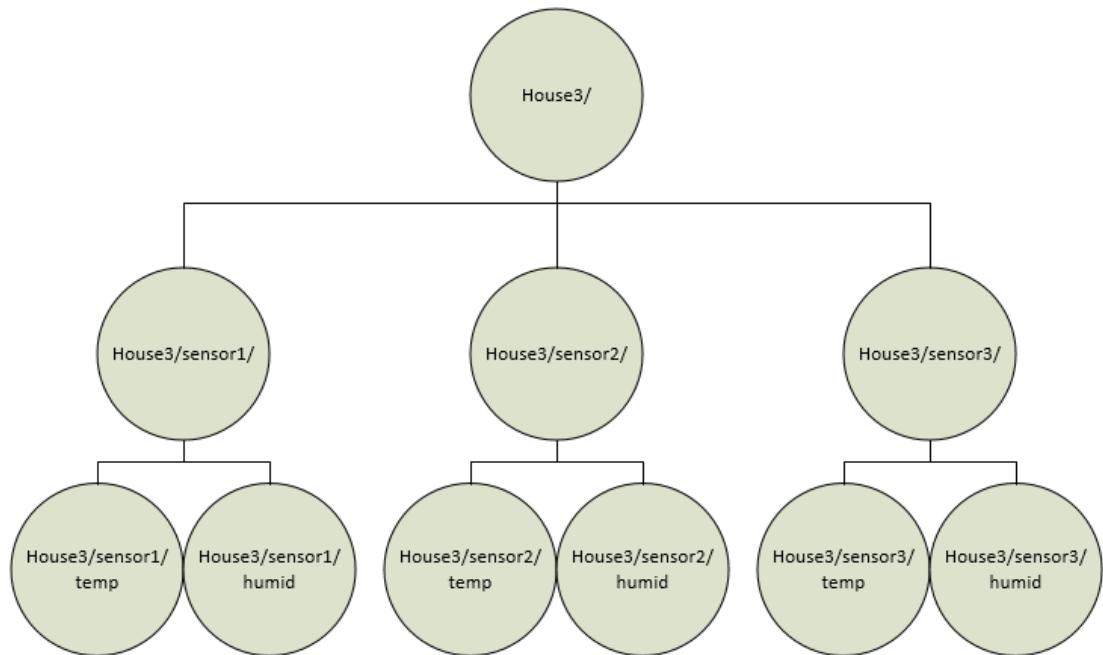
Kaikista paras tietoturva saadaan käyttämällä X509-sertifikaattia. Sertifikaatti täytyy kuitenkin laittaa erikseen jokaiselle laitteelle, mikä on hidasta, joten tätä suositellaan vain tilanteisiin, joissa laitteita on vähän tai asennus voidaan suorittaa vain laitteelle, jotka tarvitsevat erityistä suojausta. (Housley 1999.)

3.2.8 Sisällön salaus

Lähetettävien tietojen kryptaus tapahtuu lähetettävien ohjelmien välillä. Tällöin datan siirto ei tapahdu vain käyttäjän ja brokerin välissä, vaan kestää koko matkan lähettäjältä ohjelmalle. Tähän ei tarvita erillistä brokerin tiedostojen muuttamista, joten sisällön salaus sopii hyvin tilanteisiin, joissa brokerille ei ole yhteyttä. (MQTT Security Fundamentals: MQTT Payload Encryption 2017.)

3.3 Topic ja sen rakenne työssä

MQTT:ssä broker jakaa viestit niihin ohjattuihin topickeihin. Topicit voivat sijaita missä vain, kunhan palvelu on yhteydessä brokeriin. Viestin voi myös ohjata kaikkiin päätopicin alla oleviin topickeihin. Työssä käytettiin Hive-MQ- nimistä brokeria, johon yhdistettiin sensorit ja thingworxin. Työssä käytettiin puumaista rakennetta, jolloin ylimpänä on päätopic, joka kattaa koko rakennuksen. Tämän jälkeen jaetaan topicit sensorikohtaisesti. Tällä tarkoitetaan siis yhtä Arduinoa, jossa kyseinen sensori on kiinni. Sensori pystyy havaitsemaan niin kosteutta kuin lämpötilaakin. Data havainnollistetaan kuviossa 12.



Kuvio 12. Opinnäytetyön kanavien topologia

3.4 MQTT ja QoS

QoS eli Quality of Service määrittää, kuinka todennäköisesti kyseinen viesti tulee perille. Tämä sopimus on lähettäjän ja vastaanottajan välinen ja MQTT:ssä käytetään kolmea eri tasoa, jotka ovat 0,1 ja 2. Toimintaeroina eri QoS-versioilla on varmistus viestin saapumiesta. (MSV J 2016.)

3.4.1 QoS 0

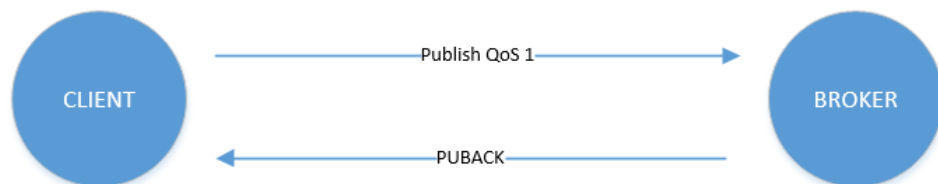
QoS 0 on alin lähetystaso. Jos lähetettävä tieto ei ole kovin tärkeää, voi tätä tasoa käyttää. Projektissa käytetään vain QoS 0, sillä lämpötilojen perillepääsy ei ole täysin välttämätöntä. Jopa 20min viive lämpötilan vastaanotossa ei tuota isoja ongelmia. QoS 0 tarkoittaa, että viestin lähettäjä lähettää viestin vain kerran ja Broker ottaa viestin mahdollisesti vastaan. Mikäli virheitä ilmenee, eikä viesti mene perille, ei tästä tule ilmoitusta kummallekkaan osapuolelle, eikä viestiä lähetetä enää uudelleen. Kuviossa 13 on QoS 0:n toiminta.



Kuvio 13. QoS 0:n toiminta

3.4.2 QoS 1

QoS 1 on astetta turvallisempi tapa lähettää tietoa eteenpäin. Lähettäjän lähetettyä viestin se muistaa lähettämänsä viestin, kunnes saa vastaanottajalta paluuviestin. Vastausviestin jälkeen lähetetty viesti poistetaan lähettävästä päästä, sillä se on vastaanotettu onnistuneesti. Mikäli vastaanottaja ei lähetä vastausta, lähettää lähettäjä viestin uudelleen vastaanottajalle, kunnes se vastaanotetaan perille onnistuneesti. Ero QoS 0:an verrattuna on kuitenkin viestien kaksinkertainen lähetetty määrä, mutta viestien varmempi vastaanotto. Viesti voi myös tulla moneen kertaan, mikäli vastaanottaja on liian kiireinen vastaamaan. Kuviossa 14 on esitelty QoS 1:n toiminta



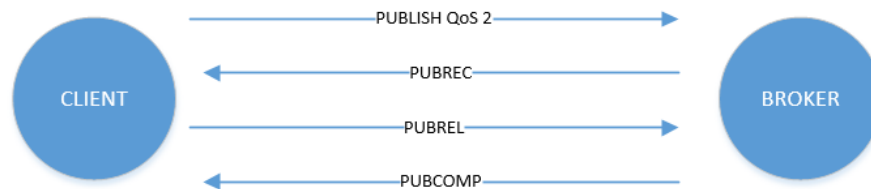
Kuvio 14. QoS 1:n toiminta

3.4.3 QoS 2

QoS 2 on kaikista varmin lähetystapa. Tässä tapauksessa viesti pääsee varmasti perille ja viesti saapuu vain kerran, eikä vahingossa kahdesti. QoS 2 toimii melkein samoin, kuin QoS 1, mutta on varmistettu viestin saapumiseksi vain kerran. Viestin saatuaan vastaanottaja tallentaa saadun viestin samanlaisten saapuvien viestien varalle ja lähettää ilmoitusviestin saapumisesta lähettäjälle. Saatuaan varmistuksen lähettäjä

poistaa viestin ja vastaa siihen PUBREL- viestillä, jolla se viestittää saaneensa vahvistuksen eikä enää lähetä viestejä. Lopuksi molemmat osapuolet tietävät, että viesti on mennyt perille ja kaikki välivaiheiden tallennetut tiedot on poistettu. Kuviossa 15 on esitetty QoS 2 toiminta.

QoS 2:en tarvitaan ylimääräisiä resursseja joka kuormittaa molempia osapuolia. Viestejä lähetetään yhteensä neljä kertaa enemmän, kuin QoS 0, mutta se on varmin tapa saada viesti perille vastaanottajalle. (MQTT Essentials Part 6: Quality of Service 0, 1 & 2 2017.)



Kuvio 15. QoS 2 toiminta

4 Elisa IoT

4.1 Yleistä

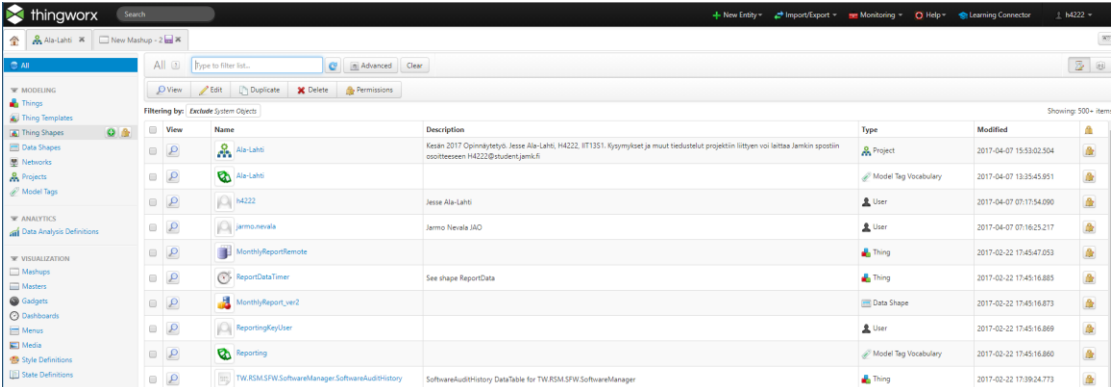
Elisa IoT- palvelu käyttää pohjanaan Thingworx Composer 7- nimistä IoT- alustaa, jonka on kehittänyt PTC. Palveluun pystyy lähettämään dataa ohjelmista tai fyysisistä sensoreista. Tätä dataa voidaan analysoida ja käsitellä palvelussa. Dataa voidaan myös esittää graafisessa muodossa Mashup-visualisoinnin avulla. Dataan voidaan myös vaikuttaa ja palvelussa voidaan myös lähettää komentoja takaisin ohjelmaan tai sensorille. Tällöin saadaan luotua interaktiivinen hallintapaneeli, jossa tarkkailija voi myös toimia. (Thingworx Delivers Industrial Innovation 2017.)

4.2 ThingWorx- käyttöliittymä

Thingworxin käyttäminen on nopeaa ja helppoa. Kotisivulta löytyy kaikki tarpeellinen ja pystyttiin tekemään montaa asiaa samanaikaisesti, sillä avattu kohde jää auki ohjelman välilehteen. Välilehtiä voi selata ja muokata samanaikaisesti. Tämä helpottaa paljon, jos kaksi tiedostoa ovat yhteydessä toisiinsa tai haluttiin kopioida tekstikohtia tiedostosta toiseen.

Kuviossa 16 kotisivun vasemmassa reunassa löydät kaikki materiaalin luomiseen tarkoitetut välineet. Näin voidaan luoda tarvittu tiedosto suoraan tästä valikosta. Esineet on sijoitettu ryhmittäin aina sen perusteella, mitä halutaan tehdä. Esimerkiksi datan visualisointiin tarkoitettujen työkalujen kohdassa ja tiedostojen tallentamiseen tarkoitettujen tietojen kohdassa.

Näytön keskellä voidaan selata palvelussa olevia tiedostoja. Tiedostoja voidaan rajata tai etsiä hakusanoilla. Esimerkiksi kaikki esineet (things), voidaan hakea kirjoittamalla hakusanaksi thing.



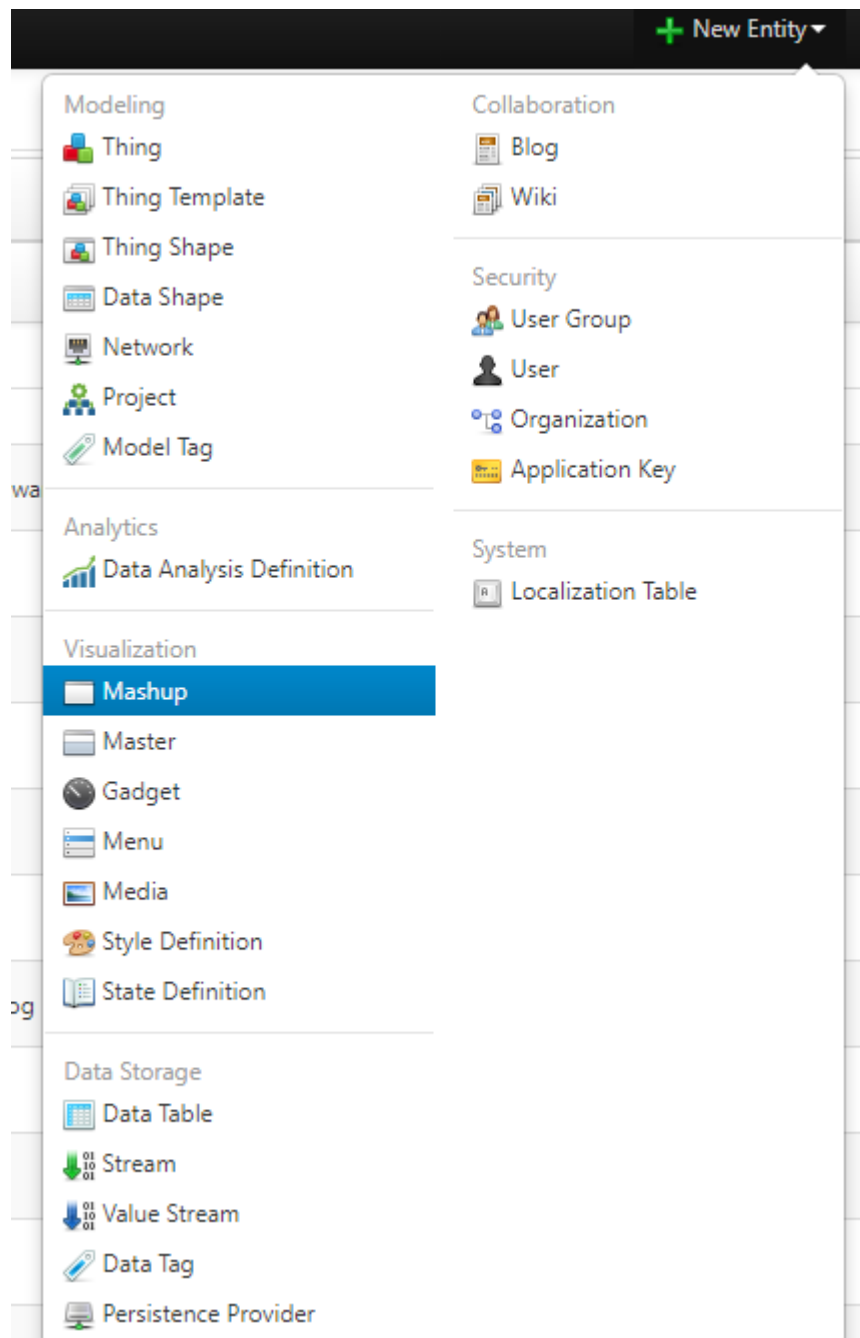
View	Name	Description	Type	Modified
	Ala-Lahri	Kesän 2017 Opinnäytetyö, Jesse Ala-Lahri, H4222, IT1351. Kysymykset ja muut tiedustelut projektiin liittyen voi laittaa Jamin spottiin osoitteeseen H4222@student.jamk.fi	Project	2017-04-07 15:53:02.504
	Ala-Lahri		Model Tag Vocabulary	2017-04-07 13:35:45.951
	H4222	Jesse Ala-Lahri	User	2017-04-07 01:17:54.090
	Jarmo.Nevela	Jarmo Nevela JAO	User	2017-04-07 01:16:25.217
	MonthlyReportRemote		Thing	2017-02-22 17:45:47.059
	ReportDataTimer	See shape ReportData	Thing	2017-02-22 17:45:16.885
	MonthlyReport_ver2		Data Shape	2017-02-22 17:45:16.873
	ReportingKeyUser		User	2017-02-22 17:45:16.869
	Reporting		Model Tag Vocabulary	2017-02-22 17:45:16.860
	TW.RDM.SP.W.SoftwareManager.SoftwareAuditHistory	SoftwareAuditHistory DataTable for TW.RDM.SP.W.SoftwareManager	Thing	2017-02-22 17:39:24.773

Kuvio 16. Thingworx-sovelluksen aloitusruutu

4.3 Materiaalin luomisen vaihtoehdot

Erilaisia vaihtoehtoja luomiseen on paljon. Näillä vaihtoehdoilla pärjätään koko projektin läpi eikä isoimmisakaan projekteissakaan tieto häviä dokumentoinnin ja organisaatiokaavion muodossa. Tässä tutkimuksessa käydään läpi melkein kaikki Thingworxin tarjoamat vaihtoehdot. Vaikka kaikkia ei tässä projektissa käytetäkään, voi

niistä olla hyötyä projektin edetessä opinnäytetyön jälkeen. Kuviossa 17 näet kaikki materiaaliveitokset.



Kuvio 17. Mahdollisuudet projektin osien luontiin

4.3.1 Thing

Thing tarkoittaa esinettä virtuaalisessa muodossa. Esineellä voi olla erilaisia arvoja ja se voi kuulua johonkin esineiden ryhmään. Esineen halutut ominaisuudet riippuvat

siitä, minkä Thing Templaten esineelle määritellään. Esine voi olla esimerkiksi polkupyörä ja sen arvoja voivat olla esimerkiksi nopeus ja sijainti. Esineen arvot liikkuvat aina esineen mukana ja sen arvot voidaan tallentaa erilliseen tiedostoon, jolloin esineen arvojen historiaa voidaan analysoida tai ilmaista graafisesti. Kuviossa 18 nähdään vaadittavat asetukset esineen luonnille.

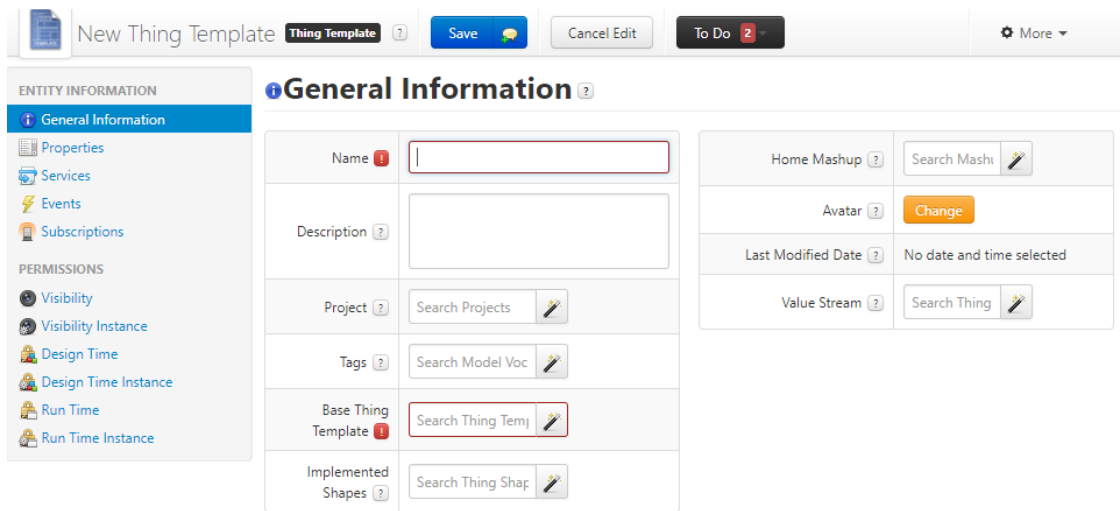
The screenshot shows the 'New Thing' configuration interface. At the top, there is a header with a gear icon, the text 'New Thing Thing', and buttons for 'Save', 'Cancel Edit', and 'To Do'. Below the header is a sidebar with 'ENTITY INFORMATION' and 'PERMISSIONS' sections. The main area is titled 'General Information' and contains several fields and controls:

Name	<input type="text"/>	Active	<input checked="" type="checkbox"/>
Description	<input type="text"/>	Home Mashup	Search Mashups <input type="text"/>
Project	Search Projects <input type="text"/>	Avatar	Change
Tags	Search Model Voc <input type="text"/>	Published	<input type="checkbox"/>
Thing Template	Search Thing Temj <input type="text"/>	Identifier	<input type="text"/> Browse...
Implemented Shapes		Last Modified Date	No date and time selected
		Value Stream	Search Thing <input type="text"/>

Kuvio 18. Esineen luonnin asetukset

4.3.2 Thing Template

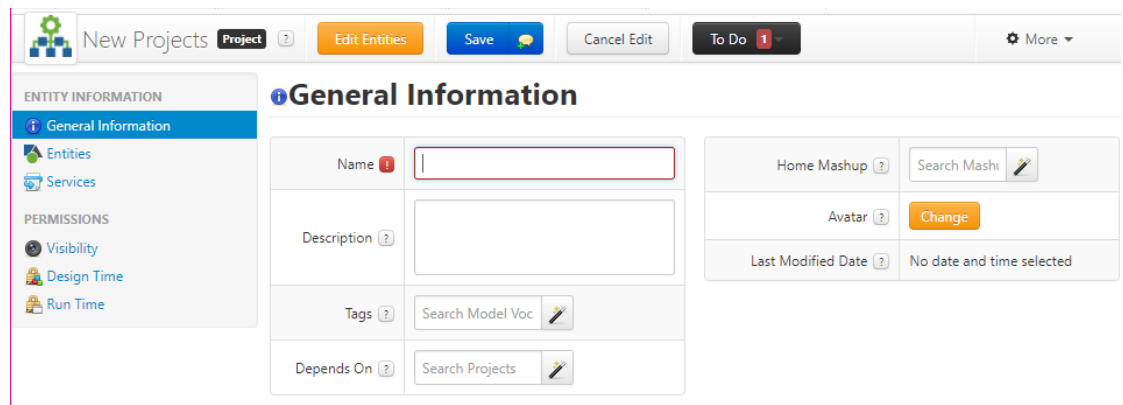
Thing Template on valmis pohja, jota voidaan käyttää esineiden luontiin. Tämä nopeuttaa esineiden luontia, mikäli samanlaisia esineitä tehdään paljon. Erilaisia valmiita pohjia voi tehdä ja ladata netistä. Tässä työssä ladataan MQTT-templatien, joka sisältää pohjan MQTT-tä hyödyntävien esineiden luontiin. Kuviossa 19 templatien asetukset.



Kuvio 19. Templaten luonnin asetukset

4.3.3 Project

Kun halutaan tehdä yksi kokonaisuus, joka sisältää niin visualisointia, esineitä ja mediaa, luodaan projekti. Kun projektille on annettu tarvittavat tiedot, voi projektiin lisätä esineitä. Näin saman projektin esineitä on helppo hallita, eivätkä tiedostot katoa muitten tiedostojen joukkoon. Kuviossa 20 projektin tarvitsemat tiedot. Esineellä voi olla vain yksi projekti, johon se kuuluu, kun taas Tägeja voi olla monta.



Kuvio 20. Asetukset projektin luonnissa

4.3.4 Model Tag

Model Tägeilla voit antaa esineelle erilaisia lyhyitä kuvauksia. Nämä helpottavat kategorioimaan esineitä ja etsimään juuri sen halutun. Toisin kuin projekteja, tageja voi

olla esineellä monta. Tageja voidaan käyttää esimerkiksi esineiden lajitteluun. Tagin luontiin tarvitaan kuvion 21 mukaiset tiedot.

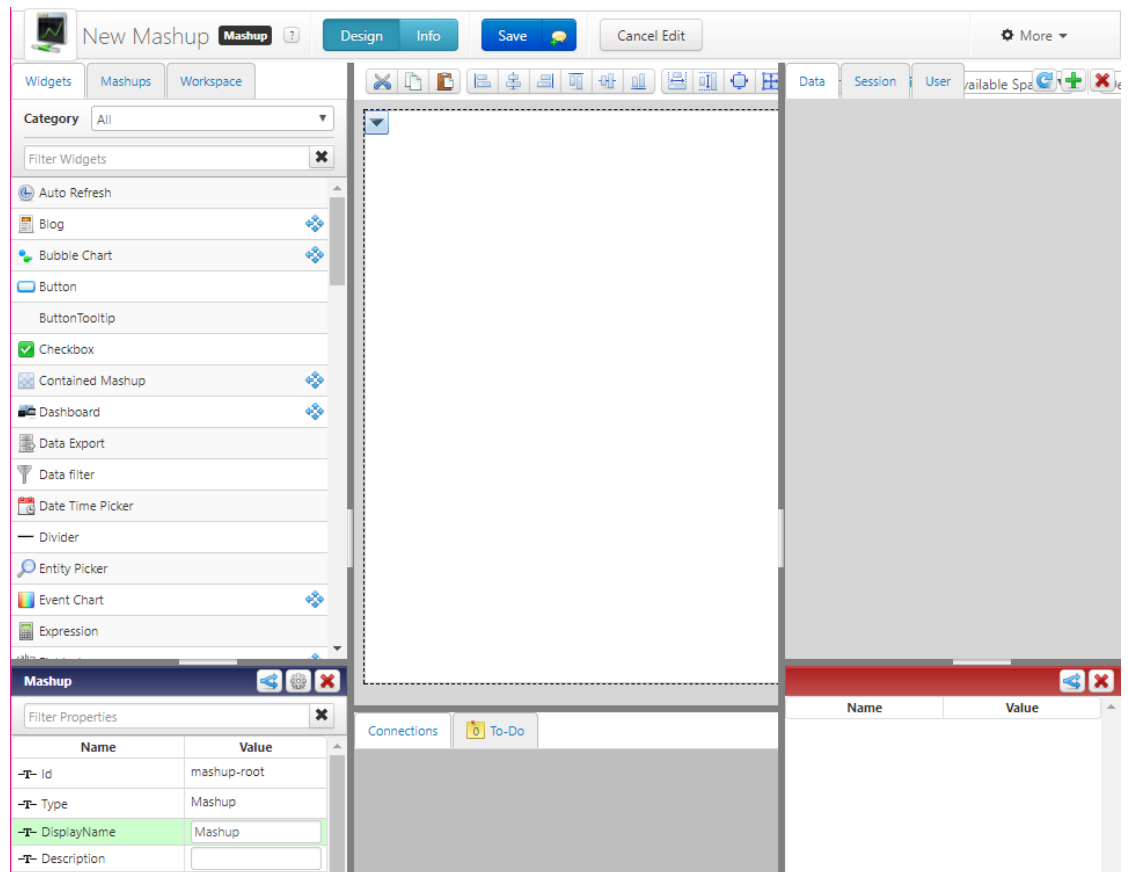
The screenshot shows a web interface for creating a new vocabulary tag. The main title is 'New Vocabulary' with a sub-tab 'Model Tag Vocabulary'. The interface includes a top navigation bar with buttons for 'Manage Terms', 'Save', 'Cancel Edit', and 'To Do'. A left sidebar contains 'ENTITY INFORMATION' with 'General Information' selected, and 'PERMISSIONS' with options for 'Visibility', 'Design Time', and 'Run Time'. The main content area is titled 'General Information' and contains a form with the following fields:

Name	<input type="text"/>	Home Mashup	<input type="text" value="Search Mashup"/>
Description	<input type="text"/>	Avatar	<input type="button" value="Change"/>
Project	<input type="text" value="Search Projects"/>	Last Modified Date	No date and time selected
Tags	<input type="text" value="Search Model Voc"/>		
Dynamic	<input type="checkbox"/>		

Kuvio 21. Mahdollisuudet tagien luontiin

4.3.5 Mashup

Mashup on näkymä, jossa voidaan visualisoida luotuja esineitä ja niiden arvoja. Tämä helpottaa arvojen lukua. Mikäli esineiden data säilötään, voidaan luoda erilaisia graafeja, miten eri esineiden arvot ovat muuttuneet. Tätä käytettiin työssä visualisoimaan huoneiston lämpövaihtelua. Kuviossa 22 tyhjä mashup. Erilaisia widgettejä, joita sijoitetaan mashupiin, voidaan joko tehdä itse tai ladata.



Kuvio 22. Tyhjä mashup-alusta

4.3.6 Gadget

Gadgetit ovat pieniä ohjelmia, joilla voidaan graafisesti esittää haluttua dataa. Näitä voi olla eri kokoisia ja näköisiä sekä eri käyttötarkoitukseen. Näitä voidaan luoda itse mashupissa. Gadget voi olla joko nappi, kalenteri, radio tai mikä vain.

4.3.7 Media

Mikäli halutaan liittää mashuppiin kuva, täytyy siitä tehdä erillinen mediatiedosto thingworxiin. Pelkkä kuvan vetäminen työpöydältä palveluun ei siis onnistu. Kuvaa voi tämän jälkeen käyttää moneen kertaan ja sitä voidaan jälkepäin muuttaa. Kuviossa 23 median lataamiseen tarvittavat tiedot.

The screenshot shows a web-based form for creating a new media entity. The form is titled "General Information" and is part of a "New Media Entity" page. The interface includes a top navigation bar with buttons for "Save", "Cancel Edit", and "To Do". A left sidebar contains "ENTITY INFORMATION" and "PERMISSIONS" sections. The main form area has the following fields and controls:

- Name:** A text input field with a red error indicator.
- Description:** A large text area.
- Project:** A dropdown menu with a search box labeled "Search Projects" and a pencil icon for editing.
- Tags:** A dropdown menu with a search box labeled "Search Model Voc" and a pencil icon for editing.
- Last Modified Date:** A field showing "No date and time selected".
- Image:** A field with a "Change" button.

Kuvio 23. Esineen luonti mashupia varten

4.3.8 Value Stream

Streameja on RemoteValueStream ja ValueStream. Kuviossa 24 voidaan valita näitten välillä samalla value streamin luomista. Näitten molempien tehtävä on tallentaa saatu data palvelimeen datan myöhempää käyttöä varten. Näiden ero kuitenkin on se, että RemoteValueStream tallentaa tiedot ulkoiselle palvelimelle, eikä luotuun tiedostoon thingworxin sisälle. Tämä on hyvä vaihtoehto, mikäli dataa tulee paljon ja/tai palvelimen tiedostot peilataan muille palvelimille. ValueStream tallentaa tiedostot suoraan samalle palvelimelle, missä palvelu pyörii.

The screenshot shows the 'New Thing' configuration interface. At the top, there is a navigation bar with a gear icon, the text 'New Thing', a 'Thing' label, a 'Save' button, a 'Cancel Edit' button, and a 'To Do' button with a red notification icon. Below this is a sidebar with 'ENTITY INFORMATION' and 'PERMISSIONS' sections. The 'General Information' tab is selected. The main content area contains a form with the following fields:

- Name:** A text input field.
- Description:** A large text area.
- Project:** A search field with a dropdown arrow.
- Tags:** A search field with a dropdown arrow.
- Thing Template:** A dropdown menu showing 'ValueStream'.
- Implemented Shapes:** A search field with a dropdown arrow.
- Active:** A checkbox that is checked.
- Home Mashup:** A search field with a dropdown arrow.
- Avatar:** A button labeled 'Change'.
- Published:** A checkbox that is unchecked.
- Identifier:** A search field with a 'Browse...' button.
- Last Modified Date:** A field with the text 'No date and time selected'.
- Value Stream:** A search field with a dropdown arrow.

At the bottom, there is a 'Persistence Provider' field with a dropdown menu showing 'ThingworxPersistenceProvider' and a close button.

Kuvio 24. Tietojen tallennuspaikan asetusten täyttäminen

4.3.9 Application key

Ulkopuolinen ohjelma, joka haluaa toimia thingworxin kanssa, tarvitsee application keyn. Tämä on avain, jota käyttämällä ohjelma saa oikeudet olla yhteydessä palvelimeen, eikä sitä tällöin estetä. Kaikki ohjelmistorajapinnat tarvitsevat toimiakseen tämän avaimen. Avaimia pystyy luomaan ja luovuttamaan halutulle ohjelmalle. Niin kuin kuviossa 25 on nähtävissä, voidaan asettaa ohjelmalle myös ip luomaan lisää tietoturva.

The screenshot shows the 'New Application Key' form. The top navigation bar includes 'New Application Key', 'Application Key', 'Save', 'Cancel Edit', 'To Do 2', and 'More'. The left sidebar shows 'ENTITY INFORMATION' with 'General Information' selected, and 'PERMISSIONS' with 'Visibility', 'Design Time', and 'Run Time'. The main form area is titled 'General Information' and contains the following fields:

Name	<input type="text"/>
Description	<input type="text"/>
Project Name	<input type="text"/> Search Projects <input type="button" value="🔍"/>
Tags	<input type="text"/> Search Model Voc <input type="button" value="🔍"/>
IP Whitelist	<input type="text"/>
Client Name	<input type="text"/>
User Name Reference	<input type="text"/> Search Users <input type="button" value="🔍"/>
keyId	<input type="text"/>

On the right side, there is a summary table:

Home Mashup	<input type="text"/> Search Mashups <input type="button" value="🔍"/>
Avatar	<input type="button" value="Change"/>
Expiration Date	No date and time selected <input type="button" value="✕"/>
Last Modified Date	No date and time selected

Kuvio 25. Asetukset avaimen luontiin

4.3.10 New User

Mikäli käyttäjällä on tarpeeksi käyttöoikeuksia palvelussa, voi hän luoda palveluun useita käyttäjiä. Käyttäjät voivat kirjautua omilla käyttäjätunnuksillaan ja salasanoillaan palveluun. Jokaiselle käyttäjälle tulisi luoda oma tunnus, jolloin sisällöntuottoa pystytään seuraamaan paremmin. Kuviossa 26 käyttäjät pystytään myös liittämään suoraan projektiin, jolloin koko projektitiimi toimii alusta asti projektin sisällä.

The screenshot shows the 'New User' form. The top navigation bar includes 'New User', 'User', 'Save', 'Cancel Edit', 'To Do 1', and 'More'. The left sidebar shows 'ENTITY INFORMATION' with 'General Information' selected, and 'PERMISSIONS' with 'User Extensions', 'User Profile Configuration', 'Visibility', 'Design Time', and 'Run Time'. The main form area is titled 'General Information' and contains the following fields:

Name	<input type="text"/>
Description	<input type="text"/>
Project Name	<input type="text"/> Search Projects <input type="button" value="🔍"/>
Tags	<input type="text"/> Search Model Vocabulary <input type="button" value="🔍"/>
Languages	<input type="text"/> <input type="button" value="Edit"/>
Password	<input type="button" value="Change Password"/>
Enabled	<input checked="" type="checkbox"/>
Locked	<input type="checkbox"/>

On the right side, there is a summary table:

Home Mashup	<input type="text"/> Search Mashups <input type="button" value="🔍"/>
Mobile Mashup	<input type="text"/> Search Mashups <input type="button" value="🔍"/>
Avatar	<input type="button" value="Change"/>
Last Modified Date	No date and time selected

Kuvio 26. Käyttäjän luonti

4.3.11 User Group

Mikäli käyttäjiä on paljon, voidaan käyttäjiä helposti lajitella ryhmiin tekemällä ryhmä. Ryhmän voi liittää myös projektiin ja sille voi antaa tageja. Ryhmän tiedoista pysytään myös seuraamaan ryhmän jäsenten määrää. Tämä ominaisuus on hyvä esimerkiksi kaikkien ryhmän tai esimerkiksi luokan jäsenten luontiin. Kuviossa 27 luodaan ryhmä.

The screenshot shows a web interface for creating a new user group. At the top, there is a navigation bar with a 'New Group' button, a 'User Group' dropdown, and several action buttons: 'Edit Members', 'Save', 'Cancel Edit', and 'To Do' (with a notification icon). Below this is a sidebar menu under 'ENTITY INFORMATION' with options for 'General Information' (selected), 'Members', and 'PERMISSIONS' (Visibility, Design Time, Run Time). The main content area is titled 'General Information' and contains a form with the following fields:

- Name:** A text input field with a red border and a red exclamation mark icon.
- Description:** A larger text area with a question mark icon.
- Project:** A dropdown menu with a search icon and the text 'Search Projects'.
- Tags:** A dropdown menu with a search icon and the text 'Search Model Vocabulary'.

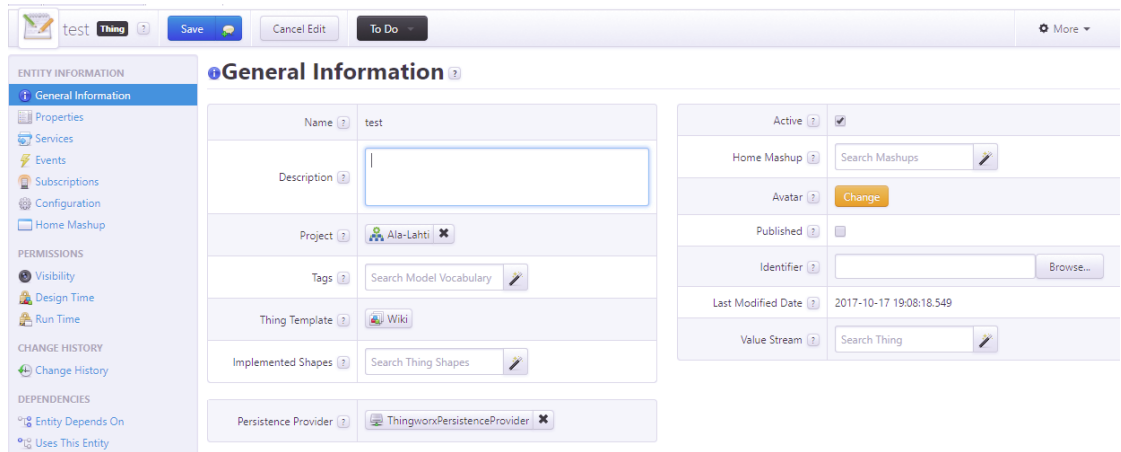
On the right side of the form, there are two additional fields:

- Avatar:** A field with a question mark icon and a 'Change' button.
- Last Modified Date:** A field with a question mark icon and the text 'No date and time selected'.

Kuvio 27. Ryhmän luonti

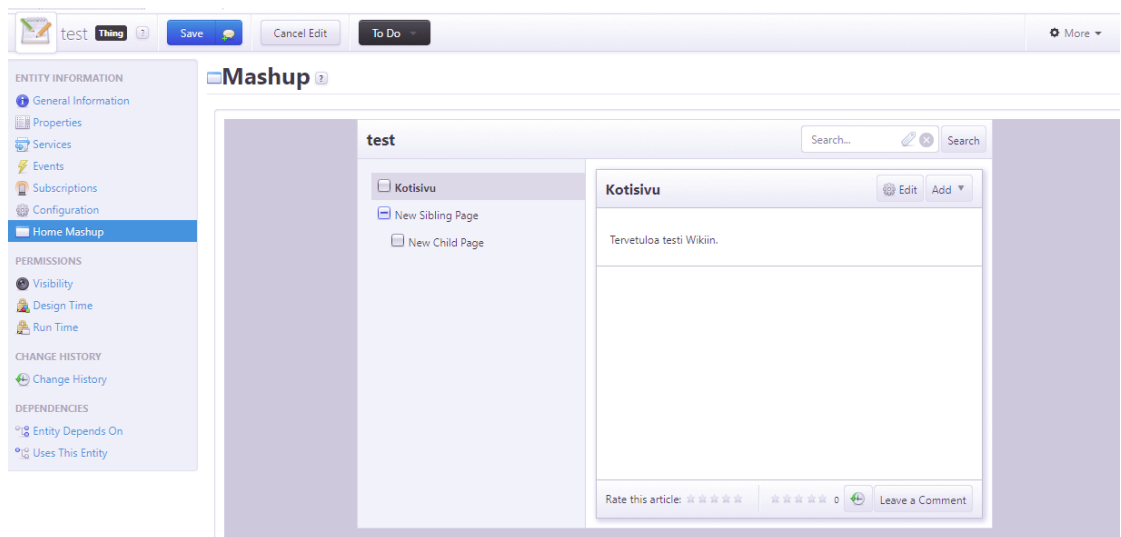
4.3.12 Wiki

Massiivisissa projekteissa voi olla tarvetta myös omalle wikille. Wikiin voi kerätä hyödyllistä tietoa eri projektin osista ja sitä voidaan pitää tietokirjana projektin osien muuttamiselle tai kehittämiseksi. Wikin tiedot voi tallentaa joko ulkoiselle tallennustilalle tai thingworxin tarjoamalle levyille. Nimen ja mahdollisten muitten tietojen antamisen jälkeen voidaan siirtyä Home Mashup -välilehteen, jossa Wikin tekeminen tapahtuu. Kuviossa 28 luodaan testi niminen wiki projektiin Ala-Lahti, joka tallennetaan thingworxin tarjoamaan tiedostopalvelimeen.



Kuvio 28. Wikin asetukset

Wikiin on helppo luoda sivuja ja muokata niitä. Sivujen lisääminen tapahtuu valitsemalla oikealta kohdan Add. Täältä voit lisätä myös alasisun pääsivun alle. Wikin tekstejä voidaan myös kommentoida ja arvioida. Kommenttien perusteella wikin ylläpitäjä pystyy korjaamaan wikiä. Kuviossa 29 wikin visuaalinen ilme.



Kuvio 29. Wikin visuaalinen luonti asetusten asettamisen jälkeen

4.3.13 Style Definition

Mikäli koko projektin käyttöliittymä ja näkyvät kohdat halutaan mukauttaa esimerkiksi yrityksen graafiseen ilmeeseen, voidaan graafinen ilme määrittellä Style Definitionin avulla. Kuviossa 30 voidaan määrittää tekstien fontti, koko ja muu värimaailma.

The screenshot shows the 'New Style Definition' interface. The top bar includes a 'New Style Definition' button, a 'Style Definition' dropdown, a 'Save' button, a 'Cancel Edit' button, and a 'To Do' notification with a red '1'. The left sidebar is titled 'ENTITY INFORMATION' and contains 'General Information', 'Style Information' (selected), and 'PERMISSIONS' (Visibility, Design Time, Run Time). The main panel is titled 'Style Definition' and contains the following settings:

Display String ?	<input type="text"/>
Background Color ?	<input type="color"/>
Secondary Background Color ?	<input type="color"/>
Foreground Color ?	<input type="color"/>
Font Bold ?	<input type="checkbox"/>
Font Italic ?	<input type="checkbox"/>
Font Underline ?	<input type="checkbox"/>
Image ?	<input type="text" value="Search Media"/>
Line Color ?	<input type="color"/>
Line Thickness ?	<input type="text" value="1"/>
Line Style ?	<input type="text" value="Solid"/>
Text Size ?	<input type="text" value="11px (M)"/>

Kuvio 30. Graafisen ilmeen suunnittelu

5 Projektin toteutus

5.1.1 MQTT lisäosan lataaminen

Projekti aloitettiin lataamalla kaikki tarvittavat osat palveluun. Datat siirrossa käytetään MQTT-protokollaa. Valitettavasti Thingworx ei kuitenkaan tätä suoraan tarjoa,

vaan palveluun täytyy ladata erillinen MQTT- lisäosa. Lataaminen onnistuu kauppapaikasta oheisen linkin kautta: <https://marketplace.thingworx.com/tools/mqtt>. Kauppapaikan käyttö vaatii tunnukset, jotka jokainen voi kuitenkin luoda ilmaiseksi. Tässä tutkimuksessa käytettiin tunnuksien luomiseen tutkijan opiskelijasähköpostia. Tunnusten tekoon voi kuitenkin käyttää mitä tahansa sähköpostia, mutta tällä haluttiin poistaa mahdollisten käyttäjäongelmien muodostumista, jos kyseiseltä sähköpostiosoitteelta ei löydy myös tunnuksia itse palveluun. Pakatun tiedoston lataamisen jälkeen se siirretään Thingworksiin painamalla ruudun yläpalkissa olevaa import-painiketta. Tämän avulla paketin osat saadaan palveluun käytettäväksi. Kuviossa 31 paketti on purettu palveluun.

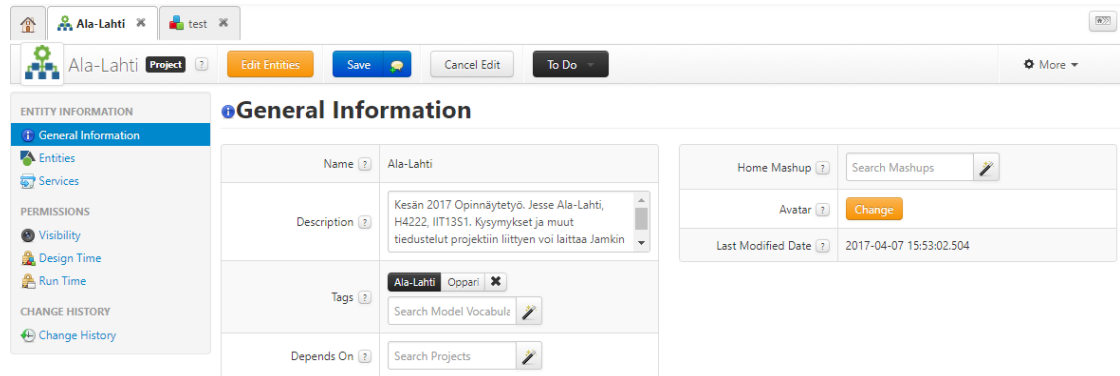
Filtering by: *Exclude System Objects* Showing: 500+ items

<input type="checkbox"/> View	Name	Description	Type	Modified	
<input type="checkbox"/>	MQTTConnection	MQTT connection interface	Thing Template	2017-08-06 17:49:30.705	
<input type="checkbox"/>	MQTTSubscriber	MQTT subscriber interface	Thing Template	2017-08-06 17:49:30.686	
<input type="checkbox"/>	MQTT	MQTT interface	Thing Template	2017-08-06 17:49:30.685	

Kuvio 31. MQTT-lisäosan tiedostojen tuonti

5.1.2 Projektin osien luonti

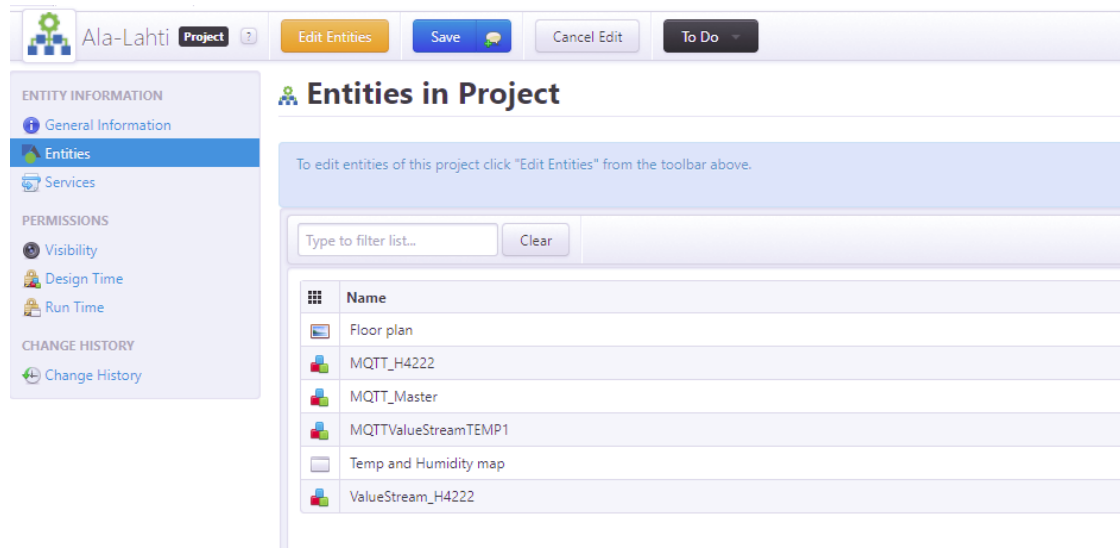
Kun työn eri osia alettiin luomaan, tehtiin ensimmäiseksi projektitiedosto kuviossa 32. Projektille annettiin nimeksi Ala-Lahti. Tätä projektia käytettiin kaikkeen tähän opinnäytetyöhön liittyvään, jolloin kaikki opinnäytetyön eri osat löytyvät helposti Ala-Lahti- projektin alta. Koska palvelussa on myös muita käyttäjiä, projektin dokumentaatioon lisätään projektin luojan yhteystiedot ja miksi projekti on tehty.



Kuvio 32. Projektin luonti ja tietojen asetus

Kuviossa 33 nähdään kaikki projektiin tehdyt tiedostot, jotka ovat seuraavat:

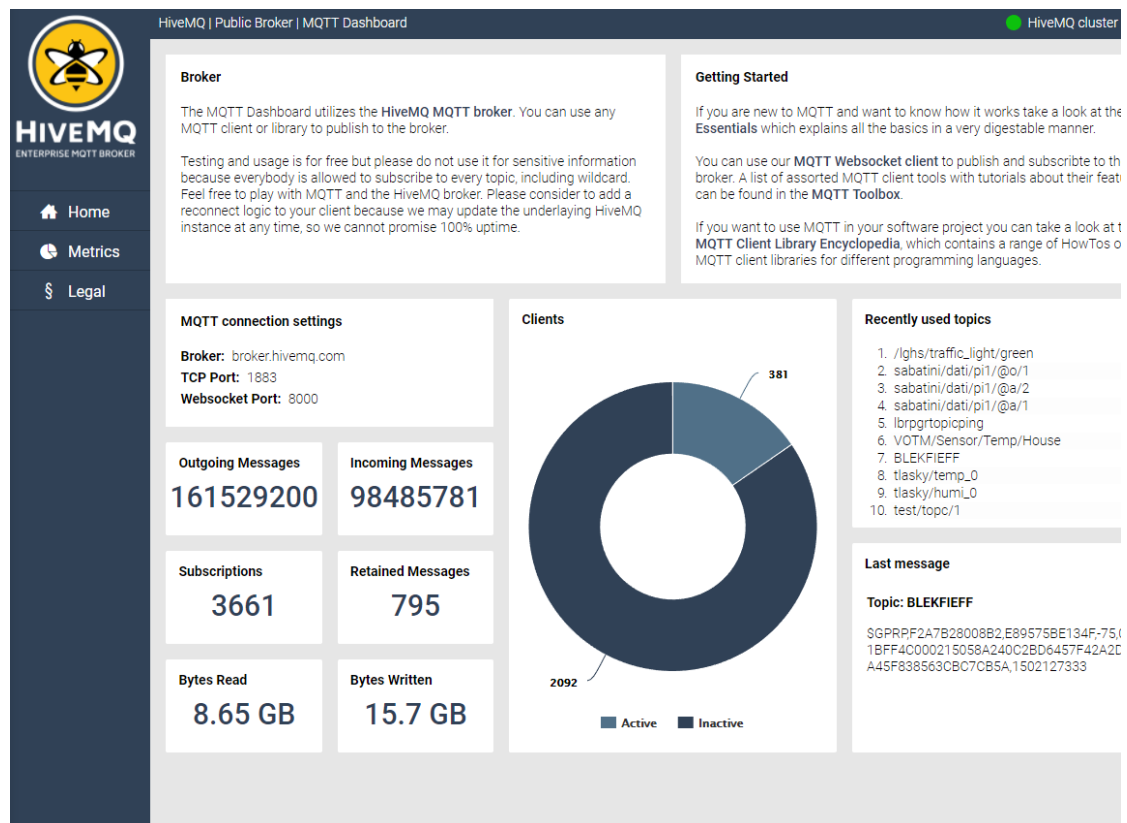
1. Talon pohjapiirrustuksen
2. MQTT- ominaisuuksia sisältävä esine, joka vastaanottaa sisääntulevan datan
3. Tietokantatiedoston, joka tallentaa sisääntulleiden tietojen arvot ja mashup
4. ja mashupin, jossa kaikki tiedot esitetään visuaalisesti.



Kuvio 33. Projektiin liitetyt tiedostot

5.1.3 MQTT- Broker

Seuraavaksi toteutettiin MQTT- Brokeri. Brokerin voi tehdä joko itse käyttämällä esimerkiksi Raspberry Pi- tietokonetta ja asentamalla tälle Mosquitto- nimisen ohjelman. Verkosta löytyy kuitenkin monta vapaasti käytettävää brokeria, jotka toimivat yhtä hyvin. Valmiina olevissa brokereissa tietosuoja on kuitenkin heikkoa eli enemmän tietoturva vaativissa projekteissa tulee käyttää salattua yhteyttä ja mahdollisesti myös omaa suojattua brokeria. Työssä käytetään HiveMQ:n tarjoamaa palvelua, joka on ilmainen ja löytyy osoitteesta broker.hivemq.com. Kuviossa 34 näehdään palvelun. Brokerin tilannetta pystytään seuraamaan HiveMQ:n verkkosivuilta, jossa näkyy myös 10 viimeisintä topickia. Tämänkin takia tärkeitä viestejä ei kannata lähettää yleisten, ei suojattujen ilmaisten palveluiden kautta.

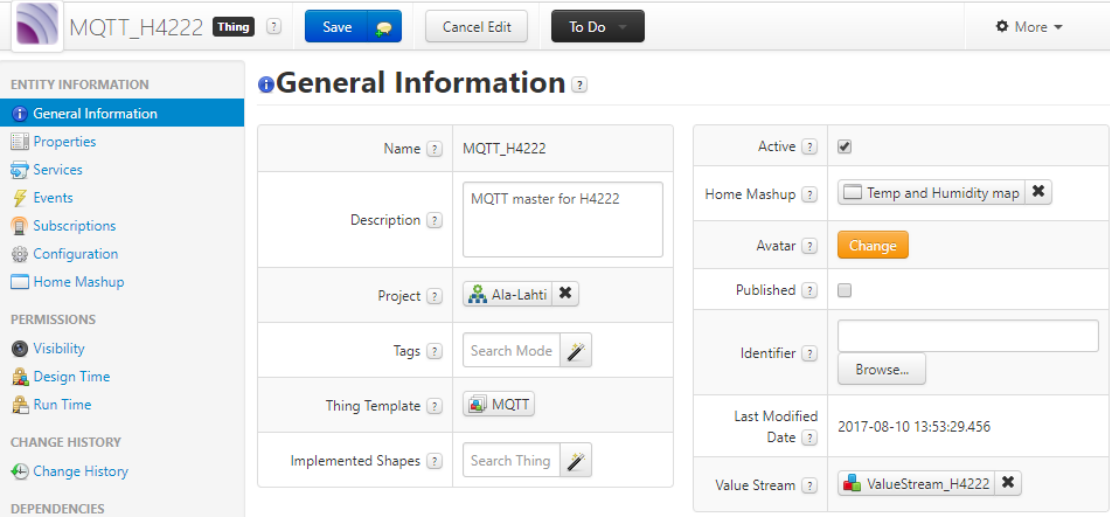


Kuvio 34. Brokerin yhteystiedot ja tiedot

5.1.4 Esineen arvojen luominen

Luodun MQTT- pohjaa käyttävä esine on kaiken ydin. Se vastaanottaa datan, jota sensorit lähettävät, tallentaa datan haluttuun tiedostoon ja on yhteydessä haluttuun

MQTT- Broker palvelimeen. Tälle esineelle annettiin kuviossa 35 nimeksi MQTT_H4222 opiskelijatunnukseni mukaan.



The screenshot shows the configuration page for an entity named MQTT_H4222. The interface includes a top navigation bar with 'Save', 'Cancel Edit', and 'To Do' buttons. A left sidebar lists various entity management options like Properties, Services, Events, Subscriptions, Configuration, Home Mashup, Permissions, and Change History. The main content area is titled 'General Information' and contains two panels. The left panel lists entity details: Name (MQTT_H4222), Description (MQTT master for H4222), Project (Ala-Lahti), Tags (Search Mode), Thing Template (MQTT), and Implemented Shapes (Search Thing). The right panel shows configuration options: Active (checked), Home Mashup (Temp and Humidity map), Avatar (Change), Published (unchecked), Identifier (Browse...), Last Modified Date (2017-08-10 13:53:29.456), and Value Stream (ValueStream_H4222).

Kuvio 35. MQTT-osien asetusten asettaminen

Kuviosta 36 löydettiin esineen perustiedot täytettynä. Tiedosto on mukana Ala-Lahti-projektissa, sisältää MQTT- ominaisuuksia, käytettiin Temp and Humidity map- mashupissa ja tallennettiin saadut tiedot ValueStream_H4222 tiedostoon.

The screenshot displays the 'Properties' configuration page for an MQTT_H4222 entity. At the top, there are buttons for 'Save', 'Cancel Edit', and 'To Do'. Below this is a navigation menu on the left with categories like 'General Information', 'Services', 'Events', 'Subscriptions', 'Configuration', 'Home Mashup', 'PERMISSIONS', 'Visibility', 'Design Time', 'Run Time', 'CHANGE HISTORY', and 'DEPENDENCIES'. The main area is titled 'Properties' and contains a table of 'My Properties'.

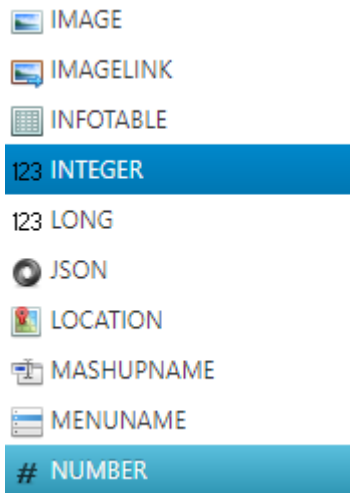
Edit	Name	Type	Alerts	Additional Info	Default Value	Value	DataChange
	# temp1		0 Alerts			22.0	Set Value: 0
	# humid1		0 Alerts			37.0	Set Value: 0
	# temp2		0 Alerts			21.0	Set Value: 0
	# humid2		0 Alerts			35.0	Set Value: 0
	# temp3		0 Alerts			21.0	Set Value: 0
	# humid3		0 Alerts			30.0	Set Value: 0
	# sauna		0 Alerts			0.0	Set Value: 0
	New Property		0 Alerts				Set Value

Below the table is the 'New Property' configuration form. It includes sections for 'General Property Info' (Name, Description, Category), 'BaseType Info' (Base Type: # NUMBER, Units, Min Value, Max Value, Has Default Value), 'Aspects' (Persistent, Read-only, Logged), and 'Data Change Info' (Data Change Type, Change Threshold). At the bottom right, there are 'Cancel', 'Done', and 'Done and Add' buttons.

Kuvio 36. Tietojen hakemisen asetukset

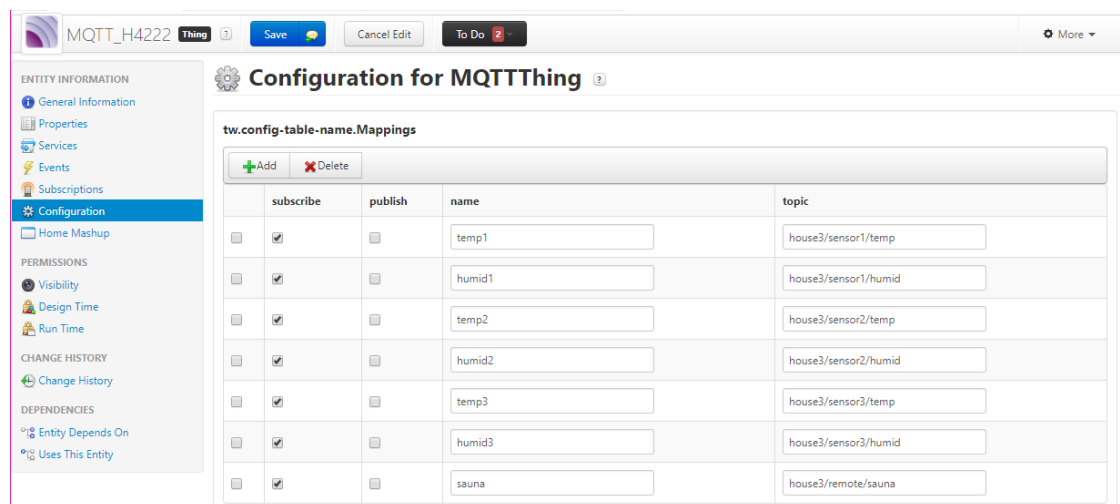
Kuviossa 37 voidaan nähdä luodut tunnisteet, joilla esineiden arvot voidaan tunnistaa. Temp1 tarkoittaa ensimmäisen sensorin lämpötilaa ja humid3 kolmannen sensorin kosteutta. Ominaisuuden arvo voidaan lukea sarakkeesta Value. Nämä tiedot päivittyvät, mikäli uutta tietoa saadaan.

Uuden ominaisuuden luontiin kysytään tietoja, joilla se voidaan tunnistaa, saatavan arvon tai datan arvoa. Oikealla löytyvä Logged on työssä erittäin tärkeä, sillä sen asettamalla päälle, kaikki saadut arvot tallennettiin ennalta määriteltyn tiedostoon. Base Type määrittää, mitä ominaisuus käsittelee. Tämä voi olla esimerkiksi kuva tai gps- reittitieto.



Kuvio 37. Tietomuodot luotaville tiedoille

Jotta esineen ominaisuudet saavat arvon, asetuksista jokaiselle ominaisuudelle säädettiin oma nimi ja topic. Ne asetettiin myös kuviossa 38 jokaisen subscribe-tilaan. Tällöin kuunteltiin asetettua topickia. Jotta arvo saadaan halutulle ominaisuudelle, tulee näiden nimet olla samat.



Kuvio 38. Tarkemmat polut tietojen hakemiselle

Seuraavaksi asetettiin palvelimen tiedot, mihin oltiin yhteydessä. Mikäli halutaan, että tieto on suojattu ja vain tietty käyttäjä voi lähettää palveluun arvoja, voidaan käyttää SSL-salausta hyväksymällä useSSL- kohdan. Koska lämpötilojen lähetys ei kuitenkaan ole salaista, voidaan syöttää kohtaan serverName kohtaan palvelimen

osoite. Erityisesti kannattaa ottaa huomioon, mitkä arvot syötetään kohtiin connectTimeout ja retryInterval. Mikäli näitä tietoja ei muuteta, ei esine ota enää vuorokauden jälkeen vastaan minkäänlaisia tietoja. Koska halutaan vastaanottaa tietoja pitkältä aikaväliltä, asetetaan kuviossa 39 kohtaan retryInterval arvon 20 sekuntia. Tällöin palvelu yrittää yhdistää palvelimeen 20 sekunnin välein, mikäli yhteyttä ei ole. Tällöin ollaan yhteydessä suurimman osan ajasta.

JDBC Settings

Name	Value
clientIdFormat	<input type="text" value="/Thingworx/{s}/t"/>
password	<input type="button" value="Change Password"/>
qos	<input type="text" value="0"/>
connectTimeout	<input type="text" value="300"/>
serverName	<input type="text" value="broker.hivemq.com"/>
retryInterval	<input type="text" value="20"/>
serverPort	<input type="text" value="1883"/>
userId	<input type="text"/>
timeout	<input type="text" value="5000"/>
useSSL	<input type="checkbox"/>

Kuvio 39. Brokerin asetukset thingworxissa

Mikäli kaikki säädöt on tehty oikein, pitäisi kohdassa Properties ja MQTT properties olla arvona true kuvion 40 mukaisesti. Tämä tarkoittaa, että ollaan yhteydessä palvelimeen ja ollaan valmiita vastaanottamaan dataa.

MQTT (ThingTemplate) - Properties

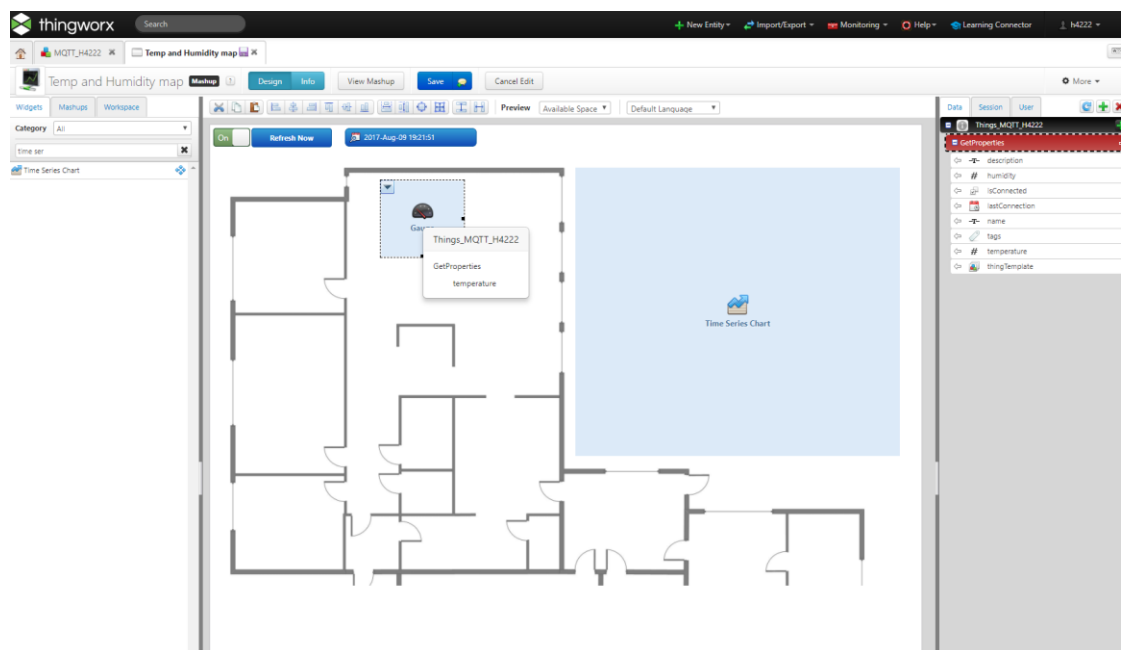
Connectable

Name	Type	Alerts	Additional Info	Default Value	Value	DataChange
lastConnection		0 Alerts			2017-08-10 13:53...	
isConnected		0 Alerts		false	true	

Kuvio 40. Thingworx on onnistuneesti yhdistänyt brokeriin

5.1.5 Mashupin luonti

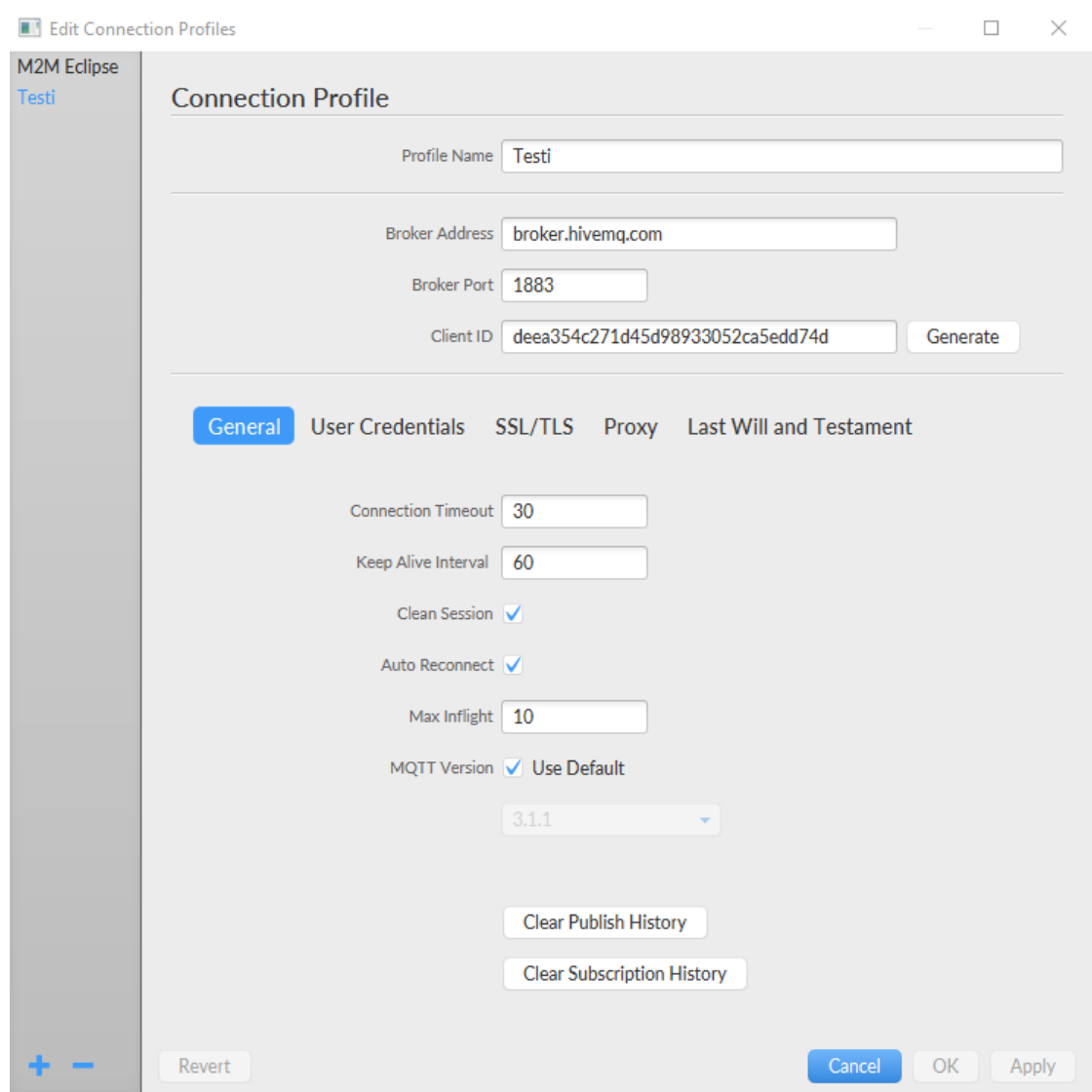
Mashup mahdollistaa datan esittämisen graafisesti. Mashup voi toimia myös interaktiivisesti, esimerkiksi mashup-näkymän kautta voidaan ladata tiedostoja palvelimelle tai antaa tietoja tekstimuodossa. Tässä työssä käytetään kuitenkin sensoridatan esittämiseen rakennuksen eri osissa ja kaikkien tilojen sensorien datahistorian esittämisessä. Mashup luodaan avaamalla oikealla ylhäällä oleva vihreä plus-merkki ja valitsemalla sieltä kohta mashup. Tällöin saadaan eteen tyhjä aukeama. Vasemmassa reunassa löytyy widgetit, joita voidaan lisätä mashupiin. Tämän alle ilmestyy valitun widgetin arvot, joita voidaan muuttaa. Ruudun alhaalta löytyvät mashupin kytkennät. Kytkennöistä voidaan seurata, mistä widget saa datansa, päivitetäänkö dataa tietyn väliajoin ja lähetetäänkö data johonkin widgetin jälkeen. Kuviossa 41 ruudun oikeassa reunassa on varattu tila esineille, joista voidaan tuoda data widgetteihin.



Kuvio 41. Tietojen liittäminen kuvaan mashupissa

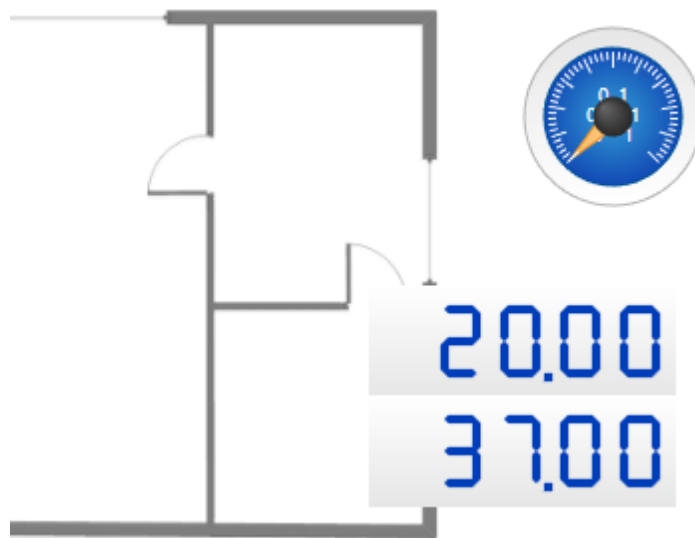
5.1.6 Datan lähetys palveluun

Teknisten ongelmien ja aikataulun vuoksi päädyttiin korvaamaan fyysiset sensorit ja toteuttamaan datan lähetyksen käyttämällä MQTT.fx ohjelmaa. Tällä saatiin korvattua fyysiset laitteet ja ohjattua palvelun lähettämään korvaavaa dataa palveluun. Palveluun luotiin javascript-koodi, joka löytyy liitteestä 1. Koodin tarkoitus on lähettää tietyn väliajoin jokaisen sensorin lämpötilat ja kosteudet kappaleen 3.3 mukaisesti. Tämän jälkeen pidettiin tauko, jota seurasi päivitettyjen tietojen lähetys. Näin saadaan mallinnettua oikean sensorin toimintaa ja päivityssyklejä ja pienennettyä mahdollisia yhteysvirheitä, joita saattaisi tulla sensorien sijoittelusta tai akunkestosta. Data saatiin lähetettyä brokerille, josta se pystytään hakemaan Thingworx- palveluun. Kuviossa 42 nähdään MQTT.fx asetukset.



Kuvio 42. MQTT.FX:n asetukset

Dataa lähetettiin palveluun myös kännykässä olevan mymqtt- sovelluksen avulla. Mikä tahansa MQTT:ä käyttävä sovellus pystyy lähettämään tietoja palveluun, joka antaa paljon erilaisia mahdollisuuksia yksinkertaisten palveluiden tekemiseen. Kännykän tarjoamaa palvelua käytettiin lähettämään 1 tai 0 topickiin house3/sauna joka on nähtävissä kuviossa 43. Tällä haluttiin esitellä mahdollisuus analysoida saunan ja saunan välikaton lämmön vaihtelevuutta toisiinsa. Kuviossa 44 nähdään komentorivimäisesti palvelun toiminta.



Kuvio 43. Saunan mittari mashupissa

```




MQTT.fx - 1.4.2
File Extras Help
Testi [Connect] [Disconnect]
Publish Subscribe Scripts Broker Status Log
2017-10-08 13:08:46,215 INFO --- Start App : Style: LIGHT /styles/mqttfx_theme_light.css
2017-10-08 13:08:46,495 INFO --- Start App : An update is available.
2017-10-08 13:08:48,709 INFO --- ScriptingManager : Found action with name: Switch Fountain Test
2017-10-08 16:52:34,784 INFO --- BrokerConnectorController : onConnect
2017-10-08 16:52:34,862 INFO --- MqttFX ClientModel : MqttClient with ID deea354c271d45d98933052ca5edd74d assigned.
2017-10-08 16:52:39,135 INFO --- PublishController : publish
2017-10-08 16:52:39,138 INFO --- MqttFX ClientModel : attempt to add PublishTopic
2017-10-08 16:52:39,139 INFO --- MqttFX ClientModel : successfully published message to topic house3/sensor1/temp (QoS 0, Retained false)
2017-10-08 16:52:45,234 INFO --- PublishController : publish
2017-10-08 16:52:45,235 INFO --- MqttFX ClientModel : attempt to add PublishTopic
2017-10-08 16:52:45,235 INFO --- MqttFX ClientModel : successfully published message 23 to topic house3/sensor1/temp (QoS 0, Retained false)

```

Kuvio 44. MQTT.FX:n toiminta komentorivillä

5.1.7 Datat liittäminen mashuppiin

Kuvioon 45 on lisätty haluttu pohjapiirustus, yhden sensoridataa näyttävän widgetin ja historiikkaa näyttävän widgetin. Mashuppiin on myös tuotu aikaisemmin luotu MQTT-esine, johon on ajettu saapuva data. Tiputusvalikosta on valittu esineeseen tuotavista arvoista haluttu ja vedetty se widgetiin. Tämän jälkeen on asetettu datat, joiden historia haluttiin katsoa Time Series Chart- widgettiin. Tämä onnistuu ruudun vasemmasta alanurkasta. Haluttu datat valittiin tiputusvalikosta ja sille annettiin kuvaava nimi. Esimerkiksi ensimmäisen sensorin lämpötiladatalle annettiin tieto olohuoneen lämpötilasta, joka viittaa sensorin sijoitukseen kiinteistössä. Tällöin sensorien lämpötilaa on helppo seurata.

DataField1	temp1
-T- DataLabel1 ⇄	Olkkarin Lämpötila
-T- SeriesType1	Use Chart Setting ▾
-T- SeriesMarkerType1	Use Chart Setting ▾
SeriesStyle1	 ✕
SeriesDataStyle1	State Formatting
DataField2	humid1
-T- DataLabel2 ⇄	Olkkarin kosteus
-T- SeriesType2	Use Chart Setting ▾
-T- SeriesMarkerType2	Use Chart Setting ▾
SeriesStyle2	 ✕
SeriesDataStyle2	State Formatting
DataField3	temp2
-T- DataLabel3 ⇄	Keittiön lämpötila
-T- SeriesType3	Use Chart Setting ▾
-T- SeriesMarkerType3	Use Chart Setting ▾
SeriesStyle3	 ✕

Kuvio 45. Mashuppiin lisätyt tiedot ja niiden nimet

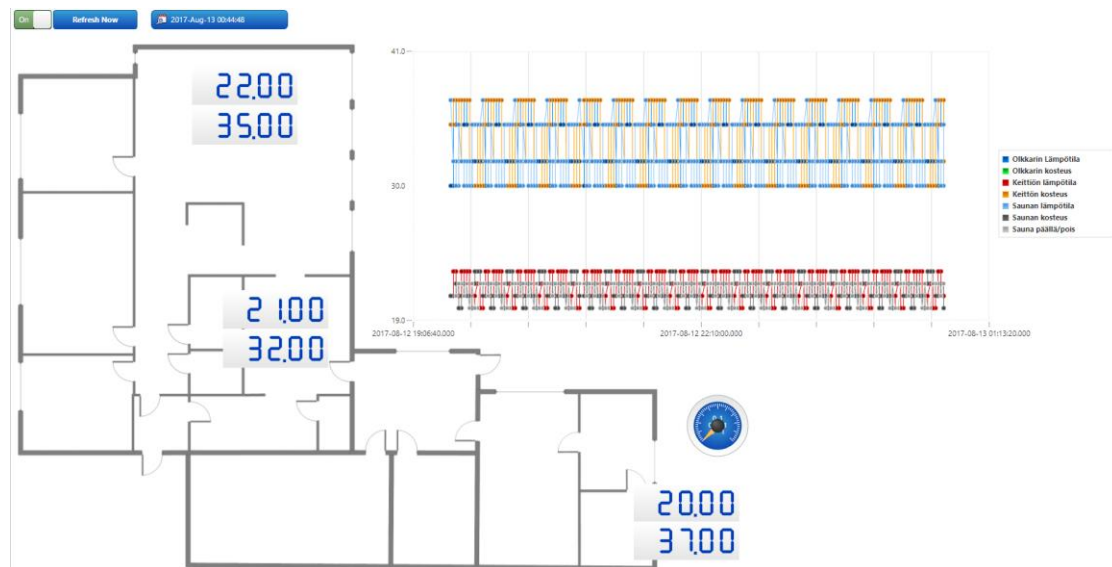
6 Testaus ja tulokset

Kun kaikki data on sijoitettu widgetteihin, tallennettu mashuppiin ja se tarkastettiin. Painettaessa painiketta View Mashup päästiin katsomaan miltä tuotos näytti toimivana kokonaisuutena. Koska kaikki säädöt olivat oikein lämpötilat ja kosteus vaihtuvat automaattisesti aina uuden datan saavuttua niin kuin kuviossa 46 näkyy. Myöskin tähän asti saatu data esitettiin oikealla puolella olevassa graafissa. Graafi päivittyi myös aina uuden datan saavuttua automaattisesti. Vastaanottokykyä testattiin myös lähettämällä nopeasti eri tietoa samalle näytölle. Kaikki mittarit vaihtoivat kuitenkin tietoja heti, kun uusin lukema oli julkaistu. Ruudun oikeassa reunassa oleva kaavio, joka muuttui vain asetetun aikavälin mukaan, ei kuitenkaan muuttanut päivitysvälillä saapuvan datan määrästä ja nopeudesta riippuen. Koska thingworx oli todella iso kokonaisuus, testattiin työssä vain kaikista kriittisimpiä ominaisuuksia, jota työssä

käytettiin. Nämä palvelut kuitenkin käytiin läpi kappaleessa 4.3 ja niiden toimivuus testattiin ohjetekstejä ja kuvia ottamista varten nopealla aikataululla.

Myös datan lähetystä testattiin, josta kerrottiin lisää kappaleessa 3.1.2. Tulevia jatkotutkimuksia varten on hyvä tietää toiminnan kaksisuuntaisuuden luomat mahdollisuudet. Työssä saatiin toimintatutkimuksen tavoitteen mukaisesti kartutettua koko matka datan luomisesta siirtämiseen ja sen esitykseen. Testauksessa saatiin selvitettyä myös hyvin eri sensorien toteutuksessa muodostuneet rajoitteet.

Pitemmän aikavälin tulosten näyttäminen on kuitenkin suoraan sidoksissa tiedon tallennuspaikkaan. Mikäli käytetään ulkoista tallennuspalvelinta ja yhteys näiden välillä katkeaa, voi vanhakin data kadota näkyvistä. Työssä kuitenkin käytettiin pelkästään Thingworxin tarjoamaa tallennustilaa, joten kaikki tallennettu data on samalla palvelimella työn kanssa.



Kuvio 46. Mashup toiminnassa

7 Pohdinta

Opinnäytetyön lopuksi jäi käteen paljon teoreettista sekä käytännöllistä tietoa. Työ oli ennen työn aloittamista tarkoitus toteuttaa siirtämällä anturidata käyttämällä

DHT11 lämpötila- ja kosteusanturia Arduinon ja ESP8266:n kanssa, mutta pitkän yrittämisen jälkeen tästä luovuttiin. Osasyynä luopumiseen oli se, että Arduinolla ei todennäköisesti pystytty tarjoamaan tarpeeksi virtaa osan käyttöön. Tähän olisi tarvittu erillinen virtalähde, joka ei ehtisi ajoissa opinnäytetyön palautusajankohtaan mennessä. Toinen syy sensoridatan emuloimiseen oli pitkään kestänyt ongelmanratkaisuperiodi osien ohjelmoinnin kanssa. Tähän kulutettiin työn tekemisen ajasta noin 50% joka oli jälkeinpäin mietittynä turhaa. Tämä on myös osasyynä opinnäytetyön palautuksen viivästymiseen suunnitellusta kesän 2017 lopusta. Koska tärkeintä työssä kuitenkin oli thingworxin esittely, päätin emuloida datan käyttämällä MQTT.FX -ohjelmaa, joka ajoi Javascript- ohjelmalla dataa sivulle. Tämä onnistuikin suhteellisen helposti ja työ saatiin eteenpäin. Mikäli työn sensorit voitaisiin nyt aloittaa uudelleen samalla ajalla, valittaisiin NodeMCU- korvaamaan Arduinoa ja ESP8266:sta heti, sillä yhteysongelmia näissä syntyi vähemmän. Vinkkejä Thingworxin käyttöön löytyi huonosti internetistä. Melkein kaikki ohjeet oli ohjattu suoraan nopeammin käyttöön otettavaan ja ilmaisiin palveluihin. Thingworx on kuitenkin pienen opetteluun jälkeen hyvin helppokäyttöinen ja yksinkertainen. Myös hyvät datan graafiseen esittämiseen annetut työkalut olivat iso plussa. Monipuolinen kauppapaikka, josta sai tarpeellisia lisäosia, oli hyvin tarpeellinen. Mikäli palvelua ei ollut valmiiksi, pystyi sen lataamaan ja asentamaan nopeasti kauppapaikasta.

Olen myös tyytyväinen thingworxin datan graafisen esityksen tarkkuuteen. Tästä voidaan tehdä myös omia johtopäätöksiä. Tästä esimerkkinä graafeissa näkyvä saunan sensorin lämpötila ja saunan mittari, joka näyttää joko saunan olevan päällä tai pois. Tästä voidaan tehdä toteamus, jossa saunan oleminen päällä vaikuttaa suoraan saunan lämpötilaan. Saunan sammuttamisen jälkeen lämpötila laskee hitaasti normaaliin huonelämpötilaan. Tämä tieto löydetään kätevästi suoraan graafeista ja niitä on helppo analysoida.

Opinnäytetyöni palauttamisen jälkeen on tutkijalla tarkoitus tutustua Thingworxiin ja rakentaa erilaisia järjestelmiä IoTin maailmassa. Parannusehdotuksia työlle olisi fyysisten sensorien ohjelmoinnin ja rakentamisen parempi dokumentointi ja testaus. Työn tarkoitus oli kuitenkin selvittää perusteet thingworx-palvelun toiminnasta, jossa raportti hyvin suoriutuu.

Lähteet

- Arduino Uno REV3. 2017. Viitattu 31.10.2017 <https://store.arduino.cc/usa/arduino-uno-rev3>
- Aosong(Guangzhou) Electronics Co.,Ltd. Temperature and humidity module. Viitattu 10.6.2017 <https://akizukidenshi.com/download/ds/aosong/DHT11.pdf>
- Benchoff, B. An sdk for the esp8266 WIFI chip. 2014. Viitattu 25.10.2014. <http://hackaday.com/2014/10/25/an-sdk-for-the-esp8266-wifi-chip/>
- Benchoff B, 2015, A Dev board for the esp lua interpreter, Viitattu 5.11.2017 <https://hackaday.com/2015/01/01/a-dev-board-for-the-esp-lua-interpreter/>
- Elisa launches new internet things service thingworx rapid application development platform. 2017. Viitattu 2.7.2017. <https://www.thingworx.com/about/news/elisa-launches-new-internet-things-service-thingworx-rapid-application-development-platform/>
- Hilitski, F. 2016. Connecting MQTT.fx to AWS IoT. Viitattu 9.11.2017. <http://www.fhilitski.com/2016/09/connecting-mqtt-fx-to-aws-iot/>
- Hirsijärvi, S. Remes, P. Sajavaara, P. 2007. Tutki ja kirjoita. 14 p. Helsinki: Tammi.
- Housley, R. 1999. Internet X.509 Public Key Infrastructure. Viitattu 11.11.2017. <https://www.ietf.org/rfc/rfc2459.txt>
- IBM. Eurotech, MQTT V3.1 Protocol Specification. Viitattu 8.10.2017. <https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>
- Koivula, U. Suihko, K. & Tyrväinen, J. 2002 Mission: Possible. Opas opinnäytteen tekijälle, Pirkanmaan ammattikorkeakoulu.
- Lady, A. DHT11,DHT22 and AM2302 Sensors. Julkaistu 29.7.2012. Muokattu 20.10.2017. Viitattu 31.10 2017. <https://learn.adafruit.com/dht/overview>
- MSV, J. 2016. Viitattu 10.11.2017. <https://thenewstack.io/messaging-reliability-persistence-mqtt/>
- MQTT Essentials Part 6: Quality of Service 0, 1 & 2 2017. Viitattu 11.11.2017. <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels>
- MQTT Security Fundamentals: MQTT Payload Encryption. 2017. Viitattu 10.11.2017. <https://www.hivemq.com/blog/mqtt-security-fundamentals-payload-encryption>
- MyMQTT 1.0. 2017. Viitattu 11.11.2017. <https://mymqtt.soft112.com/>
- OASIS Open. 2014. Viitattu 11.11.2017. <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf>.
- Statista. 2017. IoT number of connected devices worldwide. Viitattu 29.10.2017. <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>

Thingworx Delivers Industrial Innovation. 2017. Viitattu 10.11.2017.

<https://www.ptc.com/en/products/iot>

What is Arduino? 2017. Viitattu 10.11.2017

<https://www.arduino.cc/en/guide/introduction>

Woolf, N. DDoS attack that disrupted internet was largest of its kind in history, experts say. 2016. Viitattu 29.10.2017.

<https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet>

Liitteet

Liite 1. Antureiden datan emulimiseen käytetty skripti

```

var Thread = Java.type("java.lang.Thread");
function execute(action) {
  out("Test Script: " + action.getName());
  for (var i = 0; i < 999999999; i++) {
    sensor1temp1();
    Thread.sleep(500);
    sensor1humid1();
    Thread.sleep(500);
    sensor2temp1();
    Thread.sleep(500);
    sensor2humid1();
    Thread.sleep(500);
    sensor3temp1();
    Thread.sleep(500);
    sensor3humid1();
    Thread.sleep(100000);
  }
  action.setExitCode(0);
  action.setResultText("done.");
  out("Test Script: Done");
  return action;
}

function sensor1temp1() {
  out("Sensori 1 lamptila on 20°C");
  mqttManager.publish("house3/sensor1/temp", "20");
}

function sensor1temp2() {
  out("Sensori 1 lampotila on 21°C");
  mqttManager.publish("house3/sensor1/temp", "21");
}

function sensor1temp3() {
    out("Sensori 1 lamptila on 22°C");
    mqttManager.publish("house3/sensor1/temp", "22");
  }

function sensor1temp4() {
    out("Sensori 1 lampotila on 23°C");
    mqttManager.publish("house3/sensor1/temp", "23");
  }

function sensor1humid1() {
  out("Sensori 1 kosteus on 30%");
  mqttManager.publish("house3/sensor1/humid", "30");
}

function sensor1humid2() {
  out("Sensori 1 kosteus on 32%");
  mqttManager.publish("house3/sensor1/humid", "32");
}

function sensor1humid3() {
  out("Sensori 1 kosteus on 35%");
  mqttManager.publish("house3/sensor1/humid", "35");
}

function sensor1humid4() {
  out("Sensori 1 kosteus on 37%");
  mqttManager.publish("house3/sensor1/humid", "37");
}

function sensor2temp1() {
  out("Sensori 2 lampotila on 20°C");
  mqttManager.publish("house3/sensor2/temp", "20");
}

function sensor2temp2() {

```

```

    out("Sensori 2 lampotila on 21°C");
    mqttManager.publish("house3/sensor2/temp", "21");
}
function sensor2temp3() {
    out("Sensori 2 lampotila on 22°C");
    mqttManager.publish("house3/sensor2/temp", "22");
}
function sensor2temp4() {
    out("Sensori 2 lampotila on 23°C");
    mqttManager.publish("house3/sensor2/temp", "23");
}
function sensor2humid1() {
    out("Sensori 2 kosteus on 30%");
    mqttManager.publish("house3/sensor2/humid", "30");
}
function sensor2humid2() {
    out("Sensori 2 kosteus on 32%");
    mqttManager.publish("house3/sensor2/humid", "32");
}
function sensor2humid3() {
    out("Sensori 2 kosteus on 35%");
    mqttManager.publish("house3/sensor2/humid", "35");
}
function sensor2humid4() {
    out("Sensori 2 kosteus on 37%");
    mqttManager.publish("house3/sensor2/humid", "37");
}
function sensor3temp1() {
    out("Sensori 3 lampotila on 20°C");
    mqttManager.publish("house3/sensor3/temp", "20");
}

function sensor3temp2() {
    out("Sensori 3 lampotila on 21°C");
    mqttManager.publish("house3/sensor3/temp", "21");
}
function sensor3temp3() {
    out("Sensori 3 lampotila on 22°C");
    mqttManager.publish("house3/sensor3/temp", "22");
}
function sensor3temp4() {
    out("Sensori 3 lampotila on 23°C");
    mqttManager.publish("house3/sensor3/temp", "23");
}
function sensor3humid1() {
    out("Sensori 3 kosteus on 30%");
    mqttManager.publish("house3/sensor3/humid", "30");
}
function sensor3humid2() {
    out("Sensori 3 kosteus on 32%");
    mqttManager.publish("house3/sensor3/humid", "32");
}
function sensor3humid3() {
    out("Sensori 3 kosteus on 35%");
    mqttManager.publish("house3/sensor3/humid", "35");
}
function sensor3humid4() {
    out("Sensori 3 kosteus on 37%");
    mqttManager.publish("house3/sensor3/humid", "37");
}

function out(message){
    output.print(message);
}

```