

Henri Hakala

LÄMPÖTILANVALVONTAJÄRJESTELMÄ OSANA ESINEIDEN INTERNETIÄ

Kandidaatintyö
Tekniikan ja luonnontieteiden tiedekunta
Tarkastaja: Mikko Salmenperä
Huhtikuu 2023

TIIVISTELMÄ

Henri Hakala: Lämpötilanvalvontajärjestelmä osana esineiden internetiä
Kandidaatintyö
Tampereen yliopisto
Teknisten tieteiden kandidaatin tutkinto-ohjelma, Automaatiotekniikka
Huhtikuu 2023

Esineiden internet (engl. Internet of Things, lyh. IoT) on laaja ja jatkuvasti kasvava konsepti, joka on kehittynyt erittäin nopeaa vauhtia 2000-luvun loppupuolelta lähtien. Esineiden internet koostuu älykkäistä laitteista ja sovelluksista, jotka kommunikoivat keskenään verkon yli. Tärkeänä osana esineiden internetiä on myös saada informaatiota ihmisen rajapintaan, jotta sitä voidaan hyödyntää. IoT-järjestelmä voi koostua useista eri laitteista tai sisältää monia alijärjestelmiä. Jokaiselta IoT-järjestelmän osa-alueelta vaaditaan kykyä kommunikoida keskenään ilman ihmisen suorittamia toimintoja.

Kandidaatintyön tavoitteena oli selvittää mitä vaatimuksia esineiden internetin järjestelmiltä vaaditaan. Kandidaatintyö toteutettiin pääosin kirjallisuuskatsauksena. Kirjallisuuskatsauksessa käytettiin aineistona aiheeseen liittyviä tieteellisiä artikkeleita, kirjoja ja laite- sekä protokollavalmistajien verkkosivuja. Kandidaatintyön tarkoituksena oli toteuttaa yksinkertainen IoT-järjestelmä kirjallisuuskatsauksesta saatujen tulosten perusteella. Käytännön työn avulla voitiin osoittaa IoT-järjestelmän toimivuus todenmukaisessa ympäristössä.

Kandidaatintyö koostuu kirjallisuuskatsauksesta sekä käytännön työstä. Kirjallisuuskatsauksessa perehdyttiin erityisesti esineiden internetin rakenteeseen ja sen arkkitehtuuriin, sekä järjestelmien vaatimuksiin laitteiston ja tiedonsiirron näkökulmasta. Käytännön työn osuudessa toteutettiin yksinkertainen lämpötilanvalvontajärjestelmä. Käytännön työn toteuttamiseen käytettiin ennalta määrättyä laitteistoa sekä tiedonsiirto-protokollaa, joka valikoitui järjestelmältä vaadittujen ominaisuuksien perusteella.

Kirjallisuuskatsauksessa tehdyn selvityksen perusteella tärkeimmät tekijät järjestelmän toimivuuden kannalta ovat vakaa internet-yhteys, yhtenäinen tiedonsiirto-protokolla, selkeä arkkitehtuuri ja hyvä tietoturva. Tulosten perusteella voidaan päätellä, että laiteympäristö sekä järjestelmän käyttötarkoitus huomioiden oikean tiedonsiirto-protokollan valinta on erittäin tärkeää. Käytännön työn perusteella voidaan todeta, että vaatimusten mukainen järjestelmä voidaan toteuttaa ja liittää osaksi esineiden internetiä. Sopivin tapa toteuttaa yksinkertainen vaatimukset täyttävä lämpötilanvalvontajärjestelmä on MQTT (Message Queue Telemetry Transport) -tiedonsiirto-protokolla, Raspberry Pi 4 -tietokone ja RuuviTag-anturit.

Avainsanat: Esineiden Internet, IoT, IoT-järjestelmä, Tiedonsiirto, Tiedonsiirto-protokolla

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

SISÄLLYSLUETTELO

1. JOHDANTO	1
2. ESINEIDEN INTERNET	3
2.1 Historiaa	3
2.2 IoT:n määritelmä	3
2.3 Käyttökohteita	4
2.4 Saatavat hyödyt	5
3. JÄRJESTELMÄ OSANA ESINEIDEN INTERNETIÄ	7
3.1 Vaatimukset järjestelmältä	7
3.1.1 Tiedonsiirto	7
3.1.2 IoT-arkkitehtuuri	8
3.1.3 Tietoturva	9
4. TOTEUTETTAVAN JÄRJESTELMÄN TEKNOLOGIAT	10
4.1 Laitteisto	10
4.1.1 Raspberry Pi 4	10
4.1.2 RuuviTag	11
4.2 Ohjelmointikieli	12
4.3 Tiedonsiirtoprotokollien vertailu ja valinta	13
4.3.1 MQTT	13
4.3.2 CoAP	14
4.3.3 OPC UA	14
4.3.4 Valittu tiedonsiirtoprotokolla	15
5. LÄMPÖTILANVALVONTAJÄRJESTELMÄ	16
5.1 Käytännön työn tavoite	16
5.2 Vaatimukset järjestelmältä	16
5.3 Järjestelmän esittely	17
5.3.1 Arkkitehtuuri	17
5.3.2 RuuviTag-antureiden ohjelma	18
5.3.3 Ihmisen rajapinnan ohjelma	19
6. YHTEENVETO	20
6.1 Johtopäätökset	20
6.2 Ideoita järjestelmän jatkokehitykselle	20
LÄHTEET	22
LIITE A: ANTUREIDEN OHJELMISTOKOODI	24
LIITE B: IHMISEN RAJAPINNAN OHJELMISTOKOODI	25

KUVALUETTELO

Kuva 1.	<i>Esineiden internetin integraatio (muokattu lähteestä [17])</i>	4
Kuva 2.	<i>IoT-arkkitehtuurin neljä tasoa (muokattu lähteestä [19])</i>	8
Kuva 3.	<i>Työssä käytettävä Raspberry Pi 4 Model B -tietokone</i>	11
Kuva 4.	<i>Työssä käytettävät RuuviTag-anturit</i>	12
Kuva 5.	<i>MQTT julkaise/tilaa -arkkitehtuuri (muokattu lähteestä [9])</i>	13
Kuva 6.	<i>CoAP arkkitehtuuri (muokattu lähteestä [8])</i>	14
Kuva 7.	<i>OPC UA asiakkaan arkkitehtuuri (muokattu lähteestä [11])</i>	15
Kuva 8.	<i>Järjestelmän arkkitehtuuri</i>	17
Kuva 9.	<i>Lämpötilanvalvontajärjestelmän ihmisen rajapinta</i>	19

LYHENTEET JA MERKINNÄT

CoAP	Constrained Application Protocol
HTML	Hypertext Markup Language
IETF	The Internet Engineering Task Force
IoE	Internet of everything
IIoT	Industrial Internet of Things, teollinen esineiden internet
IoT	Internet of Things, esineiden internet
JSON	JavaScript Object Notation
MQTT	Message Queue Telemetry Transport
M2M	Machine to machine
OPC UA	OPC Unified Architecture
RAM	Random Access Memory, hajasaantimuisti
TCP/IP	Transmission Control Protocol / Internet Protocol
TLS/SSL	Transport Layer Security/Secure Sockets Layer
WLAN	Wireless local area network, langaton lähiverkko
XML	Extensible Markup Language
MAC	Media Access Control
C	Celsius

1. JOHDANTO

Internet-yhteyden muodostaminen on nykyään mahdollista lähes kaikkialla maailmassa, ja ihmiset ovatkin alkaneet pitää sitä jo melkein itsestäänselvyytenä. Lähes jokaiselta löytyy taskusta älypuhelin, joka on yhteydessä internetiin, ja keskivertosuomalaisen kodista saattaa löytyä jopa kymmeniä verkkoon yhteydessä olevia laitteita. Laitteiden määrän valtavan kasvun vuoksi viimeisen 20 vuoden aikana termi IoT (Internet of Things, esineiden internet) on muodostunut yhdeksi avainsanoista.

Esineiden internet on konseptina varsin uusi, ja se on teknologiana vasta kasvuvaiheessa. Yleisesti sillä tarkoitetaan järjestelmiä, joiden tiedonsiirto, seuranta ja ohjaus voidaan toteuttaa Internet-verkon kautta [5]. Vuoden 2021 loppuun mennessä asennettujen IoT-laitteiden määrän arvioidaan olevan lähes 36 miljardia laitetta maailmanlaajuisesti, ja sen ennustetaan kasvavan yli 75 miljardiin laitteeseen vuoteen 2025 mennessä. Vuodelle 2022 IoT:n markkina-arvoksi arvioitiin noin 212 miljardia maailmanlaajuisesti. [12]

Nopeasti digitalisoituva maailma ja IoT ovat mahdollistaneet sen, että lähes jokainen nykypäivänä toteutettu automaatio- ja tietotekninen sovellus on joiltain osin yhteydessä internetiin. Harvinaisempaa on löytää järjestelmä, joka ei ole yhteydessä internetiin, sillä uudet teknologiat mahdollistavat verkon yli kulkevaa tiedonsiirtoa tehokkaammin kuin koskaan aiemmin.

Järjestelmien, jotka ovat osana esineiden internetiä, täytyy täyttää tietyt vaatimukset. Jotta järjestelmä voi toimia osana esineiden internetiä, tiedonsiirron täytyy olla sujuvaa ja suurilta osin ihmisestä riippumatonta. Yhteisen tiedonsiirtotavan valinta järjestelmälle on avain sen toimivuuteen. Langattomassa tiedonsiirrossa tietoturva on yksi tärkeimmistä asioista, jonka täytyy olla kunnossa. Uusien tietoturvaa parantavien tekniikoiden myötä myös tietoturvaa kiertävät teknologiat ovat lisääntyneet yhtä nopeasti, joten tietoturvan pitää olla ajan tasalla. Myös ympäristöltä, jossa järjestelmä toimii, vaaditaan tiettyjä ominaisuuksia ja mahdollisuuksia.

Monet yrityksissä olevat tilat vaativat tarkkaa lämpötilanvalvontaa. Useita IoT-pohjaisia ratkaisuja löytyy jo markkinoilta. Lämpötilanvalvontajärjestelmä on hyvä esimerkki IoT-järjestelmästä, jossa tiedonsiirron reaaliaikaisuus on tärkeää. Lämpötilanvalvontajärjestelmää on myös helppo testata eri tilassa kuin missä sen on tarkoitus toimia.

Työn tavoite on tutkia järjestelmän liittämistä osaksi esineiden internetiä ja toteuttaa yksinkertainen järjestelmä tilanteeseen sopivan tiedonsiirtoprotokollan ja laitteiston avulla. Tarkoituksena ei ole toteuttaa mahdollisimman laajaa ja täydellisesti toimivaa järjestelmää, vaan tarkastella järjestelmältä vaadittuja ominaisuuksia ja sen paikkaa IoT-arkkitehtuurissa.

Työ koostuu kolmesta osasta. Ensimmäinen osa sisältää luvut 2 ja 3, joissa käsitellään taustatietoa ja työhön liittyvää teoriaa. Luvussa 2 käsitellään esineiden internetiä, sen historiaa ja ominaisuuksia. Luvussa 3 käsitellään järjestelmän liittämistä osaksi esineiden internetiä ja sitä, mitä ominaisuuksia järjestelmältä vaaditaan. Toinen osa työstä sisältää luvut 4 ja 5. Luvussa 4 tutkitaan mahdollisia toteutustekniikoita, joilla järjestelmä saadaan osaksi esineiden internetiä, sekä perehdytetään lukija laitteistoon, jolla käytännön osuus toteutetaan. Luvussa 5 esitellään käytännön työn toteutus valituilla toteutustekniikoilla. Kolmas ja työn viimeinen osuus sisältää luvun 6. Luvussa 6 esitellään saadut tulokset, yhteenveto työstä ja mahdollisia kehityskohteita järjestelmään.

2. ESINEIDEN INTERNET

Esineiden internet on erittäin laaja käsite, joka ei ole yksiselitteinen. Sen jatkuva kehittyminen aiheuttaa paljon eroa näkemyksiin siitä, mitä esineiden internetillä tarkoitetaan.

Tässä luvussa esitellään IoT:n (Internet of Things, esineiden internet) historiaa, määritelmää sekä mahdollisia käyttökohteita ja IoT:n vaikutusta kyseisissä käyttökohteissa.

2.1 Historiaa

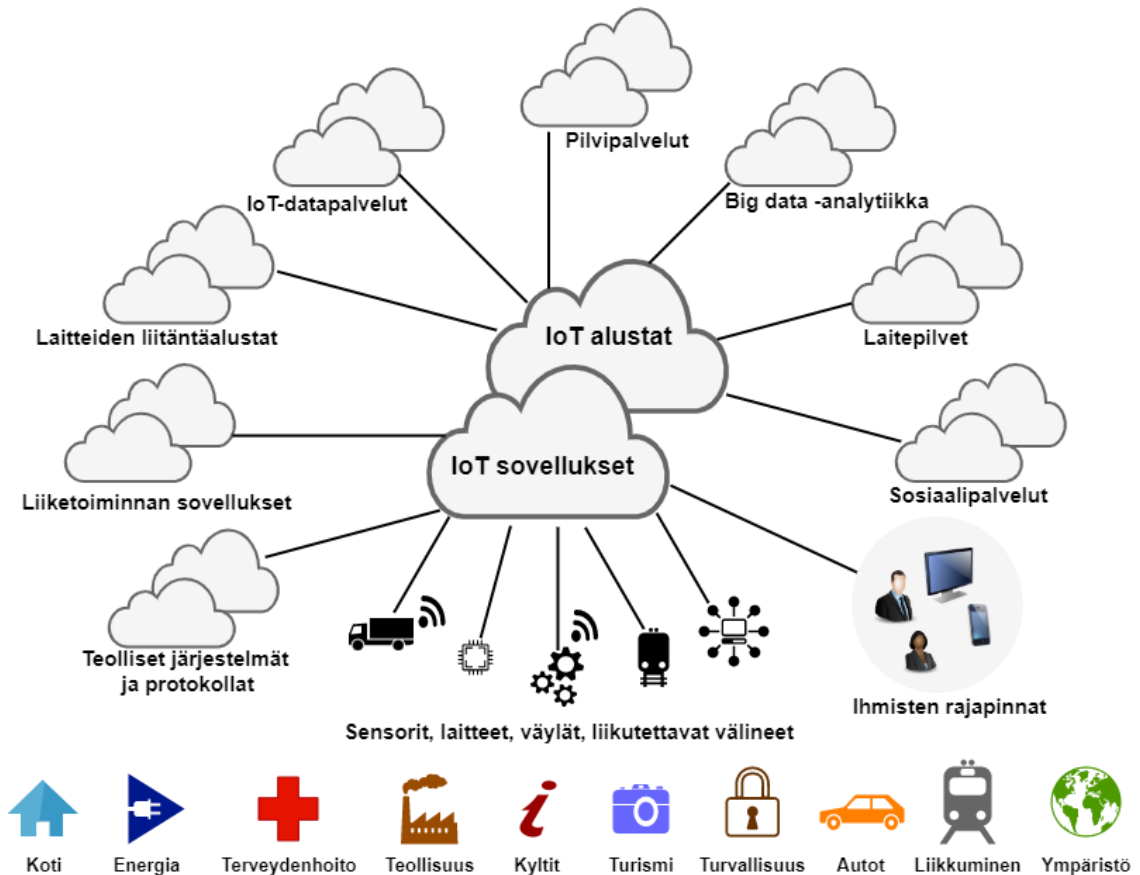
IoT:n aikakauden sanotaan varsinaisesti alkaneen 2000-luvun loppupuolella, jolloin internetiin yhteydessä olevien laitteiden määrä ylitti ihmisten väkiluvun maapallolla [6]. Suurin syy laitteiden valtavaan määrän lisääntymiseen oli älypuhelinien saapuminen markkinoille.

Termi "Internet of Things" oli esillä jo kuitenkin noin 10 vuotta aiemmin, kun Kevin Ashton toi sen esille. Ashtonia pidetään termin keksijänä, ja hän sai idean yrityksen tuotantoketjun yhdistämisestä internetiin työskennellessään Procter & Gamble -nimiselle yritykselle vuonna 1999 [6]. Ashton [1] kertoo haastattelussa, että ennen kyseistä hetkeä tietokoneet olivat vain aivoja ilman varsinaisia aisteja ja ne eivät tieneet mitään, mitä niille ei ole kerrottu.

2010-luvulta aina tähän päivään asti IoT-laitteiden määrän kasvu on ollut erittäin nopeaa. Uusia päälle puettavia laitteita, jotka ovat yhteydessä internetiin tulee jatkuvasti lisää. Näistä hyvinä esimerkkeinä toimivat esimerkiksi älykellot ja -rannekkeet ja uusimpina tulokkaina älysormukset.

2.2 IoT:n määritelmä

Yleisesti termillä IoT viitataan älykkyydellä varustettujen jokapäiväisten esineiden ja järjestelmien liittymisenä toisiinsa verkon yli [20]. IoT on konseptina erittäin laaja, eikä sille voi asettaa yhtä yksiselitteistä määritelmää, sillä kyseessä on jatkuvasti kehityksessä oleva teknologia. Pääasiassa silti aina kun puhutaan esineiden internetistä, kyseessä on laaja määrä toisiinsa liitettyjä älykkäitä laitteita ja sovelluksia, joiden yhdessä toimiminen ei ole ihmisen toimintojen varassa.



Kuva 1. Esineiden internetin integraatio (muokattu lähteestä [17])

Kuva 1 esittää hahmotelman esineiden internetin mahdollisuuksista. Kyseinen kuva määrittelee hyvin kokonaisuuden, jota kutsutaan esineiden internetiksi. IoT konseptina sisältää huomattavasti laajemman kokonaisuuden kuin vain älykkäät laitteet, jotka ovat liitettyinä toisiinsa. Huomattava määrä kuluttajalle näkymättömiä palveluita ovat avaintekijöitä koko esineiden internetin toimivuudessa. Pilvipalveluiden yleistymisen on aiheuttanut läpimurron tiedon jakamisen helppoudessa ja nopeudessa. Suurin osa IoT:n toiminnallisuutta onkin piilotettu erinäisiin pilvipalveluihin. Joskus IoT:stä puhuttaessa tulee esille termi IoE (Internet of Everything). IoE:n määritelmää voidaan pitää lähes samana kuin IoT:n määritelmää ja se määritellään ihmisten, prosessien, datan ja esineiden välisenä kommunikaationa [14].

Tavoitteena esineiden internetillä on siis kerätä valtava määrä tietoa ja jakaa sitä tarvittavien laitteiden, sovellusten ja ihmisten kesken. Lähestulkoon jokaisella toimialalla aina terveydenhuollosta julkiseen liikenteeseen IoT on nykypäivää.

2.3 Käyttökohteita

Teollisuus on ollut aina edelläkävijä internet pohjaisissa ratkaisuissa verrattuna yksityis- henkilöihin. Vuonna 1999 Ashtonin kehittämä tuotantoketjun yhdistäminen internetiin

vaikutti huomattavasti internetin käyttöön ja sen yleistymiseen teollisuudessa. Konsepti IloT(Industrial Internet of Things, teollinen esineiden Internet) onkin alkanut esiintyä teollisuudessa tämän yleistymisen myötä. IloT:lla tarkoitetaan sensoreita ja laitteistoa, jotka ovat yhteydessä teollisiin applikaatioihin yhteisen verkon kautta [3]. Käyttökohteita teknologialle onkin lähes rajaton määrä.

IoT löytyy nykypäivänä myös kaikkien kodeista. Samaan langattomaan verkkoon on mahdollista yhdistää laitteet televisioista ja älypuhelimista kahvinkeittimiin ja kiukaisiin. Saunan voi laittaa lämpenemään puhelimesta olevan sovelluksen kautta, ja kahvin lämpötilaa voi tarkkailla esimerkiksi tietokoneen näytöltä. Yleisimmin IoT näkyy nykyaikaisissa kotiautomaatiojärjestelmissä, ja niiden kautta on alettu puhua älykodeista. Kodin automaatiojärjestelmään voi olla yhteydessä esimerkiksi useita valvontakameroita, sisä- ja ulkolämpötilan anturit, valaistus ja lämmitys. Kaikkea tätä voidaan hallita esimerkiksi älypuhelimella verkon yli.

2.4 Saatavat hyödyt

IoT:n tarkoituksena on tuottaa lisäarvoa järjestelmille. Keräämällä mahdollisimman laajasti tietoa laitteista, niiden käyttötavasta ja käyttöympäristöstä saadaan tulevaisuudessa parannettua käyttäjäkokemusta ja parannettua järjestelmiä toimimaan paremmin niiden käyttötarkoituksessa. Hyvänä esimerkkinä laajasta tiedonkeruusta toimii digitaalinen kaksonen. Digitaalisen kaksosen ideana on simuloida järjestelmää mahdollisimman tarkasti, jotta sen käyttöikä ja mahdollisia virhetilanteita voidaan ennustaa jo ennen niiden tapahtumista. IoT on mahdollistanut digitaalisen kaksosen konseptina, sillä se mahdollistaa reaaliaikaisen tiedonkeruun useista eri laitteista ja informaatio voidaan koota yhdeksi suureksi tietokannaksi, josta digitaalinen kaksonen voidaan koota.

Yrityksissä markkina-arvon näkökulmasta on ensisijaisena tavoitteena parantaa sitä, miltä tuotteet ja palvelut näyttävät ulospäin nykyisille ja tuleville asiakkaille [16]. Digitaalinen kaksonen on hyvä esimerkki siitä, mitä palveluita yritykset voivat tarjota esineiden internetin ansiosta. Lisäarvon tuottaminen järjestelmälle IoT-pohjaisten ratkaisujen kautta hyödyttää sekä yritystä että asiakasta.

Yksityishenkilöillä IoT:n hyödyt näkyvät suurimmaksi osaksi helpotuksena arkisiin asioihin ja palveluiden lisääntymisenä. Fyysinen kontakti ja ajankäyttö laitteiden kanssa voidaan minimoida etäkäytöllä, kun operoitavan järjestelmän ei tarvitse olla käden ulottuvilla.

Tulevaisuudessa IoT:stä saatavat hyödyt lisääntyvät edelleen jatkuvasti kehittyvien teknologioiden myötä. Keskustelua konsepteista, joissa koneet voisivat hoitaa nykyään ihmistä vaativia asioita, on ollut jo kauan esillä. Esimerkiksi postin jakaminen tai kauppastosten kotiin tuominen voisivat olla mahdollisia asioita, jotka ainoastaan koneet hoitaisivat IoT:n ansiosta. Reaaliaikaisen tiedonsiirtämisen myötä näistä toiminnoista ihmisen poistaminen olisi varsin mahdollista.

3. JÄRJESTELMÄ OSANA ESINEIDEN INTERNETIÄ

Lähes mikä tahansa laite, jolla on toimiva internet-yhteys, voidaan liittää osaksi IoT:tä. Joitain vaatimuksia kyseiseen operaation on varsinkin tietoturvan osalta. Datan käsitteilyssä ja siirtämisessä on aina omat haasteensa, ja datan julkisuutta ja siirtotapoja valvotaankin erittäin tarkasti. Laitteiden liittäminen toisiin laitteisiin verkon yli kuulostaa yksinkertaisena nykypäivänä, mutta todellisuudessa mahdolliset yhteensopivuus ongelmat nousevat varsin usein pinnalle.

Tässä luvussa esitellään esineiden internetin arkkitehtuuria, tietoturvaa ja mitä vaatimuksia järjestelmältä vaaditaan, jotta se voi toimia osana esineiden internetiä. IoT-arkkitehtuuri on avainasemassa ja sen esittelyssä painotetaan järjestelmän paikkaa arkkitehtuurissa.

3.1 Vaatimukset järjestelmältä

Järjestelmä koostuu useista eri laitteista ja se pystytään rajaamaan toimintaympäristönsä. Järjestelmä voi koostua myös useista eri alijärjestelmistä. Jotta järjestelmä toimii oikein, sen sisäisten laitteiden täytyy jakaa yhteinen kommunikointitapa, sillä täytyy olla yhtenäinen ja selkeä arkkitehtuuri ja sen tietoturvan täytyy olla vahva verkon yli käytettävässä tiedonsiirrossa. Mahdollisia järjestelmän sisäisiä laitteita voivat olla esimerkiksi anturit, sensorit, tietokoneet ja kommunikointiväylät.

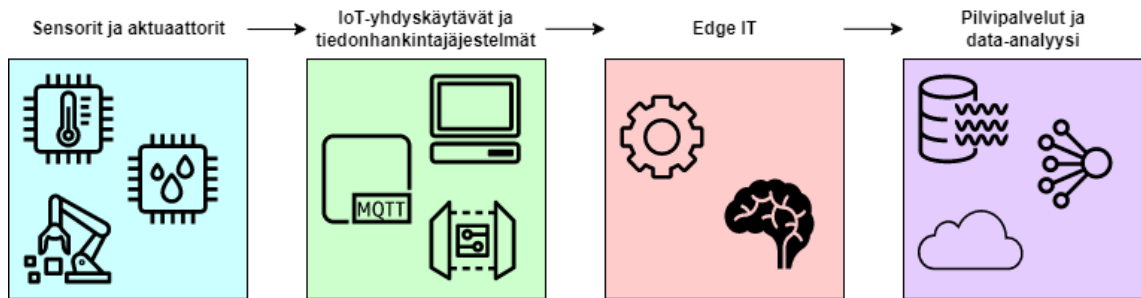
3.1.1 Tiedonsiirto

Suurin ratkaiseva tekijä puhuttaessa esineiden internetin sisäisestä järjestelmästä on tiedonsiirto ja kommunikaatio verkon yli. Ilman laitteiden välistä kommunikaatiota ei voida puhua IoT-järjestelmästä, sillä kokonaisuudessaan tarkoituksena on poistaa ihmisen suorittama tiedonsiirto laitteelta toiselle.

Kommunikaation yhtenäisyys on myös tärkeää. Samaa kommunikaatiotapaa käytettäessä, voidaan varmistua siitä, että tiedonsiirto on sujuvaa ja välttyään mahdollisilta ylimääräisiltä datan muokkauksilta. Yhtenäinen kommunikaatio takaa myös mahdollisuuden tehdä helppoa integraatiota järjestelmien välillä. Useimmat ongelmat järjestelmien fuusiossa liittyvät tiedonsiirtoon ja tapaan, jolla tietoa siirretään.

3.1.2 IoT-arkkitehtuuri

IoT:lle ei ole olemassa virallista standardisoitua arkkitehtuuria, joka olisi käytössä kaikkialla. Hyvän arkkitehtuurin valitseminen on erittäin tärkeää järjestelmän toimivuuden kannalta. Arkkitehtuurin tulisi mahdollistaa selkeä järjestelmän eri osa-alueiden rajaaminen, sekä järjestelmän muokkaaminen poistamalla tai lisäämällä laitteita ilman, että se vaatii koko arkkitehtuurin läpi käymistä. Hyvästä IoT-arkkitehtuurista on erilaisia näkemyksiä, mutta suurin osa niistä noudattaa samaa kaavaa.



Kuva 2. IoT-arkkitehtuurin neljä tasoa (muokattu lähteestä [19])

IoT-arkkitehtuuri jaetaan yleisimmin neljään eri tasoon Kuva 2 mukaan. Ensimmäinen taso koostuu sensoreista ja aktuaattoreista. Tämän tason laitteiden tehtävänä on kerätä informaatiota tai toimia saadun informaation perusteella.[19] Erilaiset sensorit ja anturit voivat kerätä esimerkiksi lämpötilasta, ilmanlaadusta tai ihmisten kulkemisesta. Aktuaattoreiden tehtävänä on suorittaa toimintoja kerätyn tiedon perusteella. Jotta aktuaattorille saadaan annettua komentoja, täytyy tiedon kulkea ensin eteenpäin sensoreilta.

Toinen IoT-arkkitehtuurin tasoista koostuu IoT-yhdyskäytävistä ja tiedonhankintajärjestelmistä. Yksinkertaisimpana esimerkkinä tiedonhankintajärjestelmästä toimii tietokone. Tiedonhankintajärjestelmän tehtävänä on kerätä dataa sensoreilta ja mahdollisesti tallentaa sitä ennen, kuin se siirretään yhdyskäytävää pitkin seuraavalle arkkitehtuurin tasolle.[19] IoT-yhdyskäytävä toimii arkkitehtuurissa molempiin suuntiin. Sen avulla tietoa siirretään sensoreilta tietokoneelle, josta taas yhdyskäytävää pitkin seuraavalle tasolle.

Kolmannen arkkitehtuurin tason nimi on Edge IT. Edge IT:n tarkoituksena on suodattaa ja esiprosessoida dataa, jotta pilvipalveluihin siirrettävä kuorma pienentyisi. IoT-järjestelmä voi toimia ilman kyseistä arkkitehtuurin tasoa, mutta siitä saatavat hyödyt ovat suuria.[19]

Arkkitehtuurin neljäs ja viimeinen taso sisältää pilvipalvelut ja data-analyysit. Informaation säilöminen pilvipalveluihin on avainasemassa hyödyn tuottamiseksi järjestelmän käyttäjille. Pilvipalveluihin voidaan sisällyttää tehokkaita työkaluja datan analysoimiseksi, kuten tekoälyn tai koneoppimisen avulla toimivia applikaatioita.[19]

3.1.3 Tietoturva

Tietoturva on yksi tärkeimmistä asioista puhuttaessa internetiin yhteydessä olevista järjestelmistä. Järjestelmä sisäinen tiedonsiirto on suojattava hyvin, sillä usein kyseessä on mahdollisesti arkaluontoista informaatiota kuten salasanoja, henkilötietoja tai prosessiparametreja. Tärkeää onkin tietää, kuinka laajasti tiedonsiirtoa täytyy turvata ja minkälainen tieto on arkaluontoisempaa kuin toinen.

IoT-laitteissa tiedon salausteknologia on yleisesti sisäänrakennettu tiedonsiirtoprotokollaan. Yksi yleisimmistä salausteknologioista internetin ylittävässä tiedonsiirrossa on TLS/SSL (Transport Layer Security/Secure Sockets Layer).

4. TOTEUTETTAVAN JÄRJESTELMÄN TEKNOLOGIAT

Tässä luvussa esitetään lukijalle tietoa käytännön työssä käytettävästä laitteistosta ja tutkitaan mahdollisia ratkaisumalleja IoT-järjestelmän tiedonsiirron kannalta. Toteutustekniikka kattaa fyysisen laitteiston sekä käyttäjälle näkymättömän osion laitteen toiminnallisuudesta kuten esimerkiksi tiedonsiirtoprotokollan ja ohjelmointikielen.

4.1 Laitteisto

Tässä kappaleessa esitellään järjestelmän toteuttamisessa käytettävä fyysinen laitteisto ja perehdytetään lukija laitteiston ominaisuuksiin. Käytettävä fyysinen laitteisto oli ennalta määritetty.

4.1.1 Raspberry Pi 4

Raspberry Pi 4 on noin korttipakan kokoinen tietokone, jota käytetään paljon pienprojekteissa sen koon ja yhteensopivuuden takia [18]. Raspberry Pi 4 pystyy käytännössä kaikkeen samaan kuin normaali tietokone, mutta sen suorituskyky on heikompi tilan optimoinnin takia. Raspberry Pi 4 sisältää USB-C-liitännällä olevan tehonlähteen, kaksi Micro HDMI -porttia, kaksi USB 2 -porttia, kaksi USB 3 -porttia ja gigabitin Ethernet-liitännän. Se on saatavilla erikokoisilla RAM-muistin (Random Access Memory, hajasaantimuisti) määrillä aina yhdestä gigabitistä kahdeksaan gigabittiin asti [13]. Kovalevynään se käyttää Micro SD -korttia.



Kuva 3. Työssä käytettävä Raspberry Pi 4 Model B -tietokone

Työssä Kuva 3 mukainen Raspberry Pi 4 toimii valvontajärjestelmän tietokoneena, jossa ohjelma pyörii ja siihen yhdistetään myös RuuviTag-sensoreita, jotka keräävät vaadittavaa mittausdataa.

4.1.2 RuuviTag

RuuviTag on langaton Bluetoothin avulla toimiva anturi, jolla voi mitata lämpötilaa, ilman-
kosteutta, ilmanpainetta ja liikettä [15]. RuuviTag on suosittu kuluttaja- ja yrityskäytössä,
ja sen avulla voi toteuttaa pienellä vaivalla mittauksia vaativia osioita esimerkiksi kotiau-
tomaatio käyttöön. Mitoiltaan RuuviTag on noin 5 senttimetriä halkaisijaltaan ja se painaa
30 grammaa. Paristolla toimiva anturia suositellaan käytettävän $-20\text{ }^{\circ}\text{C}$... $+70\text{ }^{\circ}\text{C}$ (Cel-
sius) käyttölämpötiloissa.[15]



Kuva 4. Työssä käytettävät RuuviTag-anturit

Työssä käytetään kolmea kappaletta Kuva 4 näkyviä RuuviTag-antureita. Langattoman teknologian ansiosta ne voidaan tilanteen mukaan sijoittaa tilan kannalta kriittisiin paikkoihin, joista halutaan mittausdataa.

4.2 Ohjelmointikieli

Python3 on tulkittava ohjelmointikieli, joka on helppo opetella. Sitä käytetään laajasti pienprojekteissa sen monipuolisuuden ja laajan kirjastovalikoiman ansiosta. Python3 kirjastoihin kuuluu vakiona useita internet protokollia kuten HTML (Hypertext Markup Language), XML (Extensible Markup Language) ja JSON (JavaScript Object Notation) sekä tuki sähköpostien käsittelyyn. [2]

Python3 valikoitui työssä käytettäväksi ohjelmointikieleksi puhtaasti perustuen yhteensopivuuteen sekä omiin vahvuuksiin ohjelmointikielissä. Valmiita kirjastoja ja laajaa dokumentaatiota RuuviTag-antureiden integroimiselle on laajalti tarjolla, mikä helpottaa huomattavasti niiden kanssa työskentelyä.

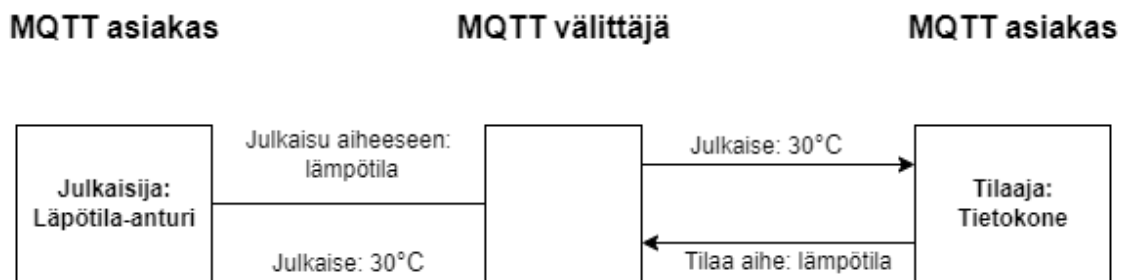
4.3 Tiedonsiirtoprotokollien vertailu ja valinta

IoT-järjestelmissä kaikki tiedonsiirto tapahtuu verkon yli, jolloin oikean tiedonsiirtoprotokollan valinta osoittautuu todella tärkeäksi. Protokollan valinta riippuu siitä, mitä järjestelmältä vaaditaan ja kuinka korkeaa suorituskykyä siltä haetaan.

Python3-ohjelmointikielestä löytyy paljon kirjastoja eri tiedonsiirtoprotokollille. Tässä osiossa vertaillaan kolmea eri protokollaa ja arvioidaan niiden sopivuutta yksinkertaisen järjestelmän toteuttamisen kannalta Raspberry Pi/Python3-ympäristössä.

4.3.1 MQTT

MQTT (Message Queue Telemetry Transport) on OASIS standardisoitu IoT-tiedonsiirtoprotokollaksi tarkoitettu julkaisemiseen ja tilaamiseen perustuva tiedonsiirtoprotokolla. Se on erittäin kevyt käyttöinen ja sitä suositetaan erityisesti laitteiden kommunikoinnissa silloin, kun verkon kaistanleveys on rajallinen ja ohjelmistokoodien koko on pientä. MQTT on laajasti käytössä useissa teollisuuden ratkaisuissa aina autoteollisuudesta tietoliikenne-ratkaisuihin. [9]



Kuva 5. MQTT julkaise/tilaa -arkkitehtuuri (muokattu lähteestä [9])

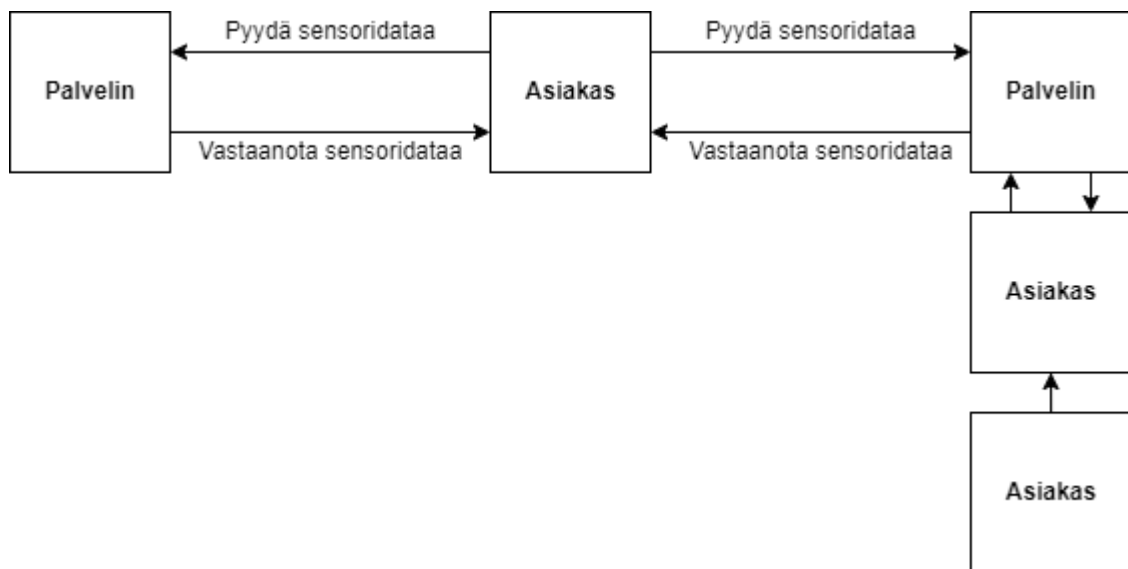
Kuva 5 havainnollistaa MQTT:n toimintaperiaatetta. Sen toiminta perustuu eri aiheisiin julkaisemiseen ja eri aiheiden tilaamiseen. Aiheeseen julkaisevaa asiakasta kutsutaan julkaisijaksi ja aiheita tilaavaa asiakasta tilaajaksi. Asiakas voi tilata useita aiheita ja vastaanottaa niihin julkaistua tietoa reaaliaikaisesti. Kaikki tieto kulkee MQTT-välittäjän kautta, jonka tehtävänä on ohjata tieto oikeaan osoitteeseen. Välittäjä mahdollistaa myös sen, että laitteiden ei tarvitse olla tiedonsiirtoprotokollan näkökulmasta tietoisia toisistaan, vaan niiden täytyy ainoastaan tunnistaa välittäjä, joka on yksi MQTT:n tuomista eduista. [7]

Mahdollisuus jatkuvalla datan seurannalle on yksi MQTT:n suurimmista vahvuuksista. Kerran luotu aihe on avoin ja siihen saa aina välittäjän kautta yhteyden. Tämän avulla

voidaan vähentää ylimääräistä liikennettä verkon yli ja saada tietoa siirrettyä tehokkaammin. Jatkuva datan seuranta on yksi tärkeimmistä vaatimuksista huomioon ottaen lämpötilanvalvontajärjestelmän perusvaatimukset, sillä joissain toimintaympäristöissä pienikin lämpötilaero voi aiheuttaa vakavia seurauksia.

4.3.2 CoAP

CoAP (Constrained Application Protocol) on IETF:n (The Internet Engineering Task Force) standardisoima tiedonsiirtoprotokolla. Protokolla on suunniteltu IoT -käyttöön ja etenkin M2M (Machine to machine) applikaatioihin. Sen toiminta perustuu pyyntö- ja vastausviesteihin. [4]

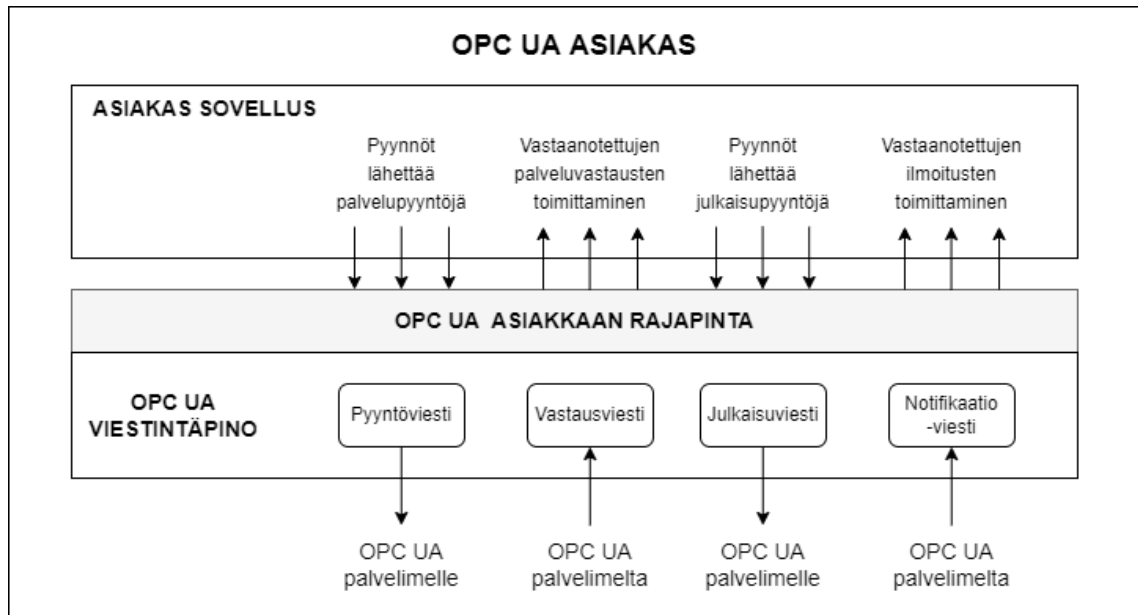


Kuva 6. CoAP arkkitehtuuri (muokattu lähteestä [8])

Kuva 6 havainnollistaa CoAP-protokollan toimintaperiaatetta. Pyyntö- ja vastausviesteihin perustuvassa tekniikassa data ei ole jatkuvasti saatavilla kuten MQTT-protokollan kanssa, vaan yhteys pitää aina avata uudelleen pyyntöviestillä.

4.3.3 OPC UA

OPC UA (OPC Unified Architecture) on vuonna 2008 julkaistu alustariippumaton palvelukeskeinen arkkitehtuuri. Sen toiminta perustuu asiakas/palvelin -tyyppiseen kommunikointiin pyyntö- ja vastausviestien avulla. [10]



Kuva 7. OPC UA asiakkaan arkkitehtuuri (muokattu lähteestä [11])

Kuva 7 havainnollistaa OPC UA -arkkitehtuuria asiakkaan näkökulmasta. Suurimpana erona MQTT- ja CoAP-protokollisiin OPC UA tarjoaa valmiin rajapinnan asiakkaan ja palvelimen väliseen kommunikaatioon. Haittapuolena rajapinnassa on se, että sitä ei pystytä muokkaamaan. Rajapinta on erittäin laaja ja se mahdollistaa paljon tarvittavia operaatioita. Kuitenkin pienprojekteissa ohjelmistokoodin koko kasvaa rajapinnan myötä tarpeettoman suureksi, eikä läheskään kaikkia rajapinnan ominaisuuksia tarvita.

4.3.4 Valittu tiedonsiirtoprotokolla

Tässä kappaleessa vertailluista tiedonsiirtoprotokollista MQTT valikoitui työssä käytettäväksi. MQTT sisältää käytännön työn toteuttamisen kannalta kaiken tarpeellisen toiminnallisuuden, sekä sillä on pienprojektin toteuttamisen näkökulmasta huomattavia etuja verrattuna kahteen muuhun vertailussa esiteltyihin protokollisiin. Python3 valinta ohjelmointikieleksi vahvasti päätöstä valita MQTT, sillä sen tuki Python ohjelmointikieleen on erinomainen. MQTT on protokollana myös erittäin kevyt ja se toimii hyvin rajallisella kaistanleveydellä, joka on hyvä ominaisuus jatkuvasti päällä olevassa järjestelmässä.

5. LÄMPÖTILANVALVONTAJÄRJESTELMÄ

5.1 Käytännön työn tavoite

Käytännön työn tavoitteena on toteuttaa yksinkertainen lämpötilanvalvontajärjestelmä, joka toimii osana IoT:tä. IoT:n määritelmän mukaan järjestelmän tulee olla kykeneväinen verkon yli tiedonsiirtoon ja tietoa pitää saada kuljetettua ihmisen rajapintaan. Työssä käytetään kappaleessa 4 esiteltyä laitteistoa, sekä valittuja toteutustekniikoita.

5.2 Vaatimukset järjestelmältä

Järjestelmä koostuu Raspberry Pi 4 tietokoneesta sekä kolmesta RuuviTag-anturista. Fyysiseltä kooltaan järjestelmä on erittäin pieni, joten se on mahdollista sijoittaa moniin erilaisiin tiloihin ja ympäristöihin. Rajoittavia tekijöitä järjestelmän sijoittamiselle ovat:

- Tietokoneen (Raspberry Pi) sijainti
 - a. Virtalähde vaatii verkkovirtaa, sijoitus pistorasian läheisyyteen.
 - b. Ylikuumenemisen vaara, ei liian ahtaaseen ja kuumaan tilaan.
 - c. Internet-yhteytys. Ethernet-kaapeli tai WLAN (wireless local area network, langaton lähiverkko).
- RuuviTag-antureiden sijainti
 - a. Bluetooth yhteys tietokoneeseen, mieluiten samassa tilassa kantaman sisällä.
- Ympäristö
 - a. RuuviTag-anturit ovat IP67 suojattuja, Raspberry Pi ei ole.
 - b. RuuviTag-anturit voidaan tarvittaessa sijoittaa ulkotiloihin, mutta Raspberry Pi tietokoneen täytyy olla kuivassa sisätilassa.

IoT-järjestelmän yksi tärkeimmistä tavoista tuottaa arvoa käyttäjälleen on saada kuljetettua tarpeellista tietoa loppukäyttäjälle asti, joten yksi selkeimmistä vaatimuksista järjestelmälle on helppo pääsy ihmisen rajapintaan.

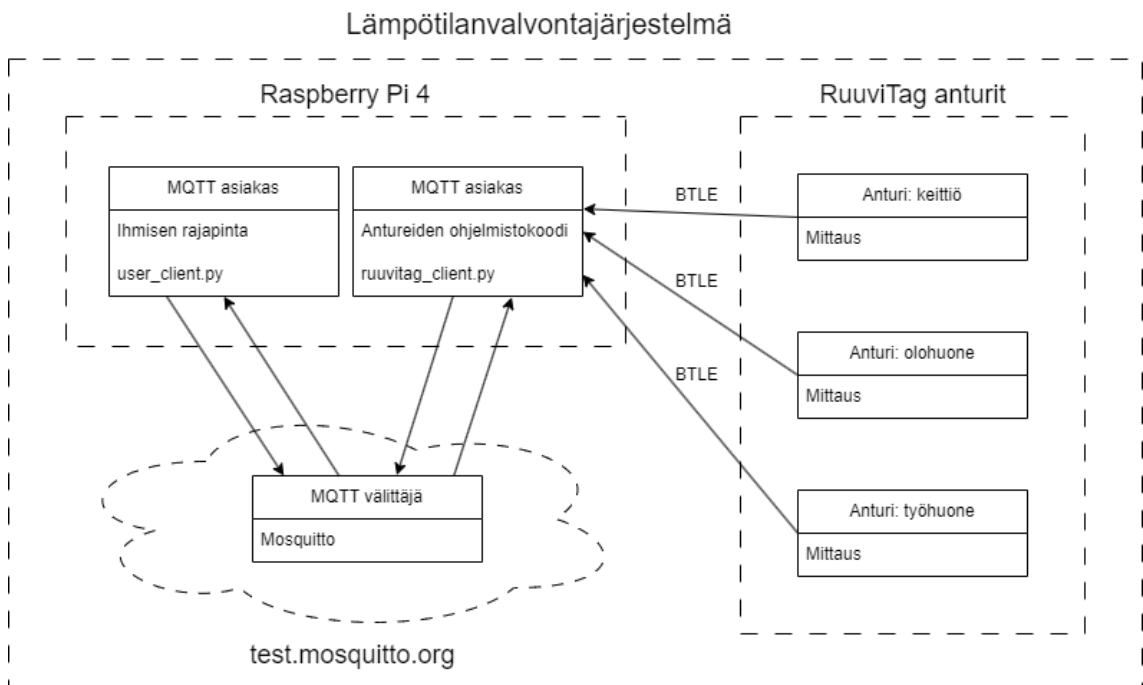
5.3 Järjestelmän esittely

Tässä kappaleessa esitellään valmiin järjestelmän lopputulosta arkkitehtuurin näkökulmasta, sekä esitetään lukijalle enemmän tietoa järjestelmän osa-alueiden toiminnasta. Järjestelmä toteutettiin toimimaan kodin sisäverkossa ja kodin sisätiloissa.

5.3.1 Arkkitehtuuri

Järjestelmässä päädyttiin käyttämään MQTT tiedonsiirtoprotokollaa, joten tiedonsiirron kannalta sen arkkitehtuuri on Kuva 5 mukainen. Kaikki järjestelmän käyttämä toiminnallisuus on keskitetty yhteen Raspberry Pi 4 tietokoneeseen. Tietokoneella ovat käynnissä ihmisen rajapinnan ja antureiden MQTT asiakasohjelmat.

Vaikka toteutetussa järjestelmässä sen kaikki osa-alueet ovat käynnissä samalla tietokoneella, on silti mahdollista käyttää esimerkiksi toista tietokonetta ihmisen rajapinnan MQTT asiakasohjelman suorittamiseen. MQTT välittäjää päädyttiin suorittamaan pilvessä *test.mosquitto.org* kautta. Jos tieto, jota julkaistaisiin aiheisiin, olisi arkaluontoista, olisi myös mahdollista suorittaa lokaalisti asentamalla Mosquitto-ympäristö suoraan Raspberry PI -tietokoneelle. Tällöin data pysyisi varmasti turvattuna ja vain samassa lähiverkossa olevat laitteet pääsisivät siihen käsiksi, jos Raspberry PI:n IP-osoite on tiedossa.



Kuva 8. Järjestelmän arkkitehtuuri

Kuva 8 esittää kokonaisuudessaan järjestelmän ylimmän tason arkkitehtuurin. Kolme RuuviTag-anturia sijoitettiin huoneistoon ennalta määrättyihin sijainteihin, joista jokainen lähettää jatkuvalla syklillä sijainti, lämpötila sekä ilmankosteus -tietoa. Data siirtyy Bluetooth yhteydellä Raspberry Pi -tietokoneella, jossa sen vastaanottaa MQTT-asiakkaana toimiva antureiden lähdekoodi *ruuvitag_client.py*, joka on esitelty liitteessä A. Data julkaistaan antureiden lähdekoodin toimesta MQTT välittäjän kautta erinäisiin aiheisiin. Ihmisen rajapinnan lähdekoodi *user_client.py*, toimii myös MQTT asiakkaana. Se tilaa anturiohjelmiston toimesta julkaistuja aiheita välittäjän kautta ja näyttää aiheissa olevan informaation reaaliaikaisena käyttöliittymällä. Ihmisen rajapinnan lähdekoodi on esitelty liitteessä B. MQTT liikenteeseen käytettiin avoimen lähdekoodin Paho-MQTT kirjastoa.

5.3.2 RuuviTag-antureiden ohjelma

RuuviTag-antureiden ohjelman pääasiallinen tarkoitus on vastaanottaa mittausdataa jokaiselta anturilta, muokata se oikeaan muotoon ja julkaista sitten MQTT avulla ennalta määrättyihin aiheisiin. Antureiden lähdekoodin toteutuksessa käytettiin hyödyksi Python3 kirjastoja *json*, *paho-mqtt* ja *ruuvitag-sensor*, joista kaksi jälkimmäistä ovat avoimen lähdekoodin kirjastoja MQTT protokollan toteuttamiseen, sekä RuuviTag antureiden ja tietokoneen väliseen kommunikointiin.

Jokainen anturi yksilöitiin niiden henkilökohtaisen MAC-osoitteen (Media Access Control) avulla, jolloin saatiin varmistettua datan julkaiseminen oikeaan aiheeseen. Yhteensä kolmeen aiheeseen, *olohuone/info*, *tyuhuone/info* ja *keittio/info* julkaistiin JSON-muotoista dataa seuraavasti:

```
{'sijainti': 'olohuone', 'lampotila': 20.93, 'ilmankosteus': 37.2}
{'sijainti': 'tyuhuone', 'lampotila': 20.67, 'ilmankosteus': 32.2}
{'sijainti': 'keittio', 'lampotila': 20.55, 'ilmankosteus': 31.57}
```

JSON-muotoista dataa on helppo käsitellä, sillä siihen valmiina olevan Python3 kirjaston avulla arvojen erottelu avaimen mukaan on yksinkertaista. Liitteessä A esitellyn lähdekoodin toimintaperiaate on erittäin yksinkertainen. Ohjelma toimii jatkuvatoimisesti pitäen MQTT-yhteyden auki ja hakemalla jatkuvasti tietoa RuuviTag-antureilta, kunnes se suljetaan. Liitteen A lähdekoodi on kommentoitu ja kommentteista selviää funktioiden ja toimintojen tarkempi kuvaus.

5.3.3 Ihmisen rajapinnan ohjelma

Ihmisen rajapinnan tehtävänä on IoT-järjestelmissä saada välitettyä tietoa loppukäyttäjälle, eli ihmiselle asti. Järjestelmään toteutettiin yksinkertainen ihmisen rajapinta lämpötila ja ilmankosteus -arvojen monitorointia varten. Liitteessä B esitetyn ihmisen rajapinnan ohjelman pääasiallinen tehtävä on tilata aiheet, joihin antureiden ohjelma julkaisee tietoa, ja näyttää reaaliaikaista tietoa tietokoneen näytöllä. Ohjelma toimii jatkuvasti, kunnes se suljetaan. Aina kun edellisessä kappaleessa mainittuihin aiheisiin julkaistaan dataa, niin monitori näkymä päivittyy.

Anturin sijainti	Lämpötila	Ilmankosteus
Keittiö	23.71 °C	30.74 %
Olohuone	24.25 °C	36.77 %
Työhuone	23.64 °C	31.82 %

Kuva 9. Lämpötilanvalvontajärjestelmän ihmisen rajapinta

Kuva 9 esittää kuvan lämpötilanvalvontajärjestelmän käyttöliittymästä. Käyttöliittymä on tarkoitettu pelkästään monitorointia varten. Ikkunassa esitetään reaaliaikaista tietoa eri antureiden tuottamasta mittausdatasta. Liitteessä B esitellyssä käyttöliittymän lähdekoodin toteutuksessa *user_client.py* käytettiin Python3 kirjastoja *tkinter*, *json* sekä *paho-mqtt*. Käyttöliittymän lähdekoodin funktioiden ja toiminnan tarkempi kuvaus näkyy liitteessä B olevissa kommentteissa koodin seassa.

6. YHTEENVETO

Tässä kandidaatintyössä tutkittiin esineiden internetiä ja selvitettiin mitä sen osana olevilta järjestelmiltä vaaditaan. Työn alussa tehtiin kirjallisuuskatsauksen keinoin selvitystä esineiden internetistä käsitteenä ja konseptina, sekä selvitystä siitä miten järjestelmä saadaan liitettyä osaksi IoT:tä. Työn lopussa toteutettiin käytännössä yksinkertainen lämpötilanvalvontajärjestelmä, joka toimi esimerkkinä IoT -järjestelmästä ja todisti teknikoiden käytännön toimivuuden.

6.1 Johtopäätökset

Kirjallisuuskatsauksen tulosten perusteella IoT on alati muuttuva käsite, jonka määritelmä ja käyttökohteet ovat jatkuvassa muutostilassa. Jatkuvasti kasvava IoT-laitteiden määrä muuttaa käsitystä siitä, mitä IoT pitää sisällään jatkuvasti. Järjestelmien liittäminen osaksi IoT:ta vaatii selkeän arkkitehtuurin, vakaan internet-yhteyden, yhtenäisen tiedonsiirtoprotokollan sekä hyvän tietoturvan. Kaikki nämä vaatimukset ovat erittäin tärkeässä asemassa toimivan ja luotettavan järjestelmän toteuttamisen kannalta.

Käytännön työn avulla osoitettiin todeksi vaatimusten paikkansapitävyys järjestelmän toteuttamisen keinoin. Järjestelmän toteuttamisessa haastavimmaksi asiaksi osoittautui Raspberry PI tietokoneen ja ohjelmointikielen kirjastojen yhteensopivuus. Käytännön työn toteuttamisessa käytettiin ennalta määrättyä laitteistoa, joka vaikutti paljon siihen, minkälainen järjestelmästä tuli. Lämpötilanvalvontajärjestelmästä tuli onnistunut yksinkertainen esimerkki IoT-järjestelmästä. Käytännön työn toteutuksessa tietoturvan osuus jäi vähäiseksi, sillä järjestelmän kommunikaatio tapahtui kodin sisäverkon sisällä.

6.2 Ideoita järjestelmän jatkokehitykselle

Toteutettu järjestelmä oli erittäin yksinkertainen ja sitä pystyisi laajentamaan. Järjestelmästä voisi toteuttaa esimerkiksi konfiguroitavan, jolloin sen käyttäjällä olisi mahdollisuus määritellä paikat RuuviTag-antureille ja päättää mitä dataa tuodaan näkyviin ihmisen rajapintaan. Toteutetun järjestelmän käyttäminen vaatii myös ulkoisen näyttöpäätteen yhdistämisen Raspberry PI tietokoneeseen. Käyttöliittymän voisi toteuttaa HTML-sivuna, jolloin kaikilla saman sisäverkon laitteilla olisi pääsy monitoroimaan antureiden antamaa tietoa. Mahdollisella reaaliaikaisen tiedon tallennuksella, voitaisiin myös päästä tarkastelemaan mittaushistoriaa.

Jos järjestelmä asennettaisiin tilaan, jossa pienilläkin lämpötila ja kosteus eroilla on merkitystä, siihen voisi toteuttaa myös hälytysjärjestelmän esimerkiksi sähköpostiviestien muodossa. Tietoturvaan on myös syytä perehtyä enemmän. Pilvipalvelun kautta suoritettava MQTT-välittäjä *test.mosquitto.org* tarjoaa myös suojattua tiedonsiirtoa, joka perustuu avainten vaihdolla tehtyyn tiedon välittämiseen. Jos siirrettävä tieto olisi arkaluontoisempaa, kuin lämpötila- tai ilmankosteustietoa, tätä olisi syytä käyttää.

LÄHTEET

- [1] A Gabbai. "Kevin Ashton describes the Internet of Things," Smithsonian Magazine. 2015. Saatavissa (viitattu 31.3.2022): <https://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/>
- [2] Applications for Python. Python Software Foundation. URL: <https://www.python.org/about/apps/>. (viitattu 23.4.2022)
- [3] H. Boyes, B. Hallaq, J. Cunningham, T. Watson. The industrial internet of things (IIoT): An analysis framework. Computers in industry. 2018;101:1–12.
- [4] CoAP, RFC 7252 Constrained Application Protocol. URL: <https://coap.technology/>. (viitattu 25.4.2022)
- [5] Esineiden internet. 18.2.2022. URL: https://fi.wikipedia.org/wiki/Esineiden_internet (viitattu 31.3.2022)
- [6] D. Hanes, G. Salgueiro, R. Barton. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things. 1st edition. Cisco Press. 2017.
- [7] GC. Hillar. MQTT essentials - a lightweight IoT protocol : the preferred IoT publish-subscribe lightweight messaging protocol. 1st edition. PACKT Publishing; 2017. s.11
- [8] S. Kuber. MQTT vs CoAP: What's the best protocol for IoT Solutions? EMnify. URL: <https://www.emnify.com/developer-blog/mqtt-vs-coap>. (viitattu 3.5.2022)
- [9] MQTT: The Standard for IoT Messaging. OASIS. URL: <https://mqtt.org/>. (viitattu 25.4.2022)
- [10] OPC Technologies, Unified Architecture. OPC Foundation. URL: <https://opcfoundation.org/about/opc-technologies/opc-ua/>. (viitattu 3.5.2022)
- [11] OPC UA Online Reference, 6.2 OPC UA Clients. OPC Foundation. URL: <https://reference.opcfoundation.org/v104/Core/docs/Part1/6.2/>. (viitattu 16.5.2022)
- [12] J. Stedward. The Ultimate List of Internet of Things Statistics for 2022. Findstack. Päivitetty 15.2.2022. URL: <https://findstack.com/internet-of-things-statistics/> (viitattu 5.4.2022)
- [13] Raspberry Pi 4. Raspberry Pi Foundation. URL: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (viitattu 5.4.2022)
- [14] A. Rayes, S. Salam. Internet of Things from Hype to Reality: The Road to Digitization. Cham: Springer International Publishing AG; 2022.
- [15] Ruuvitag tuotteen kotisivut. Ruuvi Innovations Oy. URL: <https://ruuvi.com/ruuvitag/> (viitattu 5.4.2022)

- [16] T. Saarikko, UH Westergren, T. Blomquist. The Internet of Things: Are you ready for what's coming? *Business horizons*. 2017;60(5):667–76. URL: <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S000768131730068X> . (viitattu 11.4.2022)
- [17] O. Vermesan, P. Friess, P. Guillemin, R. Giaffreda, H. Grindvoll, M. Eisenhauer, et al. *Internet of Things beyond the Hype: Research, innovation and deployment*. River publishers. 2015.
- [18] What is a Raspberry Pi? Raspberry Pi Foundtaion. URL: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/> (viitattu 5.4.2022)
- [19] N. Zubovich. What is IoT Architecture, 4 stages of IoT Architecture. Sumasoft. URL: <https://sumatosoft.com/blog/what-is-iot-architecture-4-stages-of-iot-architecture> . (viitattu 26.4.2022)
- [20] F. Xia, LT Yang, L. Wang, A. Vinel. Internet of Things. *International journal of communication systems*. 2012;25(9):1101–2.

LIITE A: ANTUREIDEN OHJELMISTOKOODI

```

import json
import paho.mqtt.client as mqtt
from paho.mqtt import publish
from ruuvitag_sensor.ruuvi import RuuviTagSensor

# Callback funktio yhteyden mahdollista menettämistä varten
def on_connect(client, userdata, flags, rc):
    print(f"Yhdistetty MQTT välittäjään palautusarvolla {str(rc)}")

    client.subscribe("$SYS/#")

# Funktio datan esikäsittelyyn, luodaan json muotoinen paketti MQTT viestiksi
# datapaketti julkaistaan aiheeseen, joka on määritelty MAC osoitteen ja si-
# jainnin perusteella
def handle_data(data):
    json_data = {
        "sijainti": tags[data[0]],
        "lampotila": data[1]['temperature'],
        "ilmankosteus": data[1]['humidity'],
    }
    print("Anturitieto vastaanotettu anturista: " + tags[data[0]])
    print(json_data)

    mqtt_topic = topics[data[0]]
    mqtt_msg = json.dumps(json_data)
    publish.single(mqtt_topic, mqtt_msg, hostname=mqtt_broker)

# Hakee dataa antureilta jatkuvasti
def main():
    macs = tags.keys()
    RuuviTagSensor.get_data_async(handle_data, macs)

if __name__ == "__main__":
    # MQTT välittäjän osoite
    mqtt_broker = "test.mosquitto.org"

    # MQTT aiheet MAC osoitteiden mukaan
    topics = {"CF:A3:0A:D0:44:2C": "keittio/info",
              "C5:A2:79:FB:54:2A": "tyuhuone/info",
              "FC:93:AE:0F:1E:3B": "olohuone/info"}

    # Ruuvitagien MAC osoitteet ja sijainnit kovakoodattuna
    tags = {"CF:A3:0A:D0:44:2C": "keittio", "C5:A2:79:FB:54:2A": "tyuhuone",
            "FC:93:AE:0F:1E:3B": "olohuone"}

    # Luodaan MQTT yhteys, sekä aloitetaan ohjelman looppi
    client = mqtt.Client()
    client.on_connect = on_connect
    client.connect(mqtt_broker, 1883, 60)
    client.loop_start()
    main()

```

LIITE B: IHMISEN RAJAPINNAN OHJELMISTOKOODI

```

from tkinter import *
import json
import paho.mqtt.client as mqtt

# Callback funktio yhteyden mahdollista menettämistä varten
def on_connect(client, userdata, flags, rc):
    print(f"Yhdistetty MQTT välittäjään palautusarvolla {str(rc)}")

# Kun aiheeseen julkaistaan uutta dataa, päivitetään monitori
def on_message(client, userdata, msg):
    print(msg.topic + ": " + msg.payload.decode())
    info = json.loads(msg.payload)
    location = info["sijainti"]
    temp = info["lampotila"]
    temp_str = str(temp) + " " + chr(176) + "C"
    humidity = info["ilmankosteus"]
    humidity_str = str(humidity) + "%"
    if location == "keittio":
        l_t_k.configure(text = temp_str)
        l_t_k.update()
        l_h_k.configure(text=humidity_str)
        l_h_k.update()
    elif location == "olohuone":
        l_t_o.configure(text = temp_str)
        l_t_o.update()
        l_h_o.configure(text=humidity_str)
        l_h_o.update()
    elif location == "tyuhuone":
        l_t_t.configure(text = temp_str)
        l_t_t.update()
        l_h_t.configure(text=humidity_str)
        l_h_t.update()

if __name__ == "__main__":
    # MQTT välittäjän osoite
    mqtt_broker = "test.mosquitto.org"

    # Luodaan simppele monitorointi UI lämpötilanvalvontaa varten
    window = Tk()
    window.title("Lämpötilanvalvontajärjestelmä")
    window.geometry("680x280+680+280")
    window.resizable(0,0)
    window.config(bg = "#2C3C3F")
    header_frame = Frame(window, bg = "#2C3C3F")
    body_frame=Frame(window, bg = "#2C3C3F")
    header_frame.pack(pady = 15)
    body_frame.pack()

    # Otsikko
    header_label = Label(
        header_frame,
        text = "Lämpötilanvalvontajärjestelmä",
        font = ("consolas", "24", "bold"),

```

```
        bg = "#2C3C3F",
        fg = "#CAF6FF"
    )
header_label.pack()

# Tunnisteet
l_loc = Label(
    body_frame,
    text="Anturin sijainti",
    font = ("consolas", "18"),
    bg = "#2C3C3F",
    fg = "#CAF6FF"
)

l_temp = Label(
    body_frame,
    text="Lämpötila",
    font=("consolas", "18"),
    bg="#2C3C3F",
    fg="#CAF6FF"
)

l_hum = Label(
    body_frame,
    text="Ilmankosteus",
    font=("consolas", "18"),
    bg="#2C3C3F",
    fg="#CAF6FF"
)

# Sijainnit
l_k = Label(
    body_frame,
    text="Keittiö",
    font=("consolas", "15"),
    bg="#2C3C3F",
    fg="#CAF6FF"
)

l_o = Label(
    body_frame,
    text="Olohuone",
    font=("consolas", "15"),
    bg="#2C3C3F",
    fg="#CAF6FF"
)

l_t = Label(
    body_frame,
    text="Työhuone",
    font=("consolas", "15"),
    bg="#2C3C3F",
    fg="#CAF6FF"
)

# Lämpötila-arvot
l_t_k = Label(
    body_frame,
    font=("radioland", "20"),
    bg = "#2C3C3F",
    fg = "#00D2FF"
```

```

)
l_t_o = Label(
    body_frame,
    font=("radioland", "20"),
    bg="#2C3C3F",
    fg="#00D2FF"
)
l_t_t = Label(
    body_frame,
    font=("radioland", "20"),
    bg="#2C3C3F",
    fg="#00D2FF"
)

# Ilmankosteus arvot
l_h_k = Label(
    body_frame,
    font=("radioland", "20"),
    bg="#2C3C3F",
    fg="#00D2FF"
)
l_h_o = Label(
    body_frame,
    font=("radioland", "20"),
    bg="#2C3C3F",
    fg="#00D2FF"
)
l_h_t = Label(
    body_frame,
    font=("radioland", "20"),
    bg="#2C3C3F",
    fg="#00D2FF"
)

# Asetetaan monitorin elementtien sijainnit kohdilleen
l_loc.grid(row=0, column=0, pady=5, padx=50)
l_temp.grid(row=0, column=1, pady=5, padx=10)
l_hum.grid(row=0, column=2, pady=5, padx=10)
l_k.grid(row=1, column=0, pady=5, padx=50)
l_o.grid(row=2, column=0, pady=5, padx=50)
l_t.grid(row=3, column=0, pady=5, padx=50)
l_t_k.grid(row=1, column=1, pady=5, padx=10)
l_t_o.grid(row=2, column=1, pady=5, padx=10)
l_t_t.grid(row=3, column=1, pady=5, padx=10)
l_h_k.grid(row=1, column=2, pady=5, padx=10)
l_h_o.grid(row=2, column=2, pady=5, padx=10)
l_h_t.grid(row=3, column=2, pady=5, padx=10)

# Luodaan MQTT client
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.connect(mqtt_broker, 1883, 60)
client.subscribe("keittio/info")
client.subscribe("olohuone/info")
client.subscribe("tyohuone/info")

# Aloitetaan ohjelma looppi

```



```
client.loop_start()  
window.mainloop()
```