Jere Willman

Petisensorin ja älykkään yhdyskäytävän yhteensopivuustestaus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

12.5.2016

Metropolia

Tekijä Otsikko Sivumäärä Aika	Jere Willman Petisensorin ja älykkään yhdyskäytävän yhteensopivuustestaus 44 sivua + 2 liitettä 12.5.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Verkot ja pilvipalvelut
Ohjaaja	Lehtori Harri Ahola

Insinöörityö toteutettiin osana laajaa projektikokonaisuutta, jonka tarkoituksena oli tutkia erilaisia mahdollisuuksia ja toteutustapoja terveysdatan hyödyntämiseen. Mukana projektissa Metropolia Ammattikorkeakoulun lisäksi ovat Aalto-yliopisto, Helsingin ja Uudenmaan Sairaanhoitopiiri, Murata, FiHTA (Finnish Healthtech Association) ja NurseBuddy.

Opinnäytetyössä oli tavoitteena tutkia asioiden internetin nykytilaa, toimintaa ja tulevaisuutta sekä älykästä yhdyskäytävää, joka mahdollisesti sopisi yhteen Murata SCA11H -sensorin kanssa. Pohjaksi valittiin Tampereen yliopiston WSN OpenAPI Gateway. Tavoitteena oli saada yhdyskäytävä ja petisensori toimimaan yhdessä niin, että niistä voitaisiin rakentaa kokonaisuus, jonka toimivuutta pystyttäisiin testaamaan suuremmassa mittakaavassa.

Työssä kävi ilmi, etteivät valitut yhdyskäytävä ja sensori toimineet yhdessä. Työssä lähdettiin tutkimaan, mistä tämä johtui ja olisiko ongelmaa mahdollista korjata. Tutkimuksissa kävi ilmi, että toimimattomuus johtui sensorin ja yhdyskäytävän erilaisesta toteutuksesta todennuksen osalta. Sensorin käyttämä datarakenne vaikutti oikealta, mutta sitä ei päästy kokeilemaan. Tämä johtui siitä, että todennuksen epäonnistuessa sensori ja yhdyskäytävä eivät kommunikoineet lainkaan keskenään.

Tutkimuksessa päätettiin rakentaa palvelinsovellus, jonka avulla sensorilta saatiin dataa talteen sensorin toiminnan tutkimista varten. Lopuksi päätettiin kokeilla, onko tutkimisen kohteena ollut yhdyskäytävä kuitenkin toimiva ja onko sille mahdollista saada toteutettua jokin yhteensopiva ratkaisu. Yhdyskäytävää varten tehtiin esimerkkisensorisovellus, joka todensi oikein ja lähetti yhdyskäytävälle dataa niin kuin oli tarkoitettu. Näin pystyttiin todentamaan, että yhdyskäytävä oli toimiva huolimatta siitä, ettei petisensori sen kanssa yhteen toiminutkaan.

Kaiken selvitetyn tiedon pohjalta voidaan todeta asioiden internetin olleen yksi suurimmista puheenaiheista teknologia-alalla viime vuosina. Siinä missä asioiden internetistä löytyvä tieto saattaa vielä pitkälti olla mainospuhetta, näyttää sen tulevaisuus kuitenkin valoisalta. Suuret kansainväliset yritykset ovat osoittaneet kiinnostuksensa asioiden internetin oletettua biljoonien dollarien markkinoita kohtaan, minkä valossa erilaisia asioiden internetiä hyödyntäviä toteutuksia varmasti nähdään tulevaisuudessa.

Avainsanat	IoT, asioiden internet, älykäs yhdyskäytävä, tietoverkot

Author Title Number of Pages Date	Jere Willman Compatibility testing of smart gateway and bed sensor 44 pages + 2 appendices 12 May 2016
Degree	Bachelor of Engineering
Degree Programme	Information technology
Specialisation option	Networking and cloud computing
Instructor	Harri Ahola, Senior Lecturer

This thesis was part of wide project entity where the main purpose was to research different possibilities and methods to implement good use of health care data. In addition to Metropolia University of Applied Sciences the project involved Aalto University, The Hospital District of Helsinki and Uusimaa, Murata, Finnish Healthtech Association and NurseBuddy.

The main purpose of this thesis was to investigate the current situation in the area of internet of things and to test if it would be possible to get Murata SCA11H bed sensor to work with WSN OpenAPI Gateway. The aim was to get the sensor to communicate with the gateway to see if they would make a match to be tested in a wider scale.

During the research it was noted that the selected gateway and sensor were not working together. The focus was on finding out the reasons for this. The research showed that the incompatibility between the gateway and sensor was caused by the difference in the main implementation of authentication. The general structure the sensor was using indicated it would work with the gateway but it could not be verified because of the authentication failure.

Because of the sensor not communicating with the selected gateway it was decided to create a simple server that could be used to capture sensor data. With the captured data it was possible to research how the sensor communicates outbound. The final phase was creating a simple sensor application so the gateway could be tested and compared to the way the bed sensor was designed. By creating a working sensor application that authenticated and communicated with the gateway it was checked that the gateway was really a working product even the sensor did not communicate with it.

After all the research an assumption can be made that the internet of things has been one of the biggest topics in the field of technology during the last few years. While it looks like the information available is mostly companies advertising their products, it can be said that the internet of things is here to stay. The biggest IT companies in the world have started to invest money to the field interested in the market of trillions of dollars. More and more internet of things products will be seen in the future.

Keywords	IoT, internet of things, gateway, network

Sisällys

Lyhenteet

API	Application Programming Interface, ohjelmointirajapinta, jonka avulla eri sovellukset voivat keskustella ja vaihtaa tietoja keskenään
ARM	Advanced RISC Machines, pienikokoisten laitteiden käyttöön soveltuva prosessoriarkkitehtuuri
D2D	Device to device, kommunikointi laitteelta laitteelle
D2S	Device to server, kommunikointi laitteelta palvelimelle
NAT	Network Address Translation, tekniikka, jonka avulla IP-osoitteita muutetaan reaaliajassa
S2S	Server to server, palvelinten välinen kommunikointi
SSL	Secure socket layer, protokolla verkkoliikenteen salaukseen
VPN	Virtual private network, teknologia, jonka avulla luodaan virtuaalinen virtuaalinen lähiverkko
Wi-Fi	Langaton verkkoteknologia, yleisesti käytössä
WSN	Wireless Sensor Network, langaton anturiverkko

1 Johdanto

Insinöörityö on osa Metropolia Ammattikorkeakoulun laajaa projektikokonaisuutta, johon on saatu teknologiateollisuuden myöntämää apurahaa. Mukana projektissa ovat myös Aalto-yliopisto, Helsingin ja Uudenmaan Sairaanhoitopiiri, Murata, FiHTA (Finnish Healthtech Association) ja NurseBuddy. Erilaisten ratkaisujen ja osa-alueiden tutkimiseksi tehdään useita opinnäytetöitä, yhteisenä tavoitteena selvittää erilaisten toteutusten mahdollisuuksia ja toteutustapoja terveysdatan hyödyntämiseen. Tässä opinnäytetyössä tavoitteena on toteuttaa sairaalakäyttöön soveltuva testitoteutus hyödyntäen Muratan petisensorin tarjoamaa sensoriteknologiaa ja verkon älyä yhdyskäytävän muodossa. Opinnäyteyössä huomio kiinnittyy suurelta osin valitun IoT-yhdyskäytävän toimintaan ja petisensorin lähettämään dataan, tavoitteena saada ne toimimaan yhdessä.

Opinnäytetyössä tutustutaan valitun yhdyskäytävän lisäksi myös kaiken internetin eli IoT:n tai IoE:n toimintaan ja sen tuomiin haasteisiin ja mahdollisuuksiin. Työssä selvitetään asioiden internetin keskeisimpiä käsitteitä ja aiheeseen läheisesti liittyviä teknologioita, kuten tiedonsiirtoprotokollia. Tarkoituksena on myös tutkia asioiden internetin tämän hetken tilaa, niin kuluttajamarkkinoilla kuin muillakin teollisilla sektoreilla.

Opinnäytetyössä toteutetaan tutkimus siitä, olisiko Tampereen yliopiston julkaisemaa OpenAPI WSN Gatewayta mahdollista saada toimimaan yhdessä Muratan valmistaman SCA11H-petisensorin kanssa. Mahdollisten yhteensopivuusongelmien ilmetessä kokeillaan vaihtoehtoisesti jonkinasteista esimerkkiratkaisua, jonka avulla voidaan kokeilla ja todentaa, onko OpenAPI WSN Gatewayta mahdollista saada toimimaan jonkin sensorin kanssa. Samalla voidaan tutkia yhdyskäytävän ominaisuuksia ja mahdollisuuksia. Opinnäytetyössä paneudutaan myös OpenAPI WSN Gatewayn ja SCA11H-sensorin toimintaperiaatteisiin.

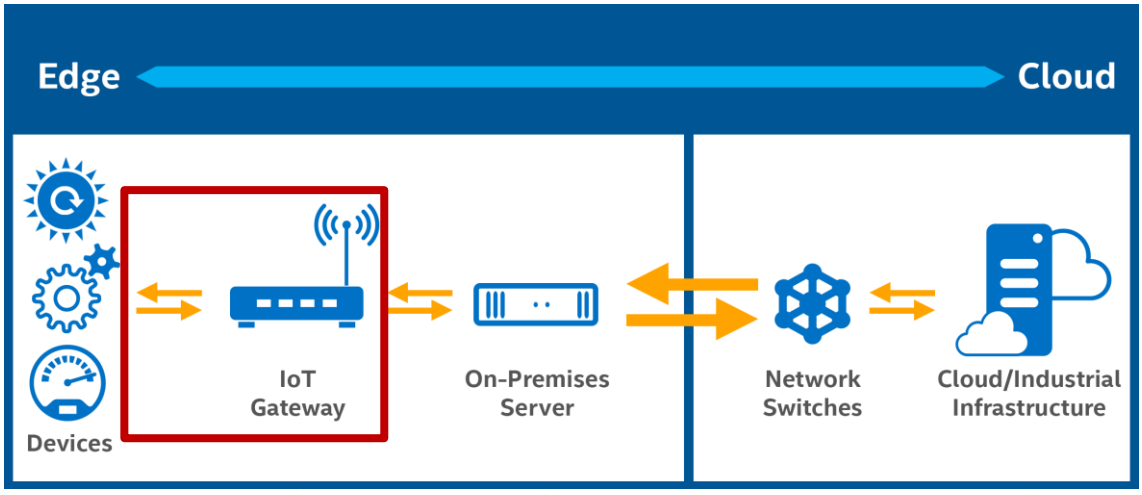
Ensimmäisessä käsittelyluvussa pohditaan asioiden internetin tämänhetkistä tilannetta ja sen mukanaan tuomia elämää helpottavia tuotteita sekä katsastetaan hieman tulevaan. Luvussa 3 esitellään asioiden internetin teknistä rakennetta eli yhdistämistä verkkoon, datan analysoimista, protokollia ja tietoturva-aspektia. Luku 4 avaa opinnäytetyön käytännön tutkimuksen lähtökohtia eli SCA11H-sensoria ja OpenAPI WSN Gatewayn asennusta. Luvussa 5 esitellään tarkemmin tarkastelun kohteena olleen tutkimuksen käytännön toteutusta.

2 Asioiden internet käytännössä

2.1 Yleistä

Tarkka englanninkielinen kuvaus asioiden internetille kuuluu seuraavasti: "A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual “things” have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network" (Friess & Vermesan 2016: 15). Tämän määritelmän mukaisesti termi on ymmärretty ja sen mukaisesti sitä käsitelty on tässä työssä.

Asioiden internet koostuu useista osa-alueista, sensoreista ja verkkoinfrastruktuurista taustajärjestelmiin. Opinnäytetyön käytännön toteutus painottuu kuvan 1 punaisella laatikolla merkittyyn alueeseen eli sensorin ja älykkään yhdyskäytävän väliseen liikennöintiin ja valitun yhdyskäytävän, OpenAPI WSN Gatewayn, toimintaan. Koska sensorin sisäistä toimintaa ei voida tutkia, pyritään tutustumaan sen tuottamaan verkkoliikenteeseen ja toimintaperiaatteisiin ulkoa käsin, jotta se saataisiin toimimaan valitun yhdyskäytävän kanssa yhdessä. Myös muita osa-alueita tarkastellaan, jotta kokonaiskuva asioiden internetistä ja sen ongelmista ja mahdollisuuksista muodostuisi.

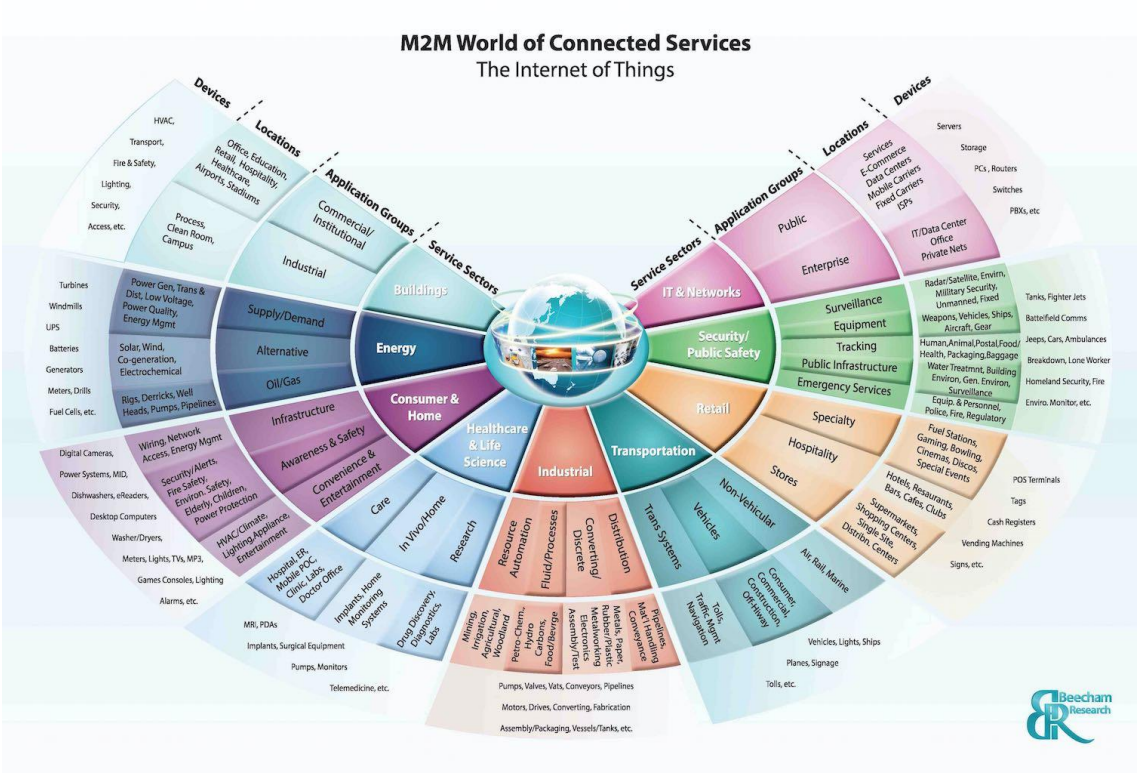


Kuva 1. Intelin suunnittelema havainnekuva verkon eri osa-alueista (End-to-End Security for Industrial Automation 2016).

Älykkäällä IoT-yhdyskäytävällä tarkoitetaan asioiden internetiä varten suunniteltua sovellusta tai laitetta, johon ”asiat” tai sensorit ottavat yhteyden. Yhdyskäytävää tarvitaan,

sillä laitteet ja sensorit yhdistäessään pääpalvelimelleen tuottavat paljon dataa ja verkkoliikennettä. Sensoreiden ja päätelaitteiden määrän kasvaessa kaiken datan käsitteleminen sellaisenaan ylikuormittaisi niin verkkoinfrastruktuurin kuin palvelimen suorituskyvyn. Älykäs yhdyskäytävä voi mahdollisesti tehdä viisaita päätöksiä sen suhteen, mitä hautaan välittää kohti pilveä (cloud) kuvassa 1 oikeassa laidassa.

Asioiden internet voidaan jakaa yhdeksään palvelusektoriin, joiden avulla eri toteutuksia ja kokonaisuuksia voidaan luokitella. Kuva 2 mallintaa ja avaa sektorit ja niiden sisältämät osa-alueet. Sektoreiden avulla voidaan hahmotella pirstaleinen ja tuore tietotekniikan ala helposti ymmärrettävään muotoon ja kokonaisuuksiin. Opinnäytetyö käsittelee terveydenhuollon ja terveystieteen sektoria, tarkemmin sairaalakäyttöön suunnitellun petisensorin toimintaa ja yhdistämistä älykkääseen yhdyskäytävään.

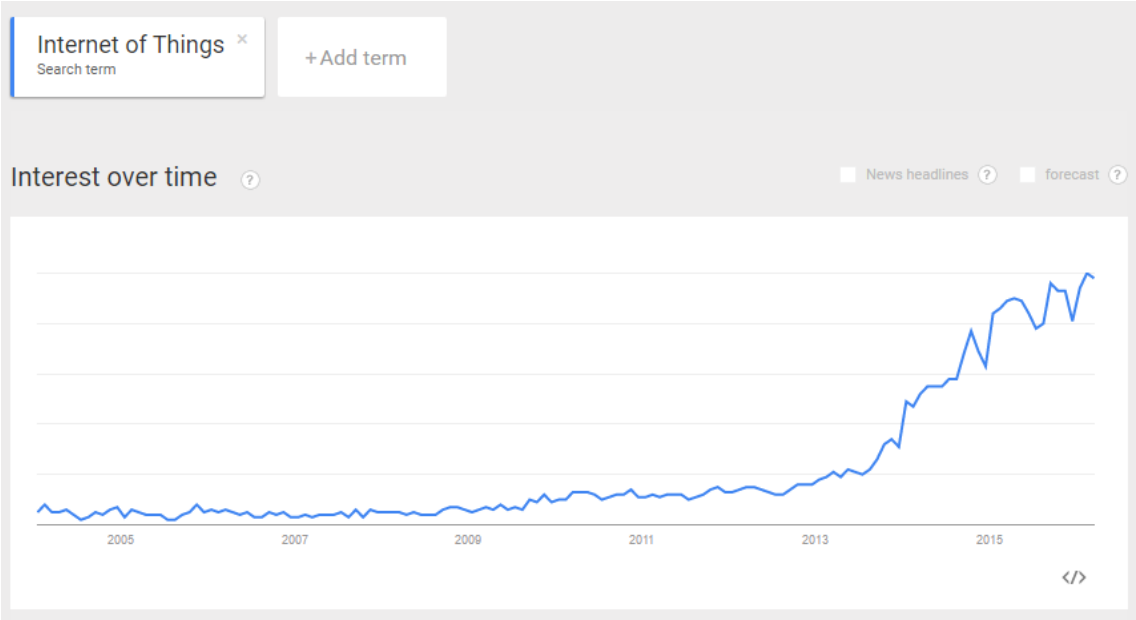


Kuva 2. Asioiden internetin eri sektorit (M2M/IoT Sector Map 2016).

Terveydenhuollon ja terveystieteen sektori pitää sisällään muun muassa terveydenhuollon ja sairaaloiden toimintaa helpottavat ratkaisut esimerkiksi potilaiden terveydentilan seurannan ja leikkaussalien toiminnan parantamiseksi. Loppukäyttäjien tai potilaiden ko-

tona ratkaisut voisivat olla esimerkiksi älykkäitä sydämentahdistimia tai muita terveydentilaa seuraavia järjestelmiä. Tutkimuksissa voitaisiin tutkia lääkkeiden jälkivaikutuksia tai laboratorioiden laitteiden toimintaa.

Asioiden internet on vuoden 2013 jälkeen alkanut nousta ihmisten tietoisuuteen. Kuten kuvasta 3 voidaan havaita, termi ”Internet of Things” on ollut käytössä jo vuonna 2004, mutta hakuvolyymit ovat vasta viime vuosina alkaneet maailmanlaajuisesti kasvaa. Tietoisuuden kasvuun on luultavasti vaikuttanut uusien laitteiden julkaisu, asioiden internetin yleistyminen ja uudet innovaatiot kuluttajamarkkinoille, kuten älykellot.



Kuva 3. Google Trends -kuvaaja hakusanalla ”Internet of Things” aikavälillä 1/2004 – 3/2016 (Google Trends 2016).

Asioiden internet yhdistää älykkäät laitteet toisiinsa ja internetiin. Sensorit ovat yleensä osa suurempaa kokonaisuutta, jossa sensorit on integroitu valmiiksi johonkin toteutukseen, esimerkiksi kelloon. Tästä esimerkkinä on älykello Pebble Time, johon sen myyntiin tullessa oli integroitu liikesensoreita, jotka vasta myöhemmin otettiin ohjelmistopäivityksen myötä käyttöön muun muassa ihmisen liikkuvuuden ja yleisen kunnon seuraamiseen (Kristoffer 2015). Esimerkiksi tällä tavalla sensorit voivat skannata dataa ja toimittaa sen johonkin ihmisille hyödyllisiin tarkoituksiin, kuten internetsivulle tai tässä tapauksessa matkapuhelinsovellukseen. Erilaisilla sensoreilla voidaan haluta valvoa myös esimerkiksi tehtaiden tai niissä toimivien koneiden tilaa.

Mobiili- ja langattomat verkot ovat nykymaailmassa yleistyneet ja muodostuneet lähes varmaksi tavaksi saada internetyhteys sijainnista riippumatta. Tämän vuoksi on voitu kehitellä sensoreita ja laitteita, jotka toimivat ennen saavuttamattomissa paikoissa, kuten syrjäseuduilla. Asioiden internetin yleistymiseen on vaikuttanut tekniikan kehittyminen ja sen mukana sensorien ja verkkokorttien hintojen laskeminen. Ennen pienten sensorien verkkoon saattaminen olisi ollut paljon kalliimpaa kuin nykyään – parhaimmillaan sensorien hinnat ovat laskeneet jopa sadasosaan 2000-luvun alusta. (Lesser 2015.) Yhä useampiin laitteisiin tullaan sijoittamaan jonkinlainen tietokone, sillä erilaisten ARM-ratkaisujen (Acorn RISC Machine) hinnat laskevat. Muutaman vuoden kuluttua alle eurolla saa riittävän tehokkaan tietokoneen, jonka voi integroida melkein mihin vain. Tällä hetkellä yksi halvimmista koneista on Raspberry zero, jonka hinta on viiden dollarin molemmin puolin.

Asioiden internetin toteutuksia tehdessä voidaan käyttää hyväksi laitteiden keskinäistä kommunikointia (D2D eli device to device), joka on luonut asioiden internetille uusia mahdollisuuksia ja haasteita. Kerätty tieto tulee toimittaa laitteelta palvelimelle (D2S eli device to server) ja palvelinten tulee osata välittää tietoa palvelimelta toiselle (S2S eli server to server). Näiden uusien ideoiden kehittyessä asioiden internetistä on pikkuhiljaa tullut menestyvä ja kannattava palvelu. (Schneider 2013.) Tätä esitellään myös kuvassa 5 sivulla .

Asioiden internetin tulevaisuuden voidaan kuvailla näyttävän valoisalta siinä mielessä, että isot yritykset ovat alkaneet panostaa ja sijoittaa sekä aikaa että rahaa siihen. Esimerkiksi Business Insider uutisoi helmikuussa 2016, että Cisco ostaa Jasper Technologies -nimisen yrityksen 1,4 miljardilla dollarilla. Jasper Technologies tekee pilvipalveluihin perustuvia IoT-toteutuksia. (Bort 2016.) Uusia laitteita ja ideoita voidaan sanoa tulevan jatkuvasti lisää, ja niillä yritetään nopeasti tehdä rahaa.

Sensorien hinnat laskevat ja koot pienenevät teknologian kehittyessä, joten lisääntyvässä määrin tulevaisuudessa sensoreita laitetaan jopa ihmisiin ja lemmikkeihin. Esimerkiksi Suomessa koiriin voidaan jo nykyisin sijoittaa mikrosiru, jonka avulla esimerkiksi eläinlääkäri voi tunnistaa löytyneen koiran (Mikrosiru 2016). Myös erilaiset elämän kannalta hyödylliset sensorit, kuten Googlen älypiilolinssiprojektissa käyttämä glukoosisensori, alkavat pikkuhiljaa yleistyä. Googlen projektissa piilolinssien avulla pystyttäisiin seuraamaan esimerkiksi kyyneleiden glukoosiarvoja ja näin analysoimaan diabetesta sai-

rastavan ihmisen tilaa. Googlen piilolinssin sisään on integroitu sensori ja langaton lähetin, jotka pystyvät tuottamaan ja raportoimaan lukemia kerran sekunnissa eli lähes reaaliajassa. (Otis & Parviz 2014.)

Googlen piilolinssien lisäksi maailmalla on jo alettu istuttaa sensoreita ihmisiin. Esimerkiksi Ruotsissa Epicenter-niminen yritys on alkanut laittaa RFID-siruja vapaaehtoisten työntekijöidensä käsiin, jolloin työntekijät voivat kulkea ja tulostaa työpaikalla ilman kulkulupia. Siru toimii työntekijöiden kulkutunnisteena. Lisäksi työntekijät voivat maksaa lounaansa henkilökohtaisilla siruillaan, käteisen tai korttimaksujen sijaan. (Mearian 2015.) Kuten Epicenterin tapauksessa, voi ihmisten siruttamisen ideana siis olla se, että ihminen pyrittäisiin varmasti tunnistamaan. Tämä voisi olla tärkeää esimerkiksi maissa, joissa kaappauksia tapahtuu paljon, sillä ihminen voitaisiin sirun avulla kiistatta tunnistaa. Suomessa alettiin vuodesta 2009 eteenpäin myönnettyihin passeihin istuttaa älysiruja ja integroida niihin muun muassa haltijan sormenjälkiä turvallisuuden parantamiseksi (Fingerprints to be included in new passports as from 29 June 2009).

Tulevaisuudessa voidaan olettaa, että maksaminen esimerkiksi matkapuhelimilla tulee yleistymään, sillä maailman suurimmat matkapuhelinyritykset, kuten Samsung, Apple ja Google, ovat kukin alkaneet panostaa omiin mobiilimaksupalveluihinsa Apple Payhin, Samsung Payhin ja Google Payhin (Hristov 2016). Kun tulevaisuudessa suurimmassa osassa puhelimista on maksuominaisuus ja monet kaupat hyväksyvät sen maksutapanaan, sen käyttö myös todennäköisesti yleistyy. Lähimaksaminen on muutenkin yleistynyt pikkuhiljaa myös Suomessa vuodesta 2013 alkaen. Lähimaksamisen ideana on se, että asiakas voi hoitaa pienten, eli alle 25 euron arvoisten ostosten maksamisen ilman, että hänen tarvitsee allekirjoittaa kuitteja tai painaa tunnuslukuja (Lähimaksaminen 2016).

Accenturen tutkimuksen mukaan vuonna 2014 87 % valtaväestöstä ei ymmärtänyt asioiden internetiä terminä eikä sen tulevaisuutta tai arvoa (The Internet of Things: The Future of Consumer Adoption 2016: 5). Valtaväestön tietoisuuden puutteesta huolimatta suuret yritykset ovat huomanneet sen tuoman markkinaraon. Verkkojätti Ciscon arvioiden mukaan vuoteen 2020 mennessä maailmassa on 50–200 miljardia verkkoon yhdistettyä laitetta. Intelin mukaan verkkoon kytkettyjä laitteita tulee olemaan yli 200 miljardia. Kaikkien näiden miljardien laitteiden tulee olla optimoitu verkkoyhteydelle, niin sovelluksen kuin laitteiston tasolla. Autoistakin yli 90 prosenttia on yhteydessä verkkoon vuonna 2020, väittää espanjalainen verkkoyhtiö Telefonica. ”Industrial Internet” tulee olemaan

suurempi kuin Kiinan tämän hetken talous, tulevaisuudessa tehtaiden ja muiden älyllä ohjattavat laitteet ovat noin 10–15 biljoonan dollarin markkinat. Laitteiden suuri määrä aiheuttaa haasteita niin verkkolaitteistolle kuin laitteita tukeville palvelimillekin. Pilven ansiosta tehtaiden automaatio kasvaa ja tehtävien tehokkuus nousee. (Sun 2016.) Kaikesta tästä voidaan päätellä, että asioiden internet ei ole ohimenevä tapaus.

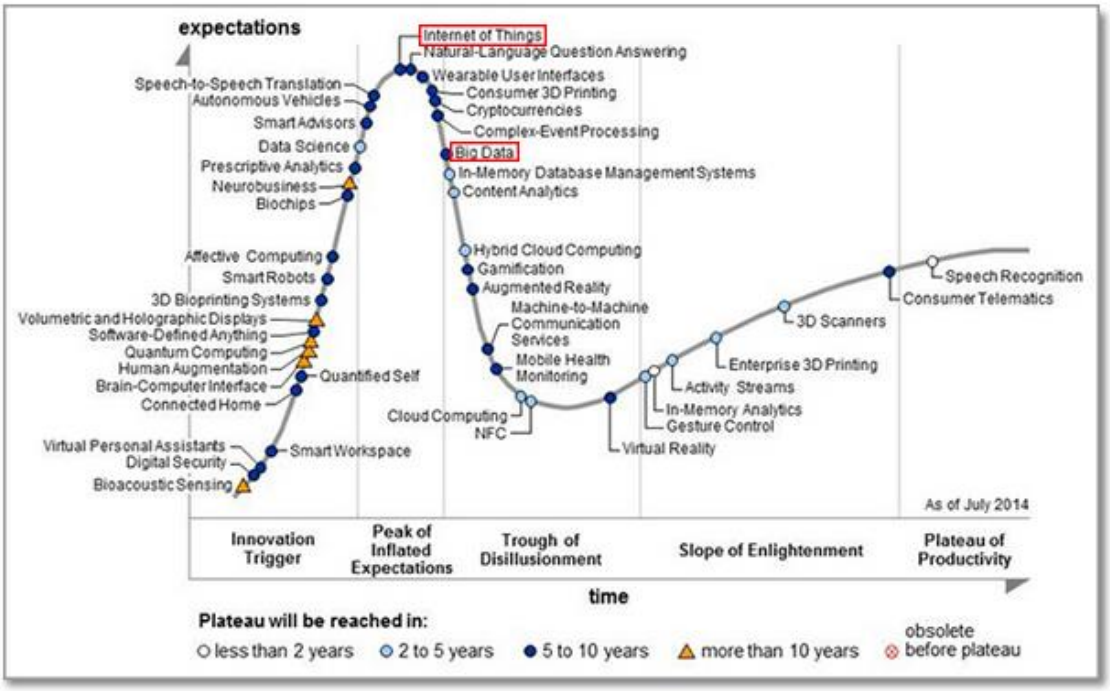
Asioiden internetin ja muutenkin verkossa olevien laitteiden määrän kasvaessa tulee IPv4-osoitteista pula. IPv4-osoitteita on vain reilu neljä miljardia, kun IPv6-osoitteita on noin 340 sekstiljoonaa (Parkhurts 2004). Matkapuhelimia yksin myydään jo yli miljardi vuodessa, ja niistä suuri osa on yhteydessä verkkoon. Kun tähän lisätään asioiden internetin tuomat muut laitteet, jotka haluavat myös kommunikoida verkkoon, on tulevaisuudessa laitteiden määrä ja niiden tarvitsemien IP-osoitteiden määrä erittäin suuri (Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013 2014). Luvuista voidaan havaita, että osoitteet loppuvat kesken, vaikka NAT eli osoitteenmuunnos olisi käytössä mahdollisimman usein. Kommunikointi kumpaankin suuntaan hankaloituu IPv4:ää käytettäessä, mikäli NAT on välissä muuttamassa osoitteita. Tästä syystä IPv6 tai ainakin tuki sille olisi hyvä saada käyttöön mahdollisimman pian, sillä sen mukana osoitteita tulee niin paljon, ettei osoitteenmuutosta välttämättä tarvittaisi enää ja kaikki halutut laitteet voisivat olla saavutettavissa jouhevasti mistä vain ja milloin vain ilman porttiohjauksia. Myös hyvin toimivat verkkoyhteydet ovat tärkeä osa kokonaisuuksia, jotta tietoa saadaan tallennettua sensoreiden ja laitteiden välimuistien lisäksi myös pysyvästi tietokantoihin.

2.2 Asioiden internet ja big data

Asioiden internet on big datan kanssa ollut viime vuosien suurimpia puheenaiheita teknologiapiireissä. big datalla tarkoitetaan massiivisen datamäärän keräämistä ja sen louhimista. Koska tallennustila on nykyään suhteellisen halpaa ja sen hinta edelleen laskee, halutaan usein kerätä kaikki mahdollinen data, mitä vain suinkin on mahdollista kerätä. Kaiken datan tallentaminen kuitenkin johtaa siihen, että tarpeellisen tiedon sekaan eksyy suunnaton määrä turhaa dataa. Tietoa kerääntyy suuria määriä esimerkiksi sensoreista, kun niitä asennetaan ympäri maailmaa seuraamaan ihmisten ja asioiden toimintaa ja kun ne lähettävät keräämiään tietoja verkkoon. Suuri osa datasta koostuu muun muassa aika- ja paikkatiedosta, sillä esimerkiksi puhelinten käyttäjät usein hyväksyvät puhelimen paikannuspalvelut ja jättävät paikannuksen päälle. Paikannuspalvelujen ollessa päällä

raportoi puhelin sijaintitietoa esimerkiksi Androidin tapauksessa Googlelle, ja näin paikkatietoa kerääntyy suuria määriä.

Gartnerin hypekäyrän avulla pystytään ennustamaan teknologian kehitystä ja uusien innovaatioiden tulevaisuutta. Erilaisilla merkeillä kuvataan sitä aikaa, joka teknologian käyttöönoton normalisoitumisessa kestää. Käyrä itsessään kertoo, kuinka suurta keskustelua teknologia kerää. Kuvasta 4 voidaan päätellä, että Internet of Things on ollut syksyllä 2014 yksi keskustelluimmista aiheista teknologia-alalla, ja sen pallosta voidaan päätellä, että se tulee olemaan tavanomainen asia noin viiden – kymmenen vuoden kuluttua eli 2020-luvun taitteessa. Big data on jo ohittanut suurimman hypekynnyksensä, ja se tullaan lähiaikoina lähes unohtamaan, kunnes se vaivihkaa muuttuu osaksi ihmisten tavallista elämää.



Kuva 4. Gartnerin hypesykli on todettu realistiseksi ja toimivaksi tavaksi ennustaa tulevaisuuden tietoteknisiä suuntauksia (McLellan 2015).

Yksi big datan suurimmista ongelmista piilee siinä, miten massiivisista tietomääristä saadaan louhittua ja karsittua vain tarpeellinen ja käyttökelpoinen tieto. Tiedon löytämisen lisäksi ongelmana on, että käyttökelpoinen data vanhenee usein nopeasti, jolloin tärkeän tiedon löytämisen tulee myös onnistua nopeasti. Tänään tärkeä data saattaa olla jo ensi

viikolla ”turhaa” tai vanhentunutta. Tilastollisesti tosin olisi hyvä pitää tallessa kaikki mahdollinen tai vähänkin tärkeä sensoridata, jolloin voidaan jälkikäteen tehdä tilastollista tutkimusta ja analysoida, kuinka maailma on kehittynyt ja kuinka trendit ovat muuttuneet tiedon tallennuksen aikana. (McLellan 2015.) Suurten datamäärien analysoimisessa on onnistuttu muun muassa aikaisemmin mainitussa Google Maps -karttapalvelussa, jonka avulla pystytään ennakoimaan, kuinka liikennevirrat kehittyvät ja käyttäjät voivat suunnitella matkansa välttäen ruuhka-aikoja ja tukoskohtia.

2.3 Käytännön toteutuksia

Joissain kaupungeissa on jo onnistuttu tekemään pysäköintipaikkojen suhteen niin, että asukas voi kotoa käsin varata paikan jostain päin kaupunkia ja järjestelmä pitää huolen, että paikka löytyy. On havaittu, että vaikka etukäteen varattavat älypysäköintipaikat olisivat kalliimpia käyttää, ihmiset maksavat niistä mielellään esimerkiksi huolettomuuden ja aikataulun helpottumisen vuoksi. Kaupunki on myös kerännyt suuremmat pysäköintitulot kuin ennen, nimenomaan älypysäköintiruutujen ansiosta. Lisäksi älypysäköintiruutujen ansioista kaupunki säästää hurjat summat, kun pysäköinninvalvojia ei tarvitse palkata seuraamaan autojen pysäköintimaksuja ja aikoja.

Joissain pysäköintihalleissa on sijoitettu pysäköintiruutujen yläpuolelle sensorit, joiden avulla tunnistetaan vapaat pysäköintiruudut. Sensoreiden avulla voidaan automaattisesti laskea esimerkiksi kuinka monta vapaata pysäköintiruutua hallissa on. Lisäksi sensorit voivat laskea auton pysäköintiajan ja näin ilmoittaa, mikäli auto on liian kauan parkissa. Lisäksi sensorin yhteyteen on mahdollista sijoittaa esimerkiksi vihreä valo ilmoittamaan vapaasta tai punainen valo varatusta pysäköintiruudusta. Näin asiakkaat voivat kauempaa havaita vapaat pysäköintiruudut: näin on toimittu esimerkiksi Kauppakeskus Sellossa Espoossa, jonne on asennettu Siemensin SIPARK-järjestelmä. Pysäköintihallin ulkopuolelle voidaan lisäksi sijoittaa numeronäyttö, joka ilmoittaa, paljonko vapaita paikkoja vielä on jäljellä, ja jopa verkkosivuilla saatetaan kertoa, kannattaako halliin tulla autolla. Pysäköintihallin sisällä älykkäät näytöt näyttävät nuolilla, missä suunnassa ja paljonko pysäköintipaikkoja on vapaana, ja näin voidaan ohjata liikennevirtoja vapaiden pysäköintipaikkojen suuntaan.

Asioiden internetin tuomat mahdollisuudet vaikuttavat liikenteen lisäksi erityisesti teollisuuteen ja sairaaloihin. Sairaalat voivat hyötyä erilaisten sensorien käytöstä erityisesti

potilaita seuratessa ilman, että ihmisen tarvitsee erikseen käydä tarkastamassa esimerkiksi sykettä. Teollisuuden laitteiden toimintaa voidaan seurata ilman ihmisen läsnäoloa ja niiden toimintaa raportoida keskitetysti. Näin laitteiden toimintaa voidaan optimoida ja jopa ennakoida huoltoja tai tulevia vikoja. Suuren arvon teollisuudelle asioiden internet tuokin muun muassa automaation kautta, jolloin usein tuotannon kallein komponentti eli ihminen voidaan irtisanoa.

Sairaaloissa laitteita voidaan käyttää potilaiden kunnon seuraamiseen ja esimerkiksi peti- tai lattia-anturilla voidaan seurata potilaiden unta ja toimintaa. Lattiasensorilla voidaan huomata, mikäli potilas kaatuu tai muuta poikkeavaa tapahtuu, esimerkiksi että potilas karkaa huoneesta. Sensoreiden lisäksi on helppo lisätä esimerkiksi hätäpainikkeita, jos potilaalle tulee akuutti tarve saada paikalle sairaanhoitaja. (Virtual Patient Observation: Centralize Monitoring of High-Risk Patients with Video 2016.) Sairaaloissa nämä ovat arvokkaita toteutuksia, joiden avulla voidaan hälyttää apua paikalle, vaikkei apua tarvitseva sitä kykenisi itse tilaamaan.

Syyskuussa 2015 Espanjassa sateet aiheuttivat suuria tulvia ja useita ihmisiä jopa kuollut vesien tulviessa kaduille ja tulvien huuhtoessaan mukanaan jopa autoja (The rain in Spain claims four lives and causes chaos across south coast 2015). Espanjan tunnettujen sääilmiöiden vuoksi siellä on suunniteltu järjestelmiä, jotka ovat alkaneet helpottaa paikallisten elämää ääriolosuhteissa. Barcelonassa esimerkiksi on onnistuttu virtaussensoreiden avulla hallitsemaan kaupungin vesiputkijärjestelmiä niin, että suurien vesimäärien tai tulvien tullessa toimitetaan vedet eri kautta kuin tavallisena aikana, jolloin veden virtaus on paljon pienempää. Näin yritetään välttää tulvia ja vesi saadaan juoksutettua nopeasti pois teiltä. Järjestelmä on toteutettu putkistoon asennetuilla virtausantureilla. (Barcelona City Council 2016.)

Vuosittainen teknologiatapahtuma CES 2016 (Consumer Electronics Show) tarjosi kuluttajille harppauksen tulevaisuuden sensoriteknologiaa ja asioiden internetiä silmälläpitäen, niin autojen kuin kauko-ohjattavien koptereiden osalta. Myös autot alkavat olla aina yhdistettynä internetiin. Monet autot jopa asentavat itse ohjelmistopäivityksensä, pitävät huolen järjestelmistään ja varaavat myös mahdollisesti itse huoltonsa, kun niiden aika esimerkiksi kilometrien osalta koittaa. Myös vuoden trendi vaikuttaisi olevan jo edellisvuosina alkanut puheentunnistus ja siihen reagoiminen. Kymmenet yritykset ilmoittivat CES 2016 -tapahtumassa alkavansa tukea Amazon Echoa, joka on puhetta ymmärtävä ja puheella toimiva laite. (Higginbotham 2016.)

Autoihin integroitujen älykkäiden sensorien ja sovelluksien avulla ihmiset voivat hallita kotivaloja ja lämmitystä esimerkiksi autosta puheentunnistusta hyväksikäyttäen. On siis käynyt ilmi, että asioiden internetissä ei ole kyse vain itse laitteista vaan sensorit, laitteet ja sovellukset tuottavat yhdessä palvelun, josta ihmiset ovat valmiita maksamaan. Esimerkiksi älyilmastointi ei ole kovin houkuttelevia sellaisenaan, mutta kun se saadaan integroitua esimerkiksi Googlen omistamaan Nestiin ja näin keskitetysti hallittua kaikkia kodin älykkäitä laitteita, tulee kokonaisuuksista huomattavasti houkuttelevampia. Googlen nykyisin omistama Nest tekee kotiautomaatioon tarkoitettuja termostaatteja ja turvajärjestelmiä, jotka on ohjelmoitu älykkäiksi ja itsestään oppiviksi.

Koteihin tuotavasta älyteknologiasta ovat alkaneet kiinnostua myös kodinkoneyritykset: esimerkiksi Whirlpool on kertonut tekevänsä yhteistyötä Amazonin kanssa. Whirlpoolin tavoitteena on, että laitteet osaisivat tilata esimerkiksi pyykinpesuainetta sen loppuessa ilman, että ihmisen tarvitsee kantaa huolta pesuaineiden riittävyydestä. On siis havaittavissa, että tulevaisuudessa kuluttajille kohdistetut tuotteet ovat pelkkien monimutkaisten tuotteiden sijaan elämää helpottavia palveluita. (Higginbotham 2016.) Palveluita ja kokonaisuuksia suunnitellessa tulee kuitenkin miettiä, kuinka toteutukseen voidaan lisätä uusia ominaisuuksia ja kuinka se toimii muiden ratkaisujen kanssa.

3 Asioiden internet – tekninen toteutus

Asioiden internetissä laitteet halutaan yleensä saada yhdistettyä verkkoon. Tätä varten tarvitaan jonkinasteinen yhdyskäytävä, jonka kautta laite yhdistää määränpäähänsä. Asioiden internetin ja älykkäiden laitteiden yleistyessä ja datan määrän räjähtäessä enää ei välttämättä riitä tavallinen oletusreitti (default gateway) ja reititin, vaan halutaan ottaa käyttöön erikseen asioiden internetiä varten suunniteltu ja toteutettu yhdyskäytävä.

Tulevaisuudessa olisi tärkeää, että yhdyskäytävät tai jopa itse sensorit tekevät itse viisaita päätöksiä, kuten esimerkiksi pakkaavat ja puskuroivat tietoa. Jos esimerkiksi valvontakamera lähettää hallintapalvelimelle kuvaa jatkuvasti, myös silloin kun ei olisi mitään kuvattavaa, tulee paljon hukkaliikennettä ja -dataa. Jos kameroita on kymmeniä, satoja tai jopa tuhansia, verkkoliikenteen ja turhan datan määrä on valtaisa. Tällöin olisi tärkeää, että kamerat osaisivat itse tunnistaa esimerkiksi liikkeen ja välittää kuvaa vain silloin, kun se on valvonnan kannalta oleellista. Sama pätee myös sensoreihin. Esimerkiksi petisensoriin olisi hyvä kehittää sen verran älyä, että se osaa havaita, mikäli ihminen

poistuu sängystä. Tyhjän sängyn tuottaman datan välittäminen eteenpäin on turhaa, koska tällöin ei haluttua sykettä tai muutakaan tietoa ihmisestä saada talteen. Sensorin tuottama turha verkkoliikenne vain kuormittaa verkkolaitteita ja mahdollista yhdyskäytävää ja tuottaa turhaa dataa, joka pitää joka tapauksessa jotenkin analysoida. Muutamalla sensorilla ei vielä liikennettä tai dataa juurikaan tule, mutta mikäli sensoreita lisätään satoja tai tuhansia, alkaa dataa esimerkiksi sairaalan kokoisessa kompleksissa kerääntyä ja liikkua suuria määriä.

Jotta datasta saataisiin yhdyskäytävän avulla karsittua turhat tiedot ja mahdollisesti myös pakattua massiivista datan määrää, täytyy yhdyskäytävän olla tavallista oletusyhdyskäytävää järkevämpi. Myös mikäli yhdyskäytävän verkkoyhteys katkeaa, olisi mahdollista pitää siinä jonkinlaista puskuria tai välimuistia, jottei katkoksen aikana tuotettu ja kerätty data katoa. Järkevän yhdyskäytävän rakentaminen on myös hinnan osalta kannattavaa, koska yhden keskitetyn komponentin hinnan nousu ei välttämättä vaikuta kokonaiskustannuksiin paljoa.

3.1 Protokollat ja alustat

Sensoreita yhdistetään verkkoon eri tavoilla. Esimerkiksi älykellot yhdistävät itsensä yleensä matkapuhelimeen Bluetoothin avulla, ja puhelin hoitaa reitin internetiin yleensä 3G:n, 4G:n tai langattoman verkon avulla. Tämä tapa ei olisi mahdollinen tai järkevä esimerkiksi petisensorissa, koska sairaalassa sensoreita saattaisi olla satoja tai tuhansia. Koska Bluetoothia ei ole suunniteltu satojen tai edes kymmenien laitteiden samanaikaiseen kommunikointiin eikä pitkien etäisyyksien kantamiin, ei se välttämättä olisi toimiva ratkaisu. Tällöin jokin muu langaton lähiverkkoteknologia, kuten langaton verkko (WLAN) ja älykäs yhdyskäytävä, voisi mahdollisesti olla toimivampi ratkaisu. Langattoman verkon käyttöönottamisessa olisi myös se hyvä puoli, että esimerkiksi useimmissa sairaaloissa sen vaatima infrastruktuuri on jo rakennettu eikä suuria investointeja välttämättä verkkoyhteyden osalta tarvittaisi.

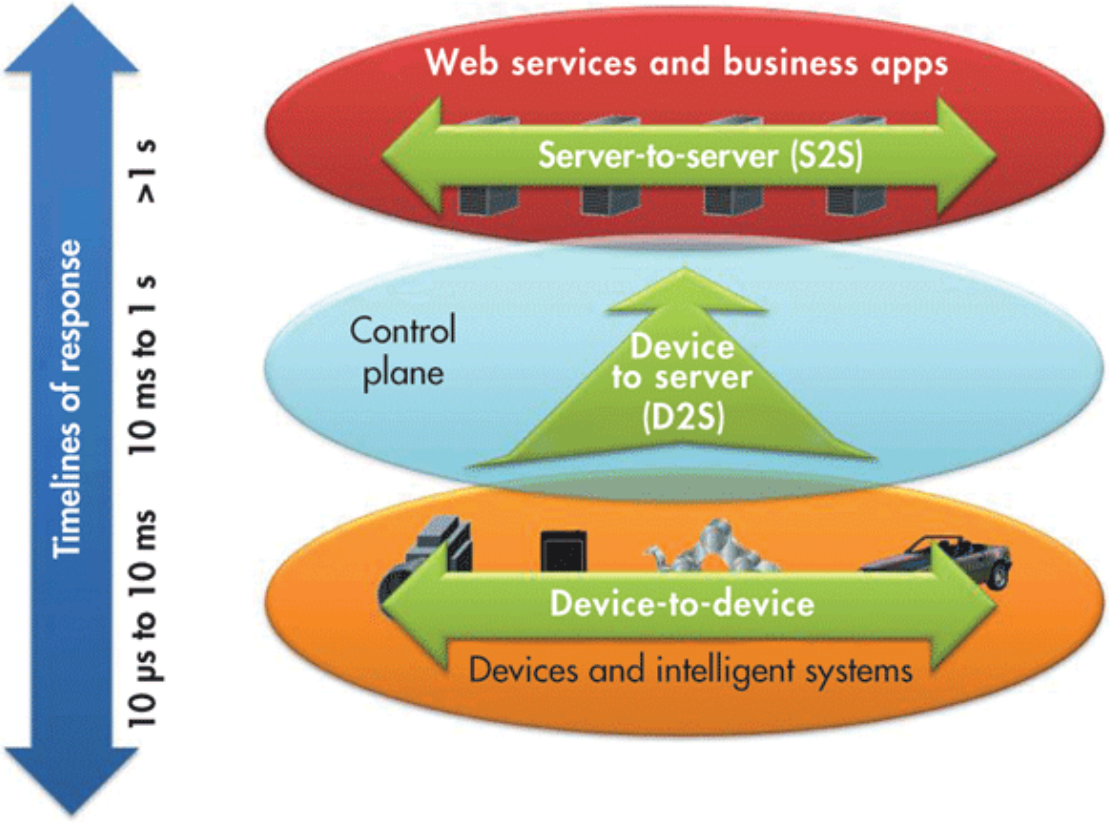
Mikäli halutaan tai on tarve käyttää muita teknologioita kuin tavallista langatonta verkkoa, se voidaan mahdollistaa yhdyskäytävällä. Älykkään yhdyskäytävän avulla voidaan sensorin kommunikointi hoitaa monilla erilaisilla tiedonsiirtotekniikoilla, jolloin sensori voi olla yhdistettynä internetiin esimerkiksi ZigBeellä eli eräällä langattomalla lähiverkkoteknologialla ja silti käyttää samaa yhdyskäytävää kuin langattomalla verkolla toimivat sensorit.

Riittää kun sensori toimittaa datansa yhdyskäytävän tukemalla datarakenteella. (Architecture 2016)

Erilaisia langattomia sensoriverkkoja (WSN) on monenlaisia, kuten esimerkiksi tähtiverkko, jossa kaikki sensorit kommunikoivat suoraan yhdyskäytävän kanssa. Lisäksi on muunlaisia verkkoja, joissa sensorit toimivat toisilleen tukiasemina ja näin saadaan sensoriverkko laajaksi ilman, että älykkäitä yhdyskäytäviä tarvitsee ripotella useampia. Sensorien tuottama data tulee sen jälkeen toimittaa ulkoiselle palvelimelle ja sieltä mahdollisesti seuraavalle palvelimelle. Palvelimen tulee toimittaa kerätty data mahdollisesti takaisin laitteelle, analyysiohjelmistolle tai ihmisille tutkittavaksi.

Laitteiden välinen kommunikointi toteutetaan yleensä jo olemassa olevilla protokollilla. Protokollat valitaan yleensä sen mukaan, kuinka reaaliaikaisesti dataa halutaan toimittaa. Yleisimpiä protokollia on MQTT, jota käytetään datan keräämiseen ja palvelimen kanssa kommunikointiin. Se on tarkoitettu laitteiden väliseen kommunikointiin (M2M), ja on suunniteltu olemaan mahdollisimman kevyt resurssien säästämisen vuoksi (MQTT 2016). Toinen on XMPP, joka on avoin protokolla suunniteltu ja laitteiden (tai ihmisten) yhdistämiseen palvelimeen (D2S). Alun perin XMPP on kehitetty Jabberin, avoimen lähdekoodin yhteisön, toimesta tarkoituksena olla vaihtoehtona aikansa suljetun lähdekoodin sovelluksille (An Overview of XMPP 2016). Lisäksi eräs tunnettu standardi on DDS (Data Distribution Service), jonka tarkoituksena on toimia nopeana, luotettavana ja skaalautuvana laitteiden välisenä (M2M) protokollana (What is DDS? 2016).

Laitteiden välinen kommunikointi voidaan jakaa kolmeen luokkaan: S2S, D2S ja D2D. Kuvan 5 havainnollistus selventää tätä. Kategorioittain voidaan olettaa, kuinka pitkiä viiveet ovat kommunikoinnissa, ja täten kokonaisuuksia suunnitellessa voidaan ottaa huomioon, minkälaisilla viiveillä järjestelmät voivat toimia. Kuvasta voi nähdä, kuinka laitteelta palvelimelle kommunikointi voi kestää jopa sekunnin, kun taas laitteiden välinen kommunikointi hoituu tavallisesti millisekunneissa.



Kuva 5. Havainnollistus siitä, kuinka eri asteilla laitteet kommunikoivat sekä niiden väliset oletetut viiveajat (Schneider 2013).

Asioiden internetiä varten on tarjolla myös erilaisia valmiiksi rakennettuja reitittimiä, jotka voivat toimia yhdyskäytävänä ja näin yksinkertaistavat verkon topologiaa ja alentavat mahdollisesti toteutuksien kokonaishintaa. Tällä hetkellä Intelillä on tarjolla neljä yhdyskäytäväsarjaa, jotka on toteutettu asioiden internetiä silmälläpitäen. Jokainen sarja on suunniteltu juuri tietynlaiseen tarpeeseen, jotta jokaiselle niitä tarvitsevalle löytyisi sopiva. Pienin sarja DK50 on tarkoitettu sovelluskehittäjille ja harrastelijoille, jotka kehittelevät IoT-sovelluksia. Mukaan tulee kuuden kuukauden mittainen järjestelmälisenssi, joka sisältää laajan kirjon Intelin omia ominaisuuksia. DK50-sarjaa ei saa käyttää tuotantoympäristössä. DK100-sarja on tarkoitettu teollisuuden käyttöön, ja se sisältää samantyyppiset ominaisuudet kuin DK50, mutta sitä saa käyttää tuotantoympäristössä eikä siinä ilmeisesti ole kuuden kuukauden mittaista lisenssiä. DK200 on tarkoitettu vähävirtaisiin toteutuksiin ja mukaan tulevat samat ominaisuudet kuin DK50- ja DK100-sarjoissa. DK100 ja DK200 sisältävät samanlaisen SoC X1020D -prosessorin. Viimeinen ja jykevin sarja on DK300, joka soveltuu suurimpiin ja monimutkaisimpiin toteutuksiin. Siinä

on Intel Atom -prosessori ja kaikki ominaisuudet, mitä Intel omiin yhdyskäytäviinsä tarjoaa. (Intel 2016.)

Viime vuosina myös Cisco on alkanut kiinnostua asioiden internetin mukanaan tuomista mahdollisuuksista. Verkkolaiteyhtiönä sillä on asioiden internetin suhteen valtavat mahdollisuudet ansaita, kun miljardeja uusia laitteita pitäisi saada yhdistettyä verkkoon. Aikaisemmin Cisco on hehkuttanut ”Fog Computingia” eli että datan analysointi saataisiin hoidettua mahdollisimman lähellä sen lähdettä, esimerkiksi älykkäässä yhdyskäytävässä. Ciscon mukaan pelkkä älykäs yhdyskäytävä ei enää riitä, vaan datan analysointi pitäisi saada siirrettyä vieläkin lähemmäs sensoria, mielellään integroitua sensoriin. Tätä kutsutaan termillä ”Mist Computing”. Sen tarkoituksena olisi, että data-analyysin tapahtuessa välittömästi voitaisiin myös toimia välittömästi ja halutut toimenpiteet tehdä ilman viivettä. Esimerkiksi jos kaupassa seurataan asiakkaita, voitaisiin asiakkaan saapuessa liikkeeseen välittömästi esittää tälle sopiva mainos. Datan tulisi edelleen toki kulkea myös pilveen tarkempaa analyysia varten, mutta tavoitteena on mahdollistaa päätösten teko nopeasti ja paikallisesti. Rethink-iot:n arvioiden mukaan Ciscon tarkoituksena on kehittyä pelkästään verkkolaitteita tarjoavasta yrityksestä samanlaiseksi data-analytiikkajätiksi kuin Google. (Cisco pushes IoT analytics to the extreme edge with mist computing 2014)

Myös Microsoft haluaa päästä mukaan asioiden internetin mukanaan mahdollistamiin tuottoihin. Tällä hetkellä lähes kaikki sulautetut laitteet on rakennettu jonkinlaisen Linux-järjestelmän päälle. Koska asioiden internetin tulevaisuus näyttää hyvin valoisalta, haluaa Microsoftkin luonnollisesti mukaan. Microsoft Windows IoT Core on suunniteltu toimimaan Raspberry Pi 2:lla, MinnowBoard Maxilla ja DragonBoard 410c:llä. MinnowBoard Maxilla voidaan ajaa myös tavallista Windows 10 -käyttöjärjestelmää. Raspberry Pi 3 -tuettu versio, Windows 10 IoT, on Insider Preview -vaiheessa, jolloin halukkaat voivat ottaa sen testikäyttöön ennen julkista jakoa. (Windows compatible hardware development boards 2016.) Microsoftin kehityssuunta on siis hyvin havaittavissa.

Microsoft Windows 10 IoT:n ohjelmointiin tarvitsee kehittäjäversion Visual Studio 2015, jonka käyttöjärjestelmävaatimuksena on Windows 10. Visual Studiolla voidaan sitten ohjelmoida Windows 10 IoT -yhteensopivia ohjelmia. Windows 10 IoT:n heikkona puolena on se, ettei sen mukana tule minkäänlaista työpöytäympäristöä, vaan sitä konfiguroidaan internetselaimella tai komentorivillä. Positiivisena puolena voidaan nähdä Windowsin

UWP API:t (Application Program Interface), joiden avulla voidaan samaa sovellusta suorittaa tuhansissa Windowsin tukemissa laitteissa, sekä järjestelmän keveys. (Learn about Windows 10 IoT Core 2016.)

3.2 Data-analytiikka

Koska asioiden internetin myötä sensoridataa kertyy sensorien lukumäärän kasvaessa paljon, on kaikelle datalle myös tehtävä jotain. Tämä viimeinen vaihe on sitä varten, että tiedon tallennuksen lisäksi siitä pitäisi jotenkin saada selville kaikki arvokas tieto ja sen jälkeen vielä saada se esitettävään muotoon. Tällöin tuotteesta saataisiin palvelu, josta asiakkaat olisivat valmiita maksamaan.

Pilvipalveluiden käytössä on suuri hyöty varsinkin jos sensorien määrä kasvaa. Oman infrastruktuurin rakentaminen on kallista, ja lisäksi infrastruktuurin laajentaminen tulevaisuudessa luo käytännön haasteita niin ylläpidon kuin hintansa vuoksi. Palvelimia laajennettaessa joudutaan usein uudelleenkäynnistämään ja tutkimaan, toimivatko päivitetyt komponentit vanhojen ratkaisujen kanssa. Lisäksi järjestelmiä on ylläpidettävä esimerkiksi komponenttien rikkoutumisen vuoksi. Kun esimerkiksi kiintolevyjä ja tuulettimia lisätään kymmeniä, on todennäköistä, että jokin niistä pettää käytön myötä. Kaikesta mainitusta aiheutuu ongelmia palveluiden toimivuuden kannalta järjestelmien ollessa sammuksissa (downtime). Pilvipalveluiden avulla tulevaisuudessa voidaan helposti skaalautua käytännössä rajattomasti ilman, että infrastruktuurista tarvitsee huolehtia. Ulkoisia palveluita käyttäessä tulee kuitenkin huomioida lait. Suomen laki määrittää tarkkaan, mihin ja kenelle potilastietoja voidaan luovuttaa ja kenen vastuulla tietojen suojaaminen ja ylläpito on. Petisensorien tapauksessa tulee ehdottomasti siis tutustua Suomen lakiin ennen käyttöönottoa. (Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä 2016.)

Opinnäytetyössä tutkimuksen kohteena olleessa OpenAPI WSN Gatewayssa oli mukana ominaisuus, joka tallensi sensoridatan tietokantaan, jotta tallennettua dataa voitaisiin jälkikäteen hallita. Lisäksi mahdollisena ominaisuutena tuli mukana myös WSN Monitor, jonka avulla pystyttiin reaaliajassa seuraamaan yhdyskäytävään yhdistäviä sensoreita ja niiden lähettämiä arvoja. WNS Monitorilla pystyi myös visualisoimaan sensorien toimittamaa dataa reaaliajassa erilaisien kuvaajien avulla. WSN Monitoria käsitellään lisää työn viimeisessä luvussa.

Mikäli kokonaisuuksista halutaan hyödyllisiä, on oikean ja mielenkiintoisen tiedon löytämiseksi nähtävä paljon vaivaa. Analytiikkaa varten jotkut yritykset tarjoavat valmiita ratkaisuja näiden ongelmien ratkaisemiseen. On olemassa myös avoimen lähdekoodin analysointiohjelmistoja, joita voidaan suorittaa halvalla tai ilmaiseksi. Tällöin toki tuotteen ylläpito ja asennus sekä muut ongelmat todennäköisesti jäävät ylläpitäjän omalle vastuulle, eikä tukea löydy kuin foorumeilta ja mahdollisesti wiki-sivustoilta.

Tallennetulle tiedolle tulee tehdä jotain, jotta se saadaan esitettyä ihmiselle niin, että siitä on jotain hyötyä. Suuren tiedonmäärän visualisointi on haastavaa, koska raakadata on ihmiselle varsin turhaa: tietokannasta numeroita luettaessa ei ihmiselle muodostu järkevää kuvaa lukujen trendeistä tai tarkoituksesta. Tätä varten data tulee saattaa esitettävään ja helposti ymmärrettävään muotoon. Mikäli data tallennettaisiin esimerkiksi Googlen pilveen, voitaisiin visualisointiin käyttää esimerkiksi Googlen visualisointirajapintaa, joka piirtää graafit miellyttävään muotoon. (Google Visualization API Reference 2016.) Suurien pilvipalveluiden käytössä voidaan yleensä helposti ottaa käyttöön myös liikenteen salaus. Lisäksi tietokannat ovat turvallisesti tallessa, heikoimmaksi kohdaksi jäädessä ylläpitäjän tai käyttäjän tekemät konfiguraatiovirheet. Kuitenkin esimerkiksi yhdysvaltalaisten palveluiden käytössä on hyvä ottaa huomioon sikäläiset lait, jotka voivat velvoittaa yrityksiä luovuttamaan tietojaan esimerkiksi paikallisen hallituksen käyttöön.

3.3 Asioiden internet ja tietoturva

Koska asioiden internetissä laitteet keskustelevat usein ulkomaailmaan päin eivätkä vain sisäverkossa, yhteyksien ja laitteiden turvaaminen nousee tärkeäksi osaksi toteutusta. Asioiden internetin ja sen alan yksi suurimmista ongelmista onkin tietoturva. Sensoreiden koskemattomuuden takaaminen eli se, etteivät ulkopuoliset pääsisi kolkuttelemaan tai muuten käsiksi niihin, nousee tärkeäksi osaksi toteutuksia ja kokonaisuuksia. Rikollisia eivät välttämättä kiinnosta itse sensorit, vaan niitä voidaan käyttää hyökkäysvektorina. Älylampun tai -leivänpaahtimen hakkerointi tuskin luo suurta lisäarvoa itsessään, mutta jos ne ovat liitettynä yrityksen verkkoon, voidaan huonon toteutuksen tuomien aukkojen avulla päästä käsiksi arvokkaaseen dataan. Esimerkkinä on LIFX-älyvalo, jossa on langaton verkkoyhteys, jotta ihmiset voivat hallita sitä matkapuhelimillaan. Tutkijat onnistuivat sen kautta kuuntelemaan langattoman verkon yhdistämiseen tarvittavat tunnukset. Rikollisen päästessä yrityksen sisäiseen verkkoon on mahdollista aiheuttaa vakavaa vahinkoa. (WakeField 2014)

Koska laitteita tehdään paljon ja mahdollisimman halvalla, niiden tietoturvasta huolehtiminen on kasvavissa määrin hankalampaa – myös vanhoja laitteita pitäisi päivittää. Ei siis riitä, että uudet tai vuoden vanhat laitteet pidetään ajan tasalla. Asioiden internet tuo mukanaan myös paljon uusia yrittäjiä, joilla ei mahdollisesti ole kokemusta turvallisista verkkojärjestelmistä ja kokonaisuuksista. Lisäksi tietoturva ei tuota lisäarvoa tuotteelle, jolloin sen hienosäätäminen ei yritykselle ole välttämättä kannattavaa. Tietoturvaa tulisi kuitenkin painottaa, sillä vahingon sattuessa voi koko yrityksen maine olla vaakalaudalla.

Hengenvaarallisena esimerkkinä voidaan mainita Jeep Cherokee, jonka tietokonejärjestelmät onnistuttiin murtamaan. Murtautuja pääsi käsiksi lähes kaikkiin auton järjestelmiin ja onnistui näin hallitsemaan jopa auton jarrua ja kaasua. Auton haavoittuvuus mahdollisti hyökkääjän pääsemisen käsiksi autoon etänä ilman, että hyökkääjän tarvitsi koskea autoon tai edes fyysisesti tietää sen sijaintia. Hyökkäyksen onnistumiseen riitti, että hyökkääjän liikenne tuli saman operaattorin verkosta, kuin missä auton järjestelmä oli. Jeepin onneksi murtajat toimivat oikein aukon löydettyään ja kertoivat siitä yritykselle ennen aukon löytämisen julkistusta. Jeep toimi myös oikein, sillä kaikki autot oli aukon julkistuksen aikaan elokuussa 2015 jo päivitetty niin, että löydetyt aukot oli korjattu eikä haavoittuvia autoja liikkunut enää liikenteessä. (Drozhzhin 2015.)

Monien mielestä pelkkä SSL (Secure Socket Layer) ei enää riitä liikenteen salaamiseen. Viimeaikaisista paljastuksista on käynyt ilmi, että salausalgoritmeihin on onnistuttu piilottamaan heikennettyjä osia. Täten myöskään VPN-yhteydet (Virtual Private Network) eivät välttämättä ole enää turvassa, joten turvallisuuden ja yhteyden turvallisuuden parantamiseksi tulisi käyttää useamman kerroksen salausta päällekkäin (Sullivan 2014). Lisäksi SSL-avainten vaihtelu ja todentelu jälkikäteen saattaa olla hankalaa. Muutenkin nykyään on suositeltavaa käyttää useamman kerroksen salausta, esimerkiksi SSL + VPN pelkän SSL-salauksen sijaan. Ei voida täysin luottaa vain yhteen tekniikkaan, vaan järjestelmät pitäisi suojata useammalla tavalla oletuksena, että jokin niistä pettää.

Laitteet voidaan liittää VPN-verkkoon, jolloin kaikki liikenne kulkee luotettua ja salattua putkea pitkin. Myös sensoreiden tarjoamat palvelut tulisi minimoida, jotta sensoriin ei pääsisi ulkopuolisia käsiksi. Sensori voisi hyvin käydä itsekseen noutamassa tietonsa ulkoiselta palvelimelta esimerkiksi salattua VPN-putkea pitkin ilman, että sensoriin pääsee kukaan suoraan käsiksi ilman konfigurointitilaan käynnistämistä. Näin myös tietoturva paranee, kun palvelimella voidaan verifioida tiedot. Lisäksi palvelimen tietoturvan

ylläpitäminen on turvallisempaa, helpompaa ja halvempaa kuin jokaisen sensorin päivittäminen ja päivitysten asennuksen tilan seuranta. Alan kehittyessä tähän luultavasti sovitaan yleisiä toimintaperiaatteita ja standardeja, joita seuraamalla toteutuksista voidaan tehdä turvallisia.

3.4 Standardointi

Asioiden internetin standardointi on aivan alkutekijöissään. Ratkaisut on yleensä toteutettu kukin omalla tyylillään, eikä yleisiä toimintaperiaatteita tunnu olevan. Suuret rikkaat yritykset rakentavat omia kokonaisuuksiaan ja sovittavat laitteensa toimimaan vain omien järjestelmiensä kanssa ja tästä syystä asiakkaiden tai loppukäyttäjien on hankala rakentaa toteutuksia omiin tarkoituksiinsa. Suurien yrityksien pilveen liitettyjen ratkaisujen ongelmaksi muodostuu myös se, että toteutukset toimivat vain niin kauan kuin valmistaja ylläpitää verkkopalveluitaan. Standardien ja yhteisten pelisääntöjen puuttuessa laitteet jäävät helposti käyttökelvottomiksi, jos valmistaja ei enää voi tai halua ylläpitää niitä.

Standardien ja yhteisten pelisääntöjen puute sekä alan yleinen pirstaleisuus aiheuttaa loppukäyttäjille ongelmia. Kun laitteet käyttävät omia toteutuksiaan, ne eivät toimi yhteen muiden laitteiden kanssa. Pahimmillaan pirstaleisuus tarkoittaa sitä, että esimerkiksi uudet älyvalot eivät toimisi vanhojen kanssa samalla sovelluksella ja käyttäjä joutuisi vaihtamaan sovellusta halutessaan hallita eri lamppua. Lisäksi jos älylampun valmistaja hylkää toteutuksensa, esimerkiksi konkurssin tai suuren tappion vuoksi, mahdollisesti kaikki valmistajan laitteet lakkaavat kokonaan toimimasta.

Standardoinnin ja yleisten käytäntöjen puute käy ilmi myös alan sanastosta. Sanasto ja termit ovat suomeksi erittäin puutteelliset, ja jopa englanniksi samoista asioista käytetään useita eri nimityksiä yksinkertaisimpana esimerkkinä ”internet of things” ja ”internet of everything”, jotka tarkoittavat käytännössä samaa. Tilanne on sama suomen kielellä: käytössä on useita eri termejä, kuten kaiken internet, asioiden internet ja esineiden internet. Lisäksi yritykset tuovat toteutuksiensa mukana omia termejään ja näkemyksiään järkevistä toteutusperiaatteista. Esimerkiksi Ciscon termit ”fog” ja ”mist” computing ovat muodostumassa yleisiksi termeiksi asioiden internetin alalla. Selkeyden vuoksi läpi opinnäytetyön on käytetty suomenkielistä termiä asioiden internet, sillä termi kattaa hyvin laajasti niin esineet kuin muutkin laitteet, joita ollaan liittämässä internetiin.

Linux Foundation ilmoitti 2013 loppuvuodesta työskentelevänsä Qualcommin kanssa AllSeen Alliance -nimen alla tavoitteenaan luoda avoimen lähdekoodin toteutus asioiden internetille. Nykyään AllSeen Alliancessa on mukana maailman suurimpia yrityksiä jopa yli 100, esimerkkeinä LG, Qualcomm ja Sony (The innovative companies that support AllJoyn 2016).

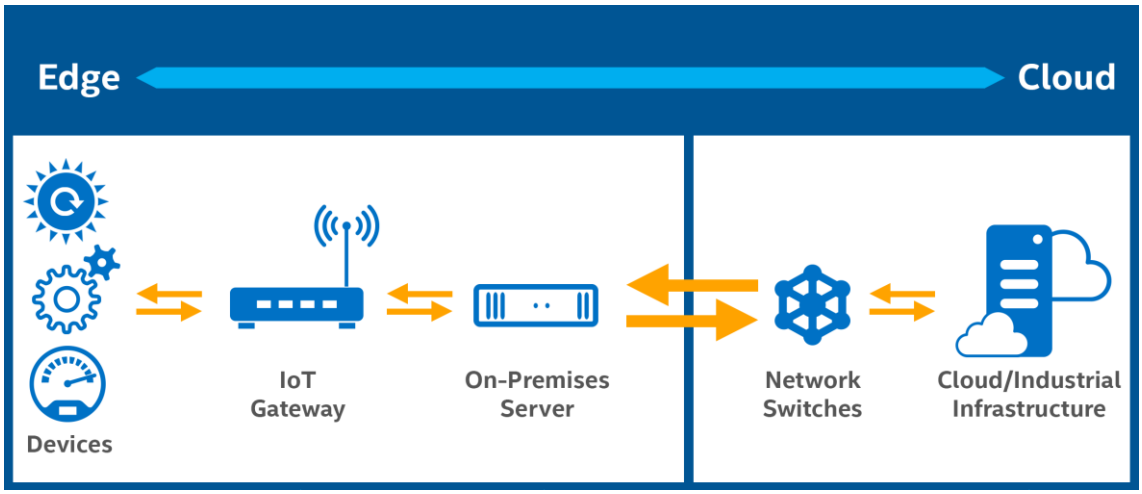
Tammikuussa 2015 AllSeen Alliance ilmoitti työstävänsä ja auttavansa asioiden internetin standardien muodostumisessa. Allseen Alliance on Quallcommin luoman pohjan päälle rakentanut ja julkaissut AllJoyn Gateway Agentin, joka mahdollistaa turvallisten pilvipalveluiden rakentamisen AllJoyn kanssa yhteensopiville laitteille. AllJoyn Gateway Agent voidaan asentaa Linuxille tai OpenWRT:tä käyttäville laitteille, kuten langattomille reitittimille. AllJoyn Gatewayn 1.0 -versiossa on valmius end-to-end –turvalle (End to end encryption) ja salaukselle yhdyskäytävän ja laitteen (device) välille. (Gateway Agent Project 2016.)

4 Tutkimuksen esittely

Tampereen yliopiston kehittämä ja julkaisema WSN OpenAPI Gateway (WOAG) otettiin opinnäytetyön aikana testikäyttöön aikomuksena tutkia sen mahdollisia käyttötapoja ja -tarkoituksia. Tämä yhdyskäytävä on yleinen (general) ja avoin yhdyskäytävä, joka on suunniteltu skaalautuvaksi niin suuriin kuin pieniinkin sensoriverkkoihin. Palvelualustan arkkitehtuuri on suunniteltu ja toteutettu niin, että yhdyskäytävän molemmilla puolilla on avoimet sovellusrajapinnat, joita käyttämällä dataa saadaan virtaamaan laitteelta yhdyskäytävälle ja yhdyskäytävältä erilaisille taustajärjestelmille esimerkiksi pilveen, palvelimille ja sovelluksille. Yhdyskäytävä on avointa lähdekoodia, ja kuka tahansa voi sen halutessaan ladata yhdyskäytävän verkkosivuilta joko lähdekoodeineen tai pelkkänä asennuspakettina.

WSN OpenAPI Gateway on skaalautuva yhdyskäytävätoteutus, jonka avulla voitaisiin toteuttaa itse yhdyskäytävän lisäksi myös tietokantapalveluita erilaisille sensoreille ja niiden tietojen tallennukselle. Skaalautuvuuden vuoksi voitaisiin yhdyskäytävän eri palaset asentaa eri palvelimille. Esimerkiksi yhdyskäytävä ja tietokanta voitaisiin kuorman tasaamisen nimissä asentaa eri palvelimille. Opinnäytetyössä kuitenkin asennettiin koko paketti kerralla mahdollisten ongelmien ja konfigurointivirheiden minimoimiseksi. Petisensori asettuu kuvassa 6 kategoriaan ”devices”; se on siis verkon reunalla (edge) ja sen

tavoitteena on yhdistää ja välittää dataa yhdyskäytävälle. Yhdyskäytävä tekee datalle konfiguroidut toimenpiteet, kuten toimittaa mittaustuloksia eteenpäin esimerkiksi pilveen (cloud) tai tallentaa dataa niitä esimerkiksi paikalliseen tietokantaan.



Kuva 6. Sensorin ja yhdyskäytävän osat kokonaisuudessa (End-to-End Security for Industrial Automation 2016).

4.1 SCA11H-sensori

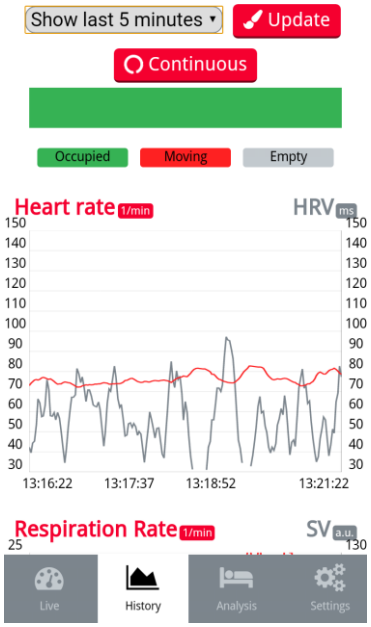
Opinnäytetyössä käytetty testisensori oli Muratan valmistama SCA11H, langaton petisensori. Se sijoitetaan sängyn runkoon, kylkeen tai patjan päälle, kuitenkin niin, ettei sensori jää vahingossa nukkuvan ihmisen alle. Lisäksi sensorin ja ihmisen välillä tulisi olla mielellään vähintään kahdenkymmenen senttimetrin etäisyys toisistaan mittausvirheiden minimoimiseksi. Sensori tunnistaa värähtelyt ja näin sykkeen ja nukkuvan ihmisen mahdolliset liikkeet. Jotta sensori toimii kunnolla, se tulee asentaa samoin päin kuin ihminen nukkuessaan sängyssä eli pitkittäin sängyn mukaan. Virran sensori saa tavallisesta pistorasiasta, joten virransaanti on taattu sähkökatkoja lukuun ottamatta, kunhan sensori on asennettu paikalleen. Sensoriin ei siis tarvitse vaihtaa paristoja tai ladata akkuja. (Product Datasheet 2015.)

Sensorin konfigurointi suoritettiin käynnistämällä se konfigurointitilaan. Konfigurointitilaan päästäkseen tuli sensorista ottaa takakansi irti käynnistyksen yhteydessä, ja näin sensori toimi langattomana tukiasemana tarjoten konfigurointia varten graafisen käyttöliittymän. Sensorin tarjoamaan tukiasemaan yhdistämällä päästiin konfiguroimaan ase-

tuksia halutuiksi. Konfiguroinnin jälkeen sensori käynnistettiin uudelleen, jolloin se käynnistyi normaaliin tilaan ja alkoi toimittaa konfiguroitua tehtäväänsä. Sensori ei osaa käsitellä saapuvaa liikennettä, vaan se pystyy vain lähettämään mittaustuloksiaan konfiguroituun IP-osoitteeseen käyttäen sille asetettua langatonta verkkoa. Sensorin käyttöjärjestelmä oli vielä testausvaiheessa, eikä esimerkiksi liikenteen salaus SSL:llä vielä toiminut.

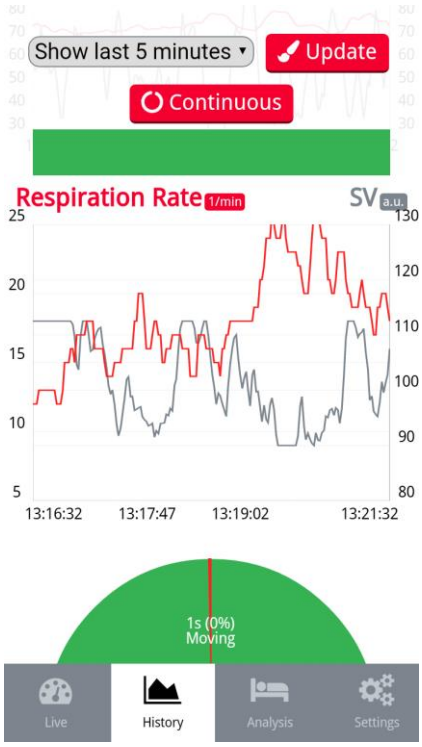
Sensorissa oli kaksi eri tilaa, joihin sen pystyi konfiguroimaan. Ensimmäinen niistä oli paikallinen tila (local mode), joka oli tarkoitettu itsenäiseen toimintaan niin, että sensori tallentaa skannatut tiedot omaan välimuistiinsa ja jokin palvelu tai sovellus voi käydä noutamassa datan käyttöönsä. Toinen mahdollinen tila oli pilvitila (cloud mode), jonka tarkoituksena oli, että sensori ei itse tallenna dataa vaan toimittaa sitä eteenpäin yhdyskäytävälle. Paikallinen tila toimi, ja sitä voitiin käyttää Muratan tarjoamilla Android- ja Windows-sovelluksilla. Testisovellukset olivat tarjolla Muratan verkkosivuilla sensorin ohjeiden ja dokumentaatioiden ohessa. Android-sovellus visualisoi sensorin havaitseman datan, jonka avulla pystyttiin seuraamaan ja tallentamaan ihmisen sykettä ja respiraatiota. (Product Datasheet 2015.)

Kuvasta 7 voidaan havaita punaisella viivalla esimerkki-ihmisen syke ja harmaalla viivalla syketaajuus.



Kuva 7. Muratan demosovellus, jossa piirrettynä sensorin havaitsema syke ja syketaajuus.

Kuvasta 8 voidaan seurata ihmisen hengitystä ja iskutilavuutta. Lisäksi sovellus piirsi piirakkamallisen kuvaajan ihmisen liikkeistä, jolloin voidaan havaita, kuinka paljon ihminen liikkuu nukkuessaan. Pitkällä aikavälillä arvoista voidaan tehdä johtopäätöksiä ihmisen terveydentilasta ja esimerkiksi unen laadusta.



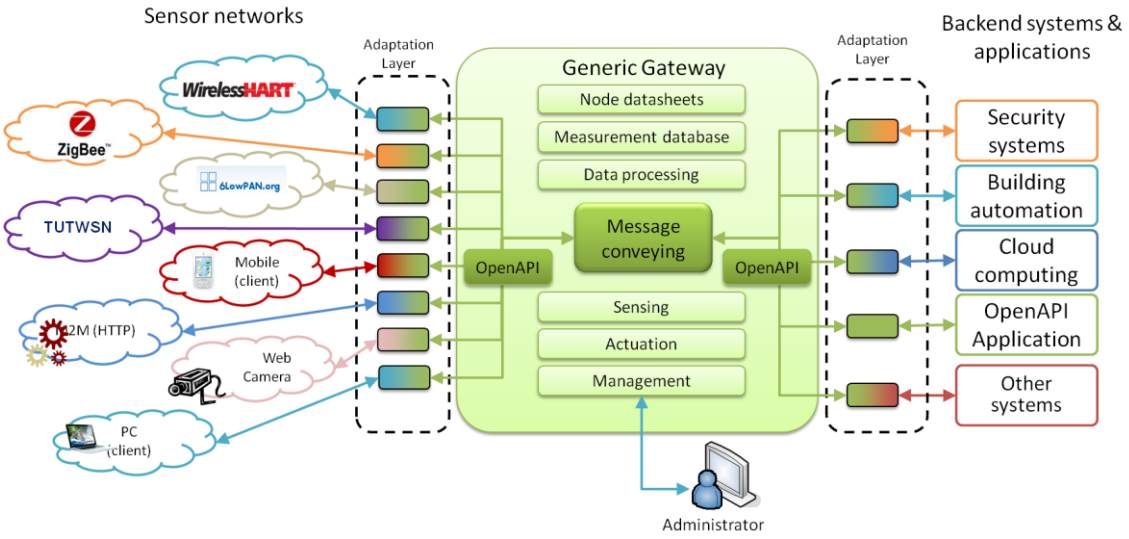
Kuva 8. Muratan demosovellus, jossa piirrettynä sensorin havaitsema hengitys ja iskutilavuus.

4.2 Yhdyskäytävän asennus

WSN OpenAPI Gateway asennettiin Windows 7 -virtuaalikoneelle, jota käytettiin kannettavalla tietokoneella. Virtuaalikoneen verkkokortti sillattiin (bridge) ilman osoitteenmuutosta (NAT), jotta sille saatiin oma IP-osoite. Näin virtuaalikone pystyi kommunikoimaan omalla osoitteellaan ulkomaailmaan ja sensori pystyi välittämään tietoa suoraan sille. Verkkokortin siltauksella vältettiin myös mahdolliset palomuuriongelmat isäntäkoneen osalta sekä porttiohjauksen mukanaan tuomat haasteet ja ongelmat, joita olisi voitu joutua selvittämään.

Kuten kuvasta 9 voidaan päätellä, sensoriverkot vasemmalla puolella ovat mitä tahansa verkkoja, jotka keskustelevat avoimen rajapinnan kautta yhdyskäytävän kanssa, ilman että tiedonsiirtotekniikasta tarvitsee huolehtia. Mahdolliset taustajärjestelmät voivat

myös käyttää yhdyskäytävän tarjoamaa rajapintaa, jonka avulla yhdyskäytävän ja muiden järjestelmien välinen kommunikointi toimisi geneerisesti. Järjestelmänvalvoja hallitsee sensorien lähettämää dataa ja niiden mahdollista tallennusta WSN OpenAPI Gatewayn työkaluilla.



Kuva 9. Ideaali tilanne, jossa yhdyskäytävä on keskellä. Sensorit, taustajärjestelmät ja sovellukset kommunikoivat avoimen rajapinnan kautta yhdyskäytävälle. (Architecture 2016.)

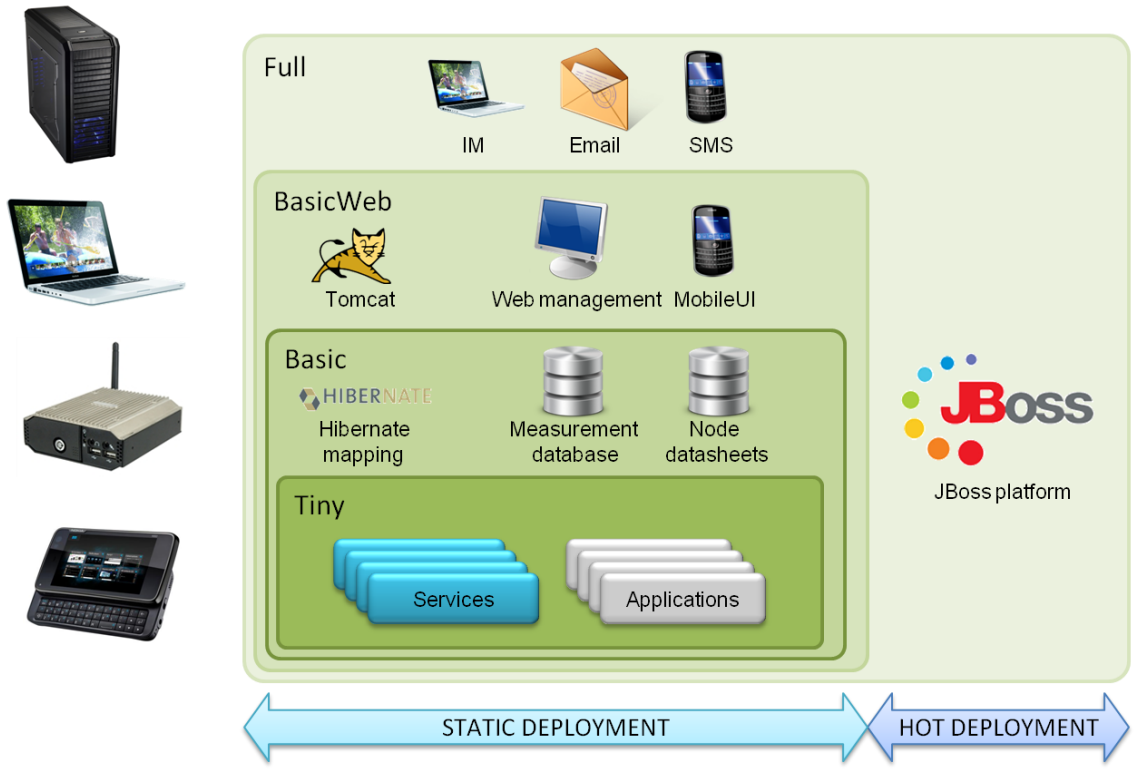
Yhdyskäytävän asennetaan suorittamalla paketin mukana tullut asennusohjelma. WSN OpenAPI Gateway -tuotteet on jaettu neljään osaan (Features 2016). Asennuksen yhteydessä valitaan asennuksen laajuus: Full, BasicWeb, Basic tai Tiny. WSN OpenAPI Gatewayn esivaatimukset ovat niin vaatimattomat, että käytännössä millä tahansa nykyaikaisella tietokoneella pystytään sitä suorittamaan. Kevyimmän asennuksen suositusvaatimuksina on Intel Atom, 200 megatavua muistia ja 50 megatavua levytilaa. Jos halutaan asentaa raskaampi versio, esimerkiksi Basic, suositukset nousevat muistin osalta gigatavuun, prosessorin ja levytilan osalta pätevät edelleen samat suositukset. On kuitenkin otettava huomioon mahdollisen tietokannan koko ja sille tarvittava levytila tulevaisuutta silmälläpitäen. (Generic WSN Gateway Installation 2016.)

Asennusta varten tulee olla graafinen ympäristö, koska asennusohjelma on graafinen. Yhdyskäytävän asennusta varten isäntäkäyttöjärjestelmässä tulee olla asennettuna Java J2SE 1.6. Asennuspaketin mukana tulevat kaikki muut tarvittavat komponentit järjestelmän asentamista ja suorittamista varten. Asennuksen esivaatimuksena suorituskykyvaatimuksien ja Javan lisäksi ovat käyttöjärjestelmänä Linux tai Microsoft Windows

2000 SP3 tai uudempi. (Generic WSN Gateway Installation 2016.) Koska käyttöjärjestelmänä myös Linux käy, voitaisiin WSN OpenAPI Gateway mahdollisesti asentaa vaikka tavalliseen kaupalliseen reitittimeen esimerkiksi OpenWRT:n päälle. Näin ei tarvittaisi erillistä yhdyskäytävälaitetta vaan tukiasema toimisi samalla älykkäänä IoT-yhdyskäytävänä, langattomana tukiasemana ja reittinä internetiin.

Asennuksen aluksi tulee suunnitella, minkäasteinen asennus yhdyskäytävästä tullaan tekemään. Esimerkiksi pienimmässä ja kevyimmässä Tiny–järjestelmässä mukana ovat vain toiminnan kannalta keskeisimmät vaadittavat komponentit. Järjestelmän mukana ei tule esimerkiksi tietokanta- tai muistipalveluita. Tinyn asetuksia ei voida myöskään editoida suorittamisen aikana, vaan asetusmuutokset tulevat voimaan vain uudelleenkäynnistyksen yhteydessä. Basic-järjestelmän mukana tulee Tinyn ominaisuuksien lisäksi myös muita ominaisuuksia, kuten tietokanta mahdollisesti tallennettavalle sensoridatalle. Lisäksi tarjolla on myös BasicWeb ja Full. BasicWeb sisältää Tomcat-web-palvelimen, jonka avulla verkkoa, sensoreita ja niiden kokonaisuuksia voidaan hallita. Laajimpana asennusmuotona tarjolla on Full, joka sisältää edellä mainittujen ominaisuuksien lisäksi erilaisia viestitysominaisuuksia, joihin ei tällä erää nähty tarpeelliseksi tutustua. (Features 2016.)

Paketin laajuutta valittaessa tulee miettiä, mihin tehtävään järjestelmä tulee. Kuvasta 10 voidaan havaita, kuinka paketin laajuus määrittää ominaisuudet ja kuinka laajempi paketti sisältää aina suppeampien pakettien ominaisuudet. Ominaisuudet on kategorisoitu järkeviin kokonaisuuksiin, jotta halutun järjestelmän toteuttaminen olisi mahdollisimman helppoa ja yksinkertaista.



Kuva 10. WSN OpenAPI Gatewayn tuoteperhe ja erikokoisten asennuspakettien sisältämät ominaisuudet ja erot suhteessa toisiinsa (Features 2016).

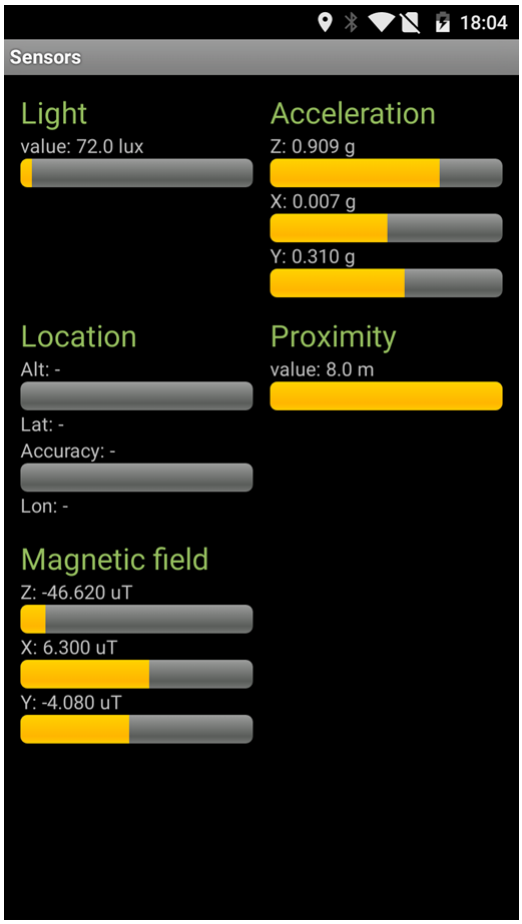
Asetusten konfigurointi on mahdollista tehdä käsin muokkaamalla asetustiedostoa, mutta mukana tuli myös graafinen asetustenmuokkaussovellus, joka näytti valmiiksi muokattavat kentät ja esimerkkiarvot. Konfigurointitiedosto sijaitsi oletuksena asennushakemistossa. Yhdyskäytävän testausta varten paketin mukana tulee pieni testiohjelma, joka lähettää tietokoneen sensoridataa yhdyskäytävälle. Näin voidaan testata, että yhdyskäytävä on asentunut oikein ja että asetukset ovat kunnossa. Testiohjelmalla ei kuitenkaan voida testata verkon tai yhteyksien toimivuutta, koska sovellus lähettää tiedot vain paikallisesti käytettävän järjestelmän paikalliseen verkko-osoitteeseen (localhost), eikä tieto kulje missään vaiheessa verkkokortille asti.

Asennettaessa BasicWeb- tai Full-tason tuoteperhe, voidaan sensoreita ja niiden kokonaisuuksia hallita Tomcat-palvelinohjelmistolla. Kokonaisuuksien hallinta onnistuu käynnistämällä ensin palvelinohjelmisto ja avaamalla sen jälkeen selainyhteys palvelimen, eli yhdyskäytävän porttiin 8080. Kokonaisuuksien luominen helpottaa suurten sensorimäärien hallintaa ja ryhmittelyä.

4.3 Testisensori

Yhdyskäytävän asennuksen jälkeen otettiin testikäyttöön WSN OpenAPI Gatewayn kanssa yhteensopiva sensorisovellus. Verkkosivuilla oli tarjolla Andorid-sovellus, jonka avulla yhdyskäytävää voitiin testata ennen kuin alettiin suunnitella SCA11H:n ja yhdyskäytävän yhteensovittamista. Sovellus käytti puhelimessa tarjolla olevia sensoreita datanlähteenä ja lähetti niitä konfiguroituun osoitteeseen. Datalähteinä toimivat esimerkiksi valoisuusanturin valodata tai etäisyysanturin ilmoittama etäisyys puhelimesta. (Android Node 2016.) Sovelluksen asennus saatiin suoritettua asentamalla apk-sovellustiedosto Android-puhelimeen. Ennen asennusta piti Androidin asetuksista käydä sallimassa ulkopuolisten sovellusten asennus, jottei käyttöjärjestelmä estäisi sovelluksen asennusta (Protect against harmful apps 2016).

Asennuksen jälkeen tuli sensorisovellus ensimmäiseksi konfiguroida. Konfigurointi vaati kohde IP osoitteen, portin, sensorille käyttäjätunnuksen ja salasanan, verkon nimen sekä sensorille nimen (node identifier). Mikäli asetukset oli konfiguroitu oikein, sovellus tunnistautui ja yhdisti asennettuun yhdyskäytävään. Yhdistämisen jälkeen sensorin asetuksista voitiin vielä valita, mitä sensoridataa haluttiin mitata ja lähettää ja kuinka usein. Tässä tapauksessa päivitysväliksi valittiin 5 sekuntia; näin testidataa saatiin kerättyä mahdollisimman paljon ja nopeasti. Sovelluksella pystyi myös esikatsomaan ja tarkastamaan saatavilla olevat sensorit ja niiden toiminnan ennen käyttöönottoa. (Android Node 2016.) Kuvasta 11 voidaan havaita, että esimerkiksi valo- ja etäisyyssensorit toimivat, mutta sijaintia ei saatu toimimaan luultavasti sovelluksen kanssa yhteesopimattoman Android-version takia.



Kuva 11. Yhdyskäytävän kanssa yhteensopiva Android-älypuhelinsovellus ja puhelimessa toimivat sensorit ja niiden tuottamat arvot.

5 Yhdyskäytävän ja sensorin käyttöönotto

Yhdyskäytävän ja sensorin todennuksessa tapahtuvien eroavaisuuksien vuoksi ne eivät pystyneet sellaisenaan kommunikoimaan keskenään. Spesifikaatioiden mukaan sensori käyttää datan välittämiseen WSN OpenAPI Gateawayn tukemaa rakennetta, mutta todennukset toimivat eri tavalla.

Todennuksien eroavaisuuksien vuoksi yhdyskäytävää ja sensoria ei saatu keskustelemaan keskenään, mutta niiden toimintaa voitiin kuitenkin tutkia. Molempien tekniset spesifikaatiot olivat internetissä tarjolla, ja niitä voitiin vertailla. Vertailun avulla päästiin selville siitä, mistä johtui, etteivät ne suostuneet kommunikoimaan tai toimimaan yhdessä.

5.1 WSN OpenAPI Gatewayn yhteyden toimintaperiaate

Valitun yhdyskäytävän spesifikaation mukaan se on rakennettu niin, että se vain odottaa saapuvaa yhteyttä. Oletuksena palvelu on päällä portissa 10994, johon asiakas eli tässä tapauksessa petisensori, voi aloittaa yhteydenoton. Yhteyden muodostamisen tulisi alkaa asiakkaan todennuspyynnöllä (AuthenticationRequest) kuvan 12 mukaisesti. Todennuspyynnön tulisi sisältää asiakkaan käyttäjätunnus ja salasana. (WSN OpenAPI Technical Specification 2013, 16.) Mikäli tunnukset ovat oikeat, yhdyskäytävä vastaa kuvan 13 mukaisesti http-vastauskoodilla 200, eli ”OK, Operation was successful” (Hypertext Transfer Protocol (HTTP) Status Code Registry 2016; WSN OpenAPI Technical Specification 2013, 16). Todennuksen jälkeen asiakkaan ja yhdyskäytävän välinen istunto pysyy avoinna, eikä uusia todennuksia tarvita. Noin viiden minuutin hiljaisuuden jälkeen yhteys kuitenkin katkeaa, jolloin asiakas joutuu todentumaan uudestaan. (WSN OpenAPI XML Introduction 2013: 1.)

Listing 1: *Authentication request*

```
<AuthenticationRequest version="1" xmlns="urn:wsn-openapi:acf" method="password">
  <Parameter name="username" value="user"/>
  <Parameter name="password" value="pass"/>
</AuthenticationRequest>
```

Kuva 12. Esimerkki yhdyskäytävän vastaanottamasta validista todennuspyynnöstä käyttäjätunnuksella ”user” ja salasanalla ”pass” (WSN OpenAPI Technical Specification 2013: 23).

Listing 2: *Authentication response*

```
<AuthenticationResponse version="1" xmlns="urn:wsn-openapi:acf" responseCode="200"/>
```

Kuva 13. Esimerkki yhdyskäytävän vastauksesta onnistuneen todennuksen jälkeen, vastauskoodilla 200 (WSN OpenAPI XML Introduction 2013: 1).

Yhteyden muodostuksen jälkeen sensori toimii datalähteenä yhdyskäytävälle ja yhdyskäytävä vastaanottaa asiakkaalta tulevan datan. Mikäli data on yhdyskäytävän ymmärtämässä muodossa, voidaan sille tehdä halutut toimenpiteet, esimerkiksi piirtää kuvaajaa tai tallentaa data tietokantaan. Yhdyskäytävä voi vastaanottaa todennetuilta asiakkailta viestejä käytännössä rajattoman määrän. Viestit voidaan lähettää joko niin, että viestissä on sensorin tuottamaa dataa, tai niin, että viesti sisältää sensorin datat pidemmältä aikaväliltä. WSN OpenAPI Gatewayn version päivittyessä on siihen lisätty ominaisuuksia ja ymmärrystä järkevämmistä ja monipuolisemmista viestintämuodoista, kuvat

14 ja 15. Uudemman version rakenne mahdollistaa suuremman määrän dataa pakettia kohti, kun vanhemman version viestit sisältävät vain yhden tietueen pakettia kohti. Vanhemman version viestirakenteet ovat edelleen tuettuja uudemmissa versioissa.

Listing 3.2: *Simple measurement with one value.*

```
<Data version="1.7" xmlns="urn:wsn-openapi:sidf">
 <Network id="1">
  <Node id="2">
   <Sensor id="3" >
    <Measurement quantity="Temperature" unit="C" time="2009-07-03T11:24:46+00:00" >
     <Component>23.0</Component>
    </Measurement>
   </Sensor>
  </Node>
 </Network>
</Data>
```

Kuva 14. Esimerkkiviestin sisältö yhden sensorin yhden hetken datasta (WSN OpenAPI Technical Specification 2013: 34).

```
<SIDF version="1.3" xmlns="urn:wsn-openapi:sidf">
 <Network id="1">
  <Node id="2">
   <Sensor id="3" >
    <Measurement quantity="Temperature" unit="C" time="2009-07-03T11:24:46+00:00">
     <Component>23.0</Component>
    </Measurement>
    <Measurement quantity="Acceleration" unit="mg" time="2009-07-25T14:24:47+00:00" >
     <Component id="x">142.0</Component>
     <Component id="y">46.0</Component>
     <Component id="z">895.0</Component>
     <Component id="roll" unit="degree">8.0</Component>
     <Component id="pitch" unit="degree">2.0</Component>
     <Component id="total">907.36156</Component>
    </Measurement>
   </Sensor>
   <Sensor id="5" >
    <Measurement quantity="Temperature" unit="C" time="2009-07-25T14:24:46+00:00" >
     <Component>25.0</Component>
    </Measurement>
   </Sensor>
  </Node>
 </Network>
</SIDF>
```

Kuva 15. Esimerkki yhdyskäytävän vastaanottamasta vanhemman version 1.3 viestistä (WSN OpenAPI XML Introduction 2013: 2).

```
<Data version="1.7" xmlns="urn:wsn-openapi:sidf">
 <Network id="my_network">
  <Node id="mynode">
   <Sensor id="thermometer">
    <Measurement quantity="temperature" unit="C" time="2012-12-24T19:00:00Z">
     <Component/>
     <Values tick="sec">
     0,12.2
     59,12.3
     121,12.2
     160,12.8
     </Values>
    </Measurement>
   </Sensor>
  </Node>
 </Network>
</Data>
```

Kuva 16. Esimerkki yhdyskäytävän vastaanottamasta uudemman version 1.7 viestistä, jossa dataa on neljältä ajanhetkeltä: 0, 59, 121 ja 160 (WSN OpenAPI Technical Specification 2013: 35).

5.2 Petisensori SCA11H:n yhteyden toimintaperiaate

Valmistajan spesifikaation mukaan SCA11H-sensori on toteutettu niin, että se ottaa oletuksena yhteyttä porttiin 80. Todennus käyttää POST-viesteihin standardin RFC 2616 mukaista protokollaa. Uudemmissa kuin 2.4.0-firmware -versioissa palvelimen portin voi muuttaa. Käytössä olleeseen sensoriin oli asennettu juuri 2.4.0, joten porttia voitiin tässä tapauksessa muuttaa.

Sensorin tuottama todennusviesti sisältää käyttäjätunnuksen ja base64-koodatun salasanan, esimerkiksi “Basic dXNlcjpwYXNz“ kuten kuvissa 17 ja 18. Sensori odottaa, että palvelin vastaa, ja alkaa vasta sen jälkeen toimittaa dataa. Palvelimen vastausta odotetaan sen takia, että sensori saa synkronoitua kellonaikansa samaksi kohteensa kanssa. Yhdyskäytävän http-vastauksessa otsikon (header) tulee sisältää kellonaika, jota sensori käyttää hyväkseen kellonaikansa päivittämiseen. Todennuksen jälkeen sensori alkaa lähettää dataa palvelimelle. Onnistuneen yhteydenmuodostuksen jälkeen sensori ei kuuntele enää palvelimelta tulevia vastauksia, vaan lähettää dataa konfiguroiduin aikavälein. Sensori lähettää datan niin, että jokaisen viestin alussa on todennusheader, ja tämän jälkeen data XML-muodossa. (SCA11H Cloud server interface specification 2015.)

```
POST /data/push/ HTTP/1.1
Host: your.cloud.server
Authorization: Basic dXNlcjpwYXNz
Content-Type: application/x-openapi-sidf+xml
Content-Length: LEN
```

Kuva 17 Esimerkki petisensorin lähettämästä todennusviestistä (SCA11H Cloud server interface specification 2015: 4).

```
POST /data/push/ HTTP/1.1
Host: your.cloud.server
Authorization: Basic dXNlcjpwYXNz
Content-Type: application/x-openapi-sidf+xml
Content-Length: 1609

<Data version="1.7" xmlns="urn:wsn-openapi:sidf">
    <Network id="test_network">
        <Node id="test_node">
            <Sensor id="0">
                <Measurement quantity="BioSignal" time="2014-03-
                07T13:18:04+00:00">
                    <Component id="heart rate" unit="bpm" />
                    <Component id="respiration rate" unit="rpm"/>
                    <Component id="relative stroke volume" unit="µl"/>
                    <Component id="heart rate variability" unit="ms"/>
                    <Component id="measured signal strength"/>
                    <Component id="status"/>
                    <Component id="beat-to-beat time" unit="ms"/>
                    <Component id="beat-to-beat time -1" unit="ms"/>
                    <Component id="beat-to-beat time -2" unit="ms"/>
                    <Values tick="sec">
                        0,81,11,45,212,1547,1,553,631,0
                        1,84,11,37,43,4280,2,611,0,0
                        2,83,12,39,41,14280,2,911,0,0
                        ...
                    </Values>
                </Measurement>
            </Sensor>
        </Node>
    </Network>
</Data>
```

Kuva 18 Esimerkki petisensorin lähettämästä viestistä, jossa todennus on http-lähetyksen (post) ylätunnisteessa ja skannattuja arvoja ajanhetkillä 0, 1 ja 2 (SCA11H Cloud server interface specification 2015: 6).

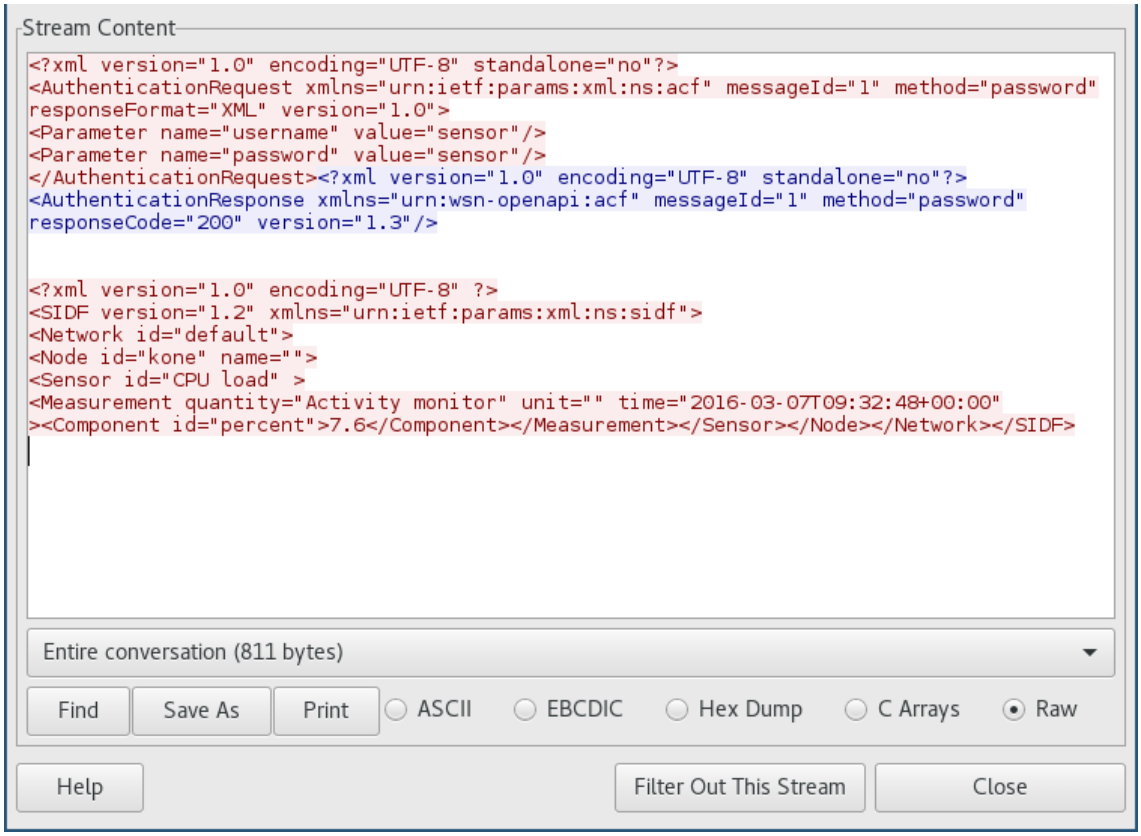
5.3 Petisensorin ja yhdyskäytävän toimintaperiaatteiden vertaaminen

Kuvassa 19 on verkkoliikennekaappaus onnistuneesta todennuksesta Wireshark-ohjelmalla ja yhden sensorin onnistuneesti lähettämästä datasta. Paketti 15547 tulee sensorilta sisäverkon osoitteesta 192.168.0.120 yhdyskäytävän osoitteeseen 192.168.0.222 porttiin 10994 ja sisältää spesifikaatioiden mukaisen todennusviestin. Todennusviestiin yhdyskäytävä vastaa http-tilavastauskoodilla (http response status code) 200 eli OK, kuvassa paketti 15548. Onnistuneen todennuksen jälkeen sensori alkaa lähettää onnistuneesti dataa kofiguroiduin aikavälein, paketit 15553 ja 15555. Lopuksi sensorisovelluksen sulkemisen vuoksi sensori katkaisee yhteyden yhdyskäytävän kanssa, kuvassa paketti 15577.

No.	Time	Source	Destination	Protocol	Length	Info
15544	1810.522530465	192.168.0.120	192.168.0.222	TCP	74	50618 → 10994 [SYN] Seq=0 Win=29200 Len=0
15545	1810.522995699	192.168.0.222	192.168.0.120	TCP	74	10994 → 50618 [SYN, ACK] Seq=0 Ack=1 Win=
15546	1810.523025736	192.168.0.120	192.168.0.222	TCP	66	50618 → 10994 [ACK] Seq=1 Ack=1 Win=29312
15547	1810.523148016	192.168.0.120	192.168.0.222	TCP	359	50618 → 10994 [PSH, ACK] Seq=1 Ack=1 Win=
15548	1810.559559549	192.168.0.222	192.168.0.120	TCP	244	10994 → 50618 [PSH, ACK] Seq=1 Ack=294 Wi
15549	1810.559590862	192.168.0.120	192.168.0.222	TCP	67	50618 → 10994 [PSH, ACK] Seq=294 Ack=179
15550	1810.766137963	192.168.0.120	192.168.0.222	TCP	67	[TCP Keep-Alive] 50618 → 10994 [PSH, ACK]
15551	1810.766604803	192.168.0.222	192.168.0.120	TCP	78	10994 → 50618 [ACK] Seq=179 Ack=295 Win=6
15553	1811.525117104	192.168.0.120	192.168.0.222	TCP	318	50618 → 10994 [PSH, ACK] Seq=295 Ack=179
15554	1811.725109449	192.168.0.222	192.168.0.120	TCP	66	10994 → 50618 [ACK] Seq=179 Ack=547 Win=6
15555	1811.725177028	192.168.0.120	192.168.0.222	TCP	153	50618 → 10994 [PSH, ACK] Seq=547 Ack=179
15556	1811.927758852	192.168.0.222	192.168.0.120	TCP	66	10994 → 50618 [ACK] Seq=179 Ack=634 Win=6
15577	1819.368957132	192.168.0.120	192.168.0.222	TCP	66	50618 → 10994 [RST, ACK] Seq=634 Ack=179

Kuva 19. OpenAPI WSN Gatewayn ja sensorin välistä liikennettä.

Kuvan 20 kuvakaappauksessa on luettavissa yhteyden aikana liikkuneiden pakettien sisältö; punaisella sensorin lähettämät ja sinisellä yhdyskäytävän lähettämä liikenne. Ensimmäisestä punaisesta paketista voidaan havaita, kuinka salasana ja käyttäjätunnus ovat ”sensor” ja ”sensor”. Vastauksesta voidaan havaita vastauskoodi 200, joka kertoo sensorille, että dataa voi alkaa lähettää ja että todennus on onnistunut. Lähetetystä datasta voidaan havaita viestin rakenne: kuinka kentät ”Network id” on default, node id on ”kone” ja name on tässä tapauksessa jätetty tyhjäksi. Myös sensorin id voidaan havaita eli ”CPU load”. Viestin lähetysaika ja esimerkkisensorin, eli sovellusta suorittavan tietokoneen prosessorin, kuormituksen arvo prosentteina 7,6 %.



Kuva 20. Raakadataa, josta voidaan lukea pakettien sisältö ja todeta, kuinka todennus on onnistunut ja dataa on saatu välitettyä sensorin ja yhdyskäytävän välillä.

Petisensori ei tarkasta yhteyttä muodostaessaan, mitä vastauksia vastaanottavalta yhdyskäytävältä tulee. Mikäli yhteys vain suostutaan avaamaan, sensori alkaa lähettää dataa konfiguroituun osoitteeseen. Koska OpenAPI WSN Gateway ei ymmärrä sensorin yhdistysyrityksiä eikä täten suostu avaamaan yhteyttä, petisensori ei saa synkronoitua kelloaan. Testikäyttöön voidaan luoda esimerkiksi erittäin yksinkertainen pythonpalvelin, joka suostuu vastaanottamaan ja avaamaan kaikki tulevat yhteydet. Tätä voi tarkastella kuvassa 21. Palvelimen ei tarvitse ymmärtää sensorin käyttämää rakennetta, sillä liikennettä voidaan tallentaa esimerkiksi Wiresharkilla tai tcpdumpilla. Vaikka palvelinsovellus ei ymmärrä edes http-lähetystä, alkaa sensori yhteyden muodostuttua lähettää dataa kyseiseen osoitteeseen.

```
import SimpleHTTPServer
import SocketServer

PORT = 10994

Handler = SimpleHTTPServer.SimpleHTTPRequestHandler

httpd = SocketServer.TCPServer(("", PORT), Handler)

print "serving at port", PORT
httpd.serve_forever()
~
~
~
~
~
~
~
"python/webserv.py" 16L, 222C written                          14,0-1        All
```

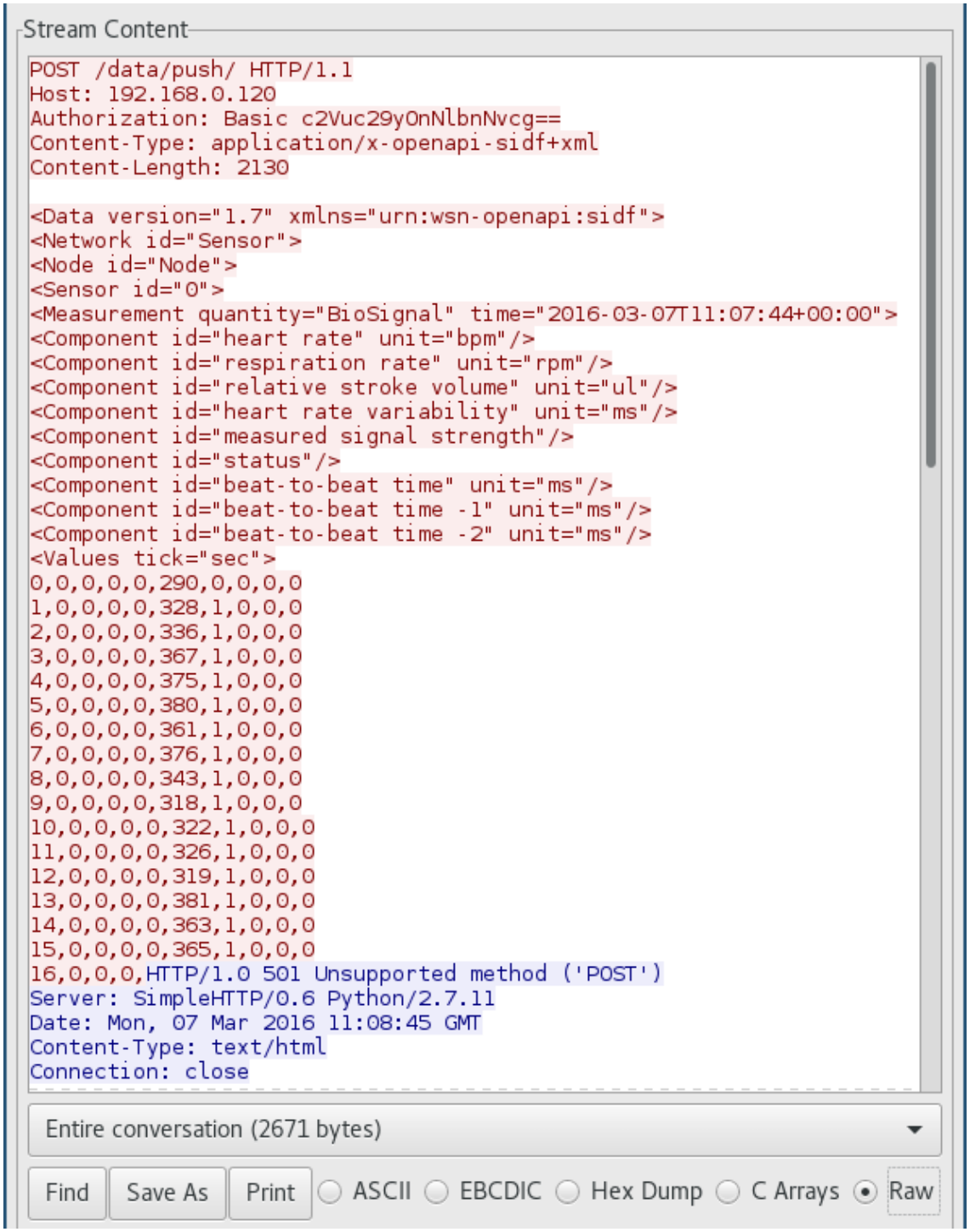
Kuva 21. Yksinkertaisen Python-palvelimen lähdekoodi, josta voidaan havaita muun muassa käytössä ollut portti 10994 (SimpleHTTPServer – Simple HTTP request handler 2016).

Kuvan 22 paketeista 273-275 voidaan havaita, kuinka petisensori ja Python-palvelin muodostavat yhteyden. Paketti 276 on ensimmäinen datapaketti ja sisältää petisensorin analysoimat arvot. Tähän palvelin vastaa 501, eli ”Unsupported method ’POST’”. Vaikka Python-palvelin ei viestiä ymmärtänyt, data kuitenkin saatiin talteen.

No.	Time	Source	Destination	Protocol	Length	Info
273	100.459851614	192.168.0.208	192.168.0.120	TCP	58	4099 → 10994 [SYN] Seq=0 Win=4608 Len=0 MSS=1152
274	100.459904941	192.168.0.120	192.168.0.208	TCP	58	10994 → 4099 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
275	100.463841984	192.168.0.208	192.168.0.120	TCP	54	4099 → 10994 [ACK] Seq=1 Ack=1 Win=4608 Len=0
276	100.463904497	192.168.0.208	192.168.0.120	TCP	1206	[TCP segment of a reassembled PDU]
277	100.463938266	192.168.0.120	192.168.0.208	TCP	54	10994 → 4099 [ACK] Seq=1 Ack=1153 Win=31104 Len=0
278	100.464678074	192.168.0.120	192.168.0.208	TCP	96	[TCP segment of a reassembled PDU]
279	100.465012020	192.168.0.120	192.168.0.208	HTTP	392	HTTP/1.0 501 Unsupported method ('POST') (text/html)
280	100.467503423	192.168.0.208	192.168.0.120	HTTP	1193	POST /data/push/ HTTP/1.1 (application/x-openapi-sidf+xml)

Kuva 22. Python-palvelimen ja petisensorin välistä liikennettä.

Liikenteestä voidaan todeta, että ongelma petisensorin ja OpenAPI WSN Gatewayn kanssa on siinä, että yhdyskäytävä ei ymmärrä petisensorin todennusyritystä eikä täten suostu vastaamaan sensorin yhdistysyrityksiin. Mikäli yhteys muodostuisi, olisi mahdollista, että yhdyskäytävä ymmärtäisi sensorin lähettämää dataa ja pystyisi sitä käsittelemään, sillä spesifikaatioiden mukaan petisensorin lähettämä data vaikuttaa uudemman version 1.7 mukaan oikeanlaiselta ja oikearakenteiselta. Kuva 23 kuvaa Python-palvelimen ja sensorin välisen liikenteen: kuinka sensori lähettää todennusyrityksensä jälkeen dataa onnistuneesti ja kuinka Python-palvelin vastaa ”Unsupported method ’POST’”.



Kuva 23. Python-palvelin ja petisensorin välistä liikennettä raakadatana. Kuvasta voidaan lukea petisensorin ja palvelimen välisen liikenteen sisältö.

5.4 Toimiva esimerkkiratkaisu

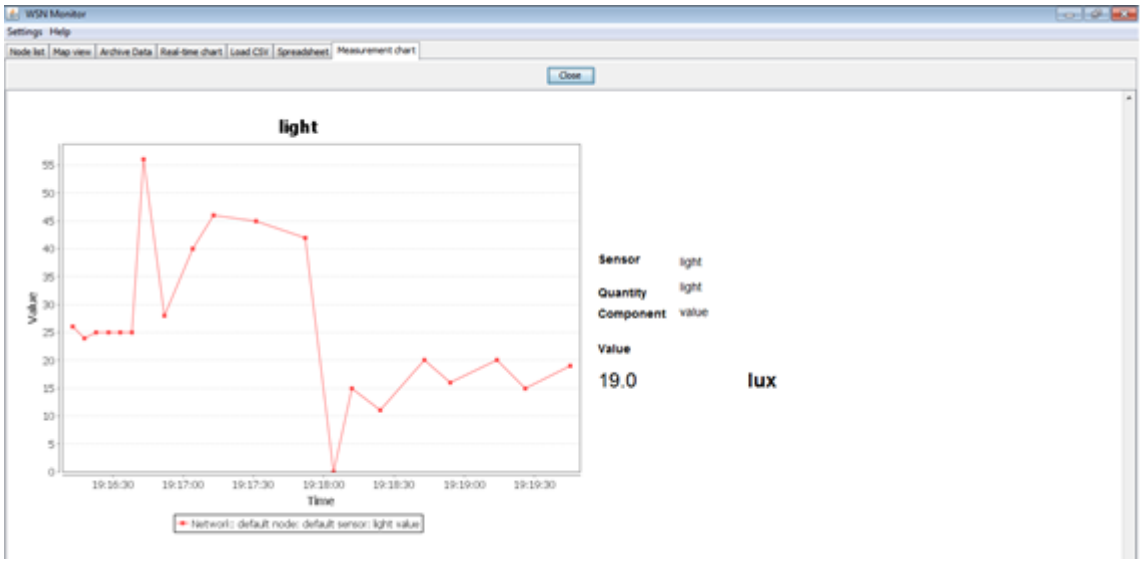
Koska Muratan sensori ei toimi sellaisenaan WSN OpenAPI Gatewayn kanssa, päätettiin, että tehdään tilalle toimiva esimerkkiratkaisu. Esimerkkiratkaisussa otettiin käyttöön Raspberry Pi, joka toimii datanlähteenä ja toimittaa arvoja yhdyskäytävälle. Raspberry Pin käyttöjärjestelmäksi valittiin Raspbian sen virallisuuden ja toimintavarmuuden vuoksi. Käyttöjärjestelmä asennettiin Raspberry Pin Micro-SD muistikortille Raspbianin ohjetta seuraamalla (Raspbian Installer 2016). Esimerkkisovelluksen tarkoituksena oli todistaa, että WSN OpenAPI Gateway voi toimia myös muiden kuin valmiiksi annettujen ratkaisujen kanssa. Tarkoituksena ei ole rakentaa vahvaa tietoturvaa eikä salata liikennettä, vaan saada vain sovellus toimimaan niin, että se todentaa yhteytensä yhdyskäytävän kanssa ja yhdyskäytävä ymmärtää saapuvan liikenteen niin, että sitä voidaan käsitellä WSN OpenAPI Gatewayn työkaluilla.

Sovelluksen pohjana käytettiin OpenAPI WSN Gatewayn sivustolla tarjolla ollutta lähdekoodia, jonka pohjalta ohjelmoitiin Pythonilla halutut ominaisuudet ja toimitettiin haluttu data yhdyskäytävän osoitteeseen. Raspbian sisältää oletuksena Python-kääntäjän, mutta ohjelman toimivuuden kannalta ja ohjelmoinnin helpottamisen vuoksi asennettiin python2-psutils-paketti Raspbianin virallisesta pakettilähteestä (repository). Psutil on järjestelmien tilan seuraamista varten rakennettu Python-moduuli, jonka avulla voidaan helposti seurata esimerkiksi prosessorin kuormitusta tai muistinkäyttöä. (Rodola 2016.)

Valmiin lähdekoodin pohjalta oli helppo rakentaa sovellus, joka yhdistää yhdyskäytävään, tunnistautuu ja alkaa lähettää haluttua dataa. Sovellus toimi lähes suoraan sellaisenaan, vain muutamia kohtia jouduttiin muuttamaan. Kun prosessorin kuormitusta saatiin lähetettyä onnistuneesti, todettiin, että WSN OpenAPI Gateway toimii hyvin ja lähdekoodin pohjalta olisi kätevä alkaa rakentaa sensoria, joka lähettäisi haluttua dataa yhdyskäytävälle. Ohjelmaa suoritettaessa sille annettiin komentoriviargumentteina halutut asetukset, kuten kohdeosoite, käyttäjätunnus ja salasana sekä kohdeosoitteen portti. (Liite 2.)

Datan analysointiin projektissa koekäytettiin OpenAPI WSN Gatewayn mukana tullutta työkalua, jonka avulla oli mahdollista esittää halutut tiedot graafisessa muodossa. Esimerkiksi jos haluttiin seurata valosensorilla huoneen valoisuutta reaaliajassa, voitiin sensorin arvot piirtää graafiseen kuvaajaan. Esitetyssä grafiikassa pystyakselilla on automaattisesti skaalautuva valon määrän mittari ja vaaka-akselilla aika. Kuvasta 24 voidaan

havaita, kuinka valojen sytyttäminen ja kädellä sensorin peittäminen vaikuttavat kuvaajaan, ja näin voidaan todeta, että valittu yhdyskäytävä eli WSN OpenAPI Gateway, ja sensoriohjelma toimivat hyvin yhdessä.



Kuva 24. WSN OpenAPI Gatewayn mukana tulleen työkalun, WSN Monitorin, piirtämää kuvaajaa sen vastaanottamasta datasta Android-sovellukselta. Kuvaajassa on piirtynyt sovelluksen lähettämää arvoa puhelimen valoanturista.

6 Yhteenveto

Opinnäytetyö oli osa Metropolia Ammattikorkeakoulun laajaa projektia. Työssä tarkasteltiin asioiden internetin tilaa, sen toimintaa ja tulevaisuutta. Kaiken löydetyn teoriatiedon ja käytännön havaintojen pohjalta voidaan todeta, että asioiden internetin tulevaisuus näyttää valoisalta ja että erilaisia toteutuksia ilmestyy markkinoille suurten yritysten ja erilaisten innovaatioiden tuomina. Asioiden internet on lyömässä läpi niin kuluttajamarkkinoilla kuin teollisella puolella. Sen mukanaan tuomat ansaintamahdollisuudet kiinnostavat suuria kansainvälisiä yrityksiä, kuten Cisco, Microsoft, Google ja Intel, jotka haluavat saada osansa oletetuista biljoonien dollarien markkinoista.

Asioiden internetin kenttään on alkanut ilmestyä ryhmittymiä kuten AllSeen Alliance, joka Linux Foundationin johdolla yrittää luoda alalle yleisiä toimivia toteutusmalleja ja standardeja. AllSeen Alliance on myös julkaissut avoimen lähdekoodin yhdyskäytävän, jota voidaan käyttää asioiden internetin toteutuksissa. Kentän pirstaleisuuden vuoksi toteutukset tuntuvat olevan erittäin spesifisiä, esimerkiksi SCA11H-sensorin on tarkoitettu toimivan vain WSN OpenAPI Gatewayn kanssa. Mikäli tarve vaatii, toiseen yhdyskäytävään ei voida helposti vaihtaa tai yhdistää kokonaisuutta muihin toteutuksiin. Asioiden internetistä on paljon tietoa, mutta suurin osa siitä on yleistä hypetystä ja myyntipuhetta tosiasioiden ja toimivien totutusten jäädessä taka-alalle, joten lähdekritiikki on hyvä pitää mielessä.

Insinöörityössä tutkittiin olisiko Muratan SCA11H-sensori ja Tampereen yliopiston avoimen lähdekoodin WSN OpenAPI Gateway mahdollista saada toimimaan yhdessä. Työssä tutkittiin, kuinka WSN OpenAPI Gatewayhin yhdistäminen on tarkoitettu toteutettavan ja kuinka SCA11H vastaisi näitä vaatimuksia. Työssä kävi ilmi, että vaikka sensorin teknisen spesifikaation perusteella sensorin olisi tarkoitus toimia yhdessä valitun yhdyskäytävän kanssa, ei se todennuksien erilaisuuden vuoksi kuitenkaan suostunut yhdistämään yhdyskäytävään. Tämän vuoksi työssä haluttiin lähteä selvittämään, löytyisikö kuitenkin jokin ratkaisu, jonka kanssa WSN OpenAPI Gateway ylipäänsä toimisi. Testiratkaisuksi tehtiin Raspberry Pillä toimiva sovellus, joka todentaa itsensä WSN OpenAPI Gatewayn kanssa ja toimittaa sille onnistuneesti haluttua dataa. Testidataksi valittiin Raspberry Pin prosessorin kuormitus prosentteina. Kaiken kaikkiaan voidaan sanoa, että WSN OpenAPI Gateway voisi olla toimiva kokonaisuus, mikäli käytettävä sensori olisi toteutettu sen tukemalla tavalla.

Lähteet

A new step towards the “smart-water”optimisation strategy. 2016. Verkkodokumentti. Barcelona City Council. <http://smartcity.bcn.cat/en/telemanaging-irrigation.html> Luettu 13.1.2016.

An Overview of XMPP. 2016. Verkkodokumentti. XMPP. <http://xmpp.org/about/technology-overview.html> Luettu 7.3.2016.

Android Node. 2016. Verkkodokumentti. Tampere University of Technology. <http://www.tkt.cs.tut.fi/research/gwg/android_client.html> Luettu 10.3.2016.

Architecture. 2016. Verkkodokumentti. Tampere University of Technology. <http://www.tkt.cs.tut.fi/research/gwg/architecture.html> Luettu 10.3.2016.

Bort, Julie. 2016. Cisco is buying Internet of Things company Jasper Technologies for $1.4 billion. Verkkodokumentti. Business Insider Inc. <http://uk.businessinsider.com/cisco-buys-jasper-for-14-billion-2016-2?r=US&IR=T> Luettu 3.2.2016.

Cisco pushes IoT analytics to the extreme edge with mist computing. 2014. Verkkodokumentti. Rethink Internet of Things. <http://rethink-iot.com/2014/12/19/cisco-pushes-iot-analytics-extreme-edge-mist-computing-2/> Luettu 10.3.2016.

Drozhzhin, Alex. 2015. Black Hat USA 2015: The full story of how that Jeep was hacked. Verkkodokumentti. AO Kaspersky Lab. <https://blog.kaspersky.com/blackhat-jeep-cherokee-hack-explained/9493/> Luettu 15.2.2016.

End-to-End Security for Industrial Automation. 2016. Verkkodokumentti. Intel. <http://www.intel.com/content/www/us/en/industrial-automation/topics-and-trends/security/overview.html> Luettu 23.3.2016.

Equip Hospitals for the Future with Samsung. 2016. Verkkodokumentti. Samsung. <http://www.samsung.com/global/microsite/cebit2015/equip-hospitals-for-the-future-with-samsung.html> Luettu 1.2.2016.

Features. 2016. Verkkodokumentti. Tampere University of Technology. <http://www.tkt.cs.tut.fi/research/gwg/features.html> Luettu 18.1.2016.

Fingerprints to be included in new passports as from 29 June. 2009. Verkkodokumentti. Suomen suurlähetystyö ja pääkonsulaatit: Washington, New York ja Los Angeles. <http://www.finland.org/Public/default.aspx?contentid=166960&nodeid=35831&culture=en-US> Luettu 10.3.2016.

Friess Peter. & Vermesan Ovidiu. 2016. Internet of Things – From Research and Innovation to Market Deployment. Verkkodokumentti. River Publishers. <http://www.internet-of-things-research.eu/pdf/IERC_Cluster_Book_2014_Ch.3_SRIA_WEB.pdf> Luettu 15.3.2016.

Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013. 2014. Verkkodokumentti. Gartner. <http://www.gartner.com/newsroom/id/2665715> Luettu 10.3.2016.

Gartner's 2015 Hype Cycle for Emerging Technologies Identifies the Computing Innovations That Organizations Should Monitor. 2015. Verkkodokumentti. Gartner. <http://www.gartner.com/newsroom/id/3114217> Luettu 10.3.2016.

Gateway Agent Project. 2016. Verkkodokumentti. Allseen Alliance. <https://wiki.allseenalliance.org/gateway/gatewayagent> Luettu 10.3.2016.

Generic WSN Gateway Installation. 2016. Verkkodokumentti. Tampere University of Technology. <http://www.tkt.cs.tut.fi/research/gwg/documentation.html> Luettu 14.1.2016.

Google Trends. 2016. Verkkodokumentti. Google. <https://www.google.com/trends/> Luettu 25.3.2016.

Google Visualization API Reference. 2016. Verkkodokumentti. Google. <https://developers.google.com/chart/interactive/docs/reference#drawtoolbar> Luettu 8.2.2016.

Higginbotham, Stacey. 2016. The 6 Things CES Taught Us About The Internet of Things. Verkkodokumentti. Fortune. <http://fortune.com/2016/01/11/ces-internet-of-things/> Luettu 10.3.2016.

Hirstov, Victor. 2016. Apple Pay vs Samsung Pay vs Android Pay: comparison. Verkkodokumentti. PhoneArena. <http://www.phonearena.com/news/Apple-Pay-vs-Samsung-Pay-vs-Android-Pay-comparison_id77632> Luettu 10.3.2016.

Hypertext Transfer Protocol (HTTP) Status Code Registry. 2016. Verkkodokumentti. The Internet Assigned Numbers Authority. <http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml> Luettu 1.3.2016.

Intel® IoT Gateway Development Kits. 2016. Verkkodokumentti. Intel. <http://www.intel.com/content/www/us/en/embedded/solutions/iot-gateway/development-kits.html> Luettu 11.2.2016.

Kristoffer, Joseph. 2015. A Gift for Every Pebbler: Introducing Pebble Health and Firmware 3.8. Verkkodokumentti. Pebble. <https://blog.getpebble.com/2015/12/15/health/> Luettu 1.3.2016.

Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä. 2007. 159/9.2.2007.

Learn about Windows 10 IoT Core. 2016. Verkkodokumentti. Microsoft. <http://msiot.github.io/content/en-US/IoTCore.htm> Luettu 18.3.2016.

Lesser, Adam. 2015. Declining sensor costs open up new consumer applications. Verkkodokumentti. Gigaom. <https://gigaom.com/2015/01/25/declining-sensor-costs-open-up-new-consumer-applications/> Luettu 14.1.2016.

Lähimaksaminen. 2016. Verkkodokumentti. Korttiturvallisuus. <https://www.korttiturvallisuus.fi/Kaupassa/Lahimaksaminen/> Luettu 15.3.2016.

M2M/IoT Sector Map. 2016. Verkkodokumentti. Beecham Research. <http://www.beechamresearch.com/article.aspx?id=4> Luettu 14.4.2016.

Manual Configuration of GWG. 2016. Verkkodokumentti. Tampere University of Technology. <http://www.tkt.cs.tut.fi/research/gwg/configuration.html> Luettu 14.1.2016.

McLellan, Charles. 2015. The internet of things and big data: Unlocking the power. Verkkodokumentti. Zdnet. <http://www.zdnet.com/article/the-internet-of-things-and-big-data-unlocking-the-power/> Luettu 2.3.2016.

Mearian, Lucas. 2015. Office complex implants RFID chips in employees' hands. Verkkodokumentti. <http://www.computerworld.com/article/2881178/office-complex-implants-rfid-chips-in-employees-hands.html> Luettu 6.2.2016.

Mikrosiru. 2016. Verkkodokumentti. Kennelliitto. <http://www.kennelliitto.fi/kasvatus-ja-terveys/tunnistusmerkinta/mikrosiru> Luettu 1.3.2016.

MQTT. 2016. Verkkodokumentti. MQTT. <http://mqtt.org/> Luettu 17.3.2016.

Otis, Brian & Parviz, Babak. 2014. Introducing our smart contact lens project. Verkkodokumentti. Google. <https://googleblog.blogspot.fi/2014/01/introducing-our-smart-contact-lens.html> Luettu 16.1.2016.

Parkurst, William R. 2004. Internet Addressing and Routing First Step. Verkkodokumentti. Cisco. <http://www.ciscopress.com/articles/article.asp?p=348253&seqNum=7> Luettu 10.3.2016.

Press, Gil. 2014. It's Official: The Internet Of Things Takes Over Big Data As The Most Hyped. Verkkodokumentti. Forbes. <http://www.forbes.com/sites/gilpress/2014/08/18/its-official-the-internet-of-things-takes-over-big-data-as-the-most-hyped-technology/#630956961aaa> Luettu 17.3.2016.

Product Datasheet. 2015. Verkkodokumentti. Murata. <http://www.murata.com/~/media/webrenewal/products/sensor/accel/sca10h_11h/product%20specification%201323%20rev1%20sca11h%20product%20datasheet%20eng.ashx?la=en-us> Luettu 10.3.2016.

Protect against harmful apps. 2016. Verkkodokumentti. Google. <https://support.google.com/nexus/answer/2812853?hl=en> Luettu 10.3.2016.

Raspbian Installer. 2016. Verkkodokumentti. Raspbian. <https://www.raspbian.org/RaspbianInstaller> Luettu 10.3.2016.

SCA11H cloud server interface specification. 2015. Verkkodokumentti. Murata. <http://www.murata.com/~/media/webrenewal/products/sensor/accel/sca10h_11h/product%20specification%201325%20rev1%20sca11h%20cloud%20server%20interface%20specification%20eng.ashx?la=en-us> Luettu 14.1.2016.

Schneider, Stan. 2013. Understanding The Protocols Behind The Internet Of Things. Verkkodokumentti. Electronic Design. <http://electronicdesign.com/iot/understanding-protocols-behind-internet-things> Luettu 10.3.2016.

SimpleHTTPServer – Simple HTTP request handler. 2016. Verkkodokumentti. Python Software Foundation. <https://docs.python.org/2/library/simplehttpserver.html> Luettu 18.3.2016.

Sun, Leo. 2016. Internet of Things in 2016: 6 Stats Everyone Should Know. Verkkodokumentti. <http://www.fool.com/investing/general/2016/01/18/internet-of-things-in-2016-6-stats-everyone-should.aspx> Luettu 10.3.2016.

Sullivan, Nick. How the NSA (may have) put a backdoor in RSA's cryptography: A technical primer. 2014. Verkkodokumentti. Ars Technica. <http://arstechnica.com/security/2014/01/how-the-nsa-may-have-put-a-backdoor-in-rsas-cryptography-a-technical-primer/> Luettu 15.3.2016.

Rodola, Giampaolo. 2016. A cross-platform process and system utilities module for Python. Verkkodokumentti. Github. <https://github.com/giampaolo/psutil> Luettu 18.3.2016.

The innovative companies that support AllJoyn. 2016. Verkkodokumentti. Allseen Alliance. <https://allseenalliance.org/alliance/members> Luettu 21.3.2016.

The Internet of Things: The Future of Consumer Adoption. 2015. Verkkodokumentti. Accenture. <https://www.accenture.com/t20150624T211456__w__/us-en/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Technology_9/Accenture-Internet-Things.pdf> Luettu 10.3.2016.

The rain in Spain claims four lives and causes chaos across south coast. 2015. Verkkodokumentti. The Local. <http://www.thelocal.es/20150907/alert-as-southern-spain-battles-severe-flooding> Luettu 10.3.2016.

Virtual Patient Observation: Centralize Monitoring of High-Risk Patients with Video. 2016. Verkkodokumentti. Cisco. <http://www.cisco.com/c/en/us/products/collateral/physical-security/video-surveillance-manager/white_paper_C11-715263.html> Luettu 13.1.2016.

Wakefield, Jane. 2014. Smart LED light bulbs leak wi-fi passwords. Verkkodokumentti. BBC. <http://www.bbc.com/news/technology-28208905> Luettu 10.3.2016.

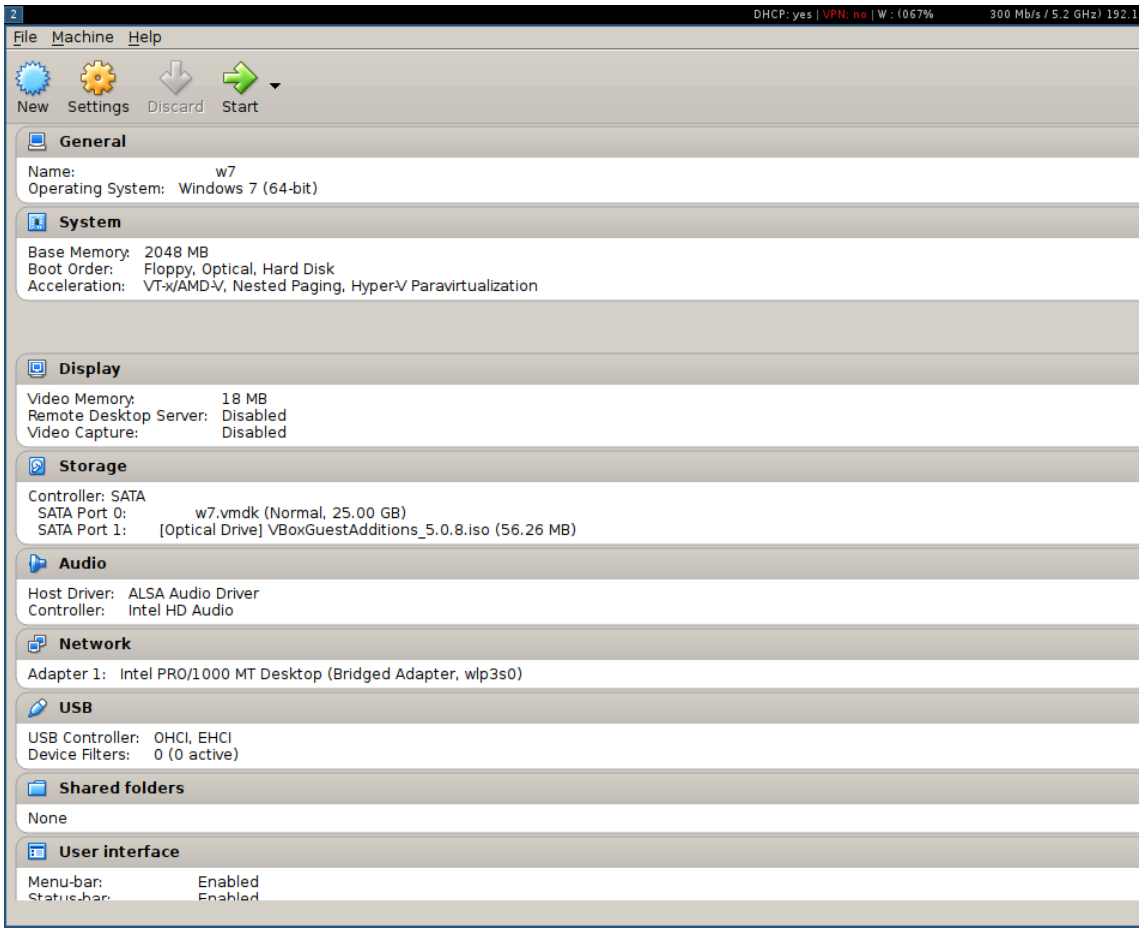
What is DDS? 2016. Verkkodokumentti. Object Management Group. <http://portals.omg.org/dds/what-is-dds-3/> Luettu 21.3.2016.

WSN OpenAPI Technical Specification. 2013. Verkkodokumentti. Tampere University of Technology. <http://www.tkt.cs.tut.fi/research/gwg/downloads/WSN_OpenAPI_Specification_r1.0.pdf> Luettu 15.3.2016.

WSN OpenAPI XML Introduction. 2013. Verkkodokumentti. Tampere University of Technology. <http://www.tkt.cs.tut.fi/research/gwg/downloads/WSN_OpenAPI_Brief.pdf> Luettu 15.3.2016.

Virtuaalikoneen asetukset

Virtualisointialustana käytettiin VirtualBoxia. OpenAPI WSN - virtuaalikoneelle allokoitiin 2 gigatavua muistia ja riittävästi kiintolevytilaa. Verkkokortti sillattiin, jotta käyttöjärjestelmälle saatiin asetettua käyttöjärjestelmän asetuksista staattinen IP-osoite. Käytetty käyttöjärjestelmä oli Windows 7.



Esimerkkisovelluksen lähdekoodi

Lähdekoodi, jolla Raspberry Pi ohjelmoitiin yhdistämään OpenAPI WSN Gatewayhyn. Alkuperäinen koodi löytyy OpenAPI WNS Gatewayn sivuilta.

```
# template for ACF message
xml_msg_acf = Template("""\
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<AuthenticationRequest xmlns="urn:ietf:params:xml:ns:acf" messageId="1" method="password" responseFormat="XML"
version="1.0">
<Parameter name="username" value="$username"/>
<Parameter name="password" value="$password"/>
</AuthenticationRequest>""")

# template for SIDF measurement message
xml_msg_sidf_start = Template("""\
<?xml version="1.0" encoding="UTF-8" ?>
<SIDF version="1.2" xmlns="urn:ietf:params:xml:ns:sidf">
<Network id="$network">
<Node id="$node" name="$nodename">
<Sensor id="$sensor" >
<Measurement quantity="$quantity" unit="$unit" time="$time" >""")
xml_msg_component=Template("""<Component id="$id">$value</Component>""")
xml_msg_end="</Measurement></Sensor></Node></Network></SIDF>"

def send_xml(sensor, quantity, unit, components):
    info = {"network":network,
    "node":node,
    "nodename":nodename,
    "sensor":sensor,
    "time":strftime("%Y-%m-%dT%H:%M:%S+00:00", gmtime()),
    "quantity":quantity,
    "unit":unit
    }
    print xml_msg_sidf_start.substitute(info)
    s.send(xml_msg_sidf_start.substitute(info))
    for c in components:
        print xml_msg_component.substitute({"id" : c, "value" : components[c]})
        s.send(xml_msg_component.substitute({"id" : c, "value" : components[c]}))
    print xml_msg_end
    s.send(xml_msg_end)
    s.send("\n")
    print "\n\n"

def send_cpu_load():
    send_xml("CPU load", "Activity monitor", "", {"percent" : psutil.cpu_percent(interval=1)})

if __name__ == '__main__':
    if len(sys.argv) < 7:
        print "usage: host port user password interval networkId nodeId (nodeName)"
    else:

        # Connect to the server
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((sys.argv[1], int(sys.argv[2]))) # host, port

        # Authentication
        s.send(xml_msg_acf.substitute({"username": sys.argv[3], "password": sys.argv[4]}))
        s.send("\n")

        network = sys.argv[6]
        node = sys.argv[7]
        try:
            nodename = sys.argv[8]
        except:
            nodename = ''

        # Update sensor values
        try:
            while (1):
                send_cpu_load()
                sleep(int(sys.argv[5]))
        except:
            s.close()
            raise
```