

Google Cloud Platform IoT ratkaisuiissa

Elmeri Kinnunen

Tekijä(t) Elmeri Kinnunen	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Raportin/Opinnäytetyön nimi Google Cloud Platform IoT ratkaisuihin	Sivu- ja liitesivumäärä 35 + 0
<p>Tämän opinnäytetyön tavoitteena oli tutkia mitä on IoT eli esineiden internet ja miten Google Cloud Platform toimii IoT-laitteiden pilvipalveluna. Tarkoituksena oli myös selvittää mitä eri tekniikoita ja protokollia tarvitaan laitteen yhdistämiseksi Googlen pilvipalveluun. Työn teoriaosuuden tutkimuksen lisäksi aihetta tutkittiin toiminnallisessa tutkimuksessa.</p> <p>Teoriaosuudessa tutkittiin ja esiteltiin esineiden internetiä sekä sen sisältämiä tekniikoita ja protokollia. IoT:n teoria tutkimuksen perusteella syvennyttiin Google Cloud alustaan esineiden internetin näkökulmasta keskittyen olennaisiin ominaisuuksiin tämän tutkimuksen kannalta. Teoriaosuuden tavoitteena oli myös antaa selkeä ymmärrys toiminnallisessa osuudessa käytettävistä tekniikoista.</p> <p>Toiminnallisen osuuden tutkimuksessa rakennettiin Raspberry Pi älylämpömittari, joka kytkettiin Googlen Cloud Platform alustan sisältämään Cloud IoT palveluun. Yhdistämistä ja laitteen lähettämän datan käsittelyprosessia tutkittiin yleisestä IoT-laitteiden näkökulmasta, jotta tutkimuksesta olisi hyötyä myös muiden IoT-laitteiden ja Googlen pilvipalvelun välisessä käytössä. Toiminnallisella osuudella pyrittiin tukemaan teoriaosuudessa esiteltyjä asioita ja antamaan tarvittavaa tietotaitoa, jotta lukija osaisi yhdistää IoT-laitteen ja Google Cloud Platform alustan.</p>	
Asiasanat IoT, Raspberry Pi, Google, pilvi	

Author(t) Elmeri Kinnunen	
Degree Programme Business Information Technology	
Raport/thesis name Google Cloud Platform in IoT solutions	Number of pages and appendix pages 35 + 0
<p>The goal of this thesis was to study what is the Internet of Things (IoT) and how the Google Cloud Platform works as a cloud service for IoT Devices. The purpose was also to research what different technologies and protocols are needed to connect the IoT device to Google's cloud service. In addition to the research of the theoretical part, the thesis includes functional part.</p> <p>The theoretical part was studied and presented the IoT, as well as the techniques and protocols it contains. The IoT theory research was deepened into the Google Cloud Platform from the perspective of the Internet of Things and focused on the essential features for this thesis. The aim of the theoretical part was also to give a clear understanding of the techniques used in the functional part.</p> <p>In the functional part, a smart thermometer was built from Raspberry Pi which was connected to the Google Cloud Platform's Cloud IoT service. The Connection and data handling processes were presented from the general perspective of IoT so that the research would also be useful for use between all types of IoT devices and Google's cloud service. The goal of the functional part was to demonstrate the presented theoretical part techniques in the practice and provide the necessary know-how to enable the reader to connect the IoT device to the Google Cloud Platform.</p>	
Keywords IoT, Raspberry Pi, Google, cloud	

Sisällys

1	Johdanto	1
	Käsitteet	2
2	IoT	4
2.1	Esineiden Internet	4
2.2	Standardit ja protokollat	5
2.3	MQTT	7
2.4	HTTP	8
2.5	Tietoturva	9
3	Google Cloud Platform	12
3.1	Google pilvialustana	12
3.2	IoT Core	13
3.3	Cloud Pub/Sub	14
3.4	Dataflow	15
3.5	Cloud Functions	15
3.6	BigQuery	15
3.7	Tietoturva	16
4	Raspberry Pi	17
4.1	Ominaisuudet	17
4.2	Raspberry Pi IoT-alustana	18
5	Raspberry Pi älylämpömittariksi	19
5.1	Google Cloud Platform käyttöönotto	19
5.2	Ulkolämpömittarin rakentaminen	21
5.3	Laitteen käyttöönotto	23
5.4	Lämpömittarin järjestelmän rakentaminen	25
5.5	Laitteen havainnointi Googlen alustoilta	28
6	Pohdinta	30
7	Lähteet	33

1 Johdanto

Internettiin kytkettävä esine tunnetaan ilmiönä Internet Of Things (IoT) eli esineiden internet. Nykyisin yhä useammat elektroniset laitteet kytketään internetiin, sillä ihmiset haluavat tehdä päivittäisistä tehtävistä helpompia tai tehokkaampia ja yritykset haluavat kerätä dataa. Dataliikenne kulkee yleensä pilven kautta päätelaitteelle ja siksi pilviratkaisu on merkittävä osa IoT-kokonaisuuden rakennetta.

Opinnäytetyöni teoriaosuudessa selvitetään mitä ovat IoT-laitteet ja mitä eri teknologioita sekä riskejä niihin liittyy. IoT-tutkimuksen jälkeen tutkitaan Google Cloud Platform alustan käyttöä IoT-laitteiden kanssa sekä selvittää IoT-laitteiden ja Googlen pilvialustan yhdistämiseen käytettäviä tekniikoita ja eri protokollia. Tämän opinnäytetyön tavoitteena on tehdä kattava selvitys aiheesta ja tarjota lukijalle tarvittavaa tietoa laitteiden yhdistämiseen ja hallinnointiin Google Cloud Platformin avulla.

Toiminallisessa osuudessa tehdään yhden piirilevyn Raspberry Pi tietokoneesta ulkolämpömittari tarvittavien komponenttien avulla ja se tullaan yhdistämään Googlen pilvialustaan. Tutkimuksessa käydään kaikki vaiheet läpi Raspberry Pi:n asennuksesta ja Python ohjelmoinnista pilvipalvelun käyttöön asti, mutta pääsääntöisesti työssä kuitenkin keskitytään yhdistämisvaiheeseen sekä käyttöönottoon.

Toiminallisen osuuden tutkimuksen on tarkoitus selvittää Raspberry Pi piirilevyn ja Google Cloud Platform alustan yhdistäminen sekä siihen vaadittavat asiat. Eri vaiheiden haasteita ja mahdollisuuksia tarkastellaan omien kokemusten pohjalta ja niistä koostetaan analyysi. Tutkimuksessa kartoitetaan myös eri toiminnallisuudet ja vaatimukset, jotta tutkimuksesta olisi hyötyä useampien IoT-laitteiden yhdistämisessä Googlen pilvialustalle.

Tärkeimmät tutkimuskysymykset:

1. Mikä on Google Cloud Platform?
2. Mitä toiminnallisuuksia Googlen Cloud Platform palvelussa on IoT-laitteille?
3. Mitä vaaditaan Raspberry Pi laitteen ja Googlen pilvialustan yhdistämiseen?

Käsitteet

Asynkroninen kommunikaatio

Reaaliaikaisen kommunikoinnin vastakohta. Lähettäjän ja vastaanottajan välinen kommunikointi ei ole riippuvainen kummastakaan osapuolesta.

Big Data

Massadata. Suuri ja jatkuvasti kasvava tietomäärä.

Bottiverkko

Usean ohjelmistorobotin muodostama verkko.

Client-Server

Tietoliikenne arkkitehtuuri malli, jossa ennalta määritetyn yhteyden sijaan kommunikointi tapahtuu asiakkaan ottaessa yhteyttä palvelimeen.

DDOS

Distributed Denial of Service. Hajautettu palvelunestohyökkäys

DNS

Domain Name Server eli nimipalvelin. Muuntaa verkkotunnukset IP-osoitteiksi.

GPIO

General-purpose input/output. Monikäyttöinen liitäntäportti elektroniikkapiireissä.

GUI

Graphical User Interface. Graafinen käyttöliittymä.

JSON

JavaScript Object Notation. Tiedon välitykseen käytettävä yleinen tiedostomuoto.

JWT

JSON Web Token. Käyttöoikeuksien varmentamiseen käytettävä JSON-pohjainen menetelmä.

PSK

Pre-Shared Key. Langattoman lähiverkon salasana.

PuTTY

Avoimenlähdekoodin SSH ja telnet komentorivi emulaattori.

RS256

Julkisen ja yksityisen avainparin epäsymmetrinen salausalgoritmi menetelmä.

SDK

Software development kit. Paketti ohjelmistokehityksen työkaluja.

SSH

Secure Shell. Salattu tietoliikenne protokolla. Yleisesti käytetty laitteiden väliseen etäyh-teyteen.

TLS/SSL

IP-verkkojen salausprotokolla. Tunnetaan nykyisin Transport Layer Security. Ennen Se-cure Sockets Layer.

Wireshark

Tietoliikenteen analysointiin käytettävä tietokoneohjelma.

WLAN

Wireless local area network. Langaton lähiverkko.

2 IoT

IoT on lyhenne englanninkielisestä käsitteestä Internet of Things eli suomennettuna esineiden internet. IoT-laite voi olla mikä tahansa fyysinen esine, joka on kytketty internettiin ja sitä voidaan seurata ja ohjata etänä verkon kautta. Yleisenä määrityksenä voidaan sanoa, että esineellä tulee olla internet yhteys sekä joku toiminnallisuus. Yleensä laitteessa on myös sensoreita, joilla ympäristöä voidaan havaita. (Tietotekniikan termitalkoot 2017)

2.1 Esineiden Internet

Esineiden internet käsitteellä ei ole yhtä ja oikeaa määritelmää, mutta käytännössä kaikki internettiin yhteydessä olevat laitteet, joilla on yhtenä komponenttina IP:n omaava tietokone, ovat IoT-laitteita. Esineen ei toisaalta tarvitse olla itse kytkettynä internettiin, kunhan se voidaan tunnistaa jonkun yksilöllisen tunnisteiden perusteella välillisesti toisen laitteen avulla. Tällainen esine voi olla esimerkiksi postipaketti. (Tietotekniikan termitalkoot 2017)

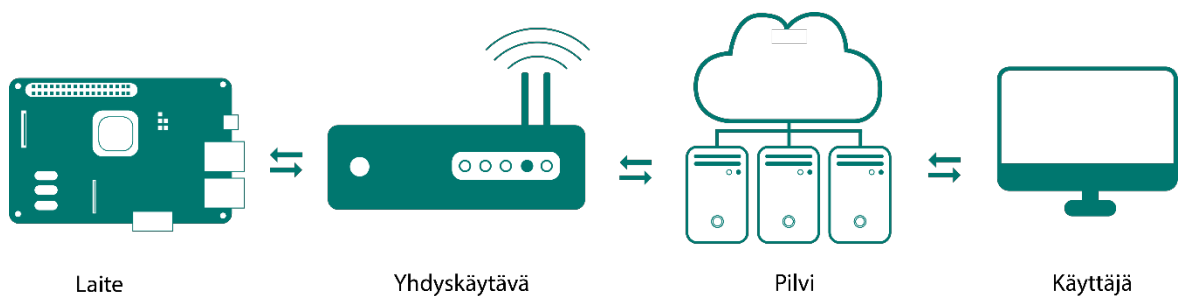
IoT-laitteiden määrä on jatkuvasti noususuhdanteessa ja yhä useammat uudet laitteet siirtyvät internettiin, vaikka suoranaista tarvetta sille ei olisikaan. Myös vanhoja laitteita yhdistetään lisälaitteiden avulla internettiin.

Teollisuudessa vastaavaa tekniikkaa kutsutaan teolliseksi internetiksi ja sen avulla yritykset voivat kerätä dataa laitteistaan ja havaita laitteen sisä- tai ulkopuolella olevia tapahtumia sensoreiden avulla. Kerätyn tiedon avulla voidaan tehostaa prosesseja ja säästää kustannuksissa. Laite voi esimerkiksi ilmoittaa ennakoivasti huoltotarpeesta tai viasta, jolloin se voidaan huoltaa ilman vaikutusta tuotantoon. Näin välttyään myös turhilta korjauksilta sekä säästetään kustannuksia. Tiedon jatkuva kerääminen myös vähentää tai poistaa manuaalisen raportoinnin tarpeen, koska laite lähettää kaikki tapahtumat tietokantaan, josta voidaan koostaa haluttu analyysi. Kertynyttä dataa voidaan myös hyödyntää laitteen toimivuuden tehostamiseksi. (Iotfinland.net 2018)

Rakennusteollisuudessa IoT on otettu käyttöön varsinkin betonitöiden osalta. Useat rakennusyrietykset ovat kehittäneet omia järjestelmiä ja laitteita betonin kosteuden seurantaan. Lujas SmartConcrete anturijärjestelmä on Lujabetonin kehittämä älykäs betonin mittaustajärjestelmä. Sen anturit keräävät tietoa kosteudesta ja tallentavat kaiken datan reaaliajassa pilvipalveluun. Järjestelmän avulla voidaan tarkasti arvioida, milloin betoni on tarpeeksi kuivaa seuraavan työvaiheen aloittamiseksi. Betonin mittausjärjestelmät säästävät turhilta mittauskustannuksilta ja rakennusvirheilta sekä tekevät aikataulusta tehokkaamman. (Lujabetoni 2018)

Kotiautomaatio on kuluttajakäytössä yleinen IoT-muoto, joka on tehnyt vahvasti tuloaan ja yhä useammasta kodista löytyy nykyisin älyvalaisimia, sensoreita tai jokin älykodinkone. Kuluttajille suunnattujen laitteiden tarkoituksena on tehdä ihmisten elämästä helpompaa ja samalla säästää rahaa tai aikaa. Esimerkiksi älyvalaisimet voidaan ohjata automaattisesti sammumaan huoneesta, jossa ei ole ketään ihmistä, jolloin säästyy sähköä ja ihmisen ei tarvitse itse painaa katkaisijasta. Kodissa olevat anturit voivat myös suojata vesivahingoilta tai vastaavilta kustannuksilta ja tuhoa aiheuttavilta katastrofeilta.

IoT-ympäristö muodostuu yleensä ainakin neljästä tarvittavasta tasosta, jotka näkyvät kuvassa 1. Ympäristö koostuu laitteesta, yhdyskäytävästä, pilvialustasta sekä loppukäyttäjistä. Laitteen ja pilven välillä on fyysinen yhdyskäytävä laite ja sen tarkoitus mahdollistaa verkkoliikenne niiden välillä. Usein yhdyskäytävä myös suodattaa tai muuntaa dataa ymmärrettävään muotoon, jota voi tarkastella esimerkiksi yhdyskäytävän järjestelmänvalvoja. Dataa saatetaan prosessoida myös yhdyskäytävän ja pilven välillä, mutta nykyisin monet toiminnot ovat siirtymässä enemmän pilvialustoille.



Kuva 1. yksinkertainen IoT-infrastruktuuuri

2.2 Standardit ja protokollat

Esineiden internet pitää sisällään valtavan määrän eri protokollia ja standardeja, joista yleisimmät on esiteltynä tässä työssä. Tarkemmin esitellään työn kannalta oleelliset protokollat sekä toiminnallisissa osuudessa käytettävät protokollat.

WLAN eli Langaton lähiverkko käyttää yleensä standardia IEEE 802.11 ja se kattaa IoT-laitteiden OSI-mallissa fyysisen kerroksen sekä siirtoyhteyserroksen muodostaman peruserroksen. WLAN käyttää internet yhteyden muodostamiseen 2,4 MHz, 5 MHz tai tulevaisuudessa 6 MHz taajuutta. Matalammalla taajuudella kantama on pidempi kuin korkeammalla taajuudella, mutta korkeammalla taajuudella siirtonopeus on nopeampi. WLAN taajuuksilla on useita kanavia, joita pitkin lähetin tai reititin voi siirtää dataa ja 6MHz taajuus tarjoaakin lisää kanavia, joiden turvin voidaan vähentää yhteysongelmia ruuhkautu-

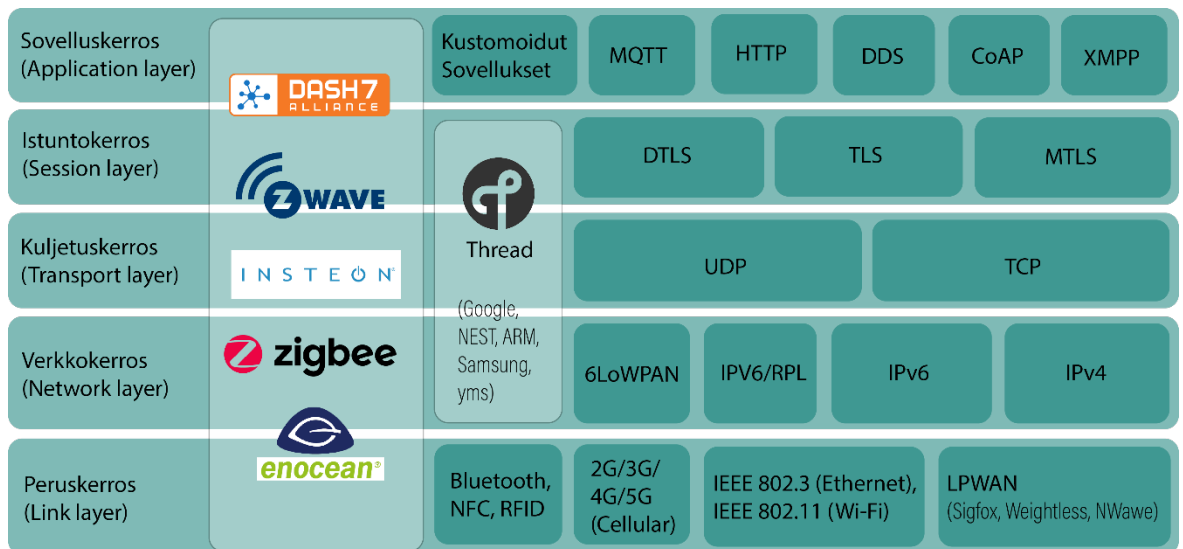
neilla kanavilla. Etenkin ympäristöissä, joissa on useita langattomia yhteyksiä, laitteet eivät välttämättä löydä vapaata kanavaa, ja sen vuoksi ne joutuvat käyttämään samaa taajuutta muiden WLAN verkkojen kanssa. (Fleishman 2020, luku 4 & 6) IoT-laitteiden yhteyksissä käytetään eniten WLAN tekniikkaa sen datan siirtonopeuden ja halvan hinnan vuoksi. Samaan siirtonopeuteen yltää myös cellular eli matkapuhelinverkko sekä Ethernet. Matkapuhelinverkko on kuitenkin kalliimpi ja Ethernet yhteydessä usein kaapelit vaikeuttavat IoT-laitteen sijoittamista haluttuun paikkaan.

IP eli Internet Protocol on verkkokerroksen protokolla, joka varmistaa TCP/IP-pakettien kuljettamisen palvelimien ja käyttäjien välillä. Paketissa oleva IP-otsake sisältää kohdeosoitteen, jonka lisäksi paketissa on myös tieto portista ja sovelluskerroksen dataa. (Kavelová & Dostálek 2006, luku 1) IP-osoite on numerosarja, joka yksilöi ja määrittää internetissä olevien laitteiden sekä palveluiden osoitteet. Ulkoverkkoon näkyy yksi osoite ja sisäverkossa oleville laitteille on omat osoiteavaruudet, jotka eivät näy ulkoverkkoon. (Kavelová & Dostálek 2006, luku 6) IPv4 osoite sisältää vain 4 tavua ja verkossa olevien laitteiden määrä on jatkuvassa kasvussa, jolloin osoitteet uhkaavat loppua kesken. Tästä syystä on kehitetty 16 tavuinen IPv6, joka keventää myös IP-otsakkeen rakennetta muuttamalla usein turhat ja pakolliset kentät vapaaehtoisiksi. (Kavelová & Dostálek 2006, luku 8) Sisäverkossa olevan IoT-laitteen kanssa voidaan keskustella suoraan ulkoverkosta määrittämällä laitteelle verkon reitittimeen avoin portti, josta verkkoliikenne pääsee kulkemaan.

TCP (Transmission Control Protocol) on UDP-protokollan lisäksi toinen yleisimmin käytetyistä kuljetuskerroksen protokollista, joka tekee saumatonta yhteistyötä verkkokerroksen IP-protokollan kanssa. TCP huolehtii datan eheän siirtämisen lähettäjän ja vastaanottajan välillä muodostamalla yhteyden kolmitiekättelyllä, jossa yhteyden avaaja lähettää SYN-paketin vastaanottajalle ilmoittaakseen halutusta yhteyden avaamisesta. Vastaanottaja ilmoittaa, että on vastaanottanut pyynnön ja palauttaa SYN/ACK-paketin lähettäjälle. Yhteyden avaaja vastaa vielä lopuksi vastaanottajalle ACK-paketilla kuitatakseen SYN/ACK-paketin saapuneen. TCP-segmentissä eli kehyksessä on IP otsake, joka sisältää kohdeosoitteen ja TCP otsake, joka sisältää kohdeportin. Kehyksessä on myös jokaisen lähetetyn paketin järjestysnumero, jolla TCP huolehtii pakettien oikean saapumisjärjestyksen ja varman siirtoyhteyden. Mikäli joku paketti katoaa siirron aikana ja vastaanottaja ei lähetä kuittausta lähettäjälle, lähetetään paketti uudelleen. Lopuksi yhteys lopetetaan yhteyden avauksen kaltaisesti kolmitiekättelyllä FIN ja ACK-paketeilla. Lopetus voidaan myös tehdä nelitiekättelyllä, jossa molemmat osapuolet lähettävät FIN-paketin ja kuittaavat ACK-paketilla vastapuolen FIN-paketin saapuneen. (Kavelová & Dostálek 2006, luku 9)

UDP (User Datagram Protocol) eroaa TCP-protokollasta esimerkiksi siten, ettei se varmista pakettien toimitusta vastaanottajalle. UDP ei myöskään suorita yhteyden alussa ja lopussa kättelyä vastaanottajan kanssa. (Kavelová & Dostálek 2006, luku 1) UDP:n avulla dataa voidaan siirtää ilman laitteiden välistä yhteyttä ja siksi sitä käytetäänkin usein reaaliaikaiseen tiedonsiirtoon, kuten videon ja äänen lähettämiseen suoratoistona. IoT-laitteissa UDP-protokollaa voidaan hyödyntää kaikissa ratkaisuiissa, joissa tarvitaan reaaliaikaista datan siirtoa, kuten kodin turvajärjestelmissä.

Itserakennetut IoT-laitteet sekä yritysten mukautetut laitteet käyttävät enimmäkseen yhden tason protokollia toimiakseen, mutta esimerkiksi usein kuluttajille myytävät internettiin kytkettävät laitteet käyttävät yleisiä monen tason protokollia tai standardeja. Yleisimmin käytetty ZigBee standardi ja Z-Wave protokolla käyttävät omia taajuuksiaan tiedonsiirtoon ja lisäksi molemmat tekniikat sisältävät kaikki alla esitetyt kerrokset peruskerroksesta sovelluskerrokseen asti (Kuva 2).

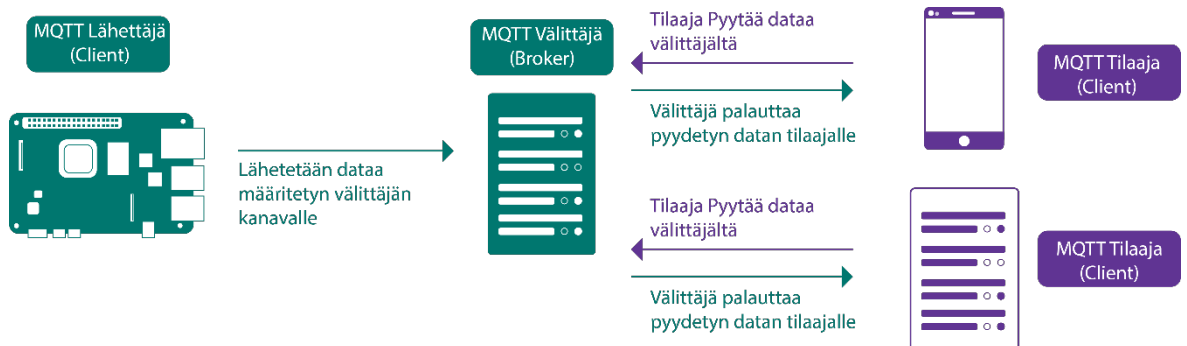


Kuva 2. Yleisimmät IoT-protokollat ja standardit kuvattuna OSI-mallin tavoin (Mukaillen Ledger 2016)

2.3 MQTT

MQTT on standardoitu IoT-laitteiden kanssa viestintään käytettävä kevyt ja suorituskykyinen protokolla, jonka lyhenne tulee englannin kielen sanoista Message Queuing Telemetry (Cope s.a.). Se on suunniteltu laitteille, joilla on tarkasti määritetty toiminallisuus sekä ympäristöihin, joissa on pieni kaistanleveys, pitkä viive yhteydessä tai epävakaata verkko (MQTT 2020). MQTT on HTTP-protokollan rinnalla toinen yleisimmistä IoT-sovellustason protokollista ja IoT-laitteen on käytettävä ainakin toista niistä, jotta laite voidaan yhdistää Googlen pilveen.

Publish/Subscribe eli vapaasti suomennettuna julkaisija/tilaaja arkkitehtuurimallia toteuttava MQTT välittää viestejä lähettäjältä tilaajille viestinvälittäjän välityksellä, joka yleensä on *broker* nimellä kutsuttava palvelin. Poikkeuksellisesti esimerkiksi Raspberry Pi voi toimia sekä lähettäjänä että välittäjänä samanaikaisesti. Käytännössä julkaisija lähettää viestejä välittäjän määritetyille kanavalle, josta vastaanottajat sitten tilaavat kyseisen lähettäjän viestejä (Kuva 3). Monelta-monelle mallin avulla yhdellä välittäjällä voi olla useita lähettäjiä ja tilaajia. Mikäli tilaaja ei saa yhteyttä reaaliaikaisesti välittäjään, ei data pääse katoamaan varastointi ominaisuuden vuoksi ja tilaaja voi noutaa datan myöhemmin asynkronisen kommunikaatio mallin tavoin. Tilaajan ei myöskään tarvitse tietää kuka tai mikä lähettäjä on, jolloin itsenäisten laitteiden tai järjestelmien välillä kommutaatio on turvallista. (MQTT s.a.b; Google Cloud 2020c).



Kuva 3. MQTT tilaaja/julkaisija arkkitehtuuri (Mukaillen MQTT s.a.).

Publish/Subscribe protokollan lisäksi MQTT tunnetaan machine to machine (M2M) tekniikan käyttäjänä. M2M eli laitteelta laitteelle tekniikka tarkoittaa kahden tai useamman laitteen välillä tapahtuvaa suoraa kommunikointia langattoman tai langallisen verkon yli. Ero IoT-tekniikkaan on siis pilven puuttuminen päätelaitteen ja hallittavan laitteen väliltä.

MQTT itsessään ei sisällä mitään salausrakennetta tai muutaakaan suojausta datan salaukselle, koska se on haluttu pitää kevyenä. Versiossa 3.1 ja siitä uudemmissa versioissa lähetettäviin paketteihin on voinut laittaa käyttäjän ja salausavaimen. Halutessaan varman turvallisuuden saa kuitenkin TCP/IP vaiheessa käytettävällä TLS yhteydellä, joka on salattu. Portti 8883 on tarkoitettu salatulle liikenteelle TLS yhteyttä käyttäessä ja portti 1883 salaamattomalle liikenteelle. (MQTT 2020)

2.4 HTTP

HTTP (Hypertext Transfer Protocol) on sovelluskerroksen protokolla, jonka tehtävänä on hakea palvelimelta yleensä selainpohjaiseen näkymään tekstiä, kuvia, videoita, sivun muotoilua ja paljon muuta dataa. Käytännössä selaimesta lähetetty komento avaa TCP-

yhteyden palvelimelle, jota pitkin HTTP-kutsu kulkee. Pyyntö voi kulkea suoraan tai epä-suorasti välityspalvelimien kautta ja palauttaa sitten palvelimelta pyydetyn näkymän tai toiminnon käyttäjän näkymään. Vastauksen saatuaan selain sulkee yhteyden tai käyttää samaa yhteyttä heti uudelleen seuraavaan pyyntöön. (MDN contributors 2019)

HTTP on toinen Googlen käyttämistä protokollista IoT-laitteiden kanssa. MQTT:tä käytetään lähinnä IoT-laitteiden kanssa, kun taas HTTP-protokollaa käytetään laajasti yleensä selainpohjaisissa asiakas-palvelin (client-server) mallia vaativissa ratkaisuissa. Samaa tekniikkaa voidaan käyttää myös IoT-laitteiden kanssa lähettämällä käskyjä tai pyyntöjä laitteelle, joka palauttaa pyydettyt arvot tai suorittaa käsketyn toiminnon.

HTTP-metodit (MDN contributors. 2019):

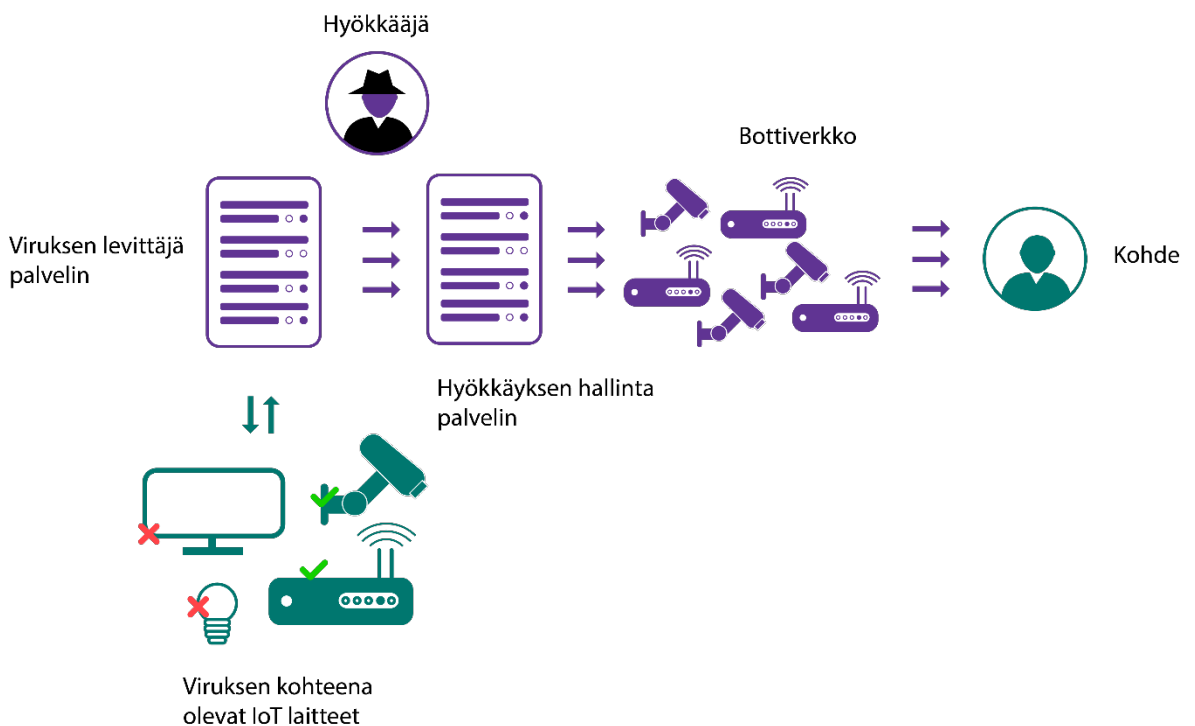
- GET - Hakee määritetyn resurssin.
- HEAD – Hakee sivuston otsikko tiedot.
- POST – Lähettää tehdyt muutokset.
- PUT – Korvaa nykyisen näkymän pyydetyllä tietosisällöllä.
- DELETE – Poistaa määritetyn resurssin.
- CONNECT – Pyytää tunneliyhteyden avausta kohde resurssin palvelimelle.
- OPTIONS – Pyytää kohde resurssin ominaisuuksia.
- TRACE – Suorittaa kaiku -testin resurssille johtavan reitin selvittämiseksi.
- PATCH – Käytetään resurssin osittaisen muutoksen tekemiseen.

HTTPS (Hypertext Transfer Protocol Secure) on suojattu versio ja nykyisin yleisempi verkkosivujen käytössä kuin HTTP. Yksinkertaisuudessaan HTTPS on TLS/SSL-protokollan ja HTTP-protokollan yhdistelmä. TLS on SSL-protokollasta vain uudempi versio, mutta toimii samankaltaisesti. TLS salaa sivustolle syötetyt tiedot ennen niiden lähettämistä käyttäjänäkymästä sivuston palvelimelle. Salauksen omaavilla sivustoilla on myös varmenne, joka todentaa sivun aidoksi ja siten käyttäjä voi varmistua olevansa oikealla sivustolla.

2.5 Tietoturva

Tietoturvaongelmat ovat olleet yleinen puheenaihe IoT-laitteista keskusteltaessa ja aiheesta onkin syytä keskustella, sillä laitteiden tietoturvassa on havaittu olevan aukkoja. Syy suuriin tietoturva puutteisiin johtuu luultavasti siitä, että laitteiden kehityksessä yritykset saattavat laiminlyödä tietoturvallisuutta, koska kuluttajatkaan eivät vielä osaa vaatia tarpeeksi kattavaa tietoturvaa (F-Secure 2020a). Kilpailu IoT-sektorilla kiihtyy ja uudet tuotteet on saatava nopeasti markkinoille, jolloin tietoturvan kehitys ja testaaminen saattavat jäädä taka-alalle. Tietoturva-aukot mahdollistavat esimerkiksi laitteen kaappaamisen hyökkääjän käyttöön, laitteen avulla voidaan tunkeutua sisäverkkoon tai laitteen keräämä data saattaa loukata tietosuojaa.

Yksi tunnetuimmista IoT-haittaohjelmahyökkäyksistä on Mirai niminen bottiverkko (botnet), joka hyödyntää IoT-laitteita botteinaan. Botti voi olla yksittäinen laite tai järjestelmä, jota hyökkääjä käyttää laajassa mittakaavassa hyökätäkseen valittuun kohteeseen. Yksittäisen IoT-botin laskentateho on mitätön, mutta yhdistämällä useita laitteita, saadaan aikaiseksi kuvassa 4 oleva suuren laskentatehon omaava verkko. Mirai hyödyntää IoT-laitteita, joiden oletuskäyttäjää ja salasanaa ei ole vaihdettu, jolloin se ei käytä hyökkäämiseen muita mahdollisia järjestelmän tietoturva-aukkoja. Ensimmäinen laaja Mirai hyökkäys oli vuonna 2016, jolloin bottiverkkoon oli kaapattu 100 000 laitetta ja Mirai suoritti bottien avulla hajautetun palvelunestohyökkäyksen (DDoS) Dyn nimipalvelimeen (DNS). (Kaspersky s.a.) Nimipalvelimella hallinnoi monia tunnettuja verkossa toimivia palveluita eli vaikutukset olivat suuria, kun palvelut lakkasivat toimimasta.



Kuva 4. Bottiverkon kerääminen ja hyökkäys kuvattuna.

Tietosuojaloukkauksien riski ja todennäköisyys kasvaa, mikäli elektronisten laitteiden valmistajat tekevät yhä useammista laitteista IoT-laitteita, vaikka sillä ei olisi kuluttajalle suurta käytännön hyötyä (Anderson Technologies 2019). Myös F-Securen tutkimusjohtaja Mikko Hyppönen arvioi raportissaan, että tulevaisuudessa suurin osa ilman älytoimintoja olevat laitteetkin ovat verkossa datan keräämisen vuoksi (F-Secure 2020a). Yritykset saattavat kerätä dataa asiakkaistaan ja saattavat käyttää sitä kohdistettuun mainontaan tai myydä dataa muille yrityksille. Esimerkiksi äly-TV valmistaja Vizio myöntää televisioiden keräävän dataa käyttäjistä ja yrityksen teknologiajohtaja Bill Baxter kertoo, että dataa

hyödynnetään paremman käyttäjäkokemuksen tarjoamiseksi (The Vergecast 2019). Kuluttajalta kysytään suostumusta datan keräämiseen esimerkiksi laitteen käyttöönoton yhteydessä hyväksyttävien ehtojen muodossa.

IoT-laitteiden avulla asuntoon, autoon tai verkkoon murtautuminen on myös noussut esiin viime vuosina. IoT-laitteiden tietoturva-aukkojen avulla voisi mahdollisesti päästä käsiksi muihin samassa verkossa oleviin laitteisiin tai verkossa liikkuvaan dataan. Huonosti suojattu kodin turvallisuusjärjestelmäkin voitaisiin sammuttaa hyökkääjän toimesta esimerkiksi IP kameran oletussalasanaa hyödyntäen. Yksi uusimmista kohteista on avaimettomat autot, joiden keskuslukitusta ja käynnistystä voi ohjata esimerkiksi WLAN tai Bluetooth yhteydellä. Muutamia autovarkautapauksia tiedetään, joissa on hyödynnetty edellä mainittuja yhteyksiä.

Markkinoille on tullut tietoturvalaitteita, jotka suojaavat kaikkia siihen liitettyjä laitteita tietokoneen virustorjuntaohjelman tavoin. Kuluttajien kotiverkoissa harvemmin on fyysistä palomuuria, jolloin reitittimeen kytketyt laitteet eivät ole suojattu mitenkään ilman erillistä virustorjunta sovellusta. IoT-laitteisiin ei erillistä virustorjuntaa luonnollisesti saa, jolloin hyökkäykselle altistumisen riski kasvaa. F-Securen kehittämä tietoturvareititin kantaa nimeä SENSE ja se toimii kaikkien siihen liitettyjen laitteiden verkkoliikenteen suojana. (F-Secure 2020b) Tietoturvaa voi myös parantaa vahvoilla salasanoilla sekä hyvällä ja suojatulla verkkoarkkitehtuurilla.

3 Google Cloud Platform

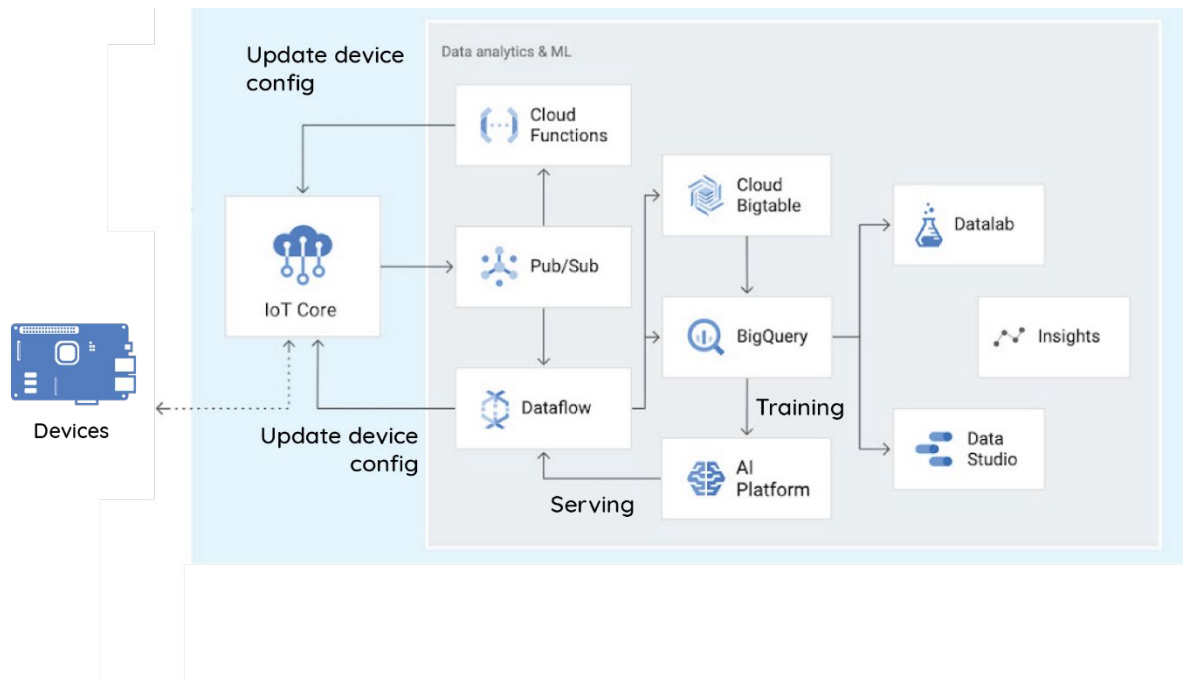
Googlen kehittämä Cloud Platform on suunnattu yritysten pilvialustaksi ja on yksi maailman johtavista pilviratkaisuiden palveluntarjoajista. Google tarjoaa laajasti työkaluja ja palveluita datan varastointiin ja prosessointiin, kuten virtuaalipalvelimia, tietoturvaa, laitehallintaa, tietokantaratkaisuja sekä analyyttisiä työkaluja. (Google s.a.a)

3.1 Google pilvialustana

Keskittämällä kaikki yrityksen tuottamat palvelut samalle palveluntarjoajalle säästyy kustannuksia ja se tehostaa kommunikointia esimerkiksi tietokannan ja sovelluksen välillä. Google tarjoaa käytännössä kaikkia erityyppisiä pilvipalveluita eli Infrastructure as a Service (IaaS), Platform as a Service (PaaS) ja Software as a Service (SaaS) tyyppisiä pilviratkaisuita.

Cloud Platform alustalla on selainpohjainen graafinen käyttöliittymä pilvipalveluiden hallintaan. Päävalikosta löytyy kaikkien palveluiden hallintapaneelit ja päävalikon yläosaan voi kiinnittää haluamansa palvelut, josta ne on helppo löytää. Yleisen hallintapaneelin etusivulla näkyy kaikki oletustiedot ja esimerkiksi API kutsujen monitorointi ikkuna. Hallintapaneelia on mahdollista muokata piilottamalla tai lisäämällä ikkunoita ja erityyppisiä visuaalisia graafeja. Aktiivisuus välilehdellä pystyy seuraamaan oman käyttäjän ja projektien tapahtumia loki tyyppisessä muodossa. Jokaisen muutoksen leimassa näkyy päivämäärä ja aika, tehty muutos sekä muutoksen tekijä.

Google Cloud IoT on kattava paketti sisältäen tarvittavat työkalut IoT-laitteiden hallintaan, datan prosessointiin, analysointiin ja säilytykseen (Kuva 5). Google tarjoaa työkaluja pilvessä työskentelyyn sekä IoT-ympäristön reuna (edge) laitteella. Edge on laite, joka kustelee pilven ja esineiden internet laitteiden tai sensoreiden välillä. Cloud IoT paketin hinta määräytyy datan käytön mukaan ja siihen sisältyy kaikkien eri moduulien ja työkalujen käyttö. Datan käyttö lasketaan sisään ja ulospäin virtaavan datan määrästä yhteen laskettuna. Alle 250MB/kk on ilmaista, 250MB-250GB/kk maksaa 0.0045 \$/MB, 250GB-5TB/kk maksaa 0.0020 \$/MB ja 5TB tai enemmän/kk maksaa taas 0.0045 \$/MB. (Google s.a.a)



Kuva 5. Googlen IoT-rakenne. (Mukaillen Google Cloud 2020i)

Yleisimpiä käyttötarkoituksia IoT-palveluiden lisäksi ovat esimerkiksi verkkosovelluksen isännöinti, tekoälyn ja algoritmien suorittaminen tai massadatan (Big data) prosessointi. Ainakin useimmat palvelut ovat saatavilla kustannustehokkaasti siten, että tarvittava teho- kapasiteetti tai tallennustila muuttuu tarpeen mukaan. Prosessit eivät silloin ole turhaan käynnissä täydellä teholla, koska tehot nousevat vasta tarpeen vaatiessa. (Google 2020b)

Google tarjoaa laadukkaiden palveluiden lisäksi valtavan määrän dokumentaatiota ja ohjeita palveluiden käyttämiseksi. Vakaiden palveluiden lisäksi Google tarjoaa kattavan verkkoinfrastruktuurin maailman laajuisesti. Verkko kattaa 24 palvelinkeskusta maailman laajuisesti, joista yksi on Suomessa, Haminassa. Palvelinkeskusten lisäksi verkossa on 144 solmukohtaa, jotka lisäävät verkon vakautta, luotettavuutta ja tehokkuutta. (Google s.a.b)

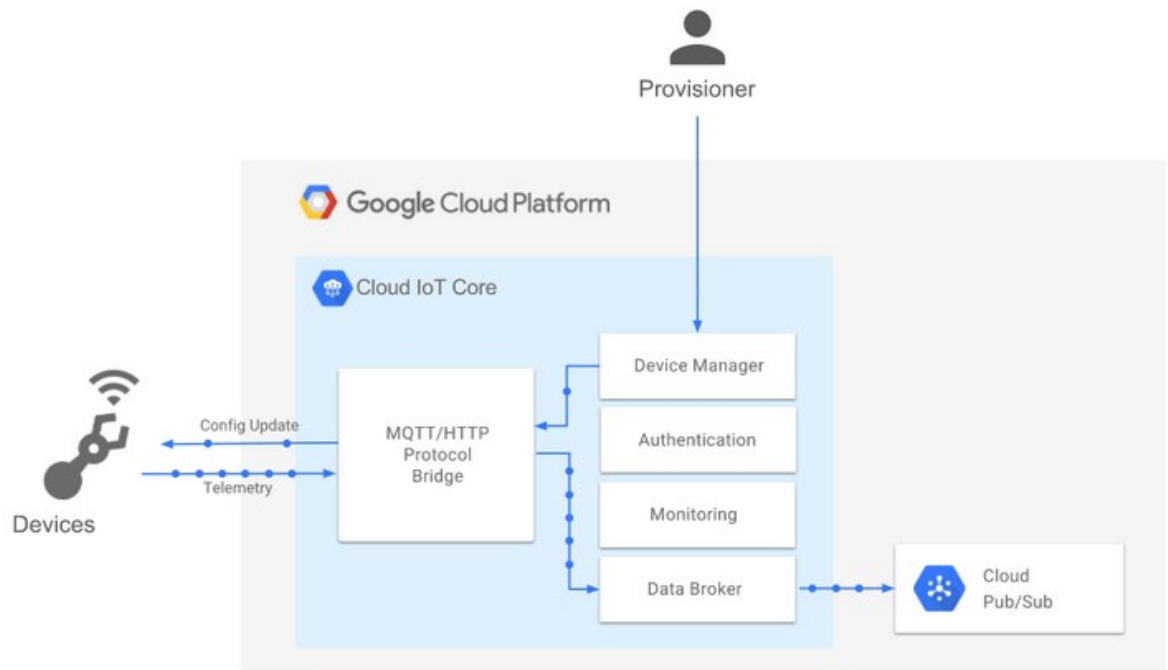
3.2 IoT Core

IoT Core on nimensä mukaisesti Cloud IoT kokonaisuuden ydin, joka toimii IoT-laitteiden ja Googlen Cloud palveluiden välillä välittäen dataa molempiin suuntiin. IoT Coren avulla järjestelmänvalvoja voi seurata yhdistettyjen laitteiden toimintoja, mahdollisia virheilmoituksia ja laitteiden fyysistä tilaa reaaliajassa. (Google Cloud 2020d)

Googlen tukemat protokollat IoT käytössä ovat HTTP ja MQTT. IoT Core on yhteydessä liitettyihin laitteisiin HTTP tai MQTT siltojen kautta. Google suosittelee käyttämään HTTP-protokollaa, mikäli ei tiedetä kumpi protokolla sopisi yhdistettäviin laitteisiin paremmin tai

jos tekijä on aloittelija. Googlen omat API palvelut käyttävät myös HTTP-protokollaa oletuksena. (Google Cloud 2020j)

Pääkomponentteina IoT Coressa toimivat laitehallinta ja protokollasillat (Kuva 6). Protokollasillat MQTT ja HTTP varmistavat IoT Coren ja laitteen välisen kommunikoinnin lisäksi laitteiden yhdistämisen pilveen. Laitehallinta mahdollistaa laitteiden rekisteröimisen sekä niiden hallinnoinnin. Yhdistetyn laitteen data ohjataan IoT Core moduulista edelleen Pub/Sub palvelun määritetyille kanavalle, josta se voidaan edelleen ohjata muille pilvipalveluille. (Google Cloud 2020d)



Kuva 6. IoT Core palvelun toiminta kuvattuna. (Google Cloud 2020d)

3.3 Cloud Pub/Sub

Googlen Pub/Sub palvelu käyttää saman tyyppistä asynkronista eli ei-reaaliaikaista lähettäjä/tilaaja mallia, jota esiteltiin kappaleessa 2.3. Pub/Sub palvelu voi vastaanottaa ja säilyttää lähettäjältä tullutta dataa ja välittää datan muille määritetyille palveluille tai vastaanottajille, jotka voivat tehdä HTTPS kutsuja `pubsub.googleapis.com` osoitteeseen. Google Cloud IoT rakenteessa tilaajat ovat yleensä Cloud Functions tai Dataflow palvelut, mutta kolmannen osapuolen tilaajat voivat olla yhteydessä Pub/Sub palvelun kanaville. (Google Cloud 2020c)

3.4 Dataflow

Apache Beam SDK avoimen lähdekoodilla rakennettava tiedonsiirtoputki (pipeline) järjestelmä voidaan toteuttaa Googlen pilvialustalla Dataflow palvelulla suoratoistona. Google tarjoaa myös valmiita Dataflow malleja, jotka on valmiiksi optimoitu tarvittaviin prosesseihin. Dataflow on suunniteltu laajoihin tiedon prosessointi kokonaisuuksiin, jossa linjastossa voidaan määrittää alkuperäisen datan tietokanta, välillä tapahtuvat palvelut ja prosessit sekä lopullinen tietokanta. (Google Cloud 2020g)

Dataflow palvelun dataputkisto toimii pienellä viiveellä, jolloin se voidaan ohjelmoida toimimaan reaaliaikaisesti. Järjestelmä taipuu moneen käyttötarkoitukseen ja se voidaan myös ajastaa suorittamaan haluttuja linjastoja esimerkiksi öisin, jolloin muut toiminnot eivät kulu tarvittavia tehoja, eivätkä rahalliset kustannuksetkaan eivät pääse nousemaan. (Google Cloud 2020g)

3.5 Cloud Functions

Skaalautuva ja palveliton Cloud Functions palvelu tarjoaa alustan, jossa voi suorittaa yksinkertaisia itseohjelmoituja ohjelmia pilvi-infrastruktuurin sisällä. Palvelun avulla pilviympäristö voidaan räätälöidä yritykselle sopivaksi Java, JavaScript, Go tai Python ohjelmointikielillä. IoT käytössä Cloud Functions palvelulla voidaan tehdä esimerkiksi useita laukaisimia (trigger), jotka tietyin ehdoin käynnistävät määritetyt tapahtumat (event). Tapahtumat voivat olla datan tallentamista tietokantaan tai datan prosessointia ennen tallentamista. Raakadatan prosessoinnin jälkeen valmis data voidaan viedä IoT Coren läpi Dataflow palvelun avulla suoraan tietokantaan tai haluttujen pilvipalveluiden läpi. Cloud Functions palvelun avulla voidaan myös tehdä ohjelmia, joilla voidaan hallita laitetta ja lähettää dataa tai käskyjä pilvestä laitteelle. (Google Cloud 2020e)

3.6 BigQuery

Petatauvuihin asti skaalautuva BigQuery on Google Cloud alustan täysin hallinnoitu tietokantakokonaisuus. Se sisältää tietokannan lisäksi työkalut tehokkaaseen datan analysoimiseen ja käsittelyyn. BigQuery on käytettävissä selainpohjaisen Console hallintapaneelin kautta, perinteisellä komentorivi ohjelmalla ja REST API teknologian välityksellä. REST API toimintoja voidaan käyttää esimerkiksi Java, .NET ja Python ohjelmointikielellä toteutetuissa ulkopuolissa järjestelmissä. BigQuery tietokanta pohjautuu SQL syntaksiin, joten kyselyitä voidaan tehdä SQL kielellä. (Google Cloud 2020f)

3.7 Tietoturva

Pilveen lisättävät laitteet käyttävät todentamiseen yksityisen ja julkisen avaimen muodostamaa avainparia. Jokaisella laitteella oleva avainpari tekee pilven ja laitteen välisestä liikenteestä eheän ja turvallisen. MQTT ja HTTP käyttävät JWT menetelmää eli JSON Web Token varmistusta JSON paketeissa esimerkiksi epäsymmetrisen RS256 avainparin muodossa. Tyypilliseen tapaan JSON sisältää otsikon ja sisällön, mutta JWT menetelmässä jokaisessa JSON tiivisteeseen otsakkeessa on myös varmenne osio, jolla voidaan varmistua pyyntöjen tulevan varmistetusta lähteestä. MQTT yhteydeltä vaaditaan lisäksi SSL/TLS-protokollan suojausta. (Google Cloud 2020g) Google ei mainitse, että SSL/TLS suojausta vaadittaisiin HTTP-protokollaa käytettäessä, mutta nykyisillä standardeilla ja suosituksilla suojattua HTTPS-yhteyttä tulisi mieluiten käyttää HTTP-yhteyden sijasta.

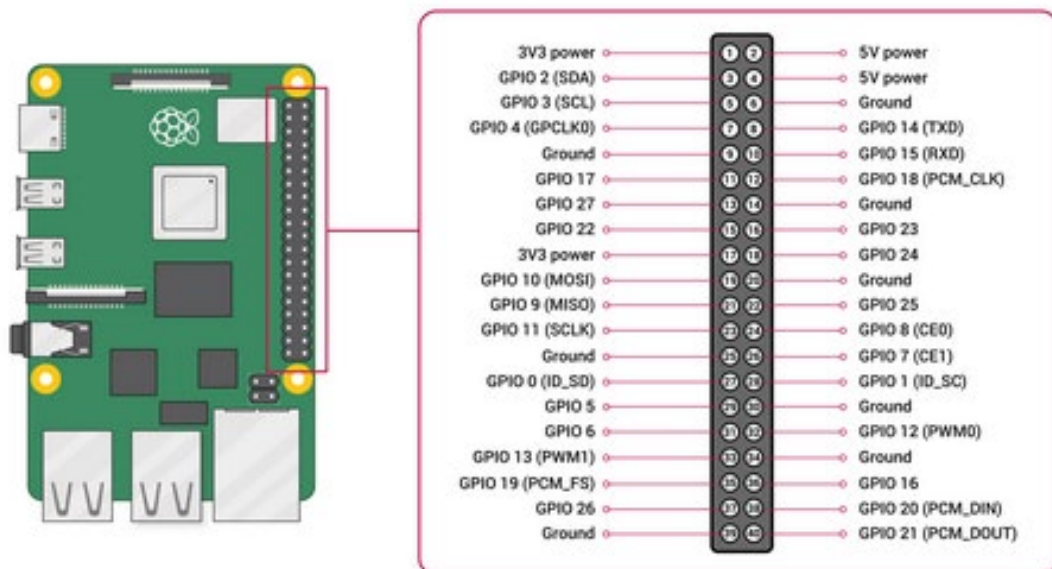
Laitteen rekisteröintivaiheessa valitaan toinen tai molemmat protokollat, jonka lisäksi luodaan avainpari. Toiminto voidaan tehdä Google Consolen graafisella käyttöliittymällä tai komentorivityökalua apuna käyttäen. Onnistuneen rekisteröinnin päätteeksi saadaan tiedosto, joka sisältää yksityisen avaimen ja tiedosto tulee viedä yhdistettävälle laitteelle. Myös laitteella voidaan generoida yksityinen avain ja avain liitetään Googleen rekisteröintivaiheessa. Laitteen ollessa valmis JSON konfiguroinnin osalta, generoidaan laitteella yksityisestä avain -tiedostosta JWT paketti ja lähetetään se IOT Coren HTTP tai MQTT sil- lalle. Silta vertaa yksityistä avainta julkiseen avaimeen ja vastaa sitten laitteelle yhteyden onnistumisesta tai epäonnistumisesta. (Google Cloud 2020g)

4 Raspberry Pi

Yhden piirilevyn tietokone ei sisällä mitään irrallisia komponentteja kaiken ollessa integroitu samalle piirilevyille. Raspberry Pi Foundation valmistaa reilun luottokortin kokoisia laitteita, jotka lukeutuvat yhden piirilevyn tietokoneiksi. Tämä isobritannialainen voittoa tavoittelematon hyväntekeväisyysjärjestö kehitti edullisen piirilevyn ja Raspberry Pi OS (ennen Rasbian) käyttöjärjestelmän tietotekniikan opettamistarkoitukseen. Ensimmäinen Raspberry Pi 1 Model B+ julkaistiin vuonna 2014. (Raspberry Pi Foundation s.a.)

4.1 Ominaisuudet

Raspberry Pi tietokoneesta on useita eri malleja, mutta kaikista eri piirilevyistä löytyy tärkein ominaisuus eli 26–40 kappaletta GPIO (general-purpose input/output) pinnejä. Käytettävissä on erityyppisiä datapinnejä, maadoituspinnejä sekä 3.3V ja 5V virtapinnejä (Kuva 7). Niiden avulla piirilevyyn voidaan kytkeä esimerkiksi muita piirilevyjä, antennejä tai sensoreita. Raspberry Pi omat lisäosat kuten kamera ja näyttö voidaan kytkeä valmiiden liittimien kautta. Raspberry Pi käyttää 5V virtaa USB-C tai mikro-USB väylän kautta. (Raspberry Pi Foundation s.a.)



Kuva 7. Raspberry Pi GPIO pinnit. (Raspberry Pi Foundation s.a.)

Käyttöjärjestelmä vaihtoehtoja on useita, joista suurin osa on Linux pohjaisia. Järjestön kehittämä avoimenlähdekoodin Raspberry Pi OS on Debian pohjainen käyttöjärjestelmä, joka on optimoitu Raspberry Pi alustoille ja se sisältää itsessään suuren määrän valmiita ohjelmistoja esimerkiksi koodaamisen tai laiterakentamisen opetteluun. Käyttöjärjestelmä on saatavilla työpöytäversiona tai kevyemmällä komentorivi-käyttöliittymällä. Suosituttu

käyttöjärjestelmiä ovat myös Lakka, Ubuntu ja Windows IoT Core. Käyttöjärjestelmä asennetaan Micro SD muistikortille toisella tietokoneella ja muistikortti kytketään sitten piirilevyn muistikortinlukijaan. Muistikortilla on myös laitteen tallennustila, jonka vuoksi muistia suositellaan olevan ainakin 16GB.

4.2 Raspberry Pi IoT-alustana

Yleisimmät IoT käytössä olevat Raspberry Pi piirilevyt ja niiden ominaisuudet on kuvattu taulukossa 1. Raspberry Pi on yleisesti ottaen yksi parhaimmista alustoista rakennettavien IoT-laitteiden tekemiseen sen monipuolisuuden ansiosta. Markkinoilta löytyy muitakin vastaavia piirilevyjä mm. Asuksen, Nvidian ja Intelin valmistamia tuotteita, mutta kilpailijoiden tuotteet ovat hinta/teho suhteeltaan kalliimpia.

Taulukko 1. Käytetyimmät Raspberry Pi alustat IoT-laitteissa.

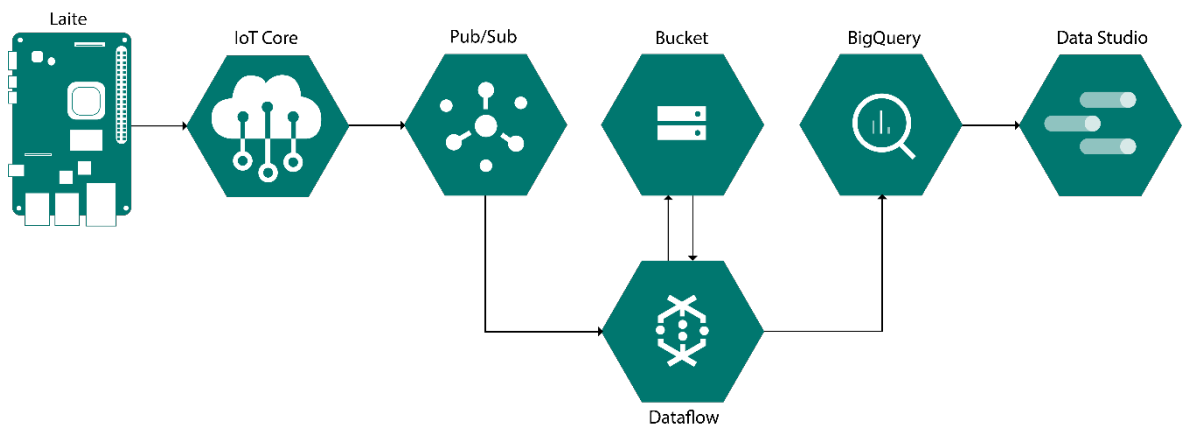
Malli	Nopeus	RAM	USB portit (kpl)	Ethernet	Wi-Fi	Bluetooth	Hinta (\$)
Raspberry Pi 3 Model B	1200MHz	1GB	4	Kyllä	802.11n	4.01	\$35
Raspberry Pi 3 Model A+	1400MHz	512MB	1	Ei	802.11ac/n	4.02	\$25
Raspberry Pi 3 Model B+	1400MHz	1GB	4	Kyllä	802.11ac/n	4.02	\$35
Raspberry Pi 4 Model B	1500MHz	2GB	2xUSB2, 2xUSB3	Kyllä	802.11ac/n	5.00	\$35
Raspberry Pi 4 Model B	1500MHz	4GB	2xUSB2, 2xUSB3	Kyllä	802.11ac/n	5.00	\$55
Raspberry Pi 4 Model B	1500MHz	8GB	2xUSB2, 2xUSB3	Kyllä	802.11ac/n	5.00	\$75
Raspberry Pi Zero	1000MHz	512MB	1	Ei	Ei	Ei	\$5
Raspberry Pi Zero W	1000MHz	512MB	1	Ei	802.11n	4.01	\$10
Raspberry Pi Zero WH	1000MHz	512MB	1	Ei	802.11n	4.01	\$15

Raspberry Pi 3 julkaistiin 2016, joka korvasi edeltäjänsä 1 ja 2 mallin huomattavasti tehokkaammalla 64 bittisellä prosessorilla sekä tehokkaammalla näytönohjaimella. Tämän hetken tehokkain malli Pi 4 Model B 8GB vastaa jo keskihintaisen kannettavan tietokoneen teholuokkaa. Teho voi olla ylimitoitettu pelkkien sensoreiden oletusyhdykskäytäväksi, mutta sensoreiden dataa voitaisiin esimerkiksi prosessoida laitteella ennen sen lähettämistä pilveen.

Zero mallit sisältävät huomattavasti vähemmän tehoa kuin edellä mainitut mallit, mutta ne soveltuvat paremmin ratkaisuihin, joissa pienestä koosta ja alhaisesta virrankulutuksesta on hyötyä. Zero malleilla voidaan esimerkiksi rakentaa huomaamattomia valvontakameroita tai akkukäyttöisiä sensoreita, joiden on mahdollista pieneen tilaan. Hinnaltaankin Zero mallit ovat tuntuvasti muita halvempia.

5 Raspberry Pi älylämpömittariksi

Tutkimuksessa haluttiin selvittää, kuinka IoT-laite kytetään Googlen Cloud Platform alustalle ja kuinka alustan päälle rakennetaan toimiva kokonaisuus hyödyntäen teoriavaiheen tutkimusta. Googlen pilvialustaan päädyttiin, koska se tarjosi hyvät lähtökohdat tämän työn kaltaiseen testaamiseen ja omaa osaamista Googlen ympäristössä haluttiin syventää. Laitteen rungoksi valikoitui Raspberry Pi 4 B. Raspberry piirilevyyn päädyttiin tehokkuuden, GPIO pinnien sekä käyttöjärjestelmä mahdollisuuksien vuoksi. Yhtenä kriteerinä oli ohjelmiston toteuttaminen Python ohjelmointikielellä ja Raspberry Pi:n tukemat käyttöjärjestelmät mahdollistivat myös sen. Laite rakennettiin itse, koska haluttiin päästä tutkimaan myös IoT-laitteen rakenteita tarkemmin, vaikuttamaan laitteen käyttäytymiseen ja lähetettävän datan sisältöön. Uusimman 4. sukupolven malli tarjosi keskusmuistia 4GB, josta olisi myös hyötyä järjestelmää laajennettaessa tulevaisuudessa. Piirilevyn valinnan jälkeen aloitettiin suunnittelemaan laitteen toimintaa, pilvi kokonaisuutta sekä muiden komponenttien valintaa. Projektin rakenne on avattu kokonaisuudessaan kuvassa 8.



Kuva 8. Tutkimuksen rakenne kokonaisuudessaan.

5.1 Google Cloud Platform käyttöönotto

Ennen työn aloitusta luotiin uusi Gmail tili, jolla voitiin perustaa uusi projekti Google Cloud Platform alustalle. Käyttäjälle aktivoitiin Googlen tarjoama kolmen kuukauden kokeilu jakso, joka sisälsi noin 250 € arvosta käyttörahaa. Projektille annettiin nimi *home-temp*, jonka jälkeen projektipohja oli valmis ja järjestelmää lähdettiin rakentamaan sen päälle. Graafisen käyttöliittymän ohjauspaneelistä lisättiin palveluita APIs & Services nimiseltä välilehdeltä. Dataflow, Cloud Pub/Sub ja Cloud IoT palveluiden API toiminnot aktivoitiin projektiin, jotta dataa voitiin kuljettaa pilven eri palveluiden välillä.

API toimintojen aktivoimisen jälkeen siirryttiin samassa valikossa pääsy tietojen rekisteröintiin. Rekisteröinnin tarkoituksena oli tehdä todennus liitettävälle laitteelle. Käytännössä

siis Googlen pilveen tehdylle projektille kerrottiin, mikä laite saa olla siihen yhteydessä. Rekisteröinnissä luotiin ensin palvelukäyttäjän avain, jolle määritettiin nimi, id, rooliksi omistaja sekä käytettävän avaimen tyyppi JSON muotoon. Palvelukäyttäjä on virtuaalinen käyttäjä, joka ei ole kukaan todellinen henkilö. Palvelukäyttäjä veloitettiin tässä tapauksessa avaamaan yhteys laitteelta pilveen ja omistajan roolin ansiosta se pystyi viemään dataa pilven projektiin. Vastaavan todennuksen olisi voinut luoda myös manuaalisesti Googlen Cloud alustalla tai generoimalla yksityisen ja julkisen RSA avainparin yhdistettävän laitteen komentorivityökalulla.

Onnistuneen palvelukäyttäjän avaimen luonnin jälkeen saatiin rekisteröitävälle laitteelle JSON tiedosto, joka sisälsi palvelukäyttäjän ja projektin tiedot sekä yksityisen avaimen. JSON tiedoston luonnin yhteydessä syntyi myös julkinen avain, joka jäi Googlen pilveen tehtyyn projektiin. Julkisella avaimella voitiin myöhemmässä vaiheessa todentaa saapuva yksityisen avaimen paketti ja avata todennettu yhteys laitteeseen. Projektiin olisi voinut liittää muitakin laitteita samalla JSON-paketilla ja ohjata laitekohtaista liikennettä eri Pub/Sub kanaville. Kyseinen JSON-todennuspaketti on myös aina lähetettävä laitteelta Googlle uuden yhteyden avaamiseksi, jos yhteys jotain syystä keskeytyy.

API valikosta siirryttiin IoT Coren välilehteen, josta rekisteröitiin laite antamalla sille ID, valitsemalla HTTP-protokolla ja alueeksi Eurooppa. Lisäksi laitteelle luotiin tässä rekisteröinnissä Pub/Sub palvelun kanava (topic), josta muut pilvipalvelut voisivat tilata lähetettyä dataa. Tässä vaiheessa ei ollut vielä merkitystä, että mikä laite on kytketty tai millaista dataa se lähettää. Rekisteröinnillä tehtiin valmius yhden laitteen yhteyttä varten.

Datalle tarvittiin myös tietokanta, joten otettiin käyttöön BigQuery luomalla sille datajoukko (dataset). Aikaisempien palveluiden kaltaisesti valittiin datajoukolle nimi, alueeksi Eurooppa sekä haluttu salaus. Salauksen olisi voinut tehdä Cloud Key Management Servicen maksullisella avainparilla, jossa avainparin tyyppiin olisi päässyt vaikuttamaan, mutta tässä työssä sille ei nähty tarvetta ja päädyttiin käyttämään Googlen hallinnoimaa salaus-avainta. Tässä vaiheessa voitiin luoda BigQueryn datajoukolle myös taulu, koska jo suunnitteluvaiheessa päätettiin laitteen lähettävän lämpötilan FLOAT muodossa sekä kellonajan ja päivämäärän TIMESTAMP muodossa. Taululle määritettiin nimeksi *outdoorTemp* ja tietue, joka sisälsi lämpötilan nimellä *temp* ja lämpötila otannon aikatiedot nimellä *time*. Tyypeiksi lämpötilalle ja aikatiedolle asetettiin suunnitellut FLOAT ja TIMESTAMP.

Varastointi kontti eli Storage bucket on Cloud Storage kokonaisuuden palvelu, joka toimii BigQueryn rinnalla datan varastointi paikkana. Kontille määriteltiin tuttuun tapaan nimi ja alueeksi Eurooppa. Varastointi muotoja oli valittavissa pitkäaikaisesta arkistointi säilytyksestä tavalliseen lyhytaikaisempaan säilytykseen. Tähän työhön valittiin tavallinen säilytys

muoto, jossa ei ollut mitään pakollisia kriteereitä, koska tarkoituksena oli säilöä dataa konttiin vain hetkellisesti. Pidempiaikaisissa säilytysvaihtoehdoissa oli vähimmäis- varastointiaika, ennen datan siirtämistä uudelleen. Salaukseksi valittiin taas Googlen hallinnoima avain maksullisen avainparin sijasta. Käytännössä laitteesta lähetettävä data ohjattiin tähän konttiin ja kontista BigQueryn tietokantaan, josta sitä voitiin myös hallita SQL kyselyillä. Dataa ei olisi ollut pakko kuljettaa kontin välityksellä, jos seuraavaksi esiteltävä Dataflow olisi tehty täysin kustomoituna. Valmista Googlen kaavaa käytettäessä kontti kuitenkin vaadittiin.

Dataflow palvelun käyttöönotto ja määrittäminen oli haastavin kaikista tähän asti määritetyistä palveluista. Dataflow palvelulla dataputkisto olisi voitu ohjelmoida täysin kustomoidusti esimerkiksi Python ohjelmointikielen avulla. Tässä työssä päädyttiin kuitenkin käyttämään Googlen valmista kaavaa, joka luo kokonaisen datan prosessointi järjestelmän. Järjestelmän tarkoituksena oli kuljettaa data Pub/Sub kanavasta, kontin kautta aina BigQuerylle asti. Pohjaksi valittiin *Cloud Pub/Sub Topic to BigQuery* kaava, johon määritettiin nimen ja alueen lisäksi Pub/Sub kanavan nimi, tietokannan taulun nimi, kontin nimi sekä prosessien enimmäismäärä.

Dataflow kaavaan määriteltiin kanava, taulu ja kontti tässä muodossa:

- Kanava: projects/home-temp-292518/topics/topic_temp01
- Taulu: home-temp-292518:outdoorTempDataset
- Kontti: gs://home-temp/tmp/

Kokonaisuudessaan järjestelmässä laite lähettää datan IOT Coreen JSON pakettina, käyttäen TCP-protokollaa ja TLS salattua HTTP-protokollaa. Paketin otsakkeessa oleva projekti ID sekä Pub/Sub kanavan nimi ohjaa datan määritetylle Pub/Sub kanavalle asti. Pilvessä dataa kuljetetaan rakennetun Dataflow putkiston avulla Pub/Sub palvelusta väliaikaiseen konttiin ja kontista BigQuery tietokanta palveluun. Data Studiossa haetaan vielä data linkittämällä Cloud Platform projekti haluttuun näkymään. Projektin ensimmäinen prosessointi tapahtuu laitteessa Python koodissa, jossa geneerinen lämpötilan arvo muutetaan ymmärrettävään muotoon. Toinen ja viimeinen prosessointi tapahtuu Data Studio palvelussa, kun data järjestetään halutun datan mukaan oikeaan järjestykseen.

5.2 Ulkolämpömittarin rakentaminen

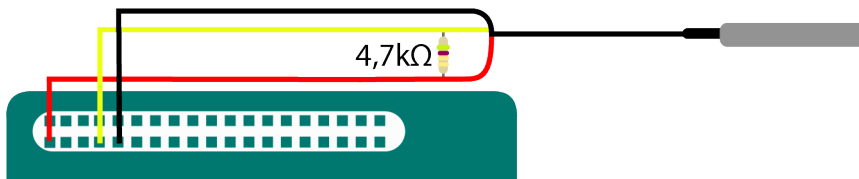
Perinteinen ulkolämpömittari haluttiin korvata älykkäämmällä versiolla ja näin päädyttiin rakentamaan sellainen itse osana toiminnallista osuutta. Anturin tyyppi rajautui digitaaliseen, koska Raspberry Pi alustoissa ei ole analogisia väyliä ja haluttiin välttää ylimääräi-

siä piirilevyjä pitämällä laite mahdollisimman yksinkertaisena. Piirilevy haluttiin pitää si-
sällä, joten anturiksi valikoitui digitaalinen ja vedenkestävä DS18B20 lämpötila-anturi kah-
den metrin johdolla. Anturissa on perinteisen ulkolämpömittarin kaltainen metallinen kap-
seli lämpötilan mittaamiseen (Kuva 10).



Kuva 9. DS18B20 anturi.

DS18B20 lämpötila-anturi vaati toimiakseen 4,7 kilo-ohmin vastuksen. Vastus juotettiin po-
sitiivisen virran sekä data kaapeleiden väliin ja juotos suojattiin vielä sähköteipillä. Juotettu
liitäntä haluttiin piiloon, joten se ratkaistiin kustomoidun kotelon johdoille varatulla osiolla.
Anturi kytkettiin piirilevyn GPIO4, 3,3 V ja maadoitus pinneihin kytkentä kaavion mukai-
sesti (Kuva 11).



Kuva 10. DS18B20 kytkentä kaavio.

Raspberry Pi 4 mallin on kerrottu kuumentuvan pienessäkin rasituksessa, joten erillistä
jäähdytystä suositeltiin. Laitteen ollessa ikkunalaudalla tai muutoin yleisessä paikassa, ei
haluttu mekaanista jäähdytystä siitä aiheutuvan äänen vuoksi ja päädyttiin kuvassa 12 nä-
kyviin passiivisiin alumiinista valmistettuihin jäähdytyssiileihin. Kustomoidulla kotelolla voi-
tiin myös varmistaa riittävä ilmanvaihto, koska lämpimän ilman johtamiseksi ulos kote-
lost, tehtiin ilmanvaihtoaukkoja kuumenevien komponenttien kohdille.



Kuva 11. Raspberry Pi 4, 3D tulostetussa kotelossa kytkettynä lämpötila-anturiin.

5.3 Laitteen käyttöönotto

Raspberryn asentamiseen tarvittiin micro-SD muistikortti ja toinen tietokone, jotta käyttöjärjestelmä voitiin asentaa muistikortille. Raspberry myy myös esiasennettuja muistikortteja, jolloin toista tietokonetta ei tarvita, mutta esiasennettu käyttöjärjestelmä ei olisi sopinut tähän tarkoitukseen. Käyttöjärjestelmän asentamista varten ladattiin Raspberrypi.org sivustolta Raspberry Pi Imager asennustyökalu sekä käyttöjärjestelmän .img asennustiedosto.

Käyttöjärjestelmäksi valittiin kevyempi 32 bittinen Raspberry Pi OS Lite, koska sensoreiden lukemiseen ja datan lähettämiseen ei tarvittu kaikkia työpöytä -version ominaisuuksia. Lite versiossa ei ole ollenkaan GUI (Graphical User Interface) pohjaista käyttöliittymää ja laitteen käyttö tapahtuu ainoastaan komentoriviä käyttäen. Lite versioon päädyttiin jo projektin suunnittelu vaiheessa, koska tiedettiin työpöytäversion käyttämisen olevan verkon yli liian raskasta ja hallinta tulisi tapahtumaan SSH yhteyden yli.

Asennustyökalulla valittiin asennus kohteeksi micro-SD muistikortti ja asennettavaksi tiedostoksi ladatun käyttöjärjestelmän .img päätteinen tiedosto. Asennuksen jälkeen Raspberryn SSH piti vielä sallia toisella tietokoneella manuaalisesti, luomalla käyttöjärjestelmän kansiorakenteen juureen `ssh` niminen tiedosto ilman mitään tiedostopäätettä. Tämän olisi voitu tehdä myös liittämällä laitteeseen näytön sekä näppäimistön ja muuttamalla SSH

asetuksia Raspberryn komentorivillä, mutta nähtiin helpommaksi vain luoda edellä mainittu tiedosto. Muistikortti laitettiin piirilevyllä olevaan lukijaan ja verkko- sekä virtakaapeli liitettiin laitteeseen, jonka jälkeen laite käynnistyi vilkuttaen punaista ja vihreää valoa. Verkon reitittimestä käytiin katsomassa laitteen IP osoite, jotta tiedettiin mihin yhteys avattaisiin. Muutaman minuutin odottelun jälkeen otettiin PuTTYlla ensimmäinen SSH yhteys haettuun IP osoitteeseen ja porttiin 22.

Raspberry vastasi nopeasti pyytäen käyttäjätunnusta ja salasanaa. Oletuksena Raspberry Pi OS käyttöjärjestelmissä käyttäjänimi on pi ja salasana on raspberry, joka vaihdettiin kirjautumisen yhteydessä. Seuraavaksi otettiin Raspberry käyttöön määrittämällä perusasetukset ja tekemällä tarvittavat toimenpiteet ennen varsinaisen lämpömittarin järjestelmän rakentamista. Käyttöönotto tehtiin seuraavilla komennoilla:

Luotiin uusi admin niminen käyttäjä ja määritettiin sille salasana:

```
sudo useradd -m admin  
sudo passwd admin
```

Lisättiin käyttäjä sudo ryhmään, jotta voitiin suorittaa komentoja pääkäyttäjän oikeuksilla:

```
sudo usermod -aG sudo admin
```

Määritettiin Wi-Fi verkon nimi ja salasana muokkaamalla wpa_supplicant tiedostoa:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Luotiin salattu salasana:

```
sudo /etc/wpa_supplicant/ wpa_passphrase <verkon nimi>
```

Edellisen komennon jälkeen saatiin palautus, jossa pyydettiin verkon salasanaa. Salasan syöttämisen jälkeen saatiin viimeisenä palautuksena salasana takaisin salatussa muodossa, joka kopioitiin ja vaihdettiin *wpa_supplicant* tiedostoon aikaisemmin määritetyn salasanan tilalle, muokkaamalla tiedostoa uudelleen komennolla: `sudo nano`

```
/etc/wpa_supplicant/wpa_supplicant.conf
```

Perusasetusten määrittämisen jälkeen laite voitiin sammuttaa `sudo poweroff` komennolla, irrottaa verkkokaapeli ja siirtää laite haluttuun paikkaan. Langattoman verkon ansioista laite ei enää tarvinnut virtakaapelin lisäksi muita kaapeleita ja se oli helppo sijoittaa ikkunalaudalle. Anturi sijoitettiin ikkunan ulkopuolelle ja laite käynnistettiin, jonka jälkeen päästiin rakentamaan lämpömittarin järjestelmää.

5.4 Lämpömittarin järjestelmän rakentaminen

Esiasennuksen jälkeen lähdettiin ensimmäiseksi asentamaan lämpöanturia, joka edellytti muutoksia *boot* nimisessä kansiossa olevaan *config.txt* tiedostoon. *Config.txt* tiedosto sisältää piirilevyn asetukset, jotka ohjaavat toimintaa käyttöjärjestelmän ja fyysisten komponenttien välillä ydin (kernel) tason tavoin. Tähän tiedostoon lisättiin *dtoverlay=w1-gpio* asetus omalle rivilleen ja tiedosto tallennettiin. Asetus *dtoverlay* sallii pinnin käytön ja *W1-gpio* on oletuksena pinni 4 eli sama pinni, johon lämpöanturin datakaapeli oli aikaisemmin kytketty. Muutosten jälkeen laite käynnistettiin uudelleen `sudo reboot` komennolla ja testattiin anturin toimivuus seuraavilla komennoilla:

Annetaan laitteelle tieto, että pinniin 4 on kytketty laite:

```
sudo modprobe w1-gpio
```

Annetaan myös tieto, että liitetyllä laitteella mitataan lämpötilaa:

```
sudo modprobe w1-therm
```

etsitään laitteen id:

```
ls /sys/bus/w1/devices
```

tarkistetaan, palauttaako laite lämpötilan:

```
cat /sys/bus/w1/devices/<laitteenId> w1_slave
```

Anturin asentamisen jälkeen aloitettiin ohjelmoiminen selvittämällä siihen tarvittavat kirjastot ja työkalut. Kohtuullisen pitkän tiedon etsinnän ja testailun jälkeen todettiin, että projektiin tarvittiin *pip*, *virtualenv*, *Google.cloud-pubsub* ja *Rpi.GPIO* kirjastot. Python kirjasto löytyi käyttöjärjestelmästä jo esiasennettuna ja *pip* tarvittiin python pakettien hallintaan. *Pip* on oleellinen työkalu Python ohjelmointia ja sillä voidaan asentaa kirjastoja ja lisäosia, jotka eivät kuulu Pythonin peruskirjastoon. *Virtualenv* eli virtuaaliympäristö (*Virtual Environment*) on kansio, johon voidaan luoda Python projekti halutuilla kirjastoilla. Kansioon tehty projekti ei huomioi laitteella olevia muita kirjastoja ja näin voidaan välttyä ylimääräisiltä virhetilanteilta. *GPIO* kirjasto mahdollistaa piirilevyn pinnien käytön ja Googlen *pub-sub* kirjastolla voidaan liittää laite aiemmin luodulle *topic_temp01* kanavalle. Kaikki tarvittava asennettiin seuraavilla komennoilla:

Asennetaan *pip*:

```
sudo apt-get install python3-pip
```

Asennetaan *virtualenv* ympäristö:

```
sudo python3 -m pip install --user virtualenv
```

Asennetaan Google Cloud Pub/Sub palvelun kirjasto:

```
pip install google-cloud-pubsub
```

Asennetaan GPIO kirjasto piirilevyn pinnien käyttämiseksi:

```
pip install Rpi.GPIO
```

Kirjastojen asentamisen jälkeen luotiin venv niminen virtuaaliympäristö komennolla: `sudo`

`python3 -m virtualenv venv` ja aktivoitiin ympäristö käyttöön komennolla: `source`

`venv/bin/activate`.

Ohjelmoinnin runko toteutettiin toisella tietokoneella ja siirrettiin testausvaiheessa python tiedostot Raspberryille. Ohjelmisto koostui kahdesta eri tiedostosta, joista kuvassa 13 oleva `sensordata.py` ohjelma lukee anturin arvon ja lopuksi palauttaa sen.

```
1  import os
2  import glob
3  import time
4
5  # valitaan pinni, datan muoto ja haetaan kansiorakenteesta tiedosto, johon arvo kirjoitetaan.
6  os.system('modprobe w1-gpio')
7  os.system('modprobe w1-therm')
8  direct = '/sys/bus/w1/devices/'
9  device_folder = glob.glob(direct + '28*')[0]
10 device_file = device_folder + '/w1_slave'
11
12 def read_r_temp(): #luetaan lämpötila-anturin antama rivi ja palautetaan luettu rivi.
13     f = open(device_file, 'r')
14     lines = f.readlines()
15     f.close()
16     return lines
17
18 def read_temp(): #prosessoidaan lämpötilan arvo ymmärrettävään muotoon ja palautetaan se.
19     lines = read_r_temp()
20     while lines[0].strip()[-3:] != 'YES':
21         time.sleep(0.2)
22         lines = read_r_temp()
23     equals_pos = lines[1].find('t=')
24     if equals_pos != -1:
25         temp_string = lines[1][equals_pos+2:]
26         temp = float(temp_string) / 1000.0
27         return temp
```

Kuva 12. `sensordata.py` ohjelma lukee anturin arvon.

Kuvan 14 `sendData.py` ohjelma ajaa 10 minuutin välein `sensordata.py` ohjelman ja muodostaa saadusta lämpötilan arvosta ja aikaleimasta paketin, jonka ohjelma lähettää lopuksi `home-temp-292518` projektin `topic_temp01` kanavalle. Datan lähettämiseen käytettiin Google Publisher Client API kirjastoa, jolla data lähetettiin TCP ja HTTP-protokollien välityksellä.

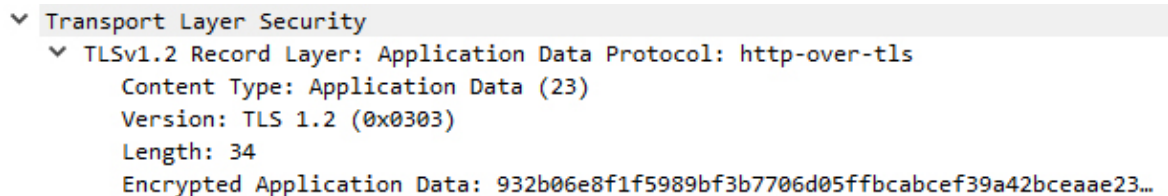
```

1  from google.cloud import pubsub_v1
2  import datetime
3  import json
4  import sensordata
5  import time
6
7  project_id = "home-temp-292518" #Googlen projektin ID
8  topic_name = "topic_temp01" #Googlen Pub/Sub kanavan nimi
9
10 publisher = pubsub_v1.PublisherClient()
11 topic = publisher.topic_path(project_id, topic_name)
12
13 collection = dict()
14
15 while True:
16     time.sleep(600) #10min viive jokaisen paketin lähetyksen välissä
17     temp = round(float(sensordata.read_temp()),2) #haetaan lämpötilan arvo
18     timenow = float(time.time())
19     data = {"time":timenow, "temp" : temp} #tehdään paketti aikaleimasta ja lämpötilan arvosta
20     print(data)
21
22     sendData = publisher.publish( #Julkaistaan data Googlen Pub/Sub kanavalle
23         topic, data=(json.dumps(data)).encode("utf-8"))
24     print("Data lähetetty onnistuneesti.")
25
26 while collection:
27     time.sleep(3)

```

Kuva 13. sendData.py ohjelma lähettää datan Pub/Sub kanavalle.

Ennen datan lähettämistä, avattiin vielä yhteys Googleen aikaisemmin saadulla yksityisnavaimen JSON tiedostolla. Tiedosto siirrettiin laitteelle samaan kansioon ohjelmisto tiedostojen kanssa. Yhteys avattiin viemällä JSON tiedosto Googleen komennolla `export GOOGLE_APPLICATION_CREDENTIALS="<teidostonNimi.json>"`. Ohjelmisto käynnistettiin yhteyden avaamisen jälkeen komennolla `python3 sendData.py`. Yhteyden toimivuus ja datan siirto varmistettiin toimivaksi, jonka jälkeen haluttiin vielä varmistua tietoliikenteen tapahtuvan halutulla tavalla. Wireshark nimisellä analysointiohjelmalla tarkkailtiin laitteen liikennettä ja voitiin havaita kuvassa 15 näkyvästä paketista, että data liikkui TLS ja HTTP-protokollan välityksellä salattuna.



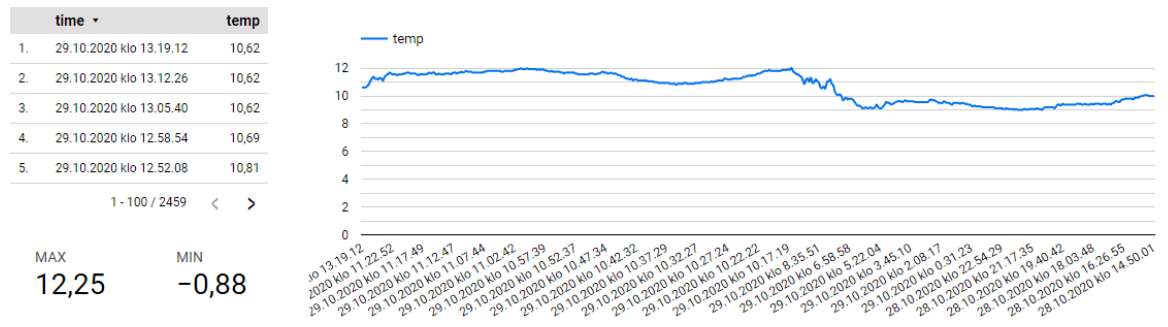
Kuva 14. Wireshark ohjelmalla havaittu paketti.

5.5 Laitteen havainnointi Googlen alustoilta

Lämpömittarin sekä Googlen pilvijärjestelmän rakentamisen ja konfiguroinnin jälkeen voitiin siirtyä tarkastelemaan ja hallinnoimaan luotua projektia. Yleisesti projektia voitiin monitoroida projektin etusivulta sekä IoT Coren välilehdeltä. Edellä mainituista näkymistä voitiin havainnoida esimerkiksi prosessorin kuormitusta, järjestelmän viivettä, rahallisia kustannuksia ja virhe lokia. Virheilmoituksia ei ollut, joten tärkeimpänä voitiin havaita, että tämän projektin kustannukset olivat kokonaisuudessaan 2,6 € vuorokaudessa. Projektin kannalta muuta tärkeää tietoa tutkittiin Pub/Sub, Dataflow ja BigQuery palveluiden monitoroinneista.

Pub/Sub palvelusta voitiin heti ensimmäiseksi huomata graafisesta kaaviosta, että laitteen lähettämä data saapui halutusti kymmenen minuutin välein, eikä virheilmoituksiakaan ollut. Dataflow palvelun monitoroinnissa oli laajemmin dataa kuvattuna, joista tärkeimpinä Dataflow palvelun varoitukset ja virheilmoitukset, dataputkiston eri vaiheiden tila ja putkistossa liikkuvan datan mahdollinen viive. Muuta mielenkiintoista dataa oli esimerkiksi palvelun päällä olo aika, käytettävän keskusmuistin määrä sekä käytössä oleva SDK versio ja mahdollinen saatavilla oleva SDK päivitys. Hyvänä ominaisuutena dataputkistoon olisi voinut määrittää hälytyksiä tapahtuvista poikkeuksista. BigQuery palvelussa voitiin tarkkailla *outdoorTemp* tietokantaa ja sen taulujen dataa. Palvelussa oli myös integroitu SQL ohjelma, jolla olisi voinut hakea dataa tietokannasta. Kaikkia edellä mainittuja palveluita olisi tarvittaessa pystynyt myös muokkaamaan samoista hallinta ja monitorointi näkymistä

Massadataa voitiin käsitellä suoraan BigQueryn SQL käyttöliittymästä taulukko muodossa, mutta yleensä dataa halutaan tarkastella myös visuaalisemmassa muodossa kaavioissa tai eri mallisissa diagrammeissa. Tässäkin työssä haluttiin selkeyttää datan tulkitsemista, jolloin käyttöön otettiin vielä Googlen Data Studio työkalu. Käyttöön otossa kirjaututtiin samalla Gmail tunnuksella Data Studio portaaliin ja valittiin haluttu projekti. Tyhjälle pohjalle valittiin vain datan lähde, joka tässä tapauksessa oli *outdoorTemp* tietokanta taulu. Graafiin valittiin lista viidestä viimeisimmästä lämpötilasta aikaleimojen perusteella, ylin ja alin lämpötila koko projektin ajalta sekä kaavio menneen vuorokauden tuloksista (Kuva 16).



Kuva 15. Data Studio näkymä. Luvut on ilmoitettu celsius- asteissa.

6 Pohdinta

Tämän opinnäytetyön tavoitteena oli tutkia mitä ovat IoT ja Google Cloud Platform ja kuinka ne liittyvät toisiinsa. Teoriaosuuden lisäksi toiminnallisen osuuden tavoitteena oli selvittää miten Raspberry Pi piirilevystä tehdään IoT-laite ja miten se saadaan yhdistettyä Googlen pilvialustalle. Molempien osuuksien tavoitteena oli tarjota lukijalle perustiedot aiheeseen perehtymistä varten sekä kuvata vaatimukset ja prosessit minkä tahansa IoT-laitteen yhdistämiseksi Google Cloud Platform alustalle.

Teoriaosuuden tutkimuksessa saatiin käsitys, että IoT-laitteet voivat olla mitä tahansa laitteita, jotka ovat suorasti tai epäsuorasti yhteydessä internettiin. IoT-laitteiden määrä on jatkuvassa kasvussa ja tulevaisuudessa suurin osa elektroniikasta tulee mitä luultavammin olemaan internetissä. Nopea kasvu ja kehitys tuovat kuitenkin mukanaan myös negatiivisia vaikutuksia kuten verkkohyökkäyksiä ja tietosuojaloukkauksia, jos tietoturvaa ei saada parannettua.

HTTP ja MQTT protokollia käytetään Googlen IoT-palveluissa, joten molemmista tehtiin kattava selvitys, jonka tuloksina voitiin todeta molempien sopivan hyvin IoT käyttöön. HTTP oli Googlen oletus protokolla ja siksi sitä päädyttiin käyttämään toiminnallisessa osuudessakin, vaikka MQTT olisi rakenteeltaan kevyempi. MQTT käyttö luultavasti alentaisi laitteen prosessien kuormitusta ja tekisi pilvenpalvelun käytöstä edullisempaa. Molempien teoriaosuuksien jälkeen aloitettiin Googlen pilvijärjestelmän ja laitteen rakentaminen. Jo teoriaosuudessa todettiin, että Google Cloud Platformin sisältämä Cloud IoT on kattava alusta monen tyyppiseen IoT käyttöön.

Toiminnallista osuutta lähdettiin suunnittelemaan pohtimalla, että millainen laite haluttaisiin tehdä ja mikä olisi tehtävissä opinnäytetyön aikataulun puitteissa. Jo ennen opinnäytetyön aloitusta Raspberry Pi ja IoT-laitteet yleisesti olivat jo kohtalaisen tuttuja aiheita, joten niidenkään opetteluun ei kulunut aikaa. Ulkolämpömittariin päädyttiin sen yksinkertaisuuden vuoksi ja tarvittavat komponentitkin löytyivät Suomesta, joten välttyttiin pitkiltä toimitusajoilta. Raspberry Pi 4 oli ominaisuuksien kannalta ylimitoitettu tähän projektiin, mutta se valikoituikin piirilevyksi lähinnä tulevaisuuden kehitysmahdollisuuksien vuoksi. Teoriaosuuden tutkimus osoittikin, että vastaavan älylämpömittarin olisi voinut toteuttaa myös tehottomammalla Zero W mallilla. Raspberryn jo ennestään tuttu alusta ja Python ohjelmointikielen osaaminen vaikuttivat myös piirilevyn valintaan.

Googlen Cloud Platform alustan päälle IoT-järjestelmän rakentamisen oli suhteellisen helppoa aiheeseen perehtymisen jälkeen ja Googlen dokumentaation avulla. Ennen toiminnallista osuutta tehty teoreettinen tutkimus helpotti Googlen IoT-rakenteen hahmottamista ja kokonaisuuden rakentamista. Hyvästä perehtymisestä huolimatta järjestelmän kokonaisuuden hahmottaminen vei useamman viikon, ennen kuin oltiin varmoja kaikista tähän projektiin tarvittavista palveluista. Googlen alusta IoT käytössä ei ollut ennestään tuttu, joten sekin vaikutti kokonaisuuden ymmärtämiseen. Palveluiden konfigurointi onnistui ilman suurempia ongelmia Googlen hyvän dokumentaation ansiosta. Ainoat ongelmat ilmaantui-
vat Dataflow palvelun kanssa, kun piti selvittää kanavan, tietokanta taulun ja kontin nimet silloin vielä tuntemattomalta alustalta. Kun ensimmäinen Dataflow oli saatu tehtyä, todettiin viikonlopun jälkeen, ettei kaikki ollut ihan kunnossa, sillä parissa päivässä rahaa oli kulunut 50 € verran. Pienen tutkimisen jälkeen huomattiin, että asetuksissa oli unohtunut rajoittaa prosessien määrä, jolloin käynnissä oli samanaikaisesti loputtomasti prosesseja, jotka kuormittivat järjestelmää. Asian korjaamiseksi määritettiin yhden prosessin tapahtuvan kerrallaan, jolloin vuorokautiset kulut laskivat 2,6 euroon.

Työssä edettiin aikataulun mukaisesti, vaikka lämpömittarin rakennusviikkojen aikana tuli-
kin kiire ylimääräisestä työajasta huolimatta. Aikataulun suunnittelussa ei ollut varattu aikaa ongelmien selvittelyyn, joita ohjelmoinnin aikana ilmeni. Muutaman päivän selvittelyn jälkeen laite onneksi saatiin valmiiksi ja yhdistettyä pilveen aikataulun mukaisesti. Kaikki asetetut tavoitteet saavutettiin lopulta viikko etuajassa ja voitiin todeta myös oman osaamisen kehittyneen. Opinnäytetyön aikana opittiin käyttämään Pub/Sub kirjastoa Python ohjelmoinnissa sekä Google Cloud Platform alustaa opittiin käyttämään jo sujuvasti. Teoriaosuudessa oma ymmärrys myös IoT-pilviteknologioiden osalta syventyi.

Internetiin kytketyt laitteet toimivat hienosti Googlen pilven kanssa ja pilvessä olevat palvelut ovat kattavia IoT-laitteiden seuraamiseen ja hallinnointiin. Käyttöliittymäkin oli selkeä ja mitään ei jääty kaipaamaan. Se voitiin kuitenkin todeta, ettei palvelun käyttö yksityiskäytössä ole kovin järkevää, sillä kustannukset ovat melko korkeat. Tämän projektin vuosittaiset kustannukset pilvipalvelun osalta olisivat jopa 950 €. Keventämällä rakennetta esimerkiksi kustomoidulla Dataflow palvelulla ja MQTT protokollalla kustannuksia voitaisiin saada alennettua.

Jatkossa laitetta on tarkoitus kehittää vielä eteenpäin kokonaisen jäähdytysjärjestelmän oletusyhdykäytäväksi. Raspberry Pi piirilevyyn tullaan kytkemään vielä sisälämpömittari sekä infrapunalähetin jäähdytyslaitteen ohjaamiseksi. Järjestelmän käynnistyksestä tulee lisäksi automaattinen eli määritetyt Python ohjelmat käynnistyvät aina laitteen käynnistyessä uudelleen. Pilviratkaisua täytyy kuitenkin vielä selvittää. Mikäli Googlen pilvipalvelun

kuluja ei saada laskettua, täytyy laite yhdistää omaan palvelimeen. Ensin yritetään kuitenkin vaihtaa HTTP-protokolla MQTT-protokollaan ja valmis Dataflow dataputkisto kustomoituun Python pohjaiseen dataputkistoon, jotta sillä saataisiin mahdollisesti pilvirakenteesta riittävän kevyt.

7 Lähteet

Anderson Technologies. 2019. AI: The Danger of IoT Data Collection. Luettavissa: <https://andersontech.com/danger-of-iot-data-collection/>. Luettu: 25.9.2020

Cope, S. s.a. Beginners Guide To The MQTT Protocol. Luettavissa: <http://www.steves-internet-guide.com/mqtt/>. Luettu: 28.9.2020

Fleishman, G. 2020. Take Control of Wi-Fi Networking and Security. Luettavissa: <https://learning.oreilly.com/library/view/take-control-of/9781947282247/>. Luettu: 1.10.2020

F-Secure. 2020a. Asiantuntijat varoittavat: esineiden internet pitää turvata nyt. Luettavissa: <https://www.f-secure.com/fi/press/p/asiantuntijat-varoittavat-esineiden-internet-piittaa-turvata-nyt>. Luettu: 25.9.2020

F-Secure. 2020b. F-Secure SENSE. Luettavissa: <https://www.f-secure.com/fi/home/products/sense/technology>. Luettu: 25.9.2020

Google Cloud. s.a.a Solve more with Google Cloud. Luettavissa: <https://cloud.google.com>. Luettu: 1.10.2020

Google Cloud. s.a.b Google Cloud IoT. Luettavissa: <https://cloud.google.com/solutions/iot>. Luettu: 2.10.2020

Google Cloud. 2020a. Google Cloud Overview. Luettavissa: <https://cloud.google.com/docs/overview>. Luettu: 1.10.2020

Google Cloud. 2020b. Get started with Google Cloud. Luettavissa <https://cloud.google.com/docs> Luettu: 1.10.2020

Google Cloud. 2020c. What is pub/sub? Luettavissa: <https://cloud.google.com/pub-sub/docs/overview>. Luettu: 9.10.2020

Google Cloud. 2020d. Cloud IoT Core overview. Luettavissa: <https://cloud.google.com/iot/docs/concepts/overview>. Luettu 9.10.2020

Google Cloud. 2020e. Cloud Functions Overview. Luettavissa: <https://cloud.google.com/functions/docs/concepts/overview>. Luettu: 10.10.2020

Google Cloud. 2020f. What is BigQuery? Luettavissa:

<https://cloud.google.com/bigquery/docs/introduction>. Luettu: 10.10.2020

Google Cloud. 2020g. Dataflow. Luettavissa: <https://cloud.google.com/dataflow>.

Luettu 15.10.2020

Google Cloud. 2020h. Device Security. Luettavissa: <https://cloud.google.com/iot/docs/concepts/device-security>. Luettu: 15.10.2020

Google Cloud. 2020i. Cloud IoT Core. Luettavissa: <https://cloud.google.com/iot-core>. Luettu: 15.10.2020

Google Cloud. 2020j. Protocols. Luettavissa: <https://cloud.google.com/iot/docs/concepts/protocols>. Luettu: 16.10.2020

lotfinland.net. 2018. Internet of Things tuo säästöjä yrityksille. Luettavissa: <http://iotfinland.net/index.php/2018/10/10/internet-of-things-tuo-saastoja-yrityksille/>. Luettu: 24.9.2020

Kaspersky. s.a. Miksi kotiverkon IoT-tietoturva on tärkeä asia Luettavissa:

<https://www.kaspersky.fi/resource-center/threats/secure-iot-devices-on-your-home-network>. Luettu: 25.9.2020

Kavelová, A. & Dostálek, L. 2006. Understanding TCP/IP. Luettavissa: <https://ebookcentral.proquest.com/lib/haaga/reader.action?docID=944992&query=>. Luettu: 1.10.2020

Ledger, D. A. 2016. mapping of many common protocols and standards used in IoT systems Luettavissa: <https://medium.com/@dledge/making-sense-of-the-myrriad-of-iot-standards-and-protocols-88dc4792ba1f>. Luettu: 28.9.2020

Lujabetoni. 2018. Lujabetonilta läpimurto betonirakenteiden kosteusturvallisuuden hallintaan: IoT -kosteusmittausjärjestelmä kokonaispalveluna koko Suomen markkinoille. Luettavissa: <https://www.lujabetoni.fi/2018/01/08/lujabetonilta-lapimurto-betonirakenteiden-kosteusturvallisuuden-hallintaan-iot-kosteusmittausjarjestelma-kokonaispalveluna-koko-suomen-markkinoille/>. Luettu: 24.9.2020

MDN contributors. 2019. An overview of HTTP. Luettavissa: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>. Luettu: 30.9.2020

MQTT. 2020. FAQ. Luettavissa: <https://mqtt.org/faq/>. Luettu: 28.9.2020

MQTT. s.a. MQTT Publish/Subscribe Architecture. Luettavissa: <https://mqtt.org/>. Luettu: 2.10.2020

Raspberry Pi Foundation. s.a. FAQs. Luettavissa: <https://www.raspberrypi.org/documentation/faqs/#introduction>. Luettu: 9.10.2020

Raspberry Pi Foundation s.a. GPIO. Luettavissa: <https://www.raspberrypi.org/documentation/usage/gpio/>. Luettu 9.10.2020

The Vergecast. 2019. CES: Privacy and smart TVs with Vizio CTO Bill Baxter. Podcast. Kuunneltavissa: <https://megaphone.link/VMP4565454603>. Kuunneltu: 25.9.2020

Tietotekniikan termitalkoot. 2017. esineiden internet. Luettavissa: http://www.tsk.fi/tsk/termitalkoot/hakemistot-267.html?page=get_id&id=ID335&vocabulary_code=TSKTT. Luettu: 24.9.2020