Miika Kärkkäinen

Datan keräys MQTT-viestiprotokolalla Beckhoff PLC-ympäristöstä pilvipalveluun

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikan tutkinto-ohjelma

Insinöörityö

17.05.2022

Tiivistelmä

Tekijä(t):	Miika Kärkkäinen
Otsikko:	Datan keräys MQTT-viestiprotokolalla Beckhoff PLC-ympäristöstä pilvipalveluun
Sivumäärä:	43 sivua
Aika:	17.05.2022
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Ammatillinen pääaine:	Automaatiotekniikka
Ohjaaja(t):	Yliopettaja, Erkki Räsänen Vanhempi Insinööri, Mahbub Rahman

---

Tämän työn tarkoituksena oli modernisoida kohdeyrityksen datan keräys, yrityksen automaatiojärjestelmistä. Datan keräyksen modernisointi toteutettiin käyttäen MQTT-viestintäprotokollaa, jolla tämä haluttu data lähetetään rajapintalaitteen avulla pilvipalveluun datan jatkokäsittelyä varten.

Työssä ensin tutkitaan MQTT-viestintäprotokollan toimintaa ja rakennetta ruohonjuuritasolla. Työ etenee tämän jälkeen tutustumaan, millaisia pilvipalveluratkaisuja löytyy ja on olemassa. Tässä osiossa myös nähdään näiden eri pilvipalveluratkaisujen erot, hyödyt sekä haitat. Työn tarkoituksena on vain saada dataa ulos kohdeyrityksen järjestelmistä, eli datan käsittely pilvessä sekä pilvi-infrastruktuurin tietäminen ei ole tälle työlle kriittistä, mutta kuitenkin hyvä tietää. Niin sanotun teoriaosuuden jälkeen työssä päästään tutkimaan käytettyjä laitteita ja ohjelmistoja. Ohjelmisto-osuus huipentuu viimeisen kappaleeseen, jossa käydään tässä työssä paljon käytettyä Beckhoffin IoT-kirjastoa tarkemmin lävitse.

Työn lopputuloksena on kohdeyritykselle rakennettu uusi datankeräysohjelmisto, joka tulee korvaamaan nykyisen datankeräyksen.

Avainsanat: MQTT, Datankeräys, Beckhoff

Abstract

Author(s):	Miika Kärkkäinen
Title:	Data acquisition using MQTT messaging protocol from Beckhoff PLC environment to cloud service
Number of Pages:	43 pages
Date:	17 May 2022
Degree:	Bachelor of Engineering
Degree Programme:	Electrical and Automation Engineering
Professional Major:	Automation Engineering
Instructor(s):	Mahbub Rahman, Senior Engineer Erkki Räsänen, Senior Teacher

---

The purpose of this thesis work was to modernize the target company's data collection from company's automation systems. The modernization of the data collection was carried out using the MQTT communication protocol, by which this desired data is sent to the cloud service via edge device for further data processing.

In this thesis, we first examine the operation and structure of the MQTT communication protocol at the grassroots level. Then we proceed to explore what kind of cloud service solutions can be found and exist. This section also shows the differences, advantages, and disadvantages of these different cloud service solutions. The purpose of this thesis work is only to get data out of the target company's systems, not processing the data in the cloud, so knowing the cloud infrastructure is not critical for this work but is still good to understand. After the theoretical part, we shall continue to study all the used equipment and software in this thesis work. The software section culminates in the last paragraph, which takes a closer look at the much-used Beckhoff IoT library in this work.

The end result of the work is a new data collection software built for the target company, which will replace the current data collection system.

Keywords: MQTT, Data acquisition, Beckhoff

Sisällys

Lyhenteet

API	Application Programming Interface eli Sovellusohjelmointirajapinta on yhteys tietokoneiden tai tietokoneohjelmien välillä.
AWCS	Automatic Waste Collection System. Automaattiset jätteenkeräysjärjestelmät.
Broker	Välittäjä. Eräänlainen palvelin, joka reitittää julkaistut MQTT-protokollan viestit tilaajalle.
Client	Asiakas. Tiedonsiirrossa palvelua käyttävä laite.
GUI	Graphical User Interface. Graafinen käyttöliittymä.
HMI	Human Machine Interface, eli ihmisen ja koneen välinen yhteys. Automaatiojärjestelmissä tällä tarkoitetaan järjestelmän graafista käyttöliittymää.
IaaS	Infrastructure as a Service eli infrastruktuuri palveluna. Tämä on yksi monesta pilvipalvelutyypistä.
IIS	Internet Information Services on Microsoftin kehittämä palvelinohjelmistokokonaisuus, joka on tarkoitettu käytettäväksi Windows-pohjaisissa palvelimissa.
I/O	Input/Output. Ohjelmoitavien logiikoiden tulot ja lähdöt.
IoT	Internet of Things. Esineiden internet.
IPsec	IP Security Architecture on joukko TCP/IP-perheeseen kuuluvia tietoliikenneprotokollia Internet-yhteyksien turvaamiseen.

IIot	Industrial Internet of Things. Teollisuuden esineiden internet.
ISO	International Organization for Standardization eli kansainvälinen organisaatio standardeille on kansainvälinen standardisoimisjärjestö.
JSON	JavaScript Object Notation on yksinkertainen avoimen standardin tiedostomuoto tiedonvälitykseen. Nimestään huolimatta tämä on JavaScriptistä täysin riippumaton.
M2M	Machine to Machine eli laitteelta laitteelle. Tämä termi tarkoittaa kahden tai useamman laitteen välistä suoraa tiedonsiirtoa riippumatta siitä, mitä tietoliikennekanavaa käyttää.
MQTT	Aikaisemmin tunnettu nimeltä Message Queue Telemetry Transport. Avoimeen lähdekoodin perustuva julkaisija/tilaajamallin mukainen tiedonsiirtoprotokolla.
OASIS	Organization for the Advancement of Structured Information Standards. Yhteisö, joka määrittää standardeja avoimen lähdekoodin ratkaisuille.
OSI	Open Systems Interconnection Reference Model, eli avointen järjestelmien yhteenliittämisen vertailumalli. Tämä malli kuvaa tiedonsiirtoprotokollien yhdistelmän seitsemällä kerroksella. Kukin kerroksista käyttää yhtä alemman kerroksen palveluja ja tarjoaa palveluja yhtä kerrosta ylemmäs.
PaaS	Platform as a Service eli alusta palveluna. Tämä on yksi monesta pilvipalvelutyypistä.
PC	Personal Computer, eli henkilökohtainen tietokone.

PING	PING on TCP/IP-protokollaan kuuluva työkalu, jonka avulla voidaan tarkistaa kohdelaitteen saatavuus tietoverkossa. Tätä käyttäessä saa käyttäjä myös paluuviestissä informaatiota yhteyden toiminnasta, kuten edestakaisen liikenteen ajan millisekunneissa.
PLC	Programmable logic controller. Ohjelmoitava logiikka.
QoS	Quality of Service. Palvelun laatu tällä viitataan yleensä tiedonsiirron laatuun.
SaaS	Software as a Service eli ohjelmisto palveluna. Tämä on yksi monesta pilvipalvelutyypistä.
SCADA	Supervisory Control And Data Acquisition, eli valvontavalvonta ja tiedonhankinta on tietokoneohjelmistotyyppi, joka on tunnettu valvomo-ohjelmisto. Tämän ohjelmiston avulla luodaan automaatiojärjestelmille graafinen valvomokäyttöliittymä.
SSL	Secure Socket Layer. Sovellusten tiedonsiirtoon käytetty salausprotokolla.
TCP/IP	Transmission Control Protocol / Internet Protocol. Tiedonsiirrossa käytettävä protokollien pino.
TLS	Transport Layer Security. Sovellusten tiedonsiirtoon käytettävä salausprotokolla.
TwinCAT	Twin Control Automation Technology. Beckhoff-automaation kehittämä ohjelmointiympäristö.
VPN	Virtual Private Network eli virtuaalinen erillisverkko. Palvelu, joka suojaa internetyhteytesi ja yksityisyytesi verkossa.

WCF	Windows Communication Foundation eli Windowsin kommunikaatio perusta on osa Microsoftin .NET-sovelluskehystä. WCF:n avulla sovellukset voivat lähettää dataa asynkronisesti päätepisteiden välillä.

1 Johdanto

Tämä insinöörityö tehtiin MariElectronics Oy:lle. Tarkoituksena on päivittää yrityksen nykyinen datankeräysjärjestelmä. Tämä päivitys rakennettiin käyttämällä MQTT-viestintäprotokollaa. MQTT-protokollan avulla haluttu data tallennetaan pilvipalveluun, jossa tätä kerättyä dataa jatkojalostetaan niin tuotekehityksen tarpeisiin kuin asiakaslaskutukseen.

Yrityksen nykyiset järjestelmät ovat vanhentumaan päin, kun Microsoft ilmoitti lopettavansa Silverlightin tukemisen. Nykyinen datan keräys on toteutettu SCADA-käyttöliittymän toimesta, joka pohjautuu WCF-palveluun, joka on englanniksi Windows Communication Foundation Service. Tämän järjestelmän GUI-osa on rakennettu käyttämällä Silverlight-tekniikkaa, jonka tukemisen Microsoft lopettaa. SCADA-palvelu toimii IIS:n pohjalla, joka tulee englannin kielen sanoista Internet Information Service.

Yrityksessä päädyttiin rakentamaan käyttöliittymät PLC-valmistajien HMI-työkalujen avulla. Tavoitteena on PLC:n ja HMI:n saumaton toiminta. SCADA-järjestelmästä luopuminen tarkoitti myös tiedonkeruujärjestelmän menettämistä. Yrityksessä päädyttiin tiedonkeruujärjestelmän uudelleenrakentamisesta pilvipohjaiseksi samalla käyttäen MQTT-viestintäprotokolaa.

Työn tavoitteena on tutustua Beckhoffin Tc3-IoT-koodikirjastoon, MQTT-viestintäprotokollaan ja miten näiden avulla saadaan lähetettyä dataa pilveen Siemensin Industrial Edge-laitteen kautta.

Tämän työn luvussa 2 tutkitaan MQTT-viestintäprotokolarakenne, ominaisuudet, sekä käydään myös MQTT:n historiaa lyhyesti lävitse. Luvussa 3 tehdään pintaraapaisu eri pilvipalveluratkaisuista, käydään lävitse niiden eroja keskenään ja mitä hyötyjä ja mahdollisia haittoja näillä on. Kappaleen 3 tutkimus jää vähäiselle, koska aihe ei suoranaisesti kuulu tähän työhön, mutta aiheesta on kuitenkin hyvä olla jonkinlainen käsitys.

Luvussa 4 käydään lävitse työssä olennaisia laitteita ja ohjelmia. Tässä luvussa myös syvennytään kuumaan aiheeseen, tietoturva ja miten tämä kerätty data saadaan turvallisesti perille pilvipalveluun.

Luvussa 5 päästäänkin lähentymään tämän työn toteutua. Viidennessä luvussa tutkitaan Beckhofffin IoT-kirjastoa, johon kuuluvat kaikki, mitä MQTT-kommunikaation tarvitsee Beckhoffin PLC-ympäristössä. Valmiin ja toimivan ohjelmakoodin sisältämien arkaluonteisten tietojen takia tässä luvussa käsitellään vain näitä valmiita Beckhoffilta saatavia kirjastoja. Valmis toimiva ohjelmakoodi sisältää aiheita ja informaatiota, jonka paljastuminen voi vaarantaa yrityksen järjestelmien toiminnan, joten tässä työssä ei niitä esitetä.

MariElectronics Oy on osa MariGroup-konsernia. Konsernin yhtiöiden valmistamia tuotteita ja järjestelmiä ovat muun muassa sähköiset hintalaput, turvallisuus- sekä terveysjärjestelmät ja AWCS-järjestelmät, eli automaattiset jätteenkeräysjärjestelmät. Tämän jätejärjestelmän brändinimenä on MetroTaifun ja järjestelmän tarkoituksena on uudistaa, tehostaa, sekä automatisoida kotitalousjätteen keräystä. Järjestelmän toiminta perustuu jätteen imemiseen alipaineella putkistoa pitkin taloyhtiöiden jätehuoneista keskitettyyn jätekeräysaseman jätekontteihin. Näin ollen kotitalousjätteen kerääminen tapahtuu vain yhdestä keskitetystä paikasta, mikä vähentää roska-autojen liikennöintiä ahtailla kaduilla. [1.]

2 MQTT-viestintäprotokolla

MQTT on avoin OASIS- ja ISO-standardi (ISO/IEC 20922) viestintäprotokolla, joka mahdollistaa tietojen vaihdon kahden koneen välillä ilman suoraa viestintäyhteyttä niiden välillä. Tämä protokolla soveltaa julkaisija/tilaaja mallia, jossa asiakas voi lähettää tai vastaanottaa dataa tai molempia. [2.]

Rakenteeltaan kevyt, avoin sekä helppo ottaa käyttöön tekee tästä protokollasta ihanteellisen tilanteisiin, jossa tarvitaan luotettavaa tiedonsiirtoa pienellä verkonkaistalla sekä koodijalanjäljellä. Näitä rajoitettuja ympäristöjä ovat muun

muassa koneelta koneelle tiedonsiirto eli M2M ja Internet of Things eli Iot. Molemmissa ympäristöissä koodin suorittamisen keveys suorittimelle sekä verkolle on enemmän kuin suotavaa. Näiden ominaisuuksien ansiosta MQTT:tä käytetään nykyään monilla eri aloilla, kuten autoteollisuudessa, teollisuudessa, televiestinnässä, öljy- ja kaasuteollisuudessa. [3; 4.]

2.1 Historia

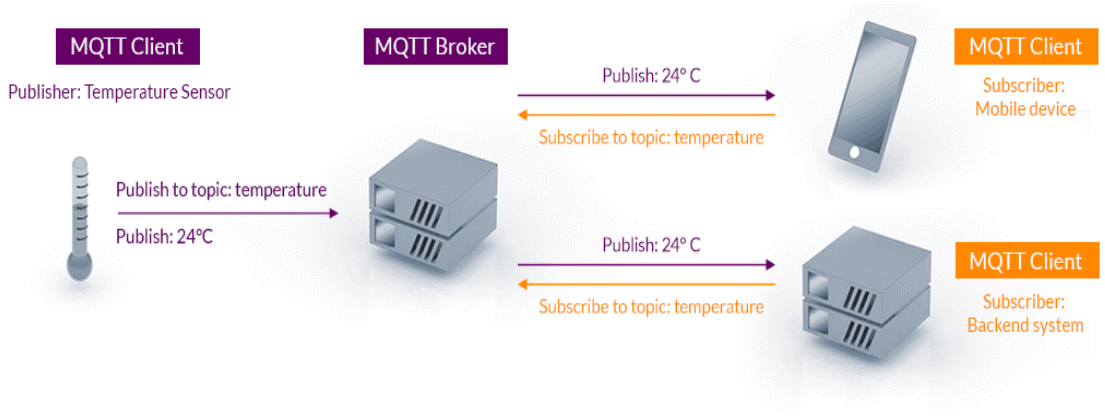
MQTT-protokollan keksivät vuonna 1999 Andy Stanford-Clark ja Arlen Nipper, koska he tarvitsivat kevyen viestintä-protokolan kyetäkseen kommunikoimaan öljyputkien kanssa satelliittien välityksellä. Andy Stanford-Clark ja Arlen Nipper määrittivät useita vaatimuksia tulevalle protokollalle, jotka ovat vielä tänäkin päivänä MQTT:n kehitystyössä tärkeitä kulmakiviä. Näitä tärkeitä vaatimuksia ovat [4]:

- viestintäprotokollan helppo käyttöönotto sekä käyttäminen
- lähetettyjen viestien laatutietotoimitukset
- kevyt, sekä koodia suorittavalle laitteelle että verkolle
- tietoisuus toiminnan vakaudesta
- protokolla ei rajoita viestien hyötykuorman rakennetta, eli protokollan täytyy olla data-agnostinen.

Vaikka näiden alkuperäisten vaatimuksien ollessa kehitys työn keskiössä, MQTT:n pääkäyttötarkoitus on muuntautunut omistetuista sulautetuista systeemeistä avoimiin esineiden internetin käyttötarkoituksiin. Tämä muutos käyttötarkoituksessa on myös herättänyt kysymyksiä, mistä MQTT on lyhenne. Vastaus tähän on, että MQTT ei ole lyhenne vaan nykyisin protokollan nimi. Aikanaan protokolla on saanut nimensä sanoista MQ Telemetry Transport, jossa termi MQ viittaa IBM:n MQ-väliohjelmistotuoteperheeseen. Protokollan käytön yleistyessä muuntui nimi Message Queuing Telemetry Tranportiksi ja nykyisin protokollan nimi on vain MQTT. [4; 5.]

2.2 Toiminta

MQTT:n toiminta perustuu tilaaja/julkaisi-malliin, joka on englanniksi käännettynä publisher subscriber model. Tämä kommunikaatioarkkitehtuuri on tututummasta asiakaspalvelinmallista poikkeava siten, että tämä irrottaa viestejä lähettävän asiakkaan kokonaan muista asiakkaisista, kun taas asiakaspalvelinmallissa palvelimeen yhdistyvä asiakas kommunikoi suoraan päätepisteen kanssa. MQTT:n tilaaja julkaisija mallissa yksikään asiakas laite ei ole tietoinen toisesta asiakkaasta, koska kaikki kommunikaatio tapahtuu kolmannen osapuolen eli välittäjän kautta, englanniksi Broker. Viestiarkkitehtuuria on havainnoitu kuvassa 1. [3; 6; 7.]



Kuva 1. MQTT-viestintäprotokolan arkkitehtuuri [3].

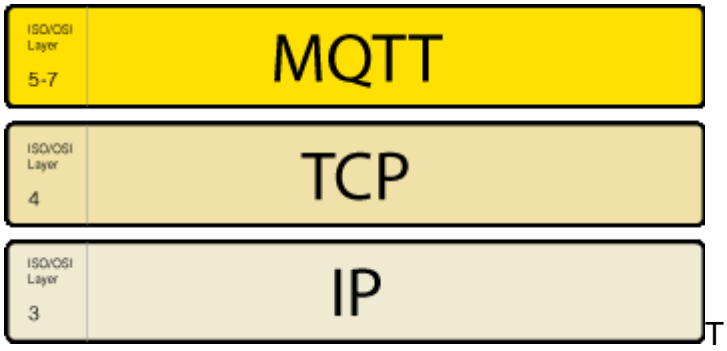
Kuvassa 1 kuvan vasemmassa reunassa oleva julkaisija-asiakas kommunikoi oman lämpötilansa kaiken keskellä olevaan välittäjään. Tähän välittäjään ovat yhteydessä kaikki tilaaja-asiakkaat, jotka saavat välittäjältä tiedon julkaisijan lämpötilasta. Tämä kuva havainnollistaa hyvin MQTT-viestintäprotokollan asiakkaiden välistä anonyymisyyttä, koska kaikki asiakkaat ovat vain yhteydessä keskellä olevaan välittäjään.

2.2.1 TCP/IP ja MQTT

TCP/IP on lyhenne sanoista Transmission Control Protocol/Internet Protocol, se on sarja viestintäprotokollia, joita käytetään verkkolaitteiden yhdistämiseen

Internetissä. Tämä protokolla määrittää, kuinka dataa siirretään internetissä tarjoamalla päästä päähän -viestintätekniikkaa. TCP/IP-protokolla määrittää, kuinka data saadaan pilkottua paketeiksi, jotka voidaan lähettää verkon yli toiselle laitteelle niin, että vastaanottava laite kykenee rakentamaan niistä tämän alkuperäisen lähetetyn viestin. TCP/IP-protokolla on rakennettu niin, että se olisi mahdollisimman omatoiminen ja kykenevä automaattiseen toipumiseen minkä tahansa verkkoon kytketyn laitteen aiheuttamista vioista. [8.]

MQTT-viestiprotokolla toimii TCP/IP-tietoliikenneprotokollan päällä ja sijoittuu siis OSI-mallin kerroksiin 5–7, kuten kuvassa 2 on esitetty. OSI tulee englannin kielen sanoista Open Systems Interconnection Reference Model [9].



Kuva 2. MQTT-protokolla toimii OSI-mallin kerroksissa 5–7 [9].

TCP/IP-protokollan päällä toiminimenen tarkoittaa sitä, että MQTT-viestintäprotokolla saa käyttöönsä kaikki TCP/IP-protokollan toiminnot niin datapakettien lähettämisestä ja vastaanottamisesta kuin virheenkorjaustekniikasta. TCP/IP-protokollan yhtenä tunnettuna ongelmana on laitteiden välisen kommunikoinnin synkronoinnin menetys. Syynä tälle on toisen laitteen kommunikaatio-ongelmat tai yksinkertainen kaatuminen. Tätä epätäydellisen yhteyden tilaa kutsutaan puoliavoimeksi yhteydeksi, englanniksi half-open connection. Puoliavoimen yhteyden aikana toimintakunnossa oleva laite yrittää toimia normaalisti, koska ei ole tietoinen toisen laitteen ongelmista. Lähetetyt paketit eivät koskaan päädy perille, joten tämä data on menetetty. Tämän ongelman ratkaisuksi MQTT:hen on lisätty pidä elossa -ominaisuus, englanniksi keep alive. Tämä ominaisuus varmistaa välittäjän ja asiakkaan

välisen yhteyden olevan auki sekä tekee välittäjän ja asiakkaan tietoiseksi yhteydestään. [10.]

Asiakas aloittaa luomalla TCP/IP-yhteyden välittäjään käyttämällä joko vakioporttia tai välittäjän operaattorien määrittelemää mukautettua porttia. Vakioportit ovat 1883 salaamattomalle tiedonsiirrolle ja 8883 salatulle SSL/TLS-tiedonsiirrolle. SSL tulee sanoista Secure Sockets Layer, TLS tulee sanoista Transport Layer Security. SSL/TLS-kättelyn aikana asiakas vahvistaa palvelimen varmenteen ja todentaa palvelimen. Asiakas voi myös antaa välittäjälle asiakasvarmenteen kättelyn aikana. Välittäjä voi käyttää tätä asiakkaan todentamiseen. Vaikka se ei ole erityisesti osa MQTT-spesifikaatiota, on tämä tullut tavaksi, että välittäjät tukevat asiakkaan todennusta SSL/TLS-asiakaspuolen varmenteilla. [11.]

MQTT-protokollan tarkoituksena on toimia myös resurssirajoitteisille ja IoT-laitteille, joten SSL/TLS-kättely ei välttämättä ole aina vaihtoehto. Tällaisissa tilanteissa todennus esitetään selkeätekstisenä käyttäjänimenä ja salasanana, jotka asiakas lähettää välittäjälle CONNECT/CONNACK-viestipaketteihin yhdistettynä. Näistä viestipaketeista CONNECT ja CONNACK on lisää luvussa 2.3.2. [11.]

2.2.2 Asiakas

MQTT-protokollassa asiakaslaitteilla on seitsemän eri komentoa käytettävänä. Nämä ovat yhdistäytyminen (CONNECT), julkaisu (PUBLISH), tilaus (SUBSCRIBE), peruutus (UNSUBSCRIBE), uloskirjautuminen (DISCONNECT), elossa pyyntö eli ping (PINGREQ) sekä varmistuksien vaihto (Auth). Viestien julkaisussa on lähetettyjen viestien laatutasoon liittyvät kuittauskomennot julkaisuviestin tunnustus (PUBACK), julkaisuvastaanotettu (PUBREC), julkaisu vapautettu (PUBREL) ja julkaisuvalmis (PUBCOMP). Näistä on lisää luvussa 2.3. [12; 13; 14]

Näistä kaikista vaihtoehdoista päätoiminnoiksi voisi sanoa viestienjulkaisua ja viestientilaamista välittäjältä. Viestien julkaisussa asiakas lähettää viestipaketin

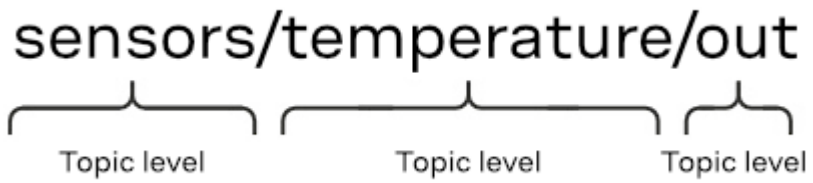
palvelimelle, jossa toimii välittäjä eikä suoraan toiselle asiakkaalle. Välittäjä hallitsee viestien eteenpäin lähettämistä käyttämällä viestipaketeissa esiintyvän aiheen avulla, englanniksi topic. Asiakas, joka haluaa vastaanottaa viestejä, joutuu MQTT-protokollan arkkitehtuurin takia tilaamaan viestit välittäjältä. Saadakseen vain haluamansa viestit asiakas tilaa nämä aiheita käyttämällä. [13; 14.]

2.2.3 Välittäjä

MQTT-protokollassa välittäjä vastaa yhteyksien, istuntojen ja tilausten hallinnasta. Päävastuuna on vastaanottaa kaikki julkaistut viestit ja lähettää ne niitä tilaaville asiakkaille. Välittäjä myös asettaa tarpeen vaatiessa viestejä jonoon niitä tilaaville asiakkaille ja lähettää ne sovitun laatutason eli QoS-tason mukaisesti. [13.]

2.2.4 Aiheet

MQTT-viestit julkaistaan ja tilataan käyttämällä viestien aiheiden avulla. Aiheet rakentuvat yhdestä tai useammasta kerroksesta, nämä kerrokset erotetaan toisistaan vinoviivan avulla (/). Aiheet ovat kirjainkoon herkkiä eikä näitä tarvitse ennakkoalustaa välittäjälle. Asiakas voi siis vain lisätä aiheen viestiin, välittäjä vastaanottaa tämän. Kuvassa 3 on mallinnettu viestipaketin aiheen rakenne. [13; 15.]



Kuva 3. MQTT-viestin aiheen rakenne [11].

Aiheitten käyttäminen mahdollistaa MQTT-protokollan sisäänrakennetun tavan järjestää dataa, kun se saapuu määränpäähänsä. Suuremmissa verkoissa tämä

hyöty alkaa näkymään kunnolla. Esimerkkinä tästä on järjestelmä, jossa on dataa lähettäviä asiakkaita useassa eri paikassa. Kaikkien viestejä julkaisevien laitteiden viestit voidaan laittaa yhteen ja jäsentää vasta määränpäässä tai viesteissä voidaan käyttää aiheita useassa kerroksessa ja ennalta jäsentää aiheitten perusteella ennen määränpäähän saapumista. [13; 14.]

Viestien tilauksissa voidaan käyttää aiheitten suodatusominaisuuksia. Näitä kutsutaan MQTT-protokollassa jokerimerkeiksi, englanniksi wildcard. Nämä suodatusmerkit tulevat aiherakenteeseen, käytetyt merkit ovat plusmerkki (+) ja ristikkomerkki (#). Plusmerkki on yksitasoinen jokerimerkki, joka vastaa mitä tahansa tietyn aihetason nimeä. Voimme käyttää tätä jokerimerkkiä tilataksemme tietyn aihetason viestit. Esimerkkinä järjestelmä tallentaa lentokoneiden laskeutumisia Euroopassa MQTT-protokollaa käyttäen. Data julkaistaisiin aiherakenteella: ”valtio/kiitoradan_tunnus/lentokentän_nimi”. Helsinki-Vantaa-lentoaseman kaikki laskeutumiset saataisiin tilaamalla aihe ”Suomi/+/Helsinki-Vantaa”. Ristikkomerkki on monitasoinen jokerimerkki, jota voimme käyttää vain aihesuodattimen lopussa viimeisenä tasona. Tämän jokerimerkin # avulla voimme siis tilata kaikki aiheet, jotka jäävät merkin oikealle puolelle. Esimerkkinä käytetään samaa kuin aikaisemmin ja nyt halutaan saada kaikki tieto laskeutuneista lentokoneista Suomessa, olisi aihe ”Suomi/#”. [11; 16.]

Lisäksi on olemassa kolmas erikoisaihemerkki, dollarimerkki ($). Tämä merkki sulkee aiheen pois kaikista juurijokerimerkkitilauksista. Yleensä $ käytetään palvelinkohtaisten tai järjestelmäviestien kuljettamiseen eikä niinkään käyttäjän määrittämässä kommunikaatiossa. [11.]

2.3 Viestipaketit

MQTT-protokolla on binääripohjainen protokolla, jossa ohjauselementit ovat binääritavuja eivätkä tekstimerkkijonoja. Aiheiden nimet, asiakastunnus, käyttäjätunnukset ja salasanat on koodattu UTF-8-merkkijonoina. MQTT-protokolla käyttää viestinnässä komentoa ja komentokuittausmuotoa, mikä

tarkoittaa sitä, että jokaiseen lähetettyyn viestipakettiin liittyy vastauskuittaus. Kuvassa 4 on esitetty tätä asiakkaan ja välittäjän edestakaista viestintää. [14; 17.]

Kuva 4. MQTT-protokollan vuorovaikutteinen viestintä [14].

Kuvassa 4 näkyy vasemmalla asiakkaan lähettämät viestipaketit ja oikealla näkyvän välittäjän vastaukset näihin. Tässä näkyy, kuinka asiakkaan yhteysviestipakettia seuraa välittäjältä yhteyden vahvistusviesti. Asiakkaan lähettämää tilausviestiä seuraa tilauksen vahvistusviesti ja julkaisuviestiä seuraa julkaisun vahvistusviesti. Tämä mekanismi on samantapainen kättelymekanismi kuin TCP-protokollassa. [14.]

MQTT-viestintäprotokolla koostuu 15 erityyppisestä viestipaketista. Taulukossa 1 on esitetty protokolassa käytetyt viestipaketit. Taulukko on muokattu OASISjärjestön, englanniksi Organization for the Advancement of Structured Information Standards, sivuilta löydettävästä MQTT-standardia käsittelevästä dokumentista. [12.]

Taulukko 1. MQTT-v5.0 protokollan viestipaketit [12].

Nimi	**Viestin Suunta**	**Selvennys**
Reserved	Kielletty	Varattu
CONNECT	Asiakkaalta Serverille	Asiakas haluaa yhdistää
CONNACK	Serveriltä Asiakkaalle	Yhteyden kuittaus
PUBLISH	Asiakkaalta Serverille tai Serveriltä Asiakkaalle	Viestin julkaisu
PUBACK	Asiakkaalta Serverille tai Serveriltä Asiakkaalle	Julkaisun kuittaus QoS 1
PUBREC	Asiakkaalta Serverille tai Serveriltä Asiakkaalle	Julkaiu QoS 2
PUBREL	Asiakkaalta Serverille tai Serveriltä Asiakkaalle	Julkaisun vapautus QoS 2
PUBCOMP	Asiakkaalta Serverille tai Serveriltä Asiakkaalle	Julkaisu valmis QoS 2
SUBSCRIBE	Asiakkaalta Serverille	Tilauksen pyyntö
SUBACK	Serveriltä Asiakkaalle	Tilauksen kuittaus
UNSUBSCRIBE	Asiakkaalta Serverille	Tilauksen lopetus pyyntö
UNSUBACK	Serveriltä Asiakkaalle	Tilauksen lopetuksen kuittaus
PINGREQ	Asiakkaalta Serverille	PING pyyntö
PINGRESP	Serveriltä Asiakkaalle	PING vastaus
DISCONNECT	Asiakkaalta Serverille	Asiakas kirjautuu ulos
AUTH	Asiakkaalta Serverille tai Serveriltä Asiakkaalle	Varmistuksien vaihto

MQTT-protokollassa on käytännössä rakenteeltaan kahdenlaisia viestipaketteja. On viestipaketteja, joilla on hyötykuorma, viestipaketteja, joilla ei ole hyötykuormaa. Hyötykuormallisia paketteja ovat yhteydenotto (CONNECT), viestin julkaisu (PUBLISH), aiheitten tilaus (SUBSCRIBE), tilauksen tunnustus (SUBACK), aihetilauksen peruutus (UNSUBSCRIBE) ja aihetilauksen peruutustunnustus (UNSUBACK). Näitä kutsutaan hyötykuormallisiksi viestipaketeiksi, koska käyttäjä voi vaikuttaa niiden sisältöön. Kaikki muut viestit ovat hyötykuormattomia viestipaketteja, joihin ei käyttäjä voi vaikuttaa sisällöllisesti. [12.]

2.3.1 MQTT-viestipakettien rakenne

MQTT-protokollan viestipaketit rakentuvat kolmesta osasta. Nämä kolme osaa ovat 2-tavuinen aina läsnästä kiinteä otsikko osioista, muuttuva otsikko osiosta sekä hyötykuormaosiosta. Koko MQTT-viestipaketin maksimihyötykuorman koko on rajoitettu 256 megatavuun. Laatutaso eli QoS-taso ei rajoita hyötykuormassa olevan viestin pituutta, eli laatutasosta riippumatta alkuperäisen julkaisuviestin koko voi olla 256 megatavuinen. [11; 12; 17]

Muuttuva otsikko sekä hyötykuormaosuus ei ole aina kaikissa viesteissä mukana. MQTT-protokollan viestipakettirakenteet ovat [17]:

- kiinteä otsikko (ohjauskenttä + paketin pituus kenttä), esimerkkinä CONNACK-paketti

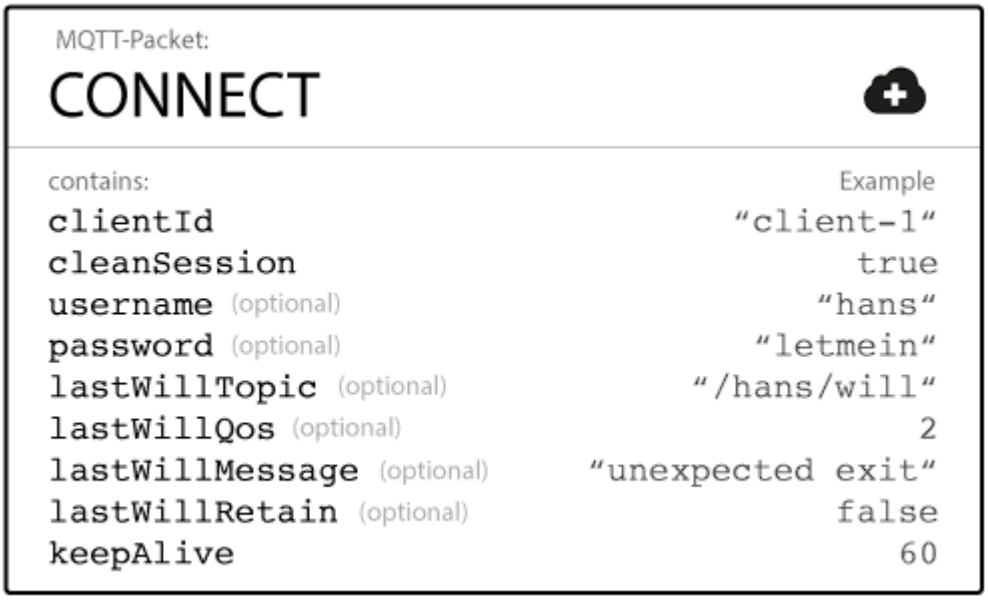
- kiinteä otsikko (ohjauskenttä + paketin pituus kenttä) + muuttuva otsikko, esimerkkinä PUBACK-paketti

- kiinteä otsikko (ohjauskenttä + paketin pituus kenttä) + muuttuva otsikko + hyötykuorma, esimerkkinä tästä on CONNECT-paketti.

Viestinvälityksen asetusten määrityksen kannalta tärkeimmät hyötykuormalliset viestipaketeista ovat CONNECT ja PUBLISH. Asiakas lähettää paketin välittäjälle, kun asiakas luo ensimmäisen kerran yhteyden viestinvälittäjään. [9.]

2.3.2 CONNECT ja CONNACK – Yhteydenluontiviestit

Asiakas aloittaa yhteyden luomisen lähettämällä CONNECT-viestipaketin välittäjälle. Välittäjän saadessa yhteydenottoviestin asiakkaalta, vastaa välittäjä asiakkaalle CONNACK-viestillä kertoakseen asiakkaalle, onko yhteydenotto hyväksytty vai ei. Tämä CONNECT-viestipaketti rakentuu kuvan 5 mukaisesti. [9.]



Kuva 5. MQTT-protokollan CONNECT-viestipaketin rakenne [9].

Kuvassa 5 esitetty CONNECT-viestipaketti lähetetään asiakkaalta välittäjälle ja tällä paketilla määritetään tuleva kommunikaatio välittäjän kanssa. Viestipaketti koostuu useasta eri määritteestä, joista toiset ovat vapaaehtoisia ja toiset eivät. Alla on listattu nämä määritteet, jotka kuuluvat CONNECT-viestipakettiin kuten kuvassa 5 on esitetty [9; 18.]:

- ClientID eli asiakkaan tunnusmäärite kertoo välittäjälle, kuka asiakas on kyseessä. Tämän arvo täytyy olla uniikki kaikille samaan välittäjään yhteydessä olevilla asiakkailla.

- CleanSession flag eli puhtaan istunnon lippu kertoo välittäjälle, millaisen istunnon asiakas haluaa. Tätä on avattu lisää luvussa 2.4.3.

- Username & password, eli käyttäjänimi- ja salasanakentät. Nämä ovat vapaaehtoisia käyttää CONNECT-viestipaketissa. Asiakas voi lähettää käyttäjätunnuksen ja salasanan todennusta ja valtuutusta varten käyttämällä näitä määritteitä. On hyvä kuitenkin huomata, ettei tämä viesti ole oletuksena salattu, joten CONNECT-viestin yhteydessä olisi

hyvä käyttää esimerkiksi TLS-salausta. Oletuksena viestin salasana näkyy verkkoa vakoilevalle vain pelkkänä tekstinä.

- LastWill eli viimeinen tahto. Tämä on kuten PUBLISH-viestipaketti, mutta lähetetään CONNECT-viestipaketin yhteydessä. Välittäjä saadessaan CONNECT-viestin yhteydessä viimeisen tahdon viestin tallettaa välittäjä tämän viestin, kunnes asiakkaan yhteys katkea. Tällöin välittäjä julkaisee tämän LastWill-viestin kaikille muille asiakkaille.

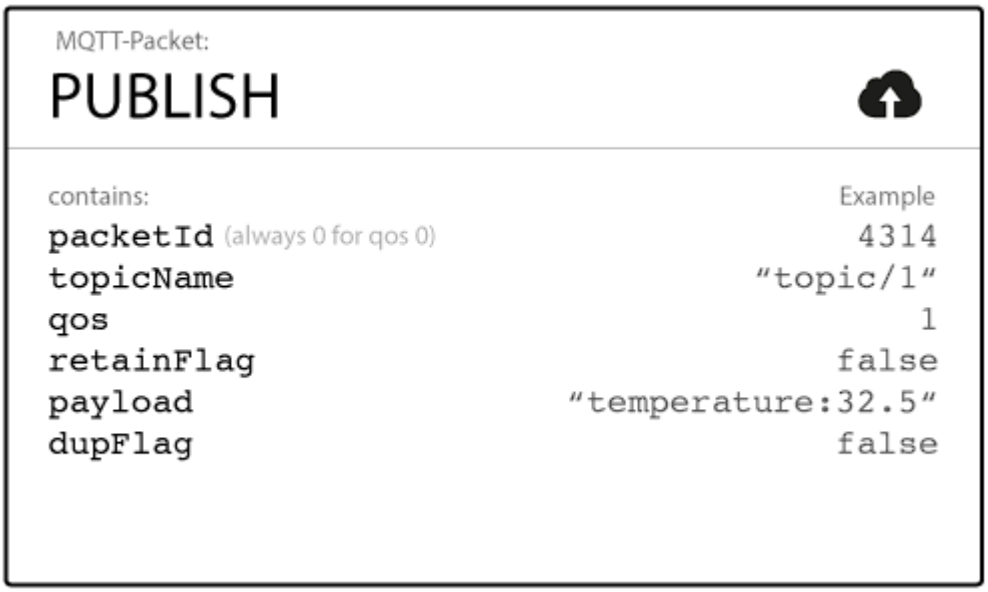
- KeepAlive eli pidä elossa on MQTT-protokollaan rakennettu ominaisuus, jolla on ratkaistu TCP/IP-protokollan puoliavoimen kommunikaation ongelma. CONNECT-viestin yhtyessä määritetään pidä elossa ominaisuuden aikaväli, vakioarvo 60 sekuntia. Tämä aikaväli kertoo, mikä on pisin aika ilman asiakkaan ja välittäjän välistä tiedostamista. Tätä avataan lisää luvussa 2.4.1.

Välittäjän saatua CONNECT-viestipaketin asiakkaalta, lähettää välittäjä CONNACK-viestipaketin asiakkaalle. Tämä viestipaketti koostuu kahdesta määritteestä: viestin vastauskoodista ja istunnon tilalipusta. CONNACK-viestin vastauskoodi käytännössä kertoo asiakkaalle, onko yhteys onnistunut vai ei. Istunnon tila lipussa välittäjä kertoo asiakkaalle, onko välittäjällä jo pysyvä istunto käytettävissä aiemmasta istunnosta asiakkaan kanssa. Pysyvän istunnon tilaa käydään lävitse luvussa 2.4.3. [9; 18.]

2.3.3 PUBLISH – Julkaisu

MQTT-asiakas voi julkaista viestejä heti, kun se saa yhteyden välittäjään. Jokaisen viestin tulee sisältää aihe, jota välittäjä voi käyttää välittääkseen viestin kiinnostuneille tilaajaasiakkaille. Tyypillisesti jokaisessa viestissä on hyötykuorma, joka sisältää lähetettävän datan tavumuodossa. MQTT-protokollaa käyttävä määrittää, kuinka julkaistujen viestien hyötykuorma on rakennettu. Eli viestejä lähettävä asiakas päättää, haluaako se lähettää binaaridataa, tekstidataa vai jopa täysimittaista XML- tai JSON-muotoa.

PUBLISH-viestillä MQTT:ssä on useita kuvan 6 mukaisia määritteitä. [18; 19; 20.]



Kuva 6. MQTT-julkaisupaketin rakenne [19].

Alle on listattu kuvan 6 PUBLISH-viestin rakenteen määritteet [18; 19; 20.]:

- pakcetId: Paketin tunniste yksilöi viestin, kun se kulkee asiakkaan ja välittäjän välillä. Paketin tunniste on merkityksellinen vain nollaa suuremmille QoS-tasoille. Asiakaskirjasto ja/tai välittäjä on vastuussa tämän sisäisen MQTT-tunnisteen asettamisesta.
- topicName: Aiheen nimi on yksinkertainen merkkijono, joka on rakenteeltaan hierarkkisesti rakennettu vinoviivalla erottimina.
- qos: Viestin palvelutaso. Tasoja on kolme: 0, 1 ja 2. Palvelutaso määrittää millainen takuu viestillä on aiotun vastaanottajan saavuttamiseksi. Tästä lisää luvussa 2.4.2.
- retainFlag: Tämä lippu määrittää, tallentaako välittäjä viestin viimeisenä tunnettuna hyvänä arvona tietylle aiheelle. Kun uusi asiakas tilaa aiheen, hän vastaanottaa viimeisen kyseisestä aiheesta säilytetyn viestin. Tästä lisää luvussa 2.4.4.

- payload: Tämä on viestin todellinen sisältö. Hyötykuormaan mahdollista lisätä, vaikka kuvia ja tai tekstiä millä tahansa koodauksella sekä salattua tietoa ja käytännössä kaikkia tietoja binäärimuodossa.
- dupFlag: Lippu osoittaa, että viesti on kaksoiskappale ja lähetettiin uudelleen, koska vastaanottaja, asiakas tai välittäjä, ei kuitannut alkuperäistä viestiä. Tämä koskee vain QoS-arvoa, joka on suurempi kuin 0. Yleensä MQTT-asiakaskirjasto tai välittäjä käsittelee uudelleenlähetyksen ja kopiointimekanismin toteutustietona.

Kun asiakas lähettää viestin MQTT-välittäjälle julkaistavaksi, välittäjä lukee viestin, kuittaa viestin QoS-tason mukaan ja käsittelee viestin tämän tasoa vastaavalla tavalla. Julkaisevaa asiakasta vain kiinnostaa saada julkaistua viesti välittäjälle ja, kun tämä viesti saapuu välittäjälle, vastuu viestistä siirtyy myös. [18; 19; 20.]

2.3.4 SUBSCRIBE & SUBACK – Tilaus ja tilausvahvistus

Vastaanottaakseen viestejä kiinnostavista aiheista asiakas lähettää MQTT-välittäjälle SUBSCRIBE-viestin. Tämä tilausviesti on hyvin yksinkertainen, se sisältää yksilöllisen tunnisteen ja aiherakenteen. Jokaisen tilauksen vahvistamiseksi välittäjä lähettää asiakkaalle SUBACK-kuittausviestin. Tämä viesti sisältää alkuperäisen SUBSCRIBE-viestin pakettitunnisteen ja luettelon palautuskoodeista. [19; 20; 21.]

2.3.5 UNSUBSCRIBE & UNSUBACK – Tilauksen peruutus ja tämän vahvistus

SUBSCRIBE-viestipaketin vastakohta on UNSUBSCRIBE-viestipaketti. Tällä viestipaketilla poistetaan olemassa olevan asiakkaan tilaus välittäjältä. UNSUBSCRIBE-viestinpaketin rakenne on samanlainen kuin SUBSCRIBE-viestipaketissa. Siinä on paketin tunniste ja aihe rakenne. Vahvistaakseen tilauksen lopettamisen välittäjä lähettää asiakkaalle UNSUBACK-kuittausviestin. Tämä viesti sisältää vain alkuperäisen UNSUBSCRIBE-viestin pakettitunnisteen. [18; 19; 20.]

2.4 MQTT-protokollan ominaisuuksia

2.4.1 Pidä elossa

Pidäelossa-ominaisuus koostuu kahdesta eri viestipaketista, joita ovat PINGREQ ja PINGRESP. Molemmat ovat hyötykuormattomia paketteja. PINGREQ on asiakkaan lähettämä paketti, jolla asiakas ilmaisee välittäjälle olevansa elossa. Välittäjän saadessa tämän PINGREQ-paketin on välittäjän vastattava PINGRESP-paketilla asiakkaalle osoittaakseen olevansa saatavilla. Tässä on lueteltuna pidäelossa-ominaisuuden muita tärkeitä kohtia [10.]:

- Asiakas voi lähettää PINGREQ-paketin milloin tahansa, kun se haluaa varmistaa, että verkkoyhteys on edelleen toiminnassa.

- Pidäelossa-pakettien maksimi lähetys intervalli on 18 h 12 m 15 s.

- Jos tämä intervalli asetetaan 0, ominaisuus jätetään pois käytöstä.

- Välittäjän on katkaistava sen asiakkaan yhteys, joka ei lähetä viestiä tai PINGREQ-pakettia puolitoista kertaa niin suuressa ajassa, kuin pidäelossa-pakettien lähetys intervalliksi on määritetty. Sama pätee, jos asiakas ei saa välittäjältä PINRESP-pakettia.

- Jos välittäjä ei saa PINGREQ-pakettia tai muuta pakettia asiakkaalta, välittäjä katkaisee yhteyden asiakkaaseen ja lähettää viimeisen tahdon ja testamenttiviestin, jos tämä on määritetty.

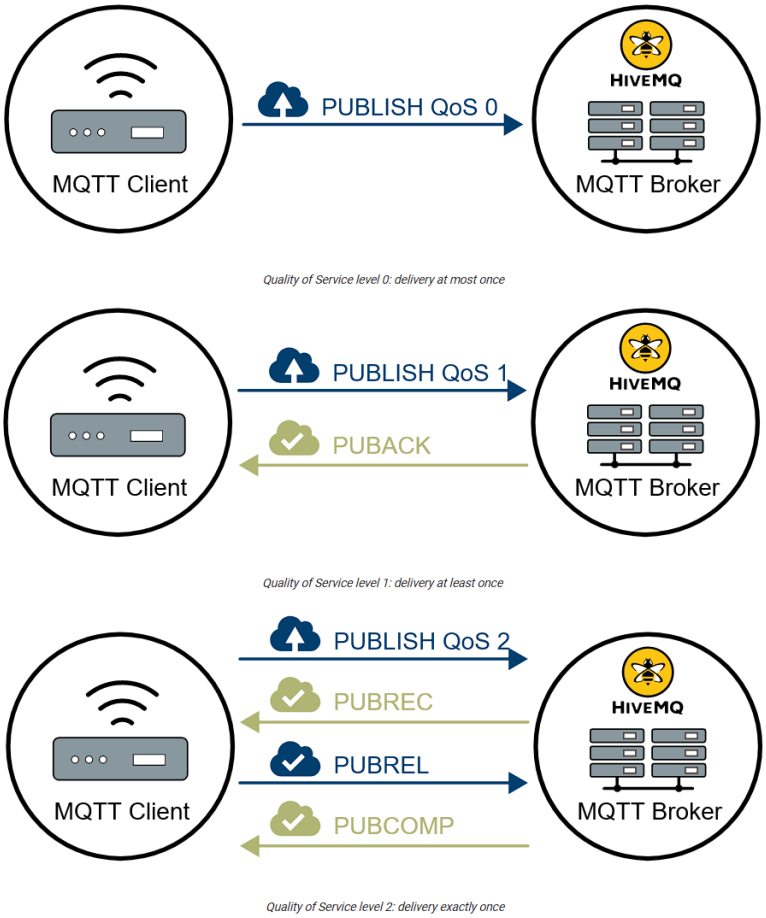
Yleensä yhteyden menettänyt asiakas yrittää muodostaa yhteyden uudelleen välittäjän kanssa, mutta joskus välittäjällä on edelleen puoliavoimessa yhteydessä asiakkaan kanssa. Tällaisessa tilanteessa tulee suorittaa asiakkaan haltuunoton. Välittäjä sulkee edellisen yhteyden samaan asiakkaaseen käyttäen asiakastunnistetta ja muodostaa uuden yhteyden asiakkaaseen. Tämä toiminta

varmistaa, että puoliavoin yhteys ei estä asiakasta muodostamasta yhteyttä uudelleen. [10.]

2.4.2 Viestinnän laatu - QoS

MQTT-protokollassa määritetään kolme eri laatutasoa QoS-tasoa. Riippuen tästä valitusta laatutasosta on MQTT-protokolla viestintä kaksisuuntaista julkaisijan ja välittäjän välillä. Laatutasolla nolla välittäjä ja julkaisija lähettää viestipaketin vain kerran tietämättään, saako vastaanottaja viestiä vai ei. Laatutasoilla yksi ja kaksi viestinnän muoto muuttuu kaksisuuntaiseksi. Tasolla yksi välittäjä ja julkaisija lähettävät viestit vain kerran, mutta odottavat kuittausviestiä takaisin viestin vastaanottajalta. Tasolla kaksi viestit varmennetaan neljän osan kättely menetelmää käyttämällä. [15, 21.]

Kuvassa 7 sekä taulukossa 1 on esitetty nämä kolme viestinnän laatutasoa sekä viestintään käytetyt viestipaketit.



Kuva 7. MQTT-protokollan QoS-laatutasot sekä tasojen vaatimat viestipaketit [19].

Kuvaa 7 lukiessa ylhäältä alaspäin laatutasot ovat QoS-taso 0, QoS-taso 1 ja QoS-taso 2. QoS-taso 0 on yksinkertaisin tapa julkaista viestipaketteja. Asiakas yksinkertaisesti vain julkaisee viestin, eikä välitä, saako välittäjä sen vai ei. QoS-taso 1 menetelmä takaa, että viesti siirretään onnistuneesti välittäjälle. Välittäjä lähettää kuittauksen takaisin viestin julkaisijalle, mutta jos kuittaus katoaa, viestin julkaisija tiedä viestin menneen perille, joten tämä tämän viestin uudelleen. Viestejä julkaiseva asiakas lähettää tällä laatutasolla viestejä, niin kauan uudelleen, kunnes saa tästä lähetetystä viestistä vastaus vahvistuksen. Tämä tarkoittaa, että lähetys on taattu, vaikka viesti voi saapua välittäjälle useammin kuin kerran. QoS-taso 2 on palvelun korkein taso, jossa lähettäjän ja vastaanottajan välillä on neljän viestin sarja, eräänlainen kättely, jolla varmistetaan, että pääviesti on lähetetty ja kuittaus vastaanotettu. Kättelyn suoritettua sekä lähettäjä että vastaanottaja ovat varmoja, että viesti on lähetetty täsmälleen kerran. [19.]

2.4.3 Pysyvä istunto

MQTT-protokollassa määritetään kahdenlaisia istuntoja: pysyviä ja jatkuvia. Jatkuvassa istunnossa vastaanottaakseen viestejä MQTT-välittäjältä asiakas muodostaa yhteyden välittäjään. Jatkuvassa istunnossa saattaa ilmetä tilanne, jossa asiakkaan ja välittäjän välinen yhteys katkeaa. Tällöin kaikki aiheet, joita asiakas tilaa menetetään ja nämä täytyy tilata uudelleen välittäjältä yhteyden takaisin muodostumisen yhteydessä. Sellaisille asiakkaille, joilla on rajoitetut resurssit, voi tämä uudelleentilaaminen olla laskennallisesti liikaa. Tämän ongelman välttämiseksi asiakas voi pyytää jatkuvaa istuntoa, kun se muodostaa yhteyden välittäjään. Jatkuvat istunnot tallentavat välittäjälle kaikki asiakkaalle tärkeät tiedot. Asiakastunnus, jonka asiakas antaa yhteydenmuodostuksen yhteydessä tunnistaa istunnon tilan, onko pysyvä vai ei. [22; 23.]

Asiakkaan yhteyden muodostus viestissä on puhtaalle istunnolle lippu, englanniksi cleanSession flag. Tällä lipulla asiakas ilmaisee välittäjälle, millaista istuntoa tarvitsee. Lipun ollessa totta asiakas ei halua puhdasta istuntoa, eli yhteyden menetyksessä hävitetään kaikki jonossa olevat viestit. Lipun ollessa epätosi välittäjä luo pysyvän istunnon asiakkaalle. Kaikki tiedot ja viestit säilytetään seuraavaan kertaan, kunnes asiakas pyytää puhdasta istuntoa. Puhtaan istunnon lipun arvo ollessa epätosi ja tilanteissa, jossa välittäjällä on jo istunto käytettävissä asiakkaalle, välittäjä käyttää olemassa olevaa istuntoa ja toimittaa aiemmin jonossa olevat viestit asiakkaalle. [22; 23.]

Asiakkaan on myös hyvä tietää yhdistäytyessään välittäjään, onko odotettavissa aiemman istunnon viestejä vai ei. Tämän tiedon asiakas saa välittäjältä CONNACK-paluuviestissä. CONNACK-viestistä löytyy lippu, jolla välittäjä ilmaisee asiakkaalle vanhan istunnon tilan. [22; 23.]

Pieni kaistanleveys ja epävakaassa verkossa toimiminen luo verkkokatkoksia usein. Pysyvä istunto ominaisuus tallentaa istunnon tilan, joten tällä välttää uudelleentilaukset, kun yhteys uudelleen muodostetaan. Tämä voi vähentää asiakkaan ja välittäjän resurssien kulutusta yhteyden muodostamisen yhteydessä. Pysyvän istunnon tilassa välittäjä tallentaa kaikki viestit, jotka tulevat välittäjälle tilaajan ollessa offline-tilassa. Välittäjä välittää nämä tallennetut viestit heti, kun asiakas muodostaa yhteyden uudelleen. Eli tämän avulla voi välttää viestien katoamisen epävakaassa verkossa. [23.]

2.4.4 Pysyvät viestit

Pysyvä viesti on normaali PUBLISH-viestipaketti, jonka säilytyslippu eli retain flag on asetettu todeksi. Välittäjä tallentaa viimeksi saadun viestin ja vastaavan laatutason kyseiselle aiheelle. Tämän jälkeen, joka kerta kun asiakas tilaa tämän kyseisen aiheen, välittäjä lähettää tämän talletetun viestin tilaajalle. Välittäjä tallettaa vain yhden viestin per aihe. Säilytetyt viestit auttavat uusia tilaajia saamaan tilapäivityksen heti aiheen tilauksen jälkeen. [24; 25; 26.]

Vaikka säilytetty viesti on tallennettu välittäjälle, se ei ole osa välittäjän ja asiakkaan välistä istuntoa. Eli vaikka se istunto lopetetaan, jossa tämän viestin alkuperäinen julkaisija on, säilytettyä viestiä ei poisteta välittäjältä. On vain kaksi tapaa poistaa säilytetty viesti. Ensimmäinen tapa one, että asiakas lähettää viestin, jossa ei ole hyötykuormaa ja merkitsee sen säilytettäväksi viestiksi. Toinen tapa on, että alkuperäisen viestinyhteydessä on käytetty viestin vanhentumisen lippua ja tämä viesti yksinkertaisesti vanhenee ja näin ollen poistuu. [25; 26.]

2.5 Tietoturva

Yleisesti ottaen tietoturva on joukko käytäntöjä, jotka on suunniteltu suojaamaan tiedot luvattomalta käytöltä tai muutoksilta. Tietoturvasta yleisesti puhuttaessa siihen liittyy varsin usein sekä tietojen tallentaminen että siirtäminen koneelta tai paikasta toiseen. Joitakin tietotyyppejä ei ole tarkoitettu yleisölle, ja niitä tulee suojata tietoturvallisuuden peruspilareilla: luottamuksellisuus, eheys ja saatavuus. Kun todelliset koneet tai esineet vaarantuvat, hyökkääjä voi vahingoittaa oikeita ihmisiä. Vaikka tietomurron kohteena ei olisi mukana oikeita ihmisiä arkaluonteisten tietojen paljastaminen voi vahingoittaa vakavasti yrityksen mainetta. [27; 28.]

Itse MQTT-viestiprotokollassa on vain muutama suoja mekanismi. MQTT:n rakentuminen TCP/IP-protokollan päälle antaa mahdollisuuden suojata siirrettävää dataa verkon jokaisessa kerroksessa. Seuraavaksi käydään lävitse näiden kerrosten tarjoamia suojauksia. [27; 29.]

2.5.1 Verkkokerroksen suojaus

Yleisesti hyvänä käytäntönä on määrittää palomuuri sallimaan tai estämään tiettyjä IP-alueita MQTT-verkosta. Myös muita tekniikoita, kuten Ipsec:iä tai VPN:ää, voidaan soveltaa sen varmistamiseksi, että vain luotettavat asiakkaat pääsevät verkkoon. [27; 29.]

Hyvänä tietoturvallisuuden periaatteena jokaisen yhteyden MQTT-välittäjään tulisi läpäistä vähintään yksi palomuuri, jos viestintä tapahtuu julkisessa verkossa. Verkonsuojaamisen kohdalla ei ole kehittynyt kaikkien hyväksymää toteutustapaa. Tämä tarkoittaa sitä, että palomuurien säännöt on räätälöitävä käyttökohteen mukaisesti. Hyvänä nyrkkisääntönä on rakentaa palomuurin asetukset niin, että tämä päästäisi vain odotetun ja halututun viestiliikenteen palomuurin takana oleville laitteille. [30.]

2.5.2 Kuljetuskerroksen suojaus

TLS/SSL-tekniikkaa käytetään yleisesti kuljetuskerroksen suojaamiseen. Tämä menetelmä on turvallinen ja todistettu tapa varmistaa, ettei tietoja voida lukea lähetyksen aikana sekä tämä tarjoaa asiakasvarmenteiden todennuksen molempien osapuolten tarkastamiseksi. [27; 29.]

TLS/SSL-tekniikka tarjoaa suojatun viestintäkanavan asiakkaan ja palvelimen välillä hyödyntäen kättelymekanismia. Tämän kättelyn aikana asiakas ja palvelin pystyvät neuvottelemaan eri parametreista suojatun yhteyden luomiseksi. Kättely valmistuessa, asiakkaan ja palvelimen välille on muodostunut salattu tapa viestiä keskenään, eikä viestintää salakuunteleva pysty avaamaan viestipakettien sisältöä. Tällaista tilannetta kutsutaan Man-In-The-Middle-Attack- eli henkilö välissä-hyökkäys. Suojaamattomassa viestinnässä tämä salakuuntelija voi lukea ja muokata verkossa liikkuvaa informaatiota. TSL/SSL-tekniikkaa käyttämällä voidaan olla varmoja siitä, ettei kukaan pääse viestintään väliin. [28; 31.]

2.5.3 Sovelluskerroksen suojaus

MQTT-protokollassa esitetty asiakastunnus, käyttäjätunnus ja salasana mahdollistavat sovellustason todennuksen ja valtuutuksen toteuttamisen. Itse protokolla tarjoaa nämä ominaisuudet. Valtuutus tai valvonta sille, mitä kukin laite saa tehdä, määräytyy tietyn välittäjän toteutuksen mukaan. Lisäksi on mahdollista käyttää hyötykuorman salausta sovellustasolla siirrettyjen tietojen turvaamiseen. [27; 29.]

3 Pilvipalvelu

Pilvellä viitataan tietotekniikassa palvelimien verkostosta koostuvaan palveluun, johon on pääsy paikasta riippumatta, kunhan laite, millä yhdistetään, kykenee kirjautumaan julkiseen verkkoon. Tavalliselle käyttäjälle pilvipalvelu toimii yleensä kuten verkkolevy eli tiedostojen tallennuspaikka, johon käyttäjällä on helppo pääsy monilla eri laitteilla. [32.]

Verkkolevyjen korvauksen lisäksi monet pilvipalveluiden palveluita tarjoavat datan säilytyksen lisäksi tämän tallennetun datan analysointia ja laskentapalveluita. Pilvessä tapahtuvaa laskentaa kutsutaan pilvilaskennaksi, englanniksi cloud computing. [33.]

3.1 Pilvipalveluiden tyypit

Pilvipalveluita on neljää päätyyppiä. Nämä ovat yksityiset pilvet, julkiset pilvet, hybridipilvet ja monipilvet. Nämä myös rakentuvat kolmesta päämallista, Infrastructure-as-a-Service IaaS eli infastruktuuri palveluna, Platforms-as-a-Service PaaS eli alustapalveluna ja Software-as-a-Service SaaS eli ohjelmistopalveluna. [30; 33; 34.]

3.1.1 IaaS – Infrastruktuuri palveluna

IaaS tarkoittaa, että pilvipalveluntarjoaja hallinnoi infrastruktuuria käyttäjän puolesta, mikä tarkoittaen kaikkia palvelimia, verkkoa ja datan säilytystä. IaaS:n avulla loppukäyttäjät voivat skaalata ja supistaa resursseja tarpeen mukaan, mikä vähentää korkeiden, etukäteisten pääomakustannusten tai tarpeettoman paikallisen tai oman infrastruktuurin tarvetta. [33; 34; 35.]

3.1.2 PaaS – Alusta palveluna

PaaS tarkoittaa, että laitteiston ja sovellusohjelmistoalustan tarjoaa ja hallinnoi ulkopuolinen pilvipalveluntarjoaja, mutta käyttäjä hoitaa alustan päällä toimivat sovellukset ja tiedot, joihin sovellus luottaa. PaaS tarjoaa käyttäjille ensisijaisesti

kehittäjille ja ohjelmoijille jaetun pilvialustan sovellusten kehittämiseen ja hallintaan ilman, että heidän tarvitsee rakentaa ja ylläpitää prosessiin tavallisesti liittyvää infrastruktuuria. [33; 34; 35.]

3.1.3 SaaS – Ohjelmisto palveluna

SaaS on palvelu, joka toimittaa ohjelmistosovelluksen. Tyypillisesti SaaS-sovellukset ovat verkkosovelluksia tai mobiilisovelluksia, joita käyttäjät voivat käyttää verkkoselaimen kautta. Ohjelmistopäivitykset, virheenkorjaukset ja muu yleinen ohjelmistojen ylläpito hoidetaan käyttäjän puolesta. SaaS:ia käyttäessä asiakkaat muodostavat yhteyden pilvisovellukseen dasboardin eli kojelaudan kautta tai API:n eli ohjelmistorajapinnan kautta. Tämä tarkoittaa sitä, että asiakkaat voivat yhdistää käytännössä minkä tahansa laitteen kanssa tähän pilvisovellukseen. Eli käytännössä laite, jolla voi pyörittää www-selain ohjelmistoa, voidaan yhdistää tähän pilvimalliin. SaaS eliminoi myös tarpeen asentaa sovellus paikallisesti jokaisen yksittäisen käyttäjän tietokoneelle, mikä mahdollistaa laajemmat ryhmä- tai ryhmäpääsymenetelmät ohjelmistoon. [33; 34; 35.]

3.1.4 Julkinen pilvi

Julkiset pilvet ovat kolmannen osapuolen pilvipalveluntarjoajien omistamia ja ylläpitämiä, jotka toimittavat laskentaresurssejaan kuten palvelimia ja tallennustilaa internetin kautta. Microsoft Azure, Amazon Web Services, Google Cloud, IBM Cloud ovat esimerkiksi julkisen pilven palveluntarjoajia. Julkisessa pilvessä kaikki laitteistot, ohjelmistot ja muut tukevat infrastruktuurit ovat pilvipalveluntarjoajan omistuksessa ja hallinnassa. [33; 34; 35.]

3.1.5 Yksityinen pilvi

Yksityinen pilvi on pilviympäristö, jossa kaikki pilvi-infrastruktuuri ja laskentaresurssit on omistettu vain yhdelle asiakkaalle. Yksityinen pilvi yhdistää monet pilvipalvelun edut kuten, skaalautuvuuden ja palvelun toimittamisen helppouden sekä tietoturvallisuuden. Yksityinen pilvi isännöidään tyypillisesti

paikan päällä asiakkaan omassa datakeskuksessa, mutta yksityinen pilvi voidaan myös isännöidä riippumattoman pilvipalveluntarjoajan infrastruktuurissa, joka sijaitsee ulkopuolisessa palvelinkeskuksessa. [33; 34; 35.]

3.1.6 Hybridipilvi

Hybridipilvet yhdistävät julkiset sekä yksityiset pilvet, mikä mahdollistaa tietojen ja sovellusten jakamisen niiden välillä. Erityisesti ja ihannetapauksessa hybridipilvi yhdistää organisaation yksityiset pilvipalvelut ja julkiset pilvet yhdeksi joustavaksi infrastruktuuriksi organisaation sovellusten ja työkuormien suorittamista varten. Tämä antaa organisaatiolle joustavuuden valita optimaalisen pilven jokaiselle sovellukselle tai työmäärälle ja siirtää työkuormia vapaasti kahden pilven välillä olosuhteiden muuttuessa. Näin organisaatio voi saavuttaa tekniset ja liiketoiminnalliset tavoitteensa tehokkaammin ja kustannustehokkaammin kuin pelkällä julkisella tai yksityisellä pilvellä. [33; 34; 35.]

3.1.7 Monipivlet

Monipilvet ovat pilviratkaisumalli, jotka koostuvat useammasta kuin yhdestä pilvipalvelusta. Kaikki hybridipilvet ovat monipilviä, mutta kaikki monipilvet eivät ole hybridipilviä. Monipilvistä tulee hybridipilviä, kun useita pilviä yhdistetään keskenään. Monipilvi ympäristö voi olla olemassa tarkoituksella, arkaluonteisten tietojen hallinnan parantamiseksi tai redundanttina tallennustilana, tai vahingossa, yleensä varjo-IT:n seurauksena. Joka tapauksessa useiden pilvien käyttö on yleistymässä yrityksissä, jotka pyrkivät parantamaan turvallisuutta ja suorituskykyä laajennetun ympäristövalikoiman avulla. [33; 34; 35.]

4 Käytetyt laitteet ja palvelut

4.1 Beckhoff

Beckhoff Automation GmbH & Co. KG on saksalainen ja maailmanlaajuisesti toimivia automaatioalan yritys. Yritys tuottaa PC-pohjaisia automaatiojärjestelmiä, komponenteista ohjelmistoon asti. Yrityksen päätuotevalikkoon kuuluvat muun muassa teolliset-PC:et, I/O- ja kenttäväyläkomponentit. [36.]

4.2 TwinCat 3 XAE

TwinCat 3 XAE eli Twin Control Automation Technology 3 eXtended Automation Engineering on Beckhoffin ohjelmistokehitysalusta. Kehitysympäristö on rakennettu Microsoft Visual Studion päälle noudattaen IEC 61131-3 -standardia ohjelmoitaville logiikoille. Visual Studioon integroituminen mahdollistaa rajapinnat muihinkin ohjelmointikieliin, kuten C:hen, C++:aan tai C#:iin. [37.]

TwinCat 3 XAE:ssa Beckhoff jakaa ohjelmointiresursseja niin kutsuttuihin komponenttiryhmiin. Näitä ryhmiä käytetään pitkälti kehittäjäympäristössä, eikä itse ohjelmistoa suorittavalle laitteelle tarvitse niitä asentaa joitakin poikkeuksia lukuun ottamatta. Tälle insinöörityölle tärkeä ryhmä on TF6xxx, josta löytyvät Beckhoffin valmiit kirjastot monelle erilaiselle teolliselle kommunikointistandardille, kuten Modbus TCP:lle tai tässä työssä paljon käytetylle IoT Communicatio (MQTT). [37; 38.]

4.3 Siemens

Siemens AG on saksalainen ja globaalisti toimiva teknologia-alan yritys. Yrityksen toimialaa ovat muun muassa tieto- ja tietoliikennetekniikka, automaatiotekniikka, terveydenhuolto ja energiatuotanto. [39.]

Teollisuuden tarpeiden kehittymisen myötä dataa on enemmän kuin koskaan aikaisemmin saatavilla. Tämä data pitäisi myös saada siirrettyä dataa luovasta

laitteesta tai järjestelmästä, datan jatkokäsittelyä varten. Datan raakana lähteestä lähettäminen esimerkiksi pilvipalveluun saattaa vaatia suuria määriä kaistaa verkosta, joka voi tulla nopeasti kalliiksi. Yhtenä ratkaisuna on ottaa reunalaskentatekniikkaa avuksi, englanniksi edge computing. Reunalaskenta mahdollistaa datan ennakkokäsittelyn, jonka seurauksena ovat pienemmät siirto- ja säilytyskustannukset. [40.]

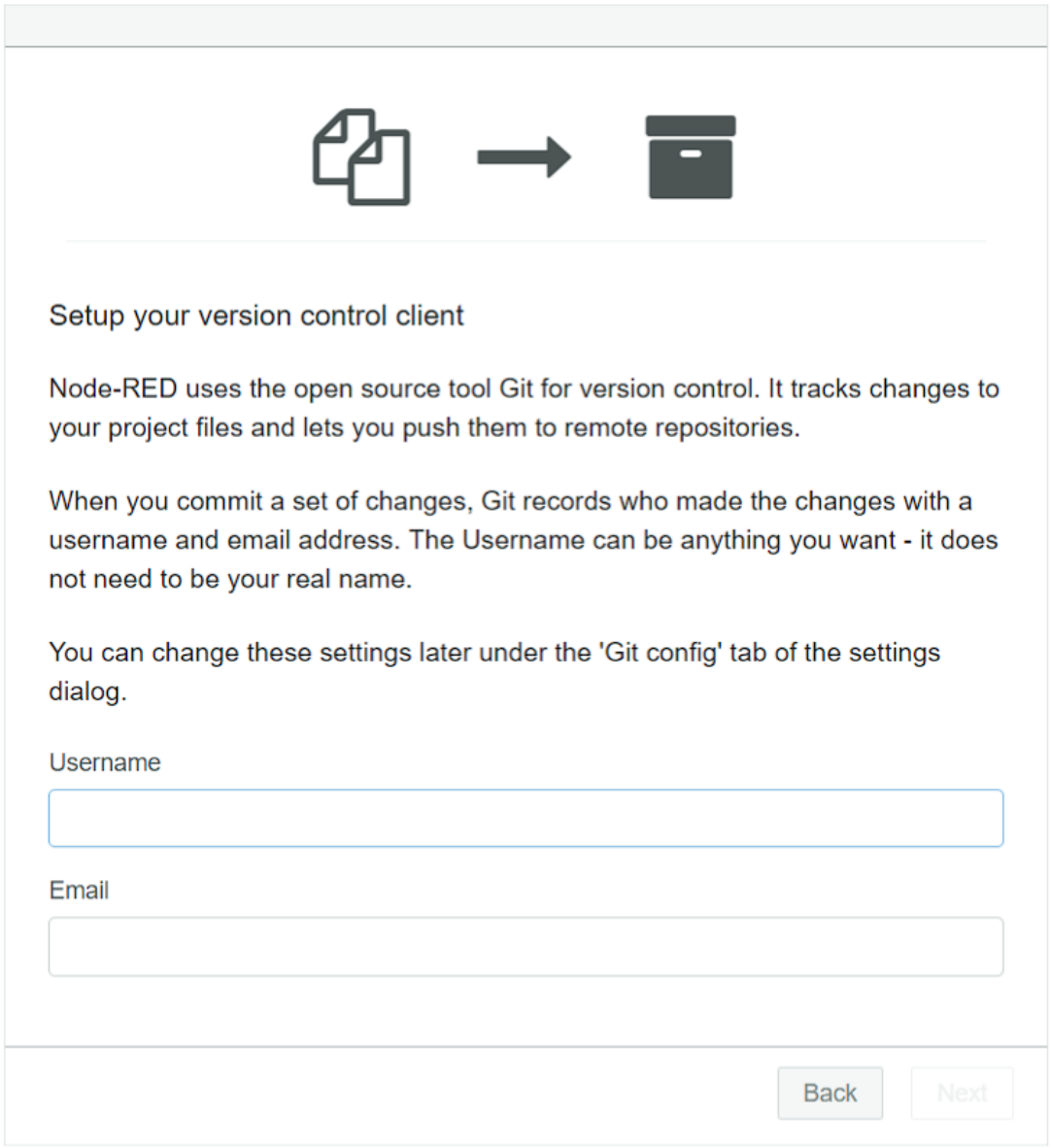
Tässä työssä on käytetty Siemensin Industrial Edgeä eli teollista reunatekniikkaa, joka toteuttaa edellä mainitun reunalaskennan sekä mahdollistaa datan jatkokäsittelyä varten siirtämisen.

4.3.1 Industrial Edge

Industrial Edge edustaa avointa, käyttövalmista reunalaskenta-alustaa, joka koostuu reunalaitteista, reunasovelluksista ja reunaliitettävyydestä. Se helpottaa tietojen keräämistä ja analysointia teollisista resursseista. Siemensin Industrial Edgestä löytyy IoT:lle tärkeitä ominaisuuksia, kuten Node Red, joka on Siemensin tapauksessa nimetty Flow Creatoriksi ja MQTT-välittäjä. [41.]

4.3.2 Flow Creator

Siemensin Industrial Edge Flow Creator on Siemensin selainpohjainen visuaalinen ohjelmointityökalu, jonka avulla voidaan luoda reunalaitteella toimivia ohjelmia. Vaikka Siemens ei missään suoraan mainitse, että kyseinen Flow Creator olisi IBM:n kehittämä Node-Red-työkalu, on tästä paljon viitteitä. Tästä esimerkkinä Siemensin ohjekirjassa, jossa kerrotaan Flow Creatorin toiminnasta. Flow Creatorin ohjeesta otetussa kuvankaappauksessa kuvassa 8 on selkeä maininta, että kyseessä on Node-Red. Ohje on luettavissa myös osoitteesta:
https://cache.industry.siemens.com/dl/files/331/109794331/att_1057284/v1/IE_flow_creator_operation_en-US.pdf [42.]



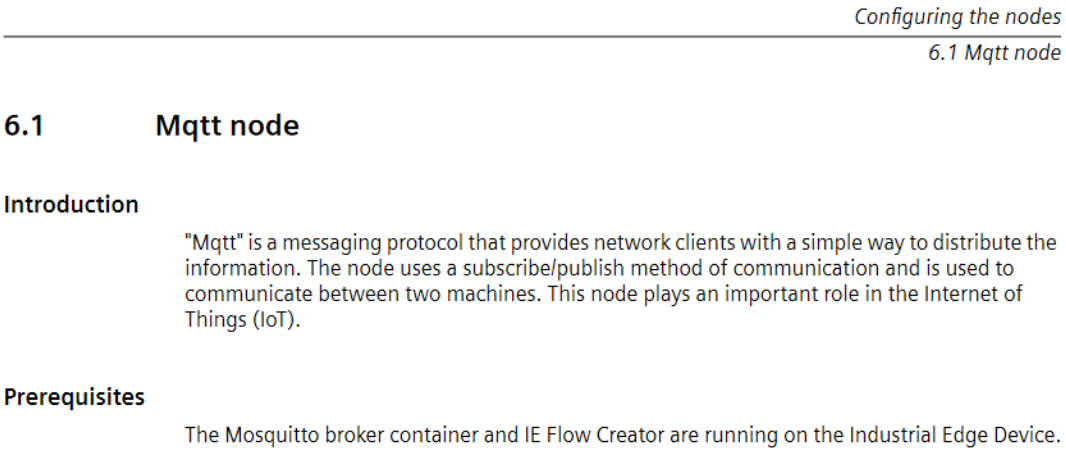
Kuva 8. Flow Creator-ohje Git-versiohallintaan kirjautumisesta [42].

Kuten kuvasta 8 voidaan epäillä, Flow Creatorin pohjana on toiminut alun perin IBM:n kehittämä Node-Red-työkalu. Node-Redillä luodut sovellukset muodostuvat nodeista eli solmuista, jotka vastaanottavat ja käsittelevät dataa, ja välittävät sen lopulta seuraavalle solmulle. Solmujen verkostosta muodostuu flow, eli virtaus. Node-Red sisältää laajat kirjastot, jossa on paljon valmiita

solmuja heti käytettäväksi. Solmuja voidaan myös asentaa lisää tai luoda itse käyttämällä JavaScripti-ohjelmointi kieltä. [43.]

4.3.3 MQTT Connector

Siemensin Industrial Edge -sovelluksista löytyy myös MQTT-välittäjä, jota Siemens kutsuu MQTT Connector eli MQTT yhdistäjäksi. Siemens ei kerro suoraan, mitä tekniikkaa MQTT välittäjään on käytetty, mutta tästä on heidän ohjeissaan viitteitä siitä, että kyseessä olisi Mosquitto-niminen MQTT-välittäjä. Kuvassa 8 on samasta Siemensin ohjeesta otettu kuvan kaappaus, jossa Siemens ohi mennen mainitsee MQTT Connectorin olevan Mosquitto.



Configuring the nodes
6.1 Mqtt node

6.1 Mqtt node

Introduction

"Mqtt" is a messaging protocol that provides network clients with a simple way to distribute the information. The node uses a subscribe/publish method of communication and is used to communicate between two machines. This node plays an important role in the Internet of Things (IoT).

Prerequisites

The Mosquitto broker container and IE Flow Creator are running on the Industrial Edge Device.

Kuva 9. Siemensin MQTT Connector saattaa olla Mosquitto-välittäjä [42].

Jos kuva 9 ei riitä, Siemensin ohjeessa, jossa käsitellään Industrial Edgen dataväyliä, on sivulla 33 mainittu vikatila, joka ei ole Mosquitto-välittäjän tukema. Ohje on saatavilla kyseisen linkin alla: https://cache.industry.siemens.com/dl/files/600/109795600/att_1062366/v1/ie_databus_enUS_en-US.pdf

Eclipse Mosquitto on avoimen lähdekoodin ohjelmisto, joka toteuttaa MQTT-viestintäprotokollan välittäjän tehtävää. Mosquitto on kevyt ja sopii käytettäväksi kaikilla laitteilla pienitehoisista yksilevytietokoneista täysiin palvelimiin. [44.]

4.4 Tietoturva

Tämän opinnäytetyön järjestelmät ovat pitkälti omassa verkossa, joka ei ole kytkettynä julkiseen verkkoon, lukuun ottamatta yksittäisiä kytköksiä. Näitä julkisen verkon liitoskohtia suojataan palomuurin avulla.

Solmukohtia suojaava palomuuri on ISO 27001 -sertifioitu, mikä on yksi kulmakivi turvallisen verkon kehityksessä ja luomisessa. Palomuuriin kytkeytyvät laitteet tunnistavat toisensa kryptografisella pariliitoksella, jotka ovat tallennettuina päätepisteissä. VPN-etäyhteyden muodostettua käyttäjän ja palomuurin välille. Kaikki dataliikenne salataan ja salaus puretaan vain yhteyden päätepisteissä eli kyseessä on end-to-end encryption eli päästä-päähän salaus. Päästä päähän-salauksen lisäksi palomuurissa käytetään todistetusti toimivia tekniikoita, kuten RSA Cryptosystemiä, AES 256-salausta, Diffie-Hellman avainten vaihtoa sekä TLS-istuntoa. [45; 46; 47.]

5 TwinCat 3 -sovellus

Opinnäytetyön ohjelmakoodi perustuu Beckhoffin tarjoamaan kirjaston käyttämiseen. Beckhoffin kirjastossa on kaikki, mitä tarvitaan MQTT-viestinnän luomiseksi. Viestien hyötykuorman rakentaminen jää tietenkin itse kehittäjän vastuulle.

5.1 Kirjastot

Asiakasohjelman tuotosta varten käytettiin pitkälti TF6701-Tc3_IotBase-kirjastoa. Hyötykuroman rakenteen ollessa JSON-string-muotoista käytettiin lisäksi TF6701-Tc3_JsonXml-kirjastoa. Tc3_IotBase-kirjaston avulla voidaan luoda MQTT-viestit ja muu kommunikaatio, Tc3_JsonXml-kirjasto taas mahdollistaa JSON-stringin luomisen. Kuvassa 10 on esitetty Tc3_IotBasen kirjastosta löytyvää funktio blokkia eli funktiolohkoa, jolla voidaan luoda ja

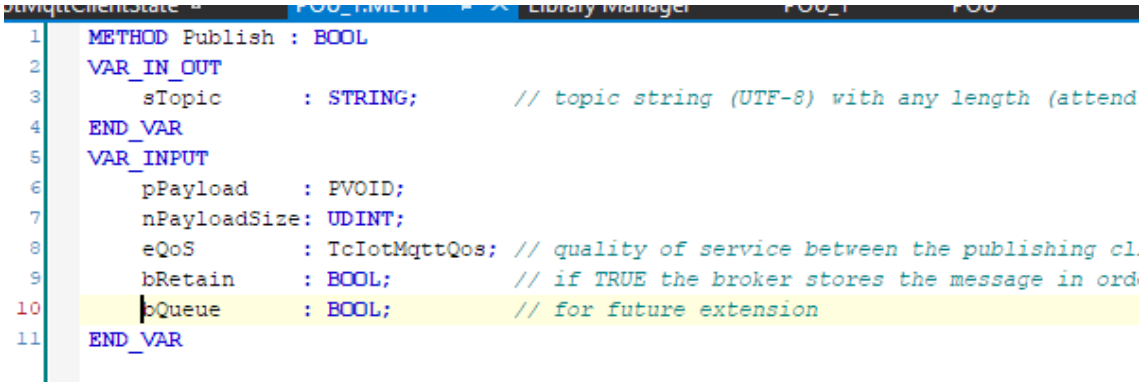
ylläpitää MQTT-yhteys välittäjään.

```
1  FUNCTION_BLOCK FB_IotMqttClient
2  VAR_INPUT
3  sClientId       : STRING(255);                  // default is generated dur
4  sHostName       : STRING(255) := '127.0.0.1';   // default is local host
5  nHostPort       : UINT := 1883;                 // default is 1883
6  sTopicPrefix    : STRING(255);                  // topic prefix for pub and
7  nKeepAlive      : UINT := 60;                   // in seconds
8  sUserName       : STRING(255);                  // optional parameter
9  sUserPassword   : STRING(255);                  // optional parameter
10 stWill          : ST_IotMqttWill;               // optional parameter
11 stTLS           : ST_IotMqttTls;                // optional parameter
12 END_VAR
13 VAR_OUTPUT
14 bError          : BOOL;
15 hrErrorCode     : HRESULT;
16 eConnectionState: ETcIotMqttClientState;
17 bConnected      : BOOL;                         // TRUE if connection to ho
18 END_VAR
```

Kuva 10. Funktio lohko FB_IotMqttClient.

Kuvassa 10 on esitetty valmista FB_IotMqttClient-funktiolohkoa, jolla voidaan luoda MQTT-viestiyhteys. Tälle funktiolohkolle annetaan kaikki tarvittavat parametrit MQTT-viestinnän luomiseksi. Nämä on käyty tämän työn luvussa 2 lävitse. Tätä funktiolohkoa ohjelmassa kutsuttaessa yrittää tämä luoda yhteyden annettuja parametrien perusteella. Yhteyden onnistuessa bConnected-lippu muuttuu todeksi. Tämän funktiolohkon muita tärkeitä ominaisuuksia on kommunikaatioon liittyvien ongelmien kertominen. Tämä tapahtuu kahdella tavalla, bError-lipun muuttuessa todeksi on ongelmatilanne päällä, sekä ongelman rajaukseen tulostaa tämä funktiolohko HRESLUT-muuttujaan MQTT-viestintään liittyviä vikakoodeja. [48.]

Kuvassa 11 on esitetty julkaisuun liittyvä metodi, jolla jatketaan FB_IotMqttClientin toimintaa [48].



```
1  METHOD Publish : BOOL
2  VAR_IN_OUT
3      sTopic       : STRING;       // topic string (UTF-8) with any length (attend
4  END_VAR
5  VAR_INPUT
6      pPayload     : PVOID;
7      nPayloadSize: UDINT;
8      eQoS         : TcIotMqttQos; // quality of service between the publishing cl
9      bRetain      : BOOL;         // if TRUE the broker stores the message in ord
10     bQueue       : BOOL;         // for future extension
11 END_VAR
```

Kuva 11. Metodi, jonka avulla tehdään MQTT-viesti julkaisuja.

Kuvassa 11 on esitetty metodi, joka jatkaa FB_IotMqttClientin toimintaa. Tätä metodia käyttämällä, voidaan tehdä MQTT-viestintäprotokollan mukaisia julkaisuja, eli viestin lähettämisiä välittäjälle. Julkaisuvaiheessa lisätään viestille aihe, jonka avulla MQTT-välittäjä käsittelee saapuneet viestit. Tässä vaiheessa voidaan myös viesteille antaa QoS-laatuluokitus, sekä aktivoida säilyvän viestin lippu. Hyötykuorma lisätään tässä vaiheessa antamalla pPayload-muuttujalle viestin sisällöksi tehty JSON-rakenne, kuten kaavassa 1 näytetään. [48.]

$$pPayload := ADR(sJsonDoc2)$$

Kaavassa 1 käytetty ADR komento on osoiteoperaatio. Eli tässä osoitetaan pPayloading-muuttujan arvoksi sama kuin sJsonDoc2 osoittamalla sJsonDoc2:seen. ADR-toiminta on samanlainen kuin käyttäisi pointereita eli osoittimia. [49.]

Näiden kahden funktio lohkojen avulla tuli rakennettua tilakone, joka käynnistyessään luo yhteyden välittäjään annettujen parametrien avulla, tehden PLC:stä asiakkaan. Tilakoneen ominaisuuksiin kuului myös yhteyden katkeamisesta johtuva vikatila. Tässä vikatilassa MQTT-tilakone ilmaisee PLC:hen yhteydessä olevalle HMI:lle vikailmoituksen, ettei saa yhteyttä.

Tilakone yrittää muodostaa yhteyden uudestaan menemällä yhteyden luontitilaan. Yhteyden luonnin epäonnistuessa useasti MQTT-tilakone yrittää hake uudelleen parametrit ja aloittaa alusta aivan alusta yhteyden luonnin.

Tilanteessa, jossa yhteys reuna laitteeseen onnistuu MQTT-tilakone mahdollistaa viestien lähetyksen. Viestien lähetys tapahtuu käyttämällä, eri viesteille rakennettuja viestin rakennemetodeja. Nämä metodit rakentavat JSON-viestin, joka sitten lähetetään reunalaitteessa olevalle välittäjälle. Reunalaitteen väittäjä vastaa viestien alkukäsittelystä hylkäämällä rakenteeltaan virheelliset viestit sekä edelleen lähetyksestä pilvipalveluun. Reunalaite kykenee tallentamaan PLC:ltä tulevia viestejä, jos yhteys pilvipalveluun katkeaa. Näin ollen PLC:n muisti ei täyty lähetystä odottavista viesteistä, eikä suoritusaikaa mene hukkaan yhteyden uudelleenluomisen aikana. PLC voi siis vain jatkaa toimintaansa, vaikka yhteyttä ei olisi. Tällaisen verkon rakenteen ansiosta dataa ei menetetä ollenkaan.

5.2 Ohjelman testaus

Ohjelmakoodin testaus tapahtui projektin aikana useassa eri vaiheessa. Ensimmäisenä MQTT-viestintäprotokollaan tutustumisen jälkeen oli luotava ohjelma, joka kykenee ottamaan yhteyttä välittäjään. Välittäjänä testeissä käytin ensin alkuun Eclipsen Mosquitto välittäjää paikallisesti omalla tietokoneella. Tämä toimiessa moitteettomasti. Siirryin käyttämään oikeaa PLC:tä, jolla yhdistin tähän omalle tietokoneelle asennettuun Mosquitto-välittäjään. Tämän kaiken onnistuessa siirryin käyttämään rajapintalaitteen välittäjää. Näin ollen minun verkkoni olikin arkkitehtuuriltaan kuten oikea järjestelmä.

Yhteyden onnistuessa rajapinta laitteen välittäjään aloin rakentamaan lopullista ohjelmakoodia. Tässä vaiheessa testaaminen muuttui viestin sisältö painiotteiseksi sekä ominaisuuspainiotteiseksi, millä tarkoitetaan sitä, että milloin ja kuinka usein viestejä lähetetään.

Viestien hyötykuorman rakenteen ollessa vastaava kuin oli määritetty, jäi enää testattavaksi, että viestit tulevat halutuista tilanteista. Viestien perille saamisen

lisäksi oli huomioitava välittäjän sekä verkon kantavuus. Tämän testaus oli yksinkertaisuudessaan stressitestaamista, eli yritystä kaataa jotain verkon komponenttia. En onnistunut koskaan saamaan Siemensin rajapintalaitteen välittäjää edes vaikuttamaan siltä, etteikö se kestäisi viestinnän tiheyttä, vaikka parhaimmillaan lähetin täysmittaisia 256 megatavun suuruisia viestejä kymmenien mikrosekuntien viiveellä. Välittäjän pyöriessä paikallisesti omalla kehitystietokoneellani sain välittäjän jumiutumaan suhteellisen helpostikin ja jopa kaadettua. En koskaan pääsyt selville kaatumisen syistä, mutta voisin epäillä, että rajapinta laite on alusta alkaen suunniteltu tällaisen kommunikaation välitykseen, kun taas tietokone on yleispätevä laite eikä ole niin optimoitu.

6 Yhteenveto

Tämän työn tarkoituksena oli modernisoida kohdeyrityksen datankeräysmenetelmä. Modernisoinnissa haluttiin datan tallettamista pilvipalveluun tämän mahdollista jatkokäsittelyä varten. Datan saamiseksi pilveen päädyttiin käyttämään MQTT-viestintäprotokollaa tämän keveyden ja helpon viestirakenteen takia.

Työssä ensin tutkittiin MQTT-viestintäprotokollan toimintaa sekä sivuutettiin pilvipalveluiden rakenteita ja toimintoja ja näiden eroja. Työn kannalta pilvipalvelun toiminnan ymmärtäminen ei ollut välttämätöntä, ainoastaan datan sinne tallettaminen, kuitenkin hyvä tietää asia, jotta data saadaan oikeassa muodossa perille asti. MQTT-tutkimuksen jälkeen esitetään käytetyt laitteet sekä ohjelmistot. Työn viimeisessä luvussa käydään PLC-ohjelman rakentumisen kannalta tärkeitä kirjastoja lävitse sekä ohjelman toiminnallisuutta yleisesti. Tarkemmin tätä ei voida salassapitovelvollisuuden nimissä käydä. Tästä siirrytään ohjelman testaukseen. Ohjelman toiminnan testauksen olin määrittänyt rakentumaan vaiheista. Ensin paikallisesti toteutetun kommunikaation toiminnallisuuden varmistaminen. Toisessa testivaiheessa testattiin, että kommunikaatio toimii yli verkon. Viimeisessä testivaiheessa oli käytettävissä samat laitteet, kuten yrityksen oikeassa automaatiojärjestelmässä.

Tässä työssä rakennettu ohjelma tullaan jalkauttamaan yrityksen automaatiojärjestelmiin seuraavien suurten päivityksien aikana.

Lähteet

1 MariMatic. 2022. MariMatic - AWCS. Verkkoaineisto. <https://www.marimatic.com/>. Luettu 22.04.2022.

2 IPCOMM GmbH. 2022. MQTT. Verkkoaineisto. <https://www.ipcomm.de/protocol/MQTT/en/sheet.html>. Luettu 06.04.2022.

3 MQTT. 2022. MQTT: The Standard for IoT Messaging. Verkkoaineisto. <https://mqtt.org/>. Luettu 03.04.2022.

4 The HiveMQ Team. 12.01.2015. Introducing the MQTT Protocol - MQTT Essentials: Part 1. Verkkoaineisto. <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>. Luettu 03.04.2022.

5 EMQ Technologies Co., Ltd. 07.07.2020. What is the MQTT protocol. Verkkoaineisto. <https://www.emqx.com/en/blog/what-is-the-mqtt-protocol>. Luettu 05.07.2022.

6 The HiveMQ Team. 19.01.2015. Publish & Subscribe - MQTT Essentials: Part 2. Verkkoaineisto. <https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe/>. Luettu 03.04.2022.

7 EMQ Technologies Co., Ltd. 17.03.2020. Introduction to MQTT publish-subscribe model. Verkkoaineisto. <https://www.emqx.com/en/blog/mqtt-5-introduction-to-publish-subscribe-model>. Luettu 07.04.2022.

8 Mary E. Shacklett, Amy Novotny ja Kate Gerwig. 13.07.2021. TCP/IP. Verkkoaineisto. <https://www.techtarget.com/searchnetworking/definition/TCP-IP>. Luettu 13.04.2022.

9 The HiveMQ Team. 17.07.2019. MQTT Client and Broker and MQTT Server and Connection Establishment Explained - MQTT Essentials: Part 3. Verkkoaineisto. <https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/>. Luettu 06.04.2022.

10 The HiveMQ Team. 16.03.2015. Keep Alive and Client Take-Over - MQTT Essentials Part 10. Verkkoaineisto. <https://www.hivemq.com/blog/mqtt-essentials-part-10-alive-client-take-over/>. Luettu 10.04.2022.

11 Corinne Bernstein, Kate Brush ja Alexander S. Gillis. 27.01.2021. MQTT (MQ Telemetry Transport). Verkkoaineisto. <https://www.techtarget.com/iotagenda/definition/MQTT-MQ-Telemetry-Transport>. Luettu 13.04.2022.

12 Andrew Banks, Ed Briggs, Ken Borgendale ja Rahul Gupta. 07.03.2019. MQTT Version 5.0. Verkkoaineisto. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf>. Luettu 12.04.2022.

13 Lee Stacey. 16.06.2020. MQTT beginner’s guide. Verkkoaineisto. <https://www.u-blox.com/en/blogs/insights/mqtt-beginners-guide>. Luettu 12.04.2022.

14 Javatpoint. 2021. MQTT protocol. Verkkoaineisto. <https://www.javatpoint.com/mqtt-protocol>. Luettu 13.04.2022.

15 Roger Light.20.09.2018. MQTT man page. Verkkoaineisto. <https://mosquitto.org/man/mqtt-7.html>. Luettu 10.04.2022.

16 Packt. 23.07.2017. Understanding wildcards. Verkkoaineisto. <https://subscription.packtpub.com/book/application_development/9781787287815/1/ch01lvl1sec18/understanding-wildcards>. Luettu 13.04.2022.

17 Steve Cope. 28.01.2021. Understanding the MQTT Protocol Packet Structure. Verkkoaineisto. <http://www.steves-internet-guide.com/mqtt-protocol-messages-overview/>. Luettu 15.04.2022.

18 EMQ Technologies Co., Ltd. 27.02.2020. How to use MQTT packet to implement publish and subscribe. Verkkoaineisto. <https://www.emqx.com/en/blog/how-to-use-mqtt-packet-to-implement-publishing-and-subscribing-functions>. Luettu 07.04.2022.

19 The HiveMQ Team. 02.02.2015. MQTT Publish, Subscribe & Unsubscribe - MQTT Essentials: Part 4. Verkkoaineisto. <https://www.hivemq.com/blog/mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe/>. Luettu 03.04.2022.

20 Steve Cope. 16.12.2021. MQTT Publish and Subscribe Beginners Guide. Verkkoaineisto. <http://www.steves-internet-guide.com/mqtt-publish-subscribe/>. Luettu 04.04.2022.

21 Assetwolf. 08.02.2022. Quality of Service (QoS). Verkkoaineisto. <https://assetwolf.com/learn/mqtt-qos-understanding-quality-of-service>. Luettu 10.04.2022.

22 The HiveMQ Team. 23.02.2015. Persistent Session and Queuing Messages - MQTT Essentials: Part 7. Verkkoaineisto. <https://www.hivemq.com/blog/mqtt-essentials-part-7-persistent-session-queuing-messages/>. Luettu 04.04.2022.

23 EMQ Technologies Co., Ltd. 18.09.2020. MQTT Session. Verkkoaineisto. <https://www.emqx.com/en/blog/mqtt-session>. Luettu 07.04.2022.

24 The HiveMQ Team. 02.03.2015. Retained Messages - MQTT Essentials: Part 8. Verkkoaineisto. <https://www.hivemq.com/blog/mqtt-essentials-part-8-retained-messages/>. Luettu 12.04.2022.

25 EMQ Technologies Co., Ltd. 06.11.2019. MQTT Retain Message. Verkkoaineisto. <https://www.emqx.com/en/blog/mqtt5-features-retain-message>. Luettu 12.04.2022.

26 Thingrex. 16.10.2021. MQTT Retained messages. Verkkoaineisto. <https://www.thingrex.com/retained_messages/>. Luettu 15.04.2022.

27 EMQ Technologies Co., Ltd. 23.08.2021. MQTT security: introduction. Verkkoaineisto. <https://www.emqx.com/en/blog/mqtt-security-part-1-introduction>. Luettu 11.04.2022.

28 The HiveMQ Team. 11.05.20215. TLS/SSL - MQTT Security Fundamentals. Verkkoaineisto. <https://www.hivemq.com/blog/mqtt-security-fundamentals-tls-ssl/>. Luettu 12.04.2022.

29 The HiveMQ Team. 20.04.2015. HiveMQ - MQTT Security Fundamentals. Verkkoaineisto.

<https://www.hivemq.com/blog/introducing-the-mqtt-security-fundamentals/>. Luettu 11.04.2022.

30 The HiveMQ Team. 16.06.2015. Securing MQTT Systems - MQTT Security Fundamentals. Verkkoaineisto. <https://www.hivemq.com/blog/mqtt-security-fundamentals-securing-mqtt-systems/>. Luettu 12.04.2022.

31 Ilya Grigorik. 15.10.2013. Transport Layer Security (TLS). Verkkoaineisto. <https://hpbn.co/transport-layer-security-tls/>. Luettu 12.04.2022.

32 Hanna Kangasniemi ja Matti Lintulahti. 10.01.2017. Mikä on pilvipalvelu?. Verkkoaineisto. <https://elisa.fi/ideat/mika-on-pilvipalvelu/>. Luettu 07.04.2022.

33 Azure. 21.11.2015. What is cloud computing?. Verkkoaineisto. <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/>. Luettu 07.04.2022.

34 Sai Vennam. 18.08.2020. Cloud Computing. Verkkoaineisto. <https://www.ibm.com/cloud/learn/cloud-computing>. Luettu 13.04.2022.

35 Red Hat Inc. 15.03.2018. Types of cloud computing. Verkkoaineisto. <https://www.redhat.com/en/topics/cloud-computing/public-cloud-vs-private-cloud-and-hybrid-cloud>. Luettu 13.04.2022.

36 Beckhoff Automation GmbH & Co. KG. 2022. Beckhoff Automation. Verkkoaineisto. <https://www.beckhoff.com/fi-fi/company/>. Luettu 20.04.2022.

37 Beckhoff Automation GmbH & Co. KG. 2022. TwinCAT automation software. Verkkoaineisto. <https://www.beckhoff.com/fi-fi/products/automation/twincat/>. Luettu 20.04.2022.

38 Beckhoff Automation GmbH & Co. KG. 2022. TF6xxx | TwinCAT 3 Connectivity. Verkkoaineisto. <https://www.beckhoff.com/fi-fi/products/automation/twincat/tfxxxx-twincat-3-functions/tf6xxx-tc3-connectivity/>. Luettu 20.04.2022.

39 Siemens AG. 2022. About us. Verkkoaineisto. <https://new.siemens.com/global/en/company/about.html>. Luettu 21.04.2022.

40 Siemens AG. 17.08.2021. Industrial Edge: the optimal technology mix. Verkkoaineisto. <https://new.siemens.com/global/en/company/stories/industry/industrial-edge.html>. Luettu 21.04.2022.

41 Siemens AG. 06.05.2021. Industrial Edge for IT specialists. Verkkoaineisto. <https://new.siemens.com/global/en/products/automation/topic-areas/industrial-edge/it-specialists.html>. Luettu 20.04.2022.

42 Siemens AG. 04.2021. Industrial Edge Flow Creator. Verkkoaineisto. <https://cache.industry.siemens.com/dl/files/331/109794331/att_1057284/v1/IE_flow_creator_operation_en-US.pdf>. Luettu 20.04.2022.

43 OpenJS Foundation. 16.03.2017. About. Verkkoaineisto. <https://nodered.org/about/>. Luettu 20.04.2022.

44 Eclipse Foundation, Inc. 08.01.2018. Eclipse Mosquitto™ An open source MQTT broker. Verkkoaineisto. <https://mosquitto.org/>. Luettu 20.04.2022.

45 ISO. 2022. ISO/IEC 27001 Information security management. Verkkoaineisto. <https://www.iso.org/isoiec-27001-information-security.html>. Luettu 20.04.2022.

46 Ben Lutkevich ja Madelyn Bacon. 25.06.2021. end-to-end encryption (E2EE). Verkkoaineisto. <https://www.techtarget.com/searchsecurity/definition/end-to-end-encryption-E2EE>. Luettu 22.04.2022.

47 Cisco Systems Inc. 2022. What Is a Firewall?. Verkkoaineisto. < https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html>. Luettu 21.04.2022.

48 Beckhoff Automation GmbH & Co. KG. 24.11.2021. TF6701 TwinCAT 3 | IoT Communication (MQTT). Verkkoaineisto. <https://download.beckhoff.com/download/document/automation/twincat3/TF6701_TC3_IoT_Communication_MQTT_EN.pdf>. Luettu 18.04.2022.

49 Beckhoff Automation GmbH & Co. KG. 2022. ADR (Address Operator). Verkkoaineisto. <https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/TcPlcCtrl_ADR.htm&id=>. Luettu 18.04.2022.