

Topias Välimäki

TwinCAT-ympäristön yhdistäminen teolliseen internetiin

Opinnäytetyö

Kevät 2020

SeAMK Tekniikka

Automaatiotekniikka



SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Tutkinto-ohjelma: Automaatiotekniikka

Suuntautumisvaihtoehto: Sähköautomaatio

Tekijä: Topias Välimäki

Työn nimi: TwinCAT-ympäristön yhdistäminen teolliseen internetiin

Ohjaaja: Jyri Lehto

Vuosi: 2020

Sivumäärä: 59

Tämän opinnäytetyön tilasi seinäjokinen mekaniikka- ja automaatioalan suunnittelutoimisto Caplan Oy. Yrityksellä oli tarve tutkia tapoja lähettää dataa TwinCAT-ympäristöstä pilvipalveluun. Työssä haluttiin myös tutustua ennalta valittuihin IoT-palveluihin.

Tutkimuksen teoriaosuuteen haettiin tietoa pilvipalveluiden erilaisista rakenteista ja pohdittiin teollisen internetin tuomia mahdollisuuksia ja haasteita. TwinCAT-ympäristön rajapintoihin perehtymisen jälkeen tutustuttiin neljään IoT-ratkaisuun: Microsoft Azure- ja Amazon AWS -palveluihin, Beckhoff TwinCAT Analytics -tuoteperheeseen sekä kotimaiseen Wapice IoT-TICKET -alustaan. Tämän jälkeen tutkimuksessa testattiin tapoja tiedon lähettämiseen käyttäen eri ohjelmointikieliä ja Beckhoff IoT Data Agent -sovellusta.

Yritys sai tutkimuksen tuloksena tietoa markkinoilla olevista pilvipalveluista ja -alustoista. Koottuun tietopakettiin sisällytettiin metodeja, jotka lukevat dataa TwinCAT-ohjelmasta ja lähettävät sitä eteenpäin MQTT- ja HTTP-kommunikaatioprotokollia käyttäen.

Avainsanat: IoT, teollinen internet, pilvipalvelu, TwinCAT, MQTT, HTTP

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Automation Engineering

Specialisation: Electrical Automation

Author: Topias Välimäki

Title of thesis: Connecting a TwinCAT environment to the Industrial Internet of Things

Supervisor: Jyri Lehto

Year: 2020

Number of pages: 59

The thesis was commissioned by Caplan Oy, an engineering office in Seinäjoki. The company expressed a need to learn new ways to collect and transfer data. The focus of this study was to examine different possibilities of connecting an IoT platform and a TwinCAT environment.

The research process included gathering information about cloud services along with the possibilities and challenges of the Industrial Internet of Things. TwinCAT environment and its interfaces were also acquainted with. In addition, the following four separate IoT solutions were examined: the cloud services Microsoft Azure and Amazon AWS, the data analysis product family TwinCAT Analytics and the IoT platform Wapice IoT-TICKET. Data was transferred by using a few programming languages and the interface application Beckhoff IoT Data Agent.

As the outcome of the study the company got information on the cloud services and platforms available on the market. Furthermore, some data transfer methods using the communication protocols MQTT and HTTP were included in the documentation.

Keywords: IoT, Industrial Internet of Things, cloud service, TwinCAT, MQTT, HTTP

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ	4
Kuvio- ja taulukkoluetelo.....	6
Käytetyt termit ja lyhenteet	8
1 JOHDANTO	10
1.1 Työn tausta	10
1.2 Työn tavoitteet.....	10
1.3 Työn rakenne	11
1.4 Caplan Oy	11
2 PILVIPALVELUT	12
2.1 Termejä.....	12
2.2 Erilaiset pilvi-infrastruktuurit	13
2.3 Esineiden internet kuluttajille.....	15
3 ESINEIDEN INTERNET TEOLLISUUDESSA	16
3.1 Teollinen internet.....	16
3.2 Ennakoiva kunnossapito	17
3.3 Teollinen internet liiketoimintamallien muuttajana	17
3.4 Teollisen internetin kasvun haasteet	20
4 TWINCAT-YMPÄRISTÖ	21
4.1 Beckhoff Automation	21
4.2 ADS.....	22
4.3 MQTT	22
4.4 REST	24
5 IOT-ALUSTOJEN KÄYTTÖKOKEMUKSIA	25
5.1 Microsoft Azure	25
5.2 Amazon AWS IoT.....	27
5.3 TwinCAT Analytics	30
5.4 Wapice IoT-TICKET	35
6 DATAN LÄHETTÄMINEN	41

6.1 Parametrit erillisessä ohjelmassa.....	41
6.2 C#-ohjelma.....	42
6.3 Python-ohjelma	45
6.4 IoT Data Agent.....	48
6.5 PLC-ohjelma	49
7 YHTEENVETO.....	52
7.1 Tutkimuksen tulokset	52
7.2 Pohdinta.....	53
LÄHTEET	55

Kuvio- ja taulukkoluetelo

Kuvio 1. Pilvi-infrastruktuurit.....	14
Kuvio 2. Datan lähetys gateway-tietokoneella.	19
Kuvio 3. MQTT-kommunikointi.....	23
Kuvio 4. Microsoft Power BI verkkoselain-, työpöytä- tai mobiilisovelluksena.....	26
Kuvio 5. PLC-ohjelman parametrit Azure-palveluun yhdistettäessä.....	27
Kuvio 6. Amazon AWS-palvelun sovelluksia.....	28
Kuvio 7. IoT Data Agent -sovelluksen parametrit AWS-palveluun yhdistettäessä.	29
Kuvio 8. Amazon AWS IoT Monitor.....	30
Kuvio 9. TwinCAT Analytics Networks.	31
Kuvio 10. Muuttujat TwinCAT Analytics -ohjelmassa.	32
Kuvio 11. Muuttujien yhdistäminen TwinCAT HMI -käyttöliittymään.	32
Kuvio 12. TwinCAT Analytics yrityksen sisäisessä verkossa.	33
Kuvio 13. TwinCAT Analytics käytössä verkon yli.....	34
Kuvio 14. IoT-TICKET-palvelun työkaluja.	35
Kuvio 15. Datanode-struktuurit IoT-TICKET-palvelun hierarkiassa.....	36
Kuvio 16. Rakennettu käyttöliittymä IoT-TICKET-palvelussa.	38
Kuvio 17. IoT-TICKET-palvelun esimerkinäkymä.....	38
Kuvio 18. Hälytyksen aktivointi IoT-TICKET-palvelussa.	39
Kuvio 19. Raportin lähetys IoT-TICKET-palvelusta sähköpostiin.....	40
Kuvio 20. Kahvamuuttujien ja ADS-yhteyden määrittely C#-ohjelmassa.	43

Kuvio 21. Muuttujien lukeminen C#-ohjelman AdsStream-datavirrasta.	43
Kuvio 22. IoTTicketClient-olio C#-ohjelmassa.	44
Kuvio 23. ADS-yhteyden muodostaminen C#-ohjelmasta PLC-ohjelmaan.....	44
Kuvio 24. Laitteen attribuutit C#-ohjelmassa.....	44
Kuvio 25. Laitteen rekisteröinti C#-ohjelmassa.	45
Kuvio 26. Esimerkki datanode-muuttujasta C#-ohjelmassa.	45
Kuvio 27. Yhteyden muodostaminen Python-ohjelmasta paikalliseen PLC-ohjelmaan.	46
Kuvio 28. Muuttujien lukeminen nimen perusteella Python-ohjelmassa.....	46
Kuvio 29. Muuttujien kirjoitus muistiavaruuteen viittaamalla Python-ohjelmassa..	46
Kuvio 30. Muuttujan kirjoittaminen kahvamuuttujan avulla Python-ohjelmassa. ...	47
Kuvio 31. Viestin lähettäminen (publish) Python-ohjelmassa.....	47
Kuvio 32. Viestin lähettäminen (post) Python-ohjelmassa.	47
Kuvio 33. IoT Data Agent -sovelluksen topologianäkymä.	48
Kuvio 34. Muuttujien valitseminen IoT Data Agent -sovellukseen.....	49
Kuvio 35. MQTT-yhteyden parametrit IoT Data Agent -sovelluksessa.....	49
Kuvio 36. Parametrit MQTT-yhteyden muodostamiseen PLC-ohjelmasta.	50
Kuvio 37. Yhteyden muodostus PLC-ohjelmasta.	50
Kuvio 38. Viestin vastaanottaminen PLC-ohjelmassa.....	51
Kuvio 39. Viestin lähettäminen PLC-ohjelmasta.	51
Taulukko 1. ADS-portit TwinCAT-ympäristössä.	42

Käytetyt termit ja lyhenteet

ADS-kommunikaatio	ADS-kommunikaatioprotokollaa (Automation Device Specification) käytetään TwinCAT-ympäristön eri työkalujen väliseen kommunikointiin (Beckhoff, [viitattu 16.1.2020]).
API	Eri sovellusten välinen rajapinta (Application Programming Interface, suom. ohjelmointirajapinta), kuten kommunikaatioprotokolla.
Big data	Valtaisa datamäärä (suom. massadata), jonka kerääminen, säilyttäminen ja analysointi eivät onnistu käyttäen perinteisiä datanhallinnointitapoja (Hilbert 2013).
Data-analytiikka	Menetelmä, jolla datasta muodostetaan erilaisten laskentamallien avulla hyödyllistä informaatiota (Academy 2019).
HMI	Käyttöliittymä (Human-Machine Interface) ihmisen ja ohjelmoitavan logiikan välille.
HTML	Merkintäkieli (Hypertext Markup Language), jolla kirjoitetut komennot verkkoselain kääntää visuaaliseen muotoon (Abeyasinghe 2008, 11).
HTTP	Tiedonsiirtoprotokolla (Hypertext Transfer Protocol), jota käytetään verkkoliikenteessä (Abeyasinghe 2008, 11).
IIoT	Teollisuuden esineiden internet tai teollinen internet (Industrial Internet of Thing), joka kuvastaa teollisuustuotannon laitteiden yhdistämistä internetiin.
IoT	Esineiden internet (Internet of Things), joka viittaa internetiin yhdistettyjen laitteiden kokonaisuuteen.
MQTT	Kevyt tiedonsiirtoprotokolla (Message Queueing Telemetry Transport), joka perustuu julkaisija-tilaajamalliin (Beckhoff 2018a, 36).

OPC UA	OPC (Open Platform Communications) on ryhmä teollisuusautomaation tiedonsiirtostandardeja. UA-loppuliite (Unified Architecture) käsittää vuonna 2008 päivitetyn määrittelyn. (OPC Foundation, [viitattu 14.1.2020].)
Pilvipalvelu	Verkon yli tarjottavaa tietojenkäsittelyä, kuten tietokanta- ja analytiikkapalveluja, sekä tiedon visuaalista esittämistä (Microsoft, [viitattu 15.1.2020]).
PLC	Ohjelmoitava logiikka tai logiikkaohjain (Programmable Logic Controller), joka on laitetta tai prosessia automaattisesti ohjaava tietokone.
Retrofit	Jälkiasennus. Verbi retrofitting (suom. asentaa tai varustaa jälkikäteen) tarkoittaa järjestelmän päivittämistä uudella osalla tai sovelluksella.
SQL-kieli	IBM:n kehittämä tiedon käsittelykieli (engl. Structured Query Language) relaatiotietokantoihin tallennetun datan hakemiseen, muuttamiseen tai lisäämiseen.
TCP/IP	Engl. Transmission Control Protocol / Internet Protocol, joka on usean verkkoliikenneprotokollan yhdistelmä (Marshall & Rinaldi 2017, 61-64).
TwinCAT	The Windows Control and Automation Technology on Beckhoffin PC-pohjainen järjestelmän ohjaus- ja ohjelmointiympäristö.

1 JOHDANTO

1.1 Työn tausta

Tämä opinnäytetyö tehtiin seinäjokiselle suunnittelutoimistolle Caplan Oy:lle. Työ käsittelee pilvipalveluiden hyödyntämistä ja datan keräämistä teollisuuden näkökulmasta. Datan kerääminen ja analysointi ovat olleet viime aikoina esillä tekniikan alalla. Ennakoiva kunnossapito, etävalvonta ja -hallinta sekä tuotantotapahtumien tilastointi ovat osa teollisen internetin tuomista mahdollisuuksista. Tällä opinnäytetyöllä pyrittiin tuomaan esille teollisuuteen sopivien pilvialustojen ominaisuuksia ja hyötyjä.

Pilvialustoja ja valmiita pilvipalveluita ovat tuoneet markkinoille useat sovelluskehittäjät ja laitevalmistajat. Muun muassa Schneider Electric tarjoaa EcoStruxure-datan käsittelysovelluksia ja Siemens omaa MindSphere-nimistä IoT-ympäristöään. Teollisuudessa on käytössä tiettyjä valmistajasta riippumattomia tiedonsiirtostandardeja, mutta yhteydenmuodostusmenetelmät eri valmistajien palveluihin eivät suinkaan ole yhdenmukaista. Tässä työssä keskityttiinkin tutkimaan eri tapoja lähettää dataa automaatioyritys Beckhoffin TwinCAT 3 -ympäristöstä pilvipalveluihin.

1.2 Työn tavoitteet

Miten dataa lähetetään logiikkaohjaimelta pilvipalveluun? Mitä työkaluja on käytettävissä datan analysointiin? Näiden kysymysten pohjalta aloitettiin tämä Caplan Oy:n tilaama tutkimustyö. Aluksi oli tutustuttava muutama dataa vastaanottavaan ja analysoivaan IoT-ratkaisuun. Toiseksi haluttiin selvittää, miten dataa voidaan lähettää TwinCAT-ympäristöstä pilvipalveluun. Työn aikana oli myös tutkittava, mitä hyötyä datan keräämisestä ja analysoinnista on asiakkaalle.

1.3 Työn rakenne

Työn toisessa luvussa käydään läpi pilvipalveluita yleisellä tasolla. Luvussa esitellään erilaisia pilvirakenteita sekä esineiden internetin hyötyä tavalliselle kuluttajalle. Kolmannessa luvussa asiaa käsitellään teollisuuden näkökulmasta. Tämän jälkeen tutustutaan TwinCAT-ympäristöön ja työssä käytettäviin rajapintoihin. Seuraavaksi työssä esitellään neljä IoT-alustaa teollisuuden sovelluskohteita silmällä pitäen. Esiteltävät palvelut ovat erilaisia ratkaisuja, eikä niitä verrata keskenään. Eri mahdollisuuksia datan lähettämiseen tuodaan esille luvussa kuusi. Tulokset kootaan yhteen seitsemännessä luvussa, jossa pohditaan lisäksi tutkimuksen vaiheita.

1.4 Caplan Oy

Työn toimeksiantaja Caplan Oy on mekaniikka- ja automaatioalan suunnittelutoimisto. Yritys perustettiin vuonna 2015 Seinäjoella. Maaliskuussa 2020 se työllisti 16 henkeä. (Caplan, [viitattu 3.4.2020].)

Yritys tarjoaa usealle teollisuuden alalle suuntautuvaa mekaniikka- ja automaatio-suunnittelua. Mekaniikkasuunnittelu käsittää tuotesuunnittelun, tuotekehityksen ja konseptisuunnittelun. Yrityksen yksi vahvoista osaamisalueista on myös tekninen laskenta, johon sisältyvät tuotteiden lujuuslaskenta ja virtuaaliset analyysityökalut. Automaatiosuunnittelupalveluita tarjotaan laitevalmistajille ja tehdasympäristöihin. Palveluihin kuuluvat muun muassa piirikaavio- ja layoutsuunnittelu, logiikkaohjelmointi ja käyttöliittymät sekä testaus ja käyttöönotto. (Caplan, [viitattu 3.4.2020].)

2 PILVIPALVELUT

2.1 Termejä

Tietotekniikan alalla käytetään nykyään usein sanaa pilvi. Tähän yhteyteen liittyy lisää termejä, kuten pilvipalvelu ja pilvilaskenta. Pilvipalvelua kutsutaan joskus nimellä pilvialusta. Alusta- ja palvelu-termien eroa onkin vaikea määrittää. Yhdyssanan loppuosa ”alusta” kuvastaa vielä keskeneräistä pilvipalvelua, jonka asiakas rakentaa tai rakennuttaa omaa liiketoimintaansa hyödyttäväksi palveluksi. Lamarre ja May (2017) kuvailevat konsulttijätti McKinsey & Companyn artikkelissa pilvialustan koostuvan ohjelmistoista ja laitteistosta eli muun muassa käyttöliittymästä, muistista, laskentatehosta, tietoturvasta ja kehitystyökaluista.

Termin ”IoT” (Internet of Things) trendikkyys on viime vuosina kasvanut ja termi esiintyy usein teollisuuden sovelluskohteisiin tarkoitettujen pilvipalveluiden tai -alustojen nimessä. IoT-palveluita ja -alustoja on saatavissa useille aloille, lukemattomiin käyttötarkoituksiin. Tee-se-itse-palvelut eli IoT-alustat ovat yleisiä teollisuudessa, jossa asiakkaiden erilaiset tarpeet huomioon ottaen on mahdollista rakentaa yleispätevää datan keruu- ja analysointipalvelua. Tämä opinnäytetyö keskittyy pilvialustoihin juuri teollisuuden näkökulmasta.

Pilvi-sana tarkoittaa tietotekniikassa palvelimien verkostoa. Palvelimilla sijaitsee käyttäjien tarvitsema tieto tai jokin sovellus, kuten sähköposti. Tavalliselle käyttäjälle pilvipalvelu, kuten Microsoftin tarjoama tiedostojen tallennuspaikka OneDrive, tarkoittaa tietojen säilyttämistä fyysisestä laitteesta riippumatta sekä tietojen helppoa jakamista turvallisesti ja nopeasti. (Kangasniemi & Lintulahti, [viitattu 16.1.2020].)

Monet pilvipalvelut, kuten Amazon AWS tai Microsoft Azure, tarjoavat lisäksi datan analytiikkaa ja pilvilaskentaa (Amazon Web Services, [viitattu 15.1.2020]). Pilvilaskennalla (engl. cloud computing) tarkoitetaan tietotekniikkapalvelujen käyttämistä verkon yli, yleensä pay-as-you-go-tyyppisellä maksupolitiikalla. Pay-as-you-go-mallissa maksaminen tapahtuu palvelua käytettäessä eikä esimerkiksi etukäteismak-

suilla. Tällainen maksupolitiikka tekee palvelusta skaalautuvan. Skaalautuvuus tarkoittaa lisäresurssien, kuten ylimääräisen laskentatehon tai tietokannan, helppoa saatavuutta. (Cloud Carib, [viitattu 16.1.2020].)

2.2 Erilaiset pilvi-infrastruktuurit

Pilvipalvelut jaetaan usein kolmeen pääluokkaan:

- SaaS (Software as a Service)
- PaaS (Platform as a Service)
- IaaS (Infrastructure as a Service). (IBM, [viitattu 19.1.2020].)

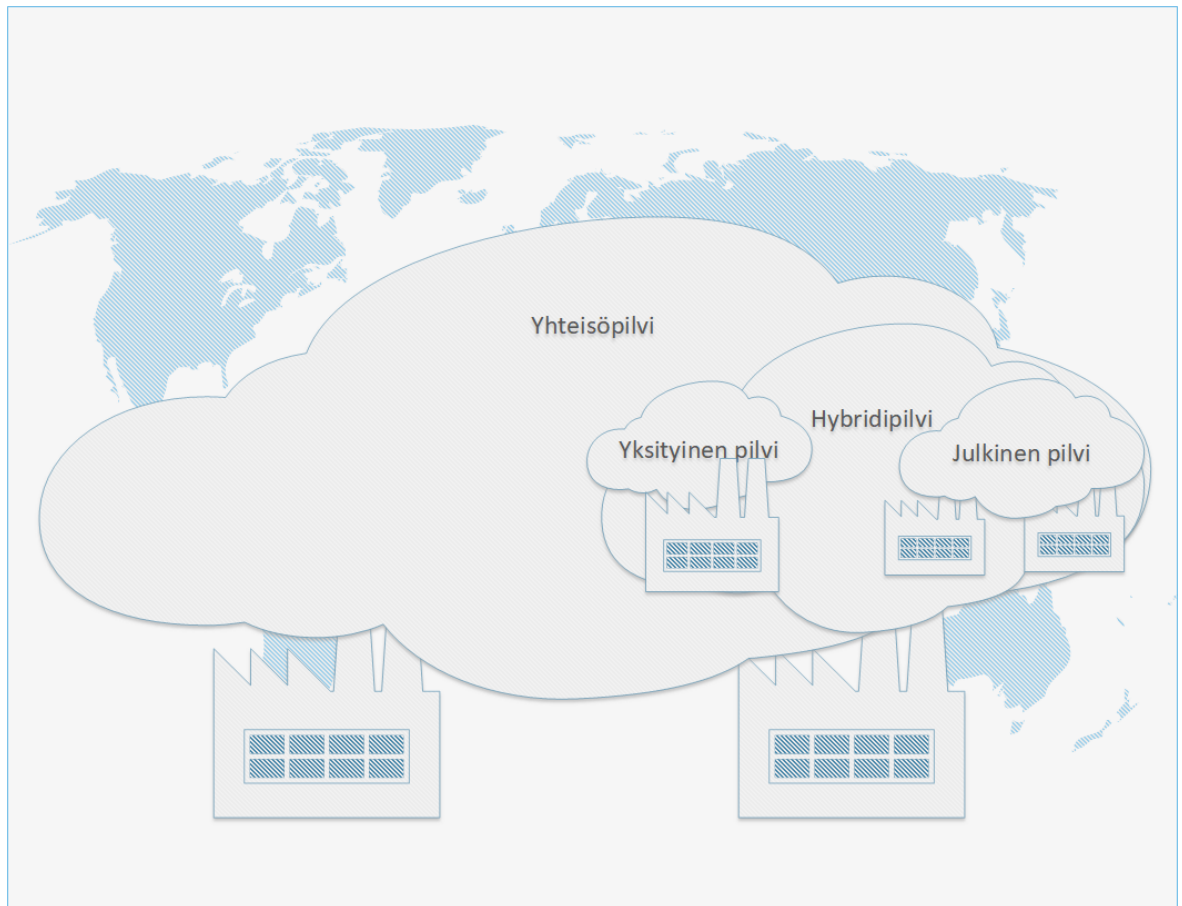
SaaS-palvelu tarkoittaa sovelluksen, kuten sähköpostin, käyttämistä verkon kautta. Käyttäjien ei tällöin tarvitse hallinnoida tai päivittää sovellusta. PaaS-palvelu puolestaan tarjoaa käyttäjälleen mahdollisuutta kehittää ja hallinnoida sovelluksiaan. Käyttäjä voi hyödyntää palveluntarjoajan työkaluja kehittääkseen ja testatakseen omia sovelluksiaan. IaaS-palvelu taas ei välttämättä sisällä ollenkaan valmiita sovelluksia tai palveluja vaan antaa asiakkaalle vapauden rakentaa oman pilvialustansa ja sovelluksensa palveluntarjoajan infrastruktuuriin eli palvelimiin. IaaS tarkoittaakin vain tiedon tallennustilan ja laskentatehon myymistä asiakkaalle. (IBM, [viitattu 19.1.2020].)

Pilvipalvelut voidaan jaotella saatavuuden perusteella neljään erilaiseen ympäristöön (kuvio 1):

- julkinen pilvi (engl. public cloud)
- yksityinen pilvi (engl. private cloud)
- hybridipilvi (engl. hybrid cloud)
- yhteisöpilvi (engl. community cloud). (Siemens, [viitattu 20.1.2020].)

Nimensä mukaisesti julkinen pilvi on kaikkien käytettävissä – vastaavasti yksityinen pilvi tarkoittaa vain yhden yrityksen tai organisaation käytössä olevaa pilvialustaa.

Hybridipilvi yhdistää nämä kaksi ratkaisua: lisäresurssien saamiseksi yksityiseen pilveen yhdistetään julkinen pilvi. Yhteisöpilvessä infrastruktuuri ja resurssit jaetaan useammalle yritykselle tai organisaatiolle. (Siemens, [viitattu 20.1.2020].)



Kuvio 1. Pilvi-infrastruktuurit.

Pilvipalvelun rakenteita eli arkkitehtuurimalleja voi olla kahdenlaisia: keskitettyjä ja hajautettuja. Keskitetyn arkkitehtuurin mallissa palvelut, datan kokoamiset ja niitä lähettävien laitteiden hallinta on koottu yhteen taustajärjestelmään. Dataa lähettävät tai lukevat laitteet eivät ole suorassa yhteydessä keskenään. Monesti tällainen arkkitehtuurimalli on riittävä. Palvelua käyttävän organisaation vaatiessa pitkälle kehittyntä ja monimutkaista teollisen internetin ratkaisua on käytettävä hajautettua arkkitehtuuria. Myös tässä mallissa taustalla on keskitetty järjestelmä, mutta päätepiisteet pystyvät kommunikoimaan keskenään itsenäisesti. Tällöin päätepiisteet ovat älykkäitä antureita. Hajautetussa arkkitehtuurimallissa tietoliikenteen viive saadaan erittäin lyhyeksi. (Collin & Saarelainen 2016, luku 11.)

2.3 Esineiden internet kuluttajille

Tavalliselle kuluttajalle suunnatut kodin laitteita verkkoon yhdistävät sovellukset kehittyvät jatkuvasti. Markkinoilla on saatavissa muun muassa mobiililaitteilla ohjattavia robotti-imureita, kodin älykkäitä turvaratkaisuja ja erilaisia valaistuksen ohjausjärjestelmiä. Liiketunnistimella varustetut kamerat lähettävät mahdollisesta tunkeutujasta ilmoituksen älypuhelimeen – tarvittaessa jopa valokuvan kera suoraan hälytyskeskukseen. Myös etävalvontajärjestelmään kuuluvat palovaroittimet, älylukot ja kosteudenilmaisimet luovat asunnon omistajalle turvallisuuden tunnetta. Sähköisten kaihtimien, valaisimien ja muiden kodin laitteiden hallinta onnistuu ääniohjauksella, vaikkapa suosittua Amazon Echo -älykaiutinta käyttäen. Mobiililaitteella voi säätää myös asunnon lämpötilaa ja ilmanvaihtoa langattomaan WiFi-verkkoon yhdistettyjen termostaattien avulla. (Kodinturvatieto 2019; CNET 2020.)

3 ESINEIDEN INTERNET TEOLLISUUDESSA

3.1 Teollinen internet

Teollinen internet eli IIoT (Industrial Internet of Things) käsitteenä saa aikaan mielenkuvan tuotantoteollisuuden laitteiden liittamisestä verkkoon. Kuitenkin General Electric Companyn julkaisemassa artikkelissa vuodelta 2012 teollinen internet koostuu teollisuuden laitteiden yhdistämisen, data-analyysin ja ennustavien algoritmien lisäksi myös esimerkiksi liikenteen älykkäästä ohjaamisesta ja terveydenhuollon laitteista (Evans & Annunziata 2012, 3).

Teollisen internetin käyttökohteet ja sovellukset yleistyvät niin Suomessa kuin muuallakin. Erilaisia ratkaisuja keksitään jatkuvasti, ja ne voidaan jakaa neljään sovel-lusalueeseen:

- laitteiden etäkäyttö: valvonta, hallinta, päivitykset ja optimointi
- kunnossapidon ennakointi sekä analytiikka
- palveluliiketoiminnan uudistaminen datapohjaiseksi
- tehtaan laitteiden autonominen ja älykäs toiminta. (Collin & Saarelainen 2016, luku 5.)

Teollista internetiä soveltava tehdas voidaan parhaimmillaan rinnastaa itseohjautuvaan autoon. Älykäs ja autonominen tehdas toimisi itsenäisesti tuotetilausten mukaan hoitaen itse tuotantoprosessin sekä ulkoiset tarviketilaukset. Tällainenkin tehdas tarvitsee silti ihmisiä, joiden tehtävänä on valvoa prosessia, parantaa ohjauksen algoritmeja sekä keksiä muita toimenpiteitä tuottavuuden parantamiseksi ja häiriöiden korjaamiseksi. Uusien etävalvonta- ja hallintajärjestelmien myötä etätyöskente-lyn mahdollisuus kasvaa. (Collin & Saarelainen 2016, luku 5.)

Etävalvontajärjestelmä kuulostaa hienolta, mutta sen rakentaminen voi olla haastavaa. Automaatiojärjestelmiä on ollut Suomessa vuosikymmeniä, mutta anturien datan yhtenäistäminen, toimivien tietoliikenneyhteyksien rakentaminen ohjainlogiikan ja pilvipalvelun välille sekä itse datan esittäminen järkevällä tavalla pilvipalvelun

käyttöliittymässä eivät ole helposti toteutettavissa. Vaativaa on myös datan analysointi, johon sisältyvät turhan datan pois jättäminen ja tärkeän datan korostaminen. (Collin & Saarelainen 2016, luku 5.)

3.2 Ennakoiva kunnossapito

Laitteiden kunnossapito helpottuu, kun vikaantumisia kyetään ennustamaan. Teollisen internetin odotetaan suuren auttavan tätä toimialaa. On alettu käyttää termejä ”ennakoiva huolto” ja ”etäkunnonvalvonta”. Hyvin toimivalla kunnossapidolla voi olla suurikin vaikutus yrityksen kannattavuuteen, sillä tällöin laitteiden käyttöaste paranee ja tuotantokatkokset vähenevät. Kunnossapidossa pilvipalvelun analytiikka ja ilmoitusominaisuudet korvaavat perinteisen päivystyksen. Palvelu voi ilmoittaa kunnossapitoryhmälle anomaliaista eli poikkeamista, esimerkiksi lämpötilan raja-arvon ylityksestä, tavallista suuremmasta virrankulutuksesta tai moottorinsuojakytkimen laukeamisesta. Ilmoitusmenetelmiä on useita. Näitä ovat esimerkiksi tekstiviesti, sähköposti tai mobiilisovelluksen lähettämä push-ilmoitus. (Collin & Saarelainen 2016, luku 5.)

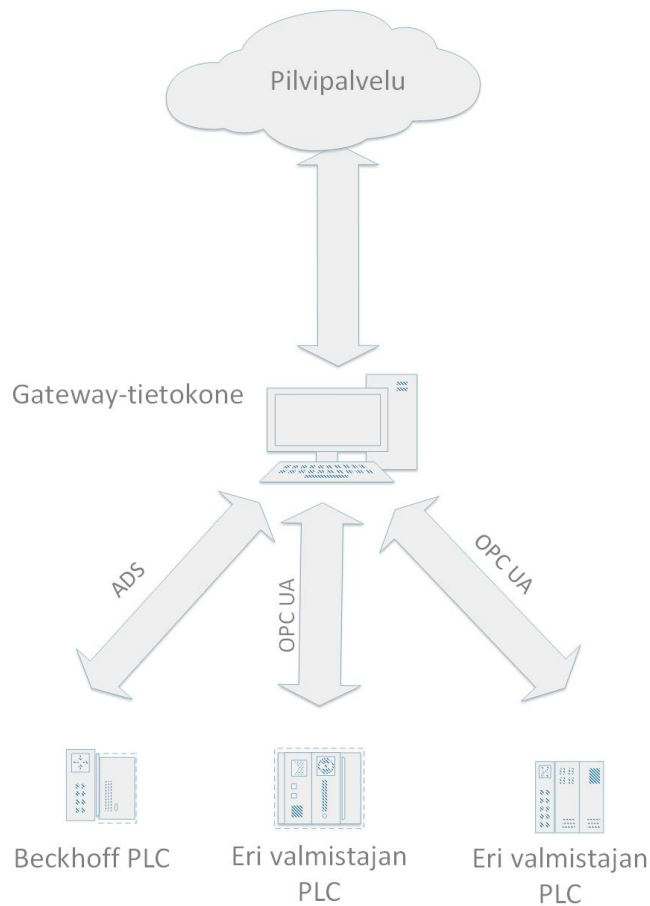
Koneoppiminen on tilastotiedettä muistuttava tekoälytieteen osa-alue, joka joskus liitetään moderniin kunnossapitoon. Koneoppimisen tavoite on saada kone oppimaan ympäristöstään ja tekemään itsenäisiä ratkaisuja. Koneeseen on ohjelmoitu algoritmeja, jotka tekevät mittaustietojen analysoinnin perusteella päätöksiä. (Collin & Saarelainen 2016, luku 12.)

3.3 Teollinen internet liiketoimintamallien muuttajana

Teollinen internet tuo lisäksi uusia liiketoimintamalleja. Yksi sellainen on PaaS (Product as a Service) eli liiketoimintamalli, jonka mukaan asiakas ostaa halutun tuloksen eikä tulosta tuottavaa laitetta. Tätä käsitettä ei sovi sekoittaa samanlyhenteiseen pilvipalvelumalliin Platform as a Service. Asiakas voi PaaS-mallia soveltaen ostaa esimerkiksi hitsausrobotin sijasta hitsaustapahtumia. Tämä tarkoittaa sitä, että asiakas maksaa hitsauslaitetoimittajalle jokaisesta hitsaustapahtumasta tietyn

summan eikä perinteiseen tyyliin osta itselleen hitsausrobottia. Mallissa vastuu suorituskyvystä siirtyy asiakkaalta laitetoimittajalle. On muitakin esimerkkejä valmiin tuotoksen ostamisesta laitteen sijasta: lentotunnit – lentokoneen moottori, tuotettu paineilma – kompressorin ja kuljetut matkat – juna. PaaS-liiketoimintamalli tarjoaa hyötyä liiketapahtuman molemmille osapuolille. Asiakkaan ei tarvitse tehdä yksittäistä isoa hankintaa, vaan hän voi jakaa kuluja pitkällekin ajanjaksolle. Yleensä PaaS-malli tarjoaa myös tuotteen kunnossapidon ja päivityksen. Näin asiakas säästyy vanhentuneen tai rikkiäisen laitteen vaivoilta. Palvelun tarjoaja eli laitteen valmistaja puolestaan saa epäsäännöllisten myyntivoittojen sijasta jatkuvaa ja tasaista kassavirtaa. Tämän lisäksi valmistajalle tulee tarkkaa palautetta tuotteen toiminnasta, mikä on varsin hyödyllistä tuotekehitykselle. Tällainen liiketoimintamalli tarkoittaa sitä, että sulavan työskentelyn varmistamiseksi laitteen valmistajan on oltava tietoinen laitteen kunnosta. Tämä onnistuu teollisen internetin mahdollistamalla etävalvonnalla. (Lombardo 2019.)

Puhuttaessa neljännessä teollisesta vallankumouksesta, Saksan valtion mainostamasta Industrie 4.0:sta eli jokaisen anturin tai toimilaitteen liittämistä verkkoon, voidaan ikääntyneessä tuotantolaitoksessa ajatella tämän muutoksen koskevan vain uusia tuotantolinjoja. Onhan uuden tuotantolinjan rakentaminen älykkäitä verkkoon yhdistettäviä antureita käyttäen varsin kallista eikä välttämättä niin kannattavaa. Tässä kohtaa esiin tulee termi retrofit, joka tarkoittaa usein vanhemman tuotantolaitoksen tuottavuuden parantamista uusilla laitteilla, esimerkiksi kosketusnäyttöpaneelilla (Austin-Morgan & Wilkins 2016). Perinteisellä ohjelmoitavalla logiikalla ohjattavat toimilaitteet ja anturit eivät ole vielä vanhentuneet teollisen internetin vallattua alaa, vaan ohjainlogiikan lähettämän datan saa pilvipalveluun analysoitavaksi käyttämällä esimerkiksi gateway-tietokonetta kuviossa 2 esitetyn mallin mukaisesti. Automaatioyritys Beckhoff esittää yhdeksi mahdollisuudeksi yhdistää logiikkaohjain teolliseen tietokoneeseen alustariippumattoman OPC UA -kommunikaatioprotokollan avulla. Tässä skenaariossa teollinen tietokone lähettää datan pilvipalveluun samalla kun logiikkaohjain keskittyy tuotantoprosessin ohjaamiseen. (Beckhoff 2018a, 9.)



Kuvio 2. Datan lähetyk gateway-tietokoneella (mukaillen Beckhoff 2018a, 10).

Datan keräämisestä on epäilemättä tuotantolaitoksille hyötyä, mutta konkreettisia esimerkkejä tarvitaan investointipäätöstä tehtäessä. Yksinkertaisessa esimerkissä käytetään sähkömoottoria ohjaavan kontaktorin "moottori käy" -signaalia, joka kytetään logiikkaohjaimelle. Kun tieto viedään tästä eteenpäin analytiikkasovellukselle, kytetään tilastoimaan moottorin käynti- ja odotusajat. Näin pystytään tunnistamaan "pullonkauloja" linjasta. Vastaavasti turhia käyntiaikoja tunnistamalla säästetään energiaa. Tuotantoprosessiin kulunutta aikaa voi myös verrata valmiiden tuotteiden määrään esimerkiksi kuukausittaisessa automaattisesti laadittavassa raportissa. Erilaisia antureita saa yhä edullisemmin valmistajien ja toimittajien lisääntyessä: laakerin lämpötilan- tai värinänmittaus on halpaa verrattuna tuotannon keskeytymisestä aiheutuviin kustannuksiin. Häiriötilastot ja häiriöalttiit laitteet voidaan asettaa näkyviin käyttöliittymässä.

3.4 Teollisen internetin kasvun haasteet

Edellisessä luvussa mainitulla PaaS-liiketoiminnalla on hyötyjen lisäksi omat haasteensa. Palvelun ydin on siinä, että vaikka laite on asiakkaan tiloissa, säilyy sen omistajuus silti valmistajalla. Tämä asettaa entistä suurempia paineita tuotekehitykselle nimenomaan tuotteen kestävyyttä ajatellen. (Lombardo 2019.)

Teolliseen internetiin keskittyneen yrityksen RUMBLEn perustaja Terri Foudray nostaa osaamispulan yhdeksi yleisimmistä esteistä teollisen internetin leviämiselle. Hän toteaa myös, että sisäisen kehitystyön sijasta yritysten tulisi hyödyntää enemmän yhteistyökumppaneiden tietotaitoa. Foudray korostaa tähän yhteyteen liittyvää tietoturvan ongelmallisuutta. Verkkoon liitettävien laitteiden lukumäärän kasvu lisää samalla hyökkäyskohteiden määrää. (Buntz 2019.)

Hyökkäyksiä voi kohdistua myös suoraan pilvipalveluihin. Palvelujen tarjoajat tiedostavat tämän huolettavan asiakkaita. Esimerkiksi Microsoft kertoo Azure-palvelullaan olevan monikerroksinen suoja. Hyökkäyksistä saadaan myös reaaliaikaista tietoa, ja hyökkäyksen loputtua palvelu laatii siitä yhteenvetoraportin (Microsoft, [viitattu 16.1.2020]). Myös Microsoftin kilpailija Amazon kertoo avoimesti AWS-palvelun tietoturvasta korostaen sen olevan varautunut palvelunestohyökkäyksiin usealla eri tavalla. Sellaisia keinoja ovat muun muassa DDos-hyökkäyksiin tarkoitettu AWS Shield, AWS WAF -palomuri sekä DNS-hyökkäyksiä ehkäisevä Amazon Route 53 (Amazon Web Services, 2018). Pilvipalveluihin kohdistuvia verkkohyökkäyksiä ovat muun muassa keinotekoisista ruuhkaa aiheuttava palvelunestohyökkäys DoS (Denial of Service), hajautettu eli usean lähteen aiheuttama palvelunestohyökkäys DDoS (Distributed Denial of Service) ja DNS-nimipalveluun (Domain Name Service) kohdistuva hyökkäys. DNS-hyökkäys kuvastaa usein välimuistin myrkyttämistä (engl. spoofing, cache poisoning), jonka takia verkkoselain ohjaa käyttäjän haitalliselle verkkosivulle. (CERT-EU 2014, 2; Kyberturvallisuuskeskus 2016, 2; Imperva, [viitattu 19.1.2020].)

4 TWINCAT-YMPÄRISTÖ

4.1 Beckhoff Automation

Saksassa 1980-luvulla alkunsa saanut Beckhoff Automation on automaatiolaitteiden ja -sovellusten toimittaja. Yrityksen liikevaihto lähenee jo miljardin euron rajaa. Suomessa Beckhoff Automation Oy:n toimipisteet sijaitsevat Hyvinkään pääkonttorin lisäksi Seinäjoella, Tampereella ja Oulussa. (Beckhoff, [viitattu 20.1.2020].)

Beckhoffin tuotteet jaetaan neljään osa-alueeseen:

- teolliset tietokoneet
- anturi- ja toimilaiteterminaalit
- liikkeenohjaus
- automaatio-sovellukset. (Beckhoff, [viitattu 20.1.2020].)

Uusin PC-pohjainen ohjausjärjestelmä TwinCAT esiintyy nimellä TwinCAT 3 XA (eXtended Automation). Sovelluskehitys ja testaus tapahtuvat ohjelmalla TwinCAT 3 XAE (eXtended Automation Engineering), johon sisältyy logiikkaohjaimissa toimiva XAR (eXtended Automation Runtime). TwinCAT 3 XAE on integroitu Microsoft Visual Studio -ympäristöön, ja se sisältää rajapintoja muun muassa C- ja C++-kielillä kirjoitettuihin ohjelmiin sekä MATLAB-laskentaohjelmistoon. Verkkoselaimella avautuva käyttöliittymä on HTML-pohjainen, mikä tekee siitä kohdelaitteesta riippumattoman. (Beckhoff, [viitattu 23.1.2020].)

TwinCAT 3 -ympäristö jaetaan komponenttiryhmiin, jollaisia ovat esimerkiksi liikkeenohjaus, käyttöliittymä ja konenäkö. TF6xxx Connectivity -ryhmä sisältää teollisen internetin työkaluja, kuten ohjelmalojkoja MQTT-kommunikointia varten, IoT Data Agent -rajapintasovelluksen sekä IoT Communicator App -mobiilisovelluksen. (Beckhoff, [viitattu 23.1.2020].)

4.2 ADS

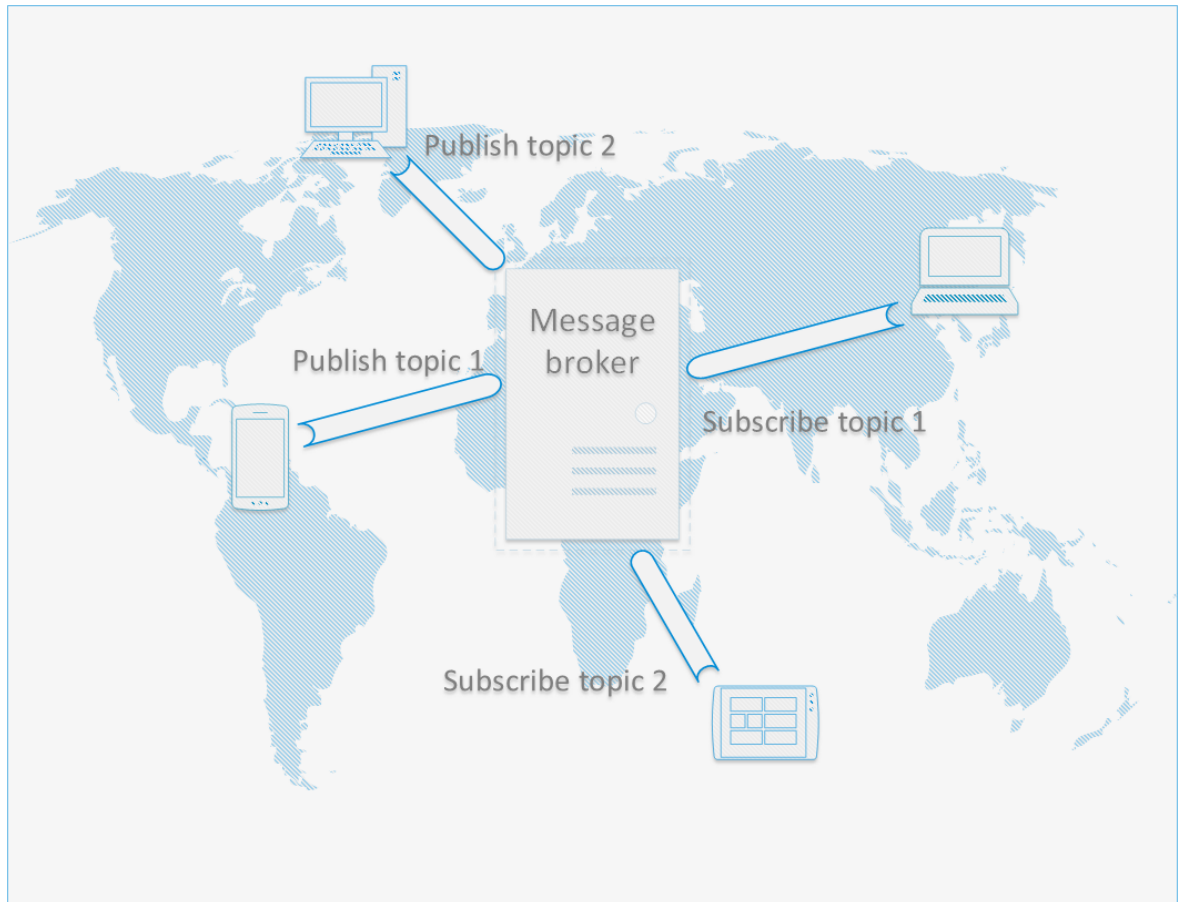
TwinCAT-ympäristössä eri sovellusmoduulien väliseen kommunikaatioon käytetään ADS-protokollaa (Automation Device Specification). Kommunikoitaessa eri laitteiden välillä ADS-protokollaa käytetään TCP/IP-protokollayhdistelmän kuljetuskerroksessa. ADS-kommunikaatioviesti koostuu viidestä osasta:

- AMS Net ID. Tämä on yhdistettävän laitteen osoite. Monissa Beckhoffin laitteissa osoite koostuu Ethernet-portin X001 MAC-osoitteen kahdeksasta viimeisestä merkistä, joiden perään on lisätty ".1.1".
- Port eli yhdistetyn laitteen portti.
- Index group, joka erottelee eri datalähetykset toisistaan saman portin sisällä.
- Index offset, joka kertoo, mistä kohdasta lukeminen tai kirjoittaminen aloitetaan viestissä.
- Len, joka kertoo datan pituuden.
- TCP port number eli TCP-yhteyden porttinumero, joka on ADS-protokollassa 48898. (Beckhoff, [viitattu 16.1.2020].)

4.3 MQTT

MQTT-protokolla (Message Queue Telemetry Transport) on yleistynyt IoT-ratkaisuissa. Suosio perustuu keveyteen ja käyttömahdollisuuteen tiedonsiirtoyhteyksissä, joissa kaistanleveys on rajallinen. Julkaisija-tilaajamalliin (publisher-subscriber) perustuva protokolla on omiaan välittämään dataa useilta kenttälaitteilta keskitettyyn pilvipalveluun. Protokollan keskeinen tekijä on message broker eli viestin välittäjä. Nimensä mukaisesti broker välittää viestin oikealle tilaajalle julkaisijan tietämättä tämän IP-osoitetta. Viestejä julkaisevat ja tilaavat asiakkaat (client) kuvion 3 mukaisesti. MQTT-viestin kaksi tärkeää osaa ovat payload ja topic. Payload (suom. hyötykuorma, tietosisältö) on viestin sisältö, kuten anturin mittaama arvo. Lähettäjällä ja vastaanottajalla on oltava tiedossa viestin tyyppi, joka voi olla esimer-

kiksi numeroarvo tai teksti. Jotta lähetetty viesti tavoittaa oikean vastaanottajan, tarvitaan viestiin identifioiva topic (suom. aihe). Suuresta datamäärästä voidaan ryhmitellä viestit jakamalla aiheet eritasoisiksi vinoviivoin eroteltuina, esimerkiksi: Tehdas/Pakkaamo/Liimauslaite/Lämpötila. (Beckhoff 2018a, 36; Beckhoff 2019a, 17-18.)



Kuvio 3. MQTT-kommunikointi (mukaillen Beckhoff 2019a, 18).

Viesti voidaan priorisoida kolmeen eri QoS-tasoon (Quality of Service):

- QoS 0. Viesti lähetetään välittämättä siitä, meneekö viesti perille.
- QoS 1. Viesti lähetetään ja varmistetaan, että se vastaanotetaan vähintään kerran. Jos vastaanottaja ei kuittaa viestiä määräajan kuluessa, se lähetetään uudestaan.
- QoS 2. Viesti lähetetään ja varmistetaan, että se vastaanotetaan vain kerran. Tämä tehdään käyttämällä kaksinkertaiseksi kättelyksi (handshake mechanism) nimitettyä toimintoa. (Beckhoff 2019a, 19-20.)

MQTT-viestin salaukseen voi käyttää esimerkiksi TLS-salausprotokollan (Transport Layer Security) kahta mekanismia:

- TLS-PSK (Pre-Shared Key), joka tarkoittaa viestin salaamista ja salauksen purkamista ennalta sovitun salasanan avulla.
- TLS-CA (Certificate Authority) eli kommunikoinnin sallimista sertifikaattitiedoston omaavalle laitteelle. (Beckhoff 2019a, 19-20.)

Beckhoff Automation Oy:n ohjausjärjestelmäasiantuntija Aapo Vuoristo (2020) antoi esimerkin, jossa MQTT-yhteyttä voi käyttää logiikkaohjaimessa suoritettavan PLC-ohjelman päivitykseen. Sama toiminto onnistuu myös esimerkiksi REST-arkkitehtuurimallia käyttäen. Päivitystä kaipaava ohjainlaite voi hakea päivitystiedoston pilvipalvelusta erillisellä sovelluksella. Sovellus vertaa pyydettyä tai syklisesti PLC-ohjelman ja pilvipalvelussa sijaitsevan ohjelman versioita. Mikäli PLC-ohjelma on päivityssovelluksen ilmoituksen mukaan vanhentunut, logiikkaohjain asettaa itsensä config-tilaan (konfiguraatio), varmuuskopioi nykyisen ohjelman, lataa uudemman version PLC-projektin boot-kansioon ja asettaa itsensä takaisin run-tilaan (ohjelman suoritus).

4.4 REST

REST (Representational State Transfer) on suosittu teknologia verkkosovellusten ja niiden rajapintojen ohjelmoinnissa. Tällä mallilla toteutettua rajapintaa (API, Application Programming Interface) kutsutaan nimellä RESTful API. Yksi REST-teknologiassa käytettävä tiedonsiirtoprotokolla on HTTP, jonka avulla voidaan siirtää muun muassa keveytensä vuoksi suosittua JSON-formaatin (JavaScript Object Notation) dataa. (Abeyasinghe 2008, 7-8, 11, 16.)

HTTP-protokolla sisältää tiedonsiirtoon käytettäviä request-metodeja (suom. pyyntö). Näitä ovat esimerkiksi get ja post. Get-metodia käyttämällä voidaan hakea dataa valitusta lähteestä. Post-metodin avulla voidaan luoda tai päivittää muuttujan arvo halutussa lähteessä. (Abeyasinghe 2008, 11-13.) Tässä opinnäytetyössä REST-teknologiaa käytettiin C#-ohjelmointikielellä toteutetussa sovelluksessa, joka yhdisti TwinCAT PLC -ohjelman Wapicen IoT-TICKET-palveluun.

5 IOT-ALUSTOJEN KÄYTTÖKOKEMUKSIA

5.1 Microsoft Azure

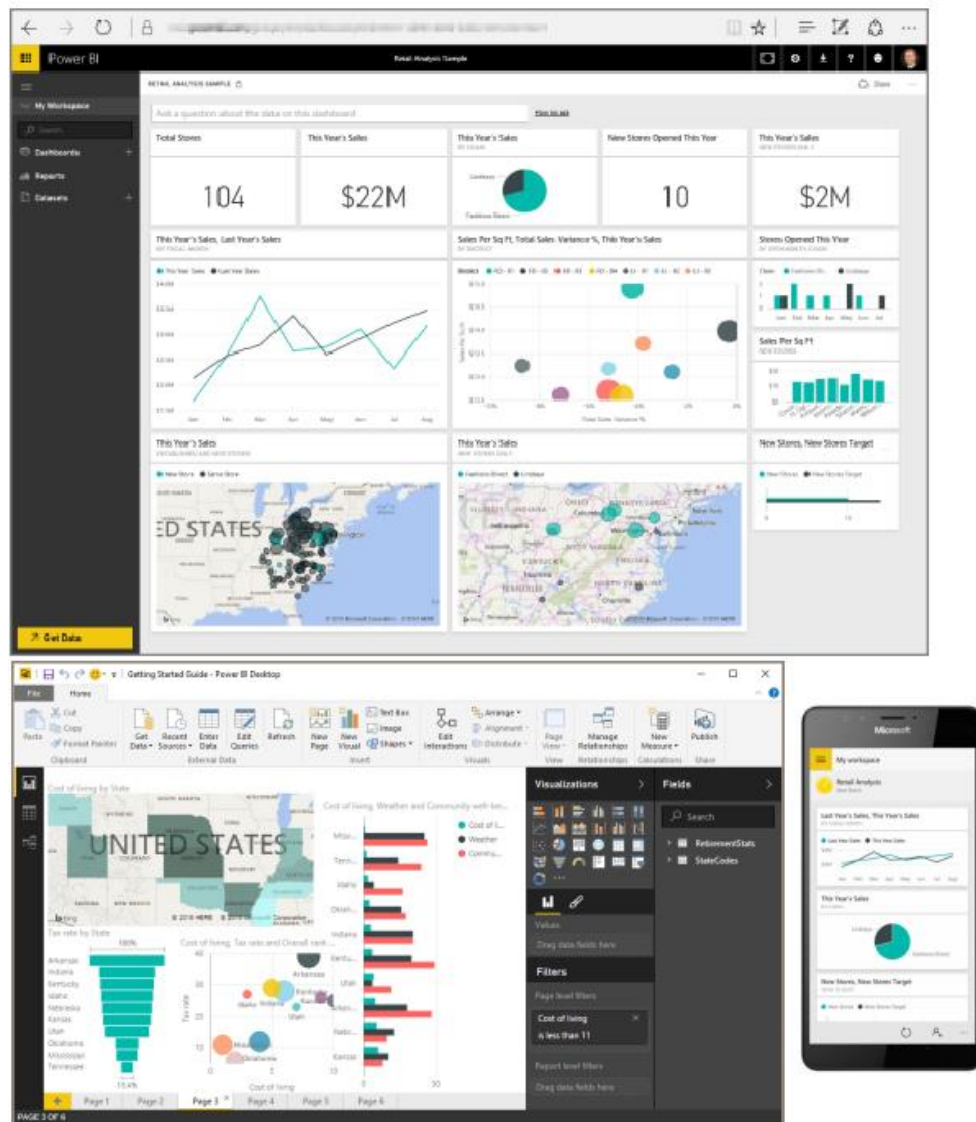
Microsoft antaa käyttäjilleen mahdollisuuden tutustua ilmaiseksi pilvipalvelu Azureen sekä osaan siihen liittyvistä sovelluksista. Tilin luominen ja palveluun tutustuminen onkin tehty helpoksi. Kirjautumisen jälkeen palvelu esittelee lyhyesti tärkeimmät ominaisuutensa. Palveluun on saatavissa runsaasti englannin- ja suomenkielisiä dokumentteja sekä verkkokursseja osoitteesta <https://docs.microsoft.com/en-us/azure/>. Azuren käyttöönottossa ei ole välttämättä suurta hyötyä muiden, kevyempien pilvipalveluiden käyttökokemuksista. Microsoft Azure on erittäin laaja, isoihin järjestelmiin tarkoitettu palvelu. Se sisältää lukuisia eri sovelluksia ja työkaluja, joihin tutustuminen vaatii aikaa ja kärsivällisyyttä. Palvelu sopii siis hyvin isollekin organisaatiolle.

Sovellusvalikoimaan kuuluu esineiden internetin käyttötarkoituksiin suunnattu IoT Hub. Palvelun ilmaiskokeiluun sisältyvä IoT Hub Free Edition tarjoaa yhteensä 8000 viestin lähettämistä tai vastaanottamista päivässä ja 500 laitteen rekisteröitymistä. Käyttäjä voi luoda IoT Hubiin yksiköjä (unit), joiden lukumäärä vaikuttaa osaltaan palvelun kokonaishintaan. Yksiköjä on eritasoisia ja -hintaisia, alkaen alle 10 eurosta. Halvin yksikkö pystyy käsittelemään 400 000 viestiä päivässä – kallein jopa 300 miljoonaa. Hubiin voidaan luoda myös resursseja (resource), jotka on jaettavissa eri ryhmiin. Resurssille voidaan lisätä laitteita. Laitteen tunnus (device ID) ja primääriavain (primary key) ovat fyysisen laitteen palveluun yhdistämisen kannalta tarvittavia tietoja. (Microsoft, [viitattu 17.1.2020].)

Stream Analytics on datan käsittelyyn suunniteltu sovellus. Sovellukseen täytyy määrittää tulot ja lähdöt eli datan vastaanotto- ja lähetyssijainnit. Yksi mahdollisista lähdöistä on Power BI. Lisäksi dataa voidaan käsitellä SQL-kyselyllä.

Power BI -sovellukseen voi lisätä erilaisia mittareita, käyriä ja muita tapoja havainnollistaa dataa. Power BI on saatavissa mobiilisovelluksena, johon Microsoft lupaa samat ominaisuudet kuin verkkoselain- tai työpöytäsovellukseen (kuvio 4). Sovellus mahdollistaa muun muassa erilaiset ilmoitukset, kuten tekstiviestit, sähköpostit sekä

push-ilmoitukset. Käyttäjä voi määrittää ilmoituksen lähetettäväksi esimerkiksi sovitun raja-arvon ylityksestä tai alituksesta. Erilaiset data-analytiikkasovellukset, datan tuonti xlsx-muodossa taulukkolaskentaohjelmiin sekä kokoavat raportit ovat käteviä työkaluja. (Microsoft Power BI, 26.6.2018).



Kuvio 4. Microsoft Power BI verkkoselain-, työpöytä- tai mobiilisovelluksena.

Power BI -sovelluksen hinnoittelupolitiikka on joustavaa. Ilmaisversio sisältää paljon ominaisuuksia, mutta esimerkiksi datan jakaminen kollegan kanssa onnistuu vain Pro-versiolla. Embedded-version virtuaalitietokoneisiin voi valita halutun määrän ytimiä ja muistia. Tehokkaimmassa virtuaalikoneessa on jopa 32 virtuaaliydintä ja 100 gigatavun kokoinen välimuisti (Microsoft, [viitattu 3.4.2020]).

TwinCAT-laitteen yhdistäminen palveluun voidaan toteuttaa esimerkiksi luvussa 6.5 esiteltävällä `lotMqttSampleUsingQueue`-ohjelmalla. Tämä vaatii toimiakseen parametreina CA-sertifikaattitiedoston polun ja SAS-merkkijonon (kuvio 5). CA (Certification Authority) on luotetun kolmannen osapuolen sertifikaatti. SAS (Shared Access Signature) on kaksiosainen merkkijono, joka sisältää käytettävän resurssin ja siihen pääsyn sallivat merkit.

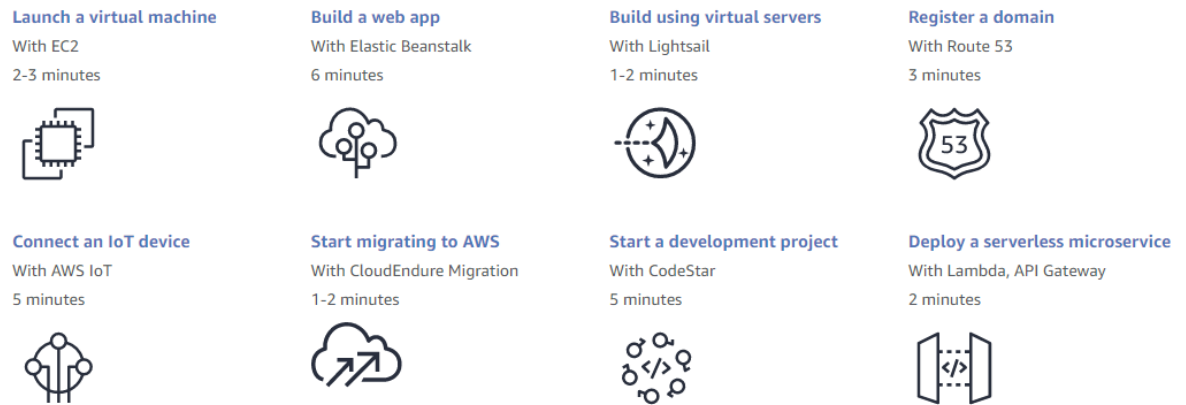
```
// CA-sertifikaatti
fbMqttClient.stTLS.sCA := 'c:\TwinCAT\3.1\Config\Certificates\MyCA.crt';
// SAS token
fbMqttClient.stTLS.sAzureSas := 'insertSasTokenHere';
```

Kuvio 5. PLC-ohjelman parametrit Azure-palveluun yhdistettäessä.

Yhteenvetona palvelusta todettakoon, että varsin monipuolinen ja jatkuvasti kehittyvä Azure on hyödyllinen data-analytiikkatyökalu vaikkapa suurelle kauppa-alan konsernille. Laajuutensa vuoksi käyttöönottoon on syytä varata aikaa. Tämän lisäksi palvelun kokonaishintaa voi olla erillisten sovellusten vuoksi vaikea määrittää etukäteen. Vaatimattomampaan datan keräämiseen ja analysointiin on tarjolla helpommin käyttöön otettavia palveluita.

5.2 Amazon AWS IoT

Amazon tarjoaa monipuolista, lukemattomiin käyttökohteisiin soveltuvaa ja koekäyttöön asti ilmaista Amazon Web Services IoT -pilvipalvelua (Amazon, [viitattu 3.4.2020]). Teollisen internetin sovelluksia varten palvelun tärkeitä osia ovat laitteet pilvipalveluun yhdistävä IoT Core, tietoturvapalvelu IoT Device Defender, laitehallintatyökalu IoT Device Management ja data-analytiikkatyökalut. Sovellusten käyttöohjeet ovat saatavissa palvelun aloitusnäköymän kautta (kuvio 6).



Kuvio 6. Amazon AWS-palvelun sovelluksia.

Aluksi palvelua käytettäessä suunnattiin IoT Core -osaan, jossa luodaan yhdistettävää laitetta edustava olio eli thing. Nämä oliot voidaan jakaa ryhmiin ja samalla lisätä laiteryhmälle ominaisia attribuutteja. Laitteen yhdistäminen verkon yli palveluun vaatii kaksi asiaa: käytännön (policy) sekä digitaalisen sertifikaatin. Käytännöt ovat lupia, joilla resursseille, kuten laitteille tai aihepiireille (topic), annetaan oikeudet muun muassa palveluun yhdistämiseen tai viestien julkaisemiseen. Luvat kirjoitetaan Create a policy -osioon yksinkertaisesti muotoon `iot:Connect` tai `iot:Publish`. Yhdellä laitteella voi olla useita lupia. Laitetta luotaessa palvelu tarjoaa ladattavaksi kolme X.509-kryptografiastandardin digitaalista sertifikaattitiedostoa: varsinaisen sertifikaattitiedoston, avaintiedoston ja Microsoft Azuren tavoin CA-sertifikaattitiedoston. Tiedostot on tallennettava dataa lähettävälle tietokoneelle, ja niiden tiedostopolku on annettava parametrina lähettävälle ohjelmalle. Ohjelma tarvitsee parametrina myös käyttäjäkohtaisen osoitteen, joka löytyy IoT Core -palvelun kohdasta Custom endpoint.

Beckhoff tarjoaa sivuiltaan ladattavaksi esimerkksiovelluksen nimeltä `lotMqttSampleAwsIoT`, jolla TwinCAT-ohjelma voidaan yhdistää AWS IoT -palveluun. Yhdistäminen voidaan toteuttaa myös luvussa 6.4 esiteltävällä IoT Data Agent -rajapintasovelluksella, jolle parametrit annetaan kuvion 7 mukaisesti.

The screenshot shows the 'Properties' dialog box for the IoT Data Agent, configured for AWS IoT. It is divided into four sections: General, Generic, Specific, and Authentication.

Key	Value
Activated	☒
ID	5
Name	AWS IoT

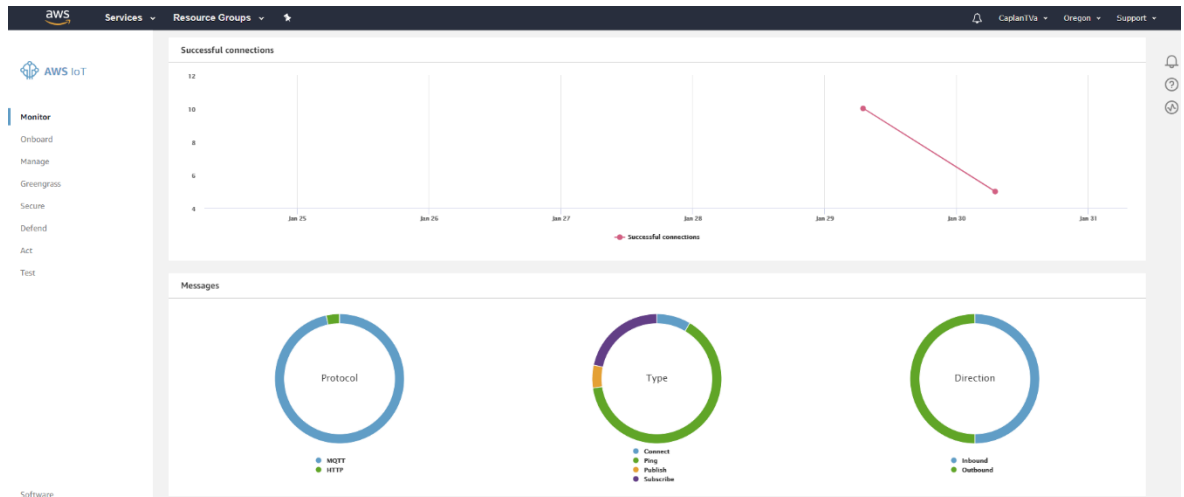
Key	Value
ClientId	CaplanTVa
Port	8883
Protocol	Aws

Key	Value
Url	https://[redacted].amazonaws.com

Key	Value
CAFile	C:\TwinCAT\3.1\Config\Certificates\root.pem
CertFile	C:\TwinCAT\3.1\Config\Certificates\[redacted].certificate.pem.crt
KeyFile	C:\TwinCAT\3.1\Config\Certificates\[redacted].private.pem.key
KeyFilePassword	
PwdInfold	6
TlsType	Client cert

Kuvio 7. IoT Data Agent -sovelluksen parametrit AWS-palveluun yhdistettäessä.

Data-analytiikka ja muut AWS IoT -palvelun sovellukset jätettiin niiden laajuuden vuoksi tässä työssä testaamatta. Laaja ja selkeä dokumentaatio auttaa käyttäjää rakentamaan halutun kaltaisen, Microsoft Azuren tavoin varsin monipuolisen IoT-palvelun. Myös AWS-palvelun laskutus tapahtuu pay-as-you-go-maksupolitiikalla, ja käyttäjän kulloinenkin kuukausimaksu on nähtävissä palvelusta (Amazon, [viitattu 4.4.2020]). AWS IoT Monitor antaa yleisnäkymän palveluun yhdistetyistä laitteista ja tiedonsiirtotavoista (kuvio 8).



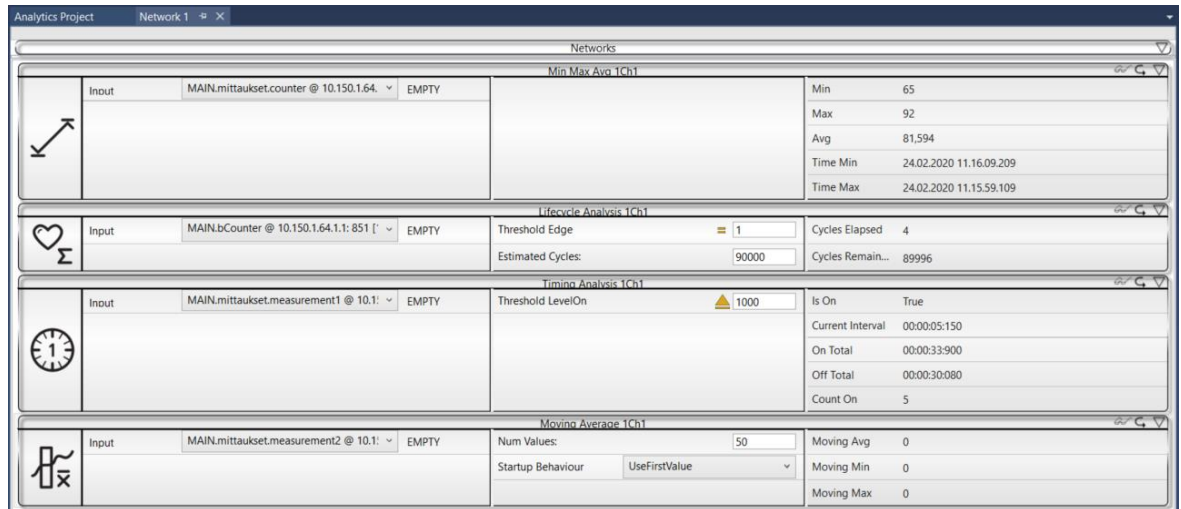
Kuvio 8. Amazon AWS IoT Monitor.

5.3 TwinCAT Analytics

Vuonna 2015 automaatiotuotteiden valmistaja Beckhoff toi markkinoille TwinCAT Analytics -tuoteperheen. Tuotteilla onnistuvat datan kerääminen, varastointi ja analysointi (Beckhoff, [viitattu 3.4.2020]). TwinCAT 3 -version käyttöliittymä (HMI) on HTML-pohjainen, joten se on käytettävissä HTML5-kielimäärittelyä tukevilla verkkoselaimilla. Tämä tarkoittaa myös sitä, että käyttöliittymään on mahdollista lisätä visualisointeja ja funktioita käyttämällä TwinCAT-ohjelman omien työkalujen lisäksi JavaScript-ohjelmointikieltä. Käyttöliittymä voidaan luoda sijoittamalla sivulle pienoishjelmia, esimerkiksi mittareita, käyriä ja taulukoita. Vaikka TwinCAT Analytics sisältääkin IoT-alustalle ominaisia työkaluja, se ei suinkaan ole valmis pilvipalvelu tai pilvialusta. Lisenssin omistavan yrityksen palvelimelle on mahdollista luoda verkon yli toimiva Analytics-sovellus, joka pilvipalvelun tavoin lukee ja analysoi dataa tietokannasta ja visualisoi sen verkkoselainpohjaisessa käyttöliittymässä.

Analytics-tuoteperhe sisältää erillisiä työkaluja. Analytics Logger -työkalu kerää dataa reaaliajassa logiikkaohjaimelta. Storage Provider tarjoaa rajapinnan tietokantaan ilman SQL-kyselykomentojen osaamista. Analysointi tapahtuu Workbench-tuotteella, joka luo valituista matemaattisista funktioista PLC-projektin ja käyttöliittymän. PLC-projekti sisältää automaattisesti generoidut ohjelmalohkot, jotka vastaavat käyttäjän valitsemia funktioita. Workbench-tuotteen One-Click Dashboard -omi-

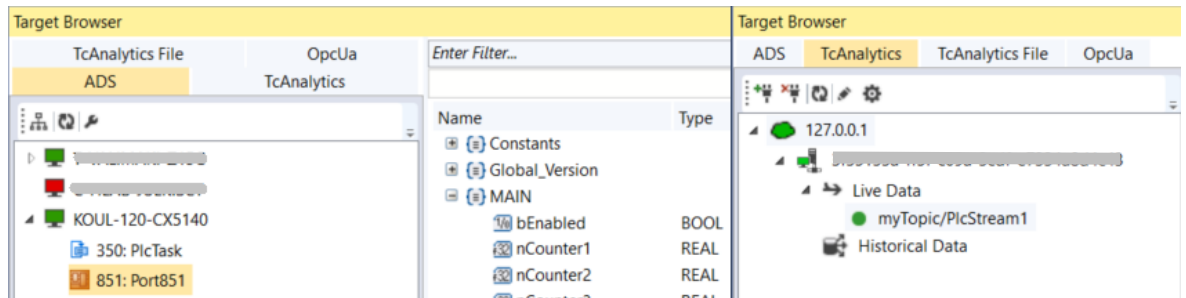
naisuus luo käyttöliittymälle perusrakenteen, joka sisältää vähintään yhden visuaalisen objektin jokaista analytiikkafunktiota varten. Perusrakenteeseen voidaan määritellä esimerkiksi yrityksen logo, värit ja käyttökieli. (Beckhoff, [viitattu 3.4.2020].)



Kuvio 9. TwinCAT Analytics Networks.

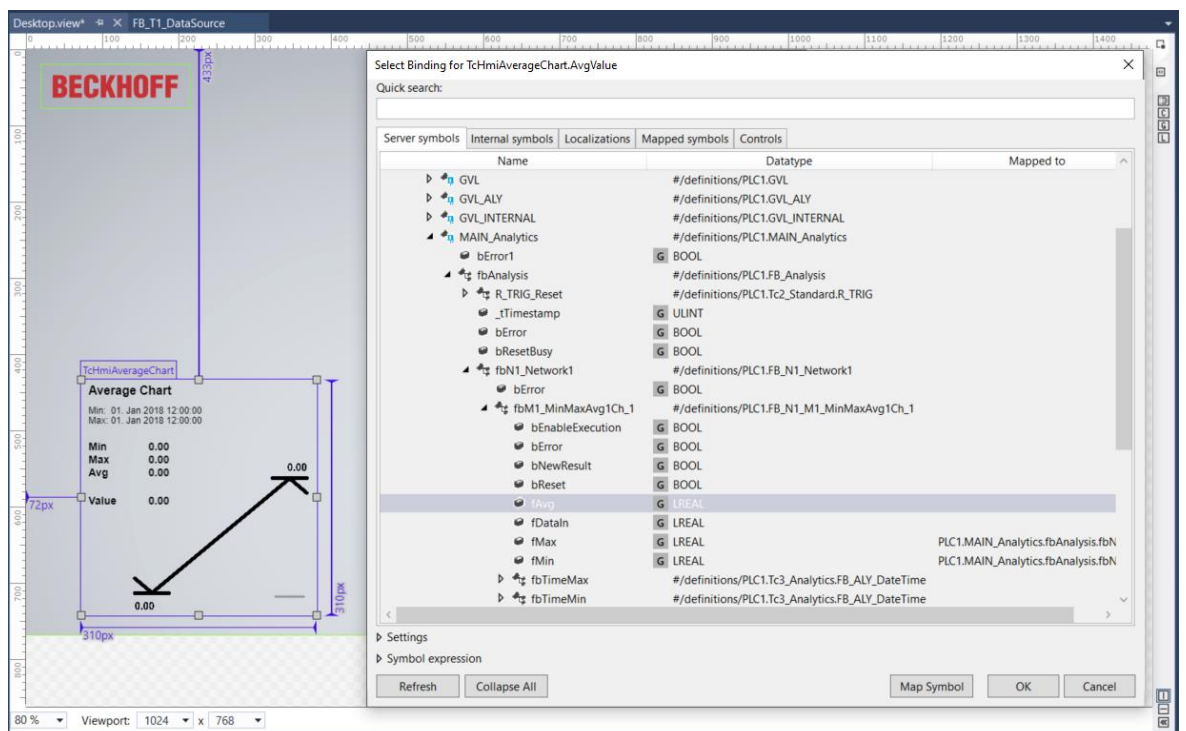
Kuviossa 9 esitellään neljä funktiota. Min Max Avg- ja Moving Average -funktioilla lasketaan annetuista muuttujista minimi-, maksimi- ja keskiarvot. Lifecycle Analysis -funktioilla voidaan käyttöliittymässä esittää esimerkiksi työkalun jäljellä oleva käyttöaika. Timing Analysis kertoo, kuinka kauan laite on ollut käynnissä tai sammutettuna.

Analytics-sovellus voi kerätä dataa ADS- tai OPC UA -yhteyksiä käyttämällä. Tieto voi tulla myös Analytics Logger -työkalun keräämänä joko binääritiedostosta tai reaaliaikaisesti MQTT-yhteyden avulla. Tiedon keräämistä kokeiltiin tässä opinnäytetyössä ADS-yhteydellä ja MQTT-viesteillä. ADS-yhteyden avulla muuttujat näkyvät Target Browser -ikkunassa (kuvio 10), kun valittu reitti ja portti ovat valittuina. Saman ikkunan TcAnalytics-välilehdelle on mahdollista luoda yhteys MQTT-viestinvälittäjään syöttämällä konfiguraatioikkunaan sen IP-osoite ja portti. Suojattua yhteyttä käytettäessä ikkunaan voidaan antaa myös käyttäjänimi, salasana ja sertifikaatteja.



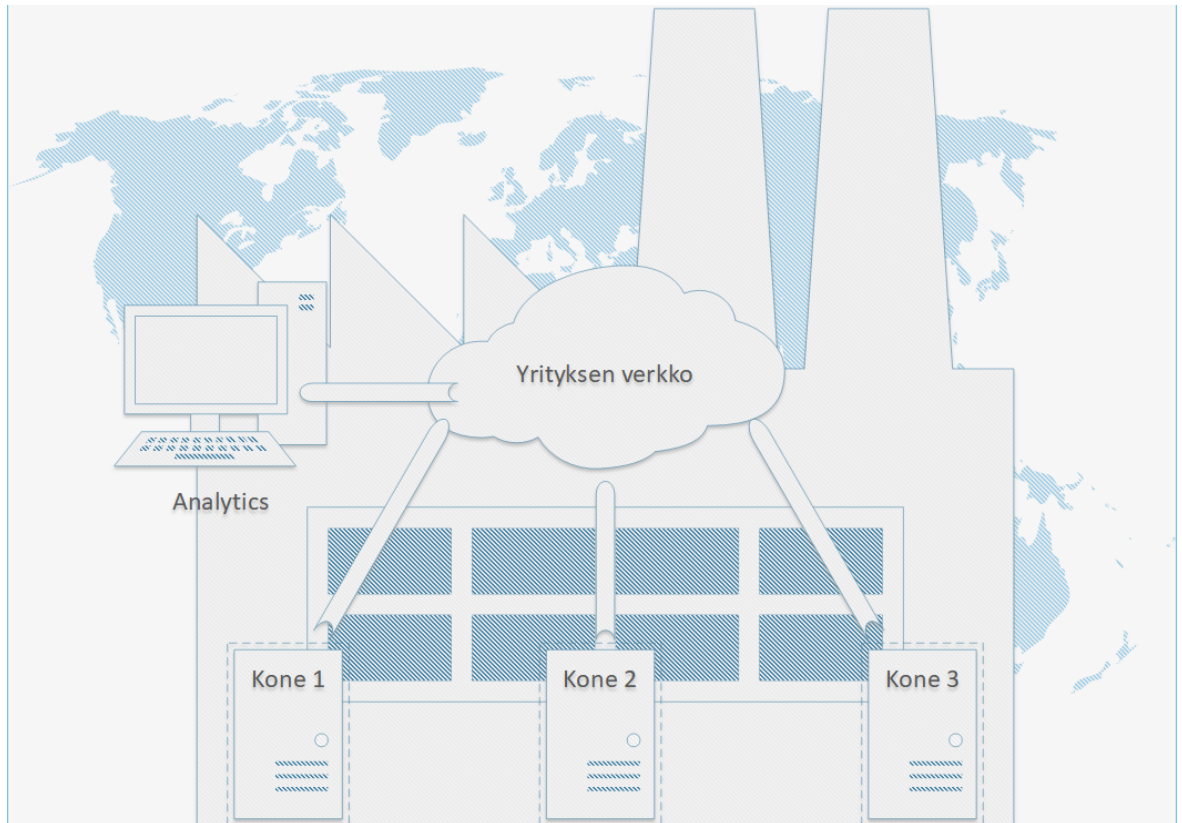
Kuvio 10. Muuttujat TwinCAT Analytics -ohjelmassa.

Kun valmis Analytics-projekti otetaan käyttöön (Deploy runtime), ohjelma luo automaattisesti TwinCAT PLC -projektin. Siinä Analytics-kirjaston funktioita ja ohjelmalohkoja käytetään IEC 61131-3 -standardin ST-ohjelmointikielellä (Structured Text). Käyttöliittymä koostuu käyttäjän valitsemista funktioista: edellä mainitut keskiarvot tai käyntiajat visualisoidaan esimerkiksi erilaisilla mittareilla tai trendikäyrillä. Automaattisesti luodun perusnäkökuvan lisäksi laskettuja tuloksia voi liittää käyttöliittymän visualisointeihin kuviossa 11 esitetyllä tavalla. Vaikka joillekin funktioille on olemassa niille erikseen tarkoitettuja visualisointiobjekteja, voidaan tuloksia esittää vapaasti missä muodossa tahansa. TwinCAT HMI -sovelluksessa on tarjolla laaja valikoima erilaisia työkaluja käyttöliittymän luontiin. Lisättyjen objektien värimaailma ja muut ominaisuudet ovat vapaasti muokattavissa.



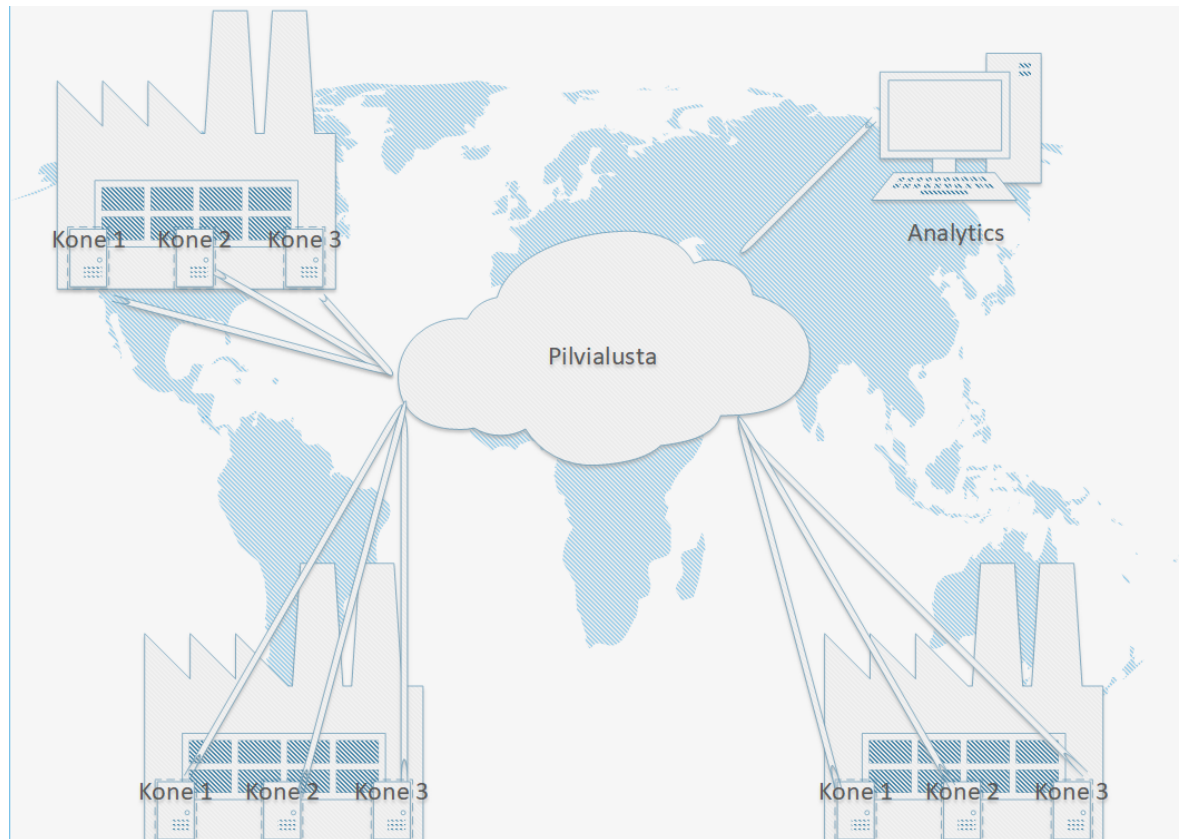
Kuvio 11. Muuttujien yhdistäminen TwinCAT HMI -käyttöliittymään.

Luotu käyttöliittymä voidaan siirtää yrityksen palvelimelle, jolloin se on henkilöstön käytettävissä yrityksen sisäisessä verkossa eli yksityisessä pilvessä (kuvio 12). Analysoinnin tulokset saadaan visualisoinnin lisäksi myös kirjoitettua tiedostoon tai lähetettyä MQTT-viestinvälittäjälle.



Kuvio 12. TwinCAT Analytics yrityksen sisäisessä verkossa (mukaillen Beckhoff 2018b).

Käyttöliittymään voi olla myös tarpeen päästä yrityksen sisäisen verkon ulkopuolelta. Palvelimelle voidaan luoda etäyhteys, jonka avulla verkkoselainpohjainen käyttöliittymä on käytettävissä (kuvio 13). Vaihtoehtoisesti Analytics-sovellus voidaan siirtää jonkin pilvipalvelun tarjoajan virtuaalikoneeseen.



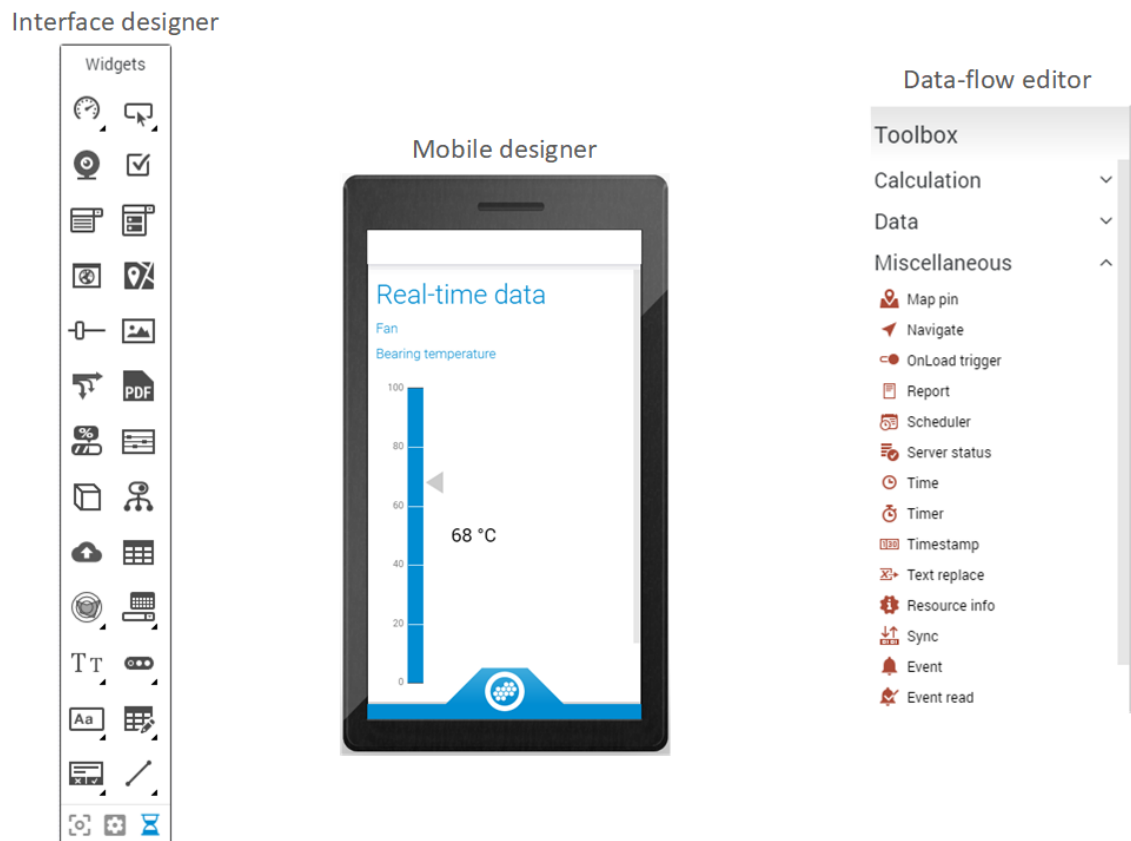
Kuvio 13. TwinCAT Analytics käytössä verkon yli (mukaillen Beckhoff 2018b).

TwinCAT HMI -projektiin eli käyttöliittymään on mahdollista luoda käyttäjäryhmiä, joille myönnetään eritasoisia oikeuksia. Käyttöliittymän voi näin rajata eri ryhmiin kuuluville käyttäjille. Esimerkiksi yrityksen myyntihenkilöstö näkee vain myyntitastistiikkaa ja kunnossapidon työntekijät pääsevät tutkimaan vain laitteiden reaaliaikaista dataa ja vikatilastoja. Analytics-sovelluksella rakennettu IoT-ratkaisu sopii siis isollekin organisaatiolle. Analytics Service Tool -työkalu on suunnattu laitteen tai tuotantolinjan käyttöönottoa ja yleisesti kunnossapitohenkilöstöä varten. Workbench-lisenssiin verrattuna huokeampi Service Tool sisältää samat funktiot ilman automaattista PLC-projektin luomista.

Oman IoT-ratkaisun luominen TwinCAT-ympäristöä käyttäen on mahdollista, mutta kokonainen sovellus tietokantayhteyksineen jätettiin tässä työssä luomatta. OPC UA -yhteyden avulla Analytics-sovelluksen yhdistäminen logiikkaohjaimeen ei rajoitu pelkästään Beckhoffin laitteisiin. Uudehkolla TwinCAT 3 HMI -työkalulla on mahdollista luoda varsin monimutkaisia ja visuaalisesti näyttäviä käyttöliittymiä.

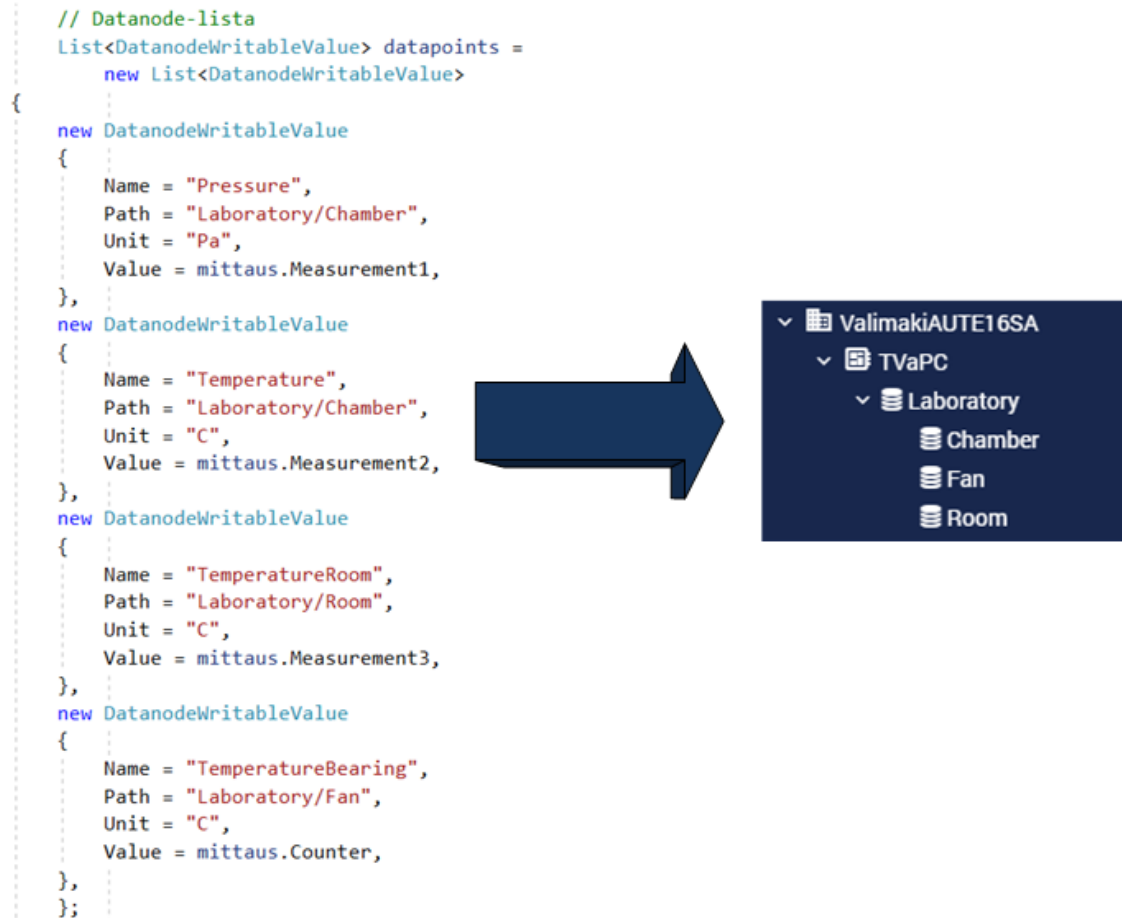
5.4 Wapice IoT-TICKET

Vuonna 1999 perustettu kotimainen ohjelmistoalan yritys Wapice tarjoaa teollisen internetin ratkaisuihin IoT-TICKET-alustaansa. Microsoft palkitsi Wapicen vuonna 2019 Partner of the Year -kilpailussa sovellusinnovaation kategoriassa. IoT-TICKET-alustan kehitys alkoi vuonna 2005, ja sillä on nykyään yli 1,6 miljoonaa käyttäjää. (Wapice 2019a, 1; Wapice 2019b.)



Kuvio 14. IoT-TICKET-palvelun työkaluja.

Palvelussa on muille pilvipalveluille tunnusomaisia työkaluja (kuvio 14), kuten lopputähtäjän käyttöliittymä eli Dashboard, lohko-ohjelmointityökalu Dataflow Editor, responsiivisesti skaalautuvan käyttöliittymän rakennustyökalu Mobile Designer sekä raporttien suunnitteluun soveltuva Report Editor. Palvelussa voidaan käsitellä big dataa sekä ohjelmoida hälytys- ja raportointimekanismeja. Yrityksen tuotteisiin kuuluu myös teollisuusolosuhteissa toimiva etähallintalaite WRM 247+, jolla tiedon kerääminen ja lähettäminen onnistuvat erilaisten liitännöiden avulla kätevästi. (Wapice, [viitattu 12.3.2020].)



Kuvio 15. Datanode-struktuurit IoT-TICKET-palvelun hierarkiassa.

Palveluun yhdistetyt laitteet näkyvät palvelun hierarkiassa (kuvio 15), jonka kolme porrasta ovat seuraavat:

- Enterprise – asiakkaan organisaatio, kuten tehdas tai muu yritys
- Device – kytkettävä laite, joita organisaatiolla voi olla useita
- Datanode – datastrukturi, joka voi yhden tai useamman arvon lisäksi sisältää muita parametreja, kuten nimen, yksikön ja palvelussa datan yksilöivän polun.

Käyttöliittymät voidaan jakaa eri hierarkiatasoihin. Esimerkiksi yrityksen johdolle on saatavissa kokonaistilannetta esittävä näkymä. Yrityksen tuotantolaitosten sijainnit

näkyvät kartassa kuvakkeina, joita painamalla laitteiden reaaliaikainen tila tulee näkyviin. Tehdasnäkymästä voidaan seurata laitteiden yksittäisten anturien mittaustietoa. (Wapice 7.4.2019.)

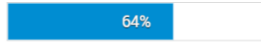
Wapice tarjoaa useita vaihtoehtoja tiedonsiirtotavoille: OPC UA- ja MQTT-protokollien lisäksi palvelu tukee yli 140:tä tiedonsiirtoprotokollaa. Yrityksen verkkosivuilta osoitteesta <https://www.iot-ticket.com/apps> on ladattavissa valmiita ohjelmakirjastoja, joiden avulla lähes mikä tahansa laite on liitettävissä palveluun. IoT-TICKET ei aseta rajoja käyttötarkoituksille: palveluun on tuotavissa tuotantolaitoksen anturidatan lisäksi vaikkapa säätietoa, laitteen huoltokirjauksia tai tuotteen käyttäjäpalautetta. Tietoa voidaan lähettää palvelusta eteenpäin kolmannen osapuolen järjestelmiin, kuten tuotannonohjausjärjestelmään, analytiikkaohjelmiston koneoppimismalliin tai raakadataa vastaanottavaan Data Lake -tyyppiseen tietojen tallennuspaikkaan. Tällainen avoimuus poistaa asiakkaan kannalta hankalan skenaarion, jossa dataa voidaan käsitellä vain yhdessä järjestelmässä. (Wapice, [viitattu 19.1.2020]; Wapice, [viitattu 12.3.2020]; Wapice 7.4.2019.)

Palvelussa käyttöliittymä visualisoidaan Interface Designer -työkalun avulla, jossa on mittarien ja kuvaajien lisäksi runsaasti toimintoja muun muassa tiedostonhallintaan varten. Käyttöliittymä rakennetaan pienoisohjelmia (widget) käyttäen, ja valmiit näkymät voidaan tarvittaessa jakaa eri välilehdille. Käyttöliittymäkomponentit ovat muun muassa painikkeita, erilaisia mittareita ja kuvaajia, taulukkonäkymiä ja listoja. Kuvien ja kolmiulotteisten laitemallien avulla käyttöliittymä voidaan visualisoida näyttäväksi.

Päivän data

Luo raportti

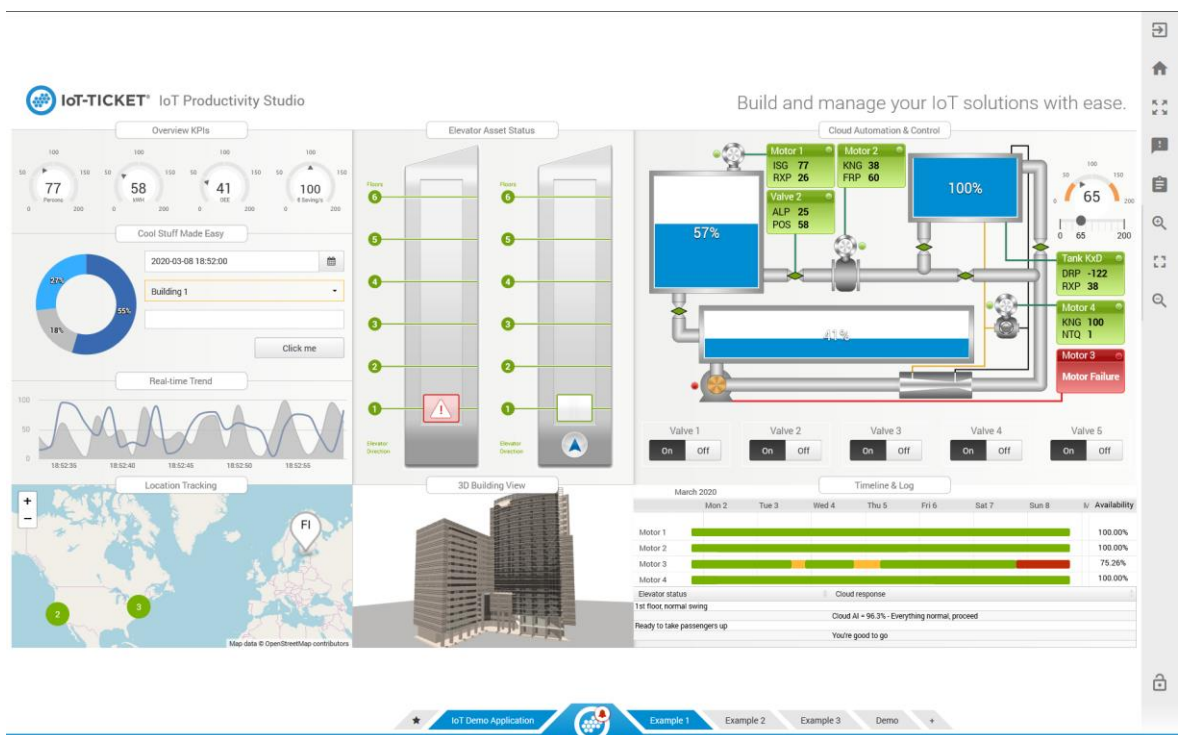
Vesitankki



MyDashboard Main page Live data Historical data

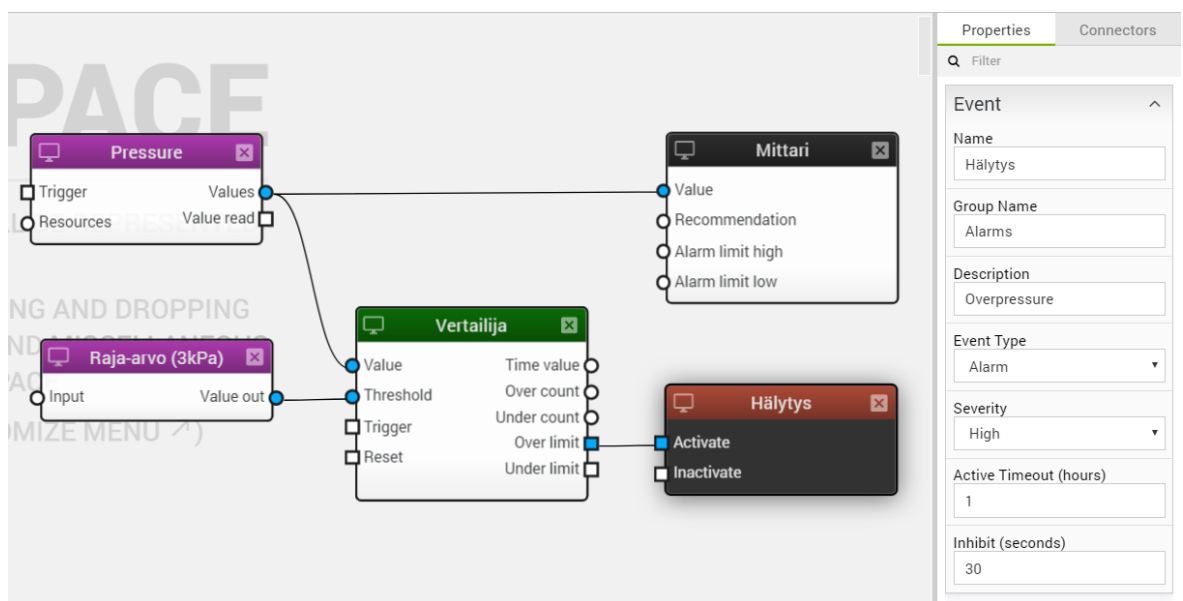
Kuvio 16. Rakennettu käyttöliittymä IoT-TICKET-palvelussa.

Monimutkaisemmat käyttöliittymät jätettiin tässä työssä rakentamatta, mutta kolmen dataa visuaalisesti esittävän välilehden luonti onnistui jo lyhyessä ajassa (kuvio 16). Asiakas voi luoda myös melko kehittyneitä sivuja kolmiulotteisesti mallinnettuine näkymineen (kuvio 17).



Kuvio 17. IoT-TICKET-palvelun esimerkinäkymä (Wapice 7.4.2019).

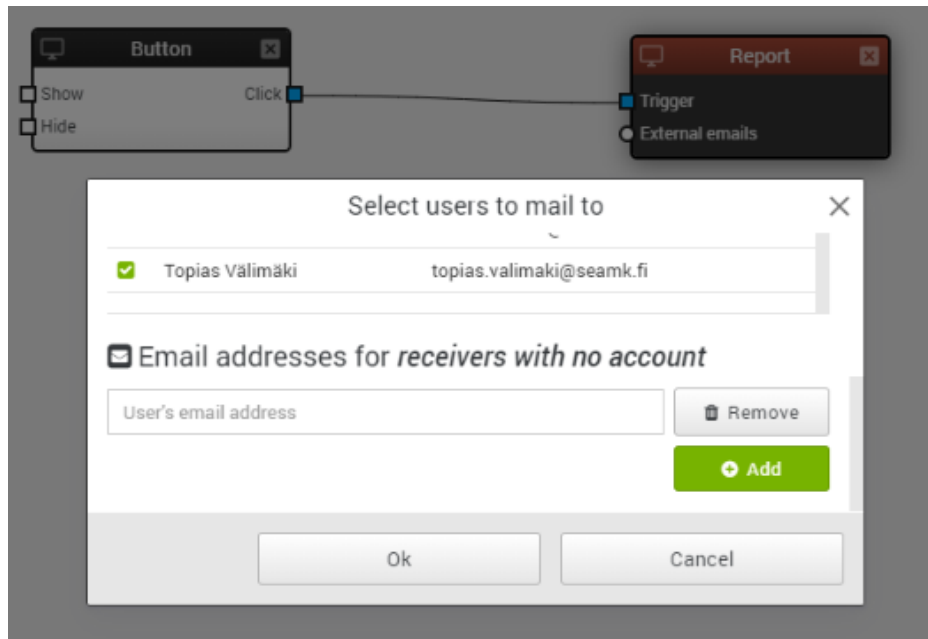
Selainpohjaisen käyttöliittymän ohjelmointi onnistuu helposti Data-flow editor -työkalulla, jossa on käytettävissä runsaasti toimintoja. Erilaiset toiminnallisuudet voidaan luoda perinteisistä ohjelmointikielistä tutuilla loogisilla porteilla, switch-case-valintarakenteilla tai sekvenssilohkoilla. Työkaluihin sisältyy myös muun muassa matemaattisia funktioita, suodattimia, ajastimia ja dataa simuloivia lohkoja. Kuviossa 18 esitetään yksinkertainen tapahtuman luonti vertailijaa käyttämällä. Mitattua arvoa verrataan raja-arvoon, jonka ylittyessä aktivoidaan hälytys. Tapahtumat jaetaan ilmoituksiin, varoituksiin ja hälytyksiin. Käyttäjän on mahdollista luoda myös oma tapahtumaluokka.



Kuvio 18. Hälytyksen aktivointi IoT-TICKET-palvelussa.

Tämänkaltaisessa ohjelmoinnissa piileekin palvelun hienous: toiminnallisuuksien luominen ja muokkaaminen eivät vaadi ohjelmointikielen osaamista, vaan palvelua käyttävä asiakas, oli tämä sitten yrityksen toimitusjohtaja tai myyntihenkilö, voi lyhyen tutustumisen jälkeen ylläpitää oman käyttöliittymänsä toimintoja.

Myös raporttien luomiseen on tarjolla käyttöliittymästä tutut työkalut, joiden avulla esimerkiksi laitteen eri antureiden tiedot on tallennettavissa vian ilmetessä. Raportteja on mahdollista lähettää myös syklisesti valittuina viikonpäivinä ja kellonaikoina, jolloin ne voivat ilmoittaa esimerkiksi järjestelmän tilastot viikon ajalta. Ne ovat saatavissa joko sähköpostiin (kuvio 19) tai ladattavissa tietokoneelle pdf-muodossa.



Kuvio 19. Raportin lähetys IoT-TICKET-palvelusta sähköpostiin.

Yksinkertaisten ja nopeiden sovellusten luomisen lisäksi palvelussa on eväitä monimutkaistenkin ohjelmistoratkaisujen tarpeisiin. IoT-TICKET mahdollistaa datan hyödyntämisen koko organisaatiossa. Samasta datasta voidaan muodostaa sekä anturidataa analysoiva sovellus että koko liiketoiminnan suoritusindikaattori KPI (Key Performance Indicator). Ohjelmistokehityksen ei tarvitse olla enää projekti tai hanke, vaan muutostarpeen havaitseva työntekijä voi tehdä itse muutoksen. Näin investoinneissa voidaan keskittyä tuottavien asioiden kehittämiseen.

Helppokäyttöisenä pilvialustana mainostettu IoT-TICKET lunasti lupauksensa käyttökokeilun jälkeen. Wapice on lisännyt YouTube-videopalveluun opetusvideoita, joiden avulla pääsee tutustumaan käyttöliittymän rakentamiseen ja palvelun ominaisuuksiin. Käyttäjän itse hallitsemassa palvelussa ei tule kilpailijoihin verrattuna yhtä suuria piilokuluja, kuten käyttöönotto- ja kehityskustannuksia. Tässä opinnäytetyössä datan lähettämiseen TwinCAT-ohjelmasta IoT-TICKET-palveluun käytettiin Seinäjoen ammattikorkeakoulun opiskelumateriaalikansiota haettua, C#-ohjelmointikielellä toteutettua esimerkkiprojektia.

6 DATAN LÄHETTÄMINEN

6.1 Parametrit erillisessä ohjelmassa

On olemassa muutamia vaihtoehtoja, joilla voidaan lähettää dataa logiikkaohjaimelta pilvipalveluun. Yhteyden voi muodostaa

- logiikkaohjaimen PLC-ohjelmassa
- logiikkaohjaimeen tallennetulla erillisellä ohjelmalla
- täysin itsenäisesti toimivalla tietokoneella, joka on muodostanut yhteyden logiikkaohjaimeen.

Kuten luvussa 3.3 todettiin, kolmas vaihtoehto sopii parhaiten varsinkin vanhempiin logiikkaohjaimiin.

Tiedonsiirtoyhteyden muodostaminen ja datan lukeminen erillisellä sovelluksella on hyvä periaate, jos halutaan PLC-ohjelman toimivan riippumatta yhteyden muodostamisesta tai datan lähettämisessä ilmenevistä ongelmista. Erillisen ohjelman rakentamiseen on tarjolla useita tapoja. Yksi niistä on Python-ohjelmointikielen ja sen TwinCAT-ympäristöön yhdistävän pyads-paketin käyttäminen (Lehmann 2015). Muun muassa Wapice tarjoaa verkkosivuillaan C#-, Java- ja Python-kielillä toteutettuja ohjelmia. Myös Beckhoffin IoT Data Agent -sovellus, joka esitellään luvussa 6.4, mahdollistaa datan lukemisen ohjainlaitteelta ja lähettämisen pilvipalveluun. Ohjainlaitteen ja pilvipalvelun välinen yhteys muodostetaan tässä sovelluksessa ilman ohjelmointikieltä.

Tavanomaisesti ADS-yhteyden luomiseen erillisestä sovelluksesta TwinCAT-ympäristöön tarvitaan ainakin kaksi parametria: laitteen identifioiva osoite verkossa eli AMS Net Id sekä ADS-portin numero. Jos lähettävä sovellus toimii samalla tietokoneella TwinCAT-ohjelman kanssa, ei AMS Net ID ole aina välttämätön. Kuten tämän työn esimerkeistä ilmenee, tavanomainen sovellus yhdistetään ADS-porttiin 851. Muut ADS-portit esitetään taulukossa 1.

Taulukko 1. ADS-portit TwinCAT-ympäristössä (Beckhoff, [viitattu 27.1.2020]).

ADS device	Port number
Message system	100
Real-time kernel	200
I/O	300
Reserved	400
NC	500
Run-time system 1 (TwinCAT 2)	801
Run-time system 2 (TwinCAT 2)	811
Run-time system 3 (TwinCAT 2)	821
Run-time system n	$791 + (n \cdot 10)$
Run-time system 1 (TwinCAT 3)	851
Run-time system 2 (TwinCAT 3)	852
Run-time system 3 (TwinCAT 3)	853
Run-time system n	$850 + n$
Camshaft controller	900

Usein TwinCAT-ohjelman muuttujien viittaukset erillisessä ohjelmassa ovat muotoa "ohjelman_nimi.muuttujan_nimi", esimerkiksi "MAIN.Variable1". Toki eri ohjelmointikielillä toteutetuissa ohjelmissa on eroavaisuuksia. Itse TwinCAT PLC -ohjelma ei vaadi tällaisessa tiedonsiirrossa lisäyksiä.

6.2 C#-ohjelma

C#-ohjelmointikieltä käytettäessä ohjelmointi on aloitettava lisäämällä projektiin TwinCAT.Ads-kirjasto. Tämä onnistuu Visual Studio -ympäristössä käyttämällä NuGet Package Manager -työkalua. Vaihtoehtoisesti TwinCAT.Ads.dll-tiedoston voi kopioida myös polusta C:\TwinCAT\AdsApi\NET\v4.0.30319. Ohjelmassa kirjastoa käytetään komennolla "using TwinCAT.Ads;". Beckhoff tarjoaa verkkosivuillaan useita esimerkkejä rajapinnan toteuttamisesta C#-kielellä: https://infosys.beckhoff.com/content/1033/tcsample_net/html/tcsample_net_intro.htm?id=1565353882024605360

Yhdistäminen PLC-projektiin tapahtuu kahvamuuttujien (handle) avulla. Kun TcAdsClient-tyyppisellä oliolla on otettu yhteys PLC-projektiin, tämän muuttujat voidaan yhdistää kahvamuuttujiin, kuten kuviossa 20 esitetään.

```
// Kahvat TwinCAT-muuttujiin
int startHandle;
int stopHandle;
int measurementDataHandle;

// Otetaan yhteys ADS-clientillä TwinCAT PLC:hen portin 851 kautta
TcAdsClient adsClient;
adsClient = new TcAdsClient();
adsClient.Connect(851); // Paikallinen TwinCAT PLC ei tarvitse AMS Net ID -parametria

// Kytetään kahvat TwinCAT-muuttujiin.
startHandle = adsClient.CreateVariableHandle("MAIN.bStart");
stopHandle = adsClient.CreateVariableHandle("MAIN.bStop");
measurementDataHandle = adsClient.CreateVariableHandle("MAIN.ST_Measurements");
```

Kuvio 20. Kahvamuuttujien ja ADS-yhteyden määrittely C#-ohjelmassa.

Kahvamuuttujia voi lukea ja kirjoittaa käyttämällä ReadAny- ja WriteAny-metodeja kuvion 21 tapaan. ReadAny-metodi tarvitsee parametreiksi kahvamuuttujan tyyppineen, kun taas WriteAny-metodille on annettava kahvamuuttuja ja sille uusi arvo. Muuttujia on mahdollista lukea myös AdsStream-tyyppisestä datavirrasta. Sen koko voidaan määrittää tavuina. Ilman parametria datavirran koko on alkuarvoltaan nolla, mutta se kasvaa muuttujien myötä. Lisätietoa tästä luokasta (AdsStream Class) löytyy Beckhoffin verkkosivuilta osoitteesta https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_adsnetref/7313041547.html&id=.

```
// Luodaan Ads-stream, jonka parametrina on muuttujien koko tavuina
AdsStream adsStream = new AdsStream(4);
adsClient.Read(measurementDataHandle, adsStream);
// Luetaan datavirtaa
BinaryReader binReader = new BinaryReader(adsStream);
adsStream.Position = 0; // Siirrytään datavirran alkuun
int sensor1 = binReader.ReadInt32();
```

Kuvio 21. Muuttujien lukeminen C#-ohjelman AdsStream-datavirrasta.

Tässä opinnäytetyössä käytettiin Seinäjoen ammattikorkeakoulun opiskelumateriaaleihin kuuluvaa ohjelmaa, joka lukee TwinCAT PLC -projektista muuttujia ja lähettää niitä IoT-TICKET-palveluun. Ohjelma vaatii TwinCat.Ads-paketin lisäksi Wapice.IoTTicket.RestClient-paketin.

Ohjelmaan kirjoitetaan käyttäjänimi ja SecureString-tyyppinen salasana (kuvio 22). Tämän tyyppinen muuttuja on kirjoitussuojattu, minkä vuoksi sen arvo asetetaan merkki kerrallaan AppendChar-komennolla. Näillä käyttäjätiedoilla IoTTicketClient-tyyppinen olio kirjautuu IoT-TICKET-palveluun.

```
// IoTTicketClient: parametreina osoite, käyttäjänimi ja salasana
var username = "LisääKäyttäjänimi";
var pwd = new SecureString();
"LisääSalasana".ToList().ForEach(pwd.AppendChar);

var client = new IoTTicketClient(
    new Uri("https://my.iot-ticket.com/api/v1/"),
    username,
    pwd);
```

Kuvio 22. IoTTicketClient-olio C#-ohjelmassa.

Muuttujia luetaan PLC-ohjelmasta kuvion 23 TcAdsClient-tyyppisellä oliolla. Paikalliseen tietokoneeseen yhdistettäessä ainoa parametri on ADS-portti. PLC-ohjelmassa mittaukset tallentuvat struktuurimuuttujaan, jota varten luodaan kahvamuuttuja.

```
// Luodaan AdsClient-oliolla yhteys
adsClient = new TcAdsClient();
adsClient.Connect(851); // Paikallinen tietokone, AMS Net ID -parametria ei tarvita
measurementDataHandle = adsClient.CreateVariableHandle("MAIN.mittaukset");
```

Kuvio 23. ADS-yhteyden muodostaminen C#-ohjelmasta PLC-ohjelmaan.

Palveluun yhdistettävälle laitteelle on mahdollista määrittää ominaisuuksia, kuten valmistaja ja malli. Tämän lisäksi voidaan luoda omia attribuutteja, joille määritetään avain ja arvo. Kuviossa 24 laitteen lisäominaisuus on OS (Operating System, suom. käyttöjärjestelmä) ja sen arvo Windows 10.

```
// Laitteen nimi
private static string myDeviceName = "TVaPC";
// Laitteen attribuutit
var device = new Device
{
    Name = myDeviceName,
    Description = "My computer",
    Manufacturer = "HP",
    Type = "PC",
    Attributes =
        new List<DeviceAttribute>
        {
            new DeviceAttribute("OS", "Windows 10")
        }
};
```

Kuvio 24. Laitteen attribuutit C#-ohjelmassa.

Laite rekisteröidään palveluun kuviossa 25 kuvatulla tavalla. Ensin verrataan laitteen nimeä luotujen laitteiden nimiin. Näin vältetään duplikaattien eli laitteiden kaksoiskappaleiden luomiselta.

```
// Listaa laitteet palvelusta
var allClients = client.GetDevicesAsync(100, 0, cts.Token).Result.Result;
// Jos laitetta ei löydy palvelusta, rekisteröidään se
DeviceDetails deviceDetails = allClients.FirstOrDefault(a => a.Name.Equals(myDeviceName));
if (deviceDetails == null)
{
    deviceDetails = client.RegisterDeviceAsync(device, cts.Token).Result;
}
```

Kuvio 25. Laitteen rekisteröinti C#-ohjelmassa.

Palveluun lähetetään tietoa kuvion 26 mukaisissa datanode-muuttujissa, joille voidaan sisällön lisäksi määrittää nimi, polku ja mitatun arvon yksikkö.

```
// Datanode-lista, jossa yksi datanode-muuttuja
List<DatanodeWritableValue> datapoints =
    new List<DatanodeWritableValue>
    {
        new DatanodeWritableValue
        {
            Name = "Temperature",
            Path = "Laboratory/Chamber",
            Unit = "°C",
            Value = mittaus.Measurement2,
        },
    };
var writeResult =
client.WriteDatapointCollectionAsync(
    deviceDetails.Id,
    datapoints,
    cts.Token).IsCompleted;
```

Kuvio 26. Esimerkki datanode-muuttujasta C#-ohjelmassa.

6.3 Python-ohjelma

Yksinkertaisen syntaksin vuoksi helposti luettavan ja varsin suosituksen Python-ohjelmointikielen ympäristöön on tarjolla runsaasti rajapintamahdollisuuksia, jotka soveltuvat eri ohjelmiin ja esimerkiksi tietokantoihin. Yksi mahdollisuus on ADS-rajapinta pyads (Lehmann 2015). Rajapintapaketti sisältää funktioita yhteyden muodostamiseksi TwinCAT-laitteisiin.

Yhteyttä muodostavan ohjelman alussa siihen tuodaan pyads-paketti komennolla "import pyads". PLC-ohjelman ollessa samalla tietokoneella käytetään yhteyden luomiseen paikallista AMS Net ID -osoitetta ja ADS-porttia 851. Nämä parametrit tallennetaan adr-muuttujaan (kuvio 27).

```

import pyads
import time
import json
import msvcrt

# Avataan portti
pyads.open_port()
time.sleep(0.5)
# Lisätään muuttujaan AMS Net Id ja ADS-portti
adr = pyads.AmsAddr('127.0.0.1.1',851)
time.sleep(0.5)

```

Kuvio 27. Yhteyden muodostaminen Python-ohjelmasta paikalliseen PLC-ohjelmaan.

Arvojen lukemiseen käytetään `read_by_name`-komentoa (kuvio 28), jolle annetaan parametreina edellä kirjoitettu `adr`-muuttuja, PLC-ohjelman muuttujien polku ja niiden tyyppi.

```

# PLC-ohjelman muuttujien arvot
meas1 = pyads.read_by_name(
    adr,
    'MAIN.Temperature',
    pyads.PLCTYPE_REAL)
time.sleep(0.3)
meas2 = pyads.read_by_name(
    adr,
    'MAIN.Pressure',
    pyads.PLCTYPE_REAL)
time.sleep(0.3)
meas3 = pyads.read_by_name(
    adr,
    'MAIN.Humidity',
    pyads.PLCTYPE_REAL)
time.sleep(0.3)

```

Kuvio 28. Muuttujien lukeminen nimen perusteella Python-ohjelmassa.

Muuttujia on mahdollista lukea ja kirjoittaa myös muistiavaruuteen viittaamalla, jolloin parametreiksi annetaan muistialue, kirjoitettava paikka alueella, muuttujan arvo ja sen tyyppi (kuvio 29).

```

# Yhdistäminen
PLC1 = pyads.Connection('127.0.0.1.1', pyads.PORT_SPS1)
PLC1.open()
# Kirjoitetaan "12345" muistiavaruuden tavuun MDW0
PLC1.write(
    pyads.INDEXGROUP_MEMORYBYTE,
    0,
    12345,
    pyads.PLCTYPE_UDINT)

```

Kuvio 29. Muuttujien kirjoitus muistiavaruuteen viittaamalla Python-ohjelmassa.

C#-kielellä kirjoitetun ohjelman tavoin PLC-muuttujia on mahdollista lukea kahva-muuttujien avulla (kuvio 30). Kahva muodostetaan `get_handle`-komennolla. Mikäli tiedonsiirtoyhteyden kaista on kapea, voidaan lisätilan saamiseksi kahva vapauttaa komennolla `release_handle`.

```
StartHandle = PLC1.get_handle('MAIN.Start')
PLC1.write_by_name(
    '',
    True,
    pyads.PLCTYPE_BOOL,
    handle=StartHandle)
PLC1.release_handle(StartHandle)
```

Kuvio 30. Muuttujan kirjoittaminen kahvamuuttujan avulla Python-ohjelmassa.

Dataa voidaan lähettää eteenpäin esimerkiksi MQTT- tai HTTP-protokollaa käyttäen. MQTT-yhteyttä käytettäessä ohjelmaan lisätään Eclipse Foundation -järjestön Paho-sovelluskirjasto. Ohjelmassa luodulla MQTT-asiakkaalla otetaan yhteys HiveMQ:n MQTT broker -osoitteeseen `connect`-komennolla. Kuvion 31 esimerkissä viesti muunnetaan JSON-muotoon ennen kuin se lähetetään `publish`-komennolla.

```
import paho.mqtt.client as mqtt

# MQTT broker -osoite
broker_address="broker.hivemq.com"
# Luodaan yhteys MQTT client -oliolla
mqttClient = mqtt.Client(client_id="TwinCAT_PLC")
mqttClient.connect(broker_address)

# Muunnetaan viesti JSON-muotoon
message = json.dumps(measurement)
# Lähetetään data, parametreina aihe (topic) ja sisältö
mqttClient.publish("myTopic", message)
```

Kuvio 31. Viestin lähettäminen (publish) Python-ohjelmassa.

HTTP-tiedonsiirtoprotokollaa käytettäessä ohjelmaan lisätään `requests`-kirjasto, jonka käyttöohjeet ovat osoitteessa <https://realpython.com/python-requests/>. Kirjaston avulla on mahdollista muun muassa vastaanottaa ja lähettää viestejä `get`- ja `post`-komennoilla. Kuviossa 32 kirjoitettuun komenttoon kuuluvat parametreina vastaanottavan tai lähettävän kohteen sijainti ja viestin sisältö.

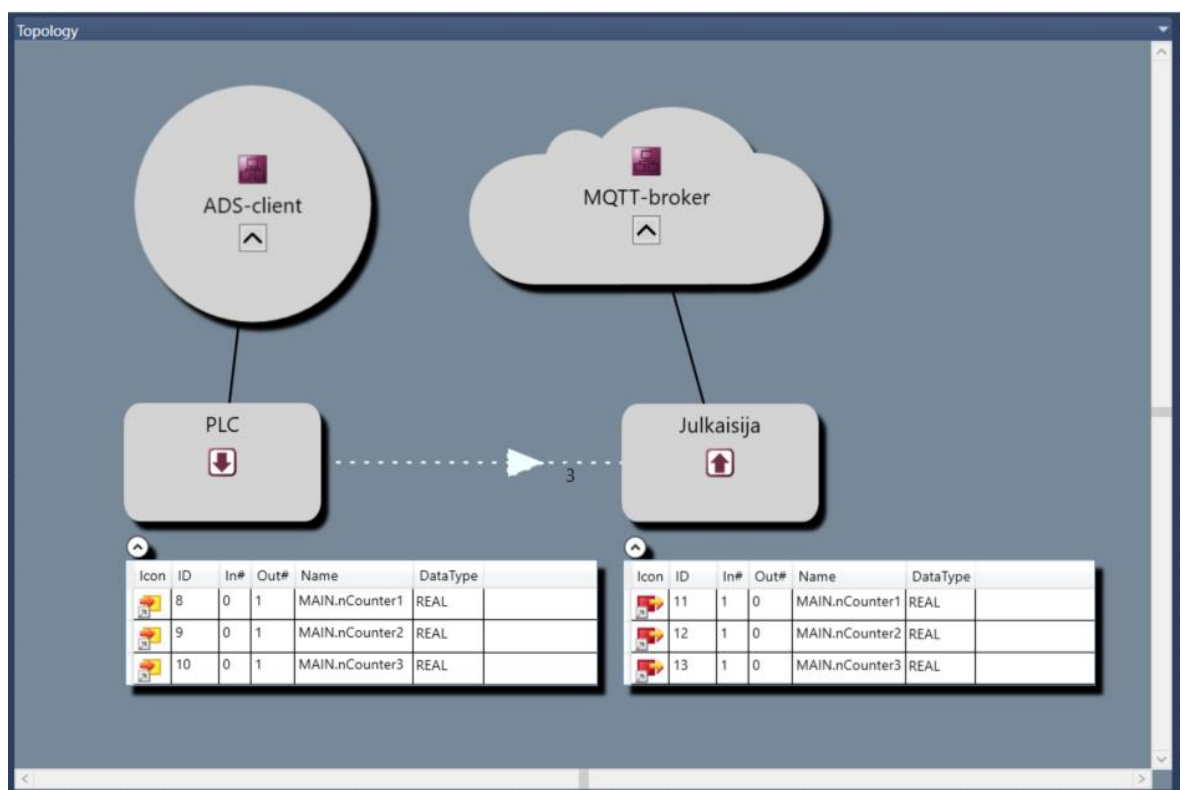
```
import requests

message = requests.post(
    'https://httpbin.org/post',
    data={'Temperature': 'TempSensor1'})
```

Kuvio 32. Viestin lähettäminen (post) Python-ohjelmassa.

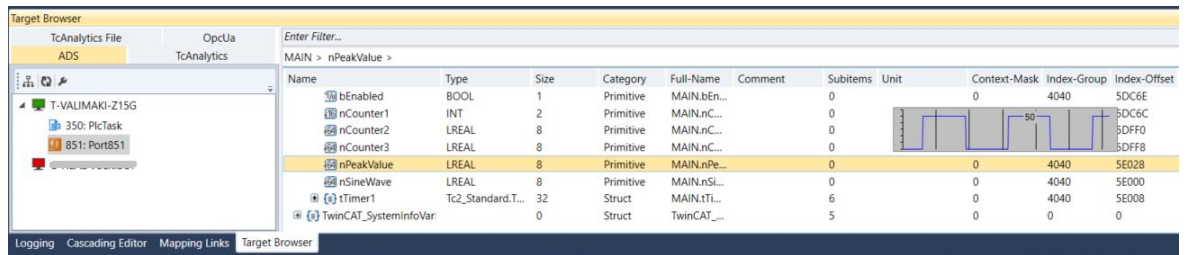
6.4 IoT Data Agent

Beckhoffin TwinCAT 3 Connectivity -sovellusryhmään kuuluva IoT Data Agent yhdistää PLC-ohjelman pilvipalveluun. Sovellus tekee tämän kahdella tavalla: ohjainlaitteelle asennettuna tai erilliseltä tietokoneelta, joka ADS- tai OPC UA -yhteyttä käyttäen lukee ja kirjoittaa muuttujia logiikkaohjaimelta. Alustasta riippumaton OPC UA -yhteys mahdollistaa yhdistämisen kolmannen osapuolen ohjainlaitteisiin. Data Agent -sovelluksessa verkkotopologian rakentaminen onnistuu kuvion 33 havainnollistamassa käyttöliittymässä ilman ohjelmointikielen kirjoittamista.



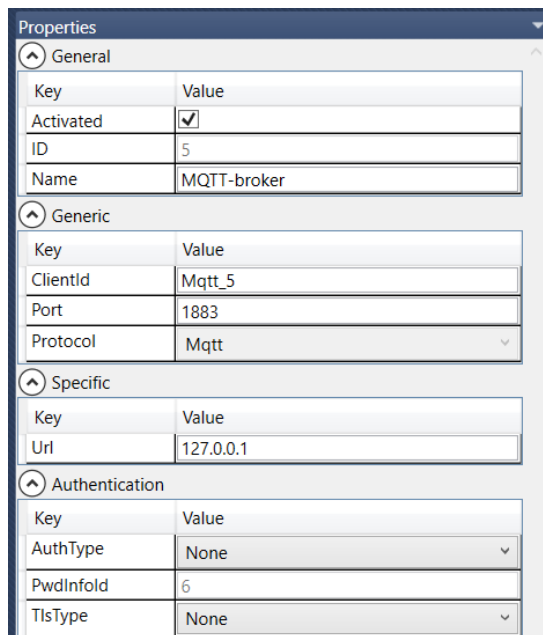
Kuvio 33. IoT Data Agent -sovelluksen topologianäkymä.

MQTT-yhteys luotiin lisäämällä ensin ADS client, jolle oli annettava parametreina dataa lähettävän logiikkaohjaimen AMS Net ID -osoite ja ADS-portti. Ohjelmaa testattaessa mittausarvoja simuloitiin paikallisesta PLC-ohjelmasta. Konfiguraatioon lisättiin myös paikallisella tietokoneella toimiva MQTT broker. Tämän jälkeen valittiin ADS-portin kautta lähetettävät muuttujat Target Browser -ikkunasta. Muuttujien arvoja voidaan tarkastella reaaliaikaisesti painamalla välilyöntipainiketta halutun muuttujan kohdalla, jolloin ohjelma esittää arvon kuvion 34 oikeassa reunassa näkyvässä viivadiagrammissa.



Kuvio 34. Muuttujien valitseminen IoT Data Agent -sovellukseen.

Kuviossa 35 näkyvistä viestien välittäjän (MQTT broker) asetuksista voidaan muuttaa esimerkiksi salausmenetelmää, kuten salasanaa tai sertifikaatteja.



Kuvio 35. MQTT-yhteyden parametrit IoT Data Agent -sovelluksessa.

Viestien välittäjänä toimi avoimen lähdekoodin Eclipse Mosquitto. Mittausarvojen vastaanotto onnistui Mosquitto Subscriber -ohjelmalla sekä samaan verkkoon yhdistetyltä matkapuhelimelta kolmannen osapuolen MQTT Dash -ilmaissovellusta käyttäen.

6.5 PLC-ohjelma

Jos yhteys halutaan selvyiden vuoksi muodostaa PLC-ohjelmassa, tässä tapauksessa TwinCAT-projektissa, Beckhoff tarjoaa useita siihen soveltuvia esimerkkiohjelmaa. Verkkosivuilta on ladattavissa yksinkertaisten MQTT-kommunikaatiota käyt-

tävien ohjelmien lisäksi esimerkkejä yhteyden muodostamisesta useaan kaupalliseen pilvipalveluun, kuten Amazon AWS-, Microsoft Azure- ja IBM Watson -palveluihin. (Beckhoff, [viitattu 21.1.2020].)

Datan lähettäminen alla esitellyssä ohjelmassa on toteutettu ST-ohjelmointikielellä. Ohjelma on hieman muokattu versio Beckhoffin sivuilta ladattavasta `lotMqttSampleUsingQueue`-ohjelmasta. Alussa on asetettava `FB_lotMqttClient`-tyyppiselle lohkolle parametrit. Tämä tehdään vain kerran. Osa parametreista on valinnaisia, ja MQTT-yhteyden muodostaminen onnistuukin vain muutamilla pakollisilla parametreilla, jotka näkyvät kuviossa 36. Saapuvat viestit tallennetaan viestijonoon, mikä onnistuu käyttämällä `fbMessageQueue`-lohkoa.

```
//
// Yhteysparametrit
IF bSetParameter THEN
  bSetParameter := FALSE;
  fbMqttClient.sHostName      := 'localhost';    // Hostin nimi/IP-osoite. Tyhjä = localhost.
  fbMqttClient.nHostPort     := 1883;           // Hostin portti.
  // fbMqttClient.sClientId   := 'MyTcMqttClient'; // Tyhjä = määritetään automaattisesti.
  fbMqttClient.sTopicPrefix  := '';            // Aihe, joka liitetään jokaiseen viestiin.
  // fbMqttClient.nKeepAlive  := 60;             // Yhteyden tarkastusväli sekunteina.
  // fbMqttClient.sUserName   := ;               // Käyttäjänimi.
  // fbMqttClient.sUserPassword := ;             // Salasana.
  // fbMqttClient.stWill      := ;               // Yhteyden katketessa lähetettävä viesti.
  // fbMqttClient.stTLS       := ;               // TLS-suojausparametri.
  fbMqttClient.ipMessageQueue := fbMessageQueue; // Saapuvien viestin tallennus jonoon.
END_IF
```

Kuvio 36. Parametrit MQTT-yhteyden muodostamiseen PLC-ohjelmasta.

Client-lohkon yhteyden muodostava `Execute`-metodi suoritetaan jokaisella ohjelmakierrolla (kuvio 37). Muuttujaan `hrErrorOccurred` tulee ilmenevän virheen koodi heksadesimaalilukuna. Virheen kuvaus saadaan vertaamalla lukua Beckhoffin tarjoamaan `ADS Return Codes` -listaan, joka on luettavissa muun muassa osoitteesta: <https://infosys.beckhoff.com/english.php?content=../content/1033/device-manager/54043195902722955.html&id=>.

```
//
// Yhteyden muodostus
// Tämä tehdään joka ohjelmakierrolla yhteyden ylläpitämiseksi.
fbMqttClient.Execute(bConnect);
IF fbMqttClient.bError THEN
  // Tähän voi lisätä myös valinnaista virnehallintaa.
  hrErrorOccurred := fbMqttClient.hrErrorCode;
END_IF
```

Kuvio 37. Yhteyden muodostus PLC-ohjelmasta.

Viestejä vastaanotetaan käyttämällä `Subscribe`-metodia (kuvio 38). Parametrina annetaan aiheen lisäksi myös QoS-taso. Jos viestijonossa on viestejä, jono puretaan

Dequeue-metodilla. Sen jälkeen viestin osat eli aihe (topic) ja sisältö (payload) tallennetaan muuttujiin.

```
//
// Viestin vastaanotto
IF fbMqttClient.bConnected THEN
    // Viestien tilaaminen
    IF NOT bSubscribed THEN
        bSubscribed := fbMqttClient.Subscribe(
            sTopic := sTopicSub,
            eQoS := TcIotMqttQos.AtMostOnceDelivery);
        IF fbMqttClient.bError THEN
            // Tähän voi lisätä myös valinnaista virreehallintaa.
            hrErrorOccurred := fbMqttClient.hrErrorCode;
        END_IF
    END_IF
END_IF
//
// Vastaanotettujen viestien purkaminen jonosta
IF fbMessageQueue.nQueuedMessages > 0 THEN
    // Dequeue-metodilla päästään käsiksi fbMessage-viestin osiin
    IF fbMessageQueue.Dequeue(fbMessage:=fbMessage) THEN
        fbMessage.GetTopic( // GetTopic() palauttaa viestin topicin
            pTopic := ADR(sTopicRcv),
            nTopicSize := SIZEOF(sTopicRcv) );
        fbMessage.GetPayload( // GetPayload() palauttaa viestin sisällön
            pPayload := ADR(sPayloadRcv),
            nPayloadSize:=SIZEOF(sPayloadRcv),
            bSetNullTermination:=FALSE);
    END_IF
END_IF
```

Kuvio 38. Viestin vastaanottaminen PLC-ohjelmassa.

Kuviossa 39 viesti lähetetään bSendMessage-muuttujalla, jos yhteys on muodostettu MQTT broker -osoitteeseen. Aihe on ennalta määritetty sTopicPub-muuttuja, ja sisältö on teksti "MyMessage", johon liitetään juokseva numerointi.

```
//
// Viestin lähetys
// Suoritetaan vain, jos yhteys muodostettu brokeriin (bConnected = TRUE)
IF fbMqttClient.bConnected THEN
    IF bSendMessage THEN // Lähetetään viesti
        i := i + 1;
        sPayloadPub := CONCAT('MyMessage', TO_STRING(i)); // Viestin sisältö
        fbMqttClient.Publish( // Viestin julkaisu
            sTopic := sTopicPub,
            pPayload := ADR(sPayloadPub),
            nPayloadSize:= LEN2(ADR(sPayloadPub))+1,
            eQoS := TcIotMqttQos.AtMostOnceDelivery,
            bRetain := FALSE,
            bQueue := FALSE);
        IF fbMqttClient.bError THEN
            // Tähän voi lisätä myös valinnaista virreehallintaa.
            hrErrorOccurred := fbMqttClient.hrErrorCode;
        END_IF
    END_IF
END_IF
```

Kuvio 39. Viestin lähettäminen PLC-ohjelmasta.

7 YHTEENVETO

7.1 Tutkimuksen tulokset

Tämän opinnäytetyön tavoitteena oli tutkia tapoja lähettää dataa logiikkaohjaimelta pilvipalveluun. Työ rajattiin käyttämällä dataa lähettävänä osana Beckhoffin TwinCAT-ympäristöä ja tutustumalla vain ennalta valittuihin dataa analysoiviin palveluihin.

Työ aloitettiin perehtymällä pilvialustoihin ja -palveluihin sekä teollisen internetin mahdollisuuksiin ja työssä käytettäviin tiedonsiirtoprotokolliin. Tämän jälkeen päätettiin tutustua Microsoft Azure- ja Amazon AWS IoT -palveluihin. Molemmissa tapauksissa ominaisuuksien testaaminen jäi niiden laajuuteen nähden hyvin suppeaksi. Tulevien käyttökohteiden ollessa luultavasti pienehköjä tehtiin työn alkupuolella päätös tutustua tarkemmin Wapicen ketterään IoT-TICKET-palveluun. Valinta osoittautuikin onnistuneeksi. Kevyen pilvipalvelun luonti onnistuu lyhyessäkin ajassa, mutta laaja työkaluarsenaali mahdollistaa kookkaankin järjestelmän etävalvonnan. Myös suhteellisen tuore TwinCAT Analytics -tuoteperhe tarjosi kiinnostavan datan analysointiratkaisun. Työssä jätettiin kuitenkin toteuttamatta laaja Analytics-sovellus tietokantoihin ja etäyhteyksiin. Näitä neljää IoT-ratkaisumahdollisuutta ei suinkaan verrattu keskenään, sillä ne ovat erilaisia palveluita. Sekä Azure-että AWS-palvelusta voisi molemmista kirjoittaa kokonaisen opinnäytetyön. TwinCAT Analytics -tuotteet eivät puolestaan muodosta valmista pilvipalvelua, mutta niiden avulla voi rakentaa toimivan, dataa visuaalisesti analysoivan IoT-ratkaisun.

Beckhoffin CX-sarjan logiikkaohjaimelta muuttujien tuominen analysoitavaksi onnistui vaivattomasti. Kolmannen osapuolen laitteiden yhdistämistä OPC UA -kommunikaatiota käyttäen ei testattu. Tähän alustariippumattomaan kommunikaatioon Beckhoff tarjoaa TF6100 TC3 OPC UA -komponenttiryhmittä. Tällaisen tiedonsiirto-ratkaisun etuna on se, että TwinCAT OPC UA Server on asennettavissa edullisiinkin logiikkaohjaimiin (Beckhoff 2019b, 10). Datan lähettämistä paikalliselta tietokoneelta testattiin kahdella tavalla: TwinCAT PLC -ohjelmasta sekä erillisistä sovelluksista, jotka muodostivat ADS-yhteyden PLC-ohjelmaan. IoT Data Agent -rajapinta-

sovellus todettiin graafisen käyttöliittymänsä vuoksi käteväksi. Sovelluksen käyttö-opas sisältää useita esimerkkejä pilvipalveluihin yhdistämisestä ja teoriaa muun muassa MQTT-kommunikaatiosta (Beckhoff 2018a). Myös Python- ja C#-kielillä toteutetuilla ohjelmilla muuttujien arvojen lukeminen ja edelleenlähetys onnistui ADS-ra-japintaa käyttäen helposti. Python pyads -kirjasto on hyvin dokumentoitu ja C#-kie-len esimerkkisovelluksia on ladattavissa Beckhoffin verkkosivuilta.

Tutustumalla verkossa luettaviin dokumentaatioihin ja Seinäjoen ammattikorkea-koulun opiskelumateriaalikansioista haettuihin esimerkkisovelluksiin yritykselle saa-tiin tietoa eri tavoista siirtää mittaustietoa TwinCAT-ympäristöstä. Useissa sovellus-kohteissa vaivaton IoT Data Agent -sovellus riittänee, mutta tulevaisuutta varten nähtiin tarpeelliseksi tutustua myös Python- ja C#-kielellä rakennettuihin sovelluk-siin. Dokumentoituja komentoja pystytään käyttämään tulevissa projekteissa tiedon-siirtoon PLC-projektista pilvipalveluun.

7.2 Pohdinta

Azure- ja AWS IoT -palveluiden poisjättäminen opinnäytetyöstä olisi jättänyt enem-män aikaa eri tiedonsiirtomahdollisuuksiin tutustumiseen ja valmiimman Analytics-tai IoT-TICKET-sovelluksen rakentamiseen. Microsoftin ja Amazonin rakentamien palveluiden suosion vuoksi niihin tutustuminen voi kuitenkin tulevaisuudessa osoit-tautua hyödylliseksi. Myös dataa lähettävässä osuudessa yhden konkreettisen ja valmiin sovellusesimerkin sijasta pidettiin tärkeämpänä luoda yleiskatsaus saata-villa oleviin rajapintoihin. TwinCAT-ympäristö valikoitui dataa lähettäväksi osapuo-leksi, koska se oli entuudestaan tuttu yritykselle. Valintaa auttoi myös helppo liitet-tävyys eri ohjelmiin ja Beckhoffin tarjoama runsas ja selkeä dokumentaatio.

IoT-alustan käyttöönotto ei ole helposti toteutettavissa, vaikka sen tekijänä olisi tuo-tantolaitos selkeine tarpeineen tai useiden valmistajien automaatiolaitteita käyttävä suunnittelutoimisto. Markkinoilla on tarjolla pilvialustoja ja valmiita palveluita useilta toimittajilta, joten asiaan perehtyminen vie aikaa. Vaikka tässä työssä tutkittiinkin suosittuja palveluita, jäi monia laajalti käytössä olevia palveluita ja alustoja testaa-matta. Esimerkiksi Siemensin MindSphere on ilmeinen vaihtoehto kokeiltavaksi var-sinkin muita Siemens-tuotteita käyttäville yrityksille.

Kotimaassa pienten ja keskisuurten tietotekniikka-alan yritysten lisäksi suuret kone- ja laitevalmistajat ovat nähneet teollisen internetin hyödyt ja investoineet omien IoT-ratkaisujensa kehitykseen. Valmiiden IoT-ratkaisujen hinnat laskenevat suomalais-ten ohjelmistoyritysten oppiessa lisää IoT-alustoista ja niiden tuomista mahdollisuuksista. Näin yhä useammat tuotantolaitokset ja esimerkiksi maataloustuottajat voivat saada merkittävää hyötyä dataa analysoivasta pilvipalvelusta. Investointia suunniteltaessa on kuitenkin otettava huomioon palvelun lisenssimaksujen lisäksi lähes varma ohjelmistomuutosten tarve. Ohjelmistoon tehtävä muutos edellyttää perinteisesti vaatimusmäärittelyn, itse toteutuksen sekä sovelluksen testauksen. IoT-TICKET-palvelun kaltainen alusta madaltaa omalta osaltaan kynnystä rakentaa IoT-ratkaisuja, kun yrityksessä ei tarvitse välttämättä olla ohjelmointiosaamista. Tällainen etävalvonnan ja data-analytiikan helppokäyttöisyys avaa ovia uusille liiketoimintamalleille, joista annettiin esimerkkejä luvussa 3.3. Räätelöintiä vaille valmis pilvialusta ei edellytä myöskään tietoturva-asioiden hallintaa, vaikkei niihin perehtyminen olekaan haitaksi. IoT-ratkaisun rakentaminen yrityksen omilla visualisointi- ja analysointisovelluksilla pakottaa perehtymään tietoturvaan.

LÄHTEET

- Abeyasinghe, S. 2008. RESTful PHP Web Services. [Verkkokirja]. Birmingham: Packt Publishing. [Viitattu 14.2.2020]. Saatavana EBSCO eBooks Collection - tietokannasta. Vaatii käyttöoikeuden.
- Academy. 2019. Mitä data-analytiikka on ja miten se pyörittää maailmaa? [Verkkosivu]. Academic Work Academy Finland Oy. [Viitattu 14.1.2020]. Saatavana: <https://www.academy.fi/news/mita-data-analytiikka-on-ja-miten-se-pyorittaa-maailmaa>
- Amazon Web Services. 2018. Denial of Service Attack Mitigation on AWS. [Verkkojulkaisu]. Amazon Web Services, Inc. [Viitattu 19.1.2020]. Saatavana: https://d1.awsstatic.com/aws-answers/AWS_DDoS_Attack_Mitigation.pdf
- Amazon Web Services. Ei päiväystä. What is cloud computing? [Verkkosivu]. Amazon Web Services, Inc. [Viitattu 15.1.2020]. Saatavana: <https://aws.amazon.com/what-is-cloud-computing/>
- Amazon Web Services. Ei päiväystä. AWS IoT. [Verkkosivu]. Amazon Web Services, Inc. [Viitattu 3.4.2020]. Saatavana: https://aws.amazon.com/iot/?nc2=h_ql_prod_it_iot
- Amazon Web Services. Ei päiväystä. AWS Pricing. [Verkkosivu]. Amazon Web Services, Inc. [Viitattu 4.4.2020]. Saatavana: https://aws.amazon.com/pricing/?nc2=h_ql_pr_ln
- Austin-Morgan, T. & Wilkins, J. 2016. Retrofit your way to Industry 4.0. [Verkkosivu]. MA Business Ltd. [Viitattu 15.1.2020]. Saatavana: <http://www.eurekamagazine.co.uk/design-engineering-blogs/guest-blog-retrofit-your-way-to-industry-4-0/147234/>
- Beckhoff. 2018a. TC3 IoT Data Agent. [Verkkojulkaisu]. Beckhoff Automation GmbH & Co. KG. [Viitattu 16.1.2020]. Saatavana: https://download.beckhoff.com/download/document/automation/twin-cat3/TF6720_TC3_IoT_Data_Agent_EN.pdf
- Beckhoff. 2018b. Creating value: IoT and Data Analytics from Beckhoff. Esite. Beckhoff Automation GmbH & Co. KG.
- Beckhoff. 2019a. TC3 IoT Communication (MQTT). [Verkkojulkaisu]. Beckhoff Automation GmbH & Co. KG. [Viitattu 23.1.2020]. Saatavana: https://download.beckhoff.com/download/document/automation/twin-cat3/TF6701_TC3_IoT_Communication_MQTT_EN.pdf

- Beckhoff. 2019b. TC3 OPC UA. [Verkkojulkaisu]. Beckhoff Automation GmbH & Co. KG. [Viitattu 10.3.2020]. Saatavana: https://download.beckhoff.com/download/document/automation/twincat3/TF6100_TC3_OPC-UA_EN.pdf
- Beckhoff. Ei päiväystä. ADS-Communication. [Verkkosivu]. Beckhoff Automation GmbH & Co. KG. [Viitattu 16.1.2020]. Saatavana: https://infosys.beckhoff.com/english.php?content=../content/1033/cx8110_hw/5091854987.html&id=7660449179006258806
- Beckhoff. Ei päiväystä. Beckhoff Automation. [Verkkosivu]. Beckhoff Automation GmbH & Co. KG. [Viitattu 20.1.2020]. Saatavana: <https://www.beckhoff.com/english.asp?beckhoff/default.htm>
- Beckhoff. Ei päiväystä. TF6701 TC3 IoT Communication (MQTT) – Samples. [Verkkosivu]. Beckhoff Automation GmbH & Co. KG. [Viitattu 21.1.2020]. Saatavana: https://infosys.beckhoff.com/english.php?content=../content/1033/tf6701_tc3_iot_communication_mqtt/99079195194274059.html&id=3284888274642960706
- Beckhoff. Ei päiväystä. TwinCAT 3 eXtended Automation (XA). [Verkkosivu]. Beckhoff Automation GmbH & Co. KG. [Viitattu 23.1.2020]. Saatavana: <https://www.beckhoff.com/english.asp?twincat/twincat-3.htm?id=347919050530>
- Beckhoff. Ei päiväystä. T_AmsPort. [Verkkosivu]. Beckhoff Automation GmbH & Co. KG. [Viitattu 27.1.2020]. Saatavana: https://infosys.beckhoff.com/english.php?content=../content/1033/tcplclib_tc2_system/27021597795287307.html&id=7968917140057171517
- Beckhoff. Ei päiväystä. TwinCAT Analytics. [Verkkosivu]. Beckhoff Automation GmbH & Co. KG. [Viitattu 3.4.2020]. Saatavana: <https://www.beckhoff.com/analytics/>
- Buntz, B. 2019. 2020 Predictions: IoT Market to Expand, but Challenges Remain. [Verkkosivu]. Informa PLC. [Viitattu 14.1.2020]. Saatavana: <https://www.iot-worldtoday.com/2019/12/11/2020-predictions-iot-market-to-expand-but-challenges-remain/>
- Caplan. Ei päiväystä. Automaatiosuunnittelu- ja mekaniikkasuunnitteluasiantuntija Seinäjoella. [Verkkosivusto]. Caplan Oy. [Viitattu 3.4.2020]. Saatavana: <https://www.caplan.fi/>
- CERT-EU. 2014. DDoS Overview and Incident Response Guide. [Verkkojulkaisu]. The Computer Emergency Response Team for the EU institutions, agencies and bodies. [Viitattu 19.1.2020]. Saatavana: https://cert.europa.eu/static/White-Papers/CERT-EU-SWP_14_09_DDoS_final.pdf

- Cloud Carib. Ei päiväystä. The Beauty of Pay As You Go Offshore Cloud Computing. [Verkkosivu]. Cloud Carib Ltd. [Viitattu 16.1.2020]. Saatavana: <https://info.cloudcarib.com/blog/the-beauty-of-pay-as-you-go-offshore-cloud-computing>
- CNET. 2020. The best smart home devices of 2020. [Verkkosivu]. CBS Interactive Inc. [Viitattu 20.1.2020]. Saatavana: <https://www.cnet.com/news/the-best-smart-home-devices-of-2020-amazon-alexa-google-assistant-apple-homekit-nest-hub-echo-show>
- Collin, J. & Saarelainen, A. 2016. Teollinen internet. [Verkkokirja]. Helsinki: Talentum. [Viitattu 14.1.2020]. Saatavana Alma Talent Bisneskirjasto -tietokannasta. Vaatii käyttöoikeuden.
- Evans, P. & Annunziata, M. 2012. Industrial Internet: Pushing the Boundaries of Minds and Machines. [Verkkojulkaisu]. General Electric Company. [Viitattu 20.1.2020]. Saatavana: https://www.ge.com/docs/chapters/Industrial_Internet.pdf
- Hilbert, M. 2013. Big Data for Development: From Information- to Knowledge Societies. [Verkkojulkaisu]. Annenberg School of Communication, University of Southern California. [Viitattu 15.1.2020]. Saatavana: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2205145
- IBM. Ei päiväystä. IaaS, PaaS, and SaaS – IBM Cloud service models. [Verkkosivu]. International Business Machines Corporation. [Viitattu 19.1.2020]. Saatavana: <https://www.ibm.com/cloud/learn/iaas-paas-saas>
- Imperva. Ei päiväystä. DNS Spoofing. [Verkkosivu]. Imperva Inc. [Viitattu 19.1.2020]. Saatavana: <https://www.imperva.com/learn/application-security/dns-spoofing/>
- Kangasniemi, H. & Lintulahti, M. Ei päiväystä. Mikä on pilvipalvelu? [Verkkosivu]. Elisa Oyj. [Viitattu 16.1.2020]. Saatavana: <https://yksityisille.hub.elisa.fi/mika-on-pilvipalvelu/>
- Kodinturvatieto. 2019. Älykoti tekee arjesta helpompaa ja turvallisempaa. [Verkkosivu]. Verisure Oy. [Viitattu 20.1.2020]. Saatavana: <https://kodinturvatieto.fi/alykoti-tekee-arjesta-helpompaa-ja-turvallisempaa/>
- Kyberturvallisuuskeskus. 2016. Ohje 3/2016 Palvelunestohyökkäysten ehkäisy ja orjunta. [Verkkojulkaisu]. Liikenne- ja viestintävirasto Traficom. [Viitattu 19.1.2020]. Saatavana: https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/file/Ohje_3_2016_Palvelunestohyokkaysten_ehkaisy_ja_torjunta.pdf

- Lamarre, E. & May, B. 2017. Making sense of Internet of Things platforms. [Verkkosivu]. McKinsey & Company. [Viitattu 20.1.2020]. Saatavana: <https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/making-sense-of-internet-of-things-platforms>
- Lehmann, S. 2015. Python wrapper for TwinCATs ADS library. [Verkkosivu]. Python Software Foundation. [Viitattu 20.1.2020]. Saatavana: <https://py-ads.readthedocs.io/en/latest/index.html>
- Lombardo, T. 2019. Moving from Product to Product-as-a-Service. [Verkkosivu]. engineering.com, Inc. [Viitattu 16.1.2020]. Saatavana: <https://www.engineering.com/IOT/ArticleID/18205/Moving-from-Product-to-Product-as-a-Service.aspx>
- Marshall P. & Rinaldi S. 2017. Industrial Ethernet. [Verkkokirja]. ISA. [Viitattu 20.1.2020]. Saatavana Knovel -tietokannasta.
- Microsoft. Ei päiväystä. What is cloud computing? [Verkkosivu]. Microsoft Corporation. [Viitattu 15.1.2020]. Saatavana: <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/>
- Microsoft. Ei päiväystä. Azure DDoS Protection. [Verkkosivu]. Microsoft Corporation. [Viitattu 16.1.2020]. Saatavana: <https://azure.microsoft.com/en-us/services/ddos-protection/>
- Microsoft. Ei päiväystä. Azure IoT Hub pricing. [Verkkosivu]. Microsoft Corporation. [Viitattu 17.1.2020]. Saatavana: <https://azure.microsoft.com/en-us/pricing/details/iot-hub/>
- Microsoft. Ei päiväystä. Power BI Embedded pricing. [Verkkosivu]. Microsoft Corporation. [Viitattu 3.4.2020]. Saatavana: <https://azure.microsoft.com/en-us/pricing/details/power-bi-embedded/>
- Microsoft Power Bi. 26.6.2018. What is Power BI? [Video]. Microsoft Corporation. [Viitattu 3.4.2020]. Saatavana: <https://www.youtube.com/watch?v=yKTSLffVGbk>
- OPC Foundation. Ei päiväystä. Unified Architecture. [Verkkosivu]. OPC Foundation. [Viitattu 14.1.2020]. Saatavana: <https://opcfoundation.org/about/opc-technologies/opc-ua/>
- Siemens. Ei päiväystä. Industrial Cloud Computing. [Verkkosivu]. Siemens Industry Software Inc. [Viitattu 20.1.2020]. Saatavana: <https://www.plm.automation.siemens.com/global/en/our-story/glossary/industrial-cloud-computing/58773>

Vuoristo, A. 2020. Ohjausjärjestelmäasiantuntija. Beckhoff Automation Oy. Updating the project via MQTT. Esitelmä. Beckhoff Best Practice -seminaari 17.1.2020. Hyvinkää.

Wapice. 2019a. IoT API User Manual. [Verkkajulkaisu]. Wapice Oy. [Viitattu 3.2.2020]. Saatavana: https://iot-ticket.com/images/Files/IoT-Ticket.com_IoT_API.pdf

Wapice. 2019b. Wapicen IoT osaaminen vei voittoon Microsoftin maailmanlaajuisessa Partner of the Year-kilpailussa. [Verkkosivu]. Wapice Oy. [Viitattu 3.4.2020]. Saatavana: <https://www.wapice.com/fi/uutiset/microsoft-application-innovation-partner-of-the-year-palkinto>

Wapice. 7.4.2019. Build production-grade IoT applications fast with IoT-TICKET®, THE IoT Productivity Studio. [Video]. Vaasa: Wapice Oy. [Viitattu 12.3.2020]. Saatavana: <https://www.youtube.com/watch?v=ZDBEcMLgeHU>

Wapice. Ei päiväystä. IoT-TICKET. [Verkkosivu]. Wapice Oy. [Viitattu 19.1.2020]. Saatavana: <https://www.wapice.com/fi/tuotteet/iot-ticket>

Wapice. Ei päiväystä. IoT-TICKET: Alusta. [Verkkosivu]. Wapice Oy. [Viitattu 12.3.2020]. Saatavana: <https://iot-ticket.com/fi/alusta>