



Tomi Isomäki

Microsoft Azure-pilvipalvelun reuna- laitteiden kartoitus ja testaus

Metropolia Ammattikorkeakoulu

Insinööri (YAMK)

Älykäs teollisuus

Opinnäytetyö

13.4.2022

Tiivistelmä

Tekijä(t):	Tomi Isomäki
Otsikko:	Microsoft Azure-pilvipalvelun reunalaitteiden kartoitus ja testaus
Sivumäärä:	57 sivua + 2 liitettä
Aika:	13.4.2022
Tutkinto:	Insinööri (YAMK)
Tutkinto-ohjelma:	Älykäs teollisuus
Suuntautumisvaihtoehto:	
Ohjaaja(t):	Yliopettaja Jarno Varteva Lehtori Tuomo Heikkinen Yksikönpäällikkö, Tuote ja kehitys, Arttu Hanhela

Automatisoidut laitteet ja prosessit vaativat yhä tarkempaa seurantaa viranomaisvaatimusten kiristyessä ja automaatioasteen kasvaessa tuotannossa sekä laitteissa. Automaatiojärjestelmien datan hyväksi käyttäminen erilaisissa kappaletavara- ja prosessituotannoissa ovat kasvaneet nykyaikana voimakkaasti ja niitä pyritään hyödyntämään erilaisilla optimoinneilla ja koneoppimisien avulla.

Teollisuudessa on vuosien varrella käytetty erilaisia järjestelmiä ohjaamaan ja keräämään dataa tuotannon laitteista. Näiden tietojen saaminen moderneihin pilvipalveluihin tuottavat kustannustehokkaan infrastruktuurin ja ohjelmistoalustas laajasti hajautettuihin järjestelmiin.

Tämän työn tarkoituksena oli tutkia kirjallisuuskatsauksen avulla mitä sopivia laiteratkaisuja löytyy teollisuudessa olevien laitteiden ja Microsoft Azure-pilvipalvelun välille. Löydettyä ratkaisua testattiin käytännössä, kuinka data saatiin välitettyä Azure-pilvipalveluun. Testattavaksi laiteratkaisuksi löydettiin Siemens Simatic IoT2050 ja Moxa UC-3100 ja UC-8100 laitteet, jotka olivat Linux pohjaisia ratkaisuja.

Testauksia varten luotiin virtuaalinen Linux-ympäristö, tämän avulla valittuja laiteratkaisuja pystyttiin simuloimaan ilman oikeaa laitekoonpanoa. Testauksien avulla saatiin hahmotettua datan välityksen arkkitehtuuria ja mitä erilaisia ohjelmistoja se vaatii, sekä kuinka nämä saadaan käyteenotettua käytännössä. Lisäksi työssä perehdyttiin Node-RED-ohjelmointiin käytännössä.

Avainsanat: Microsoft Azure, Teollinen internet, yhdyskäytävä, OPC UA, Node-RED

Abstract

Author(s): Tomi Isomäki
Title: Mapping and testing Microsoft Azure cloud edge devices
Number of Pages: 57 pages + 2 appendices
Date: 13 April 2022

Degree: Master of the degree
Degree Programme: Intelligent Technology
Specialisation option:
Instructor(s): Jarno Varteva, Principal Lecturer
Tuomo Heikkinen, Senior Lecturer
Arttu Hanhela, Head of Unit, R&D

Automated devices and processes require more and more close monitoring as regulatory requirements tighten and automation rates increase in production and equipment. The use of data from automation systems in various product handling and process productions has grown strongly today and efforts are being made to utilize them through various optimizations and machine learning.

Over the years, industry has used various systems to control and collect data from production equipment. Getting this information into modern cloud services generates the cost of efficient infrastructure and software platform for broadly distributed systems.

The purpose of this work was to use a literature review to explore what suitable device solutions can be found between devices in industry and the Microsoft Azure cloud service. The solution found was practically tested to see how the data could be transmitted to the Azure cloud service. Siemens Simatic IoT2050 and Moxa UC-3100 and UC-8100 devices, which were Linux-based solutions, were found as the device solution for testing.

A virtual Linux environment was created for the tests, which allowed the selected device solutions to be simulated without the correct device configuration. The tests were used to understand the architecture of data transmission and what different software it requires, as well as how these can be used in practice. In addition, the work focused on Node-RED programming in practice.

Keywords: Microsoft Azure, IIoT gateway, Internet of things, OPC UA, Node-RED

Sisällys

1	Johdanto	1
1.1	Teollisuus 4.0	1
1.2	Automaatiojärjestelmä	1
1.3	Pilvipalvelut	3
1.4	Tutkimusmenetelmät ja tavoitteet	4
2	Kommunikointiprotokollat	6
2.1	Yleisesti	6
2.2	OSI- ja TCP-malli	6
2.3	Teollisuuden IoT-protokollat	8
2.4	Teolliset viestintäprotokollat	9
2.5	OPC UA	10
3	Tiedonvälitys	12
3.1	Esineiden internet	12
3.2	Yhteysmallit	16
3.3	IoT-Gateway	18
3.4	Node-RED	19
4	Microsoft Azure	21
4.1	Azure Historia	21
4.2	Azure palvelutyypit	21
4.3	IoT Hub	23
5	Laitteiden valinta	24
5.1	Valinta perusteet	24
5.1.1	Asennustapa	24
5.1.2	Yhteystavat	24
5.1.3	Kustannukset	24
5.1.4	Alustat	25
5.1.5	Tietoturva	25
5.1.6	Takuu ja saatavuus	25
5.2	Vertailuun valitut laitteet	25
5.2.1	Hewlett Packard	26
5.2.2	Dell	27

5.2.3	Omron	28
5.2.4	Phoenix	29
5.2.5	Siemens	30
5.2.6	mbNET	31
5.2.7	Moxa	32
5.2.8	Welotec	33
5.3	Laitteiden vertailu	34
5.3.1	Tulosten koonti	34
5.3.2	Tulosten analysointi	34
6	Testien kuvaus ja testit	37
6.1	Testausympäristö	37
6.2	OPC UA	38
6.2.1	Valintojen perusteet testauksiin	38
6.2.2	Prosys OPC UA Simulation server konfigurointi	38
6.2.3	ThingWorx Kepware Edge konfigurointi	42
6.3	Azure IoT Hub	45
6.4	Node-RED	48
6.4.1	Valinnan perusteet testauksiin	48
6.4.2	Node-RED-ohjelmointi	48
6.4.3	Node-RED kirjastot	48
6.4.4	OPC UA kommunikointi	49
6.4.5	Microsoft Azure kommunikointi	52
6.5	Tulokset	53
7	Yhteenveto	56
	Lähteet	58
	Liitteet	63
	Laittevertailu taulukko	64
	Laittevertailu kaaviot	67

1 Johdanto

Automatisoidut laitteet ja prosessit vaativat yhä tarkempaa seuranta- ja valvontaa viranomaisvaatimusten kiristyessä ja automaatioasteen kasvaessa tuotannossa sekä laitteissa. Modernit pilvipalvelut tarjoavat kustannustehokkaan infrastruktuurin ja ohjelmistoalustat laajasti hajautettujen järjestelmien ja esimerkiksi liikkuvien työkonereiden automaatioidatan keräämiseksi, esittämiseksi ja analysoimiseksi.

1.1 Teollisuus 4.0

Nykyään Teollisuusautomaatiossa puhutaan paljon käsitteestä Teollisuus 4.0, joka sai alkunsa vuonna 2011 Saksan Hannover messuilla, jossa Bosch kuvaili tieto- ja viestintäteknikka integrointia teolliseen tuotantoon. Vuonna 2013 Teollisuus 4.0 -työryhmä esitti Saksan hallitukselle käyttöönotto suosituksia, mistä syntyi teollinen vallankumous 4.0. [International Society of Automation (ISA) 2022]

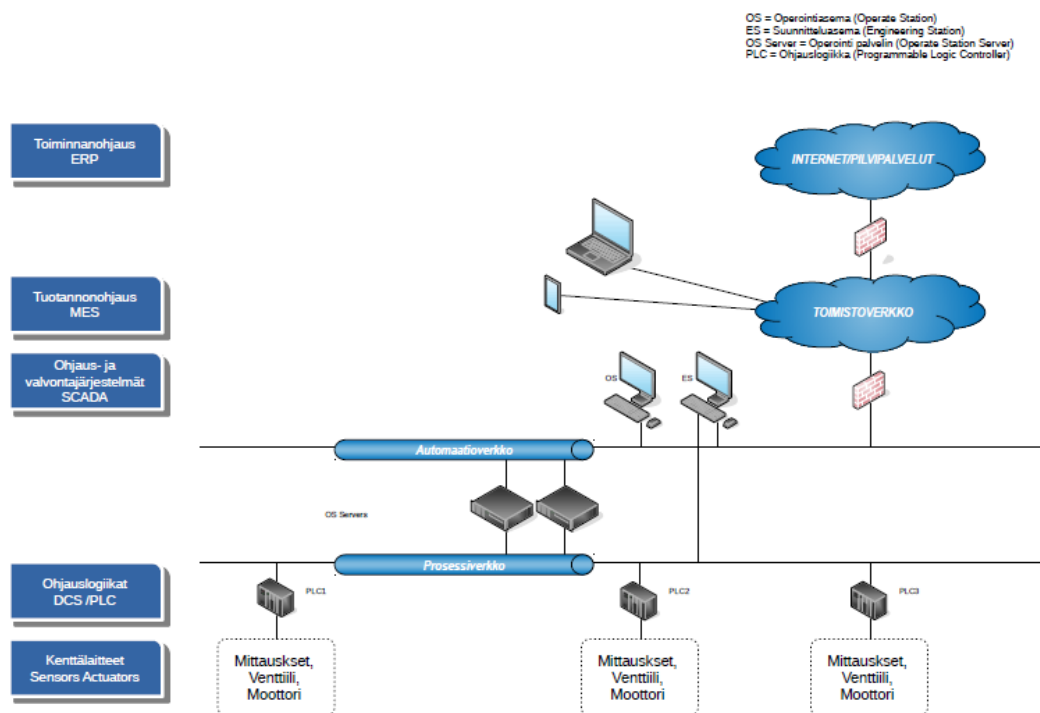
Teollisuus 4.0 tarkoittaa, kuinka fyysinen-, digitaalinen- ja virtuaalimaailma yhdistyvät luoden uusia älykkäitä tuotteita, menettelyjä ja prosesseja, sekä älykkäitä tehtaita. [International Society of Automation (ISA) 2022]

1.2 Automaatiojärjestelmä

Tiedon tallentaminen automaatiolaitteista tapahtuu niitä ohjaavien ja lukevien automaatiojärjestelmissä olevien integroitujen prosessihistoria ohjelmistojen avulla, jolla data tallennetaan paikallisesti tietokantaan.

Automaatiojärjestelmän liittämällä prosessilaitteiston ja yrityksen tietojärjestelmät yhteen. Automaation perustarkoituksena on parantaa tuotannon laatua, tehokkuutta turvallisuutta ja ympäristöystävällisyyttä [Magnus 2018; Suomen Automaatioseura ry 2007: 4]. Alla olevassa kuvassa (Kuva 1) on esitetty Automaatiojärjestelmän järjestelmäkaavion periaatekuva.

Automaatiojärjestelmän prosessien ohjaamista ja toimintaa kuvataan usein ISA-95 standardin mukaan viidessä eri tasossa [Magnus 2018: 2]. Järjestelmäkaaviossa (Kuva 1) on kuvattu vasemmalla puolella kyseiset tasot. Näistä kaksi ylintä tasoa muodostuvat yrityksen tiejärjestelmiin kuuluvista ERP (Enterprise Resource Planning) sekä MES (Manufacturing Execution System) [Suomen Automaatioseura ry 2007: 10-11]. Loput kolme tasoa tietojärjestelmien alla koostuu automaatiojärjestelmästä.

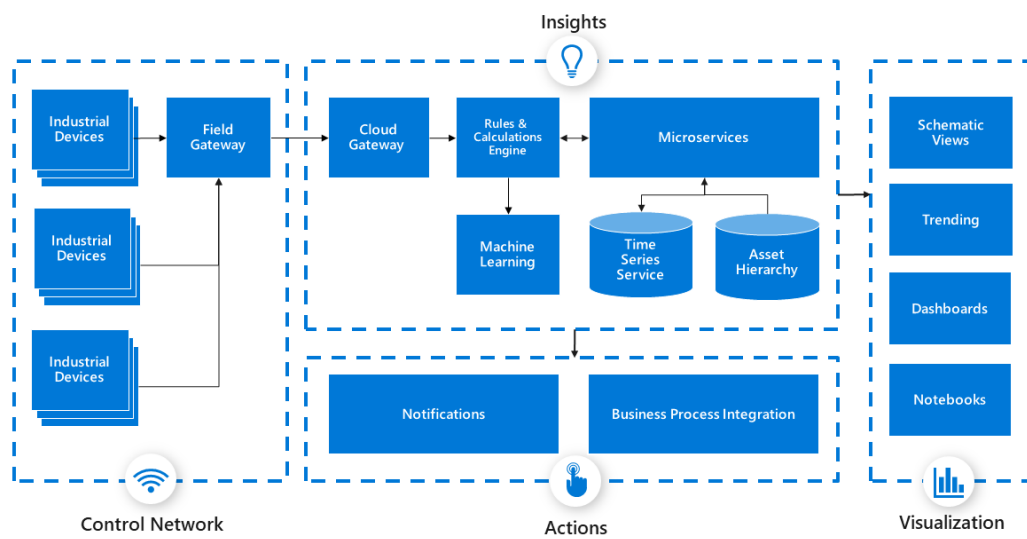


Kuva 1. Automaatiojärjestelmän järjestelmäkaavio periaatekuva yhdistettynä ISA-95 mukaiseen automaation pyramidiin.

Tehtaiden automaatio sisältää tavanomaisesti kaikki edellä olevassa järjestelmäkaaviossa (Kuva 1) kuvatut laitteet ohjauslogiikoista toiminnanohjaukseen.

1.3 Pilvipalvelut

Microsoft Azure esittää IIoT-analytiikka perusratkaisun (Kuva 2), jossa data virtaa teollisuuden laitteita pilveen yhdyskäytäviä pitkin. Pilvessä data käsitellään erilaisilla palveluilla ja ohjelmistoilla, joilla saadaan luotu erilaisia koonti ja historia datanäyttöjä tai ohjaavia toimenpiteitä ja ilmoituksia. [Microsoft 2021a; Shyam ym. 2017: 47–49]



Kuva 2. Microsoft Azure IIoT-arkkitehtuurikaavio. [Microsoft 2021a]

1.4 Tutkimusmenetelmät ja tavoitteet

Työn tarkoituksena on tutkia ja löytää kustannustehokas ja toimiva ratkaisu teollisuudessa olevien laitteiden ja Microsoft Azure-pilvialustan välille, sekä testata käytännössä ratkaisua soveltuvin osin käytännössä. Tutkimuksessa pyritään vastaamaan alla oleviin tutkimuskysymyksiin:

1. Mitä laiteratkaisuja löytyy Azure-pilvipalvelun ja logiikoiden väliseen kommunikointiin?
2. Kuinka mittausdata saadaan välitettyä Microsoft Azure-pilvipalveluun?

Tutkimuskysymyksiin pyritään löytämään ratkaisuja kirjallisuuskatsauksen avulla ja sen tuottamia ratkaisuja tutkitaan empiirisesti.

Empiiristä tutkimusta varten rakennetaan testausympäristö soveltuvin osin mitä ohjelmisto- ja laiteratkaisuja löytyy. Näiden avulla pyritään luomaan näkemys, kuinka data välittyy mittauksista Microsoft Azure-pilvipalveluun.

Työ koostuu neljästä pääkohdasta, jotka ovat:

- Tiedonvälitys ja kommunikointi protokollat
- Laitteiden valinta
- Microsoft Azure
- Testit

Tiedonvälitys ja kommunikointi protokollat sisältävät ainoastaan tämän työn kannalta oleelliset asiat, eikä näissä käydä tarkemmin kaikkia protokollia läpi.

Laitteiden valinnassa käydään läpi laitteiden valintaperusteita ja mitä laiteratkaisuja tutkimus tuotti. Tässä tutkimuksessa ei käydä kaikkia löytyneitä ratkaisuja läpi, työhön on valittu tutkimuksen kannalta oleelliset ratkaisut.

Tutkimuksessa käydään läpi Microsoft Azure-pilvipalvelua läpi siltä osin, miten sinne saa välitettyä dataa. Työssä ei käydä läpi tarkemmin tämän muita sovelluksia ja ominaisuuksia.

Testeissä pyritään käymään läpi vastaavilla laite- ja ohjelmistoratkaisuilla mitä teollisessa ympäristössäkin tullaan käyttämään. Laiteratkaisut saatetaan korvata erilaisilla simulaattoreilla kuitenkin todellisia laitteita mukaillen.

Tutkimuksessa ei syvennytä tarkemmin Microsoft Azure-pilvipalvelussa oleviin sovelluksiin. Tietoturva käydään läpi tässä tutkimuksessa vain yleisellä tasolla.

Työtä mitataan seuraavilla mittareilla:

- Työntehokkuus / Käytettävyys
- Kustannukset
- Luotettavuus

Työntehokkuus ja käytettävyys -mittarilla mitataan, kuinka löydetty ratkaisu skaalautuu erilaisiin kohteisiin ja tätä kautta myös työntehokkuuteen.

Kustannus -mittarilla mitataan, kuinka löydetty ratkaisu säästää kustannuksia projekteissa. Tässä työssä ei tulla esittämään hintoja tai laskelmia, mutta tätä kuitenkin mitataan tutkimuksessa ja sillä pyritään löytämään kustannustehokas ratkaisu.

Luotettavuus -mittarilla mitataan, kuinka löydetty ratkaisu on toimintavarma, sekä tietoturvallinen ratkaisu.

2 Kommunikointiprotokollat

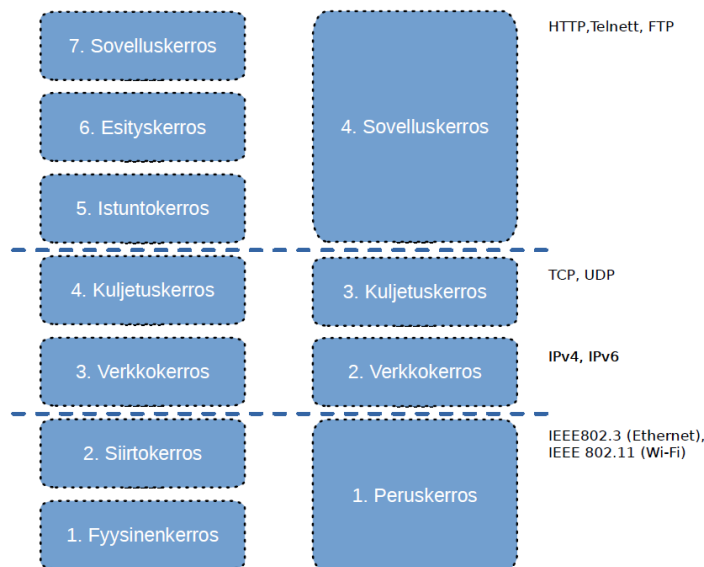
Kommunikointiprotokollia on monia riippuen käyttötarpeista ja niiden hyödyntämisestä. Tässä tutkimuksessa käydään osa niistä pintapuolisesti läpi, jotka ovat tämän tutkimuksen kannalta olennaisia.

2.1 Yleisesti

Sana protokolla tarkoittaa yhteyskäytäntöä. Laitteet ja tietokoneet kommunikoi- vat keskenään käyttämällä protokollaa. Ilman tätä ne eivät ymmärtäisivät toisi- aan. [Henrik Nygren 2018].

2.2 OSI- ja TCP-malli

OSI-malli eli Open System Interconnect model on ISO-organisaation (Internet Standard Organisation) kehittämä malli tiedonsiirtoprotokollien yhdistämisestä seitsemään kerrokseen. [Li ym., 2011] Mallia hyödynnetään mallintamaan tai ymmärtämään dataliikennettä verkossa. OSI-mallissa (Kuva 3) jokainen kerros palvelee sen ylempää olevaa kerrosta ja sitä puolestaan palvelee sen alapuolei- nen kerros.



Kuva 3. OSI- ja TCP/IP-malli mukautettu. [Li ym., 2011]

Seuraavaksi käydään pintapuoleisesti läpi OSI-mallin kaikki kerrokset.

Taso 7. Sovelluskerroksen avulla käyttäjä voi olla vuorovaikutuksessa sovelluksen tai verkon kanssa aina, kun käyttäjä päättää lukea viestejä, siirtää tiedostoja tai suorittaa muita verkkoon liittyviä tehtäviä. [Li ym., 2011]

Taso 6. Esityskerros kääntää tai muotoilee tiedot käyttäjälle näkyvään muotoon esimerkiksi kuvan pikseleiksi. [Li ym., 2011]

Taso 5. Istuntokerros määrittää, koordinoi ja lopettaa keskustelut sovellusten välillä. Sen tehtäviin kuuluu todentaminen ja yhteyden muodostaminen keskeytyksen jälkeen. Tämä kerros määrittää, kuinka kauan järjestelmä odottaa toisen sovelluksen vastausta. [Li ym., 2011]

Kerros 4. Kuljetuskerros on vastuussa tiedon siirtämisestä verkon yli. Se määrittää, kuinka paljon dataa lähetetään, mihin se lähetetään ja millä nopeudella. [Li ym., 2011]

Taso 3. Verkkokerroksen tehtävä on siirtää tietoja muihin verkkoihin. Se pakkaa tiedot oikeisiin verkko-osoitetietoihin, valitsemalla sopivat verkkoreitit ja välittämällä pakatut tiedot pinosta kuljetuskerrokseen. [Li ym., 2011]

Taso 2. Siirtokerros käsittelee tiedon siirtämistä verkon fyysiseen linkkiin ja sieltä pois. Lisäksi se vastaa datakehysten seurannasta lähettävän ja vastaanottavan laitteiston MAC-osoitteita käyttäen. [Li ym., 2011]

Taso 1. Fyysinen kerros siirtää dataa sähköisten rajapintojen avulla. Se määrittää, kuinka fyysiset yhteydet verkkoon muodostetaan ja kuinka bitit esitetään. [Li ym., 2011]

2.3 Teollisuuden IoT-protokollat

Kappaleessa 3.1 Esineiden internet käydään läpi IIoT ja sen toimintaperiaatetta. Kappaleessa käytiin myös läpi, etteivät IIoT ja IoT-laitteet eroa juurikaan keskenään toimintaperiaatteeltaan. IIoT-protokollat ovat melko hajanainen kokonaisuus. Erilaiset ryhmät ovat koonneet erilaisia omia standardeja IIoT-markkinoille, jotka voivat vaihdella protokolla pinojen laajuudeltaan tai kehykseltään. [Riedl & Yoon 2019]

Alla olevassa kuvassa (Kuva 4) on listattu lähteen [Riedl & Yoon 2019] mukaan yleisimmät IIoT käyttämät protokollapinot. Oranssit kohdat kuvastavat kuvassa protokollapinojen suojausprotokollakerroksia.

	Home Kit	XMPP	MQTT	HTTP	CoAP	LwM2M	Thread	ZigBee	Bluetooth	IPsec (VPN)
Application Layer	HAP	XMPP	MQTT	HTTP	CoAP	LwM2M	e.g. Weave, CoAP	ZigBee	Bluetooth	Any
Transport Layer		TLS			DTLS		-			
	TCP			UDP						
Network Layer	IPv4/IPv6						IPv6/6LoWPAN			IPsec
Link Layer	Any, e.g. IEEE 802.11						IEEE 802.15.4			
Physical Layer	Any, e.g. IEEE 802.11						IEEE 802.15.4			Any

Kuva 4. IIoT-protokollat ja yhteysstandardit TCP/IP-mallissa. [Riedl & Yoon 2019: 68]

Yllä olevasta kuvasta voidaan huomata, että IIoT-laitteet käyttävät IEEE 802-standardia, joista ehkä tunnetuimpia ovat IEEE 802.3 (Ethernet), IEEE 802.11 (WLAN) sekä IEEE 802.15.4 (WPAN). [Wikipedia 2021]

2.4 Teolliset viestintäprotokollat

Kappaleessa 1.2 Automaatiojärjestelmä käytiin läpi lyhykäisyydessään automaatiojärjestelmää, jossa kuvassa (Kuva 1) esitettiin tämän hierarkiatasot. Kaksi alinta tasoa kuvassa koostui Kenttälaiteet ja Ohjauslogiikat -tasoista, nämä tasot koostuvat ns. hajautetusta automaatiosta (DCS, Distributed Control System). Jotta kenttälaite- ja ohjauslogiikkatasot voisivat kommunikoida keskenään, on niihin kehitelty erilaisia protokollia.

Lähteessä [Yuqian Lu ym. 2020: 6] nämä protokollat oli kartoitettu hyvin taulukoon (Kuva 5). Tästä voidaan hyvin huomata, että nykyisin protokollat voidaan jakaa fyysisen kerroksen mukaan langattomiin, sarjaliikenneväyläisiin ja Ethernet pohjaisista kommunikointi protokolliin. Kuvasta käy hyvin myös ilmi, kuinka protokollat sijoittuvat kappaleessa 2.2 OSI- ja TCP-malli esiteltyyn OSI-malliin.

Protocol	Physical	Data link	Network	Transport	Session	Presentation	Application	Data Rate	No. of Devices
ControlNet [36]	RG-6 coaxial cables, 5Mbps	ControlNet CTDMA	ControlNet, 99 nodes	ControlNet	CIP protocol family			<5Mbps	99
DeviceNet [37]	CANbus with twisted pair cables, 1Mbps	CAN bus CSMA/NBA	DeviceNet, 64 nodes	DeviceNet	CIP protocol family			<0.5Mbps	64
Modbus-RTU or ASCII [38]	Serial cable, ex: RS-232, RS-485	Modbus	Modbus Map, 247 nodes	Modbus			Modbus client or server + interface	19.2kbps (default)	<247
PROFIBUS [39]	RS-485 cables, fiber optic cable or MBP	PROFIBUS Fieldbus data link	32 nodes, 126 with fiber optic cable	Not used	Not used	Not used	PROFIBUS DP	<12Mbps	<126
PROFINET [40]	Ethernet 10/100/1000 Mbps	Ethernet CSMA/CD	IP	TCP/UDP	Not used	Not used	PROFINET	<1000Mbps	>1000
Modbus-TCP/IP [41]	Ethernet 10/100/1000 Mbps	EtherNet	IP, 254 nodes/module	TCP port 502	Modbus TCP		Modbus client or server + interface	<1000Mbps	>1000
EtherNet/IP [42]	Ethernet 10/100/1000 Mbps	Ethernet CSMA/CD	IP	TCP/UDP	CIP protocol family			<1000Mbps	>1000
EtherCAT [42]	Ethernet 10/100/1000 Mbps	EtherNet w/EtherCAT slave&controller chip	IP with timing layer, up to 65535 nodes	TCP/UDP	EtherCAT			<1000Mbps	>1000
HART (Wired) [43]	Simultaneous hybrid analog & digital signaling, 4-20mA copper wiring	Mechanical/electrical connection transmits raw bitstream		Auto segmented transfer of large data sets, reliable stream transport, negotiated segment sizes	Not used	Not used	Command oriented, predefined data types and application procedures	1.2kbps	62
HART (Wireless) [44]	2.4GHz wireless, IEEE802.15.4 based radios, 10dbm transmission power	Secure and reliable, time synched TDMA/CSMA, frequency agile with ARQ	Power-optimized, redundant path, self-healing wireless mesh network		Not used	Not used		<250kbps	<30000

Kuva 5. Yleisimmät teollisuuden kommunikointi protokollat kartoitettuna OSI-malliin. [Yuqian Lu ym. 2020: 6]

2.5 OPC UA

OPC on lyhenne sanoista Open Platform Communication. Se on ohjelmistostandardi, joka mahdollistaa helpon liittymisen eri valmistajien automaatio-rajapintoihin. Sen takana toimii voittoa tavoittelematon OPC Foundation organisaatio, joka valvoo OPC teknisiä tietoja. Järjestöön kuuluu paljon jäseniä, laitteistoja ja ohjelmistoja. [OPC Connect 2022] OPC tuotteita on useita, joista tunnetuimpia ovat OPC Classic-arkkitehtuuriin kuuluva OPC DA (OPC Data Access) ja nykyisin OPC UA (OPC Unified Architecture) [OPC Foundation 2022a]

OPC UA on uusi arkkitehtuuri, joka julkaistiin vuonna 2008. Se on alustariippumaton palvelukeskeinen arkkitehtuuri, joka yhdistää kaikki OPC Classic-arkkitehtuuriin kuuluvat tuotteet yhdeksi laajennettavaksi kokonaisuudeksi. [OPC Foundation 2022b].

OPC UA-arkkitehtuurille on määritelty suunnittelu spesifikaatio tavoitteet, jotka ovat listattu ja suomennettu alla mukailen OPC Foundation määrittämiä [OPC Foundation 2022b]:

- Toiminnallinen vastaavuus
- Alustariippumattomuus
- Turvallisuus
- Laajennettavuus
- Skaalautuvuus

Toiminnallisella vastaavuudella tarkoitetaan, että OPC UA on pystyttävä toiminnallisesti vastaamaan vanhaa OPC Classicin toimintoja, mutta samalla parantamaan sen ominaisuuksia. [OPC Foundation 2022b]

Alustariippumattomuudella tarkoitetaan sitä, että ennen OPC Classic rajoittui Microsoft Windows käyttöjärjestelmiin, mutta nykyisin OPC UA on alusta riippumaton ja sitä pystyy käyttämään Microsoft Windows, Apple OSX tai mikä tahansa Linux-jakeluiden kanssa. [OPC Foundation 2022b]

Turvallisuudella tarkoitetaan, että OPC UA-arkkitehtuurissa on otettu huomioon tietoturvallisuus. [OPC Foundation 2022b] Tärkeimpiä ominaisuuksia ovat:

- Istunnonsalaus – jolla salataan viestit.
- Viestin allekirjoitus – jolla tarkistetaan vastaan otettujen viestien alkuperä ja eheys.
- Sekvensoidut paketit – viestien toistohyökkäyksille altistuminen eliminoidaan sekvensoinnilla
- Todennus – jokainen UA-asiakas ja -palvelin tunnustetaan varmenteilla, jotka ohjaavat sitä, mitkä sovellukset ja järjestelmät saavat muodostaa yhteyden toisiinsa.
- Käyttäjähallinta – Sovellukset voivat vaatia edellä mainittuja todennuksia ja rajoittaa ja parantaa näiden mukaan toimintoja.
- Tarkastus – Käyttäjän tai järjestelmän tekemät toiminnot kirjataan lokiin.

Skaalautuvuudella ja laajennettavuudella tarkoitetaan sitä, että OPC UA-arkkitehtuuri pystyy edeltäjänsä verrattuna esittämään datan hierarkkisena tietomallina aina yksinkertaisesta mallista monimutkaiseen. Laajennettavuudellaan OPC UA-arkkitehtuuri pystyy tulevaisuudessa laajentumaan uusiin siirtoprotokoliin, salauksiin, koodausstandardeihin tai sovelluspalveluihin ilman että se menettäisi yhteensopivuuden aikaisempien tuotteiden kanssa. [OPC Foundation 2022b]

Kappaleessa 6.2 OPC UA syvennyttään enemmän itse OPC UA-arkkitehtuurin käyttöön käytännössä ja sen konfiguroimiseen.

3 Tiedonvälitys

3.1 Esineiden internet

Esineiden internet eli IoT (Internet Of Things) ja teollinen internet eli IIoT (Industrial Internet Of Things), tarkoittaa yksinkertaisesti kaikkien esineiden yhdistämistä internettiin. Esineet ovat fyysisiä laitteita, joissa on sulautettu elektronikka ja ohjelmisto, ja ne ovat yhdistettynä internettiin. Tällaisia laitteita voivat olla esimerkiksi erilaiset älykellot, mittausanturit. [Gillis 2022]

IIoT-laitteet jakavat keräämänsä datan reunayhdyskäytävän tai jonkin muun reunalaitteen kautta pilvi- tai paikalliseen palveluun analysoitavaksi, niin kuin kuvissa (Kuva 10 ja Kuva 11). Laitteet voivat jakaa tietoa myös keskenään (Kuva 9). [Gillis 2022]

IoT ja IIoT eivät juurikaan eroa keskenään. Lähteessä [Devasia 2021] todetaan, että kaikki IIoT-laitteet ovat IoT-laitteita, mutta kaikki IoT-laitteet eivät ole IIoT-laitteita. Alla on listattu joitakin eroavaisuuksia. [Devasia 2021]

- Käyttösovellukset
- Turvallisuus ja luetettavuus
- Mittakaava

Käyttösovelluksissa nämä eroavat siten, että IIoT-laitteita on käytetty laajalti teollisissa käyttösovelluksissa ja IoT-laitteita kuluttajalaitteissa kuten ilmastointilaitteet, jääkaapit ja älykellot. [Devasia 2021]

Turvallisuudessa nämä eroavat sitten, että IIoT-laitteita käytetään paljon kriittisissä infrastruktuureissa, joista voi muodostua mittavat haitat. Tästä syystä näissä on panostettu enemmän laitteiden tietoturvaan. IoT-laitteissa tapahtuva tietoturva hyökkäys olisi häiriö eikä katastrofi, niin kuin IIoT-laitteissa. Luetettavuutta verrattaessa IIoT-laitteet kestävät pidempään, koska tuotantolaitokset ja laitteiden ovat rakennettu ajatellen niiden pidempää elinkaarta verrattuna IoT-laitteiden elinkaarta. [Devasia 2021]

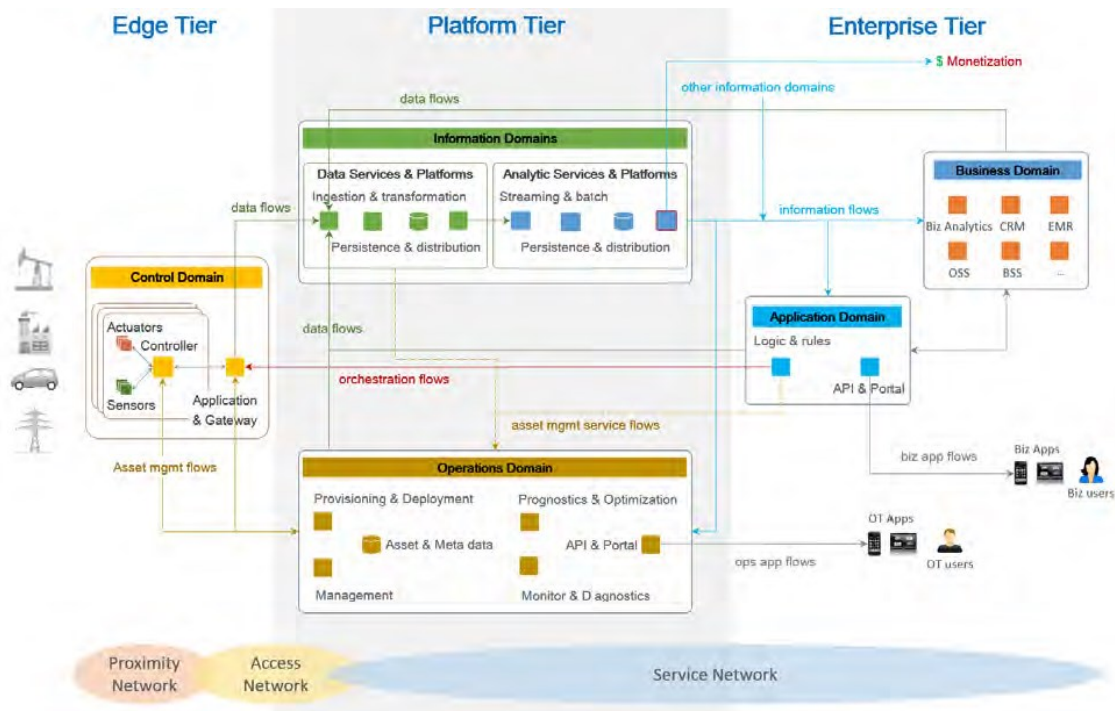
Mittakaavoja verrattaessa IoT- ja IIoT-laitteiden välillä käytössä voivat ne vaihdella suurestikin. IIoT-laitteita voi olla pienessä tehtaassa tai suuressa, mutta kuluttaja puolella IoT-laitteiden käyttö voivat olla massiivisia. [Devasia 2021]

IIoT-arkkitehtuuri malleja on monia erilaisia. Tässä työssä esitellään niistä kolme, jotka pohjautuvat IIC (Industrial Internet Consortium) esittelemiin malleihin. [Industrial internet consortium 2019: 40–48]

- Kolmikerroksinen arkkitehtuurikuvio (Three-tier-architecture pattern).
- Yhdyskäytävälitteinen reunayhteys ja hallinta arkkitehtuurimalli (Gateway-Mediated Edge Connectivity and Management architecture pattern).
- Kerrostettu tietoväylämalli (Layered Databus pattern).

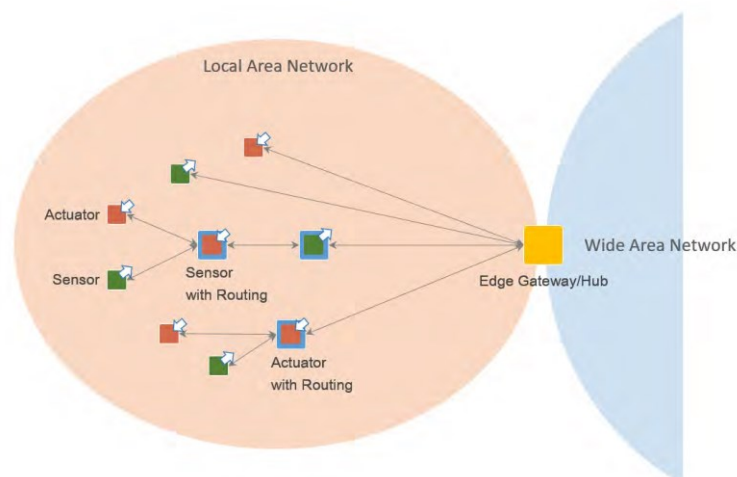
Kolmikerroksinen arkkitehtuurikuvio (Kuva 6) koostuu kolmesta tasosta, joita ovat: Reunataso (Edge Tier), Alustaso (Platform Tier) ja Yritystaso (Enterprise Tier).

Reunataso kerää tietoa laitteista ja antureista lähiverkkokäyttämällä. Alustataso vastaanottaa, käsittelee ja välittää reunatasolta tai yritystasolta tulevat viestit ja välittää ne eteenpäin seuraavalle tasolle. Yritystaso tarjoaa toimialue kohtaisia sovelluksia ja tukijärjestelmiä sekä tarjoaa rajapintoja loppukäyttäjille. Tämä voi myös antaa ohjaukskomentoja muille tasoille. [Industrial internet consortium 2019: 40–41]



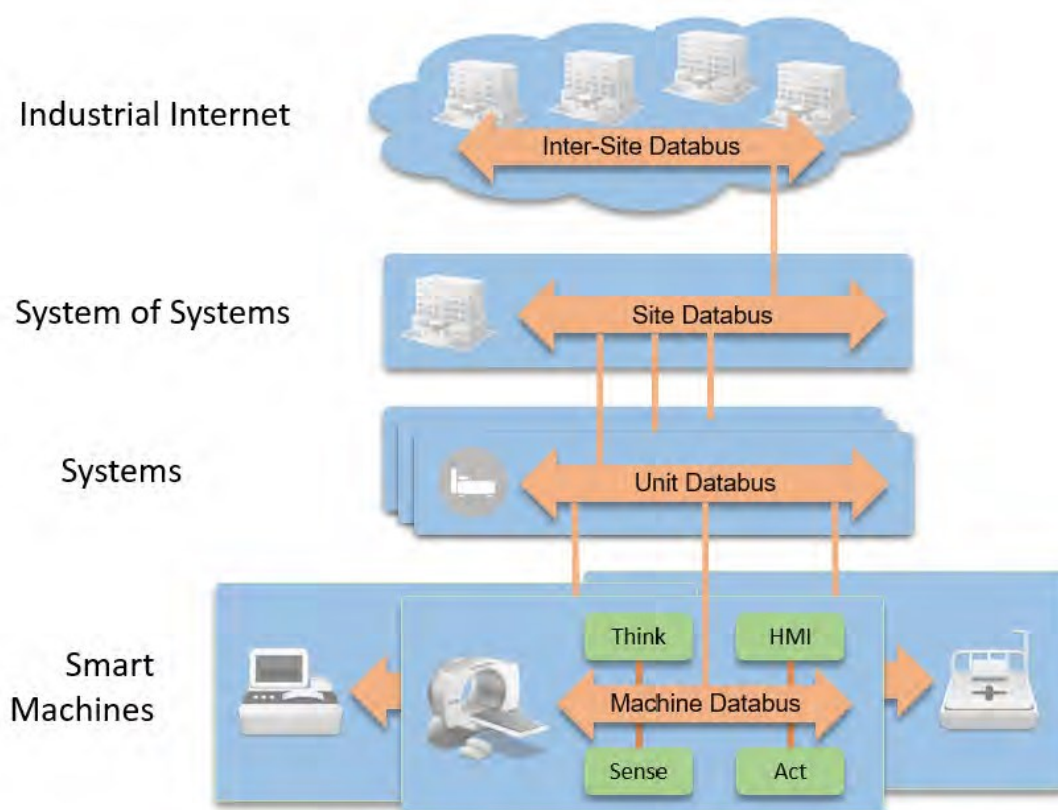
Kuva 6. Kolmikerroksinen arkkitehtuurimalli. [Industrial internet consortium 2019: 42]

Yhdyskäytävälitteinen reunayhteys ja hallinta arkkitehtuurimallissa (Kuva 7) kuvataan kuinka paikallinen verkko (LAN, Local Area Network) on kytketty reunayhdyskäytävään (Edge Gateway), joka siltaa yhteyden laajaan verkkoon (WAN, Wide Area Network). Tämän mallin päätarkoitus on yksinkertaistaa IIoT-järjestelmän monimutkaisuutta. [Industrial internet consortium 2019: 43–45]



Kuva 7. Yhdyskäytävälitteinen reunayhteys ja hallinta arkkitehtuurimalli. [Industrial internet consortium 2019: 44]

Kerrostettu tietoväylämalli (Kuva 8) koostuu neljästä tasosta. Alin taso koostuu älykkäissä laitteissa (Smart Machines), jotka käyttävät koneiden tietoväyliä (Machine Databus) paikalliseen ohjaukseen, automaatioon ja reaaliaikaiseen analytiikkaan. Järjestelmä taso (System) käyttää laite tietoväyliä (Unit Databus) valvontaan ja seurantaan, josta se käsittelee dataa oman konetietoväylän kautta järjestelmäjärjestelmä tasolle (System of System). Tämä taso käyttää toimipakkakohtaista väylää (Site Databus), joka hallitsee sen alla olevia järjestelmä- ja älylaitetasoja, joiden tietoväylät käyttävät yhteistä kaavaa (Schema) kommunikoidakseen päätelaitteiden kanssa. Ylimmällä tasolla on toimipaikkojen välinen tietoväylä (Inter-Site Databus), joka vastaanottaa ja lähettää dataa alaspäin. [Industrial internet consortium 2019: 45–47]



Kuva 8. Kerrostettu tietoväylämalli. [Industrial internet consortium 2019: 45]

Jos kerrostettua tietoväylämallia ajattelee automaation pyramidin kautta, jota on kuvattu johdannossa olevassa kuvassa (Kuva 1) vasemmalla puolella, voidaan huomata, että se koostuu paljon samoista tasoista.

3.2 Yhteysmallit

Kolme yleisintä IoT-laite yhteysmallia ovat laitteelta – laitteelle, laitteelta – pilveen ja laitteelta – yhdyskäytävään. Seuraavissa kuvissa (Kuva 9, Kuva 10 ja Kuva 11) havainnollistetaan, minkälaiset yhteyksien rakenteet ovat.

Device-to-Device eli toisin sanottu laitteiden välinen yhteysmalli (Kuva 9). Yhteys voidaan luoda kahden eri valmistajan kehittämän, mutta kuitenkin samaa protokollaa käyttävien laitteiden välille. Yhteyden hyvinä puolina voidaan pitää, että siinä on yksinkertainen rakenne ja laitteiden ylläpito ei vaadi ylläpito työtä. Tyypillinen esimerkki tällaisista ovat kuluttaja puolella esim. Zigbee-protokollaa käyttävät laitteet, joissa toinen laite on lamppu ja toinen katkaisija. Tämä yhteysmalli ei kykene kovin monimutkaisiin ohjauksiin sekä erivalmistajien protokollat voivat tuoda ristiriitoja yhteyksiä rakennettaessa. [Rose ym. 2015: 18–19]



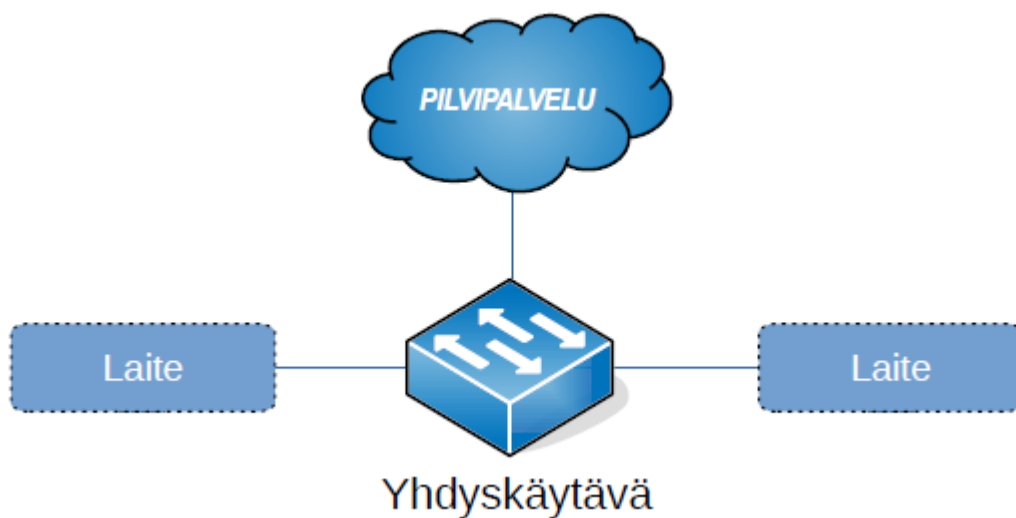
Kuva 9. IoT yhteysmalli laitteelta–laitteelle. Muokattu. [Rose ym. 2015: 19]

Device-to-Cloud eli laitteelta pilveen välinen yhteysmalli (Kuva 10). Tässä yhteys mallissa laite muodostaa yhteyden suoraan pilvipalveluun internet yhteyden avulla langallisesti tai langattomasti. Pilvipalvelu yhteys mahdollistaa laitteiden monipuolisemman käytön, koska pilvipalvelut voivat sisältää monenlaisia datan käsittely ja analysointi ohjelmia. Yhteysmallin haittapuolina voidaan pitää, että laitteiden tai pilvipalveluiden palveluiden muutokset saattavat tehdä kokolaitteis-
tosta käyttökeltottoman [Rose ym. 2015: 19–20]



Kuva 10. IoT yhteysmalli laitteelta–pilveen. Muokattu. [Rose ym. 2015: 20]

Device-to-Gateway eli laitteelta yhdyskäytävälle välinen yhteysmalli (Kuva 11). Tässä yhteysmallissa laite muodostaa yhteyden ensiksi paikalliseen yhdyskäytävään, josta yhdyskäytävä välittää datan pilvipalveluun. Yhdyskäytävä laitteiden ja pilvipalveluiden välissä mahdollistaa data analysoinnin ja pakkaamisen enne sen vientiä pilvipalveluun. Lisäksi yhdyskäytävän kommunikointi protokollat ovat monipuolisemmat, joka mahdollistaa, ettei laiteiden tarvitse olla välttämättä IoT-laitteita. [Rose ym. 2015: 20–21]



Kuva 11. IoT yhteysmalli laitteelta–yhdyskäytävään. Muokattu. [Rose ym. 2015: 21]

3.3 IoT-Gateway

IoT-Gateway (IoT-yhdyskäytävä), joka tunnetaan myös nimellä IoT Edge Gateway (IoT-reunayhdyskäytävä). Sen keskeisimmät ominaisuudet ovat kerätä dataa erilaisilta antureilta, päätelaitteilta ja järjestelmistä, analysointia, yhdistämistä tai tiedonsiirtämistä varten pilvipalveluun. [Open Automation Software 2022; Shyam ym. 2017: 50]

Kuvassa (Kuva 2) on kuvattu Azure-pilvipalvelu arkkitehtuuria, missä IoT-yhdyskäytävä on kuvattuna nimellä Field Gateway. Tästä voidaan huomata, että tyyppillisesti IoT-yhdyskäytävä sijoittuu mahdollisimman lähelle anturi ja päätelaite rajapintaa.

IoT-yhdyskäytävät voivat olla hyvinkin monipuolisia niiden sisältämien erilaisten käyttöjärjestelmien ja avoimien lähdekoodien, sekä erilaisten protokolla tukien johdosta. Tämä mahdollistaa, ettei itse päätelaitteiden ja antureiden tarvitse sisältää älyä itsestään. Sen implementoiminen mahdollistaa suurien määrien datan keruun ja analysoinnin, joka olisin paljon suurempi, mikäli anturit lähettäisivät datan suoraan pilveen. Tämän takia se on ideaalinen järjestelmiin missä ei ennestään laitteet tuo IoT-protokollia.

Lähteessä [Shyam ym. 2017] puhutaan, että IoT-yhdyskäytävä tulee tukea seuraavia alla olevia ominaisuuksia [Shyam ym. 2017: 51]:

- Paikallinen yhdistettävyys
- Verkojen ja protokollien siltaus
- Paikallinen datankäsittely
- Laite- ja omaisuudenhallinta ja hallintapiste
- Paikkasidonnainen päätös- ja sovelluslogiikka

Yhdyskäytävien ominaisuuksiin perehdytään lisää kappaleessa 5 Laitteiden valinta, jossa käydään tämän tutkimuksen kannalta tärkeimpiä ominaisuuksia ja valinta kriteereitä.

3.4 Node-RED

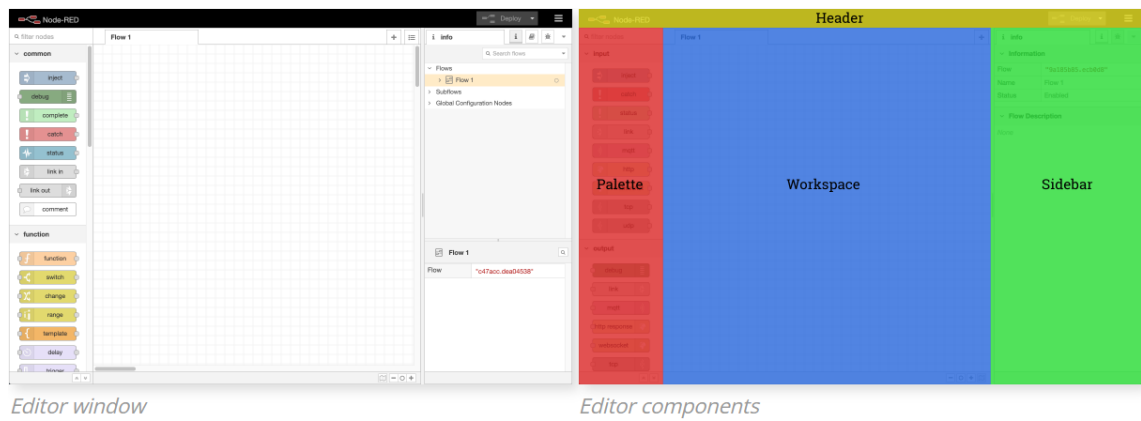
Node-RED on IBM:n kehittämä vuonna 2013 avoimen lähdekoodin visuaalinen ohjelmointityökalu [OpenJS Foundation & Contributors 2022]. Se on alhaisen koodin (low-code) ohjelmistoympäristö tapahtumapohjaisille sovelluksille. Tämä mahdollistaa nopean koodin ja vähäisellä koodin kirjoitus määrällä. [IBM 2020]

Node-RED hyödyntää Node.js sovelluksen kirjastoa, joka on alusta riippumaton avoimen lähdekoodin JavaScript-ohjelmisto kieli. [OpenJS Foundation & Contributors 2022; Node-RED 2022a]

Node-RED ohjelmointi muistuttaa FBD (Function block diagram) -ohjelmointi kieltä, jota käytetään ohjelmoimaan teollisuuden PLC:tä (Programmable logic controller).

Node-RED-ohjelmointi käyttöliittymään avautuu selaimeen, jota kautta itse ohjelmointi tapahtuu. Alla olevassa kuvassa (Kuva 12) on esitelty käyttöliittymän perusnäkyminen kuvan vasemmalla puolella. Kuvan oikealla puolella on esitelty kuva missä kuvataan erivärein mistä käyttöliittymä tarkemmin koostuu.

Koodi luodaan käyttämällä Flow(virtauksia) (Kuva 12, sininen alue), joihin työkaluriviltä (Kuva 12, punainen alue) voidaan raahata erilaisia niin sanottuja nodeja(solmuja), joita yhdistellään johdottamalla niitä keskenään. Sivun valikossa (Kuva 12, vihreä alue) voidaan katsoa ja konfiguroida noodeja, tarkastella lokeja toiminnoista tai virheistä sekä lukea käyttöohjeita noodeista. [Node-RED 2022a]



Kuva 12. Node-RED-työkalun perusnäkymä. [Node-RED 2022b]

Kappaleessa 6.4 Node-RED syvennyttään enemmän itse Node-RED-ohjelmointiin käytännössä.

4 Microsoft Azure

4.1 Azure Historia

Microsoft Azure on yhdysvaltalaisen yrityksen Microsoftin kehittämä pilvialusta. Se julkaistiin vuonna 2008 nimellä Windows Azure [Microsoft 2008]. Vuonna 2014 Microsoft muutti nimen Microsoft Azure, millä nimellä se tunnetaan tänäkin päivänä [Microsoft 2014].

Azuren palvelut sisältävät erilaisia IaaS- (Infrastructure as a Service) ja PaaS-tyyppistä (Platform as a Service) ja Software as a Service (SaaS) tyyppisiä palveluita. [Microsoft 2022a; Microsoft 2022b; Microsoft 2022c]

4.2 Azure palvelutyypit

Azure kuvaa verkkosivuilla kolme yleisintä (Kuva 13) palvelutyypit, sekä mitä ne pitävät sisällään.



Kuva 13. Microsoft Azuren palvelut [Microsoft 2022c]

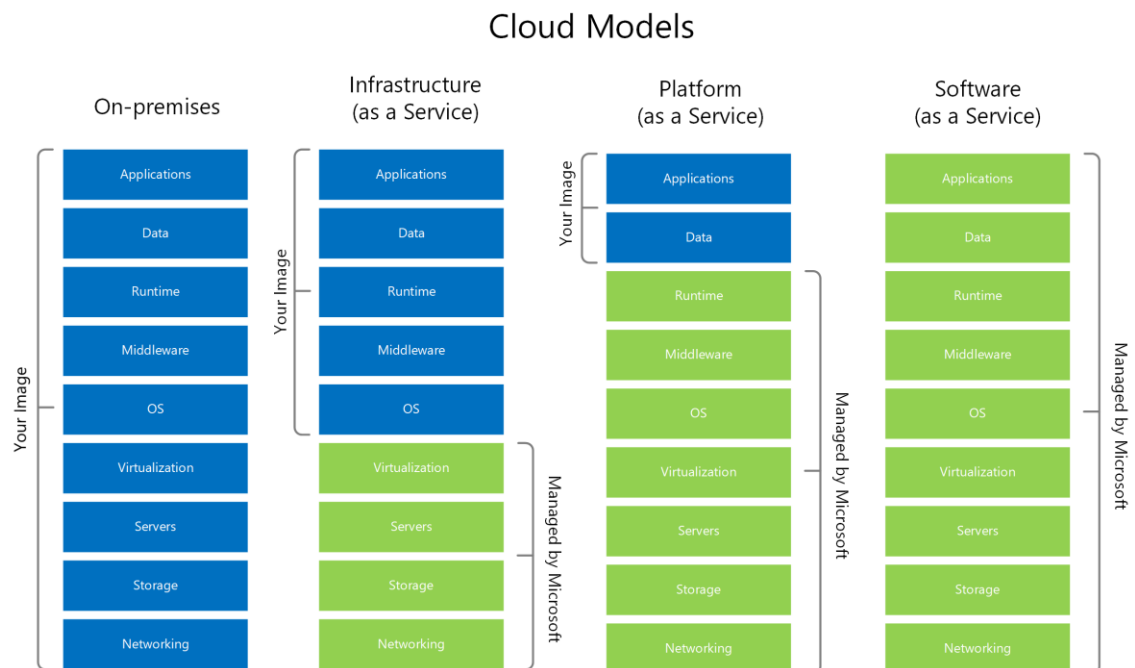
IaaS on suomennettu infrastruktuurin palvelu, joka sisältää kaiken infrastruktuurin verkkopalveluiden tukemiseen, sekä tallennus-, verkko- ja sovelluspalvelimet. [Microsoft 2022a] Tyypillisiä IaaS-palveluita ovat virtuaalikone- ja konttipalvelut.

PaaS on suomennettuna alustapalvelut, se sisältää samat palvelut kuin edellä mainittu IaaS, mutta lisäksi tämä sisältää erilaiset kehittämis- ja data-analytiikka työkalut sekä kehitysympäristöt. [Microsoft 2022b]

SaaS on suomennettua ohjelmistopalvelut, mahdollistaa ohjelmistojen kehittämisen ja hallinnan ilman omaa infraa. [Microsoft 2022c] Tyypillisiä SaaS-palveluita ovat erilaiset sähköposti, asiakas ja toiminnanohjaus palvelut.

Microsoft on listannut sivuillaan useita palveluita, jotka sisältävät mm. tietokantoja, tallennusta, data analytiikkaa ja verkkopalveluita. Palvelut ovat usein data-käyttö kohtaisia, jolloin käytettävästä laitteiden tehosta ja datanmäärästä laskutetaan sen perusteella mitä on käytetty.

Kuvassa (Kuva 14) on selvennetty kuinka pilvipalveluiden vastuut jakautuvat kussakin palvelussa. Sinisellä on kuvattu palvelun tilaajan vastuut ja vihreällä palvelun toimittajan vastuut.



Kuva 14. Microsoft Azure-pilvipalvelun ylläpitovastuut [Microsoft 2021b]

4.3 IoT Hub

Azure IoT Hub suomennettuna IoT keskitin. Se on Azure-pilvipalvelussa pyörivä palvelu, jonka tarkoituksena on toimia viestikeskuksena IoT-sovellusten ja siihen liitettyjen laitteiden välillä [Microsoft 2022d]. Kuvassa (Kuva 2), jossa esitellään Azuren arkkitehtuuri malli IIoT-verkosta on tämä kuvattu nimellä Cloud gateway.

IoT Hub tukee seuraavia laitepuolen viestintäprotokollia MQTT, MQTT WebSocket, AMQP, AMQP WebSockets ja HTTPS [Microsoft 2008]. Azure-pilvipalvelu on turvallinen ratkaisu, koska jokaiselle IoT Hub:lla Azuressa on identiteettirekisteri. Tähän tallennetaan siihen yhdistettyjen laitteiden tiedot, jonka perusteella sallitaan laitteen yhdistää siihen. [Microsoft 2022e]

IoT Hub kuvataan usein tärkeimpänä palveluna Microsoft PaaS tyyppisissä palveluissa. Se mahdollistaa tiedon lähettämisen, sekä vastaanottamisen useiden IoT-laitteiden ja Azure-pilvipalvelun välillä. [Microsoft 2022d]

5 Laitteiden valinta

5.1 Valinta perusteet

Tässä kappaleessa käydään läpi valinta kriteerit, joiden mukaan laitevalintoja lähdetään valitsemaan. Laitteiden kriteerit koostuvat pääosin laitteiden teknisistä tiedoista. Seuraavissa luvuissa käydään läpi valittujen kriteereihin vaikuttavat tekijät.

5.1.1 Asennustapa

Laitteita tullaan integroimaan erilaisiin automaatiojärjestelmiin, joten keskeiset asiat asennustavassa tulee olemaan käyttöjännite, kiinnitys ja koko. Pääsääntöisesti voidaan sanoa, että automaatio DCS-taso ja kenttälaiteet toimivat 24VDC jännitteellä, jolloin tämä jännitetaso on määritetty myös kriteeriksi laitteelle. Laitteistot sijaitsevat automaatiokeskuksissa, jotka ovat kalustettu melko ahtaalle olemassa olevissa järjestelmissä. Tästä syystä laitteen olisi hyvä olla DIN-kisko kiinnitteinen, sekä mahdollisimman pieni kooltaan, jotta laite ei veisi tilaa keskuksessa.

5.1.2 Yhteystavat

Ohjelmoitavat logiikat kommunikoivat pääsääntöisesti Ethernetilla ylemmille tasoille. Kohteita voi olla myös sellaisia laitoksia, joihin ei välttämättä ole internetiin pääsyä, jolloin datan siirtäminen voi olla hankalaa. Tämänlaisissa kohteissa voidaan vaatia joustavimpia yhteystapoja, jolloin langatonverkko tai mobiiliyhteys voi tulla kysymykseen.

5.1.3 Kustannukset

Laitteivalinnoissa pyritään löytämään mahdollisimman kustannustehokas laiteratkaisu. Valinnoissa kiinnitetään myös huomioita laitteiden lisensointiin ja näiden hallintaohjelmien hintoihin. Tavoitteena olisi löytää laite, josta ei kertyisi muita palvelukustannuksia hankintahinnan lisäksi.

5.1.4 Alustat

Alustoja on monenlaisia, joita voi olla laitevalmistajan omia käyttöjärjestelmiä, Windows tai Linux. Tavoitteena on löytää Linux pohjainen ratkaisu, jolloin voidaan puhua avoimen lähdekoodin ratkaisusta. Tämä mahdollistaa paremman ja joustavamman ympäristön laitteelle.

5.1.5 Tietoturva

Vaikka tutkimuksessa ei ole käsitelty tarkemmin tietoturvaa, otettiin tämä kuitenkin laitehankinnoissa huomioon.

5.1.6 Takuu ja saatavuus

Automaatiolaitteiden elinkaari on pitkä, ja tästä syystä pitää ottaa myös laitteiden valinnoissa huomioon. Laitteilla täytyy olla takuu ja sen saatavuus täytyy olla myös taattu.

5.2 Vertailuun valitut laitteet

Tutkimus tuotti paljon laiteratkaisuja, joista muutamat on esitelty seuraavissa kappaleissa. Laitteet voidaan jakaa karkeasti kolmeen erilaiseen ryhmään.

- PC pohjaiset ratkaisut, jotka ovat liitännöiltään, kooltaan ja suoritusarvoiltaan vastaavia mitä teollisuustietokoneet ovat.
- Sulautettu järjestelmät, jotka ovat suunniteltu kyseiseen tarkoitukseen, mutta näihin ei ole mahdollista asentaa kolmansien osapuolien ohjelmia.
- Linux pohjaiset tai avoimet ratkaisut, jotka mahdollistavat kolmansien osapuolien ohjelmat.

Seuraavissa kappaleissa käydään löydettyjä laitteita lyhykäisyydessään läpi.

5.2.1 Hewlett Packard

Hewlett Packard GL10 IoT Gateway kuuluu löydetyistä laitteista ryhmään PC pohjaiset ratkaisut. Se on suunniteltu reunayhdyslaitteeksi ja se omaa lisäksi monipuoliset liitännät, sekä laajennuspaikat lisäkorteille. GL10:stä voidaankin puhua teollisuus PC:stä sen liitäntöjen ja tehojen suhteen. GL10 on kuitenkin tutkimuksen tuottamista ratkaisuista kooltaan ja tehon tarpeeltaan, sekä hinnan suhteen suurimpia. Hewlett Packard on maailman laajuisesti tunnettu yhtiö, joten takuu asiat varmasti hoituvat myös yrityksen puolesta. Hyvinä puolina voidaan GL10:ssa pitää, että se on käyttöjärjestelmältään Linux tai Windows pohjainen, jolloin sen käyttö on skaalautuvaa erilaisiin käyttötarkoituksiin. Kuvassa (Kuva 15) näkyy GL10 reunayhdyslaite liitäntöineen.



Kuva 15. HPE GL10 IoT Gateway. [Hewlett Packard Enterprise 2022]

Lisää laitteen tiedoista selviää Liitteestä 1.

5.2.2 Dell

Dell Embedded Box PC3000 kuuluu myös löydetyistä laitteista ryhmään PC-pohjaiset ratkaisut. Siinä on myös monipuoliset liitännät, sekä laajennuspaikat lisäkorteille. Myös tästäkin voidaan puhua niin sanottuna teollisuus PC:stä. PC3000 on tutkimuksen tuottamista ratkaisuista kooltaan ja tehon tarpeeltaan sekä hinnaltaan suurin. Dell on maailman laajuinen tunnettu yritys, joka lupaa tuotteelle vuoden peruslaitetakuu. Sen käyttöjärjestelmä tuet ovat myös laajat, aina Linuxista Windowsiin, joka mahdollistaa laitteen monipuoliset käyttökohdet. Kuvassa (Kuva 16) näkyy Dell Embedded Box PC3000 reunayhdyslaite liitännäineen.

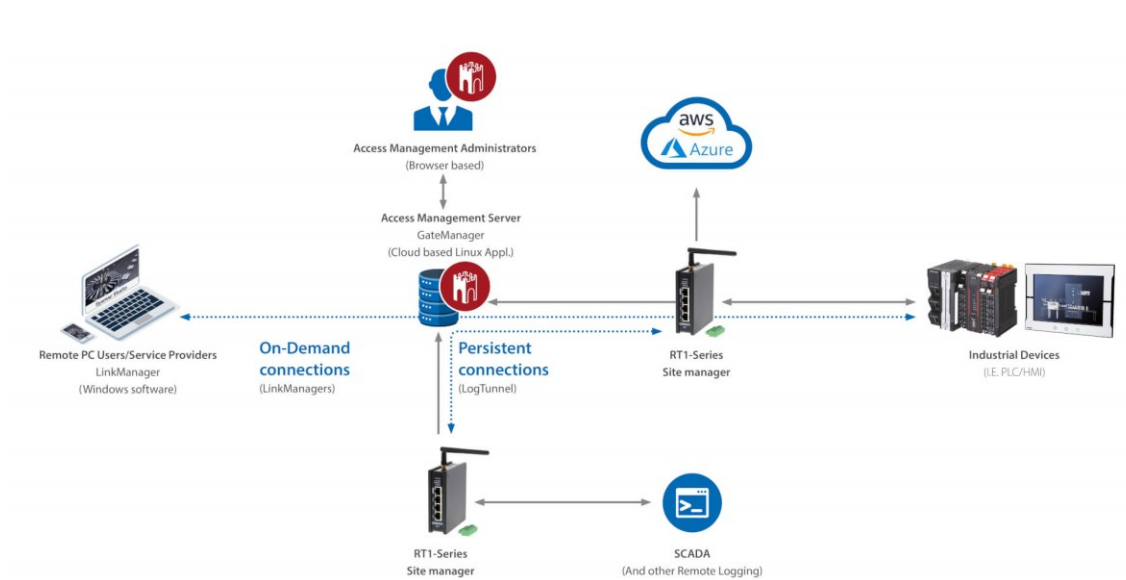


Kuva 16. Dell Embedded Box PC3000. [Dell 2022]

Lisää laitteen tiedoista selviää Liitteestä 1.

5.2.3 Omron

Omron RT1 SiteManager kuuluu löydetyistä laitteista ryhmään Sulautetut järjestelmät. Siinä on kaksi digitaalista tulo ja lähtökanavaa, joilla on mahdollista vastaanottaa tietoa tai ohjata laitetta sen kautta. Se on kompaktin kokoinen laite ja sitä on saatavana kolmea eri versiota: Mobiili 4G, LAN ja WiFi. Omron on myös maailman laajuinen tunnettu yritys, joka puolestaan takaa tuotteiden saatavuuden ja takuu käsittelyn. Käyttöjärjestelmänä laitteessa toimii Omron Sitemanager, jolla pystytään lukemaan sujuvasti ainakin Omronin omia teollisuuden laitteita. RT1 Sitemanageria hallitaan erillisellä Access Management Serverillä, johon otetaan puolestaan kiinni tietokoneella. Kuvassa (Kuva 17) on kuvattu minkälainen, sen perusverkon rakenne on.



Kuva 17. Omron Sitemanager verkkotopologia. [Omron 2022]

Lisää laitteen tiedoista selviää Liitteestä 1.

5.2.4 Phoenix

Phoenix IoT gateway kuuluu myös löydettyistä laitteista ryhmään Sulautetut järjestelmät. Se on kompakti laite, joka oli myös tutkimuksen pienimpiä. Phoenix on maailmanlaajuinen tunnettu yritys, joka puolestaan takaa tuotteiden saatavuuden ja takuu käsittelyn. Tutkimuksen edetessä kuitenkin huomattiin, että ilmeisesti kyseinen laite ei pysty kommunikoimaan Azuren pilvipalvelun kanssa vaan ainoastaan Phoenixin oman ProfiCloud kanssa, jossa pystyi käsittelemään dataa ja piirtämään erilaisia kuvaajia. Kuvassa (Kuva 18) on esitelty Phoenix IoT gateway.



Kuva 18. Phoenix IoT gateway. [Phoenix contact 2022]

Lisää laitteen tiedoista selviää Liitteestä 1.

5.2.5 Siemens

Siemens-laitteista löytyi kolme mielenkiintoista vaihtoehtoa reunayhdyskäytäväksi ja tästä syystä tutkimusta myös tarkennettiin näihin kolmeen malliin. Siemensin Simatic IOT2050 ja IOT2040 laitteet kuuluvat ryhmään Linux pohjaiset tai avoimet ratkaisut. IOT2050 Löytyy kahta eri versiota ja ne eroavat toisistaan vain keskusmuistin määrässä. IOT2050 ja IOT2040 eroavat toisistaan suorittimen, muistin, liitäntöjen ja koon osalta. Näissä laitteissa on myös monipuoliset liitännät teollisuuden sovelluksiin. Kumpikin laite versio käyttää käyttöjärjestelmänä Siemens Industrial OS, joka on Linux pohjainen käyttöjärjestelmä. Siemens on maailman laajuisesti tunnettu yritys, joka takaa tuotteiden saatavuuden ja takuu käsittelyn. Kuvassa on esitelty vasemmalla Siemens Simatic IOT2040 ja oikealla Siemens Simatic IOT2050.

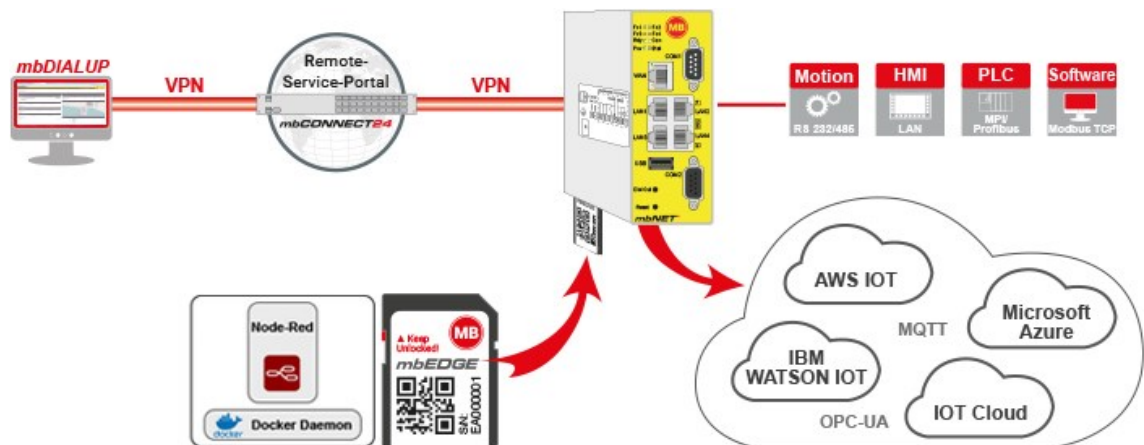


Kuva 19. Siemens Simatic IOT2040 ja IOT2050. [Siemens 2022] [Siemens 2022]

Lisää laitteen tiedoista selviää Liitteestä 1.

5.2.6 mbNET

mbNET MDH810-MHD871 oli tutkimuksen tekijälle uusi laite valmistaja, joka herätti kiinnostusta. MbNeT laiteratkaisut kuuluvat ryhmään Linux pohjaiset tai avoimet ratkaisut. Laitteet ovat pelkkiä reitittimiä, kunnes niihin lisätään mbNET linux pohjainen ohjelmisto SD-kortilla, jolloin se muuttaa reitittimet IoT-reunayhdyskäytäväksi. Laitteita saa kolmea eri versiota: WLAN, Mobiili 4G ja LAN. Laite on kompaktin kokoinen. mbNET on saksalainen laitevalmistaja, mutta sitä edustaa suomessa Sarlin Oy, joka puolestaan mahdollistaisi takuu käsittelyn Suomessa. Laitteen saatavuus jäi tutkimuksessa vähän epävarmaksi. Käyttöjärjestelmänä laitteissa toimii erillisellä muistikortilla lisättävä mbEDGE, joka on Linux pohjainen käyttöjärjestelmä. Laitteeseen saa asennettua Node-RED ohjelmiston ja Docker & container-kontti ohjelmiston, joka mahdollistaa monipuolisen kolmansien osapuolien ohjelmistojen asennuksen. Kuvassa (Kuva 20) on esitelty kyseisen reunayhdyskäytävän verkkotopologia, sekä itse laite.



Kuva 20. mbNET reititin + mbEDGE verkkotopologia. [MB connect line 2020]

Lisää laitteen tiedoista selviää Liitteestä 1.

5.2.7 Moxa

Moxa:lta löytyi kaksi mielenkiintoista vaihtoehtoa reunayhdyskäytäväksi, josta syystä tutkimusta myös tarkennettiin näihin kahteen malliin. Moxa:n UC-3100 ja UC-8100 laitteet kuuluvat ryhmään Linux pohjaiset tai avoimet ratkaisut. UC-3100 malleja löytyy montaa eri versiota ja ne eroavat toisistaan niissä olevien liitännöiden, sekä bluetooth tai Wi-fi-yhteyksien suhteen. UC-8100 malleja löytyy myös montaa eri versiota, ja ne vaihtelevat niissä olevien liitännöiden määrän, muistin ja prosessorin koon suhteen. Moxa reunayhdyskäytävissä toimii käyttöjärjestelmänä Linux, joka mahdollistaa monipuolisen ja joustavan käytön. Moxa on maailman laajuisesti tunnettu yhtiö ja se lupaa reunayhdyskäytävillään 5 vuoden takuun.



Kuva 21. Moxa UC-3100 ja UC-8100. [Moxa 2022]

Lisää laitteen tiedoista selviää Liitteestä 1.

5.2.8 Welotec

Welotec EG500 ja EG600 oli tutkimuksen tekijälle uusi laitevalmistaja. Welotec laiteratkaisut kuuluvat ryhmään Linux pohjaiset tai avoimet ratkaisut. Laitteita löytyi valmistajalta kahta erilaista, jotka eroavat keskenään prosessorin ja fyysisen koon, sekä liitännöiden määrän suhteen. Welotec käyttöjärjestelmänä pyörii Docker ohjelmisto, joka mahdollistaa monipuolisen kolmansien osapuolien ohjelmistojen asennuksen. Welotec on saksalainen valmistaja ja edustusta Suomessa ei suoraan löydetty. Takuu asia jäi myös avoimeksi, kuinka pitkä takuu laitteille myönnetään. Kuvassa (Kuva 22) on esitelty vasemmalla puolella EG500 ja oikealla puolella EG600.



Kuva 22. Welotec EG500 ja EG600. [Welotec 2022]

Lisää laitteen tiedoista selviää Liitteestä 1.

5.3 Laitteiden vertailu

5.3.1 Tulosten koonti

Kappaleissa 5.1.1–5.1.6 esiteltiin läpi reunayhdyslaitteet, joita tutkimus tuotti. Saatuja tuloksia jatko käsiteltiin sijoittamalla laitteet taulukkoon (Liite 1) ja keräämällä niihin laitteiden tiedot tutkimuksen kappaleen 5.1 Valinta perusteet esitellyjen määrityksiensä perusteella laitteiden datalehdistä, jotka kerättiin laitevalmistajien sivuilta.

Laitteiden tiedot sijoitettiin taulukkoon seuraavalla tavalla. Jos joitakin laitteiden tietoja ei löydetty, merkittiin kyseinen kohta kysymysmerkillä. Laitteiden virta ja hinta-arvio tiedot jouduttiin joissakin tapauksissa laskemaan. Tämä tarkoitti sitä, että kyseistä tietoa ei saatu, mutta sen pystyi laskemaan muiden tietojen perusteella auki. Tyypilliset tapaukset olivat valuuttamuunnokset ja virran lasku ohmin lailla tehosta ja jännitteestä. Laskujen tulokset eivät olleet välttämättä tarkkoja, mutta niiden tarkkuus riitti tulosten analysoitaviksi. Joillakin laitteilla ilmeni eri malleja, näitä ei erikseen lähdetty erittelemään taulukkoon, vaan kyseisestä laite mallista kerättiin kootusti kaikki ominaisuudet kyseisen laitteen tietoihin. Laitteiden hinnat kerättiin suuntaa antavasti internetistä löytyvien kauppojen sivuilta. Hinnat eivät ole välttämättä tarkkoja, mutta näistä pystyttiin tarkastelemaan kuitenkin tuloksia riittävän tarkasti.

5.3.2 Tulosten analysointi

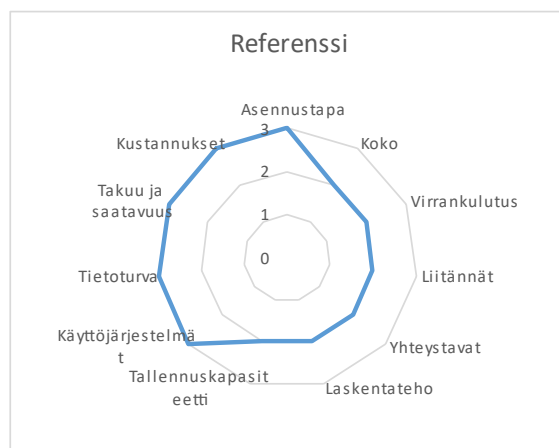
Tuloksia analysoitiin luomalla laitetaulukon laitteista valmistaja kohtaiset säteittäiset kaaviot (Liite 2), joka mahdollisti paremman visuaalisen laitteiden vertailun. Laitteiden tiedot olivat pääsääntöisesti teksti muotoisia ja tulosten esittely säteittäisellä kaaviolla oli numeerista. Tästä syystä laitteiden ominaisuudet piti arvioida numeerisesti. Arviointi taulukossa päädyttiin käyttämään arvoja 0–3, jonka tarkkuus oli laitearvioon riittävä kuvaamaan laitteen kyseistä ominaisuutta. Taulukossa (Taulukko 1) on kuvattu, kuinka laitteiden ominaisuudet arvioitiin.

Taulukko 1. Laitteiden ominaisuuksien arviointi taulukko.

Ominaisuus	Arviointi asteikko			
	0	1	2	3
Asennustapa	Tieto ei löytynyt	Jos asennus tyyppi ei ollut	Jos asennus tyyppi oli vähintään esim. rack, VESA tai Seinä	Jos asennus tyyppi oli vähintään DIN ja Seinä asenteinen
Koko	Tieto ei löytynyt	Jos pisin mitta oli yli arvon 150mm	Jos pisin mitta ei ollut yli arvon 120 mm ja yhtä kuin tai alle 150	Jos pisin mitta ei ollut yli arvon 120 mm
Virrankulutus	Tieto ei löytynyt	Jos virta oli yli 2A	Jos virta oli yli 1A ja alle 2A	Jos virta oli alle 1A
Liitännät	Tieto ei löytynyt	Jos liitäntöjä ei ollut kuin ethernet ja 1xusb	Jos liitäntöjä oli ethernet ja sarjaliikenne	Jos liitäntöjä oli ethernet, sarjaliikenne, IO ja lisäksi muita
Yhteystavat	Tieto ei löytynyt	Jos yhteyksiä ei ollut kuin ethernet	Jos yhteyksiä oli Ethernet, WLAN	Jos yhteyksiä oli Ethernet, WLAN sekä LTE
Laskentateho	Tieto ei löytynyt	Jos prosessorin teho alle 1000 MHz	Jos prosessorin teho alle yli 1000 MHz ja alle 1200 MHz	Jos prosessorin teho yli 1200 MHz
Tallennuskapasiteetti	Tieto ei löytynyt	Jos tallennus kapasiteettia ei ole	Jos tallennus voidaan tehdä SD-kortille	Jos tallennus on mSata, SSD, HDD tai yli 32Gb
Käyttöjärjestelmät	Tieto ei löytynyt	Jos käyttöjärjestelmä on laitevalmistajan oma	Jos käyttöjärjestelmä on Docker-ohjema pohjainen	Jos käyttöjärjestelmä on Linux ja lisäksi jokin muu
Tietoturva	Tieto ei löytynyt	Ei ole mainintaa	-	Mainittu
Takuu ja saatavuus	Tieto ei löytynyt	Jos takuuta ei ole tai laitevalmistaja on pieni yhtiö	Jos laite edustus suomessa, mutta ei ole tunnettu laitevalmistaja	Jos suuri kansainvälinen yhtiö ja takuu
Kustannukset	Tieto ei löytynyt	Jos hinta yli 900€	Jos hinta yli 400 € ja alle 900 €	Jos alle 400 €

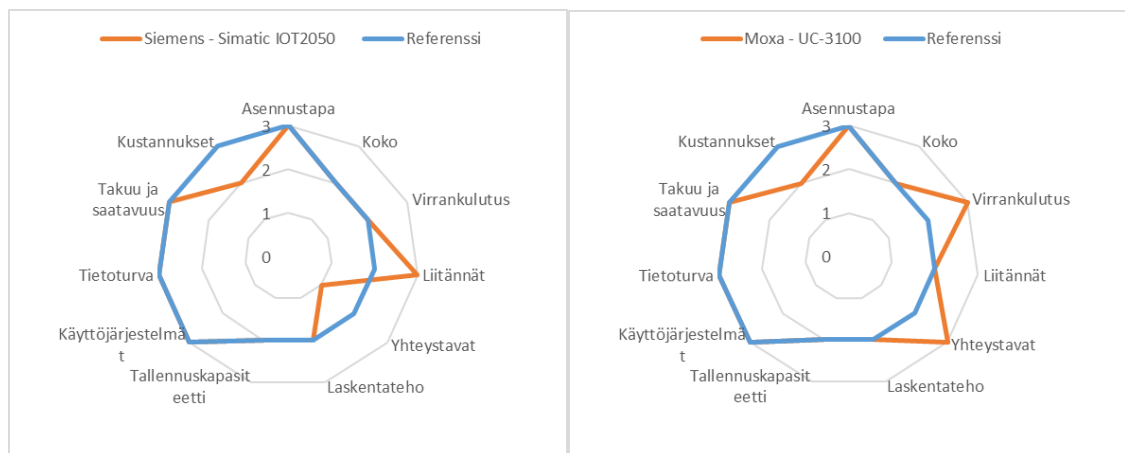
Taulukon arviointi skaalautui laitteiden tietojen mukaan, jolloin saatiin arvio löydettyjen laitteiden keskinäisistä tiedoista.

Laitevalinnoissa oli tavoitteena painottaa seuraavia ominaisuuksia: käyttöjärjestelmä, tietoturva, takuu ja saatavuus, kustannukset ja asennustapa. Näiden painotusten perusteella saatiin tuotettua referenssikäyrä (Kuva 23).



Kuva 23. Laittevertailu referenssikäyrä

Kun referenssikäyrää verrattiin laiteista laadittuihin käyriin (Liite 2) huomattiin, että parhaimmat vastaavuudet olivat Siemens IOT2050, sekä Moxa UC-3100 ja UC-8100 laitteilla. Kuvassa (Kuva 24) on kuvattu vertailun tuloksia, jossa vasemmalla on verrattu Siemens Simatic IOT2050 ja oikealla Moxa UC-3100. Kuvasta voidaan huomata, että kustannukset sekä yhteystavat eivät täysin kohtaa referenssikaavion kanssa, mutta muut ominaisuudet puolestaan vastasivat referenssi kaavion kanssa.



Kuva 24. Laittevertailu referenssikaavioon

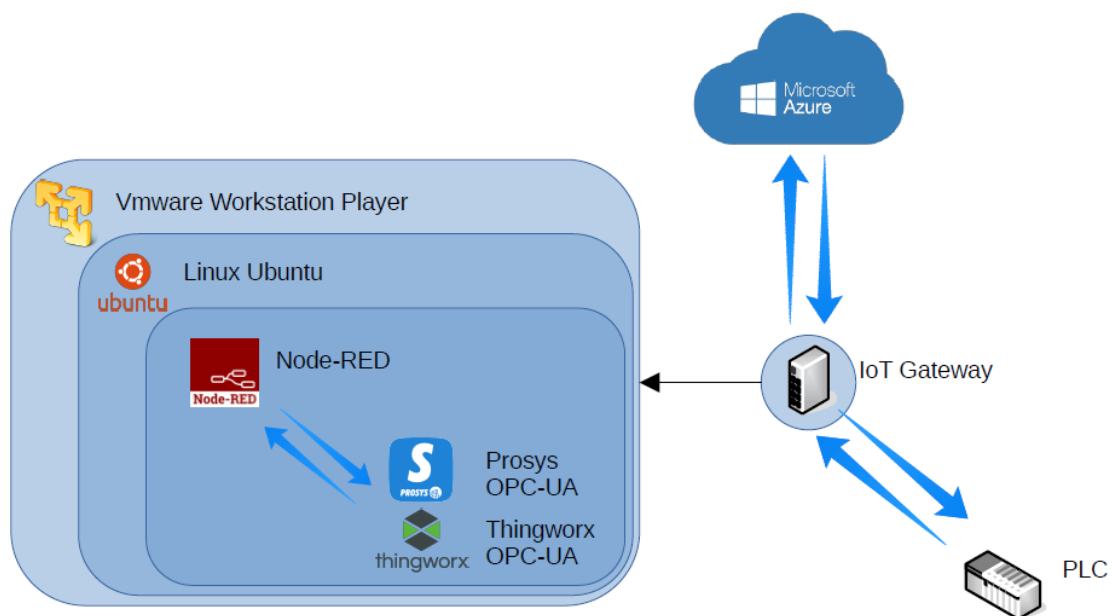
6 Testien kuvaus ja testit

6.1 Testausympäristö

Kappaleessa 5 Laitteiden valinta esiteltyjen laiteratkaisujen pohjalta luotiin testiympäristö. Laitteiksi valittiin Linux-ympäristössä toimivat reunayhdyskäytävät niiden monipuolisuuden vuoksi.

Reunayhdyskäytävää ei päästy testaamaan oikeilla laitekoonpanolla, mutta tätä varten luotiin virtuaalinen Linux Ubuntu-ympäristö, käyttämällä VMware Workstation Player-ohjelmaa. Virtuaaliympäristö simuloi testattavaa laitekoonpanoa.

Kuvassa (Kuva 25) on kuvattu simulointiympäristön ohjelmistorakenne, joka pyrittiin rakentamaan vastaavanlaiseksi kuin IoT Gateway-ohjelmistoja. Logiikoiden eli toisinsanottuna PLC:n simulointi tapahtui OPC UA serverissä itsessään.



Kuva 25. Testausympäristö.

Seuraavissa kappaleissa tullaan käymään ohjelmistoja tarkemmin läpi ja esitellään, kuinka ohjelmistot ohjelmoitiin ja kuinka niitä käytettiin testeissä.

6.2 OPC UA

6.2.1 Valintojen perusteet testauksiin

Testeihin valittiin kaksi eri OPC UA ohjelmistoa ThingWorx Kepware Edge ja Prosys OPC UA Server OPC UA serverit. Kyseiset ohjelmistot valittiin testiin niiden simulointi mahdollisuuden takia eli näillä voidaan simuloida mittauksia, jotka tulisivat oikeassa automaatioympäristössä logiikalta. Lisäksi testeihin valittiin kaksi eri ohjelmistoa, koska Prosys OPC UA Simulation server-ohjelmisto saatiin toimimaan vain numeerisilla tunnisteilla ja ThingWorx Kepware Edge saatiin toimimaan merkkijonotunnisteella. Kolmas syy kahden ohjelman valintaan oli, että tällä mahdollistettiin laitteiston joustavuus OPC UA ohjelmiston valintaan.

6.2.2 Prosys OPC UA Simulation server konfigurointi

Prosys OPC UA Simulation server toimii graafisella käyttöliittymällä. Kun tämä oli saatu asennettua onnistuneesti, näytti ohjelman päänäkymä kuvan (Kuva 26) kaltaiselta.

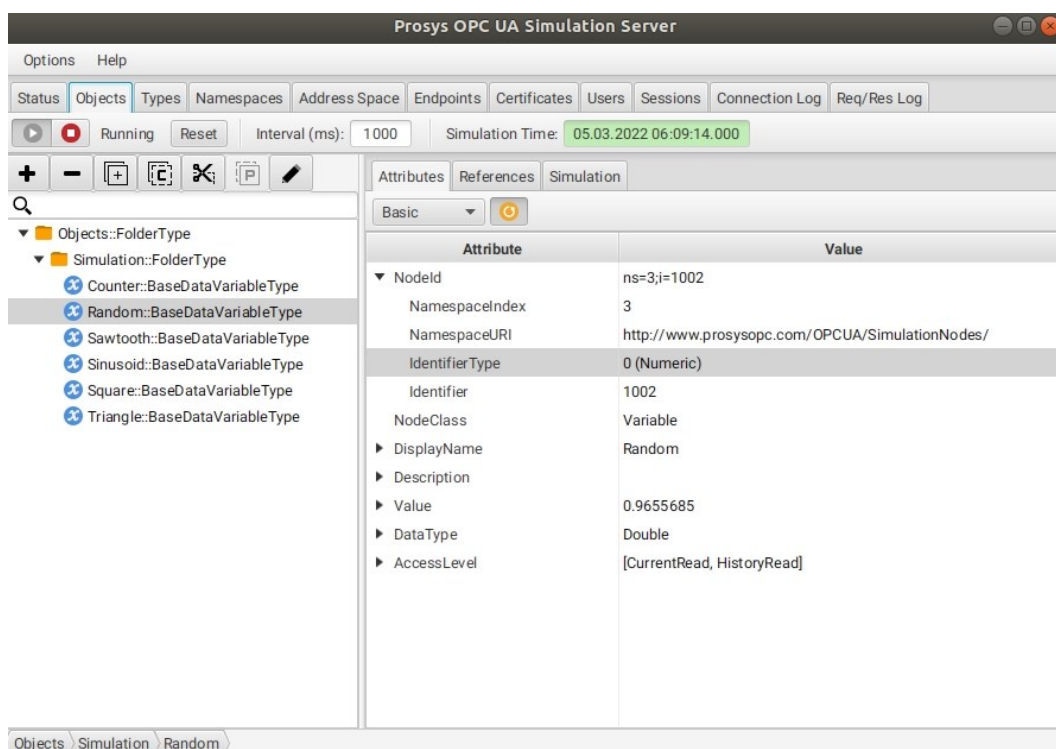


Kuva 26. Prosys Simulation server status-näkymä.

Tästä ikkunasta nähdään kaksi tärkeää asiaa Server status, että tämä on päällä eli running-tilassa ja yhteysosoite, jolla otetaan serveriin yhteyttä.

Seuraavaksi määriteltiin mittaukset, jotka tulisivat todellisuudessa PLC:itä. Näitä on tähän serveriin valmiiksi luotuna kuusi erilaista erilaisilla käyrillä tuotettuja simulointi arvoja (Kuva 27). Kuvasta voidaan myös nähdä, että nämä muuttujat käyttävät jo enemmän mainitsemaa numeerista tunnistetta (Identifier Type).

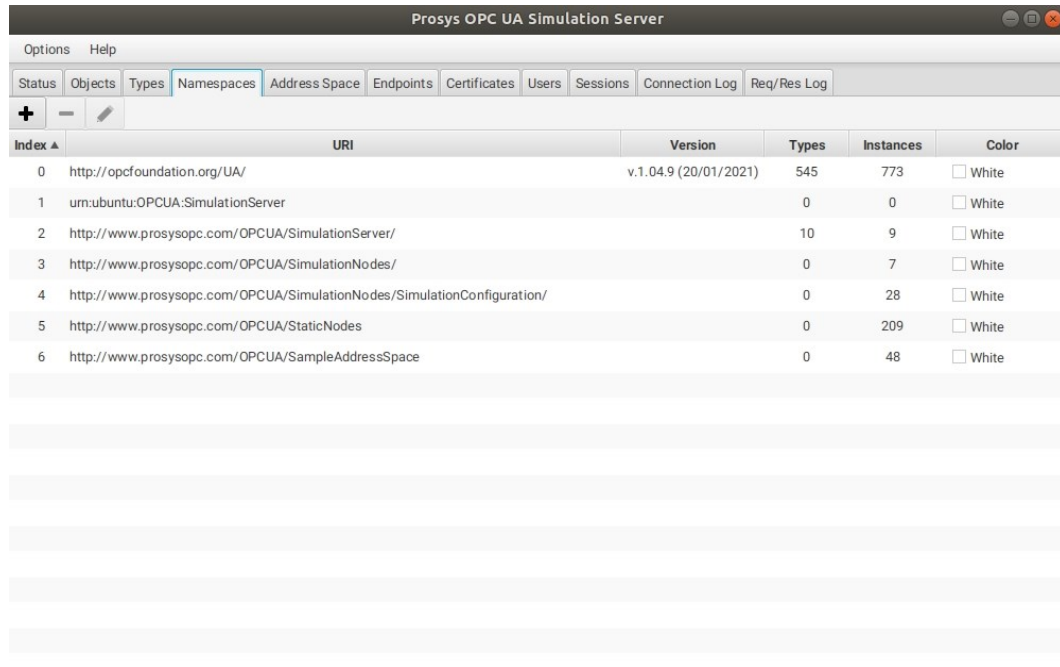
Datan luvun kannalta ovat NodeId parametrit tärkeät. Näistä kolme tärkeintä ovat NamespaceIndex, IdentifierType ja Identifier. Nämä kertovat, että mistä lähteestä data luetaan, minkälaiset sen tunnistetiedot ovat.



Kuva 27. Prosys Simulation server object-näkymä.

Namespace-välilehdellä (Kuva 28) määriteltiin niin sanottu lähde, mistä enemmän mainitut mittaukset tulevat. Mittauksia voi olla useammasta PLC:stä, jolloin Namespace-välilehdellä hallitaan mistä mikäkin mittaustieto tulee. Simulointi

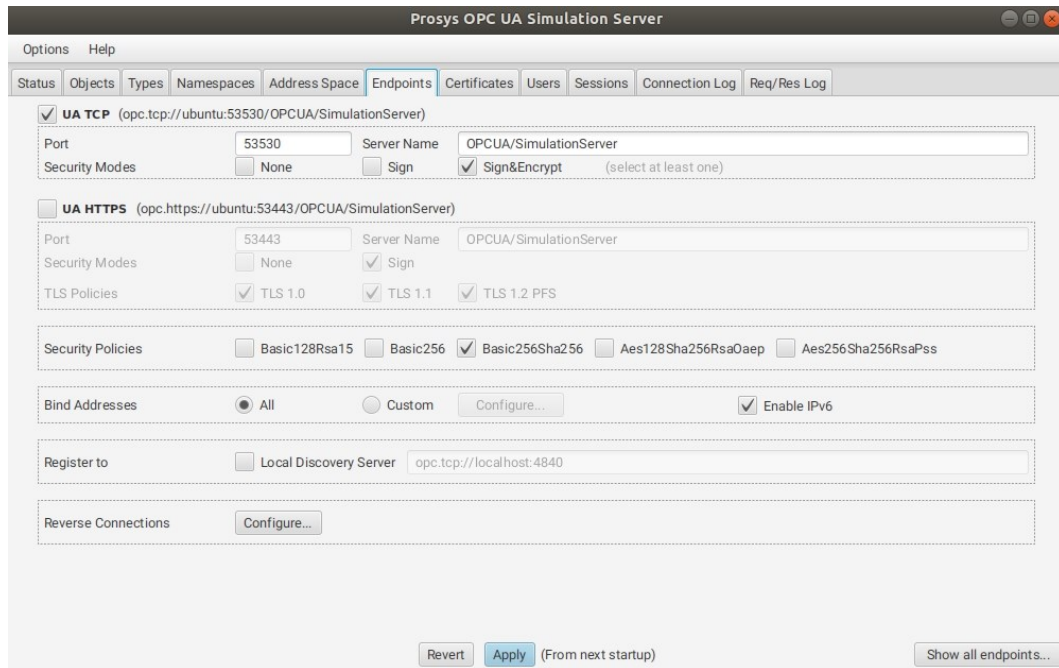
ympäristössä käytettiin simuloituja mittauksia, jolloin ne tulivat lähteestä 3. Tällöin se merkittiin mittauksen Namespaceindex:iin ns=3, niin kuin kuvassa (Kuva 27).



Index ▲	URI	Version	Types	Instances	Color
0	http://opcfoundation.org/UA/	v.1.04.9 (20/01/2021)	545	773	<input type="checkbox"/> White
1	urn:ubuntu:OPCUA:SimulationServer		0	0	<input type="checkbox"/> White
2	http://www.prosysopc.com/OPCUA/SimulationServer/		10	9	<input type="checkbox"/> White
3	http://www.prosysopc.com/OPCUA/SimulationNodes/		0	7	<input type="checkbox"/> White
4	http://www.prosysopc.com/OPCUA/SimulationNodes/SimulationConfiguration/		0	28	<input type="checkbox"/> White
5	http://www.prosysopc.com/OPCUA/StaticNodes		0	209	<input type="checkbox"/> White
6	http://www.prosysopc.com/OPCUA/SampleAddressSpace		0	48	<input type="checkbox"/> White

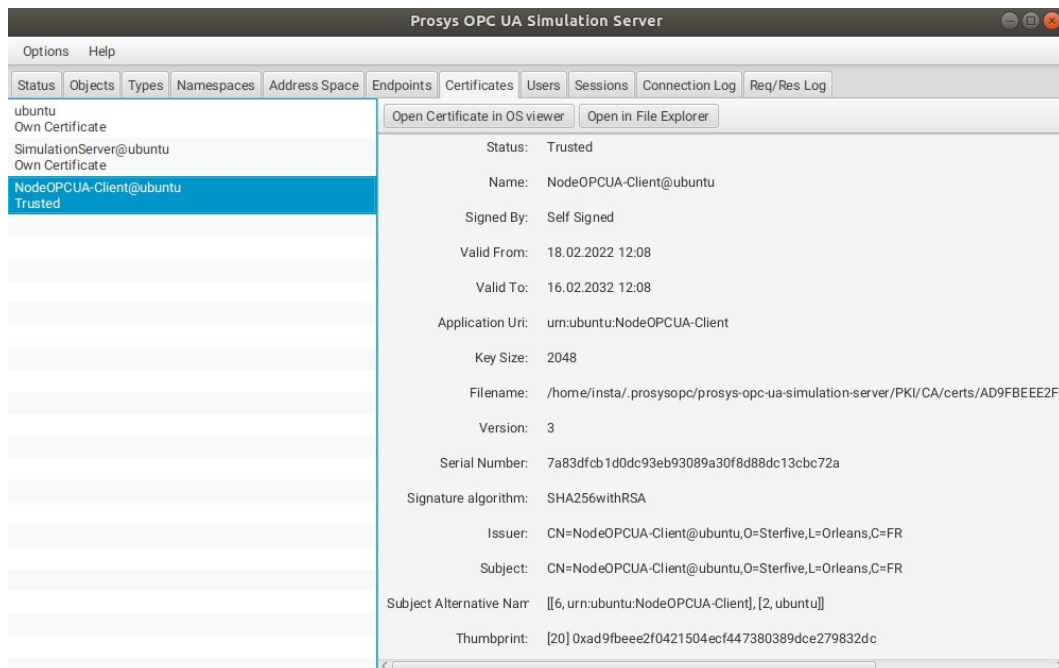
Kuva 28. Prosys Simulation server Namespaces-näkymä.

Endpoint-näkymässä (Kuva 29) määriteltiin serverin nimi ja kommunikointiportti. Lisäksi valikosta määriteltiin kommunikoinnin suojaus määrittelyjä esimerkiksi käyttäjätunnistuksen sallinta ja salauksen tyyppin valinta. Testiympäristössä käytettiin käyttäjätunnistusta ja salausta (Sing&Encrypt). Salauksena käytettiin Basic256Sha256-salausta. OPC UA ja Node-RED olivat asennettuina samaan laitteeseen, jolloin näiden välisen viestinnän salaus oli melko turhaa. Salaus tarvitsee enemmän laskentatehoa, kuin ilman salausta oleva viestintä. Tässä työssä haluttiin kuitenkin ottaa nämä tietoturvallisuuden kannalta mukaan.



Kuva 29. Prosys Simulation server Endpoints-näkymä.

Users-valikosta luotiin serverille käyttäjätunnus ja salasana. Lisäksi Node-RED-yhteyden sertifikaatti hyväksyttiin Certificates-valikosta (Kuva 30).



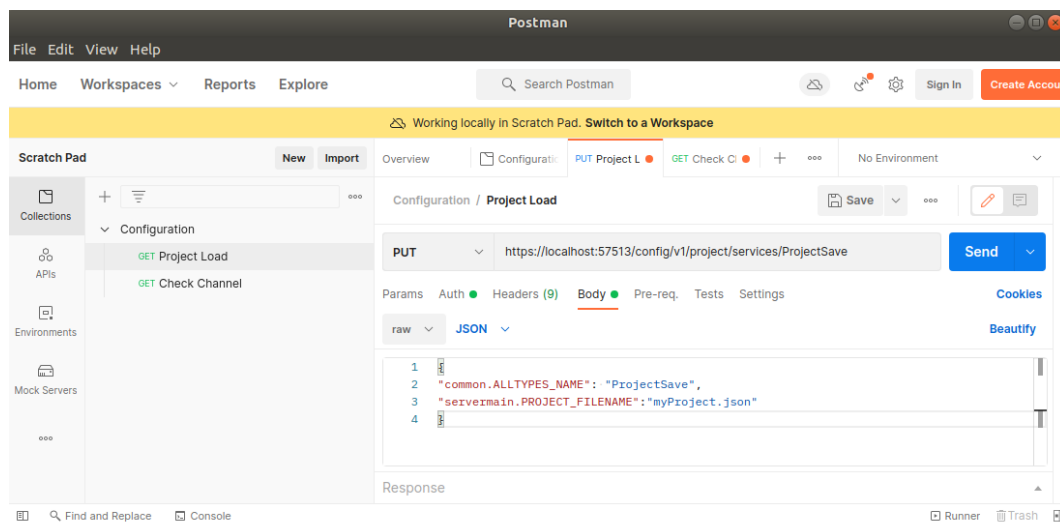
Kuva 30. Prosys Simulation server Endpoints Certificates-näkymä.

Sertifikaatti piti hyväksyä vasemmalla olevasta listassa, jotta yhteys saatiin muodostettua kyseiseen serveriin.

6.2.3 ThingWorx Kepware Edge konfigurointi

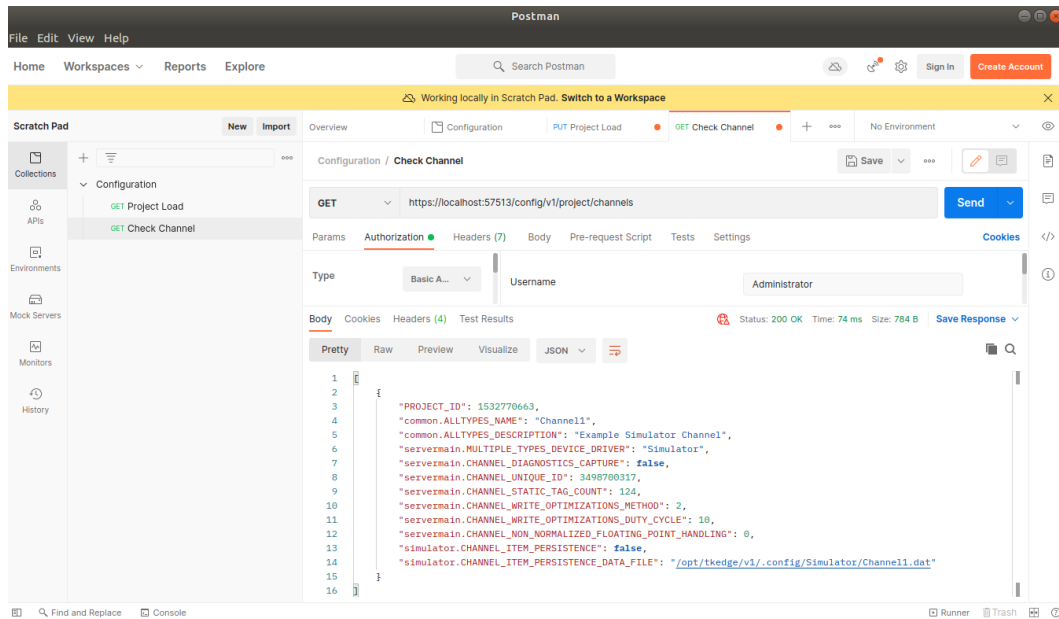
ThingWorx Kepware Edge serverin asennus ja konfigurointi tapahtui käyttämällä API-sovellusliittymää käyttäen. Tätä varten jouduttiin asentamaan myös koneelle Postman-sovellus, joka on tarkoitettu REST-rajapintojen testaukseen. Lisäksi koneelle asennettiin Unifiend Automation UaExpert client-sovellus, jolla konfiguroitua serveriä pystyi graafisesti tutkimaan paremmin.

Kuvassa (Kuva 31) ladattiin ensiksi projekti serverille hyödyntäen PUT-komentoa.



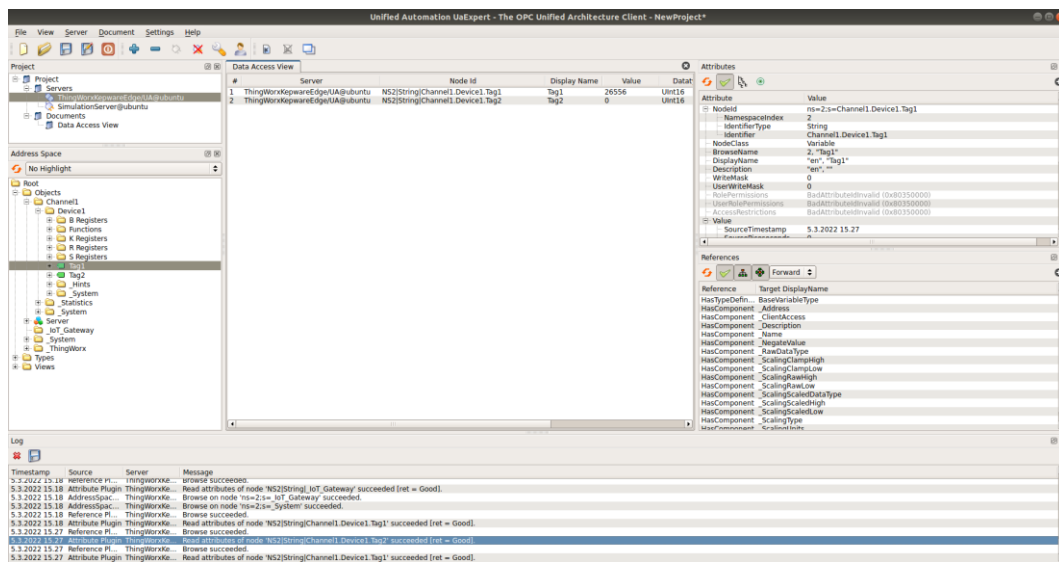
Kuva 31. Postman-sovelluksella projektin lataus.

Tämän jälkeen sovelluksella lähetettiin GET-komento (Kuva 32), jolla saatiin vastaus serveriltä, että projekti on ladattu serverille.



Kuva 32. Postman-sovelluksella projektin tarkastus.

Kun serveri oli saatu pystytettyä, pystyttiin siihen ottamaan yhteyttä UaExpert client-sovelluksella ja tarkastelemaan sen mittauksia (Kuva 33).

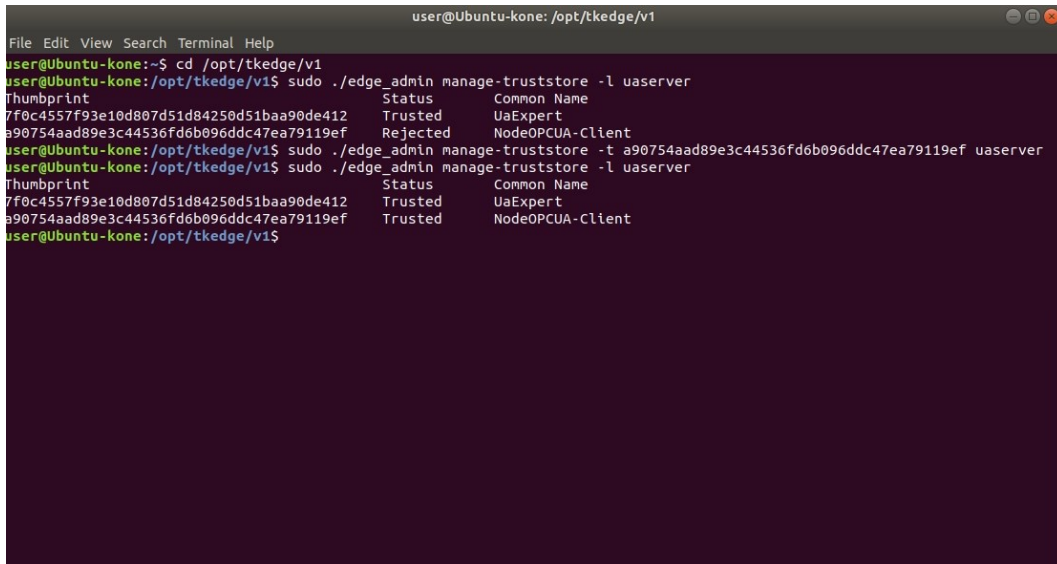


Kuva 33. UaExpert client-sovelluksen näkymä

Kappaleessa 6.2.2 Prosys OPC UA Simulation server, käytiin läpi tarkemmin läpi Nodeld tietoja. Kuten kuvasta (Kuva 33) voidaan huomata, löytyi myös tässä NodeID tiedot. Kyseisessä serverissä käytettiin merkkijonotunnistetta, joka mahdollisti mittauksien paremman tunnistuksen nimen perusteella.

Niin kuin Prosys OPC UA serverissä, käytettiin tässäkin valmiiksi luotuja simuloitua mittausta, sekä tietoturvallisuudessa käyttäjätunnistusta ja Basic256Sha256-salausta.

Käyttäjätunnus luotiin jo ohjelman asennus vaiheessa, mutta sertifikaatin asennus Node-RED-yhteydelle tehtiin käyttämällä Linux konsolia ja siinä kuvassa (Kuva 34) näkyviä komentoja.



```

user@Ubuntu-kone: /opt/tkedge/v1
File Edit View Search Terminal Help
user@Ubuntu-kone:~$ cd /opt/tkedge/v1
user@Ubuntu-kone:/opt/tkedge/v1$ sudo ./edge_admin manage-truststore -l uaserver
Thumbprint          Status      Common Name
7f0c4557f93e10d807d51d84250d51baa90de412 Trusted    UaExpert
a90754aad89e3c44536fd6b096ddc47ea79119ef Rejected   NodeOPCUA-Client
user@Ubuntu-kone:/opt/tkedge/v1$ sudo ./edge_admin manage-truststore -t a90754aad89e3c44536fd6b096ddc47ea79119ef uaserver
user@Ubuntu-kone:/opt/tkedge/v1$ sudo ./edge_admin manage-truststore -l uaserver
Thumbprint          Status      Common Name
7f0c4557f93e10d807d51d84250d51baa90de412 Trusted    UaExpert
a90754aad89e3c44536fd6b096ddc47ea79119ef Trusted     NodeOPCUA-Client
user@Ubuntu-kone:/opt/tkedge/v1$

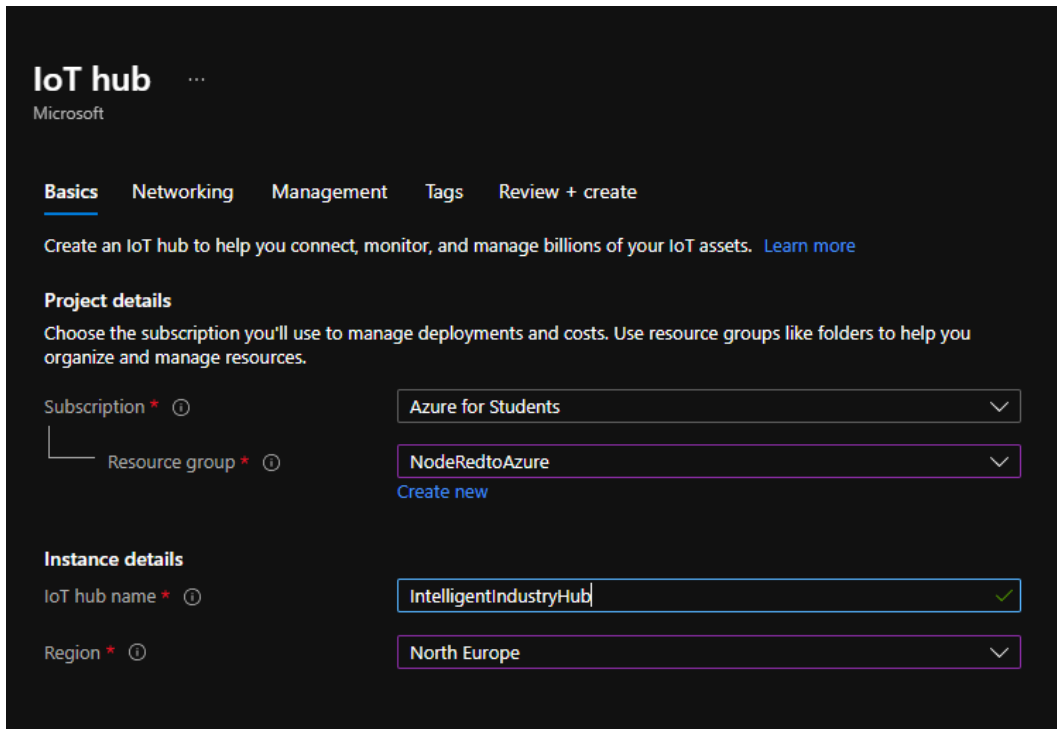
```

Kuva 34. ThingWorx Kepware Edge serverin sertifikaattien asennus.

Aluksi selvitettiin yhteyden ns. sormenjälki, jonka avulla saatiin sertifikaatti asennettua kyseiselle yhteydelle.

6.3 Azure IoT Hub

Testeissä käytettiin koulun kautta saatua Microsoft Azurea-pilvipalvelua. Se käyttöön otettiin ottamalla yhteyttä web-selaimella Azure portaaliin. Tämän jälkeen Azureen luotiin IoT Hub (Kuva 35). Tähän piti määrittää luotavan resurssin ja IoT Hubin nimi, sekä valita datakeskuksen sijainti, joka määrittää mihin data maantieteellisesti sijoittuu.



IoT hub ...
Microsoft

Basics Networking Management Tags Review + create

Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. [Learn more](#)

Project details
Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.

Subscription * ⓘ Azure for Students

Resource group * ⓘ NodeRedtoAzure
[Create new](#)

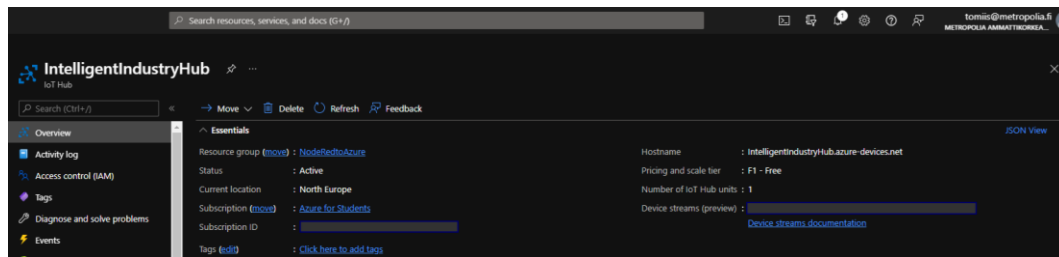
Instance details

IoT hub name * ⓘ IntelligentIndustryHub ✓

Region * ⓘ North Europe

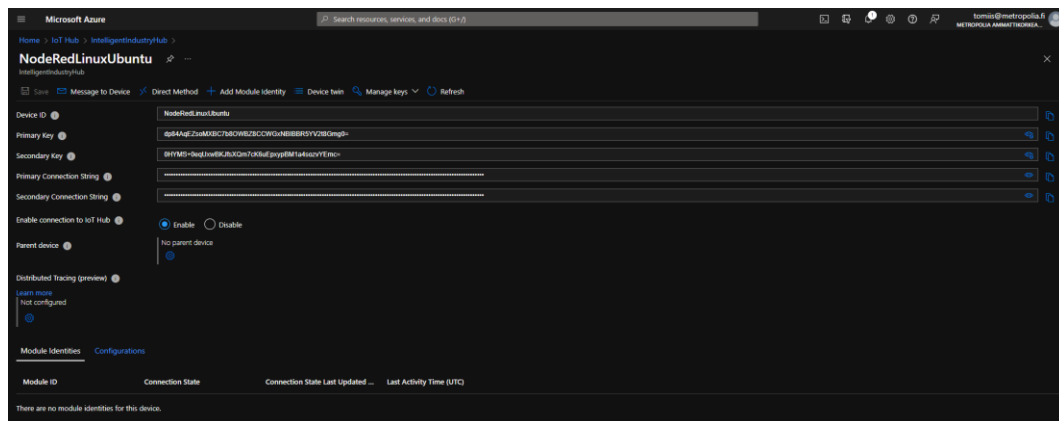
Kuva 35. Azure IoT Hub luonti

Tämän jälkeen Azure loi IoT Hub:ia jonkin aikaa, jonka jälkeen päästiin asettelemaan asetuksia itse IoT Hubiin (Kuva 36).



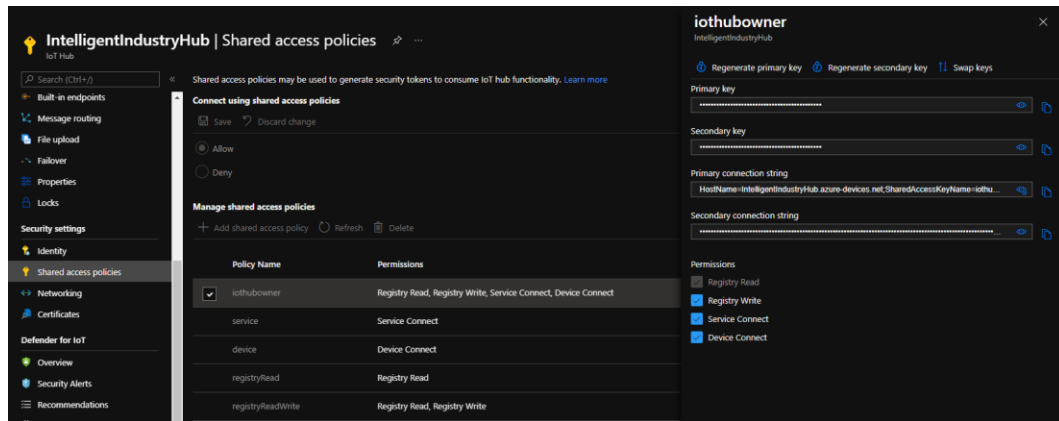
Kuva 36. Azure IoT Hub hallinta paneeli

IoT Hub-laitteen lisääminen onnistuu kahdella tapaa, laite lisätään käsin suoraan IoT Hub-valikosta Device (Kuva 37) tai sitten suoraan Node-RED-koodista lähettämällä IoT Hub:lle sen deviceId eli laitteen nimen. Laitteen lisääminen Node-RED-sovelluksen kautta käsitellään myöhemmin kappaleessa 6.4.5 Microsoft Azure kommunikointi. Kun laite luodaan manuaalisesti, luo IoT Hub-laitteille avaimet yhteyden muodostamiseen laitteelle.



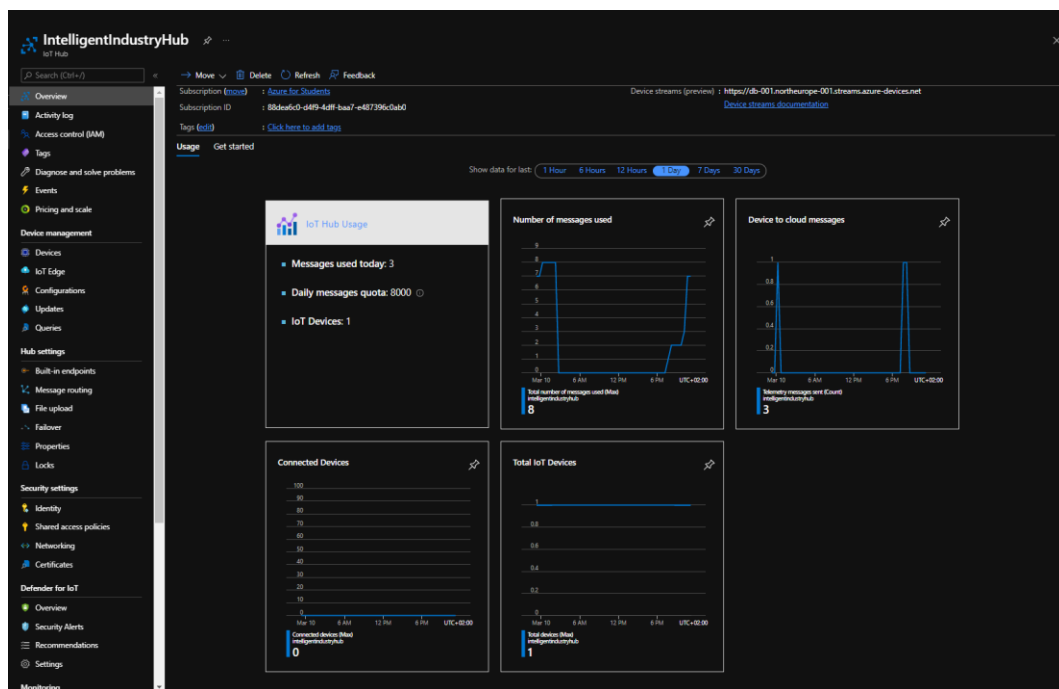
Kuva 37. Azure IoT laitteen lisäys

Testien kannalta toiseksi tärkein seikka oli selvittää IoT Hub käyttöoikeuksien yhteys-merkkijono, jolla pystyttiin laitteet lisäämään Node-RED-ohjelmasta, sekä lähettämään paluuviesti vastaanotetusta datasta. Testeissä käytettiin suurinta käyttöoikeuksia omaavaa käyttöluokkaa, joka oli nimeltään iothubowner (Kuva 38).



Kuva 38. Azure IoT Hub käyttöoikeuksien määrittämiset

Kun yhteydet oli saatu luotua ja Node-RED:stä saatiin lähetettyä dataa Azure IoT Hub:lle voitiin viestien lähetys määrää seurata luodun Azure IoT Hub-portaalista (Kuva 39).



Kuva 39. Azure IoT Hub-portaali

6.4 Node-RED

6.4.1 Valinnan perusteet testauksiin

Niin kuin kappaleessa 3.4 Node-RED todettiin, on tämä avoimen lähdekoodin ohjelmisto, jolla pystytään tuottamaan koodia nopeasti ja yksinkertaisesti. Tästä syystä myös se valittiin testeihin. Tämän lisäksi sen pystyy asentamaan helposti node.js päälle, joka puolestaan on avoimen alustan ohjelmisto, jolloin tämä sopi tarkoitukseen erinomaisesti. Myös laitevalinnoissa huomattiin, että jotkut iot-gateway:t tukivat suoraan Node-RED-ohjelmistoa, joka myös puolsi osaltaan ohjelmiston valintaa.

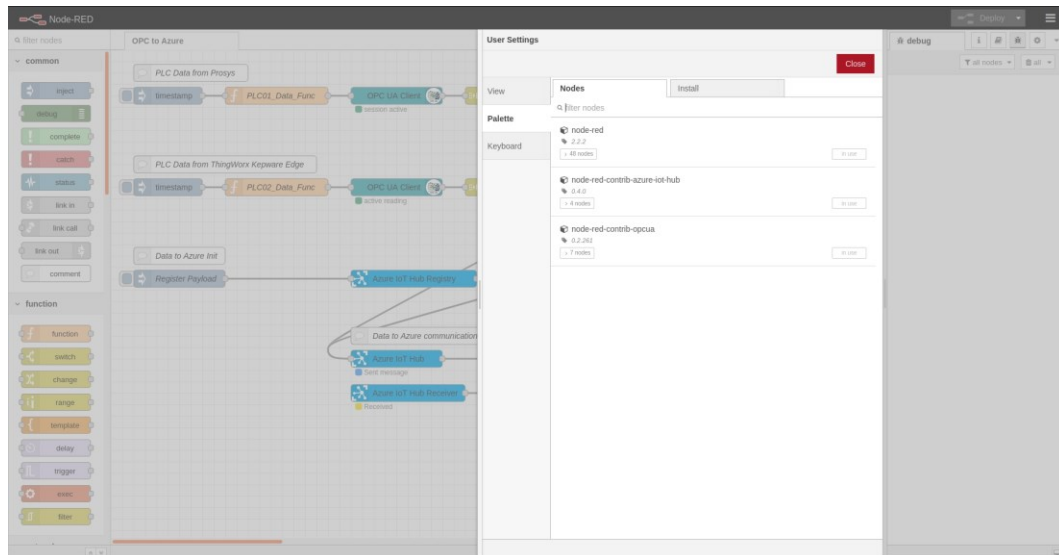
6.4.2 Node-RED-ohjelmointi

Node-RED-ohjelman rakenne koostuu kahdesta pääkohdasta, jotka ovat datan lukeminen OPC UA serveriltä ja datan lähettäminen Microsoft Azureen. OPC UA datan lukeminen on tehty kahdella erillisellä kommunikoinnilla joista toinen kommunikoi Prosys OPC UA kanssa ja toinen ThingWorx Kepware Edge OPC UA kanssa.

Seuraavissa kappaleissa tullaan käymään läpi näiden ohjelmisto rakenteita sekä niissä tarvittavia kirjastoja.

6.4.3 Node-RED kirjastot

Ensiksi varsinaisen kommunikoinnin rakentamiseen OPC UA ja Node-RED väliin piti Node-RED ladata muutamia erilaisia kirjastoja, joita pystyttiin lataamaan Node-RED palette-valikosta itse ohjelmaan. Testauksia varten ohjelmaan asennettiin seuraavat kirjastot kuvassa (Kuva 40) olevat kirjastot.



Kuva 40. Node-RED asennetut kirjastot.

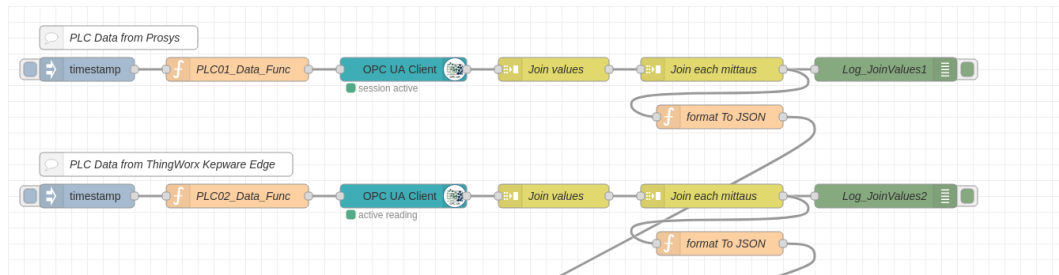
Alla on lueteltu kuvassa (Kuva 40) näkyvät asennetut kirjastojen käyttö tarkoitukset.

- Node-red-kirjasto oli valmiiksi asennettu eli se piti sisällään kaikki yleisimmät noodit.
- Node-red-contrib-azure-iot-hub-kirjaston avulla pystyttiin kommunikoimaan Microsoft Azuren pilvipalveluun.
- Node-red-contrib-opcua-kirjaston avulla pystyttiin kommunikoimaan OPC UA serverin kanssa.

Node-RED-kirjoihin löytyi ohjeita itse ohjelmasta, sekä itse ohjelman sivuilta, joissa oli myös yksityiskohtaisia esimerkkejä kirjaston käytöstä, joita myös tässä työssä hyödynnettiin.

6.4.4 OPC UA kommunikointi

Kuvassa (Kuva 41) on kuvattu OPC UA lukuun käytetty koodi, joka koostui kahdesta erillisestä kommunikoinnista joista, toinen oli Prosys OPC UA:lle ja toinen ThingWorx Kepware Edge OPC UA:lle. Kommunikointi koodit olivat muuten identtisiä keskenään, mutta ensimmäiset funktio ja opc ua client noodit olivat yksilöllisiä.



Kuva 41. Node-RED OPC UA koodit.

Seuraavaksi käydään läpi kuvassa (Kuva 41) näkyvän ohjelman toimintakuvaus.

Koodi alkoi Inject-tyyppisellä noodilla, joka on kuvattu kuvassa (Kuva 41) nimellä timestamp. Tällä noodilla pystyttiin käynnistämään virtaus koodi pätkä manuaalisesti tai ajastetusti tietyin väliajoin. Tässä testiympäristössä kummallekin kommunikoinnille rakennettiin oma käynnistyskutsunsa.

Toisena virtauksessa oli käytetty function-tyyppistä noodia, joka on kuvassa (Kuva 41) nimillä PLC01_Data_Func ja PLC02_Data_Func. Näillä pystyttiin kutsumaan haluttua mittausta OPC UA serveriltä. Esimerkkikoodi 1 on esitelty kummastakin kommunikointi funktiosta yhden mittauksen kutsu. Ensimmäinen kutsu on Prosys:stä numeerisena ja alla ThingWorx Kepware Edge:stä merkkijonona.

```
node.send({"topic":"ns=3;i=1002","datatype":"Float","plc":"PLC01","positio":"TI-1001"})
```

```
node.send({"topic":"ns=2;s=Channel1.Devicel.Tag1","datatype":"UInt16","plc":"PLC02","positio":"TI-100"})
```

Esimerkkikoodi 1. Koodi yhden mittauksen kutsusta function-noodeissa.

Toisena virtauksessa oli käytetty OpcUa - Client tyyppistä noodia, joka on kuvassa (Kuva 41) nimillä OPC UA Client. Näillä noodeilla vastaanotettiin tämän function-noodista tuleva kutsukoodi, jonka perusteella OPC UA noodi kyseli

OPC UA serveriltä kyseisen datan tietoja ja välitti vastaanotetut tiedot eteenpäin seuraavalle noodille.

Kuvassa (Kuva 41) olevat Join values ja Join each mittaus nimiset noodit olivat Join-tyyppisiä noodeja. Näiden tarkoitus oli yhdistä OPC UA Client noodilta tulevien useampien mittaus datojen vietit yhdeksi array-tyyppiseksi viestiksi.

Virtauksen viimeisenä oleva function-noodi kuvassa (Kuva 41) nimellä format To JSON tarkoitus oli muuttaa saapuvat viestit JSON-tyyppiseksi ja lisätä viestiin Microsoft Azuren kommunikoinnin kannalta tärkeät tiedot. Esimerkkikoodi 2 on esitelty, kuinka koodi oli muodostettu kyseissä noodissa. Niin kuin kappaleessa 0

Azure IoT Hub mainittiin, yhteyden muodostamisen Microsoft Azure IoT Hub:iin vaatii kaksi parametriä deviceId ja key.

```

var Mittaukset
var data
var Msg1
var NewMsg

Mittaukset = `${msg.payload[0].positio + ":" + parseFloat(msg.payload[0].payload[msg.payload[0].positio].toFixed(2))},`;
Mittaukset = `${Mittaukset + msg.payload[1].positio + ":" + parseFloat(msg.payload[1].payload[msg.payload[1].positio].toFixed(2))}`;

data = Mittaukset

msg1 = {"deviceId": "NodeRedLinuxUbuntu", "key": "dp84AqEZs-oMXBC7b8OWBZ8CCWGxNBIBBR5YV2t8Gm=", "protocol": "http", data}

newMsg = {payload: msg1};

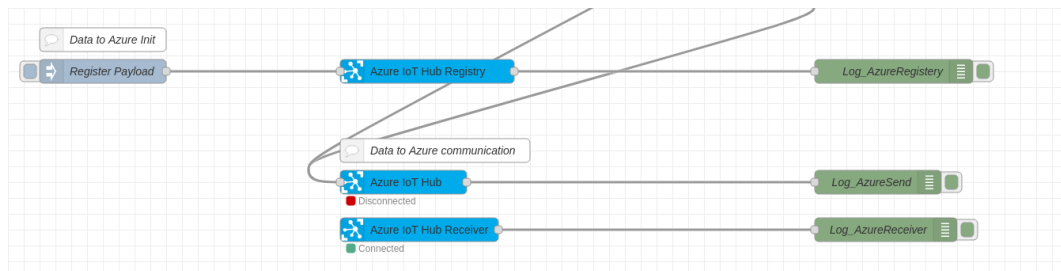
return newMsg;

```

Esimerkkikoodi 2. Koodi format To JSON-noodista.

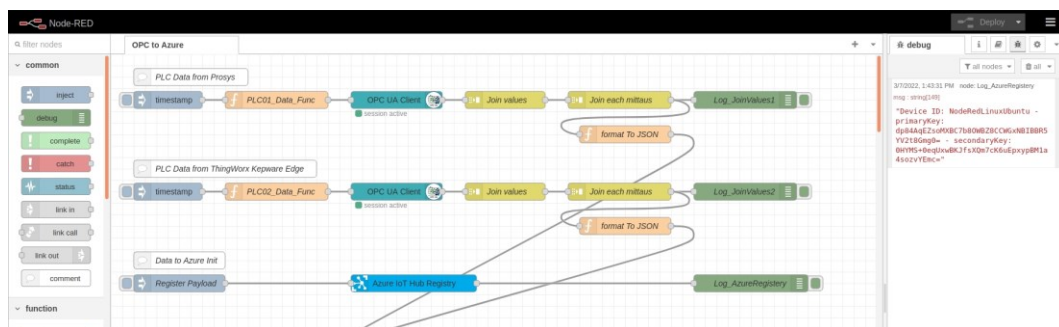
6.4.5 Microsoft Azure kommunikointi

Kuvassa (Kuva 42) on kuvattu Azure IoT Hub kirjoitukseen ja lukuun käytetty koodi, joka koostui kolmesta kohdasta. Laitteen rekisteröinti, datan kirjoitus ja luku Azure IoT Hub palveluun.



Kuva 42. Node-RED Microsoft Azure IoT Hub koodit.

Kappaleessa 6.3 Azure IoT Hub käytiin läpi, kuinka laitteen voi lisätä Azure IoT Hub:iin. Laitetta ei tarvitse välttämättä lisätä manuaalisesti Azuresta. Kuvassa (Kuva 42) esiteltä Azure IoT Hub Registry-noodi, jolla laitteen lisäys onnistuu lähettämällä siihen Injection-noodilla laitteen deviceId, joka tässä testausympäristössä oli NodeRedLinuxUbuntu. Lisäksi kyseiseen noodiin täytyi kopioida Azuresta laitteen connectionString. Kuvassa (Kuva 43) on esiteltä kuinka Azure IoT Hub vastaa laite rekisteröintiin, lähettämällä laitteen kommunikointiin tarvittavat primaryKey ja secondaryKey avaimet, joita hyödynnettiin edellisessä kappaleessa format To JSON-noodissa.

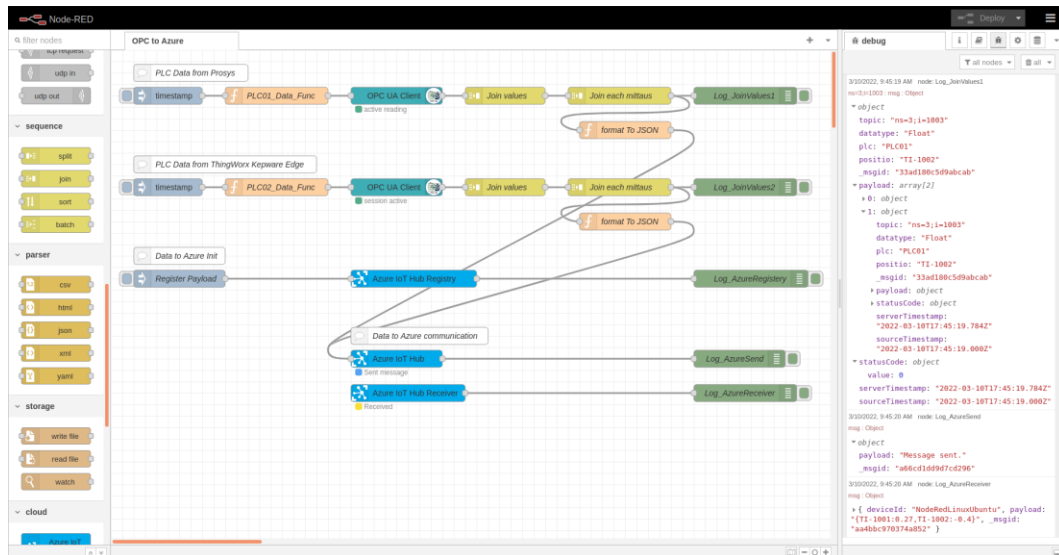


Kuva 43. Node-RED laitteen lisääminen Azure IoT Hub.

Kuvassa (Kuva 42) kaksi viimeisintä noodia Azure IoT Hub ja Azure IoT Hub Receiver tarkoitukset olivat lähettää Azure IoT Hub dataa, sekä vastaanottaa vastaus takaisin Node-RED-ohjelmaan. Azure IoT Hub-noodiin täytyi lisätä Azure palvelun Hostname. Azure IoT Hub Receiver-noodiin täytyi lisätä connectionString, jotta Azuren ja Node-RED-ohjelman välillä viestinvälitys onnistui.

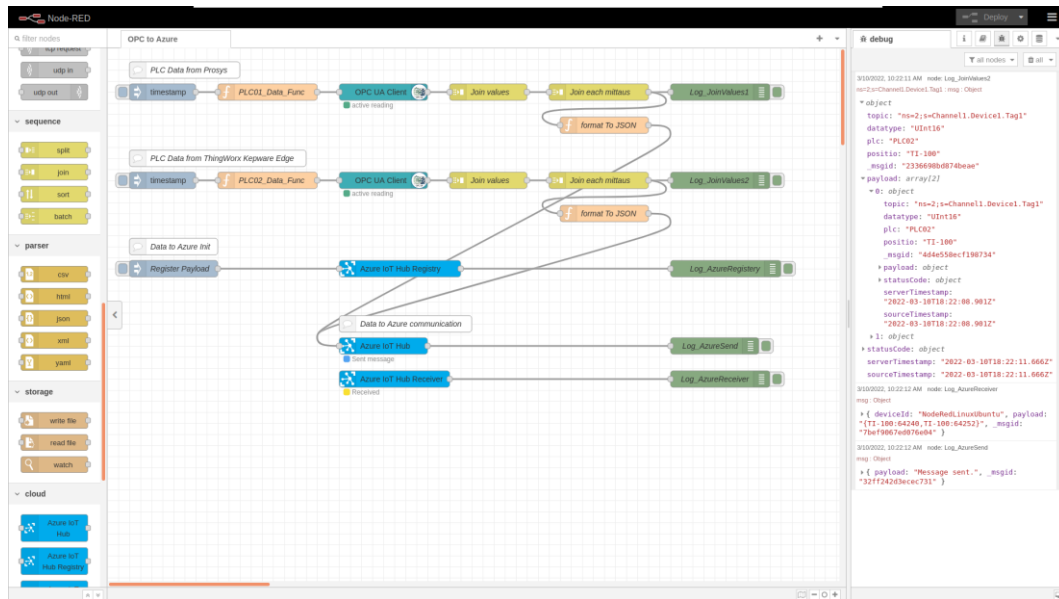
6.5 Tulokset

Kun kokonainen virtaus oli saatu rakennettua, näytti koodi kuvan (Kuva 44) kaltaiselta. Kummatkin OPC UA kommunikoinnit yhdistettiin Azure IoT Hub noodiin, josta se välitti datan Azure IoT Hubiin.



Kuva 44. Valmis Node-RED-koodi ja datan luku Prosys OPC UA-kommunikoinnista.

Kuvassa (Kuva 44) oikealla puolella on tulostunut lokia koodista vihreiden debug noodien kautta. Tästä voitiin huomata, kuinka data liikkui virtauksen mukana lopulta Azure IoT Hubiin ja sieltä takaisin. Azureen lähetetyt viestit sisälsivät OPC UA:sta saadut seuraavat tiedot: mittauksien positiot, PLC nimi, sekä mittauksien arvot. Log AzureSend-lokista voidaan huomata, että viesti oli lähtenyt. Log AzureReceiver-kohdasta nähtiin lähetetyn viestin tiedot vastaanotettuna takaisin Azure IoT Hub:sta. Kuvan (Kuva 44) lokitiedot ovat tulostuneet Prosys OPC UA kommunikoinnin tiedoista, mutta vastaavasti tiedot saatiin, sammalla tavalla haettu ThingWorx Kepware Edge:stä (Kuva 45)



Kuva 45. Node-RED koodi ja datan luku ThingWorx Kepware Edge kommunikoinnista.

7 Yhteenveto

Tämän työn tavoitteena oli tutkia mitä erilaisia laiteratkaisuja teollisuudessa olevien laitteiden ja Microsoft Azure-pilvipalveluiden väliin löytyy, sekä testata löydettyjä ratkaisuja käytännössä. Laiteratkaisujen valintoihin vaikuttivat tutkimuksen tilaajan vaatimukset laiteista.

Laittevalintojen jälkeen toteutettiin löydettyjen laitteiden käyttöjärjestelmää mukaillen testausympäristö, jolla pystyttiin simuloimaan suurilta osin oikeaa vastaavaa laitetta. Käyttöjärjestelmään asennettiin reunalaskentaa simuloiva Node-RED-ohjelmisto ja kaksi eri ohjelmisto kehittäjän OPC UA simulointi serveriä, jolla mahdollistettiin logiikkaraja pinnan kommunikoinnin simulointi. Testausympäristön pystytys antoi kokemusta Linux käyttöjärjestelmän käytöstä, ja kommunikoinnin testaukset auttoivat hahmottamaan arkkitehtuurinmallin käytännössä.

Tutkimuksen aihe ja sen laajuus tuottivat vaikeuksia työn rajauksessa. Haasteita tuotti myös työn rakenteen löytäminen, joka osittain johtui myös työn rajauksesta.

Työssä vastattiin kahteen tutkimuskysymykseen. Tutkimuskysymykset ja niiden tuottamat ratkaisut käydään tässä luvussa läpi.

1. Mitä laiteratkaisuja löytyy Azure-pilvialustan ja logiikoiden väliseen kommunikointiin?

Laiteratkaisuja tutkittiin hyvin laajasti ja aluksi mitään löydettyjä ratkaisuja ei jätetty pois. Työn edetessä työstä rajattiin laitteet pois, joiden ominaisuudet jäivät kauaksi niille määritetyistä vaatimuksista. Löydetyt laiteratkaisut pystyttiin jakamaan kolmeen eri ryhmään, jotka olivat: Teollisuus PC pohjaiset ratkaisut, sulautetut järjestelmät ja Linux pohjaiset järjestelmät.

Laiteratkaisuissa Teollisuus PC pohjaiset ratkaisut omasivat suurimman koon, laskentatehon, liitännät ja hinnan, sekä näiden lisäksi niiden käyttöjärjestelmä tuki oli laajin. Sulautetut järjestelmät ja Linux pohjaiset ratkaisuissa suurimat

erot olivat käyttöjärjestelmä. Sulautetuissa järjestelmissä käytettiin laitevalmistajan omaa käyttöjärjestelmää, jotka eivät sopineet laiteille asetettuihin vaatimuksiin.

2. Kuinka mittausdata saadaan välitettyä Microsoft Azure-pilvipalveluun?

Mittausdatan välittäminen pilvipalveluun voi tapahtua monella eritapaa. Yhteystapoja voi olla suoraa laitteelta–laitteelle, laitteelta–pilveen ja laitteelta–yhdyskäytävälle. Työn tilaaja halusi hyödyntää kommunikoinnissa reunalaskentaa ja laitteen skaalautuvuutta erilaisiin ratkaisuihin, joka puolestaan rajasi yhteystavan käyttämään yhteysmallia laitteelta – yhdyskäytävälle yhteysmallia. Tätä yhteystapaa hyödyntämällä voidaan dataa siirtää monipuolisimmin erilaisista laitteista aina erilaisiin pilvipalveluihin. Reunalaskennallisella yhteystavalla saadaan data pakattua tiiviimmin yhteen viestiin, jolloin viestien lähetykset vähenevät merkittävästi verrattuna viestien lähettämiseen erikseen pilvipalveluun.

Tutkimus täytti sille annetut vaatimukset. Tutkimus tuotti ratkaisun tutkimusongelmiin ja ne todettiin käytännössä toimivaksi ratkaisuksi. Käytäntöön jalostamisen kannalta haasteita tuotti ohjelman skaalautuvuus erilaisiin projekteihin. Vaikka OPC UA tuo suuren skaalattavuuden erilaisten laitteiden kanssa, täytyy itse Node-RED-ohjelmaa kehittää siten, että tämän skaalautuvuus olisi mahdollisimman hyvä. Lisäksi jatkotutkimuksiin täytyy panostaa laitetestauksien ja niiden tietoturvallisuuden tutkimiseen, jotka jäivät tässä tutkimuksessa niukemmalle. Tietoturvallisuus on nykypäivänä erittäin tärkeitä asia, niin siviili laitteissa kuin teollisuudessakin.

Lähteet

ABB Oy, 2007. TTT-käsikirja 2000-07, luku 5. Automaation tietoliikennetekniikka. Verkkodokumentti. Saatavilla: <http://www.oamk.fi/~kurki/automaatiolabrat/TTT/05_0_Automaation%20tietoliikenne.pdf>. Luettu 20.2.2022.

Dell, 2022. Embedded Box PC 3000. Verkkodokumentti. <<https://www.dell.com/fi/yriytykset/p/embedded-box-pc-3000/pd>>. Luettu 16.3.2022.

Devasia, A., 2021. The Main Difference Between IoT and IIoT. Verkkodokumentti. <<https://control.com/technical-articles/the-main-difference-between-iiot-and-iiot/>>. Luettu 10.2.2022.

Froehlich, A., Rosencrance, L. & Gattine, K., 2022. OSI model (Open Systems Interconnection). Verkkodokumentti. <https://www.techtarget.com/searchnetworking/definition/OSI?_gl=1*168re7l*_ga*MTkzNjQ3NzMzMC4xNjQyNTQ2MDU1*_ga_TQKE4GS5P9*MTY0NDI2Njc1NC44LjEuMTY0NDI2NjkzMS4w&_ga=2.7309876.762076623.1644243524-1936477330.1642546055>. Luettu 11.2.2022.

Gillis, A.S., 2022. What is internet of things (IoT)? Verkkodokumentti. <<https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>>. Luettu 10.2.2022.

H. Tschofenig ym., 2015. Architectural Considerations in Smart Object Networking. Internet Architecture Board (IAB).

Henrik Nygren, A.L.a.t.A.E.R.g., 2018. Johdatus tietoliikenteeseen. Verkkodokumentti. <<https://johdatus-tietoliikenteeseen-19.mooc.fi/osa-1/2-protokolla-jasoite>>. Luettu 19.1.2022.

Hewlett Packard Enterprise, 2022. HPE GL10 IoT Gateway - Overview. Verkkodokumentti. <https://support.hpe.com/hpesc/public/docDisplay?docId=c04930066&docLocale=en_US>. Luettu 16.3.2022.

IBM, 2020. Get started with Node-RED. Verkkodokumentti. <https://developer.ibm.com/learningpaths/get-started-node-red/?mhsrc=ibmsearch_a&mhq=node%20red>. Luettu 23.1.2022.

Industrial internet consortium, 2019. The Industrial Internet of Things Volume G1: Reference Architecture. Verkkodokumentti. Industrial internet consortium Saatavilla: <<https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf>>. Luettu 7.2.2022.

International Society of Automation (ISA), 2022. What Is Industry 4.0? Verkkodokumentti. <<https://blog.isa.org/what-is-industry-40>>. Luettu 26.1.2022.

International Society of Automation (ISA), 2022. What's the Difference Between Industry 4.0 and Industry 5.0? Verkkodokumentti. <<https://blog.isa.org/whats-the-difference-between-industry-40-industry-50>>. Luettu 26.1.2022.

I-scoop, 2022. Industry 4.0 and the fourth industrial revolution explained. Verkkodokumentti. <<https://www.i-scoop.eu/industry-4-0/>>. Luettu 26.1.2022.

Li, Y., Li, D., Cui, W. & Zhang, R., 2011. Research based on OSI model. Introduction, 29 May. pp.554-57. Saatavilla: <https://ieeexplore-ieee-org.ezproxy.metropolia.fi/document/6014631>. Luettu 11.2.2022.

Magnus, Å., 2018. Implementing Shop Floor IT for Industry 4.0. Gothenburg, Sweden: Chalmers University of Technology.

MB connect line, 2020. mbEDGE.advanced Manual V 2.0. Verkkodokumentti. (2.0) <<https://downloadportal.mbconnectline.com/en/mbedge.html>>. Luettu 16.3.2022.

Microsoft, 2008. Introducing Windows Azure. Verkkodokumentti. <<https://azure.microsoft.com/en-us/blog/introducing-windows-azure/>>. Luettu 17.1.2022.

Microsoft, 2013. Introducing Windows Azure. Washington 2013: Microsoft Press. Saatavilla: https://download.microsoft.com/DOWNLOAD/D/6/7/D670D322-5771-409E-BF34-5B98496DEB0A/MICROSOFT_PRESS_EBOOK_INTRODUCING_AZURE_PDF.PDF.

Microsoft, 2014. Upcoming Name Change for Windows Azure. Verkkodokumentti. <<https://azure.microsoft.com/en-us/blog/upcoming-name-change-for-windows-azure/>>. Luettu 17.1.2022.

Microsoft, 2021a. Azure industrial IoT analytics guidance. Verkkodokumentti. <<https://docs.microsoft.com/en-us/azure/architecture/guide/iiot-guidance/iiot-architecture>>. Luettu 26.1.2022.

Microsoft, 2021b. Cloud monitoring guide: Formulate a monitoring strategy. Verkkodokumentti. <<https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/strategy/monitoring-strategy>>. Luettu 14.03.2022.

Microsoft, 2021c. Azure portal overview. Verkkodokumentti. <<https://docs.microsoft.com/en-us/azure/azure-portal/azure-portal-overview>>. Luettu 19.1.2022.

Microsoft, 2022a. What is IaaS? Verkkodokumentti. <<https://azure.microsoft.com/en-gb/overview/what-is-iaas/>>. Luettu 17.1.2022.

Microsoft, 2022b. What is PaaS? Verkkodokumentti. <<https://azure.microsoft.com/en-gb/overview/what-is-paas/>>. Luettu 17.1.2022.

Microsoft, 2022c. What is SaaS? Verkkodokumentti. <<https://azure.microsoft.com/en-gb/overview/what-is-saas/>>. Luettu 17.1.2022.

Microsoft, 2022d. Introduction to Azure IoT Hub. Verkkodokumentti. <<https://docs.microsoft.com/en-us/learn/modules/introduction-to-iot-hub/>>. Luettu 14.03.2022.

Microsoft, 2022e. IoT concepts and Azure IoT Hub. Verkkodokumentti. <<https://docs.microsoft.com/en-us/azure/iot-hub/iot-concepts-and-iot-hub/>>. Luettu 1.11.2022.

Microsoft, 2022f. Azure IoT Hub. Verkkodokumentti. <<https://azure.microsoft.com/en-gb/services/iot-hub/>>. Luettu 19.1.2022.

Moxa, 2022. Arm-Based Computers. Verkkodokumentti. <<https://www.moxa.com/en/products/industrial-computing/arm-based-computers>>. Luettu 17.3.2022.

Node-RED, 2022a. About Node.js®. Verkkodokumentti. <<https://nodejs.org/>>. Luettu 23.1.2022.

Node-RED, 2022b. Editor Guide. Verkkodokumentti. <<https://node-red.org/docs/user-guide/editor/>>. Luettu 23.1.2022.

Omron, 2022. Remote Access Solution. Verkkodokumentti. <https://industrial.omron.eu/en/products/remote-access-solution#specifications_ordering_info>. Luettu 15.3.2022.

OPC Connect, 2022. Open programmers connection. Verkkodokumentti. <<https://www.opcconnect.com/>>. Luettu 19.02.2022.

OPC Foundation, 2022a. What is OPC? Verkkodokumentti. <<https://opcfoundation.org/about/what-is-opc/>>. Luettu 19.02.2022.

OPC Foundation, 2022b. Unified Architecture. Verkkodokumentti. <<https://opcfoundation.org/about/opc-technologies/opc-ua/>>. Luettu 19.2.2022.

Open Automation Software, 2022. What is an IoT Gateway? Verkkodokumentti. <<https://openautomationsoftware.com/blog/what-is-an-iot-gateway/>>. Luettu 3.2.2022.

OpenJS Foundation & Contributors, 2022. About. Verkkodokumentti. <<https://nodered.org>>. Luettu 23.1.2022.

Pal, A. & Purushothaman, B., 2017. IoT Technical Challenges and Solutions. Norwood: Artech House.

Phoenix contact, 2022. IoT gateway - CLOUD IOT GATEWAY. Verkkodokumentti. <<https://www.phoenixcontact.com/en-za/products/gatewaysproxies-cloud-iot-gateway-1031235>>. Luettu 16.3.2022.

Riedl, M. & Yoon, D., 2019. OPTimised Industrial IoT and Distributed Control Platform for Manufacturing and Material Handling Deliverable 2.1 Evaluation of IoT-platforms and Definition of the IIoT- platform Architecture. Saatavilla: <https://www.semanticscholar.org/paper/OPTimised-Industrial-IoT-and-Distributed-Control-of-Riedl-Yoon/4c80b8dcf83d6b49b493568a858fb369f588be93#extracted>.

Rose, K., Scott, E. & Lyman, C., 2015. The internet of things an overview. The Internet Society (ISOC). Saatavilla: <https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf>. Luettu 3.2.2022.

Shyam, N., Stackowiak, & Romano, C., 2017. Architecting the Industrial Internet. Birmingham: Packt Publishing Ltd.

Siemens, 2022. 6ES7647-0AA00-1YA2. Verkkodokumentti. <<https://mall.industry.siemens.com/mall/en/hu/Catalog/Product/6ES7647-0AA00-1YA2>>. Luettu 16.3.2022.

Siemens, 2022. 6ES7647-0BA00-1YA2. Verkkodokumentti. <<https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7647-0BA00-1YA2>>. Luettu 16.3.2022.

Suomen Automaatioseura ry, 2007. Automaatiosuunnittelun prosessimalli. Helsinki: Suomen Automaatioseura ry.





Welotec, 2022. IoT Edge Gateway. Verkkodokumentti. <<https://www.welotec.com/iot-edge-gateway/>>. Luettu 17.3.2022.





Wikipedia, 2021. IEEE 802. Verkkodokumentti. <https://fi.wikipedia.org/wiki/IEEE_802>. Luettu 25.2.2022.





Yuqian Lu, C.L., Wang, K.I.-K., Huang, H. & Xu, X., 2020. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. Robotics and Computer-Integrated Manufacturing, 61(0736-5845), s.101837. Saatavilla: <https://www.sciencedirect.com/science/article/pii/S0736584519302480>. Luettu 20.12.2020.

Liitteet

Laittevertailu taulukko

Valmistaja	Hewlett Packard Enterprise GL10 IoT Gateway	Dell Embedded Box PC3000	Omron RTI SiteManager	Phoenix Cloud IoT gateway
Typppi Kuva				
Linkki	https://support.hpe.com/hpesc/public/docDisplay?docLocale=en_US&docId=c04930066	https://www.dell.com/fi/xrtykset/p/embedded-box-pc-3000/pd	https://industrial.omron.eu/en/products/remote-access-solution	https://www.phoenixcontact.com/online/portal/us?url=pxc-oc-itemdetail;pid=1031235&library=usen&tab=1
Asemus tyyppi	Seinä, VESA, Rack	DIN, Seinä, VESA	DIN	DIN
Koko (WxHxD)	35.98 x 138.5 x 116.4 mm	237 x 160 x 60 mm	107 x 32 x 97 mm	22.5 x 99 x 114.5 mm
Jännitesyöttö	12VDC	12-26VDC	12-24VDC	18-30VDC
Virta	3A	5.4A		0.2A
Ethernet	2x 1Gbps	2x 1Gbps	4x 100Mbps	2x 100Mbps
Sarjaliikenne	1x (RS232/RS422/RS485)	2x (RS-232/485), 1x (RS-232/422)	1x (RS232)	1x (RS232/RS485)
IO	-	GPIO: 12 bit	2x DI, 2x DO	-
Wifi	2x USB	Wi-Fi IEEE 802.11n	1x USB	1x USB type C
USB	1x Micro HDMI, 1x HDMI, 1x VGA	4x USB 2.0, 1x USB 3.0	-	-
Display	-	ZigBee and CAN bus card	-	-
Extra portit	-	-	-	-
ExtAntenna	2x WWAN SMA FM, 2x Wi-Fi RP-SMA FM	2-Port Wifi, 2-Port LTE or 4-in-1 LTE/Wifi	1x RP-SMA	-
Mobile	-	LTE	LTE	-
Proessori	Intel Atom E3825, 1460MHz	Intel Atom E3825, 1300MHz	ARM Cortex A7 MCU, 800MHz	ARM Cortex-A8 1x 600 MHz
Muisti	4GB DDR3	1x 4GB / 1x 8GB DDR3	-	-
Kovalevy	32GB mSATA	SSD / HDD	SD-CARD	-
Käyttöjärjestelmä	Windows IoT Core, Microsoft Windows Server, Canonical Ubuntu Snappy Core, CentOS	Factory options for Ubuntu Desktop 16.04, Win7 Pro, Win7 Pro for Embedded Systems (FES7), Win10 Pro, Win10 IoT Enterprise 2015 LTSB, Win Standard Embedded 7	Omron Site Manager	Proficloud
Ohjelmit				
Hinta-arvio	968 €	1 231 €	?	400 €

Valmistaja	Siemens		Siemens	mbNet
Tyyppi	Simatic IOT2050		Simatic IOT2040	MDH810-MHD850
Kuva				
Linkki	https://mall.industry.siemens.com/mall/en/WW/Ca/taalog/Product/6ES7647-0BA00-0YA2	https://mall.industry.siemens.com/mall/en/WW/Ca/taalog/Product/6ES7647-0BA00-1YA2	https://mall.industry.siemens.com/mall/en/WW/Ca/taalog/Product/6ES7647-0AAA00-1YA2	https://mbconnectline.com/mbnet/
Asennus tyyppi	DIN, Seinä	DIN, Seinä	DIN, Seinä	DIN, Seinä
Koko (WxHxD)	37x142x100mm	37 x 142 x 100mm	144 x 90 x 53mm	48 mm x 137 mm x 140 mm
Jännitesyöttö	12/24VDC	12/24VDC	9-36VDC	10-30VDC
Virta	1,7A	1,7A		0,5A
Ethernet	2x 1Gbps	2x 1Gbps	2x 100Mbps	4x
Sarjaliikenne	1x (RS232/RS422/RS485)	1x (RS232/RS422/RS485)	2x (RS232/RS485)	1x (RS-232/485), 1x (RS-232/485)
IO	5x DI, 2x AI, 2x DO	5x DI, 2x AI, 2x DO		
Wifi	-	-		
USB	2x USB 2.0	2x USB 2.0	1x USB 2.0, 1x USB	1x USB
Display	1x DisplayPort	1x DisplayPort	-	-
Extra portit	1x Arduino, 1x mPCIe	1x Arduino, 1x mPCIe	1x Arduino, 1x mPCIe	-
EkAntenna	-	-	-	2x RP-SMA
Mobile	-	-	-	LTE
Proessori	ARM TI AM6528 GP	ARM TI AM6548 HS	INTEL Quark x1020 SoC (System on Chip)	ARM Cortex-A8 1000 MHz
Muisti	1GB DDR4	2GB DDR4	1GB DDR3	512MB
Kovalevy	microSD card	microSD card	microSD card	SD-CARD
Käyttöjärjestelmä	Linux Debia 10, SIMATIC Industrial OS	Linux Debia 10, SIMATIC Industrial OS	Linux Debia 10, SIMATIC Industrial OS	mbEDGE
Ohjelmaa	Node-RED	Node-RED		Node-RED, Docker, Portainer.io
Hinta-arvio	423 €	591 €	211 €	1 350 €

	Moxa		Welotec	
Valmistaja	UC-3100	UC-8100	EG500	EG600
Tyyppi				
Kuva				
Linkki	https://www.moxa.com/en/products/industrial-computing/arm-based-computers/uc-3100-series	https://www.moxa.com/en/products/industrial-computing/arm-based-computers/uc-8100-series	https://www.welotec.com/product/edre-gateway-for-azure-iiot-hub-eg500-series/	https://www.welotec.com/product/edre-gateway-eg600-series/
Asennus tyyppi	DIN, Seinä	DIN, Seinä	DIN, Seinä	DIN, Seinä
Koko (WxHxD)	41 mm x 89,1 mm x 128,5 mm	33 mm x 125,6 mm x 141 mm	64 mm x 92 mm x 140 mm	30 mm x 90 mm x 110 mm
Jännitesyöttö	9-36VDC	9-36VDC	12-24VDC	12-24VDC
Virta	0,5A	0,7A		
Ethernet	2x 100Mbps	2x 100Mbps	3x 1Gbps	3x 1Gbps
Sarjaliikenne	1x (RS-232/422/485)	2x (RS-232/422/485)	1x (RS-232)	1x (RS-232), 1x (RS-485)
IO				
WiFi				
USB	1x USB 2.0	2x USB 2.0	4x USB 4.0	4x USB 4.0
Display	-	-	HDMI	HDMI
Extra portit	-	-	-	-
EkAntenna	2x RP-SMA	2x RP-SMA	2x SMA	2x SMA
Mobile	LTE	LTE	LTE	LTE
Proessori	Arm v7 Cortex-A8 1000 MHz	Arm v7 Cortex-A8 1000 MHz	Intel Atom x5-E3940 / Intel Atom x7-E3950	Intel Quad Core Atom x7-E3950
Muisti	1GB DDR3	1GB DDR3	8GB	8GB
Kovalevy	8 GB eMMC, SD slots x 1	8 GB eMMC, SD slots x 1	45GB	45GB
Käyttöjärjestelmä	Linux Debian 9 kernel 4.4 (Moxa Industrial Linux)	Linux Debian 9 kernel 4.4 (Moxa Industrial Linux)	Docker cli, Docker compose, Moby Engine, Azure Edge	Docker cli, Docker compose, Moby Engine, Azure Edge
Ohjelmaa			Container	Container
Hinta-arvio	638 €	747 €	?	?

Laitevertailu kaaviot

