Tampere University

Toni Kauppinen

IoT-alustojen tietoturvan soveltuvuusvertailu CityIoT-hankkeeseen

Informaatioteknologian ja viestinnän tiedekunta
Diplomityö
Tammikuu 2021

Tiivistelmä

Toni Kauppinen: IoT-alustojen tietoturvan soveltuvuusvertailu CityIoT-hankkeeseen
Diplomityö
Tampereen yliopisto
Tietotekniikan diplomi-insinöörin tutkinto-ohjelma
Tammikuu 2021

Vuoteen 2050 mennessä maailman väestön lukumäärän on arvioitu kasvavan melkein 10 miljardiin ja 80% väestöstä tulisi asumaan kaupungeissa. Nopea väestönkasvu asettaa haasteita kaupunkien jatkuvalle kehittämiselle, niiden energiakulutukselle, ihmisten liikkumiselle, vesi- ja jätehuollolle, infrastruktuurille, yleiselle turvallisuudelle sekä terveydenhuollon järjestämiselle. "Älykäs kaupunki" käyttää hyväkseen uusimpia teknologioita ja pyrkii parantamaan kaupunkien palveluita sekä ratkaisemaan kaupunkien ongelmakohtia. Sen tavoitteena on kehittää päätöksentekokykyä, pienentää päästöjä, parantaa sosiaalista ja taloudellista laatua sekä vankistaa asukkaittensa yhteisöllisyyden tunnetta.

Esineiden Internet (IoT) on tietojenkäsittelyn konsepti, minkä avulla jokapäiväiset fyysiset laitteet voidaan yksilöidä ja liittää Internetiin. IoT on verkko, joka muodostuu keskenään kommunikoivista laitteista, jotka keräävät että tallentavat tietoa ympäristöstään ilman ihmisen apua. IoT-alusta on IoT-järjestelmän ydin jonka avulla hallitaan kaikkia laitteita, yhteyksiä, ohjelmisto- ja sovelluskerroksia. IoT-järjestelmä voi sisältää suuren määrän heterogeenisiä laitteita, jonka vuoksi turvallisuus ja yksityisyys ovat kriittisiä asioita näissä järjestelmissä. Myös tallennettujen tietojen luottamuksellisuus ja eheys on varmistettava. Tämän lisäksi on käytettävä hyviksi todettuja todennusmekanismeja, jotta luvattomat tahot eivät pääsisi käyttämään järjestelmää väärin. Toisaalta järjestelmässä pitää pystyä takaamaan käyttäjien yksityisyys, nimettömyys sekä tietosuoja.

Tämä työn tarkoitus on vertailla eri IoT-alustojen tietoturvaominaisuuksia CityIoT-hankeen vaatimusten perusteella. Vertailun apuna käytettiin hankeen omia tietoturvavaatimuksia sekä yleisesti tunnettujen alalla toimivien tahojen tietoturvaohjeistuksia. IoT-alustojen valintakriteereinä on käytetty järjestelmien pohjautumista avoimeen lähdekoodiin sekä alustan soveltuvuutta älykkäisiin kaupunkeihin.

Avainsanat: Älykkäät kaupungit, IoT-alusta, tietoturva

Abstract

Toni Kauppinen: Compatibility Comparison of IoT Platform Security in the CityIoT Project
Master of Science Thesis
Tampere University
Master's Degree Programme in Information Technology
Januar 2021

By 2050, the number of population in the world is estimated to grow to almost 10 billion and an estimated 80% of people would live in cities. Rapid population growth poses challenges to the sustainable development of cities, their energy consumption, for public transportation, water and waste management, infrastructure, public safety and the organization of health care. The "smart city" takes advantage of the latest technologies and thus strives to improve its services as well as solve the city's problem areas. The main objectives are to develop decision-making capacity, reduce emissions and problem areas, improve social and economic quality and strengthen the sense of community of its residents.

The Internet of Things (IoT) is a concept that allows everyday physical devices to be identified and connected to the Internet. IoT is a network of devices that communicate with each other and collect and store information about their environment without human interaction. The IoT platform is the core of every IoT system that manages all devices, connections, software and application layers. An IoT system can contain a large number of devices and therefore security is critical in these systems. The confidentiality and integrity of the data stored must also be ensured. In addition, proven authentication mechanisms must be used to block unauthorized access to the system. Also, the system must be able to guarantee users' privacy, anonymity and data protection.

The purpose of this thesis is to evaluate the security features of different IoT platforms based on the requirements of the CityIoT project. The comparison was based on the project's own security requirements and the security publications of well-known parties operating in the field. Criteria for selecting IoT platforms was based on open source systems and suitability specifically for smart cities.

Keywords: Smart cities, IoT platforms, information security

Alkusanat

Haluan kiittää Tampereen teknillistä yliopistoa (nykyään Tampereen yliopisto) kuluneista 10 vuodesta. Näiden vuosien aikana olen saanut syvennettyä osaamistani hypermedian parissa sekä saanut kehitettyä entisestään automaattisen tietojenkäsittelyn taitojani. Kiitän myös Professori Kari Systää diplomityön aiheesta ja siihen liittyvästä ohjauksesta.

Lisäksi haluan kiittää poikiani, Iivoa ja Edviniä, jotka ovat piristäneet päiviäni opintojen lomassa. Erityiskiitos myös vaimolleni, Niina Simoselle, saamastani tuesta opintojeni loppuun saattamisessa.

Tampereella 13.1.2021

Toni Kauppinen

Sisältö

Lyhenteet ja merkinnät

3G, 4G, 5G	Versiointi laajakaistaisille matkapuhelinverkkotekniikoille
AD	Active Directory, Microsoft Windows -toimialueen käyttäjätietokanta ja hakemistopalvelu
API	Application Programming Interface, ohjelmointirajapinta
AMQP	Advanced Message Queuing Protocol, avoin standardi viestien välittämiseen sovellusten välillä
Audit log	IT-järjestelmien tapahtumalokien tietue, tyypillisesti aktiviteettisekvenssin tai tietyn toiminnan tallenne
CDMA	Code Division Multiple Access, koodijakokanavointi, radiotien kanavanvaraustekniikka, jota käytetään laajakaistaisissa järjestelmissä
CoAP	Constrained Application Protocols, uudelleenlähetysprotokolla, joka sisältää pyyntö- ja vastausviestejä
CRM	Customer Relationship Management, asiakkuudenhallinta
CRUD	Create, Read, Update, Delete, neljä toimintoa pysyvän tiedon tallentamiseen tietokantoihin
CSV	Comma-Separated Values, tiedostomuoto, jolla tallennetaan yksinkertaista taulukkomuotoista tietoa tekstitiedostoon käyttäen pilkkua erottimena
DTLS	Datagram Transport Layer Security, tietoliikenneprotokolla, joka tarjoaa tietoturvaa datagrammipohjaisille sovelluksille
E2E	End-To-End, käsite missä huomioidaan kaikkien prosessien, palvelujen ja käyttäjien tarvitsemat tarpeet
EP	Endpoint, Kaa IoT -alustan käyttämä termi laitteista sekä muista järjestelmään liitetyistä asioista ja järjestelmistä
ERP	Enterprise Resource Planning, toiminnanohjausjärjestelmä
GDPR	General Data Protection Regulation, Euroopan Unionin yleinen tietosuoja-asetus
GSM	Global System for Mobile Communications, maailmanlaajuisesti käytetty digitaalinen matkapuhelinjärjestelmä
HTTP	Hypertext Transfer Protocol, tiedonsiirtoprotokolla, jota selaimet ja palvelimet käyttävät
HTTP POST	HTTP-metodi, jolla välitetään palvelimelle mm. selaimessa muokattavien tekstikenttien sisällöt
ICT	Information and Communication Technology, tieto- ja viestintätekniikka

IETF	Internet Engineering Task Force, Internet-arkkitehtuurin kehityksestä ja Internetin sujuvasta toiminnasta kiinnostunut avoin kansainvälinen yhteisö
IoT	Internet of Things, Esineiden Internet
IP	Internet Protocol, protokolla, jonka vastuulla on IP-tietoliikennepakettien toimittaminen perille pakettikytkentäisessä Internet-verkossa
JSON	JavaScript Object Notation, avoimen standardin tiedostomuoto tiedonvälitykseen
LDAP	Lightweight Directory Access Protocol, verkkoprotokolla hakemistopalvelujen käyttöön
LoRaWAN	Long Range Wide Area Network, tiedonsiirtoverkko langatonta ja nopeaa tiedonsiirtoa varten
M2M	Machine to Machine, suora kommunikaatioyhteys kahden laitteen välillä
MQTT	Message Queuing Telemetry Transport, kevyt julkaise/tilaa-verkkoprotokolla
OPC-UA	OPC Unified Architecture, OPC Foundation kehittämä kommunikaatioprotokolla kahden laitteen välille
OWASP	Open Web Application Security Project, online-yhteisö, joka tuottaa vapaasti saatavilla olevia artikkeleita, menetelmiä, dokumentaatiota, työkaluja ja tekniikoita web-sovellusten tietoturvan parantamiseksi
QR-koodi	Quick Response -koodi, kaksiulotteinen kuviokoodi
RBAC	Role-Based Access Control, roolipohjainen pääsynhallinta
REST	Representational State Transfer, HTTP-protokollaan perustuva arkkitehtuurimalli ohjelmointirajapintojen toteuttamiseen
RFID	Radio-Frequency Identification, radiotaajuinen etätunnistus
SaaS	Software as a Service, tarkoittaa ohjelmiston käyttöä palveluna
TCP	Transmission Control Protocol, tietoliikenneprotokolla tietokoneiden väliseen tiedonsiirtoon
TLS	Transport Layer Security, salausprotokolla, jolla voidaan suojata web-sovellusten tietoliikenne IP-verkoissa
UDP	User Datagram Protocol, yhteydetön tietoliikenneprotokolla, joka ei vaadi yhteyttä laitteiden välille
UI	User Interface, käyttöliittymä, jolla käyttäjä käyttää järjestelmää
WSN	Wireless Sensor Networks, alueellisesti hajautettujen antureiden ryhmä, joka valvoo ja tallentaa ympäristön fyysisiä olosuhteita
WWW	World Wide Web, Internet-verkossa toimiva hajautettu hypertekstijärjestelmä
XLS	Microsoft Excel -sovelluksen tiedostomuoto

1 Johdanto

1.1 Työn taustoja

Kaupungit ovat aina pyrkineet kohtaamaan asukkaittensa kysynnän tarjoamalla palveluja, jotka tukevat ja parantavat asukkaiden elämänlaatua. Vuoteen 2050 mennessä maailman väestön lukumäärän on arvioitu kasvavan melkein 10 miljardiin ja arviolta 80% väestöstä tulee asumaan kaupungeissa. Nopea väestönkasvu asettaa haasteita kaupunkien jatkuvalle kehittämiselle, niiden energiakulutukselle, ihmisten liikkumiselle, vesi- ja jätehuollolle, infrastruktuurille, yleiselle turvallisuudelle sekä terveydenhuollon järjestämiselle. [1] [2]

Vaikka kaupunkien jatkuva kehittyminen onkin haastavaa, mahdollistaa se uusien teknologioiden käytön ja siten uusien innovatiivisten ratkaisujen luomisen. Vuosien saatossa olemme laajentaneet digitaalisen tiedon avulla fyysistä ympäristöämme ja siten muokanneet työskentely- ja vuorovaikutustapojamme, joilla on ollut myös suuri vaikutus eri aloihin, kuten terveydenhuoltoon, ympäristövalvontaan, kaupunkijärjestelmiin sekä eri hallinta- ja ohjausjärjestelmiin. Teknologian avulla on myös kyetty paljastamaan monia haasteita, jotka vaikuttavat asukkaiden elämään. [3]

Älykäs kaupunki käyttää hyväkseen uusimpia teknologioita ja pyrkii siten parantamaan palveluitaan sekä ratkaisemaan kaupungin ongelmakohtia. Sen päätavoitteina ovat kehittää päätöksentekokykyä, pienentää päästöjä, parantaa sosiaalista ja taloudellista laatua sekä vankistaa asukkaiden yhteisöllisyyden tunnetta. Älykkäässä kaupungissa parannetaan esimerkiksi liikennettä ja palveluiden saavutettavuutta, edistetään kestävää kehitystä sekä toimitaan vuorovaikutuksessa asukkaiden kanssa. [4]

Joukko Suomen suurimpia kaupunkeja ovat yhdessä alkaneet suunnitella kaupunkien kestävää kehittämistä yhteisellä 6Aika-strategialla. Näihin kuuluvat Helsinki, Espoo, Vantaa, Tampere, Turku ja Oulu. Strategian avulla pyritään ratkaisemaan kaupunkilaistumisen haasteita ja kehittämään palveluita vastaamaan paremmin asukkaiden tarpeita. Strategian kehityskohteina ovat kiertotalous ja energia, liikkuminen, oppiminen, terveys ja hyvinvointi, Smart City -ratkaisut, Pk-yritysten osaaminen, työllisyys ja osaaminen sekä kärkihankkeet. [5] Tulevaisuudessa kaupungit ja niiden palvelut rakentuvat langattomuuden ja tehokkaan tietojen hyödyntämisen pohjalle. Kaupungit investoivat erilaisiin sensoriteknologioihin ja etsivät Esineiden Internetin soveltamismahdollisuuksia palvelujen kehittämiseksi. [6]

1.2 Diplomityön tavoitteet, käytetty menetelmä sekä työn rajaukset

Tämän diplomityön tavoitteena on vertailla keskenään erilaisia avoimeen lähdekoodiin pohjautuvia IoT-alustoja ja arvioida vertailun pohjalta niiden soveltuvuutta CityIoT-hankkeeseen. Apuna vertailussa käytetään hankkeen alkuperäisistä vaatimuksista johdettua erillistä dokumentaatiota [7], joka pitää sisällään tietoturvavaatimukset hankkeessa toteutetulle IoT-alustalle. Lisäksi alustojen tietoturvaa tarkastellaan alalla toimivien tahojen tuottamia ohjeistuksia vasten. Myös hankkeen referenssitoteutus tullaan tarkastelemaan vertailtavien alustojen joukossa. Hankkeen aikana on nähty tarpeen tarkastella tarjolla olevien IoT-alustaratkaisujen tietoturvaominaisuuksia ja siksi tässä työssä keskitytään vain IoT-alustojen tietoturvaominaisuuksien vertailuun, eikä oteta kantaa esimerkiksi niiden arkkitehtuuriin, käytettävyyteen tai ylläpidettävyyteen.

1.3 Diplomityön rakenne

Tämän diplomityön rakenne noudattaa seuraavan kaltaista järjestystä, johdanto kappaleen jälkeen: kappaleessa 2 käsitellään työn taustoja tutustumalla Esineiden Internet -käsitteeseen, IoT-järjestelmiin, keskeisimpiin tiedonsiirtoprotokolliin sekä tietoturvaan ja -uhkiin. Lisäksi kappaleessa esitellään diplomityössä vertailtavat IoT-alustat. Kappaleessa 3 esitellään CityIoT-hanke ja sen tietoturvavaatimukset sekä hankkeen aikana toteutettu referenssitoteutus. Kappaleessa esitellään myös alalla tunnettujen tahojen tuottamia tietoturvaohjeistuksia, joita käytetään työn vertailussa. 4. kappaleessa tehdään varsinainen vertailu valittuja IoT-alustoja hankkeen tietoturvavaatimuksia vasten sekä verrataan alustojen tietoturvaa alalla toimivien tahojen tuottamiin ohjeistuksiin. Lopuksi 5. kappaleessa esitellään vielä vertailun tulokset ja kappaleessa 6 pohditaan vertailtujen IoT-alustojen sopivuutta hankkeeseen, saatujen tulosten pohjalta. Viimeisimpänä esitetään lähdeluettelo ja liitteet.

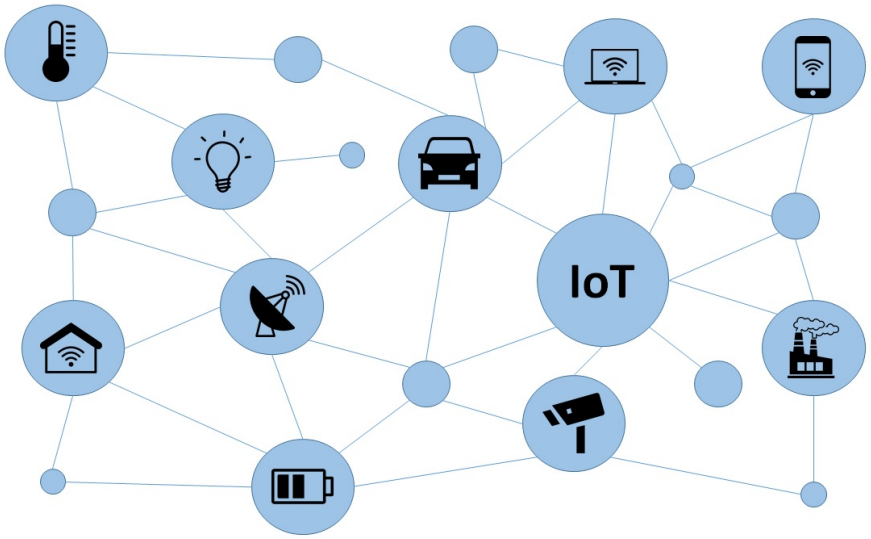
2 IoT:n taustaa ja vertailtavat alustat

Tässä kappaleessa tutustutaan Esineiden Internet -käsitteeseen sekä käydään läpi IoT:n liittyviä asioita teoreettisella tasolla. Lisäksi kerrotaan tietoturvaan liittyvistä haasteista joita IoT-järjestelmissä kohdataan. Kappaleen lopussa on esittely työn vertailussa käytetyistä IoT-alustoista.

2.1 Esineiden Internet

Esineiden Internet on tietojenkäsittelyn konsepti, minkä avulla jokapäiväiset fyysiset laitteet voidaan yksilöidä ja liittää Internetiin. IoT muodostuu verkkolaitteista, jotka kommunikoivat keskenään ja keräävät, että tallentavat tietoa ympäristöstään ilman, että ihmisen tarvitsee tehdä asialle juuri mitään. Esineiden Internet -ekosysteemi koostuu verkkoon liitetyistä älykkäistä esineistä, laitteista, älypuhelimista, tablet-tietokoneista sekä muista älylaitteista. Laitteet käyttävät sisäiseen viestintään radiotaajuista etätunnistusta (RFID), QR-koodeja, sensoreita tai langattomia teknologioita. [8] [9]

IoT-laitteet jakavat keräämänsä datan IoT-yhdyskäytävään (tai muuhun verkkolaitteeseen), josta tiedot lähetetään IoT-alustalle tallennettavaksi tai suoraan pilvipalvelulle analysoitavaksi. Kuvassa 2.1 on havainnollistettu abstraktilla tasolla erilaisten laitteiden, esineiden ja järjestelmien avulla millaisia IoT-ekosysteemit voivat olla. IoT-järjestelmän ytimenä toimii IoT-alusta.



Kuva 2.1 *Havainnollistamiskuva IoT-ekosysteemistä*

IoT:n tavoitteena on tehdä ihmisten arkipäivästä turvallisempaa ja tehokkaampaa. Tyypillisiä tämän päivän esimerkkejä ovat valaistuksen säätö automaattisesti, älypuhelimella tai puheohjauksella [10], sähkömittarin etäluku [11], sähköautojen latausasemien käyttö [12] tai vaikkapa mökin saunan kiukaan lämmittäminen kotoa käsin [13]. Teollisuudessa antureilla voidaan esimerkiksi mitata ja seurata eri laitteiden huoltotarvetta ja siten tilata huolto tarvittaville laitteille hyvissä ajoin. Kemikaalitehdasalueen kaasupitoisuuksien nousua seuraamalla voidaan suorittaa hälytyksiä, jos huomataan vaarallisia kaasupitoisuuksia ilmassa. Kiinteistöjen kunnossapitoa voidaan puolestaan seurata olosuhdetietojen avulla (ilmanlaatu, hiilidioksidimäärät, ilmankosteus). [14]

IoT-järjestelmät ja ratkaisut jatkavat kasvuaan voimakkaasti ja toimittajat kohtaavat yhä enenevissä määrin asiakkaiden ja yritysten tarpeita. Hälytyslaitteet ja teknologiat kuten äly-kaiuttimet, koneoppiminen ja 5G mahdollistavat tehokkaan järjestelmien hallinnan kodeissa ja työpaikoilla. IoT-teollisuuden kasvu vauhdittaa modernisointia ja on arvioitu, että IoT-markkinoiden vuotuinen kasvu ylittäisi 2.4 miljardia Yhdysvaltain dollaria vuoteen 2027 mennessä. [15]

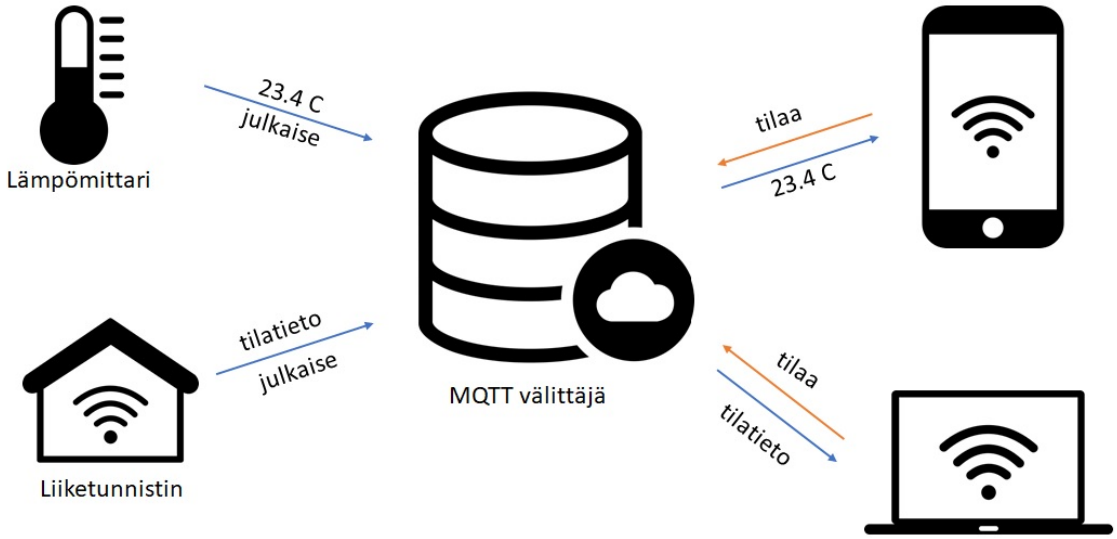
2.2 Yleisimmät protokollat

Tässä osiossa esitellään IoT-järjestelmien yleisimmin käytetyt yhteysprotokollat. Näiden yhteysprotokollien avulla välitetään tietoa eri laitteiden ja sovellusten välillä.

2.2.1 MQTT

Message Queuing Telemetry Transport on julkaise/tilaa -viestinvälitysprotokolla laitteiden välille. Se on kevyt, avoin, yksinkertainen ja helppokäyttöinen. Ominaisuuksiensa ansiosta se soveltuu käytettäväksi monissa järjestelmässä, mutta ennen kaikkea se soveltuu viestinvälitysprotokollaksi ”koneelta koneelle” -viestinnässä (M2M). Se on hyödyllinen IoT-ympäristöissä, joissa kooltaan pienet ohjelmistokoodit ovat tarpeen ja verkkokapasiteetin suuri käyttö ei ole toivottua niistä aiheutuvien kustannusten vuoksi. [16]

MQTT:ssä julkaisijalla ei ole tietoa tilaajasta, joten yhteydet asiakkaiden välillä hoidetaan aina keskitetysti välittäjän (broker) avulla. Välittäjä pitää kirjaa siitä kuka asiakas välittää tietoa aihepiireittäin (topics). Asiakassovellukset saavat tietoa tilaamalla haluamansa aihepiirin, näin tiedon päivittyessä asiakassovellus saa aina viimeisimmän tiedon itselleen. [17] Kuvassa 2.2 lämpömittari julkaisee lämpötilatiedon välittäjälle. Asiakassovellus tilaa välittäjältä aihepiirin, joka pitää sisällään lämpötilamittarin lämpötilatiedon. Lopuksi lämpötilatieto voidaan esittää päätelaitteen avulla käyttäjälle.

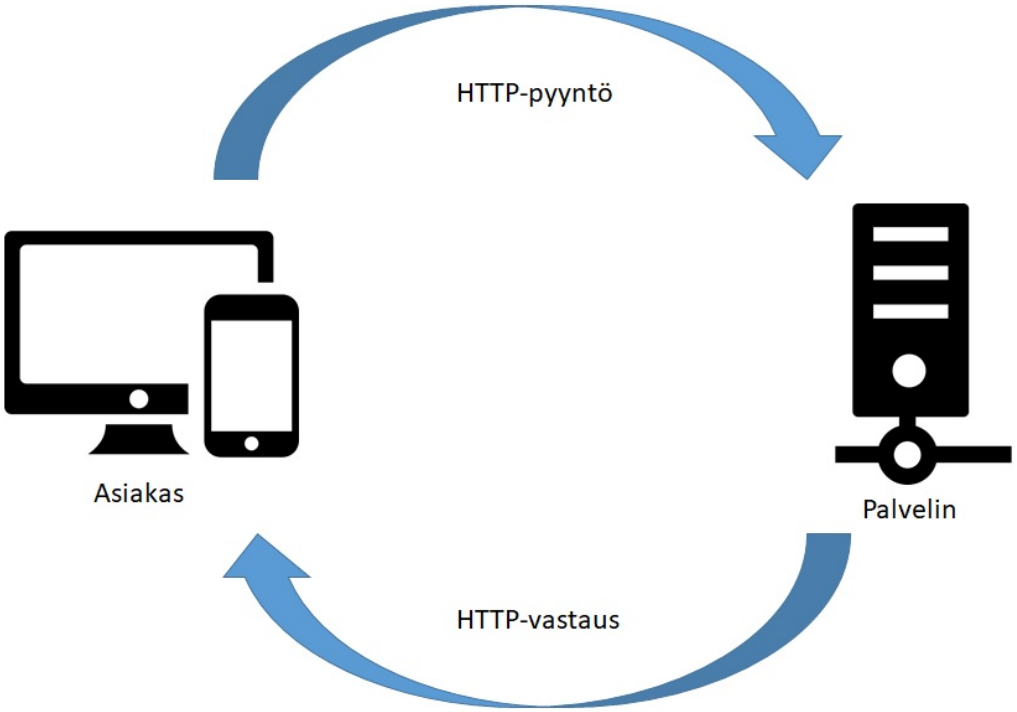


***Kuva** 2.2 MQTT-protokolla*

2.2.2 HTTP

Hypertext Transfer Protocol on pyydä/vastaa-viestinvälitysprotokolla asiakkaan ja palvelimen välille. Kuvassa 2.3 asiakas lähettää verkon yli palvelimelle pyynnön, joka sisältää tiedon tarvittavista muuttujista, tietoja asiakkaasta sekä itse viestin. Palvelimen vastaus sisältää tilatiedon (onnistunut pyyntö tai virhekoodin), tietoja palvelimesta, erilaisia metatietoja sekä itse vastausviestin.

HTTP on sovellustason protokolla hajautetuille ICT-järjestelmille. HTTP:tä on yleisesti käytetty WWW-palvelimien ja selainten väliseen kommunikointiin maailmanlaajuisesti, mutta se soveltuu myös muuhun verkkoliikenteeseen, kuten esimerkiksi IoT-verkossa olevien laitteiden väliseen kommunikointiin. [18]



Kuva 2.3 *HTTP-protokollan käyttö*

HTTP-protokollassa on määritelty eri metodeita, joita käytetään palvelimelle lähetetyissä pyynnöissä osoittamaan millä toiminnolla tiettyä resurssia kutsutaan. Taulukossa 2.1 on esiteltynä HTTP/1.1-protokollan kaikki eri metodit.

Taulukko 2.1 *HTTP-protokollan metodit [18]*

Metodi	**Kuvaus**
GET	Käytetään resurssin hakemiseen palvelimelta
POST	Käytetään tietojen lähettämiseen kohdistettuun resurssiin
TRACE	Käytetään toistamaan asiakkaan lähettämät tiedot
PATCH	Käytetään resurssin osittaiseen päivittämiseen
PUT	Käytetään korvaamaan valittu resurssi
HEAD	Käytetään hakemaan resurssi, joka on identtinen GET-pyynnön resurssin kanssa, mutta ilman vastausviestin runko-osaa
DELETE	Käytetään määritetyn resurssin poistamiseen
OPTIONS	Käytetään kuvaamaan resurssin tuettuja HTTP-menetelmiä
CONNECT	Käytetään tunnelin luomiseen kohderesurssin määrittelemälle palvelimelle

2.2.3 CoAP

Constrained Application Protocols on IETF:n standardoima uudelleenlähetysprotokolla, joka sisältää pyyntö- ja vastausviestejä. CoAP on suunniteltu resurssirajoitteisille laitteille, jotka toimivat langattomissa anturiverkoissa (WSN). CoAP:n toimintoja on seuranta, etälaitteiden hallinta ja viestien toimittamisen takaaminen hyödyntäen HTTP-komentoja (GET, POST, PUT ja DELETE) asiakkaan ja palvelimen väliseen kommunikoimiseen. Protokolla hyödyntää UDP-protokollaa ja se on jaettu kahteen alikerrokseen, viesti- ja pyyntö-vastaus alikerrokseen. Ensin mainittu takaa luotettavan viestinnän ja jälkimmäinen hoitaa REST-viestinnän.

CoAP tarjoaa DTLS-palvelua, joka mahdollistaa luottamuksellisen viestinnän, vähentää viestinnän kuormittumista ja seuraa resursseja julkaisu-tilausmekanismin avulla. CoAP:n neljä eri viestityyppiä ovat *vahvistettavissa*, *ei-vahvistettavissa*, *kuittaus* ja *erillinen vastaus*. Kuvassa 2.4 esitetään CoAP-viestin formaatti, missä sen otsikkotiedoissa ilmenee CoAP:n versio (V), siirtotapa (T), valintojen (options) lukumäärän (OC), koodi (Code) ja viestitunnus (M ID) joiden perään tulevat Token-, Options- ja Payload-kentät. [19]

Kuva 2.4 *CoAP-viestin formaatti [19]*

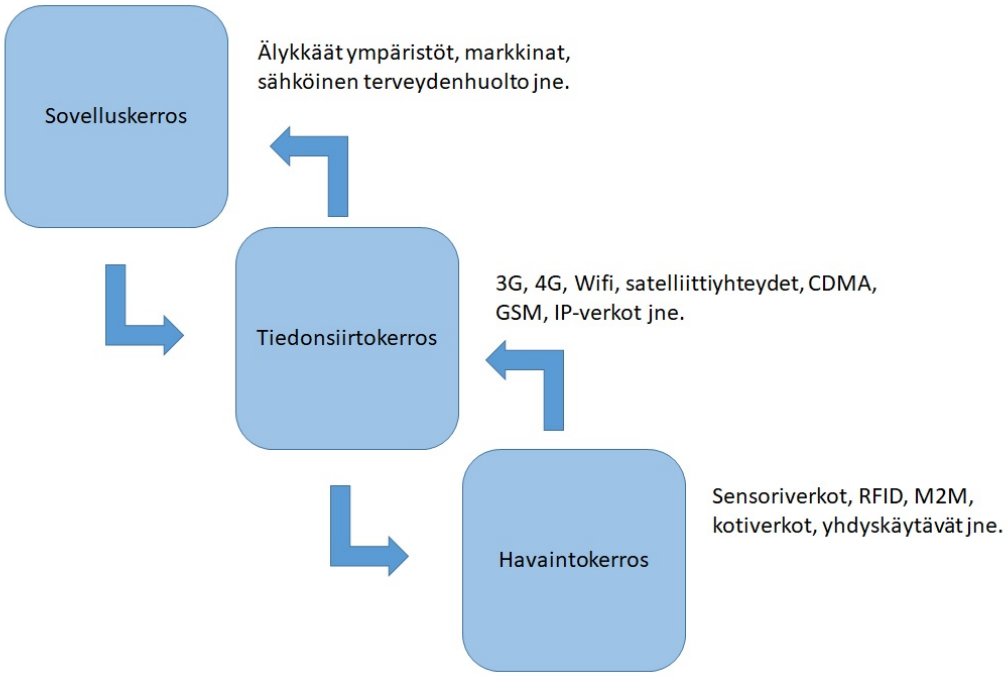
2.3 IoT:n ja älykkäiden kaupunkien tietoturva

Tietoturvalla tarkoitetaan tietojen luottamuksellisuuden, eheyden ja saatavuuden suojaamista. Luottamuksellisuudella tarkoitetaan tietojen suojaamista siten, ettei niitä aseteta saataville tai altisteta luvattomille henkilöille, osapuolille, prosesseille tai järjestelmille. Eheydellä tarkoitetaan puolestaan sitä, että tieto pysyy sellaisena kuin se on alun perin tarkoitettu. Saatavuudella varmistetaan pääsyoikeudet tietoihin vain niille henkilöille ja tahoille, joille siihen on oikeus. Tietoturvaan voidaan myös liittää muitakin ominaisuuksia kuten aitous, vastuullisuus, kiistämättömyys ja luotettavuus. [20]

Verkkohyökkäyksillä pyritään häiritsemään verkkopalveluita ja väärentämään, tai tuhoamaan palveluiden tärkeitä tietoja. Hyökkäykset tapahtuvat yleensä verkkoon kytkettyjen tietokoneiden ja laitteiden kautta. Viime vuosina tietoverkkorikollisuus on aiheuttanut merkittäviä taloudellisia vahinkoja ja tämän takia tietokonejärjestelmien tietoturva on todella tärkeää nykyaikaisessa elämässä ja yhteiskunnassa. [21]

IoT-järjestelmä voi sisältään suuren määrän heterogeenisiä laitteita, jonka vuoksi turvallisuus ja yksityisyys ovat kriittisiä asioita näissä järjestelmissä. Myös tallennettujen tietojen luottamuksellisuus ja eheys on varmistettava. Tämän lisäksi on käytettävä hyväksi todettuja todennusmekanismeja, jotta luvattomat tahot eivät pääsisi käyttämään järjestelmää väärin. Toisaalta järjestelmässä pitää pystyä takaamaan käyttäjien yksityisyys, nimettömyys sekä tietosuoja. [22]

Geneerisen IoT-järjestelmän arkkitehtuuri pitää sisällään kolme kerrosta: havainto-, tiedonsiirto- ja sovelluskerros. IoT:n turvallisuuden parantamista tulisi soveltaa kaikissa kolmessa kerroksessa, ottaen huomioon kukin haavoittuvuus ja mahdolliset hyökkäykset kullekin tasolle. [23] Kuvassa 2.5 on havainnollistettu geneerisen IoT-järjestelmän arkkitehtuuri ja kerrokset. Kuvan jälkeen käydään läpi kunkin kerroksen haasteet.



Kuva 2.5 *IoT-arkkitehtuurin kerrokset [22]*

Havaintokerros kerää dataa fyysisen laitteen kuten RFID-lukijan, anturin, GPS-laitteen tai jonkin muun laitteen välityksellä ja muuntaa saadun tiedon fyysisestä ympäristöstä digitaaliseen muotoon. Näillä laitteilla on yleensä rajalliset resurssit muistin- ja virrankäytön suhteen, jonka vuoksi tietoturvajärjestelmän asentaminen laitteeseen on vaikeaa. Tästä huolimatta, palvelunestohyökkäyksien ja anturin keräämän datan eheyden, aitouden ja luottamuksellisuuden turvaaminen on kuitenkin tarpeellista.

Ulkopuolisen tahon pääsyn laitteeseen voidaan estää vaatimalla käyttäjän todentamista ja sen avulla oikeuttaa pääsy laitteeseen. Tiedon luottamuksellisuuden varmistamiseksi tiedonsiirron yhteydessä laitteiden ja järjestelmien välillä on välttämätöntä käyttää salattuja yhteyksiä sekä tarvittaessa salata välitettävät tiedot kevyillä salaustekniikoilla rajallisten käyttöresurssien takia. Tiedon eheys ja aitous ovat järjestelmän kannalta tärkeitä asioita ja tämän vuoksi ne tulisi suojata hyvin.

Esimerkiksi RFID-järjestelmissä tunnisteisiin voidaan päästä käsiksi todennusmekanismien puuttuessa ja siten mahdollistetaan hyökkääjälle pääsy laitteen tietoihin. Jos hyökkääjä pääsee käsiksi laitteen tietoihin, hän voi lukea, muokata tai poistaa tietoja sekä tehdä kopioita tunnisteista. Tunnisteen lukija ei osaa tunnistaa onko tunniste aito vai väärennetty ja siten voi altistua saastuneen tunnisteen lukemiseen. Hyökkääjä voi huijata järjestelmää luulemaan olevansa osa järjestelmää ja saada tällä tavoin pääsyn järjestelmän muihin toiminnallisuuksiin ja saada mahdollisuuden aiheuttaa muuta tuhoa.

RFID-järjestelmissä voidaan myös suorittaa salakuuntelua järjestelmän langattomia ominaisuuksia hyväksikäyttäen. Hyökkääjä voi haistella tietoliikennettä tunnisteen ja lukijan välillä ja siten vaarantaa tiedon aitouden. Saastuneen tunnisteen avulla on mahdollista lähettää häiriösignaalia järjestelmään (DoS-hyökkäys) ja siten aiheuttaa järjestelmän toimimattomuutta tai kaatumisen. [22]

Tiedonsiirtokerroksen tehtävänä on välittää havaintokerroksen lähettämä tieto luotettavasti eteenpäin. Tiedonsiirrossa käytetään yleisiä verkkoja (matkapuhelinverkkoja, langattomia ja kiinteitä verkkoja, suljettuja verkkoja) ja siksi tiedonsiirtoprotokollat ovat tärkeässä asemassa laitteiden välisessä tiedonsiirrossa. Tiedonsiirtokerros koostuu langattomasta anturiverkosta (WSN), jonka vastuulla on siirtää data luotettavasti anturista määränpäähän ja sen ominaisuuksien avulla voidaan tiedonsiirrolla tehdä lähes täydellinen suojaus. Silti Man-in-the-Middle-hyökkäykset ja väärennetyn datan lähettäminen ovat mahdollista, koska verkoissa kulkee todella suuria määriä dataa. Tästä syystä tämän kerroksen turvamekanismit ovat erittäin tärkeitä koko IoT-järjestelmälle.

Tiedonsiirtokerroksessa olemassa olevien tietoturvamekanismien soveltaminen on vaikeaa verkossa olevien laitteiden resursseista ja verkkoratkaisuista johtuen.

Tunnisteiden käyttö todentamisessa vaaditaan, jotta ulkopuolisten pääseminen verkkoon voidaan estää sekä datan luottamuksellisuus ja sen eheys voidaan varmistaa. Hajautetut palvelunestohyökkäykset ovat yleisiä ja vakava uhka IoT-verkoissa. Tämän vuoksi niitä vastaan tulisi varautua ratkaisemalla kuinka hyökkäys saadaan pysäytettyä ja verkko palautettua normaaliin tilaan hallitusti mahdollisen hyökkäyksen jälkeen.

Hyökkääjä voi aiheuttaa häiriötä esiintymällä useampana identiteettinä ja siten häiritä jotain tiettyä verkon solmupistettä, eli tehdä ns. Sybil-hyökkäyksen, jonka avulla heikentää IoT-verkon toimintaa. Lisäksi laitteet joiden virransaanti on rajallinen ovat alttiita univajehyökkäyksille. Tämänkaltaiset laitteet on suunniteltu olemaan aika-ajoin sammuksissa säästääkseen pattereita tai akkua. Univajehyökkäyksen aikana laitetta ei kuitenkaan päästetä sammumaan vaan sitä pidetään hereillä, kunnes energiavarasto on kulutettu loppuun ja laite sammuu. Myös DoS-hyökkäykset ovat mahdollisia. Niissä IoT-verkko ruuhkautetaan suurella verkkoliikennemäärällä ja siten saadaan verkko käyttökelvottomaksi. Man-in-the-Middle-hyökkäys puolestaan mahdollistaa verkon salakuuntelun tai solmupisteiden välisen kommunikaation hallinnan. Vakavimpana hyökkäyksenä voidaan pitää haitallisen koodin injektiohyökkäystä, missä hyökkääjä pääsee solmupisteen kautta asentamaan haitallista koodia järjestelmään ja siten mahdollistamaan hyökkääjälle IoT-verkon täydellisen hallinnan. [22]

Sovelluskerros mahdollistaa personoidut palvelut, jotka vastaavat käyttäjien tarpeita. Kerroksen käyttöliittymät mahdollistavat käyttäjien pääsyn järjestelmään tietokoneen tai mobiililaitteen avulla. Suojaustarpeet vaihtelevat sovellusympäristöstä riippuen ja tiedonjako-ominaisuudet aiheuttavat monia ongelmia tietosuojan toteutumiseen, pääsynvalvontaan sekä tietojen paljastumiseen.

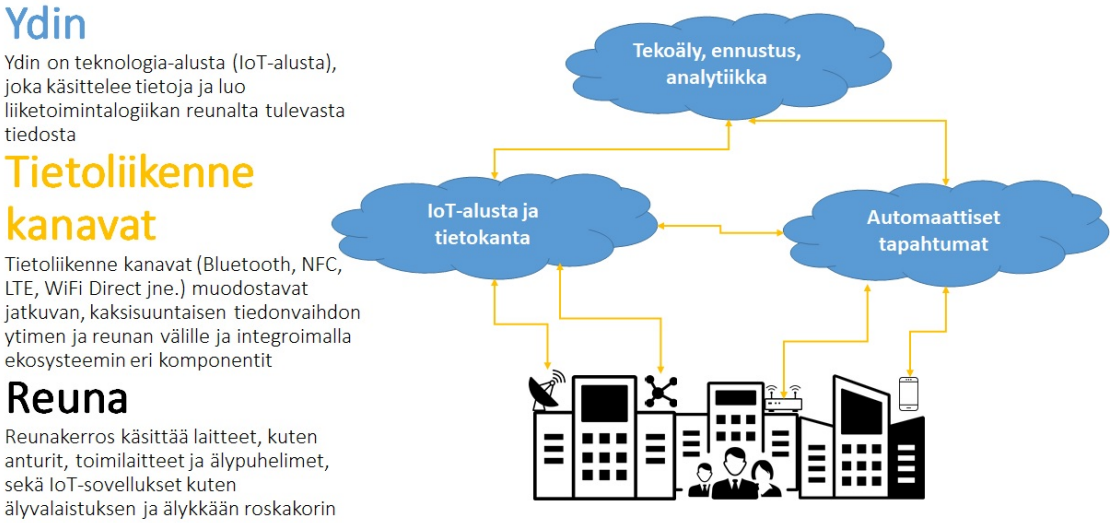
Tietoturvahaasteiden ratkaisemiseksi tulee käyttää todentamista ja yhtenäisiä sopimuksia heterogeenisessä verkossa sekä huolehtia käyttäjän yksityisyyden suojaamisesta. Merkittäviä asioita ovat myös käyttäjien tietoturvakoulutukset sekä -hallinta, joilla varmistetaan esimerkiksi oikeanlaisten salasanojen käyttö järjestelmässä.

Haitallisen koodin injektiohyökkäyksien avulla, hyökkääjä voi suorittaa haitallista koodia järjestelmässä ja sen avulla saada järjestelmän käyttäjät varastamaan tärkeitä tietoja tai mahdollistamaan pääsyn järjestelmään hyökkääjän puolesta. Kohdennetun DoS-hyökkäyksen avulla voidaan haitata tietoturvajärjestelmän toimivuutta ja siten vaarantaa käyttäjän tietosuojaa. Myös verkkoliikenteen skannauksella voidaan saada tietoa järjestelmästä ja käyttäjistä sekä aiheuttaa häiriöitä järjestelmään. Verkkourkinnan (kuten sähköpostihuijauksen) avulla hyökkääjä voi kalastella järjestelmän käyttäjistä tietoja ja siten päästä käsiksi itse järjestelmän tie-

toihin. [22]

Kaupunkien digitalisoituminen tuovat kaupungeille uudenlaisia uhkia, kyberhyökkäysten muodossa. Kyberhyökkäyksiä raportoidaan yhä useammin ja niiden tilastot nousevat vuosittain. Viime vuosina kyberhyökkäyksiä on alettu kohdistamaan datan lisäksi myös fyysiseen omaisuuteen. Seurauksina voivat olla tietovuotojen, taloudellisten menetysten ja maineen vahingoittumisen lisäksi myös häiriöt keskeisissä kaupunkipalveluissa ja -infrastruktuurissa, jotka vaikuttavat moniin eri aloihin. Tällaisia aloja ovat mm. terveydenhuolto, julkinen liikenne, lainvalvonta, sähkön jakelu sekä asumista tukevat palvelut. [24]

Älykäs kaupunki on siis kompleksinen ekosysteemi, mikä muodostuu kuntapalveluista, julkisista ja yksityisistä palveluista, ihmisistä, prosesseista, laitteista ja infrastruktuurista. Kaikki nämä osa-alueet ovat jatkuvassa vuorovaikutuksessa keskenään. Taustalla toimiva teknologia voidaan jakaa kolmeen kerrokseen, jotka muistuttavat geneerisessä IoT-järjestelmässä olevia kerroksia: reuna, ydin ja tietoliikennekanavat. Reuna-kerros pitää sisällään laitteet kuten anturit, toimilaitteet, IoT-laitteet sekä älypuhelimet. Ydin puolestaan on teknologia-alusta, joka prosessoi ja järjestelee reuna-kerrokselta tulevan datan. Tietoliikennekanavat muodostavat jatkuvan kaksisuuntaisen tiedonvälityksen reunan ja ytimen välille. [24] Kuvan 2.6 avulla on havainnollistettu älykkään kaupungin ekosysteemiä.



Kuva 2.6 *Älykäs kaupunki, ekosysteemi [24]*

Suuren tiedonsiirtotarpeen ja muuttuvien prosessien vuoksi jo valmiiksi kompleksisessa ympäristössä kohdataan uusia kyberuhkia teknologia infrastruktuurin ympärillä. Kaupungit kohtaavat ongelmia, kuten kuinka määritellä sisäinen ja julkinen

tieto, mikä tieto on kaupallista tai salattua, mitä tietoa laitteet keräävät ja kuinka sitä tallennetaan, arkistoidaan, monistetaan ja tuhotaan. Lisäksi standardien ja käytäntöjen puuttuminen aiheuttaa sen, että kaupungit joutuvat tekemään yhteistyötä eri järjestelmätoimittajien kanssa. Tämä aiheuttaa ongelmia järjestelmien yhteensopivuuden kanssa ja hankaloittaa siten järjestelmien integrointia aiheuttaen mahdollisia lisäuhkia. [24]

2.4 IoT-alustat

IoT-verkoissa kerättyä dataa pitää pystyä analysoimaan, jotta sen perusteella voidaan muodostaa päätöksiä automaattisesti. Apuna tässä voidaan käyttää IoT-alustaa, minkä avulla suuren tietomäärän hallinta helpottuu.

IoT-alusta on IoT-järjestelmän ydin ja sen avulla hallitaan kaikkia laitteita, yhteyksiä, ohjelmisto- ja sovelluskerroksia. Yleisesti voidaan sanoa, että IoT-alustan päätarkoitus on yhdistää eri teknologiakerrokset saumattomasti yhteen. Se tarjoaa tehokkaan tavan laitehallintaan ja konfigurointiin, tiedon keräämiseen ja analysointiin sekä mahdollistaa erilaisten sovellusten käytön pilvi- tai paikallisessa ympäristöissä. [25]

IoT-alustoja on toteutettu monella eri tavalla ja ne soveltuvat vaihtelevasti eri yritysten sekä toimialojen tarpeisiin. [26] Jotta IoT-alusta pystyisi toteuttamaan todellisen E2E-kokemuksen useimmissa tapauksissa, tulisi sen arkkitehtuurista löytyä kuvan 2.7 mukaiset komponentit. Nämä komponentit on esitelty tarkemmin seuraavissa kappaleissa.

Kuva 2.7 *IoT-alustan tärkeimmät arkkitehtuurikomponentit [27]*

1. **Yhteydet ja normalisointi**: mahdollistaa erilaisten protokollien ja tiedostomuotojen käytön API-rajanpinnassa sekä takaa tiedonsiirron ja vuorovaikutuksen kaikkien laitteiden kanssa.

2. **IoT-laitteiden hallinta**: varmistaa, että järjestelmään liitetyt laitteet toimivat saumattomasti mm. suorittamalla laitteiden ohjelmistokorjauksia ja -päivityksiä.

3. **Tietovarasto**: skaalautuva tallennus nostaa tietokantojen vaatimukset korkealle tasolle datan määrän, monimuotoisuuden, nopeuden ja reaaliaikaisuuden takia.

4. **Tapahtumien hallinta ja prosessointi**: Muokkaa dataa sääntöpohjaisilla tapahtumilla, jotka mahdollistavat ”älykkäiden”-toimintojen suorittamisen kerätystä tiedosta.

5. **Tiedon visualisointi**: dataa voidaan kuvata erilaisten kaavioiden, 2D- tai 3D-mallien avulla.

6. **Analytiikka**: datalle voidaan suorittaa monimutkaisia analyysejä sekä tehdä ennakoivaa analytiikkaa koneoppimisen avulla.

7. **Sovelluslaajennukset**: mahdollistaa mm. kehittäjien protoilun ja testauksen sekä uusien käyttötapausten luomisen alustan ekosysteemin avulla. Esimerkiksi käyttäjähallinta, raportointi sekä erilaisten hälytysten ja ilmoitusten lähettäminen voidaan tuoda sovelluslaajennoksina järjestelmään.

8. **Ulkoiset rajapinnat**: integroituminen kolmannen osapuolten järjestelmiin ja muihin laajempiin ITC-järjestelmiin onnistuu sisäänrakennettujen sovellusohjelmointirajapintojen (API), ohjelmistokehityspakettien ja yhdyskäytävien avulla (esim. ERP, CRM).

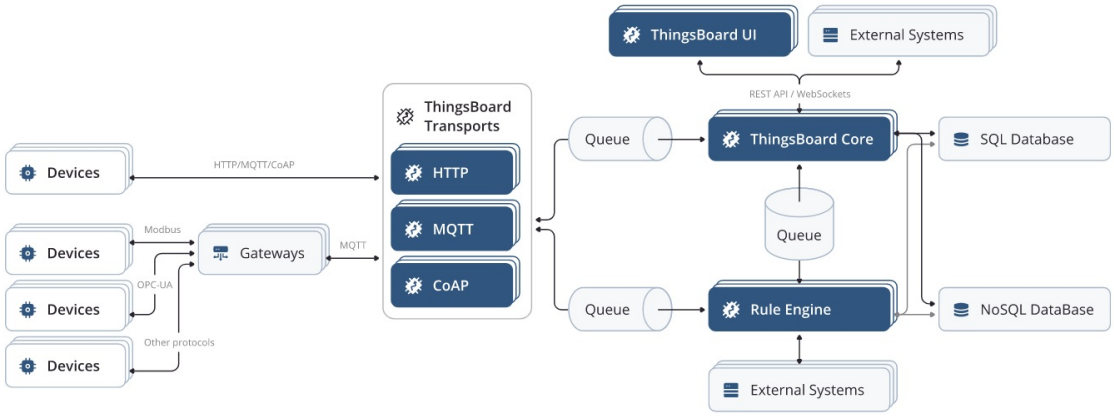
Seuraavaksi on tähän työhön valittujen IoT-alustojen esittely. IoT-alustojen valintakriteereinä olivat järjestelmien pohjautuminen avoimeen lähdekoodiin sekä soveltuvuus nimenomaan älykkäisiin kaupunkeihin.

2.4.1 ThingsBoard

ThingsBoard on ThingsBoard-yhtiön kehittämä ja ylläpitämä avoimeen lähdekoodiin pohjautuva IoT-alusta, joka mahdollistaa IoT-projektien nopean kehittämisen, hallinnan ja skaalauksen. Sitä on mahdollista käyttää ulkoisena IoT-pilvipalveluna, tai paikallisena ratkaisuna, joka mahdollistaa palvelinpuolen infrastruktuurin IoT-sovelluksille. ThingsBoardilla voidaan mm. määritellä liitokset laitteiden, eri resurssien sekä asiakkaiden välille. Alustan avulla voidaan myös kerätä ja analysoida dataa laitteilta sekä muilta resursseilta ja tietolähteiltä. Dynaamisilla kojetauluilla voidaan esittää laitteiden tuottamaa dataa käyttäjille.

ThingsBoard IoT-alustasta on olemassa kaksi eri versiota, ilmainen Community-versio sekä maksullinen Professional-versio joka on laajennettu versio ilmaisesta järjestelmästä. Eroavaisuuksia löytyy mm. laiteidenliitäntä protokollista, sääntöketjutoiminnallisuuden komponenteissa, käyttäjienhallinnassa, raportoinnissa ja alustan integraatioissa. [28]

ThingsBoard on suunniteltu olemaan skaalautuva horisontaalisesti ja se hyödyntää avoimen lähdekoodin teknologioita. Jokainen solmupiste klusterissa on suunniteltu olemaan identtinen, eikä yksittäistä vikaantumispistettä (single-point-of-failure) ole, joten ThingsBoard:ssa on hyvä viansietokyky. Yksittäisellä palvelinsolmupisteellä voidaan hallita kymmenestä sataantuhanteen laitetta ja siten yhdellä klusterilla voidaan hallita miljoonia laitteita. Useamman eri jonototeutuksen tuki mahdollistaa todella korkean jonokuorman ja siten korkean suorituskyvyn järjestelmälle. Järjestelmän laajentaminen onnistuu helpoilla kustomointiominaisuuksilla ja siten lisäohjelmien sekä sääntöketjujen muokkaaminen on mahdollista. Kuvassa 2.8 on esitetty järjestelmän avainkomponentit ja niiden rajapinnat. [29]



Kuva 2.8 *ThingsBoard-arkkitehtuuri [29]*

Järjestelmä tukee MQTT-, HTTP- ja CoAP-pohjaisia sovellusrajapintoja lait-

teiden sovelluksille ja laiteohjelmistoille. Jokaisen protokollan sovellusrajapinta tarjotaan erillisen komponentin kautta ja ne yhdessä muodostavat järjestelmän viestinvälityskerroksen. MQTT-protokollaa varten on myös toteutettu erillinen sovellusrajapinta, joka mahdollistaa yhdyskäytävän kautta useamman laitteen, tai anturin liittämisen järjestelmään.

Viesti vastaanotetaan laitteelta ja parsitaan järjestelmälle sopivaan muotoon ja laitetaan jonoon. Laite saa kuittauksen onnistuneesta viestin lähettämisestä vasta kun viesti on saatu onnistuneesti laitettua jonoon. Arkkitehtuurikuvassa 2.8 Transports-lohko kuvastaa järjestelmän viestien välitystä.

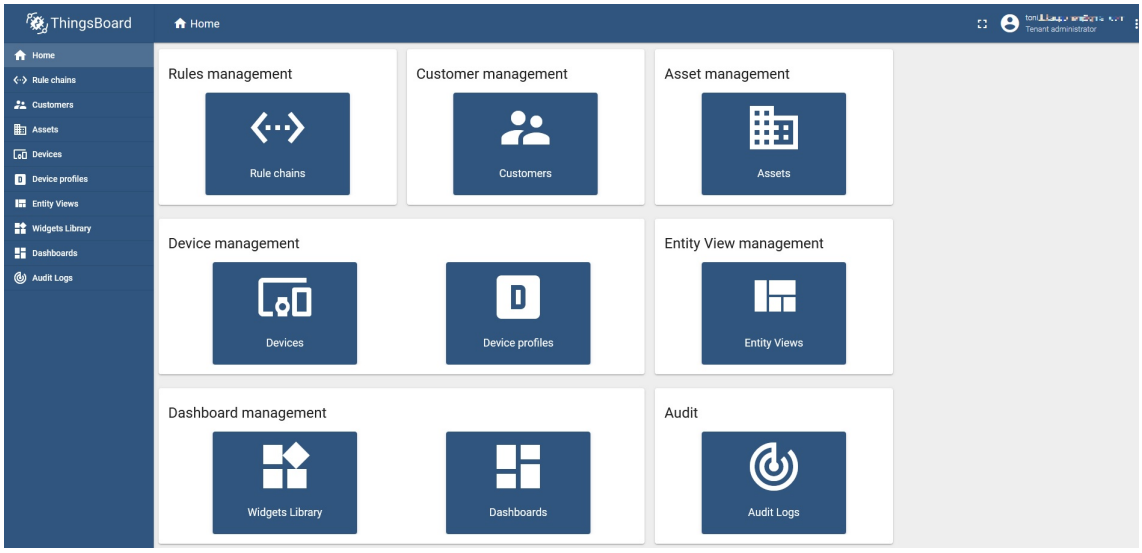
Järjestelmän ydin pitää huolen ulkoisten rajapintakutsujen ja WebSocket-tilausten käsittelystä. Lisäksi se vastaa aikatiedon tallentamisesta aktiivisten ja valvonnan alla olevien laitteiden tiloista. Ydin käyttää Actor System -järjestelmää, jolla se toteuttaa alustan tärkeimmät kokonaisuudet: liiketoimintayksiköt (tenants) ja laitteet (devices). Kuvassa 2.8 ThingsBoard Core -lohko kuvaa järjestelmän ydintä.

Kuvassa 2.8 esitetty Rule engine -komponentti on järjestelmän tärkeimpiä kokonaisuuksia. Sen vastuulla on prosessoida kaikki järjestelmään saapuvat viestit. Rule engine -komponentin toteuttama toiminnallisuus käyttää Actor System -järjestelmää, jolla se toteuttaa alustan tärkeimmät kokonaisuudet: sääntöketjut (rule chains) ja sääntönoodit (rule nodes).

Rule engine saa laitteilta saapuneen datasyötteen jonoista ja kuittaa viestin vasta prosessoituaan sen. Käytettävänä on useita strategioita, joilla voidaan ohjata järjestystä, tai viestin käsittelyä ja sanoman kuittauksen kriteerejä.

Rule engineä voidaan käyttää kahdella eri tavalla: jaettuna tai eristettynä. Jaettuna Rule engine:llä voidaan prosessoida viestit, jotka voivat kuulua useammalla eri liiketoimintayksikölle ja eristettynä vain tietyn liiketoimintayksikön viestit.

Järjestelmässä on käytettävänä kevyt selainpohjainen hallintatyökalu (ThingsBoard UI), joka on toteutettu Express.js-sovelluskehystä käyttäen. Työkalu on tilaton ja sen konfigurointi on melko vähäistä. Hallintatyökalu käyttää ulkoisen rajapinnan kautta ytimen tarjoamia palveluita. Kuvassa 2.9 on hallintatyökalun etusivunäkymä.



Kuva 2.9 *ThingsBoard-hallintatyökalu*

2.4.2 Kaa IoT

Kaa IoT -alusta on sovellettavissa mihin tahansa yrityksen IoT-projektiin. Järjestelmä tarjoaa laajan valikoiman eri toiminnallisuuksia, jotka mahdollistavat kehittäjiä rakentamaan kehittyneitä sovelluksia älykkäille tuotteille, joustavaa laitehallintaa pilviympäristössä, E2E datan prosessoinnin ja laitteiden datan analysointia. Kaikki alustan toiminnallisuudet on toteutettu mikropalveluina ja järjestelmä pohjautuu joustavaan mikropalveluarkkitehtuuriin. Tämä tarkoittaa sitä, että kaikkia toiminnallisuuksia voidaan muokata joko lisäämällä uusia, tai korvaamalla olemassa olevilla, tai kolmannen osapuolen toteutuksilla. [30]

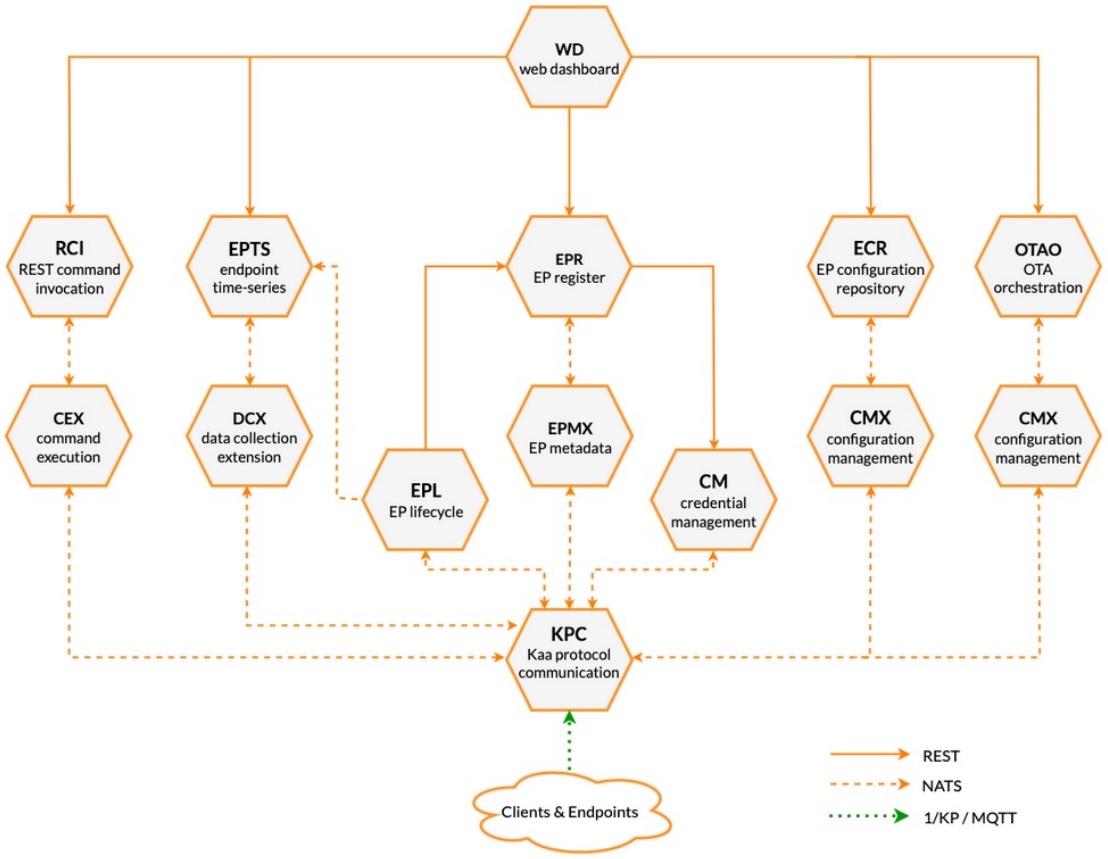
Kaa IoT-alustasta on olemassa myös avoimeen lähdekoodiin pohjautuva versio, mutta sen ylläpito on lopetettu. Toteutusten välillä on paljon eroavaisuuksia, koska järjestelmien arkkitehtuurit ovat täysin erilaisia. Poikkeuksellisesti tässä työssä on tarkisteltu kaupallisen version ilmaisversiota, vaikka se ei olekaan vaatimusten mukainen. Kaa IoT -alusta vaikuttaa mielenkiintoiselta vaihtoehdolta, koska se pitää sisällään joukon avoimeen lähdekoodiin pohjautuvia komponentteja, kuten NATS, Keycloak ja Open Distro.

Kaa IoT on pilvipohjainen IoT-alusta, jonka arkkitehtuuri pohjautuu mikropalvelumenetelmään. Jokainen Kaa-mikropalvelu on itsenäinen kokonaisuus ja niistä voidaan koota joustavasti tarvetta, tai liiketoimintatapausta vastaava kokonaisuus. Mikropalvelut ovat käytännössä mustia laatikoita, jotka toteuttavat niille määritellyn toiminnon, eikä niiden sisäisellä arkkitehtuurilla ole vaikutusta alustan kokonaisarkkitehtuuriin.

Palvelujen välinen kommunikointi tapahtuu HTTP- ja NATS-protokollien avul-

la ja viestit on dekoodattu JSON- ja Avro-muotoon. Nämä tekniikat ovat hyvin määriteltyjä ja niitä käytetään yleisesti monissa nykyajan ohjelmointikielissä, joten mikropalveluiden toteutus ei ole sidottu mihinkään tiettyyn ohjelmointikieleen. Alustassa käytetty Javaa, Go:ta ja TypeScript:iä (NodeJS) ohjelmointikielinä, mutta mikropalveluiden integrointi onnistuu myös mm. Python-, Rust- ja Scala-ohjelmointikielillä.

Kuvassa 2.10 on diagrammi muodossa esitetty, miten yleensä Kaa-mikropalvelut ovat muodostettu. Kuvan jälkeen on esitelty Kaa IoT -alustan keskeisimmät palvelut.



Kuva 2.10 *Kaa IoT -arkkitehtuuri [31]*

Kaa Protocol Communication -palvelu (KPC) kommunikoi asiakas- ja EP-sovellusten (endpoint) kanssa käyttäen jotain seuraavista viestintätavoista: selkokielistä MQTT, MQTTS (TLS salattua MQTT), MQTT WebSocketin yli tai selkokielistä HTTP. KPC on se kontaktipiste, joka suorittaa asiakkaan autentikoinnin ja EP:n todentamisen. Autentikointi tapahtuu käyttäjätunnus-salasana-kombinaation, tai SSL-varmenteen avulla ja todentaminen tapahtuu EP-tunnisteen avulla. [32]

Crendential Management -palvelu (CM) tarjoaa rajapinnan EP-tunnisteiden ja niiden tilojen hallintaa sekä toimii järjestelmään liitettyjen EP:iden tunnistajana. Tunnisteiden mahdolliset eri tilat ovat passiivinen, aktiivinen, jäädytetty tai peruutettu. [33]

Data Collection Extension -palvelu (DCX) toteuttaa Data Collection Protocol -kommunikaatiokyvykkyyden ja vastaanottaa dataa kommunikaatiopalvelulta ja lähettää sen edelleen muille palveluille tallentamista tai prosessointia varten. [34]

Configuration Management Extension -palvelu (CMX) toteuttaa Configuration Management Protocol -toiminnallisuuden, jolla jaetaan konfiguraatiotietoja EP:lle ennakoivasti push-toiminnallisuuden avulla. CMX kuuntelee EP:iden yhdistämis- ja elinkaaritapahtumia havaitakseen, milloin mahdollisia konfiguraatiopäivityksiä tarvitaan. [35]

Endpoint Register -palvelu (EPR) pitää kirjaa kaikista rekisteröidyistä EP:stä sekä niiden metadatatiedoista, joita voidaan käyttää suoraan rajapinnan kautta. Palvelu välittää EP:iden elinkaaritapahtumia, kuten EP:n rekisteröinnin tai poistamisen sekä metatietopäivitykset. [36]

Web Dashboard -palvelu (WD) tarjoaa visuaalisen hallintatyökalun alustan muiden palveluiden määrittämiseksi, käyttäjien hallintaan, EP:iden konfiguroimiseksi ja valvontaan sekä antaa mahdollisuuden visualisoida ja analysoida dataa. [37]

Järjestelmä tukee tunnettuja kevyitä IoT-protokollia laitteiden ja alustan väliseen viestintään, kuten MQTT ja HTTP. Kaa IoT -alusta mahdollistaa myös muiden IoT-protokollien käytön, mutta niiden integrointi jää toteuttajan vastuulle. Lisäksi järjestelmään voidaan liittää myös sellaisia sovelluksia, jotka käyttävät pysyvää tai yhteydetöntä verkkoyhteyttä. Riippuen tarpeesta, MQTT- ja HTTP-protokollia voidaan myös käyttää salattuna, jos kyseessä on laite joka lähettää arkaluonteista dataa. Yhdyskäytäväarkkitehtuurin avulla Kaa IoT -alustaan voidaan liittää myös laitteita, joille ei ole määritelty IP:tä. Tällaisissa tapauksissa on yleensä käytössä erillinen laite, joka kommunikoi laitteiden kanssa ja välittää datan alustan suuntaan.

Kaa IoT -alusta tarjoaa rekisterin digitaalisista kaksosista, jotka kuvaavat alustan hallinnoimia asioita, laitteita ja muita entiteettejä. Myös laitteiden attribuutteja voidaan tallentaa ja siten mahdollistetaan yksilöivän tiedon tallentaminen laitteista (kuten MAC-osoitteen, paikkatiedon tai sovellusversion). Myös monimutkaisem-

pien rakenteisien objektien lisääminen on mahdollista. Attribuuttien avulla voidaan luoda erilaisia suodattimia, jotka jakavat laitteita pienemmiksi ja hallittavammiksi kokonaisuuksiksi.

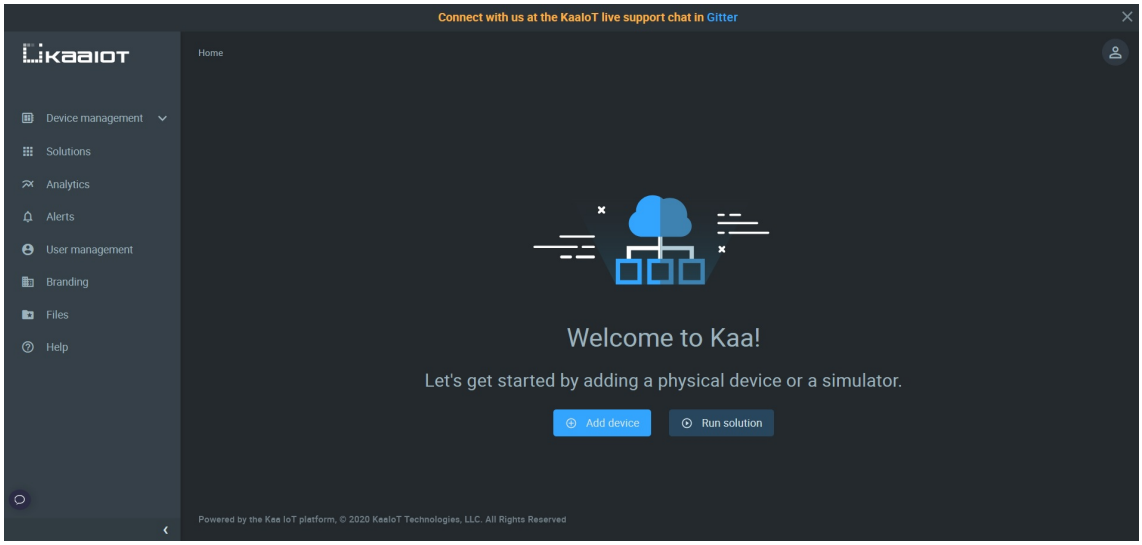
Laitteiden liittäminen järjestelmään vaatii voimassaolevat tunnistetiedot, jotka voivat olla mm. tunnisteena (token), varmenteena (certificate) tai esijaettuna avaimena (pre-shared keys). Järjestelmä pitää kirjaa laitteista ja niiden tapahtumista koko niiden elinkaaren ajan.

Kaa IoT -alustaan on toteutettu helppokäyttöinen protokolla datan keräykseen ja tämän protokollan avulla taataan luotettava datan välitys. Järjestelmään saapunut data voidaan lähettää useammalle prosessointiputkelle. Jos prosessoinnin aikana tapahtuu jotain odottamatonta, laitteelle lähetetään ilmoitus, jonka perusteella laite tietää voiko se tuhota datan vai pitäisikö se uudelleen lähettää. Laitteet voivat myös bufferoida dataa itselleen ja lähettää dataa isompina paketteina alustalle. Data voi olla rakenteellista tai epämuodollista, primitiivisiä tyyppejä (kuten numeroita tai tekstiä), tai yhdistelmiä, kuten avain-arvo-pareja, taulukoita tai sisäkkäisiä objekteja.

Alusta sisältää joukon adaptereita, joiden avulla on mahdollista lähettää dataa useampaan tietokantaan tai erilaisiin data-analyysijärjestelmiin. Raaka, epämuodollinen data voidaan myös muuntaa jäsennellyksi aikasarjaksi, joka sopii paremmin analytiikkaa ja visualisointia varten. Järjestelmään on integroitu Open Distro for Elasticsearch -teknologia kokonaisuus, minkä avulla voidaan tehdä edistynyttä analyysiä, visualisointia sekä hälytyksiä prosessoidusta datasta.

Kaa IoT -alustassa on toiminnallisuus usean liiketoimintayksikön (multitenancy) käyttöön. Tämän avulla yhdellä palvelin-instanssilla voidaan palvella useita itsenäisiä liiketoimintayksiköitä ja niiden hallitsema data on täysin eristetty toisten käytöstä. Toiminnallisuus on toteutettu Keycloak-järjestelmän avulla, joka jakaa liiketoimintayksiköt omiin alueisiinsa (realm). Tämä tarkoittaa siis sitä, että kukin liiketoimintayksikkö hallinnoi vain omia käyttäjiään, laitteita, käyttöoikeuksia, sovelluksia, kojelautoja.

Alustassa on käytössä selainpohjainen hallintatyökalu, jonka avulla voidaan hallita mm. käyttäjiä, laitteita ja käyttöoikeuksia sekä muodostaa visuaalisia kuvaajia, taulukoita ja mittarinäkymiä laitteiden keräämästä datasta. Kuvassa 2.11 on hallintatyökalun etusivunäkymä.

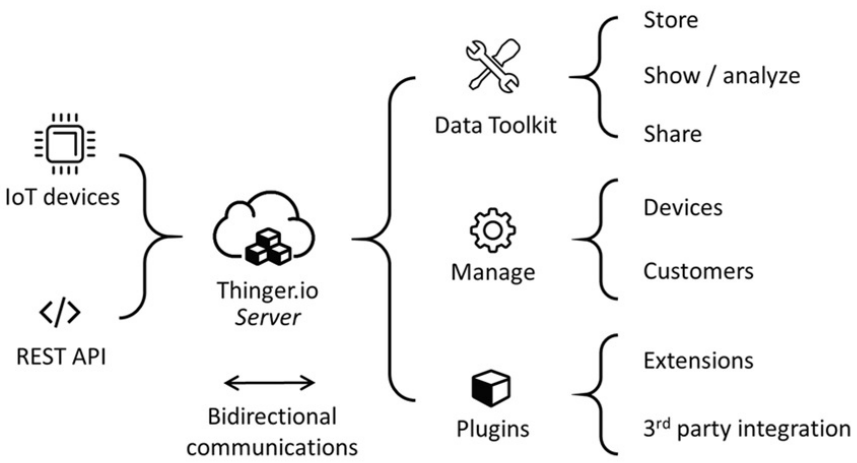


Kuva 2.11 *Kaa IoT -hallintatyökalu*

2.4.3 Thinger.io

Thinger.io on pilvipohjainen IoT-alusta, joka tarjoaa työkalut liitettävien laitteiden skaalaamiseen sekä hallintaan. Alustan tavoitteena on selkeyttää IoT:n käyttöä suurissa järjestelmissä ja tässä apuna käytetään avoimeen lähdekoodiin pohjautuvia moduuleja, kirjastoja ja sovelluksia. Alustasta on saatavilla ilmainen Maker-versio sekä useampi erillinen maksullinen versio, jotka eroavat toisistaan eri ominaisuuksilla ja toiminnallisuuksilla. [38]

Thinger.io-alusta koostuu kahdesta eri kokonaisuudesta, IoT-palvelimesta (backend) ja web-sovelluksesta (frontend). Näiden avulla voidaan helposti käyttää järjestelmän eri ominaisuuksia. Lisäksi järjestelmään voidaan laajentaa plugins-ominaisuuden avulla eri käyttötarkoituksia tai liiketoimintoja varten. Kuvassa 2.12 on esitelty Thinger.io-alustan keskeisimmät ominaisuudet.

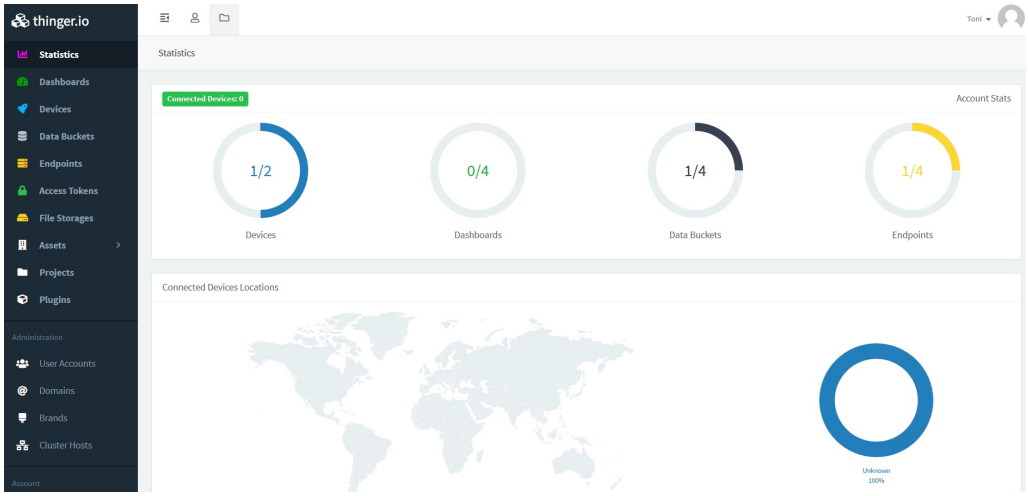


Kuva 2.12 *Thinger.io-alustan ominaisuudet [39]*

Alusta on yhteensopiva minkä tahansa laitteen kanssa riippumatta laitteen prosessorista, tuetusta verkkoliitännästä tai valmistajasta. Lisäksi Thinger.io:ssa on tuki kaksisuuntaiselle viestinnälle Linux-, Arduino-, Raspberry Pi-, MQTT-laitteiden sekä Sigfox-, LoRaWAN- ja Internet-tietolähteiden kanssa.

Datan tallentamiseen käytetään avuksi ”data ämpäreitä” (data bucket), joiden avulla mallinnetaan laitteilta tulevaa dataa sekä tarvittaessa yhdistetään sitä reaaliaikaisesti. Dataa voidaan myös esittää reaaliaikaisesti useamman rinnakkaisen kuvaajan, taulukon, mittarin tai kustomoidun näkymän avulla.

Hallintatyökalun avulla voidaan ylläpitää alustan laitteita ja käyttäjiä sekä tarkastella dataa erilaisten kojelautanäkymien avulla. Myös hallintatyökalun ulkoasun muokkaaminen on mahdollista. Kuvassa 2.13 on hallintatyökalun etusivunäkymä.



Kuva 2.13 *Thinger.io-hallintatyökalu*

3 CityIoT-hanke, tietoturvavaatimukset ja -ohjeistukset

Tässä kappaleessa on esitelty CityIoT-hanke ja hankkeen aikana toteutettu referenssitoteutus sekä määritellyt tietoturvavaatimukset. Lisäksi käydään läpi vertailussa käytettävät ohjeistukset, jotka on koostettu alalla toimivien tahojen tuottamista ohjeistuksista.

3.1 CityIoT-hanke ja referenssitoteutus

Tulevaisuuden toimijariippumaton dataintegraatioalusta -hanke (CityIoT) oli 6Aika-strategian Smart City -ratkaisujen kehittämishanke. Hankkeessa olivat mukana Oulun yliopisto, Oulun kaupunki, Tampereen kaupunki, TTY-säätiö sekä Oulun ammattikorkeakoulu. Tavoitteena hankkeella oli [40]:

- avoimen ja toimijoista riippumaton dataintegraatioalusta
- referenssiarkkitehtuurin määrittely
- järjestelmien rajapintaehtojen määrittely
- IoT-pilotointiympäristön rakentaminen
- osallistuttaa Pk-yrityksiä mukaan hyödyntämään pilottiympäristöjä uusien tuotteiden kehittämiseen
- uusien liiketoimintamahdollisuuksia luominen yrityksille IoT-tietoja hyödyntämällä
- auttaa uusien langattomien teknologioiden kehittymistä
- nopeuttaa digitalisaatiota Suomessa

Hankkeen aikana luotiin vaatimusmäärittely, jonka pohjalta voidaan kehittää IoT-alustaratkaisu hankkeen tavoitteiden mukaisesti. Vaatimusmäärittelyä varten pidettiin työpajoja, kuunneltiin loppukäyttäjien näkemyksiä ja haastateltiin asiantuntijoita, jotka edustavat erilaisia näkökulmia kuten alustaliiketoimintaa, kehittämistä ja palvelutuotantoa. Vaatimusten keräämisen aikana tutkittiin olemassa olevia IoT-ratkaisuja ja -alustoja sekä tunnistettiin niiden tarpeita ja vaatimuksia alustojen palveluiden saattamiseksi kaupunkien ja yritysten käyttöön. [41]

CityIoT-hankkeen yhtenä tavoitteena oli määritellä toimijariippumaton IoT-alusta älykkäiden kaupunkien käyttöön. Alustan tulisi tukea useita tietolähteitä, mahdollistaa datan käyttö usealle sovellusratkaisulle, skaalautua useampaa käyttötapausta varten, tarjota käyttäjien pääsynhallinta, olla helposti käyttöön otettava sekä tarvittaessa olla laajennettavissa eri tarpeita varten. Hankkeen alussa tutkittiin vaihtoehtoja alustan toteutusta varten ja FIWARE valittiin teknilliseksi pohjaksi toteutettavalle referenssialustalle. Valintaan vaikuttivat alustan tarjoamat avoimeen lähdekoodiin pohjautuvat komponentit, FIWARE yhteisön samankaltaiset tavoitteet toimijariippumattomaan alustaan sekä SmartCity-yhteisön mielenkiinto alustaa kohtaan. [42]

FIWARE on avoimeen lähdekoodiin pohjautuva IoT-alusta. Järjestelmällä pyritään määrittelemään datan hallintaa koskevat yleiset standardit, joiden tarkoituksena olisi helpottaa erilaisten älykkäiden ratkaisujen kehitystyötä, kuten älykkäät kaupungit sekä älykäs teollisuus.

Kaikkien FIWARE-ratkaisujen tärkein komponentti on FIWARE Context Broker Generic Enabler, joka toimii minkä tahansa älykkään ratkaisun ytimenä. Komponentin avulla voidaan päivittää ja jakaa dataa, eli hallita järjestelmään tallennettua dataa.

FIWARE NGSI on ydin komponentin tarjoama sovellusrajapinta, jonka avulla voidaan integroida muut komponentit alustaan. Sovellusrajapinnan määritykset kehittyvät jatkuvasti ja vastaavat työn kirjoitushetkellä NGSIv2-määrityksiä, mutta tulevat kehittymään ETSI NGSI-LD-standardin mukaiseksi. FIWARE-yhteisö on mukana kehittämässä NGSIv2:een perustuvien ETSI NGSI-LD-määrittelyitä ja he ovat sitoutuneet toimittamaan määrittelyä vastaavan avoimen lähdekoodin toteutuksen.

FIWARE Context Broker:n ympärille on saatavana runsaasti järjestelmää laajentavia Generic Enablers –kokonaisuuksia. Näiden avulla voidaan esimerkiksi hallita ja tallentaa dataa tietolähteistä, välittää dataa jatkokäsittelyyn, tarjota rajapinta IoT-laitteiden hallintaa varten, prosessoida, analysoida ja visualisoida tietolähteistä saatua dataa, tai julkaista ja kaupallistaa järjestelmään varastoitua dataa. Lisäksi alusta tarjoaa joukon kehitystyökaluja joiden avulla FIWAREn ja muiden tarjoamien komponenttien käyttöönottoon ja konfigurointiin. [43] Kuvassa 3.1 on esitelty edellä kerrotut Generic Enablers -kokonaisuudet ja kuvan jälkeen on kerrottu lyhyesti millaisia komponentteja kukin kokonaisuus voi pitää sisällään.

Kuva 3.1 *FIWARE-rakennekuva [44]*

Ydinkomponentti: Orion Context Broker toimii järjestelmän ydinkomponenttina ja toteuttaa NGSIv2-sovellusrajapinnan sekä hallinnoi järjestelmään liitettyjen laitteiden tiloja. QuantumLeap-komponentin on ytimen rinnalla toimiva komponentti ja sen avulla tallennetaan laitteiden historiatietoja aikasarjamuotoisena tietokantaan. [43]

Sisällön prosessointi, analysointi, visualisointi: Wirecloud-komponentti tarjoaa web-käyttöliittymän alustaan. Tämän käyttöliittymän avulla voidaan luoda kojelautanäkymiä tietolähteiden tuottamasta datasta loppukäyttäjille. [43]

Rajapinnat IoT-laitteille, automatiikalle ja 3rd party-järjestelmille: Ultralight-komponenttia käytetään siltana NGSIv2-sovellusrajapinnan ja tietomallin välillä sekä kommunikoimaan niiden tietolähteiden kanssa, jotka käyttävät Ultralight-protokollaan pohjautuvaa tekstipohjaista tiedonvälitystapaa (HTTP-, AMQP- tai MQTT-tiedonsiirtoprotokollia) hyödyntäen. [43]

Sovellusrajapinta, julkaisu, kaupallistaminen: Keyrock-komponentin avulla hallinnoidaan käyttäjiä ja niiden rooleja sekä mahdollistetaan OAuth2-autentikointiprotokollan käyttäminen järjestelmässä. AuthZForce-komponenttia käytetään luomaan joustavia pääsynhallintasääntöjä alustan hallinnoimaan dataan. Järjestelmän pääsynhallinnasta vastaa Wilma-palvelu ja pääsyä valvotaan Keyrock-palvelun hallinnoimien käyttäjien ja roolien sekä AuthZForcen määrittelemien sääntöjen avulla. Datan julkaiseminen tapahtuu CKAN-julkaisualustan integraatiolla. [43]

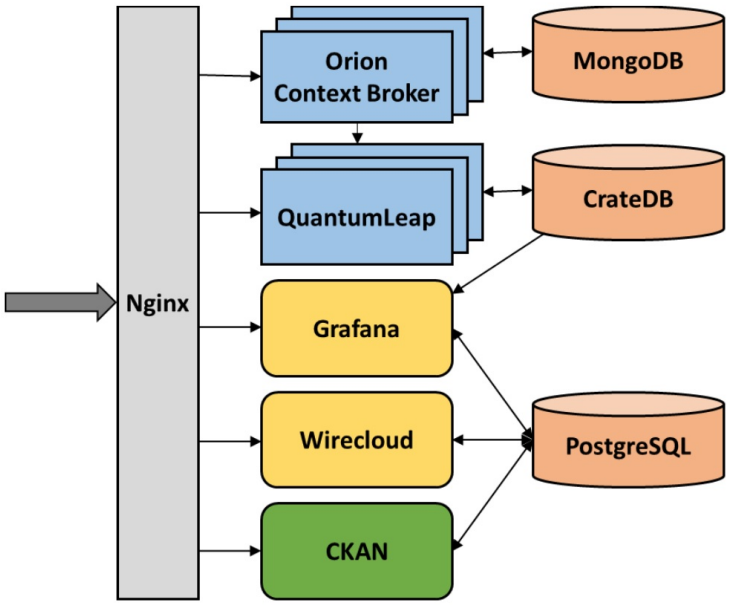
Kehitystyökalut: Alustan mukana tulee useita kehitystyökaluja, joiden avulla FIWAREn, tai kolmansien osapuolten komponenttien integrointi ja käyttöönotto helpottuvat. [43]

CityIoT-referenssitoteutus (jatkossa CityIoT FIWARE -alusta) pohjautuu FIWARE-alustaan ja sen ydinkomponentin ympärille valittuihin laajennoksiin sekä muihin avoimen lähdekoodin sovelluksiin. Nginx proxy -palvelintoteutus tarjoaa TLS-yhteyden, joka käsittelee kaiken alustalle saapuvan liikenteen ja välittää pyynnöt oikeille alustapalveluille. Proxy tarjoaa myös yksinkertaisen käyttäjien pääsynhallinnan alustan ydin komponentille ja QuantumLeap-komponentille sekä välimuistiratkaisun toistuvien pyyntöjen tehokkaampaan käsittelyyn.

Alustan ydinkomponenttina toimii FIWARE Orion Context Broker, joka tallentaa tietolähteiden lähettämän datan MongoDB-tietokantaan. Historiatietoja hallitaan QuantumLeap-komponentin avulla yhdessä ydinkomponentin kanssa. Historiatietoja tallennetaan CrateDB-tietokantaan.

Alustaan on myös liitettynä muutamia avoimen lähdekoodin sovelluksia, jotka tarjoavat työkaluja alustalle tallennetun datan käsittelyä varten. Wirecloud ja Grafana sovellusten avulla voidaan visualisoida dataa, suorittaa datan analysointia ja tehdä erilaisia hälytyksiä analytiikan pohjalta. CKAN-dataportaalin avulla puolestaan voidaan julkaista ja jakaa alustan dataa. Jokainen näistä komponenteista käyttää PostgreSQL-tietokantaa tiedon tallentamista varten sekä tarjoavat oman pääsynhallinnan käyttäjille.

Kuvassa 3.2 on esitetty CityIoT FIWARE -alustan arkkitehtuurikuvaus.



Kuva 3.2 *CityIoT FIWARE -alustan arkkitehtuurikuvaus [45]*

Lisäksi alustaan voidaan sisällyttää IoT Agent for Ultralight -komponentti. Komponentti on valinnainen ja sen käyttöönotto tapahtuu samalla tavalla, kuin Orion Context Broker:n ja QuantumLeap:n.

3.2 Tietoturvavaatimukset

Vaatimusmäärittelyn pohjalta luotiin erillinen dokumentaatio, joka pitää sisällään alkuperäisistä vaatimuksista jalostettuja IoT-alustan tietoturvaan liittyviä vaatimuksia [7]. Tietoturvavaatimukset ovat käännettynä ja löytyvät tämän työn liitteistä [Liite A]. Nämä vaatimukset ovat kirjoitettu käyttötapauksina ja ne ovat jaoteltuina seuraaviin kategorioihin:

1. **Sopimusasiat:** Tuotantokäytössä sopimuksilla selvennetään oikeuksia ja vastuita kullekin käyttäjälle. Sopimukset eivät välttämättä näy alustalle tallennettuun dataan, mutta ne määrittelevät mitä dataa voidaan tallentaa alustaan ja millä säännöillä niihin annetaan pääsyoikeuksia.

2. **Datasettien hallinta:** Datasettien hallinta pitää sisällään käyttäjien lisäämät ja muokkaamat datan sekä metatiedot.

3. **Käyttäjähallinta:** Alustalle määriteltävien käyttäjien ja käyttäjäryhmien tai organisaatioiden hallinta. Käyttäjiä voidaan määritellä joko suoraan alustalle, tai he voivat käyttää ulkoista järjestelmää, joka käyttää pääsynhallintamekanismia käyttäjien käyttöoikeuksien tunnistamiseksi. Tälläisenä protokollana voi toimia esimerkiksi OAuth2.

4. **Pääsynhallinta:** Pääsynhallinta pitää sisällään oikeuksien luonnin ja muokkauksen. Pääsynhallinnan avulla voidaan kertoa käyttäjälle, onko hänellä oikeus johonkin tiettyyn dataan. Käyttöoikeuspyynnön koko sisältö pitää pystyä varmistamaan.

5. **Pääsynhallinta kaupalliseen tietoon:** Kaupalliseksi merkitylle datalle on pystyttävä asettamaan tiettyjä sääntöjä, jotka ilmoittavat ko. tiedon olevan kaupallista. Näiden sääntöjen avulla määritellään käyttöoikeudet vain niille käyttäjille, jotka ovat ostaneet käyttöoikeudet alustan dataan.

6. **Datan käyttö- ja muokkausoikeudet:** Kun datasetti tallennetaan alustalle ja tarvittavat käyttöoikeudet ovat asetettuna, järjestelmän käyttäjille voidaan myöntää pääsy dataan. Käyttäjät voivat myös etsiä mihin tietoihin heillä on pääsyoikeudet ja tehdä hakuja tähän dataan.

7. **Ulkoisten tietolähteiden lisääminen alustaan:** Ulkoisella tietolähteellä tarkoitetaan alustan ulkopuolelta tulevaa dataa toisesta IoT-alustasta, tietokannasta tai muusta ulkoisesta tietolähteestä. Data joko haetaan toisesta ulkoisesta lähteestä ja kopioidaan IoT-alustaan, tai tarkentavan lisätiedon avulla mahdollistetaan ulkoisen tietolähteen tietojen käyttö alustalla. Tässä tapauksessa alusta toimisi eräänlaisena välittäjänä käyttäjän ja ulkoisen datan välillä.

8. **Alustaan liitettyjen IoT-laitteiden hallinta:** IoT-laitteita järjestelmään lisäävien käyttäjien hallinta. IoT-laitteet syöttävät dataa suoraan alustalle, joten sen on oltava alustan ymmärrettävässä muodossa, tai muutettuna välittäjän kautta sellaiseen muotoon.

9. **Pääsynhallinta alustan toiminnallisuuksiin:** Kun datan omistaja kohtaa ongelmia sisällönhallinnassa alustalla, esimerkiksi saatavuuden tai suorituskyvyn kanssa, pitää alustan tiettyihin toiminnallisuuksiin päästä käsiksi. Sisällönhallintaongelmien tapauksessa esimerkiksi alustan lokitiedoista voidaan saada selville mistä ongelmat voivat johtuvat

3.3 Tietoturvaohjeistukset

Tässä alikappaleessa käydään läpi työn vertailussa käytetyt tietoturvaohjeistukset, jotka soveltuvat työn rajauksen kanssa.

3.3.1 OWASP Top 10 Proactive Controls

OWASP Top 10 Proactive Controls -projekti ohjeistaa mitä tärkeitä asioita tulisi huomioida tietoturvan osalta kehitettäessä web-sovelluksia. Sen tarkoitus ei kuitenkaan ole kertoa mitä tekniikoita ja käytäntöjä tulisi noudattaa sovellusten kehittämisessä, vaan ennemminkin sitä tulisi pitää pohjana kehitystyössä ja kehittäjien tietoisuuden parantamisessa. Ohjeistuksessa otetaan kantaa mm. tietoturvavaatimuksiin, tietoturvakehyksien ja -kirjastojen käyttöön sekä datan turvaamiseen. [46]

Tässä työssä ohjeistuksesta huomioidaan vain seuraavat asiat, koska ne ovat sovellettavissa työn rajauksen kanssa:

> **C7: Enforce Access Controls:** Pääsynhallinta tai todentaminen on prosessi, jolla hyväksytään tai estetään käyttäjän, sovelluksen, tai prosessin esittämä pyyntö. Pääsynhallintaan kuuluu myös näiden oikeuksien myöntäminen ja poistaminen. Riippuen pääsynhallinnan monimutkaisuudesta, voi se liittyä moniin eri alueisiin järjestelmässä. Metadatan tai välimuistin käyttö järjestelmän skaalautuvuus tarkoituksessa eivät ole pakollisia, mutta ne pitää huomioida pääsynhallintajärjestelmässä. [47]

C8: Protect Data Everywhere: Hyökkääjät voivat varastaa tietoa web-palveluista lukuisilla eri tavoilla. Usein kohteena on arkaluonteinen tieto, kuten käyttäjätunnukset ja salasana, pankkikorttitunnukset, henkilötiedot tai yrityssalaisuudet. Siksi onkin tärkeää kategorisoida järjestelmässä oleva tieto ja määritellä näille eri kategorioille suojaussäännöt. Yleisesti voidaan käyttää seuraavanlaista luokittelua: avoin ja salainen tieto. Avointa tietoa voi säilyttää vapaammin järjestelmässä, mutta salattua tietoa tulisi pitää kryptattuna tietokannassa ja yhteyksien tulisi olla salattuja, kun tätä tietoa siirretään palvelujen välillä. Lähtökohtaisesti salaista tietoa pitäisi välttää käytettävän web-järjestelmissä, mutta useimmiten tämänkaltaisen tiedon käyttö on välttämätöntä (esimerkiksi pankkien verkkopalvelut, yhteiskunnalliset palvelut, verkkokaupat).

Kun tietoa lähetetään minkä tahansa verkon läpi, tulisi viesti lähettää aina salattuna. TLS on käytetyin ja laajasti tuettu salausprotokolla viestien suojaamiseen. Useat erilaiset sovellukset (kuten web, verkkopalvelut, mobiilisovellukset) käyttävät sitä kommunikoidakseen turvallisesti verkon kautta. TLS tulee konfiguroida oikealla tavalla, jotta viestintä tulee asianmukaisesti turvattua. Tärkein hyöty viestinvälityskerroksen salaamisesta on verkkosovellustietojen suojaaminen luvattomalta paljastumiselta ja muokkaamiselta, kun tietoa siirretään asiakas- ja palvelinsovellusten välillä. [48]

C9: Implement Security Logging and Monitoring: Käyttölokien avulla saadaan tietoa sovelluksesta sen ajon aikana. Lokeista voidaan mm. tunnistaa automaatiota käyttäen mahdolliset järjestelmää kohtaan tehtävät tietoturvarikkomukset sekä voidaan jäljittää käyttäjien tekemiä muutoksia dataan tai järjestelmän konfiguraatioihin. Joissakin maissa myös lainsäädäntö vaatii käyttölokien käytön. [49]

3.3.2 NIST Digital Identity Guidelines

Tämän dokumentaation tarkoituksena on antaa eri tahoille (virastot, viranomaiset) tekniset ohjeet digitaalisen todennuksen toteuttamiseksi tietoteknisiin järjestelmiin.

Digitaalinen identiteetti on yksilöivä kuvaus kohteesta, joka suorittaa online-tapahtumia. Digitaalinen identiteetti on aina yksilöivä tietyssä digitaalisessa palvelussa, mutta sitä ei ole tarpeen yhdistää tosielämän kohteeseen. Toisin sanoen digitaalisen palvelun käyttö ei välttämättä vaadi sitä, että taustalla olevan henkilön tosielämän identiteetti tiedettäisiin. Digitaalisen identiteetin toteutus järjestelmille on tekninen haaste, koska usein tämä tarkoittaa yksilön tunnistamista avoimen verkon ylin. Tämä tarjoaa helpon mahdollisuuden hakkerille digitaalisen identiteetin väärinkäytöksiin. [50]

3.3.3 Tietosuojavaltuutetun toimisto

Tietosuojavaltuutetun toimisto on verkkopalvelu josta voi saada tietoa monista tietosuojaan liittyvistä asioista sekä säädöksistä (kuten GDPR). [51]

Kaikki tieto millä voidaan tunnistaa, tai identifioida käyttäjä luetaan henkilötiedoiksi. Yleisimpiä henkilötiedoiksi tunnistettavia asioita ovat sähköpostiosoite (muodossa etunimi.sukunimi@yritys.fi), puhelinnumero, auton rekisterinumero, IP-osoite ja potilastiedot. EU-maissa tulee noudattaa henkilötietojen käsittelyssä GDPR-asetuksen vaatimuksia riippumatta niiden käsittelyssä käytetystä tekniikasta. Tietojen tallennusmenetelmä voi olla IT-järjestelmä, videovalvontajärjestelmä tai paperiarkisto. Tietosuojasäännösten sovellettavuuden kannalta ei ole myöskään merkitystä, onko henkilötietojen käsittely keskitetty yhteen paikkaan vai hajautettu useisiin paikkoihin, järjestelmiin tai prosessoreihin. Olennaista on, että tietyn henkilön tiedot voidaan saada keskitetystä tai hajautetusta tiedosta, joka on asetettu tietyillä kriteereillä, kuten nimellä tai henkilötunnuksella. Toisin sanoen, jos niiden käyttötarkoitus on sama, tiedot kuuluvat samaan loogiseen datatiedostoon. Loogiseen datatiedostoon ei vaikuta se, että kerättäisiinkö ne eri lähteistä, tallennettaisiinko ne eri paikkoihin tai käsittelevätkö niitä eri osapuolet. [52]

3.3.4 OWASP API Security Project

OWASP API Security -projekti keskittyy strategioihin ja ratkaisuihin sovellusten API-rajapintojen haavoittuvuuksien ja tietoturvariskien ymmärtämiseksi.

Sovellusohjelmointirajapinnat (API) ovat kriittinen osa moderneja mobiili-, SaaS- ja verkkosovelluksia. Niitä käyttävät mm. pankit, vähittäiskauppa, IoT, itsenäiset ajoneuvot sekä älykkäät kaupungit. Sovellusliittymät paljastavat sovelluslogiikan ja arkaluonteisia tietoja (kuten henkilötietoja) ja tästä syystä ne ovat altistuneet enemmän hyökkäyksille. [53]

Seuraavassa listassa on esitelty vertailussa käytettyjen OWASP API Security -projektin ohjeistukset:

Broken Object Level Authorization: Jokaiselle sovellusrajapinnan päätepisteelle, joka suorittaa toiminnon valitulle objektille, tulisi suorittaa objektitason todennus. Tällä varmistetaan, että käyttäjällä oikeus suorittaa pyydetty toiminto valitulle objektille.

Excessive Data Exposure: Sovellusrajapinnan tulisi välttää paljastamatta liikaa tietoja käyttäjälle. Jos rajapinta on suunniteltu siten, että tietoja suodatetaan vasta asiakassovelluksen päässä ennen kuin dataa esitetään käyttäjälle,

voi hyökkääjä saada verkkoliikennettä skannaamalla tietoonsa arkaluonteisia tietoja järjestelmästä.

Broken Function Level Authorization: Pääsynvalvonnan monimutkaiset käytännöt, erilaiset hierarkiat, ryhmät ja roolit sekä epäselvät rajat käyttäjille ja ylläpitäjille tarkoitetuista toiminnoista voi johtaa helposti virheellisiin valtuutuksiin. Hyökkääjät voivat hyödyntää näitä virheitä ja niiden avulla he voivat päästä käyttämään käyttäjien tai ylläpitäjien toimintoja.

4 Vertailu

Työssä toteutettu vertailu on suoritettu siten, että IoT-alustojen tietoturvaominaisuuksia tarkastellaan kappaleessa 3.2 esitettyjen CityIoT-hankkeen tietoturvavaatimuksia vasten sekä myös yleisesti käytettyihin tietoturvaohjeistuksiin. Jälkimmäisessä käytetään apuna erilaisia lähteitä, jotka on pyritty valitsemaan yleisesti alalla tunnettujen tahojen julkaisemista ohjeistuksista. Ohjeistukset on esitelty kappaleessa 3.3.

Jotta alustoja voitaisiin vertailla järkevästi keskenään, vertailussa käytetään apuna ominaisuuksien pisteyttämistä. Taulukossa 4.1 on esitelty kuinka vertailtava alusta voi saada pisteitä: jos alusta ei toteuta vaatimusta ollenkaan, annetaan 0 pistettä. Jos vaatimus toteutuu täysin tai paremmin, annetaan 1 pistettä. Pisteytystä käytetään, kun tarkastellaan hankkeen vaatimuksia sekä tietolähteitä alustan ominaisuuksia vasten. Pisteytykset löytyvät kunkin vertailtavan vaatimuksen lopuksi taulukosta tämän kappaleen sisältä. Kaikkien alustojen tulosten yhteenveto esitellään kappaleessa 5.

Taulukko 4.1 *Vertailussa käytetty pisteyttäminen*

Vertailun tulos	Pisteytys
Vaatimus ei toteudu ollenkaan	0
Vaatimus toteutuu täysin	1

4.1 CityIoT FIWARE

Datasettien hallinta

Datasetin luonti aloitetaan CityIoT FIWARE -alustalla määrittelemällä entiteetti, joka kuvastaa loogista objektia (laitetta, kulkuneuvoa, lämpömittaria tai rakennusta). Entiteeteille voidaan myös määritellä attribuutteja sekä metadatatietoja joiden avulla rikastetaan tietolähteiden lähettämää dataa. Käyttäjien ja palveluiden käyttöoikeudet määritellään erillisillä konfiguraatiotiedostoilla. Käyttäjälle ilmoitetaan virheilmoituksella, jos hänellä ei ole oikeutta lisätä dataa valitsemaansa datasettiin. Alustaan voidaan lisätä dataa vain ulkoisen rajapinnan kautta, erillistä hallintatyökalua ei ole. Myöskään suurien data määrien tuonti kerralla, eli ns. eräajot eivät ole mahdollisia. Datan omistajat voivat muokata ja poistaa datasettejä suoraan alustalta ulkoisen rajapinnan kautta. Datasetin poisto ei poista siihen liitettyjä käyttöoikeuksia, vaan ne tulee poistaa erikseen konfiguraatiotiedostoista.

Vaikka alustan dataa voidaan rikastaa todella joustavasti attribuuttien ja metadatan avulla, niiden käyttöä esimerkiksi datan näkyvyyden asettamisessa, käyttöoikeuksien tai datan omistajan määrittelemisessä, lisensoimisessa tai datan kaupallistamisessa ei ole dokumentoitu.

Taulukko 4.2 *CityIoT FIWARE pisteytys - Datasettien hallinta*

Vaatimus	Tulos	Pisteet
Datasettien luonti*	Toteutuu	1
Datan lisääminen datasettiin*	Ei toteudu	0
Datasetin muokkaaminen tai poisto*	Toteutuu	1
Metatiedon lisääminen alustassa olevaan dataan*	Ei toteudu	0
Datan kategorisointi avoimeksi ja salaiseksi**	Ei toteudu	0
Salauksen käyttö datan lähettämisessä**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Protect Data Everywhere -ohjeistus

Käyttäjähallinta

Käyttäjien lisääminen CityIoT FIWARE -alustaan tapahtuu ylläpitäjän toimesta. Jokaiselle käyttäjälle määritellään mitä alustan palveluja he voivat käyttää sekä mitä tunnistetta heidän tulee käyttää jokaisessa palvelukutsussaan. Grafana-, Wirecloud- ja CKAN-palveluissa on omat käyttäjä- ja pääsynhallintajärjestelmät, joihin käyttäjät voivat kirjautua käyttämällä käyttäjä-salasana-yhdistelmää.

Taulukko 4.3 *CityIoT FIWARE pisteytys - Käyttäjähallinta*

Vaatimus	Tulos	Pisteet
Alustan käyttäjien hallinta*	Toteutuu	1
Digitaalinen identiteetti**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** NIST Digital Identity Guidelines -ohjeistus

Pääsynhallinta

Jokaiselle käyttäjälle määritellään pääsyoikeudet sekä mitä CityIoT FIWARE -palveluita he voivat käyttää. Kullekin palvelulle määritellään käyttöoikeudet erikseen. Käyttöoikeusvaihtoehtoja ovat vain lukuoikeudet (sallitaan vain GET-toiminnot) tai ei rajoituksia ollenkaan. Oikeuksien tarkistus suoritetaan aina, kun käyttäjä tekee pyynnön alustan palveluille. Käyttöoikeuksia ei voida asettaa yksittäiselle datalle alustalla ja niitä voi hallinnoida vain alustan ylläpitäjä.

Grafana-, Wirecloud- ja CKAN-palvelut tarjoavat kaikki omat käyttäjien- ja pääsynhallintajärjestelmänsä. Kunkin palvelun pääsivu on vapaasti kaikkien käyttäjien käytettävissä ilman käyttäjätiliä ja järjestelmän ylläpitäjätilin avulla hallitaan tarvittavia käyttöoikeuksia kunkin palvelun web-käyttöliittymään. Ylläpitäjätilit luodaan alustan alustusvaiheessa.

Taulukko 4.4 *CityIoT FIWARE pisteytys - Pääsynhallinta*

Vaatimus	Tulos	Pisteet
Käyttöoikeuksien ja -tasojen määrittely datasettiin tai muuhun dataan*	Ei toteudu	0
Datasetin käyttöoikeuksien muokkaaminen tai kaiken datan muokkaus*	Ei toteudu	0
Pääsynhallintaprosessi**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Acess Controls -ohjeistus

Pääsynhallinta kaupalliseen tietoon

FIWARE Business API Ecosystem -ja CKAN-laajennuksien avulla voitaisiin kaupallistaa CityIoT FIWARE -alustan dataa. Lisäksi alustan pääsyn- ja käyttäjienhallinta tulisi toteuttaa käyttäen FIWAREn omia komponentteja [54]. CityIoT FIWARE -alustaa ei kuitenkaan ole toteutettu näin, joten datan kaupallistamista ei alustalla voida tehdä. Integroidun CKAN-palvelun avulla voidaan kuitenkin julkaista dataa avoimen datan periaatteiden mukaisesti.

Taulukko 4.5 *CityIoT FIWARE pisteytys - Pääsynhallinta kaupalliseen tietoon*

Vaatimus	Tulos	Pisteet
Kaupallisten oikeuksien määrittely tietokokoelman käyttöön*	Ei toteudu	0
Tietokokoelman käyttöoikeuksien myynti alustalta kolmansille osapuolille*	Ei toteudu	0
Tietokokoelman omistuksen siirtäminen tai muokkaus*	Toteutuu	1

* CityIoT-hankkeen vaatimus

Datan käyttö- ja muokkausoikeudet

Käyttäjät voivat hakea kaikkia alustalle tallennettuja tietoja tekemällä hakuja rajapinnan kautta. Jotta pyyntö onnistuu, tulee käyttäjillä olla määriteltynä pääsyoikeudet tarvittaviin rapapintoihin. Pyyntö listaa kaikki tiedot haetusta asiasta, mutta pääsyrajoituksia ei listata vastaukseen. Käyttäjien tulee siis tietää etukäteen mitä tietoja he haluavat hakea alustasta ja pyytää pääsyoikeuksia tämän perusteella. Jokaisen käyttäjän tulee käyttää yksilöivää tunnistetta jokaisessa alustalle tehdyssä pyynnössään. Alusta tarkistaa tunnisteen ja palauttaa pyydetyt tiedot, jos käyttäjän pääsyoikeudet ovat oikeat. Muussa tapauksessa palautetaan virheviesti (unauthorized data). Kaikki pyynnöt alustalle tulee tehdä käyttäen HTTPS-protokollaa.

FIWARE CityIoT -alustassa ilmoitusten lähettäminen voidaan tehdä Grafana-palvelun avulla, vaikka alustan dokumentaatiosta ei löytynyt tähän suoraa vastausta. Grafanassa kuitenkin on ko. toiminnallisuus, joten voidaan olettaa, että ilmoitukset voidaan tehdä käyttämällä tätä palvelua. Ilmoituksia voidaan lähettää mm. sähköpostiin, Slack-sovellukseen tai webhook-metodia käyttäen.

Alustan käyttäjiä ei koskaan todenneta vaan oikeuksien tarkastelu perustuu käytettyyn tunnisteeseen. Tästä syystä käyttäjistä ei ole tallennettu mitään tietoa alustaan. Ainoastaan Grafana-, Wirecloud- ja CKAN-palvelut toteuttavat oman käyttäjähallinan ja niiden käyttö vaatii perustietoja käyttäjästä. Nämä tiedot löytyvät ko. palvelun hallintatyökalun avulla, tai tietoja voidaan myös pyytää palvelun ylläpitäjältä.

Taulukko 4.6 *CityIoT FIWARE pisteytys - Datan käyttö- ja muokkausoikeudet*

Vaatimus	Tulos	Pisteet
Tiedon ja tietolähteiden saatavuuden tarkastelu*	Ei toteudu	0
Käyttäjä voi käsitellä ja muokata tietoa alustassa*	Toteutuu	1
Ilmoitusten vastaanottamisen tilaustiedon muuttuessa alustassa on mahdollista*	Toteutuu	1
Käyttäjät voivat nähdä mitä tietoa heistä on tallennettuna alustaan*	Toteutuu	1
Pääsynhallintaprosessi**	Toteutuu	1
Salauksen käyttö datan lähettämisessä***	Toteutuu	1
Henkilötietojen hallinta****	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Access Controls -ohjeistus

*** OWASP Protect Data Everywhere -ohjeistus

**** Tietosuojavaltuutetun toimisto Henkilötiedot

Ulkoisten tietolähteiden lisääminen alustaan

CityIoT FIWARE -alustalle voidaan liittää ulkoisia tietolähteitä erillisen rajapinnan avulla. Datan lisääminen tapahtuu samalla tavalla, kuin uuden datan lisääminen alustaan. Alusta tukee HTTP-protokollan käyttöä ulkoisen tietolähteen datan vastaanottamisessa, mutta valinnaisen laajennoksen avulla saadaan myös tuki MQTT- ja AMQP-protokollille.

Koska CityIoT FIWARE -alustan pääsynhallinta perustuu vain käyttäjän tunnisteeseen, ulkoista tietolähdettä varten on mahdollistettu erillinen rajapinta datan tuomiseen alustalle. Rajapinnan tarjoaa alustan ydinkomponentti ja sitä tulee käyttää yksilöivän tunnisteen kanssa.

Taulukko 4.7 *CityIoT FIWARE pisteytys - Ulkoisten tietolähteiden lisääminen alustaan*

Vaatimus	Tulos	Pisteet
Alustan liittäminen toiseen IoT-alustaan*	Toteutuu	1
Broken Object Level Authorization**	Toteutuu	1
Excessive Data Exposure**	Toteutuu	1
Broken Function Level Authorization**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP API Security Project -ohjeistus

Alustaan liitettyjen IoT-laitteiden hallita

CityIoT FIWARE -alustassa käyttäjät voivat lisätä laitteita käyttäen ulkoista rajapintaa. Laite vastaa yhtä entiteettiä järjestelmässä ja käyttäjät voivat lisätä entiteettejä vain, jos heillä on järjestelmän vaatima tunniste käytössään. Lisätyt laitteet lähettävät keräämäänsä dataa alustaan haluttujen asetusten mukaisesti ja näiden asetusten päivittäminen onnistuu alustan avulla lähettämällä päivityksiä laitteille ulkoista rajapintaa käyttäen.

Taulukko 4.8 *CityIoT FIWARE pisteytys - Alustaan liitettyjen IoT-laitteiden hallinta*

Vaatimus	Tulos	Pisteet
IoT-laitteiden lisääminen alustaan*	Toteutuu	1
IoT-laitteiden hallinta alustan avulla*	Toteutuu	1
Pääsynhallintaprosessi**	Toteutuu	1
Salauksen käyttö datan lähettämisessä***	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Access Controls -ohjeistus

*** OWASP Protect Data Everywhere -ohjeistus

Pääsynhallinta alustan toiminnallisuuksiin

Varsinaista käyttölokia CityIoT FIWARE -alustassa ei ole. Alustan ylläpitäjällä on kuitenkin pääsy Nginx proxy:n pääsylokeihin, joiden avulla voidaan tarkastella alustan käyttöä jossain määrin. Hankkeen vaatimukset eivät kuitenkaan toteudu tämän osalta.

Taulukko 4.9 *CityIoT FIWARE pisteytys - Pääsynhallinta alustan toiminnallisuuksiin*

Vaatimus	Tulos	Pisteet
Pääsy alustan lokitietoihin*	Ei toteudu	0
Käyttölokien käyttö*	Ei toteudu	0

* CityIoT-hankkeen vaatimus

** OWASP Implement Security Logging and Monitoring -ohjeistus

4.2 Thingsboard

Datasettien hallinta

ThingsBoardin datasettien luonti tapahtuu automaattisesti, kun laite, anturi tai jokin muu ulkoinen lähde lähettää dataa alustalle. Dataa voidaan rikastaa metatiedon avulla sääntöketju-toiminnallisuudella, mutta käyttöoikeudet määräytyvät ko. laitteelle määriteltyjen roolien mukaan. Jos käyttäjä yrittää lisätä ulkoisen rajapinnan (REST API) kautta dataa sellaiseen datasettiin johon hänellä ei ole oikeutta, ilmoitetaan siitä hänelle virheilmoituksella (unauthorized data). Alustan hallintatyökalun avulla käyttäjä voi lisätä dataa vain sellaisiin datasetteihin, joihin hänellä on oikeus. Lisäksi alustalta löytyy datantuontityökalu, jolla voidaan tuoda CSV- tai XLS-muotoista dataa alustalle nk. eräajona. Datan omistajat (liiketoimintayksikön ylläpitäjät) voivat muokata ja poistaa organisaationsa datasettejä suoraan alustalta käyttöliittymän tai ulkoisen rajapinnan kautta. Datasetin poisto poistaa myös siihen liitetyt käyttöoikeudet. Datasetteja ei voi kategorisoida näkyvyyden suhteen (avoin/salainen).

ThingsBoard-alustassa voidaan sisään tulevaa dataa rikastaa erilaisilla sääntöketjuilla. Kuitenkaan itse alusta ei varsinaisesti tarvitse kaikkia näitä tietoja, vaan niiden avulla voidaan välittää tietoa esimerkiksi kolmannen osapuolen sovellukselle, joka voi jatkojalostaa dataa mm. kaupalliseen tarkoitukseen. Datan rikastaminen on monipuolista ja metatiedon lisääminen on joustavaa, mutta käyttöoikeuksien määrittäminen ei tapahdu tätä kautta, vaan se toteutuu roolipohjaisen käyttöoikeuksien avulla (RBAC).

Taulukko 4.10 *Thingsboard pisteytys - Datasettien hallinta*

Vaatimus	Tulos	Pisteet
Datasettien luonti*	Toteutuu	1
Datan lisääminen datasettiin*	Toteutuu	1
Datasetin muokkaaminen tai poisto*	Toteutuu	1
Metatiedon lisääminen alustassa olevaan dataan*	Ei toteudu	0
Datan kategorisointi avoimeksi ja salaiseksi**	Ei toteudu	0
Salauksen käyttö datan lähettämisessä**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Protect Data Everywhere -ohjeistus

Käyttäjähallinta

Kuten aikaisemmissa vaatimuksissa on käynyt ilmi, ThingsBoardin käyttäjähallinta on roolipohjainen (RBAC). Alustaan voidaan lisätä liiketoimintayksiköitä, yksilöinä tai organisaatioina, järjestelmän ylläpitäjän (system administrator) toimesta. Liiketoimintayksikköön voidaan lisätä käyttäjiä, jotka puolestaan toimivat ko. liiketoimintayksikön ylläpitäjinä (tenant administrator). Nämä käyttäjät voivat lisätä entiteettejä (mm. laitteita, omaisuuksia, näkymiä), käyttäjiä sekä asiakkaita. He myös määrittelevät säännöt datan käyttöoikeuksille ja voivat hallinnoida tätä erilaisten ryhmien (user group, device group, customer group) avulla. Liiketoimintayksikön ylläpitäjät määrittelevät käyttäjäroolit, jotka määrittelevät käyttäjille niiden valtuudet alustan toiminnallisuuksia varten. Käyttäjät voivat kirjautua alustalle alustaan määriteltyjen käyttäjätunnusten avulla, mutta todentaminen voidaan myös tehdä ulkoisen palvelun avulla käyttäen OAuth 2.0 -autentikointiprotokollaa.

Taulukko 4.11 *Thingsboard pisteytys - Käyttäjähallinta*

Vaatimus	Tulos	Pisteet
Alustan käyttäjien hallinta*	Toteutuu	1
Digitaalinen identiteetti**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** NIST Digital Identity Guidelines -ohjeistus

Pääsynhallinta

Liiketoimintayksikön ylläpitäjät määrittelevät kaikki ko. liiketoimintayksikön käyttöoikeudet alustalle tulevaan dataan sekä datan näkyvyyden (salainen/julkinen). Oikeuksia voivat olla mitkä tahansa variaatiot CRUD-toiminnoista, mutta myös muita erityisoikeuksia voidaan määritellä, kuten vain luku- tai kirjoitusoikeudet. Oikeuksien tarkistus suoritetaan aina, kun käyttäjä suorittaa pyynnön alustan dataan. Tarpeen vaatiessa ylläpitäjät voivat myös muokata käyttöoikeuksia jo luoduille dataseteille.

Taulukko 4.12 *Thingsboard pisteytys - Pääsynhallinta*

Vaatimus	Tulos	Pisteet
Käyttöoikeuksien ja -tasojen määrittely datasettiin tai muuhun dataan*	Toteutuu	1
Datasetin käyttöoikeuksien muokkaaminen tai kaiken datan muokkaus*	Toteutuu	1
Pääsynhallintaprosessi**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Acess Controls -ohjeistus

Pääsynhallinta kaupalliseen tietoon

ThingsBoard-alusta ei itsessään tarjoa datan kaupallistamiseen liittyviä ominaisuuksia. Sen pääasiallinen tarkoitus on vain kerätä ja visualisoida dataa. Datan jatkojalostaminen voidaan kuitenkin toteuttaa kolmannen osapuolen sovelluksilla ja niille voidaan lähettää haluttua dataa ulkoisen rajapinnan kautta sääntöketjutoiminnallisuuden avulla.

Taulukko 4.13 *Thingsboard pisteytys - Pääsynhallinta kaupalliseen tietoon*

Vaatimus	Tulos	Pisteet
Kaupallisten oikeuksien määrittely tietokokoelman käyttöön*	Ei toteudu	0
Tietokokoelman käyttöoikeuksien myynti alustalta kolmansille osapuolille*	Ei toteudu	0
Tietokokoelman omistuksen siirtäminen tai muokkaus*	Toteutuu	1

* CityIoT-hankkeen vaatimus

Datan käyttö- ja muokkausoikeudet

Roolipohjaisen käyttäjähallinnan takia ThingsBoardista ei voi nähdä muiden käyttäjien ja organisaatioiden lisäämiä datasettejä. Käyttäjät voivat nähdä vain ainoastaan omalle roolilleen sallittuja datoja. Alustalla voidaan kuitenkin tehdä erilaisia näkymiä (entity view), joille voidaan määritellä käyttöoikeudet laajemmalla käyttäjäkunnalle. Näkymiin voidaan antaa kuitenkin vain lukuoikeudet, eli näkymässä olevan datan hallinta on vain näkymän omistajalla.

Liiketoimintayksikön ylläpitäjät voivat mahdollistaa käyttäjille ilmoitusten lähettämisen aina silloin, kun laitteet saavat uusia mittaustuloksia. Tämä tapahtuu

sääntöketju-toiminnallisuuden avulla ja ilmoituksen voi saada, joko sähköpostiin tai johonkin kolmannen osapuolen järjestelmään tai sovellukseen, esimerkiksi mobiilisovellukseen.

Käyttäjät voivat kysyä omalta ylläpitäjältä mitä tietoja hänestä on tallennettu järjestelmään. Tähän ei ole kuitenkaan suoraa toiminnallisuutta alustassa, vaan ylläpitäjän täytyy kerätä tarvittavat tiedot käsin järjestelmästä. Tarvittaessa myös järjestelmän ylläpitäjää tulisi pystyä konsultoimaan ko. asiaan liittyen. Käyttäjät voivat myös itse poistaa käyttäjätilinsä järjestelmästä. Hallintatyökalun avulla käyttäjä voi nähdä myös profiilistaan, mitä tietoja järjestelmässä on tallennettuna.

Taulukko 4.14 *Thingsboard pisteytys - Datan käyttö- ja muokkausoikeudet*

Vaatimus	**Tulos**	**Pisteet**
Tiedon ja tietolähteiden saatavuuden tarkastelu*	Ei toteudu	0
Käyttäjä voi käsitellä ja muokata tietoa alustassa*	Toteutuu	1
Ilmoitusten vastaanottamisen tilaustiedon muuttuessa alustassa on mahdollista*	Toteutuu	1
Käyttäjät voivat nähdä mitä tietoa heistä on tallennettuna alustaan*	Toteutuu	1
Pääsynhallintaprosessi**	Toteutuu	1
Salauksen käyttö datan lähettämisessä***	Toteutuu	1
Henkilötietojen hallinta****	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Access Controls -ohjeistus

*** OWASP Protect Data Everywhere -ohjeistus

**** Tietosuojavaltuutetun toimisto Henkilötiedot

Ulkoisten tietolähteiden lisääminen alustaan

Ulkoisen tietolähteen lisääminen on mahdollista ThingsBoardiin. Tämä tapahtuu erilaisten alustaintegraatioiden avulla. Valmiita integraatioita löytyy kuten HTTP-, MQTT-, OPC-UA- ja LoRaWAN-protokollalle, Sigfox backendille ja lukuisille IoT-laitteille, jotka käyttävät UDP- ja TCP-integraatioita. Alustalta löytyvät valmiiksi myös AWS IoT -, IBM Watson -ja Azure Event Hub -integraatiot.

Viestin saavuttua ulkoiselta alustalta ThingsBoardille, sen kuorma validoidaan alustakohtaisesti tietoturvasääntöjen mukaan. Kun viesti on todettu, kutsutaan Uplink Data Converteria keräämään tarvittavat tiedot saapuneesta viestistä. Viesti muutetaan tietolähdekohtaisesta ThingsBoardin käyttämään muotoon.

Taulukko 4.15 *Thingsboard pisteytys - Ulkoisten tietolähteiden lisääminen alustaan*

Vaatimus	Tulos	Pisteet
Alustan liittäminen toiseen IoT-alustaan*	Toteutuu	1
Broken Object Level Authorization**	Toteutuu	1
Excessive Data Exposure**	Toteutuu	1
Broken Function Level Authorization**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP API Security Project -ohjeistus

Alustaan liitettyjen IoT-laitteiden hallita

Laitteiden hallinta ThingsBoard:ssa toimii liiketoimintayksikön ylläpitäjien toimesta. Heidän vastuullaan on laitteiden lisääminen ja poistaminen sekä niiden käyttöoikeuksien hallinta. Käyttäjät voivat kuitenkin lähettää laitteille toimintoja kuten mittaustulosten kyselyjä ja laiteversion päivittämispyyntöjä. Laitteita voidaan hallita alustan käyttöliittymän tai ulkoisen rajapinnan (REST API) kautta.

Taulukko 4.16 *Thingsboard pisteytys - Alustaan liitettyjen IoT-laitteiden hallinta*

Vaatimus	Tulos	Pisteet
IoT-laitteiden lisääminen alustaan*	Toteutuu	1
IoT-laitteiden hallinta alustan avulla*	Toteutuu	1
Pääsynhallintaprosessi**	Toteutuu	1
Salauksen käyttö datan lähettämisessä***	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Access Controls -ohjeistus

*** OWASP Protect Data Everywhere -ohjeistus

Pääsynhallinta alustan toiminnallisuuksiin

Käyttölokien hallinta ThingsBoard:ssa on mahdollista järjestelmän ylläpitäjän ja liiketoimintayksikön ylläpitäjien toimesta. Käyttäjistä tallennetaan lokitietoja, jotka sisältävät tapahtumia käyttäjien tekemistä muutoksista mm. laitteisiin, omaisuuksiin, sääntöketjuihin sekä UI-näkymiin (dashboard).

Taulukko 4.17 *Thingsboard pisteytys - Pääsynhallinta alustan toiminnallisuuksiin*

Vaatimus	Tulos	Pisteet
Pääsy alustan lokitietoihin*	Toteutuu	1
Käyttölokien käyttö*	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Implement Security Logging and Monitoring -ohjeistus

4.3 Kaa IoT

Datasettien hallinta

Datasettien luonti Kaa IoT:ssa tehdään niin, että todellisen maailman fyysiset laitteet mallinnetaan “digitaalisiksi kaksosiksi” järjestelmään hallintatyökalun avulla. [55] Näitä virtuaalisia malleja kutsutaan Kaa IoT -alustassa “endpointeiksi” (EP). Malleihin voidaan myös lisätä avain-arvo-parien avulla metatietoa. Datasettien muokkaus tapahtuu myös hallintatyökalun avulla.

Käyttöoikeudet puolestaan määräytyvät siten, että järjestelmään voidaan luoda erilaisia rooleja, jotka määrittelevät mitä toimintoja ko. roolissa toimiva käyttäjä voi tehdä. Esimerkiksi “Laitteiden hallinta”-roolilla pystyttäisiin hallitsemaan kaikkia yhden liiketoimintayksikön laitteita. Yksittäiselle datasetille ei voida asettaa erikseen käyttöoikeuksia metadatan avulla, vaan ne määräytyvät käyttäjän roolin kautta.

Datan lisääminen järjestelmään tapahtuu ulkoisen rajapinnan kautta (REST API). Kullekin EP:lle määritellään tai generoidaan automaattisesti tunniste, jota tulee käyttää datan lähettämiseen alustalle. Tunnisteet ovat yksilöllisiä yhdessä Kaa-sovelluksessa olevaa tiettyä EP:tä kohden. Lähettäjä (yleensä asiakassovellus) todennetaan järjestelmän omalla käyttäjä- ja käyttöoikeuksien hallintajärjestelmällä. Todentaminen tapahtuu käyttäjätunnuksella ja salasanalla, joiden avulla saadaan lähettäjälle oma tunniste (access token). Tunnisteella mahdollistetaan pääsy alustan rajapintoihin. Yksilöllisten access- ja EP-tunnisteiden avulla asiakassovellukset voivat lähettää dataa Kaa IoT -alustalle. Jos nämä tiedot eivät löydy lähettäjän pyynnössä, palautetaan siitä virheilmoitus. Erillistä datan massatuontityökalua (eräajo) alustalla ei ole.

Datasetin poisto tapahtuu hallintatyökalun avulla ja vain itse datan lähde poistuu. Jos datasettiä varten on jouduttu luomaan erilliset käyttöoikeudet, ne eivät poistu, vaan ne pitää käydä poistamassa erikseen käyttöoikeuksien hallintajärjestelmästä.

EP:hin lisättävän metadatan avulla ei voida hallita käyttöoikeuksia, vaan nämä hoituvat erillisen käyttäjä- ja käyttöoikeuksien hallintajärjestelmän avulla. Myöskään EP:n kategorisointia näkyvyyden mukaan (avoin/salainen) ei ole mahdollista. Metadataa voidaan kuitenkin laajentaa joustavasti muihin tarpeisiin, esimerkiksi datan analysointia ja alustan eri näkymien suodattimia varten.

Taulukko 4.18 *Kaa IoT pisteytys - Datasettien hallinta*

Vaatimus	**Tulos**	**Pisteet**
Datasettien luonti*	Toteutuu	1
Datan lisääminen datasettiin*	Toteutuu	1
Datasetin muokkaaminen tai poisto*	Toteutuu	1
Metatiedon lisääminen alustassa olevaan dataan*	Ei toteudu	0
Datan kategorisointi avoimeksi ja salaiseksi**	Ei toteudu	0
Salauksen käyttö datan lähettämisessä**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Protect Data Everywhere -ohjeistus

Käyttäjähallinta

Kaa IoT- alustan käyttäjiä ja käyttöoikeuksia hallitaan erillisen Keycloak-järjestelmän avulla. Keycloak on avoimeen lähdekoodiin perustuva identiteetin ja käyttöoikeuksien hallintajärjestelmä. Liiketoimintayksikön ylläpitäjät hallinnoivat käyttäjiä ja niiden käyttöoikeuksia tämä järjestelmän avulla. Koska Kaa IoT -alusta on mikropalvelupohjainen, myös kaikki siihen liitetyt sovellukset tulee konfiguroida käyttämään tätä palvelua. Kaa-ylläpitäjäroolissa (Kaa administrator) oleva käyttäjä pystyy lisäämään, muokkaamaan ja poistamaan liiketoimintayksikön ylläpitäjiä (tenant administrator). Nämä ylläpitäjät puolestaan hallinnoivat alustan sovelluksia (applications, clients), laitteita (endpoints), käyttäjiä ja niiden oikeuksia sekä alustan erilaisia toimintoja.

Taulukko 4.19 *Kaa IoT pisteytys - Käyttäjähallinta*

Vaatimus	Tulos	Pisteet
Alustan käyttäjien hallinta*	Toteutuu	1
Digitaalinen identiteetti**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** NIST Digital Identity Guidelines -ohjeistus

Pääsynhallinta

Liiketoimintayksiköiden ylläpitäjät voivat määritellä alustan käyttöoikeuksia. Oikeudet ovat roolipohjaisia ja voivat olla mitkä tahansa variaatiot CRUD-toiminnoista sekä muunlaisia erityisoikeuksia. Oikeudet tarkistetaan käyttäjiltä sekä liitetyiltä sovelluksilta Keycloak-järjestelmän avulla aina kun pyyntö saapuu Kaa IoT- alustalle. Käyttöoikeuksia yksittäiselle datasetille ei ole automaattisesti saatavilla alustan oman hallintatyökalun kautta, mutta määrittelemällä tarkat tiedot (esimerkiksi EP ID tietoa käyttäen) Keycloak-järjestelmään tällainenkin toiminto voidaan toteuttaa. Ylläpidollisesti tämä kuitenkin saattaa olla haastavaa.

Taulukko 4.20 *Kaa IoT pisteytys - Pääsynhallinta*

Vaatimus	Tulos	Pisteet
Käyttöoikeuksien ja -tasojen määrittely datasettiin tai muuhun dataan*	Toteutuu	1
Datasetin käyttöoikeuksien muokkaaminen tai kaiken datan muokkaus*	Ei toteudu	0
Pääsynhallintaprosessi**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Acess Controls -ohjeistus

Pääsynhallinta kaupalliseen tietoon

Kaa IoT- alusta ei tarjoa työkaluja IoT-datan kaupallistamiseen. Mikropalveluarkkitehtuuri ja selkeät ulkoiset rajapinnat kuitenkin mahdollistavat alustan laajentamisen, tai mahdollisesti ulkopuolisen datan kaupallistamisjärjestelmän liittämisen osaksi alustaa.

Taulukko 4.21 *Kaa IoT pisteytys - Pääsynhallinta kaupalliseen tietoon*

Vaatimus	Tulos	Pisteet
Kaupallisten oikeuksien määrittely tietokokoelman käyttöön*	Ei toteudu	0
Tietokokoelman käyttöoikeuksien myynti alustalta kolmansille osapuolille*	Ei toteudu	0
Tietokokoelman omistuksen siirtäminen tai muokkaus*	Toteutuu	1

* CityIoT-hankkeen vaatimus

Datan käyttö- ja muokkausoikeudet

Kaa IoT:ssa voidaan hakea datasta tietoa tekemällä kutsuja ulkoisen rajapinnan avulla. Käyttäjä tai sovellus voi kuitenkin hakea tietoa vain niistä liiketoimintayksikön datoista, joihin hänellä on pääsyoikeudet. Alustaan määritellyt liiketoimintayksiköt on jaettu erillisiin alueisiin (realms) ja ne ovat eristettyinä toisistaan Keycloak-järjestelmän avulla.

Käyttäjät voivat lisätä erilaisia hälytyksiä haluamaansa datasettiin hallintatyökalun avulla saadakseen niistä itselleen muistutuksen tekstiviestillä, sähköpostilla tai johonkin muuhun haluttuun sovellukseen (kuten esimerkiksi Slack). Liiketoimintayksikön ylläpitäjät voivat myös määritellä valmiiksi hälytyksiä pohjautuen yleisimpiin käyttötapauksiin, esimerkiksi yhteys laitteeseen on ollut poikki yli minuutin tai mittaustuloksissa on selvästi poikkeavia arvoja.

Käyttäjillä on mahdollisuus kysyä liiketoimintayksikön ylläpitäjältä mitä tietoja hänestä on tallennettu järjestelmään. Kaikki tarvittavat tiedot saadaan haettua Keycloak-järjestelmästä ylläpitäjien toimesta palvelurajapinnan avulla tai käyttäjä voi omasta profiilistaan nähdä mitä tietoja järjestelmässä on tallennettuna hänestä.

Taulukko 4.22 *Kaa IoT pisteytys - Datan käyttö- ja muokkausoikeudet*

Vaatimus	Tulos	Pisteet
Tiedon ja tietolähteiden saatavuuden tarkastelu*	Ei toteudu	0
Käyttäjä voi käsitellä ja muokata tietoa alustassa*	Toteutuu	1
Ilmoitusten vastaanottamisen tilaustiedon muuttuessa alustassa on mahdollista*	Toteutuu	1
Käyttäjät voivat nähdä mitä tietoa heistä on tallennettuna alustaan*	Toteutuu	1
Pääsynhallintaprosessi**	Toteutuu	1
Salauksen käyttö datan lähettämisessä***	Toteutuu	1
Henkilötietojen hallinta****	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Access Controls -ohjeistus

*** OWASP Protect Data Everywhere -ohjeistus

**** Tietosuojavaltuutetun toimisto Henkilötiedot

Ulkoisten tietolähteiden lisääminen alustaan

Ulkoisen tietolähteen liittäminen Kaa IoT -alustalle tapahtuu sen Data Collection Extension -laajennuksen (DCX) avulla. DCX-toteutus vastaanottaa ulkoisesta tietolähteestä tulevan datan ja välittää sen edelleen tiedonkeräysadaptereille tallentamista ja/tai prosessointia varten. Alusta tukee valmiiksi MQTT- ja HTTP-protokollia, mutta muitakin protokollia on mahdollista implementoida tarvittaessa.

Taulukko 4.23 *Kaa IoT pisteytys - Ulkoisten tietolähteiden lisääminen alustaan*

Vaatimus	Tulos	Pisteet
Alustan liittäminen toiseen IoT-alustaan*	Toteutuu	1
Broken Object Level Authorization**	Toteutuu	1
Excessive Data Exposure**	Toteutuu	1
Broken Function Level Authorization**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP API Security Project -ohjeistus

Alustaan liitettyjen IoT-laitteiden hallita

Käyttäjien käyttöoikeudet määräytyvät heille määritellyn rooliin muokaisesti. Roolille voi määritellä käyttöoikeuksia, joilla on mahdollista hallita laitteita, lisätä, poistaa ja muokata laitteiden tietoja sekä lähettää toimintoja laitteille (kuten mittaustulosten kyselyjä ja laiteversion päivittämisen). Laitteiden hallinta tapahtuu hallintatyökalun tai ulkoisen rajapinnan avulla. Viestinvälityksessä käytetään aina salausta, mutta myös selkokielinen viestintä laitteiden välillä on mahdollista.

Taulukko 4.24 *Kaa IoT pisteytys - Alustaan liitettyjen IoT-laitteiden hallinta*

Vaatimus	Tulos	Pisteet
IoT-laitteiden lisääminen alustaan*	Toteutuu	1
IoT-laitteiden hallinta alustan avulla*	Toteutuu	1
Pääsynhallintaprosessi**	Toteutuu	1
Salauksen käyttö datan lähettämisessä***	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Access Controls -ohjeistus

*** OWASP Protect Data Everywhere -ohjeistus

Pääsynhallinta alustan toiminnallisuuksiin

Testatussa Kaa IoT -alustan versiossa (v1.2.0) ei ole mahdollista käyttää audit log -toiminnallisuutta. Keycloak-järjestelmässä on Events-toiminnalisuus, mutta jostain syystä sitä ei ole mahdollista laittaa päälle tässä ko. Kaa IoT -integraatiossa.

Taulukko 4.25 *Kaa IoT pisteytys - Pääsynhallinta alustan toiminnallisuuksiin*

Vaatimus	Tulos	Pisteet
Pääsy alustan lokitietoihin*	Ei toteudu	0
Käyttölokien käyttö*	Ei toteudu	0

* CityIoT-hankkeen vaatimus

** OWASP Implement Security Logging and Monitoring -ohjeistus

4.4 Thinger.io

Datasettien hallinta

Laitteiden lisäämisen jälkeen Thinger.io:ssa pitää myös määritellä datasetti (data bucket), joka listaa valitusta laitteesta tulevan datan. Datasetin voi jakaa muille saman projektin käyttäjille. Jos käyttäjä yrittää lisätä ulkoisen rajapinnan (REST API) kautta dataa sellaiseen datasettiin johon hänellä ei ole oikeutta, ilmoitetaan siitä hänelle virheilmoituksella (unauthorized data). Datasettien tuonti onnistuu eräajona CSV-muotoisista tiedostoista datantuontityökalulla. Datasetin poisto tapahtuu hallintatyökalun tai ulkoisen rajapinnan avulla ja vain itse datan lähde poistuu. Lisäksi alustalta on mahdollista tyhjentää pelkkä datasetin sisältö ilman, että datasettien tai laitteiden konfigurointi muuttuu.

Laitteille voidaan lisätä ominaisuuksia (properties), joiden avulla voidaan tallentaa lisätietoja laitteista (metadata) ylläpitäjien ja kehittäjien toimesta. Tämän avulla ei voida kuitenkaan kategorisoida dataa näkyvyyden mukaan (avoin/salainen) tai hallita datan käyttöoikeuksia. Tätä lisätietoa voidaan esittää hallintatyökalun kojetauluilla, mutta muita käyttötapauksia tälle tiedolle ei ole alustassa. Ulkoisen rajapinnan avulla tätä tietoa voidaan kuitenkin jalostaa laajemmin.

Taulukko 4.26 *Thinger.io pisteytys - Datasettien hallinta*

Vaatimus	Tulos	Pisteet
Datasettien luonti*	Toteutuu	1
Datan lisääminen datasettiin*	Toteutuu	1
Datasetin muokkaaminen tai poisto*	Toteutuu	1
Metatiedon lisääminen alustassa olevaan dataan*	Ei toteudu	0
Datan kategorisointi avoimeksi ja salaiseksi**	Ei toteudu	0
Salauksen käyttö datan lähettämisessä**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Protect Data Everywhere -ohjeistus

Käyttäjähallinta

Thinger.io:ssa ainoastaan järjestelmän ylläpitäjät voivat hallita käyttäjiä. Ylläpito voi määritellä kolmen tyyppisiä käyttäjiä: ylläpitäjiä (administrator), kehittäjiä (developer) ja peruskäyttäjiä (user). Kehittäjätili on tarkoitettu käytettävän organisaation sidosryhmille ja niillä onkin rajatut käyttöoikeudet alustan hallinnollisiin ominaisuuksiin (käyttäjät, domain-asetukset). Peruskäyttäjät puolestaan voivat nähdä

vain heille osoitettujen projektien resursseja. Järjestelmän hallintatyökaluun tunnistaudutaan käyttäjätunnuksen ja salasanan avulla. Ulkoista rajapintaa käytetään puolestaan erillisen tunnisteen avulla. Thinger.io-alustan ylläpitäjä (administrator) pystyy hallitsemaan alustan kaikkia toimintoja: IoT-laitteiden hallinta, palvelinlaajennusten ylläpito, projektien ja omaisuuden organisointi, käyttäjien hallinta sekä domainin (liiketoimintayksikköjen) ylläpito. Ylläpitäjiä voidaan määritellä useampia yhtä alustan instanssia kohden.

Taulukko 4.27 *Thinger.io pisteytys - Käyttäjähallinta*

Vaatimus	Tulos	Pisteet
Alustan käyttäjien hallinta*	Toteutuu	1
Digitaalinen identiteetti**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** NIST Digital Identity Guidelines -ohjeistus

Pääsynhallinta

Pääsynhallinta Thinger.io:ssa tapahtuu erillisten tunnisteiden (access token) avulla, joita ylläpitäjät voivat hallita. Tämä johtuu siitä, että järjestelmä on toteutettu täysin REST API -arkkitehtuuria käyttäen. Käyttäjän tekemien pyyntöjen (palvelimella oleviin resursseihin) tulee sisältää tarvittava tunniste tunnistautumista ja käyttöoikeuksia varten. Käyttöoikeuksia voidaan luoda tarpeen mukaan resurssien täysistä CRUD-toiminnoista erilaisiin listausoikeuksiin (vain lukuoikeudet). Käyttöoikeuksia voidaan muokata jälkikäteen ja muutokset tulevat voimaa heti muokkauksen jälkeen. Datan näkyvyyden määritteleminen datan omistajan toimesta ei ole mahdollista.

Taulukko 4.28 *Thinger.io pisteytys - Pääsynhallinta*

Vaatimus	Tulos	Pisteet
Käyttöoikeuksien ja -tasojen määrittely datasettiin tai muuhun dataan*	Toteutuu	1
Datasetin käyttöoikeuksien muokkaaminen tai kaiken datan muokkaus*	Ei toteudu	0
Pääsynhallintaprosessi**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Acess Controls -ohjeistus

Pääsynhallinta kaupalliseen tietoon

Kaupallistamiseen liittyviä toiminnallisuuksia Thinger.io-alustasta ei löydy. REST API -arkkitehtuurin myötä järjestelmää olisi ainakin teoriassa mahdollista laajentaa kolmannen osapuolen sovelluksilla.

Taulukko 4.29 *Thinger.io pisteytys - Pääsynhallinta kaupalliseen tietoon*

Vaatimus	Tulos	Pisteet
Kaupallisten oikeuksien määrittely tietokokoelman käyttöön*	Ei toteudu	0
Tietokokoelman käyttöoikeuksien myynti alustalta kolmansille osapuolille*	Ei toteudu	0
Tietokokoelman omistuksen siirtäminen tai muokkaus*	Toteutuu	1

* CityIoT-hankkeen vaatimus

Datan käyttö- ja muokkausoikeudet

Thinger.io-alustasta löytyy avoin palvelinrajapintakuvaus, jonka avulla käyttäjä voi tehdä pyyntöjä alustalle. Jos käyttäjällä ei ole tarvittavia käyttöoikeuksia haluttuun tietoon, annetaan tästä kuvaava virheviesti (unauthorized data). Alustalla voidaan hallita useamman liiketoimintayksikön resursseja, mutta käyttäjät voivat nähdä vain oman domainin resursseja. Lisäksi käyttäjällä voi olla käyttöoikeudet vain yhteen projektiin, joka kaventaa resurssitarjontaa entisestään. Käyttöoikeuksien lisäämisestä käyttäjät voivat olla yhteydessä alustan ylläpitäjiin.

Laitteiden mittaustulosten muuttuessa on mahdollista välittää tämä tieto halutuille käyttäjille EP-toiminnallisuuden avulla. Ilmoituksia voidaan lähettää sähköpostitse tai kolmannen osapuolen sovelluksiin kuten Telegram-sovellukseen.

Thinger.io-alustasta voidaan User API -rajapinnan avulla tarkistella tietoja käyttäjistä. Tämän ominaisuuden dokumentaatio kuitenkin oli työn alla tämän työn tekohetkellä, joten tarkempaa tietoa siitä mitä tietoja käyttäjästä voisi saada ei saatu tutkittavaksi. Hallintatyökalun avulla käyttäjä voi profiilistaan nähdä mitä tietoja järjestelmässä on tallennettuna.

Taulukko 4.30 *Thinger.io pisteytys - Datan käyttö- ja muokkausoikeudet*

Vaatimus	Tulos	Pisteet
Tiedon ja tietolähteiden saatavuuden tarkastelu*	Ei toteudu	0
Käyttäjä voi käsitellä ja muokata tietoa alustassa*	Toteutuu	1
Ilmoitusten vastaanottamisen tilaustiedon muuttuessa alustassa on mahdollista*	Toteutuu	1
Käyttäjät voivat nähdä mitä tietoa heistä on tallennettuna alustaan*	Toteutuu	1
Pääsynhallintaprosessi**	Toteutuu	1
Salauksen käyttö datan lähettämisessä***	Toteutuu	1
Henkilötietojen hallinta****	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Access Controls -ohjeistus

*** OWASP Protect Data Everywhere -ohjeistus

**** Tietosuojavaltuutetun toimisto Henkilötiedot

Ulkoisten tietolähteiden lisääminen alustaan

Thinger.io-alustaan ulkoisten tietolähteiden liittäminen tapahtuu erillisen HTTP-laitetyypin avulla. Tämän toiminnallisuuden avulla alustaan voidaan liittää mitä tahansa ulkoisia laitteita (tai järjestelmiä) ja konfiguroida niistä tulevaa dataa samalla tavalla kuin muistakin laitteista. Rajoitteena on kuitenkin se, että ulkoisen lähteen pitää pystyä lähettämään HTTP POST -viestejä sisältäen JSON-koodattua dataa.

Taulukko 4.31 *Thinger.io pisteytys - Ulkoisten tietolähteiden lisääminen alustaan*

Vaatimus	Tulos	Pisteet
Alustan liittäminen toiseen IoT-alustaan*	Toteutuu	1
Broken Object Level Authorization**	Toteutuu	1
Excessive Data Exposure**	Toteutuu	1
Broken Function Level Authorization**	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP API Security Project -ohjeistus

Alustaan liitettyjen IoT-laitteiden hallita

Ylläpitäjät ja kehittäjät voivat hallita laitteita Thinger.io-alustassa. He vastaavat laitteiden lisäämisestä ja poistamisesta sekä käyttöoikeuksien asettamisesta. Molemmat käyttäjät voivat myös lähettää toimintoja laitteille kuten mittaustulosten kyselyjä ja laiteversion päivittämisen. Laitteiden hallintaa voidaan tehdä hallintatyökalun tai ulkoisen rajapinnan kautta.

Taulukko 4.32 *Thinger.io pisteytys - Alustaan liitettyjen IoT-laitteiden hallinta*

Vaatimus	Tulos	Pisteet
IoT-laitteiden lisääminen alustaan*	Toteutuu	1
IoT-laitteiden hallinta alustan avulla*	Toteutuu	1
Pääsynhallintaprosessi**	Toteutuu	1
Salauksen käyttö datan lähettämisessä***	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Enforce Access Controls -ohjeistus

*** OWASP Protect Data Everywhere -ohjeistus

Pääsynhallinta alustan toiminnallisuuksiin

Käytetyssä Thinger.io-alustan versiossa (v2.9.7) ei ollut erillistä audit log -toiminnallisuutta. Alustan palvelimen lokituksesta voidaan kuitenkin nähdä tietoa käyttäjien tekemistä tapahtumista, mutta yksittäisen käyttäjän tapahtumien etsiminen on suoritettava ylläpitäjien toimesta käsin.

Taulukko 4.33 *Thinger.io pisteytys - Pääsynhallinta alustan toiminnallisuuksiin*

Vaatimus	Tulos	Pisteet
Pääsy alustan lokitietoihin*	Toteutuu	1
Käyttölokien käyttö*	Toteutuu	1

* CityIoT-hankkeen vaatimus

** OWASP Implement Security Logging and Monitoring -ohjeistus

5 Tulosten käsittely

Tässä kappaleessa koostetaan yhteen kappaleessa 4 saadut tulokset. Kappaleessa 6 koostetaan yhteenveto siitä kuinka hyvin vertailtavat alustat toteuttivat vaatimukset.

Tulokset on esitetty taulukkomuotoisena missä sarakkeiden otsikoissa on tieto vertailtavasta vaatimuksesta tai ohjeistuksesta. Otsikoissa on myös sulkuihin merkitty suurin mahdollinen pistemäärä mitä alusta voi saada kyseisen vaatimuksen tai ohjeistuksen vertailusta. Taulukoiden riveillä on esitetty vertailtavat alustat sekä alustojen saamat pistemäärät vertailusta.

5.1 Sopimusasiat

Vertailtavissa IoT-alustoissa ei ollut mahdollisuutta hallita sopimuksellisia asioita, joten tämän vaatimuksen osalta vertailua ei ole suoritettu. Yhteenvetona voidaan kuitenkin todeta, että sopimukselliset asiat kuten datan käyttöoikeussäännöt, pääsyoikeuksien määritteleminen sekä datan avoimuuden tasot on käsitelty muiden vaatimusten yhteydessä.

5.2 Datasettien hallinta

Tarkasteltavissa IoT-alustoissa ei ollut mahdollisuutta asettaa tiedon kategoriaa: avoin tai salattu tieto. Metadatan avulla kategorisointia voisi yrittää tehdä, mutta vaikutukset eivät kuitenkaan olisi halutun kaltaisia: tietoa pystyttäisiin vain piilottamaan toisilta käyttäjiltä. Lisäksi useimmista alustoista ei ollut saatavilla tietoa siitä salataanko tallennettua dataa vai ei. Ainoastaan Kaa IoT -alustan dokumentaatiosta löytyi tieto datan salaamisesta. Jokaisessa alustassa tiedonsiirto on aina salattua asiakkaiden ja palvelimien välille sekä järjestelmän sisäisten yhteyksien välillä. Kaa IoT -alustassa on myös mahdollisuus liittää laitteita joiden välinen yhteys ei ole salattu ja viestit kulkevat selkokielisenä. Tämä siksi, että kaikissa IoT-laitteissa ei välttämättä ole mahdollisuutta salauksen käyttöön, mikä selittyy laitteen vähäisestä käyttömuistista.

Taulukko 5.1 *Datasettien hallinta - tulokset*

Vertailu / Alusta	CityIoT-vaatimus (max. 4)	OWASP: Protect Data Everywhere (max. 2)
CityIoT FIWARE	2	1
Thingsboard	3	1
Kaa IoT	3	1
Thinger.io	3	1

5.3 Käyttäjähallinta

Työssä vertailtavissa IoT-alustoissa käyttäjien hallinta tapahtuu lähes aina salattuja yhteyksiä käyttäen erillisellä hallintatyökalulla. Poikkeuksena CityIoT FIWARE, missä käyttäjiä hallitaan erillisten palvelimelle tallennettujen konfiguraatiotiedostojen avulla. Lisäksi käyttäjän henkilötietoja ei ole pakko syöttää järjestelmiin, mutta mahdollisuus tähän kuitenkin on annettu. Jälleen kerran CityIoT FIWARE poikkeaa muista, eikä siinä ole pakko antaa mitään tietoa käyttäjästä, vaan käyttäjät identifioidaan vain tunnisteen avulla. Muissa alustoissa sähköpostitunnuksen asettaminen käyttäjätiliin on pakollista, koska tämä tieto toimii käyttäjät yksilöivänä tietona järjestelmissä sekä sitä voidaan käyttää OAuth2-protokollan yhteydessä.

Suurimmassa osassa järjestelmiä oli myös mahdollisuus käyttää ulkoista käyttäjähallintaa (AD, LDAP tai jokin muu ratkaisu) ja todentaminen tapahtuu ko. palvelun avulla. Näin ollen IoT-alustaan ei ole tarvetta tallentaa käyttäjän tietoja ollenkaan. Käyttäjien hallinnointi järjestelmissä on aina ylläpito-roolissa (IoT-alustan tai käyttäjähallintapalvelun) olevan henkilön vastuulla.

Taulukko 5.2 *Käyttäjähallinta - tulokset*

Vertailu / Alusta	CityIoT-vaatimus (max. 1)	NIST Digital Identity Guidelines (max. 1)
CityIoT FIWARE	1	1
Thingsboard	1	1
Kaa IoT	1	1
Thinger.io	1	1

5.4 Pääsynhallinta

Muut vertailtavat IoT-alustat, paitsi CityIoT FIWARE -alusta, tarjosivat samanlaisen ratkaisun pääsynhallintaan: roolipohjainen pääsynhallinta (RBAC). Käyttäjät pystyvät määrittelemään omistamaansa dataan käyttöoikeusketjun, joka tarkistetaan jokaiselta käyttäjältä, tai sovellukselta aina kun he pyytävät tietoa alustalta. CityIoT FIWARE -alusta tarjosi yksinkertaisen pääsynhallinnan mikä pohjautuu käyttäjille määriteltyyn tunnisteeseen. Tunnisteen avulla määritellään pääsyoikeudet alustan palveluihin.

Käyttöoikeuksia hallitaan järjestelmien ylläpitäjien toimesta. Ulkoista rajapintaa käytettäessä käyttäjän todentamisen apuna käytetään yksilöivää tunnistetta, joka pitää sisällyttää HTTP-pyyntöön.

Taulukko 5.3 *Pääsynhallinta - tulokset*

Vertailu / Alusta	**CityIoT-vaatimus (max. 2)**	**OWASP: Enforce Access Controls (max. 1)**
CityIoT FIWARE	0	1
Thingsboard	2	1
Kaa IoT	1	1
Thinger.io	1	1

5.5 Pääsynhallinta kaupalliseen tietoon

Etsiessä vastauksia tapaan kaupallistaa IoT-alustan hallinnoimaa tietoa, sen tietoturvalliseen toteuttamiseen ja siihen liittyviin riskeihin, vastaan ei tullut yhtään kunnollista paperia tai artikkelia. Tämä saattaa johtua siitä, että IoT-järjestelmien datan kaupallistaminen on ollut todella haasteellista, vaikka alan kasvua on hypetetty jo monia vuosia. Ongelmallista tässä on se, ettei oikeanlaista käyttötapausta tiedon kaupallistamiseen ole vielä saatu kehitettyä. Kaupungit ja asukkaat kyllä hyötyvät IoT:n tuomista eduista monellakin tapaa, mutta nämä edut ovat käyttökokemusten parantumisessa, resurssien hallinnan helpottamisessa tai pienentyneessä energian kulutuksessa. IoT-alustat eivät kuitenkaan tarvitse kaupallistamisominaisuutta tämänkaltaisten asioiden hoitamiseen, vaan nämä asiat voidaan toteuttaa pelkän datan avulla. [56] [57]

Tämän vaatimuksen alla käsiteltiin myös tietokokoelman omistajuuden siirtämien toiselle käyttäjälle, mikä on mahdollista suorittaa jokaisella alustalla ylläpitäjien toimesta.

Taulukko 5.4 *Pääsynhallinta kaupalliseen tietoon - tulokset*

Vertailu / Alusta	CityIoT-vaatimus (max. 3)
CityIoT FIWARE	1
Thingsboard	1
Kaa IoT	1
Thinger.io	1

5.6 Datan käyttö- ja muokkausoikeudet

Melkein jokaisessa vertailtavassa IoT-alustassa käyttäjien pääsyn- ja oikeuksienhallinta pohjautuu roolipohjaiseen pääsynhallintaa, kuten aikaisemmin on todettu. Poikkeuksena CityIoT FIWARE -alusta, jonka pääsynhallinta pohjautuu käyttäjän yksilöivään tunnisteeseen.

Hälytysten ja ilmoitusten tekeminen poikkeaa tietyiltä osin alustojen välillä, mutta kaikki kuitenkin käyttävät salattua viestintää (TLS) viestien välittämiseen. Näin ollen käyttäjät saavat halutut ilmoitukset muuttumattomina ja voivat luottaa tiedon oikeellisuuteen.

ThingsBoard-, Kaa IoT- ja Thinger.io-alustoissa pakollisena tietona tallennetaan käyttäjän sähköpostiosoite, joka toimii hallintatyökaluun kirjautumistunnisteena. Kaa IoT -alustassa myös käyttäjän etunimi ja sukunimi ovat pakollisia tietoja. Työn kontekstissa (älykkäät kaupungit) sähköpostiosoitteet luetaan henkilötiedoksi, koska ne todennäköisesti tulevat sisältämään liiketoimintayksikköä kuvaavan domain-osan ja näin ollen tietoja tulee käsitellä GDPR-asetuksen mukaisesti. [52]

Taulukko 5.5 *Datan käyttö- ja muokkausoikeudet - tulokset*

Vertailu / Alusta	CityIoT-vaatimus (max. 4)	OWASP: Enforce Access Controls (max. 1)	OWASP: Protect Data Everywhere (max. 1)	Tietosuojavaltuutetun toimisto: Henkilötieto (1)
CityIoT FIWARE	3	1	1	1
Thingsboard	3	1	1	1
Kaa IoT	3	1	1	1
Thinger.io	3	1	1	1

5.7 Ulkoisten tietolähteiden lisääminen alustaan

Työssä tarkasteltiin IoT-alustojen dokumentaatiota sekä tehtiin hands-on-tutkimusta pilviympäristöissä oleviin ilmaisiin kokeiluversioihin. Näiden avulla pyrittiin mahdollisimman tarkasti vastaamaan muutamaan API Security Top 10 2019 kuvattuun käyttötapaukseen. Lisäksi tarkasteltiin myös CityIoT-hankkeen vaatimusta, kuinka toisen IoT-alustan dataa voidaan tuoda alustalle. Vertailussa ei tullut eroja alustojen välillä, kuten taulukosta 5.6 löytyvät tulokset osoittavat.

Taulukko 5.6 *Ulkoisten tietolähteiden lisääminen alustaan - tulokset*

Vertailu / Alusta	CityIoT-vaatimus (max. 1)	OWASP API Security Project (max. 3)
CityIoT FIWARE	1	3
Thingsboard	1	3
Kaa IoT	1	3
Thinger.io	1	3

5.8 Alustaan liitettyjen IoT-laitteiden hallinta

Muut vertailtavat IoT-alustat, lukuunottamatta CityIoT FIWARE -alustaa, tarjosivat roolipohjaisen pääsynhallinnan ja tätä samaa tapaa käytetään myös hyväksi laitteiden hallinnassa: laitteille määritellään käyttäjäroolit, jotka tarkistetaan jokaiselta käyttäjältä tai sovellukselta aina kun ne pyytävät pääsyä laitteen tietoihin tai komentoihin alustalta. CityIoT FIWARE -alustassa käyttäjän yksilöivä tunnisteen avulla määritellään pääsynhallinta alustan laitteisiin.

Laitteille lähetetyt komennot salataan käyttäen viestien salausprotokollaa (TLS). Poikkeuksena Kaa IoT -alusta, joka mahdollistaa komentojen lähettämisen laitteille myös selkokielisenä. Tässä vertailussa ei ollut eri alustojen välillä eroavaisuuksia.

Taulukko 5.7 *Alustaan liitettyjen IoT-laitteiden hallinta - tulokset*

Vertailu / Alusta	CityIoT-vaatimus (max. 2)	OWASP: Enforce Access Controls (max. 1)	OWASP: Protect Data Everywhere (max. 1)
CityIoT FIWARE	2	1	1
Thingsboard	2	1	1
Kaa IoT	2	1	1
Thinger.io	2	1	1

5.9 Pääsynhallinta alustan toiminnallisuuksiin

Ainoastaan Thingsboard-alustasta löytyi toiminnallisuus käyttölokien tarkasteluun suoraan hallintatyökalun avulla. Datan omistaja sekä ylläpitäjät pystyvät seuraamaan alustan tapahtumia käyttäen kyseistä työkalua. Myös Thinger.io-alustalla oli mahdollista tarkastella käyttölokia ylläpitäjän toimesta.

Taulukko 5.8 *Pääsynhallinta alustan toiminnallisuuksiin - tulokset*

Vertailu / Alusta	CityIoT-vaatimus (max. 1)	OWASP: Implement Security Logging and Monitoring (max. 1)
CityIoT FIWARE	0	0
Thingsboard	1	1
Kaa IoT	0	0
Thinger.io	1	1

6 Yhteenveto

Diplomityön tavoitteena oli vertailla erilaisia avoimeen lähdekoodiin pohjautuvia IoT-alustoja ja niiden tietoturvaominaisuuksien soveltuvuutta CityIoT-hankkeeseen. Lisäksi alustojen tietoturvaominaisuuksia tarkasteltiin alalla toimivien tahojen tuottamia ohjeistuksia vasten.

Valitut IoT-alustat olivat CityIoT FIWARE, ThingsBoard, Kaa IoT ja Thinger.io. Ensimmäisenä mainittu alusta toteutettiin hankkeen aikana referenssitoteutuksena. Vertailun lopputuloksena ThingsBoard sai eniten pisteitä. Taulukossa 6.1 on esitetty vertailun tuloksista yhteenveto. Maksimipistemäärä, minkä yksi IoT-alusta olisi voinut saada, jos kaikista vaatimuksista ja ohjeistuksista olisi tullut täydet pisteet, on 31 pistettä. Yksikään alusta ei tähän päässyt ja selitys tähän löytyi CityIoT-hankkeen vaatimuksista. Vaatimusten mukaan alustan pääsynhallinta olisi pitänyt pohjautua dataan määriteltävien attribuuttien mukaan, eli alustan olisi pitänyt toteuttaa ns. attribuuttipohjainen pääsynhallinta (ABAC). Toinen isompi kokonaisuus oli alustan datan kaupallistaminen ja siihen liittyvien seikkojen hallinta. Yhdessäkään alustasta ei tätä toiminnallisuutta löydy suoraan vaan tähän tarkoitukseen pitäisi käyttää jotain olemassa olevaa toteutusta, tai mahdollisuuksien mukaan tehdä itse sellainen.

Taulukko 6.1 *Vertailun tulokset*

Alusta / **Vaatimus**	**Max.**	**CityIoT FIWARE**	**ThingsBoard**	**Kaa IoT**	**Thinger.io**
Sopimusasiat	0	0	0	0	0
Datasettien hallinta	6	3	4	4	4
Käyttäjähallinta	2	2	2	2	2
Pääsyhallinta	3	1	3	2	2
Pääsyhallinta kaupalliseen tietoon	3	1	1	1	1
Datan käyttö- ja muokkausoikeudet	7	6	6	6	6
Ulkoisten tietolähteiden lisääminen alustaan	4	4	4	4	4
Alustaan liitettyjen IoT-laitteiden hallinta	4	4	4	4	4
Pääsyhallinta alustan toiminnallisuuksiin	2	0	1	0	1
Yhteensä	**31**	**22**	**26**	**24**	**25**

ThingsBoard, Kaa IoT ja Thinger.io alustat olivat arkkitehtuurisesti todella samankaltaisia: pilvipalveluita jotka toteuttivat roolipohjaisen pääsynhallinnan sekä monipuolisen REST API -rajapinnan. Tästä syystä pisteetkin olivat lähes identtiset. Kaa IoT -alusta sai vähemmän pisteitä, koska siitä puuttui käyttölokitoiminnallisuus. Työn aikana asiaa kysyttiin alustan tekijöiltä ja he olivat sitä mieltä, että heidän käyttämään Keycloak-järjestelmään käyttölokitoiminnallisuus olisi mahdollista toteuttaa ja tätä väitettä tukivat myös Keycloak-järjestelmän oma dokumentaatio. Toiminnallisuutta ei kuitenkaan lähdetty toteuttamaan työn aikana, jotta kaikkia alustoja olisi kohdeltu tasavertaisesti vertailussa.

ThingsBoard-alusta sai puolestaan enemmän pisteitä, koska siinä on mahdollista kategorisoida dataa näkyvyyden mukaan (avoin/salainen). Tämän ominaisuuden avulla ThingsBoard-alusta sai eniten pisteitä tässä vertailussa.

Vertailtavia IoT-alustoja etsittäessä vastaan ei tullut yhtään avoimeen lähdekoodiin pohjautuvaa täysin toimittajariippumatonta alustaa. Tässäkin työssä käytetyt alusta ovat joko kokeiluversioita maksullisesta versiosta tai niistä löytyy ilmainen rajoitetuin ominaisuuksin käytettävä versio. FIWARE-alusta onkin tässä tapauksessa poikkeuksellinen ja tarjoaa hankkeen vaatimuksiin soveltuvan kokonaisuuden, vaikkakin sijoittui tämän työn vertailussa viimeiseksi. Sijoitukseen vaikutti hyvin paljon alustaan toteutettu pääsynhallinta, joka ei täysin vastannut vaatimuksia ja oli selvästi muita alustoja yksinkertaisempi.

Lähteet

[1] Department of Economic United Nations ja Social Affairs. *World Population Prospects 2019 Highlights*. PDF. Saatavilla: https://population.un.org/wpp/Publications/Files/WPP2019_Highlights.pdf. 2019.

[2] Department of Economic United Nations ja Social Affairs. *World Urbanization Prospects The 2018 Revision*. PDF. Saatavilla: https://population.un.org/wup/Publications/Files/WUP2018-Report.pdf. 2018.

[3] Clean Technol. *A Review of Technical Standards for Smart Cities*. WWW. Saatavilla: https://www.mdpi.com/2571-8797/2/3/19/htm. 2020.

[4] Bismart Business Intelligence Specialist Services. *What Exactly is a Smart City?* WWW. Saatavilla: https://blog.bismart.com/en/what-is-a-smart-city. 2019.

[5] 6Aika. *Älykkäät kaupungit tehdään yhdessä*. WWW. Saatavilla: https://6aika.fi/. 2020.

[6] CityIoT. *CityIoT – ratkaisuja kaupunkien digimurrokseen*. WWW. Saatavilla: https://www.cityiot.fi/. 2019.

[7] Mikko Nurminen. *IoT platforms, data ownership and data access*. Google docs. Saatavilla: https://docs.google.com/document/d/1A4_3X9APc-vZXrApXUKd0tFcUdJ9Lsr3FdjutsrNS4U, versio 0.3. 2019.

[8] Advanced Analytics Division IBM Software Labs Sachchidanand Singh. *Internet of Things(IoT): Security Challenges, Business Opportunities & Reference Architecturefor E-commerce*. WWW. Saatavilla: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7380718. 2015.

[9] a Mobile Virtual Network Operator (MVNO) Nina Pineda consultant of PodM2M. *M2M VS IoT: Know the Difference*. WWW. Saatavilla: https://www.peerbits.com/blog/difference-between-m2m-and-iot.html. 2020.

[10] Philips Lighting Finland Oy. *Philips Hue -tuoteperhe*. WWW. Saatavilla: https://www.philips-hue.com/fi-fi. 2020.

[11] Seppo Kärkkäinen Pekka Koponen Antti Martikainen Hannu Pihala (VTT). *Sähkön pienkuluttajien etäluottavan mittaroinnin tila ja luomat mahdollisuudet*. PDF. Saatavilla: https://www.vttresearch.com/sites/default/files/julkaisut/muut/2006/VTT-R-09048-06.pdf. 2006.

[12] Etteplan MORE Oy. *Internet of Things -pilviratkaisu ja Web-mobiilisovellus sähköautojen lataukseen*. WWW. Saatavilla: https://www.etteplanmore.com/referenssit/unified-chargers. 2020.

[13] Joonas Hämäläinen ja Timo Kantanen. *Saunan etäkäyttö mobiililaitteella*. PDF. Saatavilla: https://www.theseus.fi/bitstream/handle/10024/101448/Saunan_et.pdf?sequence=1. 2015.

[14] Digita Oy. *IoT mahdollistaa älykkäät ratkaisut teollisuuden pulmiin: "Teknologia on kehittynyt ajan saatossa ja tarjoaa nyt uusia mahdollisuuksia"*. WWW. Saatavilla: https://www.digita.fi/asiakastarinat/iot-mahdollistaa-alykkaat-ratkaisut-teollisuuden-pulmiin-teknologia-on-kehittynyt-ajan-saatossa-ja-tarjoaa-nyt-uusia-mahdollisuuksia/. 2020.

[15] Research Analyst at BI Intelligence Peter Newman. *THE INTERNET OF THINGS 2020: Here's what over 400 IoT decision-makers say about the future of enterprise connectivity and how IoT companies can use it to grow revenue*. WWW. Saatavilla: https://www.businessinsider.com/internet-of-things-report?r=US&IR=T. 2020.

[16] Rahul Gupta (IBM) Andrew Banks (IBM). *MQTT Version 3.1.1, OASIS Standard*. WWW. Saatavilla: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. 2014.

[17] The HiveMQ Team. *Publish & Subscribe - MQTT Essentials: Part 2*. WWW. Saatavilla: https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe/. 2015.

[18] J Mogul R Fielding J Gettys. *Hypertext Transfer Protocol HTTP/1,1*. WWW. Saatavilla: https://tools.ietf.org/html/rfc2616. 1999.

[19] Kamaljit Kaur Jaideep Kaur. *Internet of Things: A Review on Technologies, Architecture, Challenges, Applications, Future Trends*. PDF. Saatavilla: http://j.mecs-press.net/ijcnis/ijcnis-v9-n4/IJCNIS-V9-N4-7.pdf. 2017.

[20] Xiaofei Xu Eric Ke Wang Yunming Ye. *Security Issues and Challenges for Cyber Physical System*. WWW. Saatavilla: https://ieeexplore.ieee.org/document/5724910. 2010.

[21] Sanah Abdullahi Muaz Adamu I Abubakar Haruna Chiroma. *A Review of the Advances in Cyber Security Benchmark Datasets for Evaluating Data-Driven Based Intrusion Detection Systems*. WWW. Saatavilla: https://www.sciencedirect.com/science/article/pii/S1877050915025788. 2015.

[22] Mada Abdulrahman Aishah Abdullah Reem Hamad. *CyberSecurity: A Review of Internet of Things (IoT) Security Issues, Challenges and Techniques*. WWW. Saatavilla: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8769560. 2019.

[23] Jiafu Wan Qi Jing Athanasios V Vasilakos. *Security of the Internet of Things: perspectives and challenges*. PDF. Saatavilla: https://csi.dgist.ac.kr/uploads/Seminar/1407_IoT_SSH.pdf. 2014.

[24] Deloitte Development LLC. *Making smart cities cybersecure*. PDF. Saatavilla: https://www2.deloitte.com/content/dam/Deloitte/de/Documents/risk/Report_making_smart_cities_cyber_secure.pdf. 2019.

[25] AVSystem. *What is an Internet of Things Platform?* WWW. Saatavilla: https://www.avsystem.com/blog/what-is-internet-of-things-platform/. 2019.

[26] i-SCOOP. *IoT platforms – IoT platform definitions, capabilities, selection advice and market*. WWW. Saatavilla: https://www.i-scoop.eu/internet-of-things-guide/iot-platform-market-2017-2025/. 2020.

[27] IoT Analytics Padraig Scully. *5 things to know about the IoT Platform ecosystem*. WWW. Saatavilla: https://iot-analytics.com/5-things-know-about-iot-platform/. 2016.

[28] ThingsBoard. *ThingsBoard - Open-source IoT Platform*. WWW. Saatavilla: https://thingsboard.io/. 2020.

[29] ThingsBoard. *ThingsBoard architecture*. WWW. Saatavilla: https://thingsboard.io/docs/reference/. 2020.

[30] Kaa. *Enterprise IoT Platform with Free Plan | Kaa*. WWW. Saatavilla: https://www.kaaproject.org/. 2020.

[31] Kaa. *Architecture overview*. WWW. Saatavilla: https://docs.kaaiot.io/KAA/docs/current/Architecture-overview/. 2020.

[32] Kaa. *KPC*. WWW. Saatavilla: https://docs.kaaiot.io/KAA/docs/v1.2.0/Features/Data-collection/DCX/. 2020.

[33] Kaa. *CM*. WWW. Saatavilla: https://docs.kaaiot.io/KAA/docs/current/Features/Device-management/CM/. 2020.

[34] Kaa. *DCX*. WWW. Saatavilla: https://docs.kaaiot.io/KAA/docs/v1.2.0/Features/Data-collection/DCX/. 2020.

[35] Kaa. *CMX*. WWW. Saatavilla: https://docs.kaaiot.io/KAA/docs/current/Features/Configuration-management/CMX/. 2020.

[36] Kaa. *EPR*. WWW. Saatavilla: https://docs.kaaiot.io/KAA/docs/current/Features/Device-management/EPR/. 2020.

[37] Kaa. *WD*. WWW. Saatavilla: https://docs.kaaiot.io/KAA/docs/current/Features/Visualization/WD/. 2020.

[38] Thinger.io. *Thinger.io - Open Source IoT Platform*. WWW. Saatavilla: https://thinger.io/. 2020.

[39] Thinger.io. *Thinger.io - Overview*. WWW. Saatavilla: https://docs.thinger.io/. 2020.

[40] 6Aika. *CityIoT – Tulevaisuuden toimijariippumaton dataintegraatioalusta*. WWW. Saatavilla: https://6aika.fi/project/cityiot/. 2020.

[41] 6Aika. *6Aika: Tulevaisuuden toimijariippumaton dataintegraatioalusta (CityIoT)*. Google docs. Saatavilla: https://drive.google.com/file/d/1B48QHkRFOgvvJdbGV5OyDh-CJU_vK4dP/view. 2020.

[42] CityIoT. *Report on FIWARE Platform*. Google docs. Saatavilla: https://drive.google.com/file/d/1yueGrdArlFmz8ZzchTXWuhbgC9dKUuGN/view. 2020.

[43] FIWARE. *The Open Source platform for our smart digital future - FIWARE*. WWW. Saatavilla: https://github.com/FIWARE/catalogue. 2020.

[44] FIWARE. *What is FIWARE?* WWW. Saatavilla: https://www.fiware.org/developers/. 2020.

[45] CityIoT. *Tampere CityIoT FIWARE platform*. Google docs. Saatavilla: https://drive.google.com/file/d/1yueGrdArlFmz8ZzchTXWuhbgC9dKUuGN. 2020.

[46] The OWASP® Foundation. *OWASP Top Ten Proactive Controls 2018*. WWW. Saatavilla: https://owasp.org/www-project-proactive-controls/v3/en/0x02-about-project.html. 2018.

[47] The OWASP® Foundation. *C7: Enforce Access Controls*. WWW. Saatavilla: https://owasp.org/www-project-proactive-controls/v3/en/c7-enforce-access-controls.html. 2018.

[48] The OWASP® Foundation. *C8: Protect Data Everywhere*. WWW. Saatavilla: https://owasp.org/www-project-proactive-controls/v3/en/c8-protect-data-everywhere. 2018.

[49] The OWASP® Foundation. *C9: Implement Security Logging and Monitoring*. WWW. Saatavilla: https://owasp.org/www-project-proactive-controls/v3/en/c9-security-logging. 2018.

[50] NIST. *NIST Special Publication 800-63B - Digital Identity Guidelines*. WWW. Saatavilla: https://pages.nist.gov/800-63-3/sp800-63b.html. 2020.

[51] Tietosuojavaltuutetun toimisto. *Tietosuojavaltuutetun toimisto*. WWW. Saatavilla: https://tietosuoja.fi/etusivu. 2020.

[52] Tietosuojavaltuutetun toimisto. *Mikä on henkilötieto?* WWW. Saatavilla: https://tietosuoja.fi/mika-on-henkilotieto. 2020.

[53] The OWASP® Foundation. *OWASP API Security Project*. WWW. Saatavilla: https://owasp.org/www-project-api-security/. 2019.

[54] FIWARE. *Data Publication and Monetization*. WWW. Saatavilla: https://fiwaretourguide.readthedocs.io/en/latest/data-publication/introduction/. 2020.

[55] Lisa Seacat DeLuca. *Digital Twin: Transforming Asset Operations*. WWW. Saatavilla: https://reliabilityweb.com/articles/entry/digital-twin-transforming-asset-operations. 2020.

[56] SmartCitiesWorld Sarah Wray. *Copenhagen shares takeaways from its City Data Exchange*. WWW. Saatavilla: https://www.smartcitiesworld.net/news/news/copenhagen-shares-takeaways-from-its-city-data-exchange-2961. 2018.

[57] Forbes Jean Lawrence. *Smart Cities Improve Resident Experience, Monetize Digital Services*. WWW. Saatavilla: https://www.forbes.com/sites/oracle/2019/04/26/smart-cities-improve-resident-experience-monetize-digital-services/?sh=57c60c686a00. 2019.

A CityIoT-hankkeen tietoturvavaatimukset

1. Sopimusasiat

Sopimustiedon lisäämisestä alustalle

Datan omistajan tulee tehdä sopimus alustan omistajan kanssa. Sopimuksessa on sovittava datan käyttöoikeussäännöistä, kenellä on pääsyoikeus dataan, millä oikeuksilla dataan pääsee käsiksi sekä tulee määritellä datan avoimuuden taso.

2. Datasettien hallinta

Datasettien luonti

Datan omistaja voi luoda uutta dataa alustan tarjoamilla työkaluilla. Datasettiin voidaan asettaan käyttöoikeudet ja metatietoa (samoin kuin datan omistaja). Joissakin IoT-alustoissa datasetin luonti tapahtuu automaattisesti, eli kun dataa lisätään järjestelmään ensimmäisen kerran luodaan myös datasetti.

Datan lisääminen datasettiin

Datan omistaja voi lisätä dataa datasettiin. Lisätty data voidaan lisätä eräajolla tai jatkuvasti, esimerkiksi IoT-laitteesta tulevasta datavirrasta. Lisättyyn dataa pääsee käsiksi datasettiin asetettujen käyttöoikeuksien mukaan. Jos käyttäjällä ei ole dataan käyttöoikeuksia, ilmoitetaan tästä käyttäjälle virheilmoituksella. Jos käyttöoikeuksia ei ole, data ei tallennu alustaan.

Datasetin muokkaaminen tai poisto

Datasetin omistaja voi halutessaan muokata tai poistaa datasetin sisältämää dataa. Datasetin poiston yhteydessä myös sen käyttöoikeudet poistuvat alustalta.

Metatiedon lisääminen alustassa olevaan dataan

IoT-alustan ylläpitäjä määrittelee datalle sen omistajan, lisenssoinnin ja mahdolliset kaupalliset ehdot metatietoina. Metatietoina voidaan asettaa myös IoT-laitteiden lisäysoikeudet. Metatiedon tulisi olla mahdollisimman joustava, jolloin omistajuuden ja käyttöoikeuksien määritteleminen erillisille dataseteille ja tietolähteille olisi mahdollista.

3. Käyttäjähallinta

Alustan käyttäjien hallinta

Järjestelmän ylläpitäjä luo käyttäjät ja organisaatiot alustaan. Jokaiselle käyttäjälle on annettu tarvittavat valtuudet, joita käytetään käyttäjien identifiointiin silloin kun he käyttävät alustan toimintoja.

4. Pääsynhallinta

Käyttöoikeuksien ja -tasojen määrittely datasettiin tai muuhun dataan

Datan omistaja asettaa datalle käyttöoikeudet. Käyttöoikeudet voivat pitää sisällään minkä tahansa osajoukon 'luoda-lukea-päivittää-poistaa'-toiminnoista (CRUD) sekä mahdolliset vaihtoehdot mitkä on valittu. Käyttöoikeudet voidaan asettaa tietyille käyttäjille tiettyyn datasettiin. Käyttöoikeudet tarkistetaan aina kun käyttäjä yrittää pääsyä alustan dataan.

Datasetin käyttöoikeuksien muokkaaminen tai datan muokkaus

Datan omistaja voi määritellä käyttöoikeudet järjestelmään lisäämilleen datalle. He voivat muuttaa datan näkyvyyttä salaisesta julkiseksi, tai voivat muokata käyttäjän käyttöoikeuksia CRUD-toimintoihin.

5. Pääsynhallinta kaupalliseen tietoon

Kaupallisten oikeuksien määrittely tietokokoelman käyttöön

Datan omistaja määrittelee käytössä olevilla alustan työkaluilla datalle haluamansa tasot, mitkä halutaan antaa maksaville käyttäjille. Datan omistaja määrittelee myös muita ehtoja joita sovelletaan mm. hinnoittelumalli (laskutetaanko käyttäjää pyyntö-perusteisesti, myydäänkö heille mahdollisuus datan käytölle tietylle aikavälille). Nämä kaupalliset säännöt talletetaan alustalle, erilleen datasta (tämän tiedon lisääminen metatietoon tekee datan tallentamisesta ja muista toiminnallisuuksista tarpeettoman monimutkaista). Säännöstö ja osto-oikeudet tarkistetaan ennen kuin käyttäjälle myönnetään pääsy dataan. Kaupalliset säännöt voidaan määritellä erilaisina datasetteinä, jotkut datasetit voivat sisältää samat säännöt. Tietorekisterin tulee tarvittaessa näyttää nämä kaupalliset säännöt muiden tietojen kanssa.

Tietokokoelman käyttöoikeuksien myynti alustalta kolmansille osapuolille

Datan käyttäjä löytää alustalta saatavilla olevan datasetin tietorekisteristä, johon on liitettynä kaupalliset säännöt. Tämän jälkeen käyttäjä ostaa pääsyn dataan alustan

työkaluilla. Kun tapahtuma on viety loppuun, käyttäjän käyttöoikeudet ehtoineen tallennetaan alustalle, erikseen datasta. Nämä käyttöoikeudet tarkistetaan jokaiselta käyttäjältä, jotka yrittävät käyttää dataa.

Tietokokoelman omistuksen siirtäminen tai muokkaus

Datan omistaja voi siirtää datan omistajuuden kenelle tahansa. Datan omistaja tiedottaa muutoksista IoT-alustan ylläpitäjälle, joka suorittaa muutokset datan omistajuuteen järjestelmässä.

6. Datan käyttö- ja muokkausoikeudet

Tiedon ja tietolähteiden saatavuuden tarkastelu

Käyttäjä käyttää tietorekisteriä nähdäkseen mitä datasettejä on saatavilla alustassa ja mitä ehtoja, kuten mitä pääsyrajoituksia liittyy näihin datasetteihin. Käyttäjä voi myös nähdä tietorekisteristä, kuinka he voivat päästä käsiksi dataan. Käyttäjä voi lähettää viestin datasetin yhteyshenkilölle saadakseen lisätietoja datasetistä.

Käyttäjä voi käsitellä ja muokata tietoa alustassa

Käyttäjä hankkii yksilöivät valtuudet alustaan. Näillä valtuuksilla käyttäjä tekee pyyntöjä alustasta. Alusta tarkistaa valtuudet sekä onko todennetulla käyttäjällä käyttöoikeudet pyydettyyn dataan. Jos käyttäjällä on tarvittavat käyttöoikeudet, alusta palauttaa pyydetyn datan. Jos käyttäjällä ei ole tarvittavia käyttöoikeuksia, kerrotaan siitä käyttäjälle paluuviestillä sekä opastetaan kuinka tarvittavat käyttöoikeudet voidaan hakea. Aina kun järjestelmän käyttäjä suorittaa CRUD-toiminnan, hänen käyttöoikeudet tarkistetaan. Jos käyttäjä ei ole ostanut tarvittavia käyttöoikeuksia pyytämäänsä dataan, tulee paluuviestinä ohjeet siitä, kuinka käyttöoikeudet voidaan lunastaa.

Ilmoitusten vastaanottamisen tilaustiedon muuttuessa alustassa on mahdollista

IoT-alustan sisällön tilaajat saavat ilmoituksen datan muuttuessa. Ilmoitus voi tulla esimerkiksi sellaisessa tilanteessa, kun IoT-laite lähettää uuden mittaustuloksen IoT-alustaan. Näin ollen käyttäjän ei tarvitse tehdä jatkuvasti pyyntöjä alustalle pysyäkseen ajan tasalla.

Käyttäjät voivat nähdä mitä tietoa heistä on tallennettuna alustaan

Henkilö voi kysyä mitä tietoa heistä on tallennettuna alustalle, GDPR-asetuksen mukaan Euroopan Unionissa. Järjestelmän ylläpitäjä yksilöi toimet ja toimittaa

kaiken löytämänsä tiedon pyytäjälle. Käyttäjää ei voida tunnistaa, jos tietoja käyttäjästä ei ole saatavilla.

7. Ulkoisten tietolähteiden lisääminen alustaan

Alustan liittäminen toiseen IoT-alustaan

Tietojen lisääminen toiselta IoT-alustalta voi aiheuttaa vaikeuksia alustojen tietomallien ja mahdollisten teknisten rajoitusten yhteensovittamisesta. Ensimmäiseksi pitää tehdä sopimus käyttäjärooleille. Tämän jälkeen uusi datasetti luodaan tulevasta datasta. Käyttöoikeudet ja kaupalliset säännökset asetetaan tämän jälkeen.

8. Alustaan liitettyjen IoT-laitteiden hallinta

IoT-laitteiden lisääminen alustaan

Yksilöidyt IoT-laitteet, tai ryhmä IoT-laitteita voidaan lisätä omiin datasetteihin datan omistajan toimesta, jos alustan ylläpitäjä on antanut tähän luvan. Laitteita lisätään keräämään niistä saatua dataa IoT-alustaan. Uusien IoT-laitteiden lisäämisoikeudet on tallennettu alustaan ja ne tarkistetaan ennen laitteen lisäämistä. Jos käyttäjällä ei ole lupaa lisätä IoT-laitteita, toimintoa ei tehdä ja käyttäjälle annetaan tästä ilmoitus.

IoT-laitteiden hallinta alustan avulla

Yksilöidyt IoT-laitteet, tai ryhmä IoT-laitteita ovat hallittavissa datan omistajan toimesta järjestelmästä löytyvillä toiminnoilla, jos alustan ylläpitäjä on antanut tähän luvan. Toiminta, joka lähetetään alustalta IoT-laitteelle, voi olla esimerkiksi uuden mittaustuloksen kysely, uuden arvon asettaminen laitteelle, laitteen kalibrointi tai laitteen uudelleen käynnistäminen. IoT-alustalla voi olla tieto siitä mitä komentoja käyttäjä voi lähettää laitteille. Tämä kuitenkin vaatisi tuntemusta laitteista ja niiden komennoista, ja olisi siksi erittäin työlästä tehdä ylläpitäjien toimesta. Tällainen toiminta jätetään datan omistajan tehtäväksi, koska heillä on paras tietämys omista järjestelmistään.

9. Pääsynhallinta alustan toiminnallisuuksiin

Pääsy alustan lokitietoihin

Käyttölokit tulisi luoda jokaiselle datasetille. Lokit tulee olla saatavilla järjestelmän ylläpitäjälle sekä datan omistajalle. Lokit pitävät sisällään tietoa kuka käyttäjä on käyttänyt järjestelmää, mitä dataa he ovat käyttäneet sekä mitä muutoksia dataan

on tehty. Lokitietoihin pääsyä hallitaan käyttöoikeuksilla, vaikkakin yleensä tämän tapaista tietoa näytetään datan omistajille 'kojelauta' näkymien avulla.