



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Bayram Aslan

# Avoimen lähdekoodin IoT-alustat

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

1.4.2020

Tekijä Otsikko	Bayram Aslan Avoimen lähdekoodin IoT-alustat
Sivumäärä Aika	29 sivua + 1 liite 1.4.2020
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Tietoverkot ja sovellukset
Ohjaajat	Lehtori Marko Uusitalo
<p>Tämän insinööriyön tavoitteena oli tutkia esineiden internettiä, IoT-alustojen osa-alueita ja markkinoilla olevia avoimen lähdekoodin IoT-alustoja. Lisäksi tarkoitus oli tehdä IoT-demo, joka lähettää huoneen lämpötila- ja kosteusarvoja työssä tutkittuun avoimen lähdekoodin IoT-alustalle.</p> <p>Insinööriyössä selvitettiin esineiden internetiin liittyviä alueita, kuten IoT-alustojen rakenteita ja avoimen lähdekoodin perusteella toteutettuja IoT-alustojen ominaisuuksia. Työssä tutustuttiin avoimen lähdekoodin IoT-alustojen ominaisuuksiin käytännöllisesti ottamalla niitä käyttöön pilvipohjaisesti. Tutkimuksessa selvitettiin, mitä avoimen lähdekoodin IoT-alustat tarjoavat ja mihin käyttöön ne ovat ensisijaisesti tarkoitettu. Työssä tutkittiin, mitä yhteysprotokollia ja kirjastoja avoimen lähdekoodin IoT-alustat tukevat.</p> <p>Insinööriyön tavoitteena oleva IoT-demo toteutui onnistuneesti. Huoneen lämpötila- ja kosteusarvot saatiin mitattua ja mittatulokset lähetettiin Thingsboard-nimiseen avoimen lähdekoodin IoT-alustalle, jossa arvot näytettiin kojelaudassa visualisoituna IoT-alustan omilla työkaluilla. Demoa varten käytetty Raspberry Pi 3 pieni tietokone, jossa pyörii Python-koodi ja anturina SenseHat-niminen Raspberry Pi:hin liitettävä laajennuskortti. Laajennuskortista löytyi moneen käyttöön antureita.</p>	
Avainsanat	Esineiden internet, IoT, avoin lähdekoodi, MQTT, alusta

Author Title	Bayram Aslan Open-source IoT Platforms
Number of Pages Date	29 pages + 1 appendix 01.04.2020
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Communication Networks and Applications
Instructors	Marko Uusitalo, Senior Lecturer
<p>The objective of the graduate study was to get acquainted with the Internet of Things, the components of IoT platforms and the open-source IoT platforms on the market. In addition, an IoT demo that sends room temperature and humidity values to one of the open-source IoT platforms studied here was completed.</p> <p>In the thesis, the areas related to the Internet of Things, the structures of IoT platforms and the properties of IoT platforms implemented based on open source are introduced. In addition, the features of open source IoT platforms are introduced in practice in a cloud-based manner. The study explores what open source IoT platforms offer and what they are primarily intended for. In addition, the study discusses which connection protocol and libraries are supported by open source IoT platforms.</p> <p>The goal of the study, the IoT demo, was successfully implemented. Room temperature and humidity values were measured and sent to an open source IoT platform called Thingsboard, where the values were displayed on the dashboard as visualized by the embedded tools of the IoT platform. For this demo, a Raspberry Pi 3 small computer where the code is running and an expansion card called SenseHat to be connected to the Raspberry Pi were used. The expansion card used had sensors for various uses.</p>	
Keywords	Internet of Things. IoT, open source, MQTT, platform

## Sisällys

### Lyhenteet

1	Johdanto	1
2	Esineiden internet	2
2.1	Esineiden internetin sovellusalueet	3
2.2	Esimerkkejä IoT-tekniikalla valmistetuista tuotteista	4
3	Esineiden internetin arkkitehtuuri	5
4	Mikä on IoT-alusta?	9
5	Avoin lähdekoodi	13
6	Avoimen lähdekoodin IoT-alustat	15
6.1	Thinger	15
6.2	Freeboard	18
6.3	Thingsboard	19
6.4	DeviceHive	22
7	Lämpötila- ja kosteusarvojen visualisointi IoT-alustassa	24
8	Yhteenveto	27
	Lähteet	28
	Liitteet	
	Liite 1. Lämpötila- ja kosteusarvojen lähetys	

## Lyhenteet

IoT	Internet of Things. Esineiden internet, verkkoon kytkettyjä esineitä.
M2M	Machine-to-machine. Erilaisten järjestelmien, koneiden, laitteiden ja niihin yhdistettyjen sensorien välistä älykästä kommunikointia.
LoRa	Long Range. Yksittäinen modulaatoratkaisu, jota päätelaitteet ja reitittimet käyttävät kommunikoidessaan keskenään
MQTT	Message Queuing Telemetry Transport. Avoimeen lähdekoodiin perustuva erittäin kevyt viestiprotokolla
CoAP	Constrained Application Protocol, kevyt viestintäprotokolla, joka on suunniteltu rajoitettuja laitteita varten.
JSON	JavaScript Object Notation. Yksinkertainen avoimen standardin tiedostomuoto tiedonvälitykseen
RPC	Remote Procedure Call. Tietoliikenneprotokolla korkean tason käyttöjärjestelmätoimintoihin.
HTTPS	Hypertext Transport Protocol Secure. Siirtoprotokolla SSL/TLS -salauksella.
REST	Representational State Transfer. HTTP-protokollaan perustuva arkkitehtuurimalli ohjelmointirajapintojen toteuttamiseen.
CURL	Client Uniform Resource Locator. Komentorivityökalu, jota käytetään tietojen siirtämiseen palvelimelta toiseen.

## 1 Johdanto

Internetin käyttö on yleistynyt vauhdilla. Internet on tärkeä osa ihmisten elämässä. Internetin tärkeys on korostunut maailmanlaajuisesti ja jopa päivä ilman internetiä voi aiheuttaa nykypäivänä erittäin suuria vahinkoja. Esineiden internet kasvattaa maailmanlaajuisen internetriippuvuuden entistä enemmän. Esineiden internet -termin, englanniksi lausuttuna Internet of Things (IoT), ensimmäinen käyttö ulottuu Kevin Ashtonin esitykseen vuonna 1999. [1.] Esineiden internet termi eli IoT lyhyesti tarkoittaa älylaitteiden viestintää. Esineiden internetillä voi esimerkiksi ohjata rannekellosta monet muut elektroniikkalaitteet kommunikoimalla keskenään. Tämä teknologia mahdollistaa etänä ihmisten omien laitteiden ohjaamista missä tahansa maapalloa. Miljardien laitteiden ja anturien internetiin yhdistämisen tarve kasvattaa paljon esineiden internetin merkityksen teollisuus- ja kuluttajamaailmassa. IoT-teknologiat mahdollistavat käyttäjän laitteiden vianetsintää ilman, että tarvitsisi yritysten ottaa yhteydenottoa asiakkaille, joten tämän kaltaiset palvelut hoituvat ilman asiakashäiriötä. Esineiden internet parantaa samalla yrityksen asiakaspalvelun laatua ja toimintaa.

Nykyään esineiden internet ulottuu pienistä laitteista älykkäisiin kaupunkeihin. Esineiden internetissä käytetyn tiedon ymmärtäminen ja erilaisten tietojen analysointi on erittäin arvokasta. Jokainen on joskus etsinyt pysäköintitilaa pysäköintialueelta. Nykypäivänä pysäköintialueilla käytettävät ilmoitustaulut sisäänkäyntien kohdalla, jossa ilmoitetaan pysäköintitilan käyttöasteesta, ovat pienimpiä esimerkkejä esineiden internetistä. Täällä sijaitsevat anturit mittaavat pysäköintialueen tilan ja käyttöasteen. Kun käyttöaste analysoidaan kellonajan perusteella, syntyy dataa, jonka avulla voimme analysoida, milloin parkkipaikka on täynnä tai siellä on tyhjää.

IoT-laitteiden hallinnointia, reaaliaikaista katselua, seuranta, konfiguraatiota varten on olemassa IoT-alustoja. On monia yrityksiä, jotka ovat tulleet markkinoille tällä nuorella teknologialla omilla IoT-alustoilla. Moni niistä toimii avoimen lähdekoodin periaatteella, mikä tarkoittaa että jokainen saa käyttää ja tutustua vapaasti näihin ilman maksua. Avoimen lähdekoodin periaatteella oleva IoT-alustan lähdekoodeihin pääsy on sallittu ja mahdollistaa ohjelmistokehittäjiä tai käyttäjiä osallistumaan IoT-alustan kehittämiseen tai muuttamiseen.

Opinnäytetyön tavoitteena on tutustua avoimilla lähdekoodilla toimiviin IoT-alustoihin. Tarkoituksena on tutkia esineiden internetin rakennetta, IoT-alustan osa-alueita ja avoimen lähdekoodin periaatteella olevien IoT-alustojen ominaisuuksia, teknologioita sekä arkkitehtuureja. Lisäksi tutkitaan käyttämällä IoT-alustojen joitakin perustoimintoja ja tehdään tästä demo, josta saadaan visualisoitu huoneen lämpötila- ja kosteusarvoja. Demo tehdään tämän insinööriyön tutkimasta avoimen lähdekoodin IoT-alusta.

## 2 Esineiden internet

Esineiden internetin (engl. IoT, Internet of Things) tekniikat mahdollistavat kaikki fyysisten laitteiden kytkemisen internetverkkoon, kuten kuvassa 1 näkyy. Yksittäinen sensori, kuten lämpömittari tai jopa ajoneuvo voi olla internetiin kytkettävä esine. Näistä esineistä luetaan tietoa tai niitä voidaan ohjata internetin avulla.

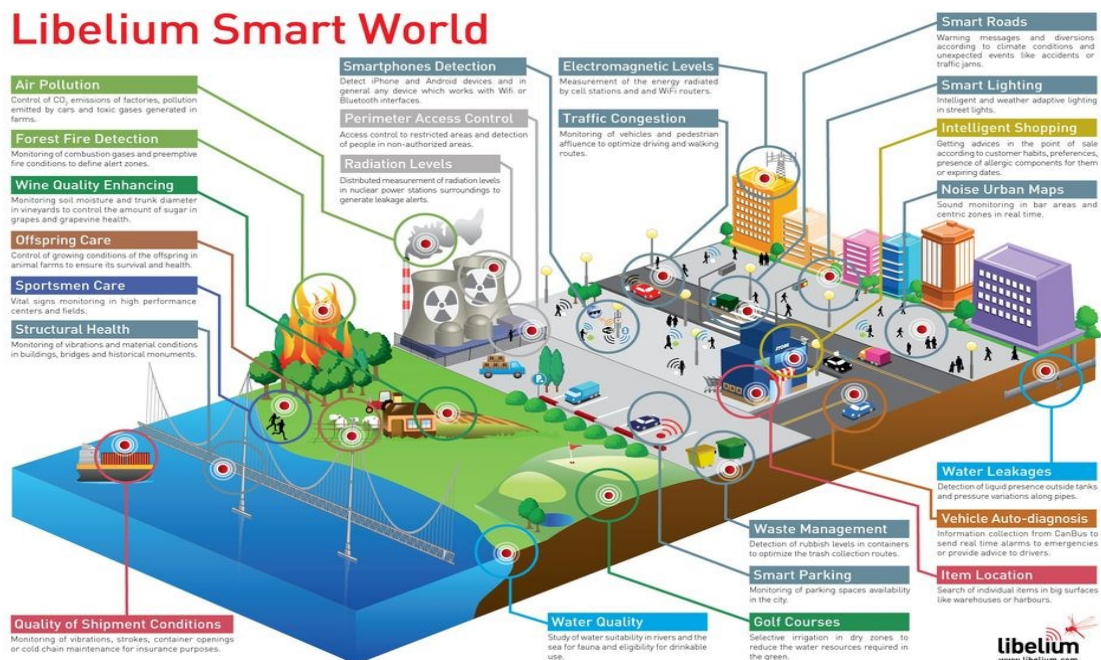


Kuva 1. Esineiden internettiin yhdistetyt esineet. [1.]

Esineiden internet on nuori termi varsinkin teollisuudessa. Kuitenkin teollisuudessa on samankaltaisia ratkaisuja hyödynnetty jo vuosikymmeniä. Termi machine-to-machine (M2M) on ollut teollisuusalalla käytössä jo ennen esineiden internetiä. Kiinteistöjen sähkömittareiden etänä lukeminen on yksi tunnetuista machine-to-machine (M2M) -ratkaisuista. Erona machine-to-machine (M2M) -tekniikasta esineiden internet on paljon monimutkaisempi ja laajempi tekniikka. Esineiden internet mahdollistaa kaikkien ympäristössämme olevien objektien kommunikaation keskenään internetin avulla, mutta machine-to-machinen kohdalla laitteiden kommunikaatio ei välttämättä edellytä internetiä. [2; 3; 4.]

## 2.1 Esineiden internetin sovellusalueet

IPv6-protokolla on ratkaisu internetosoitteiden räjähdysmäisen kasvuun. Tämä mahdollistaa erilaisten ja määrällisesti suurempien laitteiden kytkemisen internetiin. Esineiden internetiä hyödynnetään monilla alueilla. Teollisuusalueella esineiden internetiä hyödynnetään valmistusprosessien valvonnassa ja ohjauksessa, kuten kuvassa 2 esitellään. [2.]



Kuva 2. Esineiden internetin sovellusalueet. [2.]

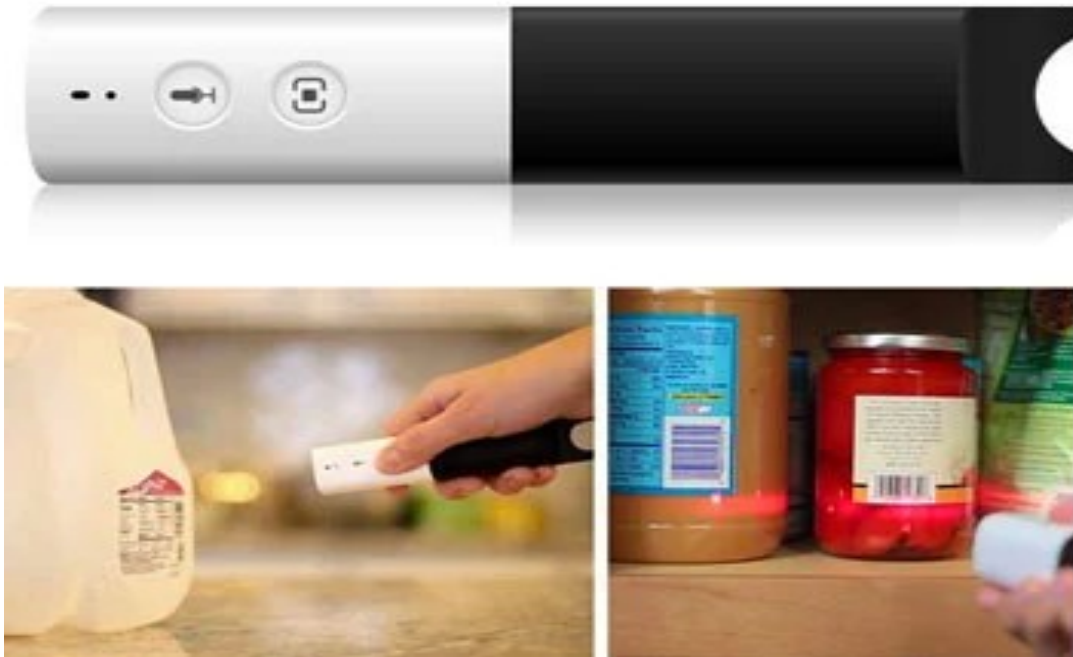


Esineiden internetiä hyödynnetään monella osa-alueella. Terveystieteiden mahdollistaa mm. älyranneke- ja sensorilaitteilla terveystietojen seurantaan. Logistiikkasektorilla esineiden internet on nykyään hyvin tärkeässä asemassa, ja sen merkitys on lisääntynyt jatkuvasti lisää. Liikenneinfrastruktuurissa ja liikennejärjestelmissä havainnointi- ja ohjauslaitteistot hoitavat valvonnan ja seurannan. Verkko- ja mobiililaitteiden avulla nämä tiedot ohjataan kansalaisten käyttöön. Esineiden internet logistiikkasektorilla näkyy esimerkiksi myös paikannusjärjestelmänä, missä käytössä olevien kaikkien kulkuneuvojen saapumisaikoja voidaan ennustella. Lämpötilan seuranta lastitiloissa on myös mahdollista. Lämpötilamuutoksista raja-arvoistaan voi esimerkiksi kuljettaja saada ilmoituksen. Jätehuollon logistiikassa IoT-tekniikkaa hyödynnetään esimerkiksi jäteastioiden tyhjennystarpeen seurannassa. Tämä onnistuu jäteastioihin asennetuilla antureilla. Näin ei tapahdu turhaa tyhjennyskäyntiä, koska anturien avulla voidaan tyhjennyspäivä ennustaa.

## 2.2 Esimerkkejä IoT-tekniikalla valmistetuista tuotteista

Internetsivu [www.postscapes.com](http://www.postscapes.com) on luetellut parhaat projektit, joissa on käytetty IoT-tuotteita:

- Amazon Dash: Käyttäjä voi lukea loppumassa olevien tuotteiden viivakoodit ja tuotteet siirtyvät suoraan ostoskoriin valmiiksi. Amazon Dash näkyy kuvassa 3.
- Älykäs bluetooth-paikannin: Tällä paikantimella voi seurata älypuhelimella omia tavaroita, kuten laukkuja ja avaimia.
- Älykkäät kodinkoneet: Talon lämmitys- ja viilennysjärjestelmät kommunikoivat zigbee-protokollan kautta muiden kodinkoneiden kanssa ja optimoivat kodin lämpötilan keräämällä hetkellisiä säätietoja. Halutessaan käyttäjä voi hallinnoida jokaisen tuotteen älypuhelimella.



Kuva 3. Amazon Dash luo ostoslistan helposti lukemalla tuotteiden viivakoodit. [3, s. 21.]

- Älykäs ovilukko: Tämä ovilukko sisältää läheisyys anturin ja on käytettävissä älypuhelimien avulla. Käyttäjä voi lukita tai avata oven etänä langottomalla yhteydellä, vaikka wifillä.
- Käyttäjän älypuhelimien läheisyysanturilla talon valot kytkeytyvät päälle automaattisesti.
- Älypistorasioilla lapset suojautuvat sähköiskuilta. [3, s. 21.]

### 3 Esineiden internetin arkkitehtuuri

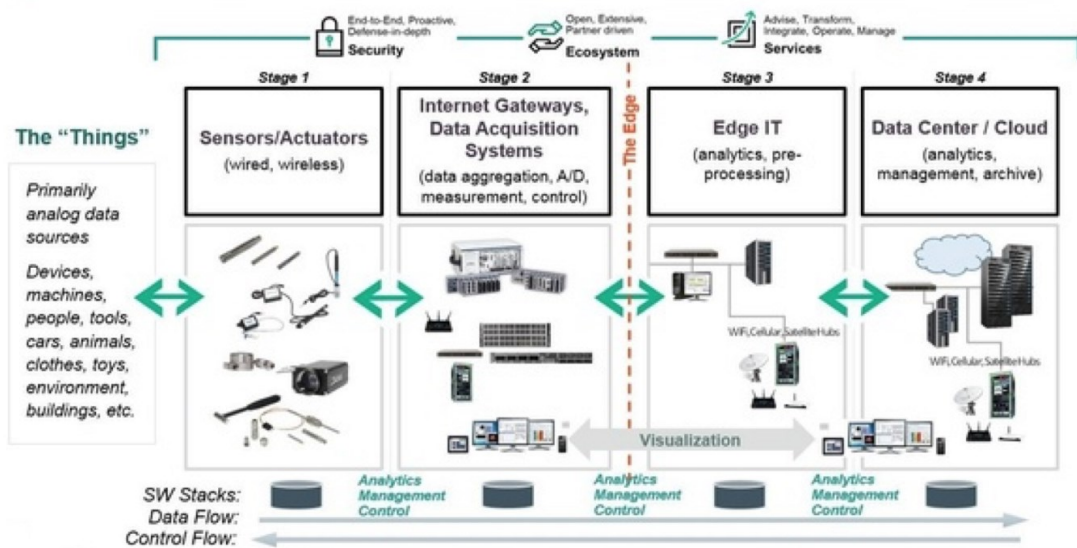
Esineiden internetin arkkitehtuurista ei ole yhtä sovittua universaalia yksimielisyyttä. Eri tutkijat ovat ehdottaneet erilaisia arkkitehtuureja. [5.] IoT-arkkitehtuuri on moniosainen järjestelmä. Nämä ovat anturit, protokollat, toimilaitteet, pilvipalvelut ja kerrokset. IoT-arkkitehtuurissa löytyy siis neljä eri vaihetta. Näistä vaiheista ei kuitenkaan ole yhtenäistä samanmielisyyttä. Esineiden internetin arkkitehtuurikerrokset erotetaan järjestelmän johdonmukaisuuden seuraamiseksi. Pohjimmiltaan IoT-arkkitehtuurikerroksia on kolme:

1. asiakaspuoli (IoT-laitekerros)
2. operaattorit palvelinpuolella (IoT-yhdyskäytäväkerros)
3. reitti asiakkaiden ja operaattoreiden yhdistämiseen (IoT-alustakerros).

Jokaisen kerroksen vaatimukset on otettava huomioon, jotta suunnitellusta tuloksesta saa todella toimivan. Lisäksi kestävä IoT-arkkitehtuurin perusominaisuuksia ovat toimivuus, skaalautuvuus, saatavuus ja ylläpidettävyys. Toimivaa IoT-arkkitehtuuria ei voi saada ilman näitä ominaisuuksia, joten nämä kaikki ominaisuudet käsitellään IoT-arkkitehtuurin neljässä vaiheessa. [5.] Neljävaiheinen IoT-arkkitehtuuri on esitelty kuvassa 4 ja koostuu seuraavista:

1. antureista ja toimilaitteista
2. internet-yhdyskäytävistä ja tiedonkeruujärjestelmistä
3. edge-reunatietotekniikoista
4. tietokeskuksista ja pilvistä.

## The 4 Stage IoT Solutions Architecture



Kuva 4. Neljävaiheinen IoT-ratkaisuarkkitehtuuri. [6.]

IoT-arkkitehtuurin eri vaiheet:

- **Vaihe 1:** Verkottuneet esineet (langattomat anturit ja toimilaitteet). Anturit ovat fyysisiä osia, joista saadaan ulkomaailmasta tietoja digitaalisessa tai analogisessa muodossa. Tällä erinomaisella ominaisuudella voidaan muuntaa ulkomaailmasta tieto analysoitaviksi tiedoiksi. Toisin sanoen on tärkeää aloittaa anturien sisällyttäminen nelivaiheiseen IoT-arkkitehtuuriin, jotta saadun tiedon voidaan tosiasiallisesti käsitellä. Toimilaitteiden osalta prosessi menee vielä pidemmälle. Toimilaitteet kykenevät puuttumaan fyysiseen todellisuuteen. Ne voivat esimerkiksi sammuttaa valon ja säätää huoneen lämpötilaa. Anturit ja toimilaitteet katavat ja säätävät kaiken fyysisessä tilanteessa. Toisin sanoen havaitsemis- ja toimimisvaiheissa saadaan tarvittavia tietoja jatkoanalyysijä varten.
- **Vaihe 2:** Anturien tiedonkeruujärjestelmät ja tietomuunnos analogisesta digitaaliseen. Vaikka IoT-arkkitehtuurissa tämä vaihe tarkoittaa edelleen työskentelyä läheisyydessä antureiden ja toimilaitteiden kanssa, myös internetyhdyskäytävät ja tiedonkeruujärjestelmät löytyvät tässä vaiheessa. Internet-yhdyskäytävät toimivat langattoman- ja langallisten lähiverkkojen kautta ja suorittavat lisäprosessointia. Tämän vaiheen elintärkeä merkitys on käsitellä edellisessä vaiheessa kerätty

valtava määrä tietoa ja puristaa se optimaaliseen kokoon lisäanalyysijä varten. Lisäksi ajoituksen ja rakenteen osalta tarvittava muuntaminen tapahtuu täällä. Lyhyesti sanottuna tässä vaiheessa tieto muutetaan digitaaliseen muotoon ja kootaan yhteen.

- Vaihe 3: Edge-reunatietotekniikkajärjestelmien ulkonäkö. Tällä hetkellä internet-arkkitehtuurin vaiheiden sisällä valmistettu tieto siirretään IT-maailmaan. Erityisesti Edge-reunatietotekniikkajärjestelmät suorittavat tässä parannetun analysoinnin ja esikäsittelyn. Se viittaa esimerkiksi koneoppimis- ja visualisointitekniikoihin. Samaan aikaan joitain lisäkäsittelyjä voi tapahtua täällä ennen tietokeskukseen pääsyä. Samoin vaihe 3 on tiiviisti sidoksissa aiempiin vaiheisiin internetin arkkitehtuurin rakentamisessa. Tästä syystä Edge-reunatietotekniikkajärjestelmien sijainti on lähellä sitä, missä anturit ja toimilaitteet sijaitsevat. Samalla on mahdollista myös oleskella etätoimistoissa.
- Vaihe 4: Tietojen analysointi, hallinta ja tallennus. IoT-arkkitehtuurin viimeisessä vaiheessa tärkeimmät prosessit tapahtuvat datakeskuksessa tai pilvessä. Se mahdollistaa perusteellisen käsittelyn ja seurantatarkistuksen palautteelle. Tässä tarvitaan sekä tietotekniikan että operatiivisen tekniikan ammattilaisten taitoja. Toisin sanoen vaihe sisältää jo korkeimman tason analyttiset taidot sekä digitaalisessa että ihmismaailmassa. Siksi muista lähteistä saatavat tiedot voidaan sisällyttää tähän syvällisen analyysin varmistamiseksi. Kaikkien laatustandardien ja vaatimusten täyttämisen jälkeen tiedot palautetaan fyysiseen maailmaan, mutta jo käsiteltynä ja tarkasti analysoituna ulkonäkönä.

Itse asiassa on olemassa myös valinnainen vaihe, jolla voi laajentaa kestävästi IoT-arkkitehtuurin rakentamisprosessia. Se tarkoittaa käyttäjän hallinnan käynnistämistä rakenteessa, jos tulokset eivät tietenkään sisällä täydellistä automaatiota. Päätehtäviä tässä ovat visualisointi ja hallinta. Valinnaisen vaiheen sisällyttämisen jälkeen järjestelmä muuttuu silmukaksi, jossa käyttäjä lähettää komentoja antureille tai toimilaitteille (vaihe 1) suorittaakseen joitain toimintoja ja, jolloin prosessi alkaa uudestaan. [6.]

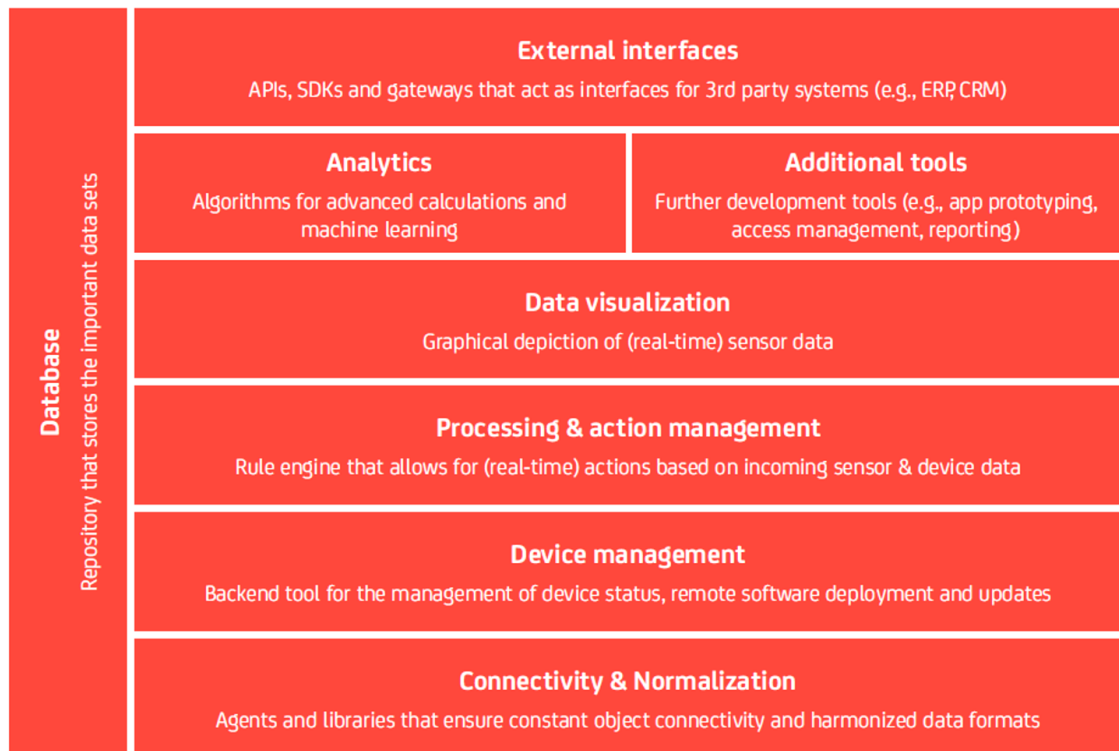
## 4 Mikä on IoT-alusta?

IoT-alusta ei ole vielä tuttu ja selkeä käsite monille ihmisille. Monet eivät vielääkään tiedä, mitä IoT-alustalla voidaan tehdä tai milloin yritysten pitäisi käyttää sitä. Vuoteen 2023 mennessä odotetaan IoT-alustojen markkinan ylittävän 22 miljardia dollaria. IoT-alustoja hyödyntämällä yritykset voivat saada valtavasti lisäarvoa ja yritysten on mahdollista alen-  
taa kustannuksia. Näin ollen myös yritysten nopea kehitys on mahdollista. IoT-alustojen käyttö teollisessa mielessä on teollisuuden 4.0 vaatimusten täyttämässä yksi vaihe. [1.]

IoT-alusta on tärkein paikka esineiden internetille, koska siinä tapahtuu todelliset esineiden internetin taikuudet. IoT-alusta tuo yhteen laitteistot, yhteydet, ohjelmistot ja sovel-  
luskerrokset. Se tarjoaa tehokkaan ratkaisun laitteiden hallintaan ja konfigurointiin, tie-  
donkeruun ja analysointiin sekä sovellusten käyttöön. Lisäksi IoT-alusta mahdollistaa yhteyden pilvipalveluihin tai paikallispalvelimiin. [8.]

Jokaisessa täydellisessä IoT-järjestelmässä on oltava välttämättömiä peruskomponentteja:

- anturit, koneet, toimilaitteet tai muut kytketyt laitteet
- liitettävyyssratkaisu
- tietojenkäsittelyohjelmisto
- käyttöliittymä/sovellus.



Kuva 5. Pitkälle kehittyneet 8-kerroksiset IoT-alustan arkkitehtoniset rakennuspalikat. [9.]

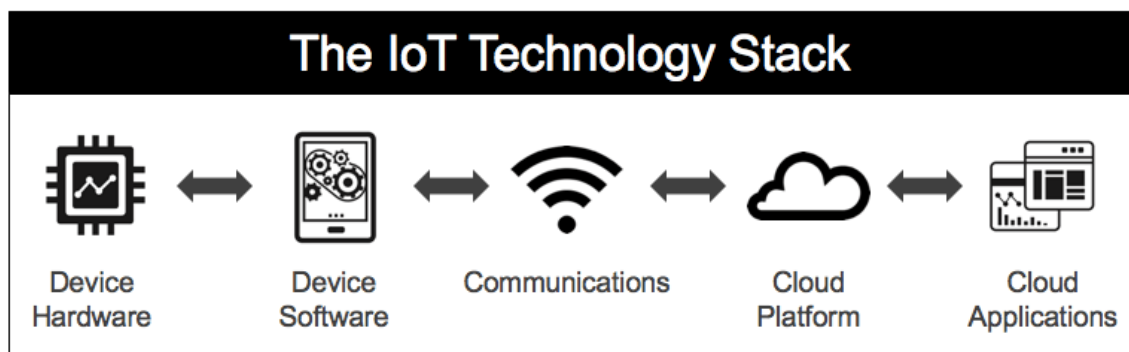
IoT-järjestelmässä on oltava laitteistoja, kuten antureita ja toimilaitteita. Nämä laitteistot keräävät tietoja ympäristöstä, suorittavat annetut toiminnot, ottavat yhteyttä toisiinsa, lähettävät tai vastaanottavat tietoa pilvipalveluista tai paikallispalvelimilta. Esimerkiksi kosteusanturi kerää ympäristöstä kosteusarvot ja lähettää nämä tiedot palvelimelle. Kasvien kastelujärjestelmä menee pois päältä, jos palvelimelta tulee komento, ja kosteusarvo on yli raja-arvon.

IoT-järjestelmä tarvitsee yhteyden, jotta laitteisto siirtää kaikki tiedot palvelimille tai vastaanottaa komennot palvelimilta. Yhteysratkaisuna on tarjolla esimerkiksi matkapuhelinverkko-, satelliitti-, wifi- tai LoRa-vaihtoehtoja. IoT-projektin vaatimuksista tai rajoituksista riippuen toimitetun yhteysratkaisun on tarjottava ominaisuuksia, jotka vastaavat niitä parhaiten kuten luotettavuus, kantama, kaistanleveys, turvallisuus ja virrankulutus.

Ohjelmiston tehtävä on kerätä, analysoida ja tehdä itsenäisiä päätöksiä tiedoista IoT-järjestelmässä, joita saa laitteistosta. Esimerkiksi se havaitsee sadeveden kosteusarvotietojen avulla ja käskee kastelujärjestelmää, ettei tänään kastella. Ohjelmisto sijaitsee palvelimessa.

Käyttöliittymä mahdollistaa vuorovaikutuksen IoT-järjestelmän ja käyttäjän välillä. Sen päätehtävä on visualisoida ohjelmistosta saadut tiedot ja esitellä ne käyttäjäystävällisellä tavalla. Esimerkiksi verkkopohjainen sovellus, joka näyttää kosteusarvot ja antaa käyttäjille mahdollisuuden kytkeä kastelujärjestelmät manuaalisesti päälle tai pois päältä.

IoT-alusta kutsutaan IoT-väliohjelmaksi, sillä se on välittäjä sovelluskerroksen ja laitteiston välillä, kuten kuva 6 esittelee. IoT-alustan päätavoite on poistaa yksittäisten teknologiakerrosten väliset esteet ja yhdistää ne tehokkaan ja saumattoman yhteistyön varmistamiseksi. IoT-alustat mahdollistavat nopean liikkeellelähdon tarjoamalla sisäänrakennettuja työkaluja ja kykyjä, jotka tekevät IoT-järjestelmistä edullisen ja helpomman käyttäjille, yrityksille ja kehittäjille. IoT-alusta helpottaa tiedonsiirtoa, viestintää, laitehallintaa ja sovellusten toimivuutta. On oltava ratkaisu, joka mahdollistaa toimimaan kaikki yhdessä, vaikka kyseessä olisi erilaisia laitteistotyyppisiä tai erilaisia yhteysvaihtoehtoja. IoT-alustat auttavat juuri näissä ongelmissa.



Kuva 6. IoT-alusta on välittäjä sovelluskerroksen ja laitteiston välillä. [10.]

IoT-alustojen edut:

- laitteiston yhdistäminen, kuten anturit ja toimilaitteet
- mahdollisuus käyttää erilaisia laitteisto- ja ohjelmistoviestintäprotokollia
- mahdollisuus tarjota laitteille ja käyttäjille tietoturva ja todennusta
- mahdollisuus kerätä, visualisoida ja analysoida antureiden ja laitteiden keräämiä tietoja
- integrointi nykyisiin yritysjärjestelmiin ja muihin web-palveluihin.



IoT-alustoja löytyy monia eri vaihtoehtoja. IoT-alustoilta osa on keskittynyt tiettyihin aloihin, osa on keskittynyt yrityksiin ja osa on keskittynyt pienempiin projekteihin. Siksi IoT-alustoja voidaan luokitella ominaisuuksien perusteella, joita IoT-alusta tarjoaa. IoT-alustojen tärkeimpinä ominaisuuksina voidaan luetella skaalautuvuus, helppokäyttöisyys, turvallisuus ja yhteistoimivuus.

Skaalautuvuudella tarkoitetaan sitä, että ominaisuuksiltaan pätevät IoT-alustat kykenisivät tarjoamaan IoT-järjestelmien laitekapasiteettien laajennuksen ja maksimaalisen suorituskyvyn olosuhteista riippumatta.

Helppokäyttöisyys- tai käyttäjäystävällisyyskriteereillä tarkoitetaan yleensä integraatio-rajapintojen joustavuutta ja kykyä hallita niiden koodia. Kehittäjät haluavat muokata laitteen ja sen sovellusten toimintoja tai kehittää lisämoduuleja helposti, joten IoT-alustan pitää olla käyttäjäystävällinen.

Edistyneellä IoT-alustalla yhteistoimivuus kriteeri on tärkeä osa, sillä laitteita ja viestintäprotokollia on monentyyppisiä. Tiedonsiirto pilvipalvelimiin tai paikallispalvelimiin erilaisilla laitteilla ja viestintäprotokollilla on tärkeä ominaisuus IoT-alustalle.

Käyttöönottovaihtoehdot ovat yrityksille tärkeä kohta. IoT-alustan pitäisi saada vaihtoehtoisesti asennettua pilvipalvelimiin tai suoraan paikallisiin palvelimiin.

Yritysten päätös IoT-alustojen valinnassa ei ole helppoa, jos ajatellaan, että markkinoilla olevien IoT-alustojen määrää ylittää 400 eri vaihtoehtoa ja jokaisen yrityksen pitää keskittyä yksilöllisiin yritystarpeisiin. Tämä vaatii perusteellisen analyysin tarjolla olevien erilaisten ratkaisujen joukosta.

IoT-alusta kykenee salaamaan siirrettyjä tietoja laitteiston ja palvelinten välillä. Lisäksi turvallinen IoT-alusta sisältää hallinnan käyttäjäpääsyihin sekä omistusoikeuden tietojen käsittelyyn. Jotkut vanhat IoT-järjestelmät vaikeuttavat IoT-tietoturvan luotettavuutta langattoman päivitystoiminnon puuttuvuuden vuoksi. [7; 8.]

Suurien määrien IoT-laitteiden lisäys (eng. onboarding) IoT-alustaan manuaalisesti on ajallisesti haastava. Tämän takia jotkut IoT-alustat tarjoavat helpotettuja ja turvallisia ratkaisuja lisätä laitteita. Helpotettu ja tietoturvallinen ratkaisu voi esimerkiksi edellyttää IoT-laitteelta laiteavainta. Laiteavain määrittellään turvallisesti IoT-laitteen tuotantovaiheessa. [12.]

## 5 Avoin lähdekoodi

Avoimeksi lähdekoodiksi kutsutaan tietokoneohjelmistojen jakelu- ja kehitystapa, jossa ohjelmiston lähdekoodit ovat asiakkaan vapaasti käytettävissä, kuten kopioiminen, muokkaaminen ja jakaminen ilman lisenssimaksuja tai lisenssin ylläpitoa. [13.] Avoin lähdekoodi (engl. open source) on Open Source Initiativen (OSI) keksimä termi, vaikka ennen tätä termiä oli käytössä toinen termi kuten ”vapaa ohjelmisto” (engl. free software), joka viittasi myös ohjelmiston ilmaisuuteen termissä käytettyjen englanninkielisten sanojen vuoksi. [14.]

Avoimella lähdekoodilla on suuria etuja. Kaikista suurimpana etuna pidetään sitä, että se estää monopolien syntymistä. Tämä tapahtuu toimittaja- ja ohjelmistoriippumattomuudella, koska avoin lähdekoodi mahdollistaa käyttäjän valita vapaasti ohjelmiston ja toimittajan erikseen. Näin ollen riskiä joutua yhden toimittajan varaan ei synny. Avoimen lähdekoodin ohjelmistojen kehitykseen tai ohjelmistovirheiden korjaamiseen ei osallistu vain yksityishenkilöjä, vaan myös isot yritykset ovat mukana. Tämän maailmalaajuisen yhteistyön avulla ohjelmistoista tulee laadukas, turvallinen ja integroitava muihin ohjelmistoihin. [14; 15.] Avoimen lähdekoodin ohjelmistoa tuottavien yritysten olemassa olevien ohjelmistotuotteiden kehitystyötä on mahdollista jatkaa myös yrityksen konkurssin jälkeen, sillä lähdekoodit ovat saatavilla kaikille. Avoin lähdekoodi antaa mahdollisuuden käyttäjille, jotka haluavat olla sitoutumatta toimittajalle, joten vaikka toimittaja hylkäisi ohjelmiston niin käyttäjän ei tarvitse välttämättä hakea tilalle toista ohjelmistoa. Suljetun lähdekoodin ohjelmiston lähdekoodit eivät ole julkisesti julkaistu, joten muut riippumattomat kehittäjät eivät pääse tutkimaan ohjelmiston toimintoihin. Siinä mielessä avoin lähdekoodi voi olla turvallisempi ja luotettavampi. [14.] Avoimen lähdekoodin periaatteella toimivien ohjelmistojen suosio kasvaa jatkuvasti enemmän ja avoimien rajapintaratkai-

sujen käyttöönotto yleistyy. Isot yritykset ovat nykyään rakentamassa ohjelmistoja avoimen lähdekoodin päälle. Esimerkkejä tunnetumpiin avoimen lähdekoodin ohjelmistoihin: Linux-käyttöjärjestelmät, Firefox-selain, OpenOffice. [15.]

Open Source Initiativen laatiman määritelmän mukaan avoimen lähdekoodin ohjelmistossa täytyy olla seuraavat ehdot:

1. Ohjelman täytyy olla vapaasti levitettävissä ja välitettävissä.
2. Lähdekoodin täytyy tulla ohjelman mukana tai olla vapaasti saatavissa.
3. Myös johdettujen teosten luominen ja levitys pitää sallia.
4. Lisenssi voi rajoittaa muokatun lähdekoodin levittämistä vain siinä tapauksessa, että lisenssi sallii korjaustiedostojen ja niiden lähdekoodin levittämisen. Voidaan myös vaatia, ettei johdettua teosta levitetä samalla nimellä tai versionumerolla kuin lähtöteosta.
5. Yksilöitä tai ihmisryhmiä ei saa asettaa eriarvoiseen asemaan.
6. Käyttötarkoituksia ei saa rajoittaa.
7. Kaikilla ohjelman käsiinsä saaneilla on samat oikeudet.
8. Lisenssi ei saa olla riippuvainen laajemmasta ohjelmistokokonaisuudesta, jonka osana ohjelmaa levitetään, vaan ohjelmaan liittyvät oikeudet säilyvät, vaikka se irrotettaisiin kokonaisuudesta.
9. Lisenssi ei voi asettaa ehtoja muille ohjelmille. Ohjelmaa saa levittää myös yhdessä sellaisten ohjelmien kanssa, joiden lähdekoodi ei ole avointa.
10. Lisenssin sisällön pitää olla riippumaton teknisestä toteutuksesta. Oikeuksiin ei saa liittää varauksia jakelutavan tai käyttöliittymän varjolla. [13.]

## 6 Avoimen lähdekoodin IoT-alustat

### 6.1 Thinger

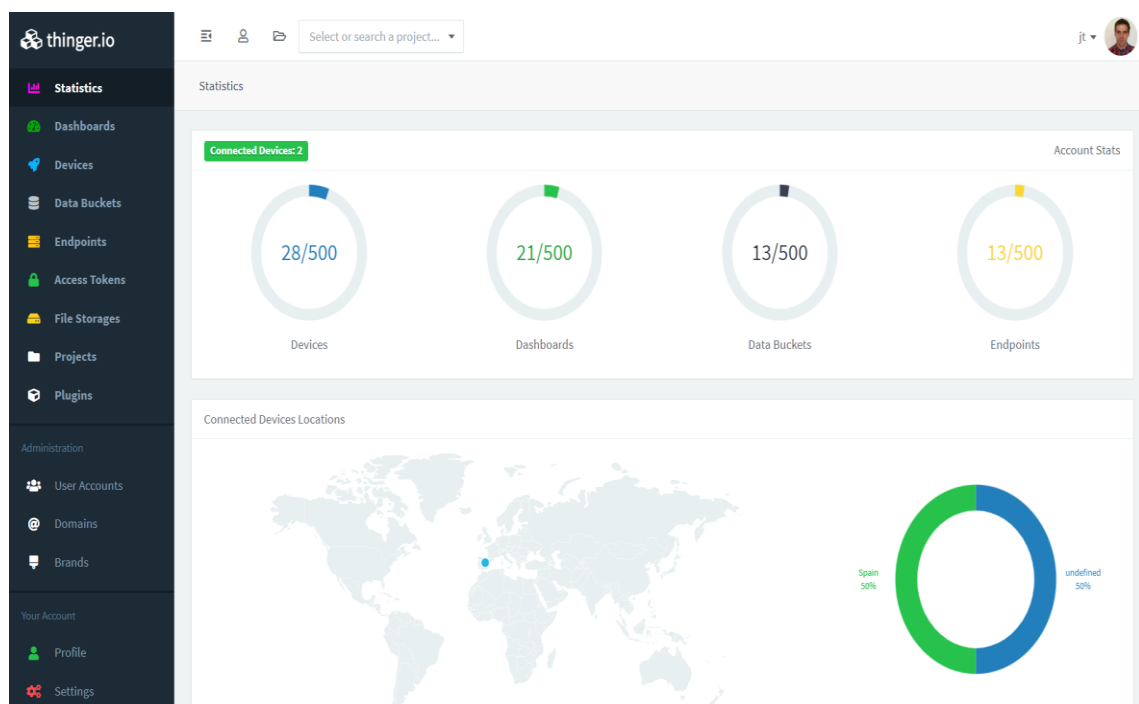
Thinger on avoimella lähdekoodilla toimiva IoT-alusta, ja se on selkeä ja yksinkertainen alusta. Alusta on pilvipohjainen eli sitä ei tarvitse asentaa paikallisesti mihinkään. Käyttämään pääsee heti, kun alustalle luodaan tili. Tilin luomiseen ei tarvita muuta kuin käyttäjänimi, sähköpostiosoite ja salasana. Ilmaisella versiolla jokainen tällä alustalla saa pienen IoT-projektinsa pyörimään. Ilmaisella versiolla kaikkea ei kuitenkaan saa, koska jotkut ominaisuudet ovat rajattua. Mikäli halutaan isompia projektia, silloin tilin päivitys maksulliseen voi olla tarpeellinen. Maksullisella versiolla saa vaihtoehtoisista pilvipalvelutarjoajalta haluamansa edistyneen pilvipalvelun. Maksullisissa pilvipalveluissa on taas eri hintaisia palveluluokkia, jotka määräytyvät muun muassa yhdistettävien laitteiden määrästä, siirrettävästä datamäärästä tai palvelinten komponenttien tehokkuudesta. Esimerkiksi halvimmillaan 29 eurolla kuukaudessa saa sellaisen paketin, jossa on mm. rajaton laitemäärä, 2 TB:n datamäärä ja palvelin, joka muodostuu komponenteista: 1 CPU, 1 Gt RAM, 40 Gt SSD.

Thinger IoT -alusta tukee Arduino- ja Linux/Raspberry pi- laitteita sekä niitä laitteita, jotka tukevat protokollia: mqtt, http ja sigfox. Lisäksi Thinger IoT -alustan omassa sivustossa myydään erityisesti Thinger-alustalle tarkoitettuja IoT -laitteita kuten ”ClimaStick” ja ”Wifi Buttonia”. ClimaStick on Arduinin ympäristöä tukeva piirilevy, joka sisältää monenlaisia antureita, joilla pääsee testaamaan IoT-projekteja mm. lämpötilojen mittausta ja näiden tietojen lähetystä Thinger-alustalle. Tällä hetkellä piirilevyn hinta Thinger-sivuston verkkokaupassa on alkaen 35 euroa. Thinger-alustan internetsivulla Thingerin omien IoT-laitteiden käyttämiseen löytyy hyviä ohjeita. Wifi Button on IoT-laite, joka voidaan käyttää käyttäjävaatimuksista riippuen minkä tahansa toimintojen suorittamiseen helposti Thinger-alustalle, kuten ilmoituksen lähettäminen sähköpostiin. Myös Wifi Button tukee Arduinin ympäristöä ja hyviä valmiita käyttöohjeita saa Thinger-sivustosta.

Thinger IoT -alustalle löytyy oma mobiilisovellus, joka on ladattavissa Apple IOS- tai Android- käyttöliittymän älypuheliiniin. Mobiilisovellus mahdollistaa valvomista Thinger-alustalle kytkettyjen laitteiden tilan sekä saa visualisoimaan laitteista tulevaa tietovirtaa älypuhelimien ruudulle. Mobiilisovelluksella voidaan myös lähettää kommentoja tai tietoja

Thinger-alustan laitteisiin. Thinger-mobiilisovellus vaatii toimiakseen laitteen token-koodin. Tämä löytyy laitteenkonfigurointi kohdassa. Token-koodi on laitteelle satunnaisesti luotu pitkä pääsytunnus. Token-koodi löytyy myös QR-muotona. Se voidaan lukea suoraan älypuhelimien kameralla. Thinger-alustalle kytkettyjen laitteista tulevia tietoja voidaan nähdä hienoilla erityyppisillä kaavioilla mobiilisovelluksessa.

Thinger IoT -alustan ominaisuuksia voidaan laajentaa erilaisilla laajennuksilla (engl. Plugins). Tällä hetkellä tarjolla ovat Node-RED-, The Things Network- ja Sigfox -laajennukset, joilla voidaan räätälöidä enemmän toimintoja ja ottaa käyttöön tarpeen mukaan. Ilmaisversio ei sisällä laajennuksien käyttöönottoominaisuutta. Asiakasyrityksille löytyy toinen tärkeä ominaisuus eli yrityksen oman logon sijoittaminen oman Thinger-alustan hallintasivulle, eli asiakasyritys saa IoT-alustan oman yrityksen väreihin sopivammaksi, mikä on tärkeä yrityksen markkinoinnin osalta. Thinger-alusta on lisensoitu MIT-lisensillä. Alustaa voi näin ollen käyttää vapaasti myös kaupallisiin tarkoituksiin.



Kuva 7. Thinger IoT -alustan hallintasivu. [16.]

Kun on kirjautunut ilmaiselle IoT-alustalle, vasemmalla kohdalla ilmestyy toimintoja sarakkeena, joita pääsee rajallisesti käyttämään. Kojelaudat (engl. Dashboards) -koh-

dassa voi luoda kojelautoja. Ilmaisella versiolla saa luoda enintään neljä kappaletta kojelautaa. Kojelaudalla päästään visualisoimaan ja valvomaan tiedot lisäämällä tänne eri näköisiä ”widgetsejä” eli vimpaimia. Vimpaimia muokataan muokkaamalla niiden kokoja tai siirtämällä niiden sijaintia kojelaudalla.

Laitteet (engl. Devices) -kohdassa luodaan laitteita. Laitteen luonti onnistuu ”Add device” -kohdan avulla. Laitteiden määrä on rajattu ilmaisversiolla kahteen kappaleeseen ja laitetypiksi voi valita ainoastaan Thinger-laitteen. Jokainen laite täytyy yksilöidä esimerkiksi määrittämällä laitteelle ID-nimi ja salasana. Luotuun laitteeseen päästään laitteen kojelaudasta, joka on valmiiksi muokattu laitteen käyttötarkoituksen mukaisesti. Laitteen kojelaudassa löytyy laitteesta tietoja, kuten laitteen IP-osoite, tilatietoa ja siirretty datan määrä. Laitteen kojelaudassa voidaan seurata laitetta reaaliaikaisesti visualisoidulla kaaviolla sekä antaa laitteelle ”Device Token” eli laitteen pääsytunnus ”Addin” kautta. Tällä pääsytunnuksella kolmansien osapuolten laitteet tai palvelut saadaan yhdistetty Thinger IoT-alustalle. Lisäksi laitteen kojelaudassa on ”Api Explorer” ominaisuus, jolla päästään testaamaan laitteen toimivuutta reaaliaikaisesti.

”Data Buckets” -kohdassa luodaan tietokantoja. Thinger IoT -alustan ilmaisversio on rajoittanut tietokantojen määrän neljällä kappaleella. Tietokannan määrittelyssä täytyy valita, milloin tieto tallennetaan. Pitää valita vaihtoehto ”From Device Resources”, jos laitteen on oltava yhteydessä IoT-alustalle jatkuvasti. Valitaan vaihtoehto ”From Write Call”, jos laite kirjoittaa tiedon tietokantaan vasta laitteen itse määrittäessä tai käyttäessä ”Rest Api”. IoT-alustalle tallennettu tietokanta voidaan ladata suoraan tietokoneelle CSV, ARFF- tai Json-muodoissa.

”Endpoints” -kohdassa voidaan määrittää erilaisia automaattisia toimintoja, kuten esimerkiksi tekstiviestin tai sähköpostin lähetys, jos laitteessa on tapahtunut tilannemuutos. Halutessa on mahdollista valita JSON-muoto tai tavallinen tekstimuoto laitteesta tulevalle viestille. Toiminnon testaus kuten sähköpostin lähetys on mahdollista painamalla ”Test Endpoint”.

”Access Tokens” -kohdassa voidaan määrittää pääsyoikeudet koko IoT-alustan hallintaan tai esimerkiksi pelkkien laitteiden hallintaan API REST:ien kautta. Tätä varten luodaan token painamalla ”Add token” ja valitaan, mihin asetuksiin halutaan päästä. Kun

Access Token on luotu, niin saadaan pitkä pääsytunnus, joka voidaan käyttää API REST:eissa.

## 6.2 Freeboard

Freeboard on avoimeen lähdekoodin periaatteisiin perustuva esineiden internet (IoT) -kojelauta, joka toimii internetselaimella pilvipohjaisesti tai paikallisesti eli lataamalla kovalevylle. Freeboard kuvailee palvelunsa omalla internetsivulla mielenkiintoisella lauseella: ” Naurettavan yksinkertaiset kojetaulut laitteillesi ”. Tällä hetkellä jotkut isot yritykset kuten Ford Motor Company, Verizon, Renesas Electronics ja BuildingLink käyttävät Freeboardin palveluja. Freeboardin hallintasivu näyttää yksinkertaiselta ja toimintojen osaltakin vaikuttaa suppealta. Freeboardin valmiiden vimpaimien kokoa ja väriä ei pysty vaihtamaan. Myöskään kojelaudan väriä ei voi vaihtaa. Kuitenkin käyttäjän luomien vimpainten värit saavat olla eri värisiä, joita luodaan JavaScript-kielellä.



Kuva 8. Freeboardin hallintasivu [17.]

Freeboard ei vastaa IoT-alustan vaatimuksia, mutta sillä hoituu helposti ja nopeasti osittain IoT-alustojen palveluita. Freeboard on kojelauta ja visualisointipalvelu, joiden avulla voidaan reaaliaikaisesti seurata ja visualisoida IoT-laitteiden lähettämät tiedot. Freeboard-palvelua pääsee kokeilemaan täysin ilmaiseksi avaamalla tilin Freeboardille. Kuten muissa IoT-alustoissa niin Freeboardillakin löytyy maksullisia toimintoja. Maksullisia palveluita saa halvimmillaan micro-paketin hintaan 12 dollarilla kuukaudessa. Se sisältää rajattoman laitemäärän, vimpainmäärän, kojelautamäärän, oman logon ja viisi kappaletta yksityiskojelautaa.

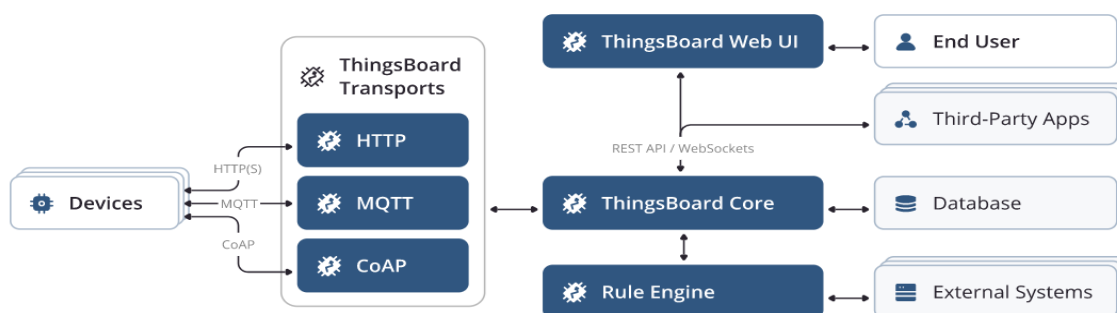
Freeboardille voi integroida dweet-viestipalvelun ja mitä tahansa web-pohjaisia sovellusrajapintoja. Kojelauta tukee älypuhelimia sillä tavalla, että se skaalautuu hyvin älypuhelinruudussa. Kojelauta on helposti jaettavissa muiden ihmisten katseluun. Siihen tarvitaan vain kojelaudan url-osoite, joka syötetään mille tahansa selaimelle. Toinen freeboard-ominaisuus on sen kyky toimia kokonaan selaimessa yksisivuisena staattisena verkkosovelluksena ilman palvelinta. Ominaisuus tekee siitä erittäin houkuttelevan käyttöliittymänä sulautetuille laitteille, joiden kyky palvella monimutkaisia ja dynaamisia verkkosivuja on rajoitettu.

### 6.3 Thingsboard

Thingsboard on avoimiin lähdekoodiin perustuva IoT-alusta. Se on tarkoitettu kaiken kokoisille yrityksille ja varsinkin teollisuusalueen yrityksiin. Tällä IoT-alustalla on kaksi eri nimistä versiota: Community Edition -versio on ilmainen ja sisältää esimerkiksi rajattomasti laitemäärän, mutta toimintoja on rajoitettu aika paljon. Professional Edition -versio on maksullinen ja sisältää laajempia toimintoja. Tässä versiossa maksut määräytyvät käytettävien laitteiden määrästä ja tietyn ajan tukipalvelusta sähköpostitse. Professional Edition -version hinta alkaa 10 USD dollarilla per kuukausi. Lisäksi Thingsboard-alusta tarjoaa erikseen maksullisia tukipaketteja kolmessa eri hintaluokassa. Hinnat määräytyvät esimerkiksi pyörittävien palvelinten määrästä, yhteyshenkilöiden määrästä tai sähköpostikonsultoinnin pituudesta. On myös toinen palvelu, ”Laitteisto Kumppani Ohjelma”. Tämä on tarkoitettu laitevalmistajien käytölle. Thingsboard tarjoaa kumppaneille mm. Thingsboardin internetsivulla ilmaisia verkkopohjaisia opasteita laitteiden asennukselle Thingsboard IoT -alustalla. Lisäksi se tarjoaa maksullisia palveluita, jos kumppanit

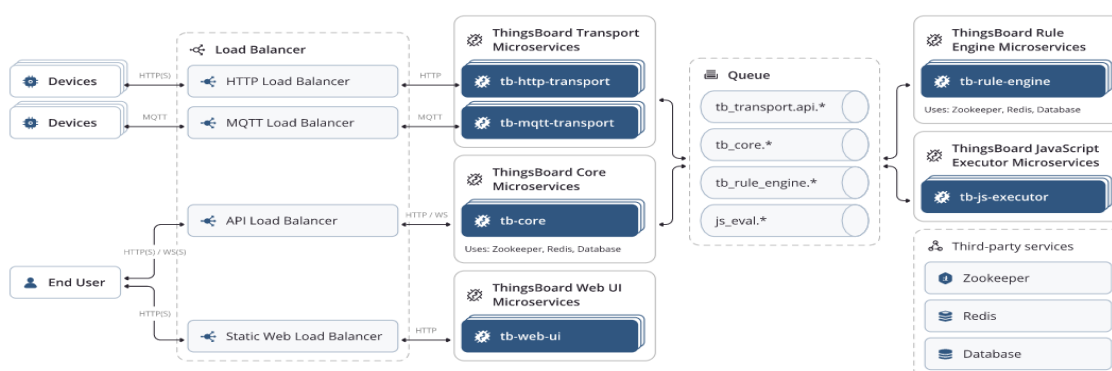


haluavat saada laajempia tukipaketteja. Tällä hetkellä Thingsboard IoT -alustaa käyttävät paljon suuria tunnettuja yrityksiä.



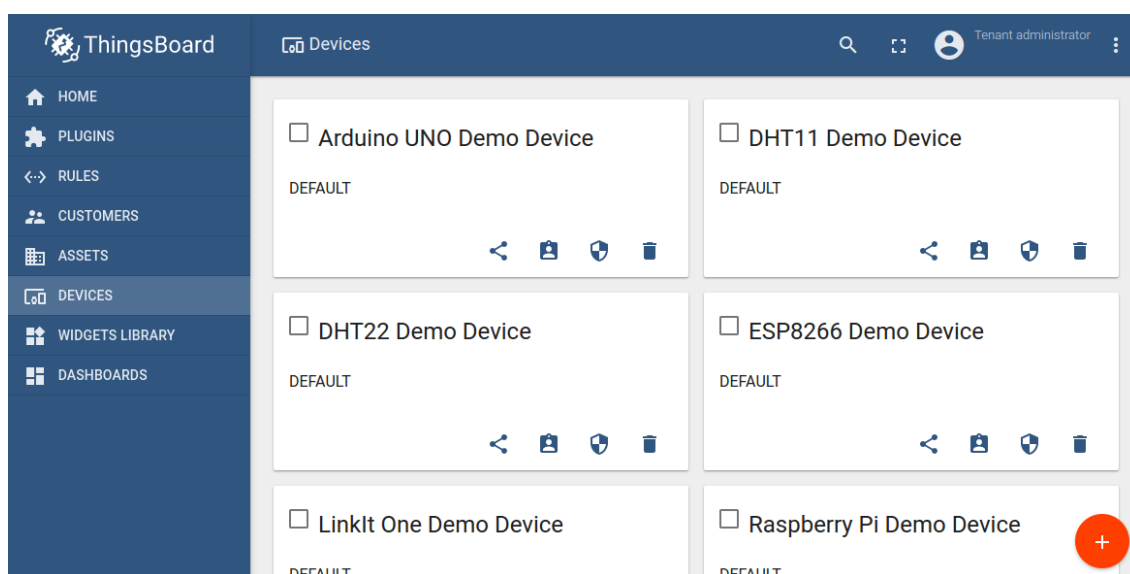
Kuva 9. Thingsboard IoT -alustan monoliittinen arkkitehtuuridiagrammi [18.]

Thingsboard mahdollistaa IoT-projektien nopean kehittämisen, hallinnan ja skaalaamisen IoT-pilvillä tai paikallisilla ratkaisuilla. Thingsboardilla voidaan määrittellä laitteen, asiakkaan ja omaisuuden (eng. asset) väliset suhteet helposti. Alustalla löytyy mm. tiedon prosessoinnin jälkeen hälytyksen lähetystoiminto, kaksisuuntainen laitteiden hallinta RPC-yhteydellä, laitteiden ja omaisuuksien tietojen keräys ja visualisointi erilaisilla vimpaimilla. Alustan ominaisuudet ovat haluttaessa laajennettavissa plugineilla. Thingsboard IoT -alusta kerää, prosessoi ja visualisoi IoT-laitteista tulevia tietoja käyttäen yhteysprotokollat MQTT, CoAP ja HTTP. Thingsboard tukee muita suosittujen kolmannen osapuolten IoT-alustojen integraatioita kuten AWS IoT:ta, AWS Kinesista, IBM Watsonia, Azure Event Hubia sekä laitteita, jotka tukevat TCP:tä, Sigfoxia, TheThingsNetworkia. Alusta on kirjoitettu monoliittiarkkitehtuurilla, mutta versiosta 2.2 lähtien alusta se on siirtynyt mikropalveluarkkitehtuurille.



Kuva 10. Thingsboard IoT -alustan mikropalveluarkkitehtuuridiagrammi [19.]

Thingsboard IoT -alustan asennus onnistuu erilaisiin paikallisiin ympäristöihin, virtuaali-koneisiin, kuten Dockeriin tai suoraan ulkoisiin palvelimiin. Thingborgin internetsivuilta löytyy hyviä opasteita erilaisiin asennusympäristöihin. Alustan voi asentaa paikallisesti esimerkiksi Windows-, Raspberry Pi-, Linux -pohjaisiin (Ubuntu, Redhat, CentOS) laitteisiin. Minimissään ohjelmiston asennus vaatii Linux-koneissa 1 GT keskusmuistin käytettäessä PostgreSQL-tietokantaa. Windows-koneissa minimikeskusmuistin määrä on 2 GT. Molemmissa tapauksissa tietokoneeseen tulee olla asennettuna Java 8 -ajoympäristö.



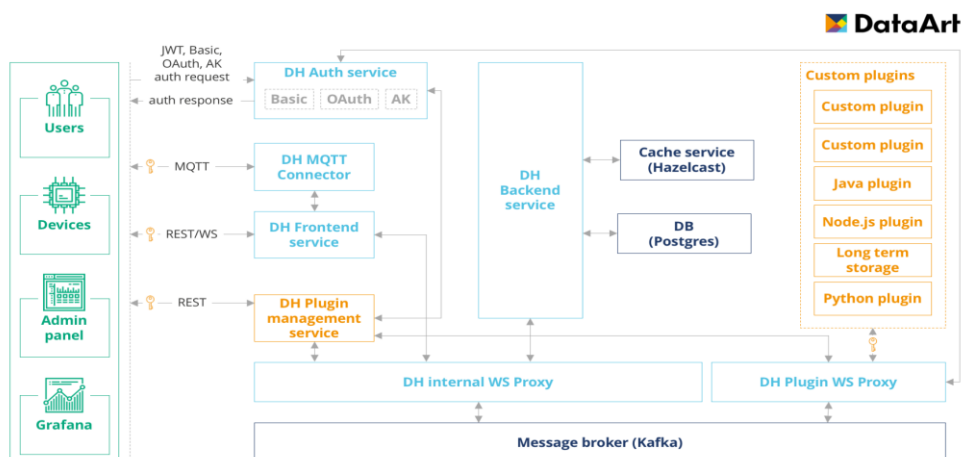
Kuva 11. Thingsboard IoT -alustan hallintasivu. [20.]

ThingsBoard on lisensoitu Apache 2.0 -lisenssillä. Näin ollen alustaa voi käyttää täysin vapaasti myös kaupallisiin tarkoituksiin. Tietoliikenteen salaus tapahtuu MQTT-palvelimen SSL:n tai HTTPS-protokollan kautta. Jokaisella laitteella on yksilölliset käyttöoikeustunnukset. Näitä käytetään yhteyden muodostamisessa, ja myös X.509-sertifikaattituki on tulossa alustalle. Alustalla tiedot tallennetaan Cassandra-tietokantaan, joka on Java-pohjainen ja tarkoitettu suurien määrien tietojen tallentamiseen. Laitteiden integrointi Thingier IoT -alustalle onnistuu "Thingsboard IoT Gateway" -ratkaisulla. Tämä ratkaisu on Thingsboardin keksimä ohjelmistokomponentti, joka on suunniteltu toimimaan Python 3.5:tä tukevilla Linux-pohjaisissa koneissa.

Thingsboard IoT -alustassa löytyy ”rule engine” -niminen ominaisuus. Tämä prosessoi IoT-laitteista tulevia viestejä ja sen kautta käynnistää erilaisia toimintoja, kuten lähettää sähköpostia, kun laitteen ennalta määritetyt raja-arvot ylittyvät. Sääntömoottorin toiminta perustuu kahden pääkomponentin eli laajennuksiin ja sääntöihin. Sääntö-pääkomponentti vastaa datan analysoinnista, kun laajennusosa suorittaa toiminnot.

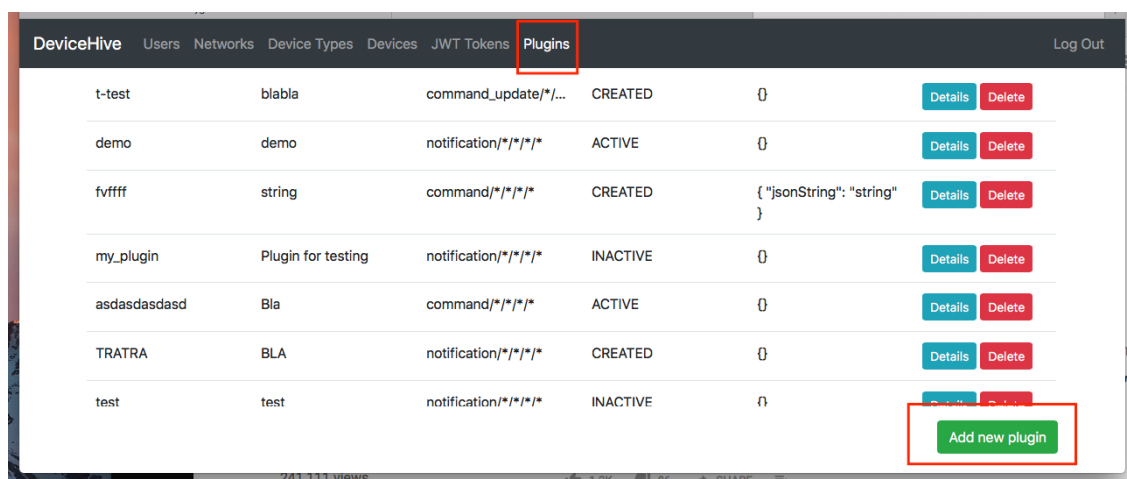
#### 6.4 DeviceHive

DeviceHive on avoimen lähdekoodin IoT-alusta. Se on lisensoitu Apache 2.0 -lisenssi-versiolla. Tämä IoT-alusta on keskittynyt erityisesti isojen tietojen analysointiin. DeviceHive IoT -alustaa pääsee käyttämään suoraan internetselaimella tai paikallisesti. Alustan voi asentaa myös Docker- ja Kubernetes -ratkaisulla ja näihin asennusratkaisuihin alustan internetsivulla löytyy ohjeita. Alustan käyttö on ilmainen, ja alusta pohjautuu mikro-palvelutarkkitehtuuriin. Ilmaista tukea alustan konsultilta on saatavilla, mutta maksullisia tukipaketteja myös löytyy, mikäli haluaa tiheämpää ja nopeampaa asiakaspalvelua. Hintavain paketti sisältää jatkuvan 7/24-tuen ja oman yhteyshenkilön. DeviceHive IoT -alustalla ei ole rajoituksia, eli alusta saa käyttää rajattomasti ja ilmaiseksi. Ainoastaan tukipaketit ovat maksullisia.



Kuva 12. DeviceHive IoT -alustan arkkitehtuuri. [21.]

DeviceHive IoT -alustalle pääsee helposti luomalla tilin ja käyttämällä pilvipohjaista käyttöönottoratkaisua, joka tällä alustalla on nimetty Playgroundiksi. Kirjautumisen jälkeen internetsivulla näkyy kolme eri hallintakonseleja: Admin Console, Dashboard ja API Console. Alustalle IoT-laitteiden lisäys, käyttäjien hallinta, laitteille token-koodien luonti, laitteiden yhdistäminen verkkoon, laitteille komentojen lähetys tai laitteista saapuvien ilmoitusten tarkkailu ja laitteiden tyyppien tai ominaisuuksien muuttaminen onnistuu Admin Consolessa eli järjestelmävalvojan konsolissa. Alustan muihin ominaisuuksiin pääsee tutustumaan ja hallinnoimaan Grafana-nimisellä laajenuksella. Tämä työkalu kerää tietoja DeviceHive-palvelimilta ja näyttää erilaisilla kojelautoilla. Grafanassa hoituu myös alustan muitakin tehtäviä, kuten tietokantojen lisäämistä. DeviceHive IoT -alustan ominaisuudet ja tehtävät on jaettu erilleen kahteen osaan. Kaikkiin toimintoihin ei siis pääse samasta ikkunasta. Kaikille RESTful-palveluille DeviceHive tarjoaa Swagger-API-kehittämistyökalun, jonka avulla voidaan testata asennusta tai tutkia DeviceHive-ominaisuuksia. Tähän työkaluun päästään ”Api Consolella” eli API-konsolilla.



Kuva 13. DeviceHive IoT -alustan hallintapaneeli. [22.]

DeviceHive IoT -alusta tukee cURL-kirjastoa. Esimerkiksi alustalle laitteen lisääminen onnistuu helposti cURL-komennolla. Alustan tukevat protokollat ovat REST, Websocket ja MQTT lisälaajenuksena. Kaikki viestintä tapahtuu JSON-viestien kautta. Lisäksi alustaa tukevat seuraavat kirjastot: Python, Node.js, Java, Android ja IOS. Alustan internetsivulla löytyy hyviä asennusohjeita ESP8266-mikro-ohjaimesta sekä monista muista IoT-antureista ja tarjoaa näihin yksinkertaisia ohjelmointirajapintoja eli API:ja. Tämä alusta

hyödyntää suurien tietojen analysoinnissa Apache Kafka-, Spark-, Cassandra- ja Elasticsearch-ratkaisuja ja tarjoaa skaalautuvia julkisia, yksityisiä tai hybridipilviresursseja. Alusta voidaan integroida muihin laitteiden pilvipalveluihin tai alustoihin käyttämällä tuettuja protokollia esimerkiksi suoraan Alexaan käyttämällä mukautettua javascript-koodia. DeviceHive plugineilla eli laajennuksilla voi tilata DeviceHive-ilmoituksia muilta pilvipalveluilta esimerkiksi lähettää anturitiedot pilvitallennuspalvelu Amazon S3:lle. DeviceHive-internetsivulla löytyy laajennusmalleja. Näitä voidaan käyttää hyväkseen tai muokata oman tarpeen mukaan.

Alustan turvallisuus on toteutettu seuraavilla ratkaisuilla:

- Laitteiden ja sovellusten viestinnässä käytetty TLS-yhteys.
- Pääsy käyttäjiin, verkkoihin tai laitteille on roolipohjainen.
- Laitteiden todennus on turvattu JSON Web Tokensilla eli lyhyeksi JWT:lla.

## 7 Lämpötila- ja kosteusarvojen visualisointi IoT-alustassa

Tämän työn lämpötila- ja kosteusarvojen visualisointiin päätettiin valita Thingsboard IoT-alusta. Päätökseen vaikutti selkeiden ohjeiden saatavuus, MQTT-protokollan tuettavuus sekä käyttöönoton helppous. Thingsboard IoT -alusta pääsee käyttämään suoraan ”live demo” -nimetyllä käyttöönotolla Thingsboardin palvelimilla, eli IoT-alustan asennusta ei tarvita.

Testausympäristöksi valittiin Raspberry Pi -pienitietokone, josta lähetetään lämpötila- ja kosteusarvot. Tässä tietokoneessa toimii Raspbian linux-käyttöjärjestelmä. Lämpötila- ja kosteusarvojen mittaamiseen käytettiin Raspberry Pi:n valmistama Sense Hat -niminen laajennuskortti, joka liitetään suoraan Raspberry Pi:hin. Tässä laajennuskortissa löytyy monien arvojen mittaamiseen antureita valmiiksi, kuten gyroskooppi, kiihtyvyy-, magnetometri-, ilmanpaine-, lämpötila- ja kosteusantureita. Koodauskielenä käytettiin python-koodia.



Kuva 14. Raspberry Pi -pienitietokone sekä kosteus- ja lämpötila-arvot SenseHatin näytössä.

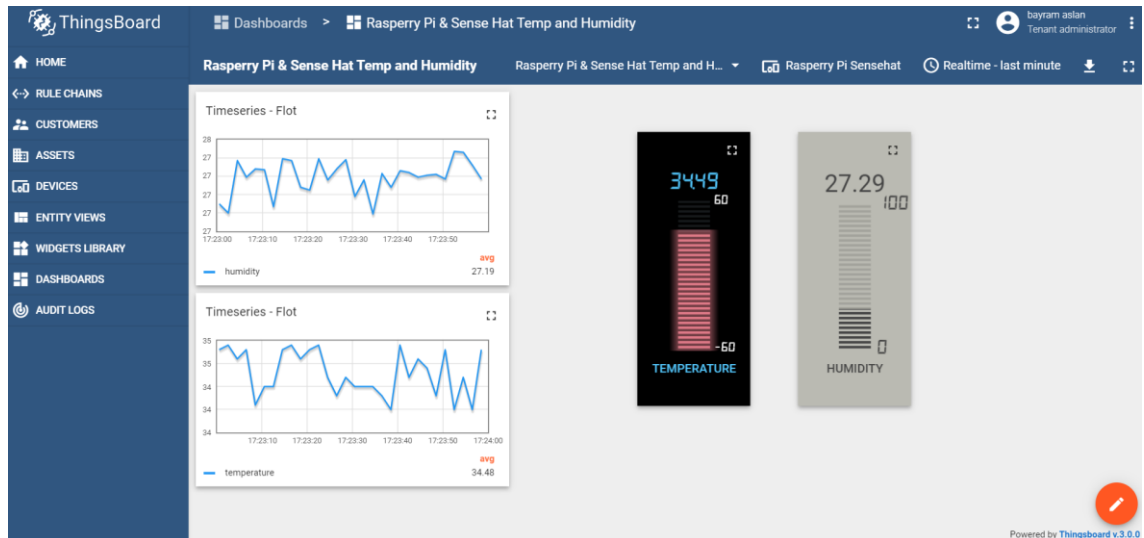
Thingsboard IoT -alustalle luotiin ensin laite, jonka token-koodi tarvittiin laitteen todennusprosessiin. Lämpötila- ja kosteusarvot tulevat tälle luodulle laitteelle.

The screenshot shows the ThingsBoard IoT platform interface. On the left is a navigation menu with options like HOME, RULE CHAINS, CUSTOMERS, ASSETS, DEVICES, ENTITY VIEWS, WIDGETS LIBRARY, DASHBOARDS, and AUDIT LOGS. The main area is titled 'RASPERRY PI SENSEHAT' and shows 'Device details'. Below this, there is a table for 'Latest telemetry' with the following data:

Last update time	Key	Value
2020-05-21 19:03:30	humidity	25.42
2020-05-21 19:03:30	temperature	35.74

Kuva 15. Laitteen hallintasivu.

Sen jälkeen lämpötila- ja kosteusarvot lähetettiin MQTT-protokollalla IoT-alustalle. Tähän tarvittiin SenseHat-, JSON- ja MQTT-python-kirjastoja. Tietokoneen pyörittämässä koodissa löytyy silmukka, joka lähettää kahden sekunnin välein arvoja IoT-alustalle sekä SenseHat-laajennuskortissa löytyvälle 64 lediselle -näytölle. Kosteus- ja lämpöarvot visualisoitiin eri vimpaimilla IoT-alustan kojelaudassa, kuten kuvassa 16. Lisäksi SenseHatin näytössä näkyvät kosteusarvot punaisella ja lämpötila-arvot vihreällä värillä, kuten kuvassa 14.



Kuva 16. Lämpötila- ja kosteusarvot Thingsboard IoT -alustan kojelaudassa.

```

pi@raspberrypi: ~
pi@raspberrypi:~$ python mgtt-raspisensehatv2.py
Temperature: 24.77°C, Humidity: 47.72%
Temperature: 24.83°C, Humidity: 47.77%
Temperature: 24.84°C, Humidity: 47.58%
Temperature: 24.94°C, Humidity: 48.62%
Temperature: 24.77°C, Humidity: 49.22%
Temperature: 24.9°C, Humidity: 49.34%
Temperature: 25.03°C, Humidity: 48.94%
Temperature: 24.95°C, Humidity: 48.51%
Temperature: 24.97°C, Humidity: 48.35%
Temperature: 24.97°C, Humidity: 48.22%
Temperature: 25.12°C, Humidity: 47.83%
Temperature: 25.05°C, Humidity: 48.22%
Temperature: 25.12°C, Humidity: 47.04%
Temperature: 25.18°C, Humidity: 47.52%
Temperature: 25.12°C, Humidity: 47.76%
Temperature: 25.19°C, Humidity: 47.62%
Temperature: 25.25°C, Humidity: 47.49%
Temperature: 25.29°C, Humidity: 47.53%
Temperature: 25.34°C, Humidity: 48.41%
Temperature: 25.36°C, Humidity: 48.74%
Temperature: 25.38°C, Humidity: 47.69%
Temperature: 25.4°C, Humidity: 47.15%
Temperature: 25.4°C, Humidity: 47.29%

```

Kuva 17. Python-koodin käynnistys terminaalissa.

## 8 Yhteenveto

Insinööriyössä tutkin IoT:n , IoT-alustojen osa-alueita sekä avoimen lähdekoodin IoT-alustojen ominaisuuksia . Lisäksi tavoitteena oli rakentaa yksi demo, jolla pyritään lähettämään huoneen lämpötila- ja kosteusarvoja IoT-alustalle visualisoitavaksi. IoT-demon toteuttamiseen valittiin Thingsboard IoT -alusta. Tämän työn yhtenä tavoitteena oleva IoT-demon rakentaminen onnistui. Siinä lopputuloksena saatiin odotusten mukaisesti lämpötila- ja kosteusarvot visualisoituna IoT-alustalle sekä SenseHat-näytölle.

Tutkimuksen aikana selvisi, että avoimen lähdekoodin IoT-alustojen ominaisuuksien välillä saattaa olla suuria eroja. Yllättävänä asiana selvisi, ettei ole yhtenäistä arkkitehtuurimallia IoT-tekniikassa. Kaikki työssä tutkitut avoimen lähdteen IoT-alustat olivat ilmaisia, kuitenkin rajoitetuilla ominaisuuksilla. Alustat tarjoavat maksullisia konsulttipalveluja. Yritysten täytyy IoT-alusta valinnassa huomioida alustan lopullista käyttötarkoitusta.

Insinööriyön aikana selvisi paljon uutta tietoa aiheesta. Tämä tutkimus mahdollisti tutustua markkinoilla oleviin avoimen lähdekoodin IoT-alustoihin sekä kartoittamaan erilaisia vaihtoehtoja aiheesta. Työn laajuudesta johtuen kaikkiin työn aiheisiin ei voitu syventyä. Avoimen lähdekoodin IoT-alustojen tutkittavaksi valittaessa otettiin huomioon käyttöönoton helppous- sekä yksinkertaisuuskriteerit. Lisäksi oppimateriaalien saatavuus oli tärkeä valintakriteeri avoimen lähdekoodin IoT-alustan valinnassa.

Insinööriyön aikana minulle selvisi IoT-tekniikan laajempi kokonaisuus. IoT-tekniikassa on laajoja erilaisia osa-alueita. Tästä syystä on tarpeen määrittää yksilöidyn osa-alueen jatkotoimenpiteitä varten ja syventyä tarkemmin. Tämän työn laajuudesta johtuen käsiteltiin teoriaosuus IoT-aiheista yleistasolla. Jatkokehitysaikana voisi olla IoT-demon toteutus laajennetuilla toiminnoilla, kuten kaksisuuntainen kommunikaatio sekä toteutus muillakin IoT-alustoilla.



## Lähteet

- 1 Esineiden internet. Verkkoaineisto. <<http://www.logistiikanmaailma.fi/logistiikka/digitalisaatio/esineiden-internet/>>. Luettu 3.4.2020.
- 2 Morgan J. 2014. Verkkoaineisto. IOT <<https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#778fc1101d09>>. Luettu 10.4.2020.
- 3 M2M ve IoT Arasındaki Fark. Verkkoaineisto. Key of change. <<https://www.keyofchange.com/tr/1236/M2M%20ve%20IoT%20Aras%C4%B1ndaki%20Fark%20Nedir%3F/>>. Luettu 4.4.2020.
- 4 Esineiden internet. Verkkoaineisto. <[https://fi.wikiversity.org/wiki/Esineiden\\_internet](https://fi.wikiversity.org/wiki/Esineiden_internet)>. Luettu 6.4.2020.
- 5 Pallavi Sethi and Smruti R. Sarangi. 2017. Verkkoaineisto. Internet of Things: Architectures, Protocols, and Applications. <<https://www.hindawi.com/journals/jece/2017/9324035/>>. Luettu 5.4.2020.
- 6 Stokes P. 2018. Verkkoaineisto. 4 Stages of IoT architecture explained in simple words. <<https://medium.com/datadriveninvestor/4-stages-of-iot-architecture-explained-in-simple-words-b2ea8b4f777f>>. Luettu 5.4.2020.
- 7 Calum McClelland, What is an IoT Platform? 2020. <<https://www.ietfoll.com/what-is-an-iot-platform/>> Luettu 4.4.2020.
- 8 What is an Internet of Things Platform? 2020. Verkkoaineisto. <<https://www.avsystem.com/blog/what-is-internet-of-things-platform/>>. Luettu 4.4.2020.
- 9 Pdraig S. 2016. <<https://iot-analytics.com/5-things-know-about-iot-platform/>>. Luettu 15.4.2020.
- 10 Elizalde D. Verkkoaineisto. <<https://danielelizalde.com/iot-platform/>>. Luettu 15.4.2020.
- 11 Büyüktanir, T ja Özer, B. 2017. Yeni Baslayanlar Icin Nesnelerin Interneti (IoT): Internet Of Things.
- 12 How to Make Internet of Things Solutions Secure by Design, Verkkoaineisto. <https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/iot-security/key-principles>. Luettu 28.3.2021.

- 13 Avoin lähdekoodi. Verkkoaineisto. <<https://coss.fi/avoimuus/avoin-lahdekoodi/>>. Luettu 5.4.2020.
- 14 Avoin Lähdekoodi. Verkkoaineisto. <[https://www.linux.fi/wiki/Avoin\\_l%C3%A4hdekoodi](https://www.linux.fi/wiki/Avoin_l%C3%A4hdekoodi)>. Luettu 5.4.2020.
- 15 Open Source. Verkkoaineisto. <<https://www.itewiki.fi/opus/open-source/>>. Luettu 5.4.2020.
- 16 Thinger console, Verkkoaineisto. <<https://docs.thinger.io/console>>. Luettu 25.4.2020.
- 17 Freeboard näkymä. Verkkoaineisto. <<https://github.com/Freeboard/freeboard/blob/master/README.md>>. Luettu 15.5.2020.
- 18 Thingsboard monolithic. Verkkoaineisto. <<https://thingsboard.io/docs/reference/monolithic/>>. Luettu 25.4.2020.
- 19 Thingsboard mikropalveluarkkitehtuuridiagrammi. Verkkoaineisto. <<https://thingsboard.io/docs/reference/msa/>>. Luettu 30.4.2020.
- 20 Thingsboard hallintanäkymä. Verkkoaineisto. <<https://thingsboard.io/docs/user-guide/ui/devices/>>. Luettu 10.5.2020.
- 21 DeviceHive arkkitehtuuri. Verkkoaineisto. <<https://docs.devicehive.com/docs/devicehive-architecture>>. Luettu 25.5.2020.
- 22 DeviceHive hallintapaneeli. Verkkoaineisto. <<https://blog.devicehive.com/2018/07/02/blockchain-devicehive-plugin/>>. Luettu 30.5.2020.

## Lämpötila- ja kosteusarvojen lähetys IoT-alustalle ja SenseHat-näytölle.

```
import time
import paho.mqtt.client as mqtt
import json
from sense_hat import SenseHat
from twoDigit.twoDigit import numToMatrix
from time import sleep

# autentikointi
THINGSBOARD_HOST = 'demo.thingsboard.io'
ACCESS_TOKEN = 'peawdP9Y3ttGFkoDlSqR'

# Tiedonkeruu- ja lahetysväli sekunneissa
INTERVAL=2

#sensehatin käynnistys
sh = SenseHat()

sensor_data = {'temperature': 0, 'humidity': 0}

next_reading = time.time()

client = mqtt.Client()

# token koodin asettaminen
client.username_pw_set(ACCESS_TOKEN)

client.connect(THINGSBOARD_HOST, 1883, 60)

client.loop_start()

try:
    while True:
        humidity = sh.get_humidity()
        temperature = sh.get_temperature()
        humidity = round(humidity, 2)
        temperature = round(temperature, 2)
        print(u"Temperature: {:g}\u00b0C, Humidity: {:g}%".format(temperature,
humidity))
        sensor_data['temperature'] = temperature
        sensor_data['humidity'] = humidity
        red = [255,0,0] # punainen
        green = [0,255,0] # vihrea
        h = int(round(humidity,0))
        t = int(round(temperature,0))
        i = numToMatrix(h,red)
        ii = numToMatrix(t,green)
        sh.set_pixels(i) # Display reading
        sleep(0.5)
        sh.set_pixels(ii)
        sleep(0.5)

# tietojen lahetys Thingboardille
client.publish('v1/devices/me/telemetry', json.dumps(sensor_data), 1)

next_reading += INTERVAL
```

```
        sleep_time = next_reading-time.time()
        if sleep_time > 0:
            time.sleep(sleep_time)
    except KeyboardInterrupt:
        pass

    client.loop_stop()
    client.disconnect()
```