

Pyry Leppälä

**TEOLLISEN INTERNETIN ALUSTAN VAATIMUSMÄÄRITTELY
JA VALINTA JOUSTAVAN VALMISTUSJÄRJESTELMÄN
DIGITALISOINNISSA**

Tekniikan ja luonnontieteiden tiedekunta
Diplomityö
Tammikuu 2019

TIIVISTELMÄ

Pyry Leppälä: Teollisen internetin alustan vaatimusmäärittely ja valinta joustavan valmistusjärjestelmän digitalisoinnissa
Diplomityö
Tampereen yliopisto
Konetekniikan DI-tutkinto-ohjelma
Tammikuu 2019

Digitalisaatio muuttaa maailmaa, teollisuutta ja yrityksiä eikä sen vaikutuksilta voi välttyä. Yrityksien on toimittava selviytyäkseen murroksesta hyödyntäen asioiden ja teollisen internetin tuomat uudet liiketoimintamahdollisuudet. Yksi konkreettinen tapa on kerätä ja hyödyntää erilaista tietoa yrityksen tuotteista sekä toimintaympäristöstä ja jalostaa siitä hyödyllistä informaatiota, jonka avulla voidaan ohjata toimintaa tai kehittää esimerkiksi uudenlaisia palveluita eri sidosryhmille.

Joustavat valmistusjärjestelmät ovat automatisoineet valmistavaa teollisuutta jo vuosikymmenien ajan, mutta perinteisesti ne ovat olleet eristettyinä tehtaiden sisälle ilman kunnollisia verkoyhteyksiä. Teollisen internetin ideologian mukaisesti FMS-järjestelmätkin (Flexible Manufacturing System) tulevat verkottumaan ja yhdistymään. Näin FMS-järjestelmien tuottama, huomattava tietomäärä saadaan hyödynnettäväksi järjestelmän ulkopuolella.

Tässä työssä tutkitaan, miten joustavan valmistusjärjestelmän digitalisointia varten valitaan teollisuusyrityksen tarpeisiin sopivin IoT-alusta (Internet of Things) laajasta kaupallisesta tarjonnasta ja mitä asioita kannattaa ottaa huomioon prosessin eri vaiheissa. Työssä kehitetään V-mallia soveltava valintaprosessi vaatimusmäärittelyn tekemiseen yrityksen tavoitteista, vertailun suunnitteluun ja toteutukseen, parhaan vaihtoehdon analysointiin ja valintaan. Kehityksessä valintaprosessissa myös varmistutaan valinnan sopivuudesta validoimalla V-mallin mukaisesti valittua IoT-alustaa yrityksen vaatimuksia ja tarpeita vasten.

Valintaprosessissa sovelletaan ohjelmistotuotannossa yleisesti käytettyjä menetelmiä esimerkiksi vaatimusmäärittelyn tekemiseen käyttäjätarinoiden muodossa sekä validointivaiheessa vaatimuksien ja tarpeiden täyttymisen varmistamiseksi testitapauksien muodossa. Alustojen kartoittamiseen ja vertailukriteerien mukaisen tiedon keräämiseen tiedonhaun, koekäyttöjen, ja referenssikeskusteluiden avulla käytetään vertailutaulukkoa, johon täyttämisen jälkeen sovelletaan Pughin matriisia keskinäisen vertailun mahdollistamiseksi. Työssä kehitettyä prosessia käytetään IoT-alustan valintaan kohdeyrityksen käyttöön, ottaen huomioon muun muassa FMS-järjestelmien käyttämän teollisuuden hitaan muutosnopeuden ja tehtaiden puutteelliset tietoliikenneyhteydet.

Siitä huolimatta, että työ keskittyy joustavien valmistusjärjestelmien erityistarpeisiin, on kehitetty prosessi sovellettavissa myös muunlaisiin korkean teknologian tuotteisiin, kunhan erityisvaatimukset selvitetään kyseisen laitteiston näkökulmasta. Prosessia soveltaessa pitää ottaa huomioon, että IoT-alustavaihtoehdot on kartoitettava jokaisella kerralla uudelleen, sillä IoT-alustojen nopeasti muuttuva ja kehittyvä tarjonta aiheuttaa tiedon vanhenemisen hyvin nopeasti.

Avainsanat: digitalisaatio, teollinen internet, FMS, joustava valmistusjärjestelmä, IoT-alusta

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ABSTRACT

Pyry Leppälä: Requirements definition and selection of industrial internet platform for digitalizing flexible manufacturing system
Master of Science Thesis
Tampere University
Master's Degree Program in Mechanical Engineering, MSc (Tech)
January 2019

Digitalization is changing the world, industries and companies and nobody is not safe from its effects. Companies must act to survive from disruption by making use of new business opportunities created with internet of things and industrial internet. One concrete way to is to collect and utilize different data coming from company's products by refining information from data. This data can be then be used to lead the actions or create new kind of services to stakeholders.

Flexible manufacturing systems have been automating manufacturing industry for many decades but those have been isolated in the factories without proper network connections. With industrial internet's paradigm, FMS systems will become networked and connected. This way data the considerable amount of data produced by FMS system can be utilized outside of the system.

In this thesis the subject of research is how industrial company can select the most suitable IoT platform for digitalizing flexible manufacturing system from wide selection of commercial offering and what kind of details need to be taken into account during different phases of the process. As a result of this thesis, a selection process is developed which uses the V-model for doing the requirement engineering starting from company's goals, planning and executing the comparison and for analyzing the results and selecting the best option. Also, part of the developed process is to validate the chosen platform according the V-model against requirements and goals of the company.

During the selection process, commonly used methods from the computing science are used to do the requirement engineering by utilizing user stories and to validate the choice using test cases to confirm that the choice is filling all company's goals and requirements. During the survey phase, gathered information from different sources, trial runs and reference discussions were collected to comparison table. After the table is filled, all individual candidates are being compared to each other using Pugh matrix. In this thesis, developed process is being used to select IoT platform for company to which thesis is made, taking account for example slow changing speed of the industries who is using FMS systems and absent of networking connection in the factories.

Despite of fact that this thesis is focusing special requirements of the flexible manufacturing systems, developed process can be utilized for other kind of products of high technology when special requirements are researched in point of view of subject. When utilizing the process, the IoT platform candidates need to be surveyed every time as rapidly changing and developing offering of the IoT platforms is causing that the information becomes outdated very quickly.

Keywords: digitalization, industrial internet, FMS, flexible manufacturing system, IoT platform

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Digitalisaatio ja asioiden internet on tullut jäädäkseen maailmaan, mutta vuonna 2015 kun teollisuus alkoi vasta heräämään tähän murrokseen, sain loistavan mahdollisuuden osallistua tähän kehitykseen rakentamalla Fastems Oy:lle perustukset IoT-aikaan tämän diplomityön muodossa. Tämän työn puitteissa tehty kehitysprojekti valmistui jo 2016 mutta työ valmistui vihdoin vuonna 2019.

Mutta ei Roomaakaan rakennettu päivässä ja nyt jälkeempäin, kun katsoo 13 vuotta kestäneitä opintoja, ei harmita lainkaan. Näihin vuosiin on mahtunut niin paljon opiskelijaelämän iloa sekä mielenkiintoisia opintoja, joista on ollut hyötyä digitalisaation aikakaudella, vaikka sellaisesta ei vielä osattu unelmoida opiskelua aloitellessa. Tästä syystä haluankin ensimmäisenä kiittää opintosuunnittelija Nina Ojalaa, joka on jaksanut kaikki nämä vuodet auttaa minua hyvin henkilökohtaisen opintokokonaisuuden sekä yliopiston byrokratian kanssa mahdollistaen viimein valmistumiseni.

Kiitokset työni tarkastajille Tero Juutille ja Timo Lehtoselle sekä Janne Kiviselle Fastemsilta työn ohjaamisesta sekä hyvästä tuesta työtä tehdessä. Haluan vielä kiittää kaikkia ystäviäni ja työkavereita kannustuksesta saattaa opinnot loppuun sekä erityiskiitokset Heini Pylkköselle sekä Vesa Vahvaselle henkisestä tuesta.

Tampereella, 31.1.2019

Pyry Leppälä

SISÄLLYSLUETTELO

1.	JOHDANTO	1
1.1	Työn tausta	1
1.2	Työn tavoitteet, tutkimuskysymykset ja rajaukset	2
1.3	Työn rakenne.....	3
2.	TUTKIMUSSTRATEGIAT JA -MENETELMÄT	5
2.1	Käytetyt tutkimusstrategiat ja -menetelmät.....	5
2.2	V-malli prosessin hallinnassa.....	6
2.3	V-mallin soveltaminen valintaprosessiin	7
2.4	Vaatimusmäärittely	10
2.5	Pughin matriisi	11
3.	VAATIMUSMÄÄRITTELYPROSESSI	13
3.1	Tavoitteet.....	13
3.1.1	Digitalisaatio ja teollinen internet.....	13
3.1.2	Liiketoiminnan tavoitteet	14
3.2	Vaatimuksien muodostus	15
3.2.1	Tarpeiden tunnistus	15
3.2.2	Tarpeiden jalostus vaatimuksiksi	16
3.3	Vaatimukset.....	22
3.3.1	Liiketoiminnan vaatimukset.....	22
3.3.2	Joustavan valmistusjärjestelmän erityisvaatimukset.....	23
3.3.3	Tekniset vaatimukset.....	25
4.	IOT-ALUSTAT	30
4.1	IoT-alustan määritelmä	30
4.2	Ominaisuudet ja erityispiirteet	30
4.3	IoT-alustat teollisessa ympäristössä.....	33
5.	IOT-ALUSTOJEN VERTAILU	36
5.1	Vertailukriteerien muodostus	36
5.2	Alustojen vertailuprosessi	39
5.3	Vaihtoehtojen kartoitus ja esivalinta.....	41
5.4	Esivalitut vaihtoehdot.....	41
5.4.1	PTC Thingworx.....	41
5.4.2	Cumulocity.....	44
5.4.3	Wapice IoT-Ticket.....	46
5.4.4	Microsoft Azure	49
5.4.5	IBM Bluemix	51
5.4.6	Amazon AWS	53
5.4.7	2lemetry.....	55
5.4.8	ELK Stack.....	57
5.4.9	Ixonos Industrial Internet Suite.....	59
5.4.10	Tieto I&I Connect	61

5.4.11	Keskustekniikka Cosmos	62
6.	VERTAILUN TULOKSET	65
6.1	Vertailutaulukko.....	65
6.2	Pughin matriisi	67
6.3	Muut tulokset	70
7.	JOHTOPÄÄTÖKSET	71
7.1	Löydökset vertailun tuloksista	71
7.2	Vaihtoehtojen karsinta	73
8.	IOT-ALUSTAN VALINTA JA VALIDOINTI	76
8.1	Valittu IoT-alusta	76
8.2	Valinnan validointi.....	77
8.2.1	Validointiprosessi.....	77
8.2.2	Validointi teknisiä vaatimuksia vasten	78
8.2.3	Validointi liiketoiminnan tavoitteita vasten.....	79
9.	VALINTAPROSESSIN JA TUTKIMUKSEN ARVIOINTI.....	81
9.1	Valintaprosessin arviointi.....	81
9.2	Tutkimuksen arviointi	81
10.	YHTEENVETO	84
	LÄHTEET.....	86

LIITE A: TYÖN RAKENNEKAAVIO

LIITE B: IOT-ALUSTOJEN VERTAILUTAULUKKO

LIITE C: PUGHIN VERTAILUMATRIISI IOT-ALUSTOISTA

KUVALUETTELO

Kuva 1.	<i>Työn läpiviennissä käytetty rakennekaavio</i>	4
Kuva 2.	<i>Perinteinen vesiputous-kehitysmalli (Sage & Rouse 2009)</i>	7
Kuva 3.	<i>Yksinkertaistettu v-mallin kaaviokuva (Osborne et al. 2005)</i>	7
Kuva 4.	<i>Työssä sovelletun V-mallin kaaviokuva</i>	9
Kuva 5.	<i>Taulukkomuotoinen käyttäjätarina hissien hätätilanteen ratkaisusta (Kujala et al. 2001)</i>	16
Kuva 6.	<i>Kaavio teknisten vaatimusten muodostuksesta ja riippuvuussuhteista</i>	26
Kuva 7.	<i>Hype-käyrän perusrakenne (Gartner n.d.)</i>	28
Kuva 8.	<i>Hype-käyrä yleisien IoT-tekniikkatekniikkien maturiteettien arviointiin (Gartner, 2015)</i>	29
Kuva 9.	<i>IoT-sovelluksen toiminnallisuus- ja teknologiavaatimuksia pinomuodossa (Porter & Heppelmann 2014)</i>	31
Kuva 10.	<i>Esimerkki systeemien systeemistä (Porter & Heppelmann 2014)</i>	33
Kuva 11.	<i>Kuluttajille tarkoitettujen asioiden internetin sekä yritystason teollisen internetin vertailua (Perry 2016)</i>	34
Kuva 12.	<i>Teknisten vaatimusten ja IoT-alustojen vertailukriteereiden riippuvuussuhteet</i>	37
Kuva 13.	<i>Täytetty vertailutaulukko alustavaihtoehdoista</i>	66
Kuva 14.	<i>Täytetty Pughin matriisi vertailuvaihtoehdoista</i>	67
Kuva 15.	<i>Pughin vertailun tulokset kaaviona</i>	69

LYHENTEET JA MERKINNÄT

FMS	Flexible Manufacturing System
IoT	Internet of Things
IIoT	Industrial Internet of Things
PDD	Property Driven Development
ERP	Enterprise Resource Planning
CRM	Customer Relationship Management
CNC	Computer Numerical Control
SDK	Software Development Kit
MQTT	Message Queuing Telemetry Transport
CoAP	Constrained Application Protocol
OPC	Open Platform Communications
HTTP	Hypertext Transfer Protocol
PoC	Proof of Concept
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
WLAN	Wireless Local Area Network

1. JOHDANTO

1.1 Työn tausta

Tämä diplomityö on tehty Fastems Oy:lle osana yrityksen digitalisaatiostrategian jalkauttamisprojektia. Fastems on suomalainen joustavia valmistusjärjestelmiä (Flexible Manufacturing System, FMS) tuottava yritys, jonka uudistetussa strategiassa tärkeänä osana on digitalisaatio ja sen hyödyntäminen uusien liiketoimintamahdollisuuksien avulla. Uudessa strategiassaan Fastems linjaa valmistuksen arvoketjujen digitalisointia ja halua olla edelläkävijä valmistavan teollisuuden digitalisaatiossa. Myöskin markkinat ovat kehittyneet siten, että asiakkaat osaavat jo koneteollisuudessakin vaatia etuja, joita digitalisaatio mahdollistaa, esimerkiksi uudenlaisien palveluiden muodossa.

Fastemsin aikaisemmassakin toiminnassa on jo paljon digitalisaation piirteitä, sillä tuotantojärjestelmien automaatioaste on jo valmiiksi todella korkea robotiikan ja kehittyneiden sovelluksien sekä teknologioiden hyödyntämisen ansiosta. Fastemsin automaatiojärjestelmät muun muassa kykenevät autonomiseen toimintaan, automaattiseen tuotannon optimointiin esimerkiksi tilauksien kriittisyyden mukaan tai reagointiin joustavasti esimerkiksi laiterikkojen sattuessa, ohjaten työt muille kykeneville laitteille. Samalla itse fyysisestä järjestelmästä ja sen tekemästä tuotantoprosessista kertyy todella paljon erilaista tietoa, mitä ei kuitenkaan käytetä hyväksi, muuten kuin mahdollisten ongelmien selvityksen apuna. Tietoa ei kuitenkaan systemaattisesti käsitelty tai viety järjestelmän ulkopuolelle, vaan tieto jäi paikallisesti järjestelmään, mistä se poistettiin määrätyn ajan kuluttua levytilan säästämiseksi. Tämä menetetty tieto on hyvin arvokasta digitaalisten palveluiden näkökulmasta, mihin tehdyillä toimenpiteillä pyritään tekemään muutos siten, että tietoa pystyttäisiin hyödyntämään mahdollisimman tehokkaasti.

Suunnitellun strategian toteuttaminen vaatii muutoksia ja kehitystä niin liiketoimintapuolella kuin teknisellä puolella, sillä valmiuksia digitalisaation hyödyntämiseen ei yrityksessä aikaisemmin ole vielä ollut. Tämä työ keskittyy strategian toteuttamisen teknisiin mahdollistajiin. Aikaisemmin digitalisaatioprojektissa oli päätetty valita sovelluskehityksen malli, jossa tekniset kyvykkyydet tarjoava ratkaisu hankintaan ulkopuoliselta toimijalta mutta toimintalogiikka ja sovellukset tehtäisiin itse omia resursseja hyödyntäen. Ratkaisuvaihtojen selvityksen tuloksena päädyttiin niin sanotun puolivalmiin IoT-alustan (Internet of Things) hankintaan.

1.2 Työn tavoitteet, tutkimuskysymykset ja rajaukset

Digitalisaation tuoman kysynnän myötä markkinoilla olevien IoT-alustojen tarjonta on kasvanut huomattavasti viimeisten vuosien aikana. Tämän työn tavoitteena on tutkia, miten yritykselle kannattaa valita IoT-alusta ja mitä asioita kannatta valintaa tehdessä ottaa huomioon. Yrityksen tavoitteena tälle työlle oli valintaprosessin määrittelyn lisäksi kartoittaa alustatarjontaa sekä löytää ja valita Fastemsille soveltuva IoT-alusta. Alustan tuli olla joustava ja mahdollistaa jatkossa monipuolinen sovellus- ja palvelukehitys ottaen huomioon joustavan valmistusjärjestelmän erityispiirteet. Näistä tavoitteista muodostettiin seuraavat, tutkimusta ohjaavat tutkimuskysymykset.

- Miten valitaan mahdollisimman hyvä IoT-alusta FMS-järjestelmän digitalisoinnin työkaluksi?
- Miten IoT-alustan vaatimusmäärittely, vertailu ja valitun ratkaisun validointi voidaan toteuttaa?
- Mitä erityisvaatimuksia joustava valmistusjärjestelmä tuottaa valintaprosessiin?

Fastemsin uudessa strategiassa ei määritelty menetelmiä, joilla digitalisaatiota pitäisi toteuttaa, joten työssä on tarkoitus tutkia, miten määritellään liiketoiminnan tavoitteiden avulla vaatimukset alustalle ja muodostetaan näistä teknisiä vaatimuksia. Tutkimuksen kohteena onkin, kuinka vaatimusmäärittelyprosessi kannattaa toteuttaa ja miten liiketoiminnan tavoitteet muutetaan teknisiksi vaatimuksiksi. Selvitettävänä on myös, miten joustavan valmistusjärjestelmän digitalisointi eroaa perinteisestä IoT:n tai teollisen internetin määritelmästä ja tuoko se mukanaan jotain erityisvaatimuksia.

Yksi tutkimusaihe on myös IoT-alustojen vaihtoehtojen kartoitus. Millä menetelmillä kaupallista tarjontaa kannattaisi tutkia, että löydettäisiin tehokkaasti parhaat vaihtoehdot vertailua varten suuresta määrästä tarjontaa. Kun vaihtoehdot on löydetty, vertailu halutaan pystyä suorittamaan siten, että vertailun tulokset olisivat merkitseviä ja joiden avulla valinta pystytään tekemään.

Kun sopiva IoT-alusta on valittu, halutaan varmistua valinnan sopivuudesta ja että se täyttää määritellyt vaatimukset sekä mahdollistaa tavoitteiden saavuttamisen. Valintaa halutaan siis validoida vaatimuksia ja tavoitteita vasten. Sopivaa menetelmää validoinnin suorittamiseen tutkitaan myös työn aikana.

Työn ulkopuolelle on rajattu alustakustannuksien vertailua lukuun ottamatta kaikki liiketoiminnan taloudelliset tai tuotepoliittiset asiat, kuten liiketoimintamallien tai uusien tuotteiden tai palveluiden käsittely. Myöskin valitun ratkaisun käyttöönotto ja verifiointi on rajattu tämän työn ulkopuolelle.

1.3 Työn rakenne

Luvussa 1 esitellään työn taustoja, tavoitteita sekä tutkimuskysymykset. Tässä kappaleessa myös esitellään työn rajaukset ja tämän raportin sisältö sekä rakenne.

Luvussa 2 määritellään käytetyt tutkimusstrategiat, joiden avulla tutkimus tullaan toteuttamaan sekä tutkimusmenetelmät, joita käyttämällä tutkimuksen aineisto kerätään. Myös tutkimuksen sekä suunniteltavan prosessin pohjalla käytetty V-malli esitellään tässä luvussa. Vaatimusmäärittelyssä ja vertailussa käytettyjen menetelmien teoria on myös esitetty tässä luvussa.

Luvussa 3 käydään läpi vaatimusmäärittelyprosessi, minkä tuloksena saadaan vertailussa käytettävät tekniset vaatimukset. Vaatimusten muodostamiseen käytetään yrityksen liiketoiminnan sekä yleisesti digitalisaation sekä teollisen internetin tavoitteita.

Luvussa 4 esitellään IoT-alustat, niiden ominaisuudet ja erityispiirteet. Luvussa käsitellään myös IoT-alustojen käyttöä teollisessa ympäristössä.

Luvussa 5 muodostetaan teknisistä vaatimuksista vertailukriteereitä, joiden avulla kasataan IoT-alustoiden vertailuun käytetty aineisto. Tässä kappaleessa esitellään käytetty vertailuprosessi ja esitellään vertailuun valitut vaihtoehdot.

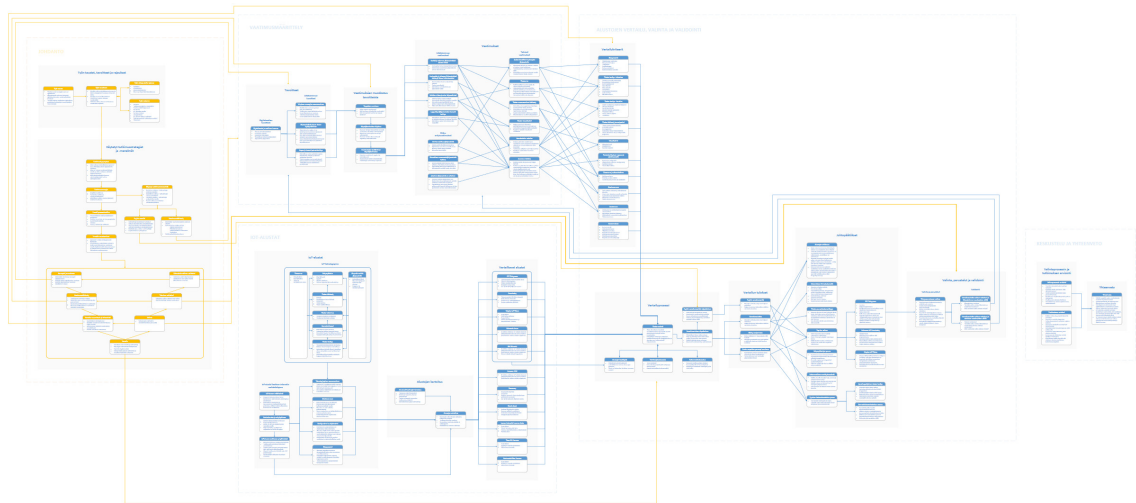
Vertailun tuloksia käsitellään luvussa 6, missä esitellään tuloksena saatu IoT-alustojen vertailutaulukko sekä alustojen keskinäinen vertailu Pughin matriisin avulla. Myös muut vertailun ohella saadut tulokset ovat esiteltyinä tässä luvussa. Luvussa 7 analysoidaan vertailun tuloksia ja esitellään tuloksista saadut johtopäätökset. Tässä luvussa myös suoritetaan osana valintaprosessia oleva vaihtoehtojen karsinta.

Luvussa 8 tehdään vertailun perusteella parhaan vaihtoehdon valinta ja suoritetaan validointiprosessi teknisiä vaatimuksia ja liiketoiminnan tavoitteita vasten. Näin saadaan varmistus siitä, että lopputuloksena valittu valinta on oikea.

Kappaleessa 9 arvioidaan kehitettyä valintaprosessia sekä sen kehitykseen tehtyä tutkimusta. Tässä kappaleessa arvioidaan prosessin toimivuutta sekä tutkimuksen onnistumista, luotettavuutta sekä merkitystä.

Työstä laadittiin kaavio, jolla visualisoitiin tutkimussuunnitelmaa, teoriaa ja itse tutkimusta. Kaaviossa edetään työn rakenteen mukaisesti vuokaaviota mukailevassa muodossa, missä asiat ovat ryhmitelty yhteen ja liitetty loogisesti toisiinsa. Kaavio on nähtävissä kuvassa 1 sekä suurempana työn liitteessä A. Kaaviossa työn teoriaosuudet ja niiden liitokset ovat kuvattu keltaisen värisillä laatikoilla sekä viivoilla. Ylätason aihekokonaisuudet ovat esitetty katkoviivalla merkityin laatikoin ja tarkemmat aiheet ovat koottu harmaalla taustavärillä merkityin laatikoiden avulla. Kaaviolla on myös hahmoteltu tämän

työn raportin rakennetta eli käsitellyt asiat etenevät ajallisesti niin tekemisen kuin raportin sisällön kannalta.



Kuva 1. Työn läpiviennissä käytetty rakennekaavio

Kaaviota päivitettiin prosessin aikana ja lopulta kaavion osioista koostettiin tämän raportin rakenne ja asiakokonaisuudet. Kaavio oli hyödyllinen työkalu, jonka avulla kokonaisuuden hallinta oli helppoa ja visuaalisuus helpotti näkemään, miten ja miksi työ etenee osiosta toiseen.

2. TUTKIMUSSTRATEGIAT JA -MENETELMÄT

2.1 Käytetyt tutkimusstrategiat ja -menetelmät

Työssä keskitytään empiirisen tutkimuksen puolelle teoreettisen tutkimuksen sijasta, sillä tutkimuksen pääpaino on etsiä ja löytää paras vaihtoehto jo olemassa olevien vaihtoehtojen tarjonnasta. Teoreettista tutkimusta tehdään enemmän vaatimusmäärittelyprosessissa, missä vaatimuksien muodostamista tutkitaan teorian kautta. Tutkimusstrategiaksi on valittu määrällinen tutkimus. Määrällisessä tutkimuksessa tutkitaan joidenkin asioiden välisiä suhteita tai eroa ja sille on ominaista tutkittavan ilmiön esittäminen tiedon, esimerkiksi numeroiden avulla sekä havaintoyksiköiden suuri määrä. (Vilka 2007). Siitä huolimatta, että tässä tutkimuksessa vertailtavien vaihtoehtojen lukumäärä ei tule olemaan erityisen korkea, eikä tutkimuksessa hyödynnetä määrälliseen tutkimukselle ominaisia tilastollisia menetelmiä, ei vaihtoehtoisena tutkimusstrategiana harkittu laadullinen tutkimus ei olisi välttämättä olisi toiminut määrällistä tutkimusstrategiaa paremmin. Laadullisessa tutkimuksessa keskitytään enemmän yksilöön ja siihen liittyvien tekijöiden tutkimiseen, mikä pienenkin joukon vertailevassa tutkimuksessa ei ole paras vaihtoehto. Määrällisen tutkimusstrategian toteuttamiseksi tutkimus- ja aineiston keruumenetelminä käytetään kartoitettavaa, vertailevaa ja kokeellista tutkimusta.

Kartoitettavaa tutkimusta hyödynnetään vertailun vaihtoehtojen etsinnässä ja esivalinnassa. Kartoitavassa tutkimuksessa ei lähtökohtaisesti ole käytössä jo olemassa olevaa teoriaa tai mallia. Tämä voi johtua muun muassa aikaisemman tutkimuksen puutteesta tai siitä, että aikaisempi tutkimus poikkeaa kyseisestä tapauksesta liikaa ja siten voisi johtaa tutkimusta harhaan. (Routio, Mallien käyttö tutkimushankkeessa, 2007). Tämän työn tapauksessa aikaisempaa tutkimusta on saatavilla yleisesti IoT-alustojen valinnasta ja vertailusta, mutta tutkimusta ei löytynyt käytännössä lainkaan IoT-alustoista valmistavan koneellisuuden näkökulmasta. Myös IoT-alustojen hyvin nopeasti muuttuva tarjonta sekä nopea kehitys aiheuttavat sen, että aikaisemmissa tutkimuksissa tehdyt johtopäätökset vanhenevat hyvin lyhyessä ajassa. Huomioitavaa on myös, että tämän työn aineisto on kerätty jo vuonna 2015, joten aineisto on työn julkaisun aikaan jo vanhaa tietoa. Kuitenkin tutkimus sekä kehitetty prosessi ei menetä merkitystään ajan saatossa aineiston tavoin.

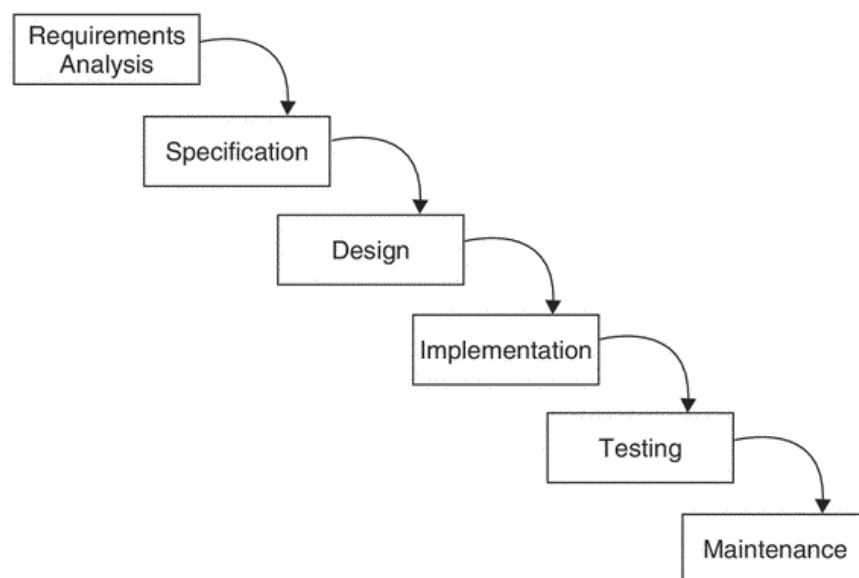
Vaihtoehtojen analysointiin käytetään vertailevaa sekä kokeellista tutkimusta. Vertailevassa tutkimuksessa tarkastellaan samankaltaisia yksilöitä tai tapauksia ja etsitään näistä eroavaisuuksia. Myös aineiston johdonmukaisuutta ja eroavaisuuksien sitä sekä seurauksia voidaan tutkia. (Routio, Vertailu, 2007). Kokeellisessa tutkimuksessa tavoitteena on tutkia yksilöitä kontrolloidussa ympäristössä tai tilanteessa. Tutkimusmenetelmässä pyrkimyksenä on mahdollistaa systemaattisten havaintojen. (Kokeellinen tutkimus, 2015). Tässä työssä IoT-alustoja vertaillaan ominaisuuksiensa suhteen ja tutkitaan alustoissa

esiintyviä eroja sekä samankaltaisuuksia. Myös löydettyjä eroavaisuuksia yritetään selittää, jotta alustojen eroavaisuuksien seuraukset ymmärrettäisiin paremmin. Kokeellista tutkimusta sovelletaan tässä työssä alustojen koekäyttöjen yhteydessä, jolloin etukäteen suunnitellut testitapaukset suoritetaan jokaisella vertailun vaihtoehdolla, keräten kokemukset ja tiedot testien tuloksista aineistoon.

2.2 V-malli prosessinhallinnassa

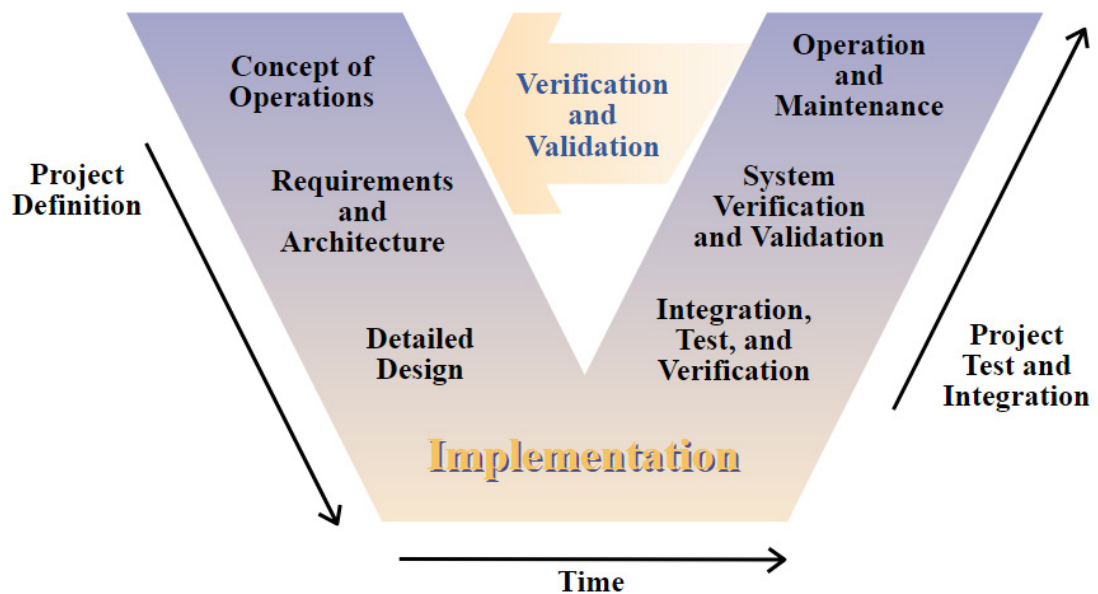
V-malli on yksi monesta systeemisuunnittelun prosessimallista ja on kehitetty erityisesti sovelluskehitysprojektien käyttöön mutta on sovellettavissa myös muun tyyppisiin projekteihin. Alun perin V-malli on kehitetty puolustusteollisuuden projektien käyttöön mutta on myöhemmin laajennettu kattamaan esimerkiksi Saksassa kaikki julkisen sektorin IT-projektit. (Das V-Modell). Kuten muutkin prosessimallit, V-malli ohjaa projektia määrittämällä tarvittavat toimenpiteet sekä askeleet siten, että projektin lopputulos olisi mahdollisimman onnistunut (Sage & Rouse 2009).

V-malli on johdettu perinteisesti käytetystä vesiputousmallista, jossa kaikki normaalissa kehitysprojektissa tarvittavat toimenpiteet ja vaiheet ovat esitetty peräkkäin suoritettavina, toisistaan riippumattomista osista siten, että yhden vaiheen loppuessa aloitetaan seuraava. Kuvassa 2 on esitetty perinteinen ylätasoinen vesiputousmalli missä on nähtävissä systeemisuunnittelun osatehtäviä kuten määrittely, suunnittelu ja testaus. (Sage & Rouse 2009).



Kuva 2. Perinteinen vesiputous-kehitysmalli (Sage & Rouse 2009)

V-mallissa vesiputousmallin tehtävien suoritusjärjestys ja ajankohta muuttuvat siten, että osasta tehtäviä tulee rinnakkaisia ja toisiinsa tiukemmin integroituja. Malli määrittää myös suoran riippuvuussuhteen vaatimuksien ja validoinnin välille (Kossiakoff et al. 2011). Kuvassa 3 on nähtävissä yleistetty v-mallin peruseriaate missä v-kuvion vasemmalla puolella suoritetaan projektin määrittelyyn liittyviä tehtäviä kuten konseptin kehitys, vaatimusmäärittely ja tarkempi suunnittelu. Kuvion pohjalla on itse projektin toteutusvaihe ja oikealla puolella on kuvattuna toteutuksen jälkeen tehtäviä projektin osia kuten testaus, validointi ja verifiointi. Näitä oikealla puolella olevia projektin vaiheita suorittaessa viitataan vasemman puolen määrittelyvaiheen suorituksiin. Esimerkiksi integraatiotestauksen ja verifiointin vaiheessa varmistetaan siitä, että toteutus täyttää tarkemman suunnittelun vaiheessa määritellyt asiat. Vastaavasti validointivaiheessa tuloksia verrataan määrittelyvaiheessa kehitettyihin vaatimuksiin, jolloin tiedetään, että toteutus vastaa sille aikaisemmin määritellyt toiminnalliset vaatimukset. (Sage & Rouse 2009)



Kuva 3. Yksinkertaistettu v-mallin kaaviokuva (Osborne et al. 2005)

V-mallia seurattaessa mitä tahansa vaiheita voidaan iteroida tarvittaessa. Esimerkiksi vaatimuksia voidaan päivittää, jos verifiointivaiheessa käy ilmi, ettei lopputulos täytä sille määriteltyjä vaatimuksia. Tai yksikkötestauksessa huomataan, että tarkempaa suunnitelmaa pitää muuttaa, että lopputulos olisi toimiva. (Buede & Miller 2016).

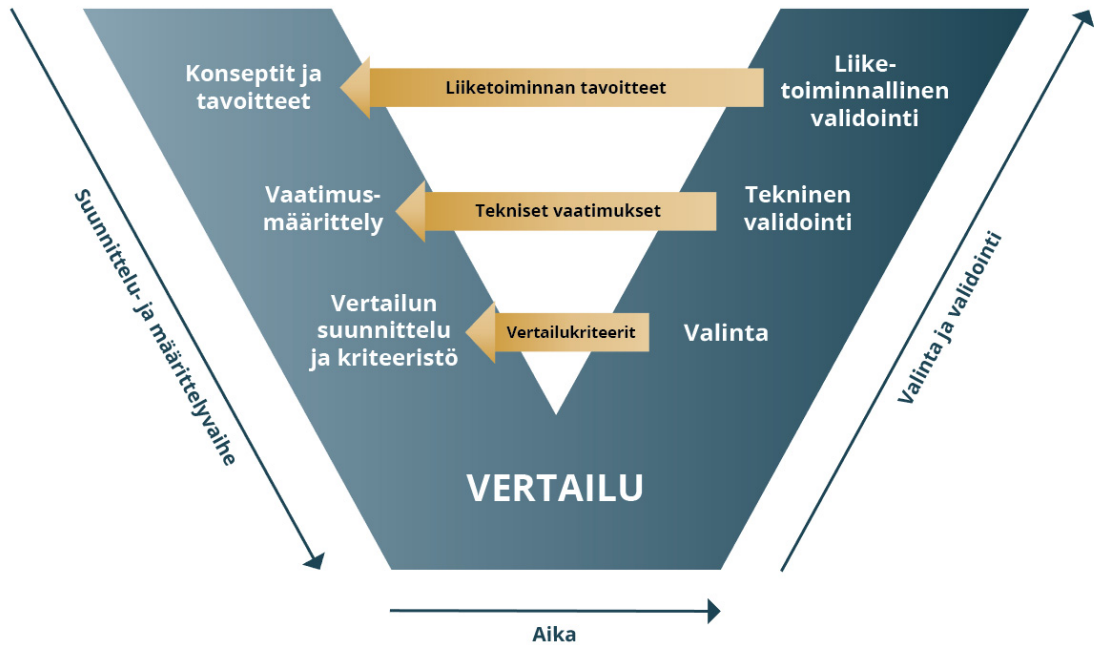
2.3 V-mallin soveltaminen valintaprosessiin

Tässä työssä sovelletaan V-mallia parhaan vaihtoehdon etsintä-, vertailu- ja valintaprosessiin. Siitä huolimatta, että V-mallia on perinteisesti käytetty ohjelmistotuotannon pro-

sesseissa, se soveltuu hyvin käytettäväksi tähän valintaprosessiin, sillä jokaisessa vaiheessa tehtävää työtä voidaan validoida ja verifioida V-mallin mukaisesti prosessin loppuvaiheessa. Pohjimmiltaan työssä käsiteltävät aiheet seuraavat ICT-hankinnoissa usein käytettyä prosessia mikä koostuu neljästä vaiheesta: vaatimusmäärittelystä sekä suunnittelusta, vaihtoehtojen kartoituksesta sekä vertailusta, toimittajan valinnasta sekä sopimuksien laadinnasta ja toteutuksesta sekä käyttöönnotosta (Kalliala & Kaskela, 2005).

V-malli valittiin kehitettävän prosessin pohjaksi siitä syystä, että V-mallissa testauksen, tai tässä tapauksessa validoinnin ja verifioinnin määrä on hyvin suuri ja näiden linkitys vaatimusmäärittelyyn on vahva. Näin voidaan varmistua siitä, että vertailu- ja valintaprosessi on täyttänyt määritellyt tarpeet sekä halutun lopputuloksen. Siitä huolimatta, että V-mallissa on mahdollisuus iterointiin, valittiin se kuitenkin osittain myös sen lineaarisuuden takia. Esimerkiksi vaihtoehtoisena prosessina pohdittu ominaisuuslähtöinen kehitys (engl. property driven development, PDD) keskittyy vaatimusmäärittelyssä kehitettyjen käyttäjätarinoiden ratkaisun iterointiin niin kauan, kunnes vaatimukset tyydyttävä ratkaisu on löydetty. (Baumeister et al. 2004). Tässä prosessissa etsitään kokonaisratkaisua, useasta vaihtoehdosta, jolloin iterointi on käytännössä mahdollista vain muuttamalla vertailukriteereitä tai vertailun vaihtoehtoja. Myöskään vaatimusmäärittelyn aikana muodostettuja käyttäjätarinoita ei hyödynnetä täysin samalla tavalla normaaliin kehitysprojektiin verrattuna ja sen sijasta käyttäjätarinoita käytetään vaatimuksien muodostamiseen kerätyistä tarpeista, joista käyttäjätarinat ovat kirjoitettu.

Kuvassa 4 nähtävissä olevasta sovelletun V-mallin kaaviokuvasta on nähtävissä, miten tutkimuksen eri vaiheita tullaan suorittamaan V-mallin mukaisesti. Vasemmassa haarassa tehdään vertailun suunnittelu- ja määrittelyvaiheeseen liittyviä tehtäviä, kaavion pohjalla tehdään itse vertailu. Oikeassa haarassa tehdään valinta ja sen validointi, käyttäen suunnittelu- ja määrittelyvaiheessa kehitettyjä kriteereitä, vaatimuksia ja tavoitteita.



Kuva 4. Työssä sovelletun V-mallin kaaviokuva

Konseptit ja tavoitteet -osiossa määritellään työn kannalta merkittävät konseptit ja käsitteet digitalisaatiosta sekä siihen liittyvät yleiset tavoitteet. Osiossa kerätään myös liiketoiminnan tavoitteet, eli tieto siitä, mitä IoT-alustalla avulla halutaan saavuttaa.

Vaatusmäärittely-osiossa selvitetään työlle sopiva menetelmä toteuttaa vaatimusten määrittely ja tunnistetaan liiketoiminnan tavoitteista tarpeet. Tarpeista muodostetaan liiketoiminnan vaatimuksia, joista taas saadaan teknisiä vaatimuksia IoT-alustalle.

Vertailun suunnittelu ja kriteeristö -osiossa määritellään IoT-alustojen vertailussa käytettävät menetelmät ja muodostetaan edellisessä vaiheessa saaduista teknisistä vaatimuksista vertailukriteerit, joita käytetään alustojen vertailussa. Tässä vaiheessa myös kartoitetaan IoT-alustojen tarjontaa ja suoritetaan esivalinta, jossa vain lupaavimmat vaihtoehdot otetaan mukaan vertailuun.

Kaavion pohjalla, toteutusvaiheessa tehdään IoT-alustojen vertailu käyttäen muodostettuja vertailukriteereitä ja Pughin matriisia. Vertailun aikana yritetään muodostaa jokaisesta esivalitusta vaihtoehdosta kokonaiskuva sen ominaisuuksista, toimintatavoista ja kyvykkyyksistä, jota vahvistettiin koekäyttämällä niin montaa esivalittua IoT-alustaa kuin mahdollista. Vertailuosiossa tehdään myös toinen karsintakierros, jossa valitaan kolme lupaavinta vaihtoehtoa ja verrataan näitä tarkemmin keskenään.

Kaavion oikeassa haarassa, aloitetaan valintaosuudesta, jossa analysoidaan vertailun tuloksia ja valitaan paras vaihtoehto vertailun sekä siinä käytettävien vertailukriteerien avulla. Valitun vaihtoehdon pitäisi olla vertailukriteerien perusteella parempi kuin muut

vaihtoehdot tai muuten kompromissina täyttää sekä vaaditut tekniset, että liiketoiminnalliset vaatimukset, joiden täytyminen varmistetaan seuraavissa osioissa.

Tekninen validointi -osiossa valittua IoT-alustaa validoidaan teknisten ominaisuuksien perusteella ja varmistetaan siitä, että alusta täyttää vaatimusmäärittelyssä muodostetut tekniset vaatimukset. Tarvittaessa teknisiä vaatimuksia voidaan seurata takaisin liiketoiminnallisiin vaatimuksiin ja varmistua siitä, että täyttyykö liiketoiminnallinen vaatimus, jos tekninen vaatimus ei täyty tyydyttävästi.

Liiketoiminnallinen validointi -osiossa IoT-alustaa validoidaan liiketoiminnan tavoitteita vasten. Näin voidaan varmistua siitä, että valittu alusta mahdollistaa liiketoiminnan näkökulmasta määritellyt ja suunnitellut asiat ja on siten sopiva valinta tekniseksi pohjaksi uusien digitaalisten palveluiden ja ratkaisujen toteuttamiselle. Näin voidaan varmistaa, että valittu ratkaisu on liiketoiminnan näkökulmasta oikea valinta.

2.4 Vaatimusmäärittely

Vaatimusmäärittely on tärkeä osa jokaista projektia vaatimuksien ollessa viimekädessä suuntaviivat, joiden perusteella projektin aikana kaikki työ tehdään. Useimmiten teollisuuden projekteissa vaatimusmäärittelyä ei tehdä riittävällä tasolla, vaan vaatimuksien analysointi jätetään tekemättä ja siirrytään suoraan suunnittelu- ja toteutusvaiheisiin. (Young 2004). Tämä ongelma oli myös tunnistettu Fastemsilla ja siksi vaatimusmäärittelyyn ja etenkin vaatimuksien analysointiin käytettiin aikaa.

Erilaisia vaatimusmäärittelymenetelmiä, malleja ja toimintakehyksiä (engl. framework) on paljon erilaisia ja jokaisessa on hyvät ja huonot puolensa. Osa lähestymistavoista ja työkaluista soveltuvat hyvin erilaisiin projekteihin ja osa on todella vahvasti fokuosoituneet vain tietyn osa-alueen vaatimuksien määrittelyyn. (Young 2004). Kaikille menetelmille yhteistä on kuitenkin tavoite yrittää yksinkertaistaa ja luoda mahdollisuudet hallita hyvin kompleksisten projektien vaatimuksia (Mahan & Dixit 2013). Käytetyn vaatimusmäärittelymenetelmän valinnassa keskityttiin yleisiin, helposti sovellettaviin menetelmiin.

Objekti-orientoituneessa (engl. object-oriented) vaatimusmäärittelymenetelmässä projektin lopputulos ja toimintaympäristö mallinnetaan tarkasti erilaisiin ryhmiin ja kokonaisuuksiin. Näitä kokonaisuuksia voidaan hallita, priorisoida ja kuvailla tarkemmin kuin isoja kokonaisuuksia. Myös objektien välisiä toiminnallisuuksia ja riippuvuussuhteita voidaan esittää helposti eri ryhmien välillä. (Mahan & Dixit 2013).

Päämääräorientoituneessa (engl. goal-oriented) menetelmässä keskitytään eri sidosryhmien tunnistamiseen ja heiltä kerättyjen päämäärien sekä tavoitteiden analysointiin, muokkaamiseen ja analysointiin (Mahan & Dixit 2013). Tässä menetelmässä tekniset nä-

kökulmat eivät ole lähtökohta, kuten useassa muussa menetelmässä. Sen sijaan menetelmässä määritellään ne kriteerit, jotka vaatimuksien ja siten lopputuloksen tulee täyttää. Koska vaatimuksia muodostetaan päämääristä, voidaan helposti muodostaa malleja, joiden avulla aivan yksittäisiä suunnittelupäätöksiä voidaan tarkasti jäljittää jopa korkean tason vaatimukseen ja tiettyyn sidosryhmään. (Lamsweerde 2001). Haasteena tässä päämääräorientoineessa menetelmässä on oikeiden sidosryhmien tunnistus ja heiltä vaatimuksien kerääminen oikealla tasolla (Mahan & Dixit 2013).

Takaisinmallinnus-menetelmässä (engl. reverse engineering) tutkitaan olemassa olevia järjestelmiä ja muodostetaan niistä kerätyn tiedon pohjalta vaatimuksia. Tämän menetelmän avulla voidaan saada muodostettua selkeämpiä vaatimuksia, kun jo valmiiksi tiedetään mitkä asiat toimivat ja mitkä eivät. Menetelmä on erityisen sopiva olemassa olevien järjestelmien uudistamiseen, kun vanhaa järjestelmää tutkimalla saadaan helposti kerättyä uudistettavien kohteiden vaatimuksia. (Mahan & Dixit 2013).

Niin sanotussa perinteisessä, mitään erityisiä menetelmiä hyödyntävässä vaatimusmäärittelyprosessissa löydetty vaatimukset ovat järjestelmättömiä, mahdollisesti huonosti hallittuja ja niiden integrointi suunnitteluvaiheessa voi olla vaikeaa (Mahan & Dixit 2013). Usein erilaisissa projekteissa vaatimusmäärittelyyn ei panosteta riittävästi, vaan halutaan edetä nopeasti ja mahdollisesti voidaan ajatella, että määrittelydokumentissa kuitenkin käsitellään näitä asioita. Pahimmassa tapauksessa toimitaan ihmisluonnolle tyypillisellä tavalla ja edetään mahdollisimman vähällä suunnittelulla suoraan toteutusvaiheeseen. Tästä syystä olisi hyvä käyttää jotakin systemaattista menetelmää vaatimusmäärittelyn tekemiseen, sillä menetelmästä riippumatta, ideana on, että käytetty menetelmä ohjaa toimintaa kohti parempaa lopputulosta. (Young 2004).

Käytettäväksi vaatimusmäärittelymenetelmäksi valittiin päämääräorientoitunut menetelmä, sillä heti alussa tärkeimmäksi sidosryhmäksi tiedettiin olevan Fastemsin johtoryhmä, joka oli uudessa strategiassa linjannut selkeitä digitalisaation tavoitteita ja päämääriä yritykselle. Näiden hyvin korkealla tasolla olevien liiketoiminnan tavoitteiden lisäksi tavoitteita tiedettiin löydettävän helposti yleisistä digitalisaation ja teollisen internetin määritelmistä, joissa kuvataan yleisiä peruseriaatteita, joita pitää ottaa huomioon. Sidosryhmien tunnistamisen ja tavoitteiden keräämisen jälkeen tavoitteista pitää löytää tarpeet. Tarpeiden tunnistamisen jälkeen nämä muutetaan vielä yleisiksi vaatimuksiksi yhdessä joustavien valmistusjärjestelmien tuomien erityisvaatimusten kanssa. Lopuksi yleisistä vaatimuksista saadaan teknisiä vaatimuksia IoT-alustoille, joita käytetään myös alustojen vertailukriteereinä.

2.5 Pughin matriisi

Vertailun aikana vaihtoehtojen keskinäiseen arviointiin käytetään Pughin matriisia, mikä on helppokäyttöinen, matriisimuotoinen arviointimenetelmä parhaan vaihtoehdon valitsemiseksi määriteltujen kriteerien perusteella. Pughin matriisin isoimpia etuja verrattuna

muihin päätöksentekotyökaluihin on, että Pughin matriisi toimii myös silloin, kun vertailukriteereitä on huomattavan paljon. Kompleksisuutta poistetaan vertailemalla ainoastaan kahta vaihtoehtoa kerrallaan keskenään, mikä lisää myös vertailun objektiivisuutta, kun vertailun tekijän ei tarvitse valita yhtä monesta vaihtoehdosta kerrallaan. (Burge 2009). Taulukossa 1 on nähtävissä esimerkki Pughin matriisin käytöstä.

Taulukko 1. *Esimerkki Pughin matriisista.*

	Vaihtoehto 1	Vaihtoehto 2	Vaihtoehto 3	Vaihtoehto 4
Kriteeri 1	S	+	S	+
Kriteeri 2	S	-	S	+
Kriteeri 3	S	S	S	+
Kriteeri 4	S	-	+	+
Kriteeri 5	S	-	+	+
Kriteeri 6	S	-	S	-
Kriteeri 7	S	+	S	-
Kriteeri 8	S	+	S	-
Kriteeri 9	S	-	S	-
Kriteeri 10	S	S	-	S
Yhteensä +	0	3	2	5
Yhteensä -	0	5	1	4
Kokonaistulos	0	-2	1	1

Pughin matriisia täyttäessä aluksi yksi vertailtavista vaihtoehdoista valitaan vertailukohdeksi. Tämä vertailukohde voi olla valittu satunnaisesti tai se voi olla vahva ehdokas, mutta vertailukohdan valinnalla ei ole niin isoa merkitystä. Jos matriisin avulla ollaan kehittämässä jotain jo olemassa olevaa ratkaisua, kannattaa vertailukohdaksi valita nykyinen ratkaisu. Vertailukohteen sarakkeeseen jokaisen kriteerin kohdalle merkitään kirjain S. Vertailua tehdessä Pughin matriisin avulla kriteerit käydään läpi jokaisen kilpailevan vaihtoehdon osalta siten, että vaihtoehtoa verrataan vertailukohteeseen. Jos kilpaileva vaihtoehto täyttää kriteerin vaatimukset paremmin kuin vertailukohde, merkataan sen sarakkeeseen plus tai jos vertailukohde on parempi, merkataan miinus. Jos molemmat vaihtoehdot täyttävät kriteerin yhtä hyvin, merkitään sarakkeeseen kirjain S. Suuremman erotuksen saavuttamiseen voidaan käyttää myös kaksinkertaisia plus- ja miinusmerkkejä. Kun kaikki sarakkeen rivit on täytetty, lasketaan plus- ja miinusmerkkien yhteismäärä sekä niiden summa, jolloin saadaan vaihtoehdon kokonaistulos verrattuna vertailukohteeseen. Tiedon avulla voidaan tehdä parhaan vaihtoehdon valinta, pudottaa heikot vaihtoehdot pois tai tehdä uusi vertailu eri vertailukohteella. (Burge 2009).

3. VAATIMUSMÄÄRITTELYPROSESSI

3.1 Tavoitteet

3.1.1 Digitalisaatio ja teollinen internet

Digitalisaatio tulee vaikuttamaan niin yksittäisten ihmisten elämään automatisoimalla prosesseja ja rutiineja. Älykkäät laitteet verkottuvat ja niistä saatavan tiedon avulla yritykset joutuvat muuttamaan toiminta- ja ajatusmallejaan pysyäkseen mukana kilpailussa. Pilvipalveluiden yleistymisen myötä, käytännössä ääretön laskentakapasiteetti mahdollistaa tiedon jalostamisen informaatioksi, minkä avulla yritykset pystyvät tuottamaan uutta liiketoimintaa tai muuttamaan nykyistä radikaalisti. (Evans & Annunziata 2012). Digitalisaation myötä useat yritykset voivatkin joutua miettimään, mikä yrityksen rooli on tai missä toimialassa se toimii. Yritykset kohtaavat uudenlaista kilpailua yrityksiltä, jotka eivät aikaisemmin ole toimineet samoissa ympäristöissä ja vaarana onkin, että uudet tulokkaat syrjäyttävät vanhoja, pitkään alalla toimineita yrityksiä, jotka eivät jostain syystä reagoi riittävän nopeasti tai lainkaan digitalisaation tuomaan muutokseen. (Porter & Heppelmann 2014).

Mikään ala ei ole turvassa digitalisaation vaikutuksilta ja yksi digitalisaation suurimmista vaikutusalueista on teollisuus ja sen kaikki toimialat. Digitalisaation sanotaankin olevan teollisuuden neljäs vallankumous ensimmäisen ollessa höyrykoneet ja kaupungistuminen 1800-luvun alussa. Toinen vallankumous tapahtui 1900-luvun alussa massavalmistuksen ja tuotantolinjojen laskiessa tuotteiden hintoja ja kolmas 1900-luvun loppupuolella, jolloin automaatio ja tietotekniikka tehostivat tuotantoa huomattavasti. Tällä hetkellä meillä on neljäs vallankumous edelleen tehostaa tuotantoa, mutta tuo mukanaan teollisuuden näkökulmia, joita ennen ei osattu siihen yhdistää. Uudenlaiset fyysiset tuotantotavat kuten materiaalia lisäävät menetelmät tai komposiittimateriaalit yhdistettynä digitaalisen maailman etuihin kuten tietoon, informaatioon sekä verkottumiseen luovat laajan pohjan teollisuuden uudistumiselle mahdollistaen tehokkuuden lisäksi esimerkiksi paremman laadun, etävalvotun kunnossapidon ja siten paremman luotettavuuden. (Hughes 2017). Tällä muutoksella on myös muita kutsumanimiä kuten ”Advanced Manufacturing” sekä kansallisia ohjelmia kuten saksalainen ”Industrie 4.0” tai yhdysvaltalainen ”Smart Manufacturing”. Myös mahdollistavien teknologioiden nimityksiä käytetään toisinaan, kuten teollinen internet tai ”Industrial Internet of Things” (IIoT) eli teollisten asioiden internet. (Hwaiyu 2015).

Teollinen internet teknologiamahdollistajana voidaan kuvata teollisten laitteiden, prosessien ja niihin liittyvien ihmisten verkostoitumisena toisiinsa sekä internettiin, mikä mah-

dollistaa niiden seurannan ja ohjauksen. Tämä mahdollistaa tuotteita, palveluita ja toimintamalleja, joita aikaisemmin on ollut mahdotonta tai hyvin vaikeaa toteuttaa teollisuudessa. Esimerkkeinä teollisten laitteiden etävalvonta ja -ohjaus, hyvin yksityiskohtainen tuotannonohjaus sekä ennustavan huollon toteutus. Uudenlaiset sensorit, analytiikkatyökalut sekä tietoliikenneyhteydet tuottavat ja kuljettavat tietoa sekä informaatiota yli yritysrajojen, lisäten näin mahdollisuuksia uuteen palveluliiketoimintaan ja erilaisten ekosysteemien luontiin, mitkä parhaimmillaan hyödyttävät useita eri alan toimijoita. (Martinsuo & Kärrri 2017). Etenkin valmistava teollisuus on jo nyt hyvin yhdistynyt lähes ekosysteemejä muodostaviin toimittaja- ja alihankintaverkostoihin, joissa toimitusketjut voivat olla hyvinkin laajalle levinneitä. Teollisen internetin mahdollistama suuri liike-mahdollisuuspotentiaali kiinnostaa lukuisia toimijoita, houkuttellen niin pieniä kuin suuria teknologiayrityksiä panostamaan niin ratkaisuiden tai palveluiden muodossa teollisuuden aivan eri mittakaavassa kuin ennen. (World Economic Forum 2015). Onkin ennustettu, että teollinen internet voi tulla nostamaan lähivuosina maailman bruttokansantuottoa 10-15 triljoonalla dollarilla, nykyisen Yhdysvaltojen bruttokansantuoton verran. Esimerkkinä teollisen internetin yhdestä sovelluskohteesta, ilmailuteollisuudessa on arvioitu, että teollisen internetin avulla olisi helposti saavutettavissa viiden prosentin säästö polttoainekustannuksissa laitteita, lentoreittejä ja toimintatapoja optimoimalla ja näin saaden yli kahdeksan miljardin dollarin vuosittain säästö. (Evans & Annunziata 2012).

3.1.2 Liiketoiminnan tavoitteet

Fastemsin digitaalisen strategian jalkauttamiprojektin lopputuloksena haluttiin luoda tekniset valmiudet kehittää uusia, digitalisaatiota hyödyntäviä palveluita, tehostaa omaa toimintaa ja löytää muita mahdollisia uusia tulonlähteitä olemalla markkinoiden edelläkävijä. Tämä olisi mahdollista hyödyntämällä järjestelmissä oleva tietoa keräämällä sekä jalostamalla siitä merkitsevää ja siten hyödyllistä informaatiota erilaisten palveluiden käyttöön sekä päätöksenteon tueksi.

Tarve oli myös uudistaa nykyinen, kankea ja vanhenemassa oleva konnektiviteettiratkaisu. Konnektiviteettiä käytetään yhteyksien luomiseen asiakkaiden järjestelmiin esimerkiksi etätuen antamiseen. Konnektiviteetti nähtiin myös uusien tulonlähteiden kriittisenä osana, kun uusia konnektiviteettiratkaisuita voitaisiin jälkimarkkinoinin lisäksi käyttää useissa erilaisissa muissa sovelluskohteissa ja uusissa palveluissa.

Tiedon keräämistä, käsittelyä ja tallentamista varten tarvittiin valmiudet, minkä suunnittelu ja toteutus olivatkin yksi Fastemsin digitalisaatioprojektin pääkohdista. Joustavat valmistusjärjestelmät korkeine automaatioasteineen tuottavat paljon tietoa järjestelmällä tehtävästä tuotannosta sekä automaatiojärjestelmästä itsestään. Tieto nykyisellään kuitenkin jää järjestelmään paikallisesti, josta se tietyn ajan kuluttua poistetaan tilan säästämiseksi. Täten tietoa käytännössä nykyisellään hyödynnetään vain asiakkaalle paikallisesti esitetyissä automaatiojärjestelmän omissa raportointinäkymissä. Osittain tämän tiedon keräämättä jättämiseen oli syynä yleinen teollisuuden ongelma, että tuotantojärjestelmät ovat

perinteisesti olleet eristyksissä olevia, omia saarekkeitaan tehtaiden sisällä puutteellisin verkkoyhteyksin. Usein tähän syynä on myös huoli tietoturvasta. Tästä syystä tietoturvan tason pitäisi olla korkeampi kuin yleisesti kyseisellä teollisuudenalalla on totuttu, että myös vaativimmat asiakkaat, kuten ilmaliikenne- ja puolustusteollisuus, hyväksyisivät käytetyt ratkaisut.

Erityisen mielenkiinnon kohteena olivat laitteisiin liittyvät elinkaari palvelut, joiden toimintatapoja haluttiin muuttaa enemmän proaktiiviseksi. Kerättyä tietoa haluttiin hyödyntää ennakoivan huollon toteutuksessa ja yleisestikin saada tukipalveluista proaktiivisempia kuin ennen. Aiemmin Fastemsin huollon ja etätuen toiminta on ollut täysin reaktiivista ja asiakkaan kanssa on asioitu ainoastaan silloin, kun asiakas itse ilmoitti ongelmasta Fastemssille. Aitoa proaktiivista ongelmien estämistä tai vian korjaamista ei ennen asiakkaan yhteydenottoa ole tehty. Joillekin asiakkaille oli saatettu tehdä erikseen sovittuja, vuosittaisia tai puolivuositteisiä huoltokäyntejä tai etänä tehtyjä järjestelmätarkistuksia.

Uudenlaisen tarjonnan lisäksi strategiassa määriteltiin myös toimintatapojen uudistaminen. Ideana oli mahdollistaa nopea, kokeileva palvelukehitys, mitä varten tarvitaan kyvykkyudet luoda itse vähintään prototyyppejä ilman kolmannen osapuolen työtä. Tällä tavalla Fastems pystyisi ketterästi asiakkaiden kanssa kehittämään palveluita ja validoimaan ideoiden toimivuus ilman hitaita alihankintaprosesseja ja muuta raskasta byrokraatia. Perinteisesti joustavat valmistusjärjestelmät ovat useimmiten yksilöllisiä ja asiakkaiden tarpeiden mukaan suunniteltuja projektitoimituksia, mutta siitä huolimatta käytetyt ratkaisut eivät saisi lisätä räätälöidyn tai muuten käsityönä tehdyn työn määrää projekteissa kohtuuttoman paljon.

3.2 Vaatimuksien muodostus

3.2.1 Tarpeiden tunnistus

Vaatimusmäärittelyprosesseissa tarpeiksi kutsutaan niitä asioita, jotka kuvaavat keinot, joiden avulla määritellyt tavoitteet voidaan saavuttaa nykyisestä tilanteesta katsoen. Tarpeita tutkiessa löydetään sekä mahdollisuuksia että ongelmia, joiden hyödyntämisestä tai ratkaisemisesta tulee muodollisempia vaatimuksia. Tarpeita tutkiessa käytetään usein niin sanottuja käyttäjätarinoita (engl. user story) joissa kuvataan tietyn sidosryhmän tai roolin suorittamia tehtäviä tai toimintaketjuja. Tällä tavoin tarpeita on helpompaa löytää ja mallintaa, kun kokonaisuutta voidaan tarkastella pienemmissä osissa ja kuten käyttäjän, tai yleisemmin tehtävien suorittajien näkökulmasta. (Kujala et al. 2001). Kuvassa 5 on nähtävissä esimerkki yhden tehtäväkuvauksen taulukkomuotoisesta dokumentoinnista.

Task sequence:	Problems and possibilities:
Step 1: When trapped in an elevator, passenger makes an emergency alarm.	<ul style="list-style-type: none"> • Passengers want to get out of the elevator as soon as possible • All kinds of passengers must be able to make an alarm call (blind, foreigners etc.) • Sometimes passengers may make false alarms unintentionally. • Passengers may be in panic. • Passengers need instant confirmation that they have created a connection to the service centre operator and that they are going to get help.
Step 2: Unoccupied service centre operator receives the emergency alarm call and asks for information.	<ul style="list-style-type: none"> • Different versions and types of remote monitoring systems. • Passenger is the only information source. • Service centre operator does not notice the emergency alarm call.
Step 3: Service centre operator completes transmission of information to the system and sends it to the area serviceman.	<ul style="list-style-type: none"> • Laborious phase for the service centre operator. • Simultaneous calls must be differentiated. • Serviceman cannot see all information. • Inadequate information from a site system. • Possibility: Instructions as to how to operate the system. • Possibility: Possibility to open phone line from Call Centre to the elevator.
Step 4: Service centre operator calls the serviceman and reads the description of the failure.	<ul style="list-style-type: none"> • Extra work for the service centre operator.

Kuva 5. Taulukkomuotoinen käyttäjätarina hissien hätätilanteen ratkaisusta (Kujala et al. 2001)

Kuvan 5 taulukossa on kuvattu tapahtumaketju, jossa hissiin jumiin jäänyt henkilö tekee ilmoituksen avuntarpeesta palvelukeskukseen. Jokaisen korkealla tasolla määritellyn tehtävän yhteyteen on listattu kyseiseen alitehtävään liittyviä mahdollisuuksia ja ongelmia. Henkilön tavoitteena on päästä pois jumittuneesta hissistä, tarpeena saada huoltomies paikalle päästämään henkilö ulos hissistä ja esimerkkinä vaatimukseksi päätyvästä ongelmasta on, että kaikenlaisten ihmisten, kuten sokeiden tai ulkomaalaisten, pitää pystyä tekemään hätäpuhelu. Tämä voi esimerkiksi muodostua vaatimukseksi, missä vaaditaan, että hätäpuheluun käytettävässä painikkeessa käytetään yleisesti hätätilanteeseen yhdistettyjä merkintöjä ja värejä tai että painikkeen yhteydessä käyttötarkoitus ilmaistaan myös pistekirjoituksen avulla.

3.2.2 Tarpeiden jalostus vaatimuksiksi

Yleisistä digitalisaation ja teollisen internetin sekä Fastemsin liiketoiminnan tavoitteista tunnistettiin yleisellä tasolla neljä kokonaisuutta tai teemaa tarpeista, joista vaatimukset tulevat koostumaan. Yleisellä tasolla tarpeet olivat seuraavat:

- Kerätä ja tallentaa järjestelmässä olevaa tietoa
- Analysoida ja jalostaa järjestelmästä kerättyä tietoa informaatioksi
- Tieto ja informaatio kaikkien sidosryhmien käytettävissä
- Loppukäyttäjäpalveluiden ketterä kehitys

Näistä tarpeita tehtiin kappaleessa 2.3.1 esitetyn mukaisesti taulukkomuotoisia käyttäjätarinoita, joissa kyseisistä tarpeista listattiin tehtäväketjuja ja kerättiin niiden suorittamisen aikana esiintyviä ongelmia ja mahdollisuuksia.

Taulukossa 2 esitetyn tarpeen ”*Kerätä ja tallentaa järjestelmässä olevaa tietoa*” tehtäväketjussa keskitytään tietoon sen kohdejärjestelmän näkökulmasta. Tieto tuotetaan automaatiojärjestelmässä asiakkaan tehtaassa ja lähtökohtaisesti tieto tallentuu paikallisesti. Nyt tieto pitäisi saada kerättyä ja asiakkaan suostumuksella siirtää Fastemsin keskitettyyn tietovarastoon hyödynnettäväksi. Kriittisimmät asiat olivat yhteyksien luominen asiakkaan tehtaassa sijaitsevaan automaatiojärjestelmään ja tiedon kerääminen siten, että asiakas saadaan hyväksymään toiminta. Myös tarve tiedon tallennukselle tietoturvallisesti ja kaikkien määräyksien mukaisesti keskitettyyn tietovarastoon on hyvin kriittinen.

Taulukko 2. Käyttäjätarina tarpeesta ”*Kerätä ja tallentaa järjestelmässä olevaa tietoa*”

Tehtäväketju	Ongelmat ja mahdollisuudet
1. Pääsyn saaminen tietoon	<ul style="list-style-type: none"> • Järjestelmissä todella paljon tietoa järjestelmästä ja sillä tehdystä tuotannosta. • Tieto suljettujen tehdasjärjestelmien sisällä. • Asiakkaat haluttomia luovuttamaan tietoa Fastemsin käyttöön. • Omistajuuskysymykset; kuka omistaa tiedon? • Tarvitaan erilliset sopimukset tiedon keruusta.
2. Tiedon kerääminen järjestelmästä	<ul style="list-style-type: none"> • Järjestelmissä on valmiiksi korkeamman tason ohjausta ja yksi piste, johon voidaan liittyä kohtuullisella työllä. • Windows-ympäristöön helppo ohjelmoida uusia sovelluksia keräystä varten. • Myös palvelinympäristössä käytetyistä tietoteknisistä laitteistoista saadaan kerättyä kunnossapitotietoa.
3. Tiedon siirtäminen tietovarastoon	<ul style="list-style-type: none"> • Tehdasympäristöissä tietoliikenne voi olla haastavaa esimerkiksi puutteellisten yhteyksien takia. • Mahdollisuus käyttää useita eri yhteystapoja muun muassa mobiiliyhteyksiä tarvittaessa. • Valittu etäyhteyksimenetelmä ei saa johtaa yhden toimittajan tilanteeseen. Toimittajaa pitää pystyä vaihtamaan tarvittaessa. • Nykyinen etäyhteyksimenetelmä on vanhentunut eikä sovellu jatkuvaan yhteyteen tiedon lähetystä varten (on-demand -yhteys). • Asiakkaan pitää hyväksyä käytetyt tietoliikenne ratkaisut. • Nykyaikana tietoturvan merkitys on todella suuri.
4. Tiedon tallentaminen tietovarastoon	<ul style="list-style-type: none"> • Tieto voidaan tallentaa joko strukturoidussa tai strukturoimattomassa muodossa. • Yksi keskitetty paikka kaiken tiedon tallentamiseen. • Tietosuojamääräykset voivat vaikuttaa muun muassa siihen, voidaanko kaikkien asiakkaiden tieto tallentaa maantieteellisesti samaan paikkaan. • Datamäärä voi olla suuri. • Tieto pitää olla helposti myös haettavissa keskusvarastosta.

Taulukossa 3 on käyttäjätarina tarpeesta ”*Analysoida ja jalostaa järjestelmästä kerättyä tietoa informaatioksi*”. Tässä vaiheessa tietoa on jo kerätty keskusvarastoon esimerkiksi sensorien antamassa muodossa ja nyt tietoa pitäisi tulkita ensiksi siten, että mitä tieto itseasiassa merkitsee ja jalostaa siitä informaatiota. Tietoa voidaan esimerkiksi mallintaa, vertailla ja yhdistellä siten, että lopputulokseksi saadaan informaatiota esimerkiksi jonkin laitteen kunnosta pelkkien raaka-arvojen sijasta. Kriittisimmät löydökset olivat tarpeellisen tiedon tunnistaminen sekä tiedon monitasoinen mallinnus. Tietoa pitäisi pystyä myös vertailemaan eri automaatiojärjestelmien välillä siitä huolimatta, että järjestelmät voivat olla hyvinkin erilaisia ja mahdollisesti myös raskaasti asiakkaalle räätälöityjä.

Taulukko 3. *Käyttäjätarina tarpeesta ”Analysoida ja jalostaa järjestelmästä kerättyä tietoa informaatioksi”*

Tehtäväketju	Ongelmat ja mahdollisuudet
1. Hyödyllisen tiedon löytäminen	<ul style="list-style-type: none"> • Tarpeellisen tiedon tunnistaminen voi olla haastavaa. • Tiedon etsintään voidaan käyttää erilaisia työkaluja muun muassa koneoppimista. • Kaikki tieto ei välttämättä ole hyödyllistä kaikille sidosryhmille. • Oikea tiedon määrä; kuinka paljon on riittävästi ja kuinka paljon liikaa?
2. Tiedon mallinnus ja normalisointi	<ul style="list-style-type: none"> • Tietoa voidaan mallintaa monella eri tasolla; osajärjestelmätasolta aivan sensoritasolle. • Voidaan aloittaa karkealla tasolla ja tarkentaa malleja ajan ja lisätiedon myötä. • Kaikki järjestelmät erilaisia; mitä käytetään normalisoinnin pohjana? • Kaikista järjestelmistä saadaan ainakin suurin piirtein samat perustiedot ja useimmiten räätälöinnit ovat rakennettu perustoiminnallisuuden päälle tai sen lisäksi. • Osassa järjestelmiä myös perustoiminnallisuudet ovat räätälöity asiakkaan tarpeiden mukaan ja näiden tunnistus voi olla vaikeaa.
3. Tiedon analysointi ja vertailu	<ul style="list-style-type: none"> • Tietoa voidaan analysoida myös automaattisesti muun muassa erilaisten raja-arvojen avulla ja luoda sääntöjä sekä hälytyksiä tiedon perusteella. • Tieto pitää olla normalisoitu siten, että vertailu eri järjestelmien kesken onnistuu. • Tietoa pitää pystyä yhdistämään useasta järjestelmästä yhteistä käsittelyä varten.
4. Informaatioksi jalostaminen ja hyödyntäminen	<ul style="list-style-type: none"> • Malleista ja/tai yhdistellystä tiedosta voidaan muodostaa erilaisia johtopäätöksiä. • Yhdistetystä tiedosta saadaan esimerkiksi keskiarvoja ja normeja, joihin voidaan verrata yksittäisiä järjestelmiä. • Esimerkiksi komponenttien kestojen arviointia ennustavan huollon päätöksenteon tueksi. • Järjestelmien käyttötietoa voidaan hyödyntää konsultaatiopalveluiden sisällön määrittämisessä.

Käyttäjätarina tarpeelle ”*Tieto ja informaatio kaikkien sidosryhmien käytettävissä*” on esitetty taulukossa 4. Sitä varten, että osattaisiin kerätä oikea tieto ja käsitellä se oikealla tavalla, pitää löytää mahdollisia tiedon hyödyntäjiä. Sidosryhmien tunnistamisessa tärkein asia on tunnistaa, onko sidosryhmä sisäinen vai ulkoinen. Sisäisiä sidosryhmiä on helpompi palvella kuin ulkoisia, sillä heidän tarpeidensa tunnistus on helpompaa, kun työskennellään saman yrityksen sisällä. Kun sidosryhmät on löydetty ja heidän tarpeensa on otettu huomioon siten, että tarvittava tieto ja informaatio on kerätty, pitää heille tarjota pääsy tai mahdollisuus hyödyntämään tietoa. Staattisten raporttien sijaan tietoa voidaan jakaa erilaisten verkkosivujen avulla rakennettujen palveluiden kautta tai esimerkiksi mobiilisovelluksien kautta. Jaettu tieto pitää olla lähtökohtaisesti ajantasaista ja paikkansa pitävää, että sitä voidaan hyödyntää päätöksenteon tukena. Tieto pitää myös olla selkeästi sekä ymmärrettävästi esitetty ja mahdollisesti jollakin tavalla visualisoituna helpottamaan tiedon omaksumista.

Taulukko 4. Käyttäjätarina tarpeesta ”Tieto ja informaatio kaikkien sidosryhmien käytettävissä”

Tehtäväketju	Ongelmat ja mahdollisuudet
1. Sidosryhmien tunnistaminen	<ul style="list-style-type: none"> • Kaikkien merkitsevien ja hyödyllisten sidosryhmien löytäminen voi olla hankalaa. • Tietotarpeet voivat olla hyvinkin erilaisia sidosryhmien välillä. • Sidosryhmät voivat olla sisäisiä tai ulkoisia. • Ulkoisien sidosryhmien tunnistus asiakkaan organisaatiossa voi olla hankalaa.
2. Sidosryhmien tarpeiden tunnistaminen	<ul style="list-style-type: none"> • Löydettyjä sidosryhmiä voidaan haastatella. • Tarpeista löytyy varmasti päällekkäisyyksiä eri sidosryhmien välillä. • Tarpeiden tunnistaminen enemmän kaupallisen tuotehallinnan haaste.
3. Tarpeiden huomiointi suunnittelussa	<ul style="list-style-type: none"> • Kaikkien sidosryhmien tarpeet pitää priorisoida ja tehdä työarvio. • Tarpeet ohjaavat loppukäyttäjäpalvelukehitystä ja teknisiä vaatimuksia. • Tarpeet voivat määrittää myös reunaehdot ja rajoituksia, jotka tarvitsevat ottaa huomioon suunnittelussa ja toteutuksessa.
4. Sidosryhmien pääsy tietoon ja informaatioon	<ul style="list-style-type: none"> • Tietoa voidaan jakaa usean eri kanavan kautta, esimerkiksi erilaisin verkkosivujen tai mobiilisovelluksien avulla. • Sidosryhmät voivat käsitellä tietoa myös koneellisesti rajapintojen kautta. • Pääsy tietoon ja informaatioon pitää olla helppoa.
5. Sidosryhmien tiedon hyödyntäminen	<ul style="list-style-type: none"> • Päätöksenteon tueksi tarvitaan ajankohtaista ja paikkansa pitävää tietoa. • Tieto ja informaatio pitää olla ymmärrettävästi ja selkeästi esitettyä sekä mieluiten visualisoituna. • Tiedon hyödyntämistä olisi hyvä päästä jotenkin mittaamaan.

Tarpeen ”Loppukäyttäjäpalveluiden ketterä kehitys” käyttäjätarina on nähtävissä taulukossa 5. Tässä käyttäjätarinassa tietoa hyödynnetään kaupallisesti erilaisten tuotteiden tai palveluiden muodossa. Palveluiden taso voi vaihdella niin sisäiseen käyttöön tarkoitettuihin palveluista täysin valmiisiin Fastemsin asiakkaiden sekä sidosryhmien käytettäväksi tarkoitettuihin tuotteisiin tai palveluihin. Myös vähemmän jalostetun informaation tai jopa raakatiedon ympärille voidaan tuottaa liiketoimintaa muun muassa tarjoamalla pääsyä tiettyyn tietoon kolmansille osapuolille. Esimerkiksi automaatiojärjestelmään liitettyjen työstökoneiden valmistajalla voi olla mielenkiintoa saada tietoa omien tuotteidensa käyttötiedosta tai virhetilanteista. Loppukäyttäjäpalveluiden kehityksestä kriittisimpiä löydöksiä olivat tiedon visualisointi sekä käyttöliittymät ja kuinka palveluita kehitetään nyt ja tulevaisuudessa, kun asiakkaiden tarpeita ei vielä tiedetä ja niiden kerääminen asiakkaalta on hankalaa.

Taulukko 5. Käyttäjätarina tarpeesta ”Loppukäyttäjäpalveluiden ketterä kehitys”

Tehtäväketju	Ongelmat ja mahdollisuudet
1. Tuote/palveluideoiden keräys	<ul style="list-style-type: none"> Asiakkaiden tarpeiden tunnistus; asiakkaat eivät välttämättä osaa itse kertoa tarpeistaan. Ideoiden/asiakastarpeiden keräys sidosryhmiltä.
2. Sisäisien ja ulkoisien palveluiden tunnistus	<ul style="list-style-type: none"> Määrittely, onko palvelu sisäiseen vai ulkoiseen käyttöön; sisäisissä oman toiminnan tehostaminen isossa roolissa. Tuotehallinnalliset haasteet; ei tunnisteta hyötyä asiakkaalle.
3. Palveluiden kehitys	<ul style="list-style-type: none"> Palveluita/tuotteita voidaan kehittää joko omin resurssein tai vaihtelevalla määrällä ulkoista kehitystä. Liian pienet kehitysresurssit. Ketterät kehitysmenetelmät. Käyttöliittymien kehittämisen pitää olla helppoa ja nopeaa.
4. Palveluiden roll-out/lanseeraus	<ul style="list-style-type: none"> Kuinka helposti asiakkaat/käyttäjät käyttävät tuotteita tai pääsevät palveluihin. Mistä palveluita/tuotteita tarjotaan ja Yrityksen sisäinen buy-in; sidosryhmät eivät ole sitoutuneet riittävästi. Palveluihin liittyvien uusien liiketoiminta- ja hinnoittelumallien määrittely voi olla vaikeaa
5. Tiedon ja informaation jakaminen	<ul style="list-style-type: none"> Tietoa voidaan visualisoida ja esittää erilaisilla käyttöliittymillä. Tietoa voidaan jakaa ohjelmallisten rajapintojen kautta. Pääsynhallinta; pääsy pitää rajata vain tarvittavaan tietoon. Asiakkaat pitää eristää toisistaan. Yleinen tietoturva ja asiakkaiden hyväksyntä.

Käyttäjätarinoiden avulla löydettyjen ongelmien ratkaisemiseksi ja mahdollisuuksien hyödyntämiseksi näistä jalostettiin vaatimuksia. Useimmiten listatuista mahdollisuuksista saatiin helposti suoria toiveita tai tiukkoja vaatimuksia toiminnallisuuden suhteen. Esimerkiksi tehtävässä ”Tiedon ja informaation jakaminen” löydettiin mahdollisuus ”Tietoa voidaan visualisoida ja esittää erilaisilla käyttöliittymillä”, jonka todettiin olevan ehdottomasti tarvittava ominaisuus ja siten suoraan muutettavissa vaatimukseksi. Näistä liiketoiminnan tarpeista saatiin yleisiä toiminnallisia vaatimuksia mutta ei niinkään rajoituksia. Näitä rajoittavia vaatimuksia ilmeni tutkittaessa automaatiojärjestelmän tuomia vaatimuksia, joita käsitellään tarkemmin kappaleessa 3.3.2.

3.3 Vaatimukset

3.3.1 Liiketoiminnan vaatimukset

Tavoitteiden kautta löydettyjen tarpeiden avulla saatiin vaatimuksia, jotka ratkaisun pitäisi täyttää liiketoiminnan näkökulmasta. Neljä suurinta teemaa, joita käsiteltiin myös tarpeiden yhteydessä, ovat suoraan käytettävissä myös vaatimuksien yhteydessä. Löydetty vaatimukset on esitelty tässä raportissa yleisellä tasolla ja ryhmitelty pääteemojen mukaisesti. Kuitenkin, kaikkien uusien ratkaisuiden vaadittiin olevan joustavia ja helposti muokattavia, käyttäen mahdollisimman paljon standardiratkaisuita tai yleisesti käytössä olevia teknologioita.

Tiedon ollessa digitalisaation mahdollistaja, tärkeänä vaatimuksena liiketoiminnan näkökulmasta on ylipäätään saada tieto järjestelmistä hyödynnettäväksi. Tätä varten tarvitaan ratkaisu tiedon keräämiseen ja siirtämiseen. Tiedon sijaitessa asiakkaan automaatiojärjestelmissä ja muusta maailmasta eristetyissä tehtaissa, tarvitaan uusia teknisiä ratkaisuja niin sovellustasolla ja laitteistotasolla. Näiden ratkaisuiden pitää pystyä liittymään niin ohjausjärjestelmään kuin automaatiojärjestelmässä sijaitsevaan tietotekniseen infrastruktuuriin ja kerätä sekä siirtää sieltä tarvittavaa tietoa. Tietoa pitää saada kerättyä vertailukelpoisena eri järjestelmien välillä ja asiakaskohtaisen räätälöinnin sijaan, ratkaisuiden tulisi olla valmiiksi joustavia, soveltuen ilman lisätyön tekemistä tai muutoksia, mahdollisimman moneen asiakasjärjestelmään. Kun tieto on saatu kerättyä, pitää se saada siirrettyä ja tallennettua keskitettyyn tietovarastoon helposti käytettäväksi. Vanha konnektiviteettiratkaisu pitäisi korvata siten, että se mahdollistaisi joustavamman sekä yksinkertaisemmän tavan yhteyksien luomiseen, että tiedon siirtämisen järjestelmien ulkopuolelle, sillä vanha konnektiviteettiratkaisu oli yksisuuntainen, vain tarvittaessa muodostettava ja tietoturvamielessä joustamaton. Asiakkaat olivat myös hyvin varauksellisia ulkopuolelta sisäänpäin avattavan yhteyden luomisesta, mikä vaati, että asiakkaat avaisivat mahdollisuuden luoda verkkoyhteyksiä omaan verkkoympäristöönsä internetistä.

Kun tieto on saatu asiakasjärjestelmistä keskitettyyn tietovarastoon, vaaditaan ratkaisu, jonka avulla tiedon analysointi ja jalostus informaatioksi on mahdollista. Kerättyä tietoa pitää olla mahdollista valvoa sekä analysoida myös reaaliaikaisesti ja löytää siitä mahdollisia trendejä tai poikkeavia arvoja, joiden perusteella voidaan reagoida esimerkiksi elinkaari palveluiden asiakastuen toimesta, ennen kuin ongelmat ilmenevät. Tärkeänä osana analysointia on vertailu muihin järjestelmiin, joten tietoa pitää pystyä normalisoimaan vertailun mahdollistamiseksi, järjestelmien ollessa käytännössä jokaisessa tapauksessa uniikkeja kokoonpanoltaan. Ratkaisuiden pitäisikin mukautua automaatiojärjestelmän ominaisuuksiin muun muassa yleistämällä tietoa tarpeen mukaan. Tulevaisuudessa analysointiin ja tiedon jalostamiseen pitää voida käyttää uusia menetelmiä kuten koneoppimista tai tekoälyä.

Tiedon jakamisen ja hyödyntämisen pitää olla helppoa ja sen pitää olla mahdollista kaikille sidosryhmille. Tarpeita tulee olemaan hyvinkin erilaisia ja edes kaikkia sidosryhmiä ei vielä tässä vaiheessa voida tunnistaa, joten ratkaisun pitää pystyä tarjoamaan jalostettua tietoa eri muodoissa hyvinkin joustavasti. Tiedon ja informaation pitää olla ajankohtaista ja paikkansa pitävää ollakseen hyödyllistä. Pääsy tietoon pitää mahdollistaa kaikille sitä tarvitseville, fyysisestä sijainnista riippumatta, mutta tietoa ja sen näkyvyyttä on myös pystyttävä rajaamaan, sillä kaikkien ei tarvitse nähdä kaikkea kerättyä tietoa tai jalostettua informaatiota. Kolmansille osapuolille voidaan antaa pääsy tietoon erilaisten loppukäyttäjäsovelluksien kautta, jolloin mahdollistetaan myös kolmansien osapuolien tiedon hyödyntämisen.

Yhtenä hyvin tärkeänä vaatimuksena oli myös loppukäyttäjäpalveluiden kehittäminen ja uusien toimintamallien hyväksi käyttäminen. Yrityksen toiminnan joustavuuden ja kehitysnopeuden lisäämiseksi, haluttiin ottaa käyttöön ketteriä toimintamalleja niin liiketoiminnallisella kuin teknisellä puolella. Näin sisäiseen tai ulkoiseen käyttöön tarkoitettuja työkaluja, tuotteita tai palveluita olisi tehokkaampaa tuottaa. Käytettyjen ratkaisuiden pitäisi siis tukea inkrementaalista, sykleissä tapahtuvaa kehitystä, joissa ensimmäiset, hyvinkin yksinkertaiset kehitysversiot saadaan tehtyä hyvinkin nopeasti ja joiden oikeellisuus voidaan validoida palautteen avulla. Myös iteratiivista ja hyvinkin kokeilevaa kehitystä pitäisi olla mahdollista tehdä siten, että erilaisten ratkaisuiden toteuttaminen ja mahdollinen hylkääminen olisi vaivatonta ja kehitysprosesseja tukevia. Joustavuutta ja ketteryyttä vaaditaan myös, jos muutoksia nähdään tarpeelliseksi tehdä, esimerkiksi uusien palveluideoiden myötä, pitäisi tarvittavat muutokset olla mahdollista tehdä jo kentällä oleviin laitteisiin ja siten mahdollistaa myös olemassa olevan laitekannan hyödyntämisen.

3.3.2 Joustavan valmistusjärjestelmän erityisvaatimukset

Joustavat valmistusjärjestelmät itsessään toteuttavat monia digitalisaation peruseriaatteita sekä vaatimuksia, etenkin tiedon ja sen käsittelyn suhteen. Haasteita kuitenkin tuottavat toimintaympäristöt, joissa joustavat valmistusjärjestelmät toimivat. Raskaalla, metallin koneistamiseen keskittyneellä teollisuudella on pitkä ja tunnettu historia asioista, missä teollisuudenala on mullistunut täysin, mutta ympäristö ei ole muuttunut niin paljoa. Esimerkiksi tietokoneella ohjelmoitavat laitteet, eli CNC-valmistuslaitteet (Computer Numerical Control), muuttivat täysin tuotantotavat, miten kappaleita työstetään, mutta edelleen tehtaissa oli halleja täynnä yksittäisiä tuotantolaitteita ja niitä käyttämässä ihmisoperaattoreita. Edelleen, siitä huolimatta, että yksittäisiä laitteita on mahdollisesti kytkeyty toisiinsa esimerkiksi joustavan valmistusjärjestelmän tai tuotantolinjan muodossa, ovat kokonaisuudet edelleen erillisiä saarekkeitaan tehtaiden sisällä. Tästä syystä esimerkiksi tietoliikenneyhteyksien luominen laitteisiin voi olla hyvin haastavaa koska yksinkertaisesti tehtaassa ei välttämättä ole vaadittavaa verkkoinfrastruktuuria olemassa jonka avulla laitteet voisivat liittyä toisiinsa tai internetiin. Tätä varten järjestelmän olisi hyvä itse toteuttaa yhteydet esimerkiksi suoraan internetiin. Itse tiedon kerääminen voi olla

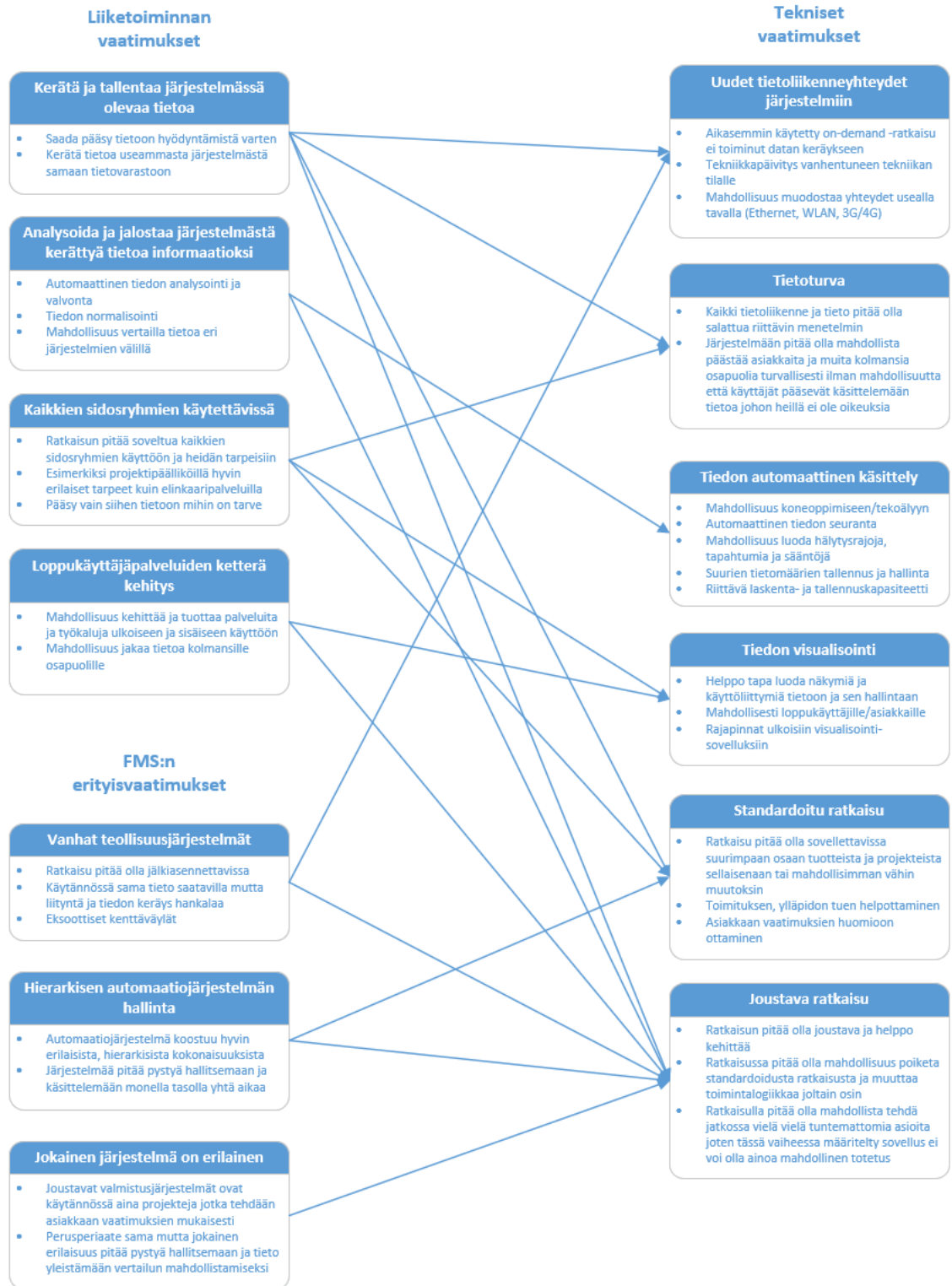
haastavaa siitäkkin syystä, että teollisuuslaitteiden- ja järjestelmien elinkaaret ovat niin pitkiä, voi käytössä olla hyvinkin laaja kirjo vanhojakin tiedonsiirtotapoja esimerkiksi erilaisten kenttävyölylien muodossa. Käytännössä myös vanhimmissa valmistusjärjestelmissä on samaa, tai samanlaista, tietoa saatavilla, esimerkiksi tuotantolukujen muodossa. Jos ja kun ratkaisun halutaan olevan jälkiasennettavissa myös jo olemassa oleviin laitteisiin, pitää varautua tiedon muuttamiseen muodosta ja väylästä toiseen.

Joustava valmistusjärjestelmä on yleensä hyvin monimutkainen systeemi, jota pelkästään fyysisen rakenteensakin puolesta voidaan tutkia hyvin monella tasolla. Koko järjestelmää voidaan kuvata todella korkealla tasolla, esimerkiksi yhtenä laitteena, jonka tehtävänä on valmistaa erilaisia kappaleita. Esimerkkejä tällä tasolla seurattavasta tiedosta on esimerkiksi kappaleiden tuotantoluvut sekä saatavuus- ja käyttöaste. Tästä alaspäin rakenteessa voidaan mennä järjestelmän laitetasolle, mitä jokainen laite ja osakokonaisuus ovat tekemässä, esimerkkinä yhden työstökoneen tekemä työvaihe ja sen seuranta. Rakennetta voidaan mennä hyvinkin syvälle sekä komponenttitasolle, jossa seurataan yhden kokonaisuuden, kuten kuljettimen toimintaa, että viimeisenä yksittäisen osan tai sensorin ja niiden tuottaman tiedon tarkasteluun. Koska järjestelmää voidaan tutkia useammalla tasolla, pitäisi olla mahdollista mallintaa näitä tasoja myös tietoa tallentaessa ja käsiteltäessä. Tätä hierarkkista rakennetta pitää pystyä tarvittaessa yleistämään siten, että eri järjestelmät ovat keskenään vertailtavissa.

Joustavat valmistusjärjestelmät ovat käytännössä aina asiakkaan tuotantoon ja tarpeisiin erikseen suunniteltuja projekteja. Näin ollen lähes jokainen järjestelmä on erilainen mikä tuottaa haasteita esimerkiksi mallintamisen suhteen. Kompleksisen järjestelmän mallintaminen ja kokonaisvaltaisen tiedonkeruun toteuttaminen myös vievät paljon aikaa, joten käytännössä järjestelmistä tullaan saamaan erilaisia datapisteitä ajan saatossa. Tiedon tallennuksen ja analysoinnin aikana tämä pitää ottaa huomioon esimerkiksi yleistämällä tietoa ja muutenkin toimia joustavasti esimerkiksi rakenteiden ja tietomallien luonnissa ja hallinnassa. Perustoiminallisuudet ovat kuitenkin käytännössä samat järjestelmästä toiseen, mutta osassa järjestelmiä myös näitä on vahvasti räätälöity asiakkaan tarpeisiin. Yksittäistapaukset ja poikkeukset tulevat olemaan hyvinkin yleisiä mutta käytettyjen ratkaisujen pitäisi vähintään mahdollistaa näiden todella erilaisten järjestelmien tiedonkeruu ja tiedon esittäminen mahdollisesti karsien analyysitoiminnallisuuksista. Toiminta pitää varmistaa siten, että voidaan aina esittää yksinkertaisempaa tai yleisempää tietoa jos järjestelmästä ei jostain syystä saada haluttua tietoa kerättyä. Esimerkkinä tuotantojärjestelmässä voi olla asiakaskohtaiset laitteiden tilat ja nimet näille tiloille. Tällaisessa järjestelmässä ei voida esimerkiksi tilojen perusteella kovin tarkkaa analyysiä tai vertailua tehdä, mutta järjestelmän nykyinen tila, räätälöityine tiloineen voidaan esittää esimerkiksi loppuasiakkaalle mobiilisovelluksen kautta, jolloin myös raakatiedolla voi olla arvoa lisäävää merkitystä.

3.3.3 Tekniset vaatimukset

Fastemsin liiketoiminnalta kerättyjen tavoitteiden pohjalta, vaatimuksiksi muunnetut tarpeet sekä joustavan valmistusjärjestelmän erityisvaatimukset kirjoitettiin teknisiksi vaatimuksiksi. Nämä tekniset vaatimukset antavat suoria reunaehdoja, joiden sisällä uuden ratkaisun pitää toimia ja mitä toiminnallisuuksia ratkaisun pitää toteuttaa. Kuvassa 6 on esitetty kaavio teknisten vaatimusten muodostuksesta ja riippuvuussuhteista liiketoiminnan ja joustavan valmistusjärjestelmän vaatimuksiin.

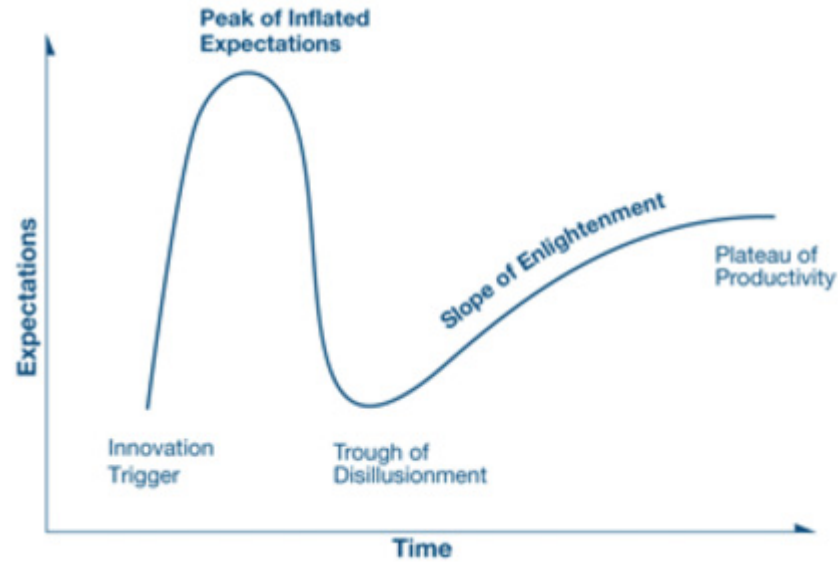


Kuva 6. Kaavio teknisten vaatimusten muodostuksesta ja riippuvuussuhteista

Kaaviossa on listattu tärkeimpiä vaatimuksia kasattuna ylätason otsikkojen alle. Tärkeimmät löydökset olivat seuraavat:

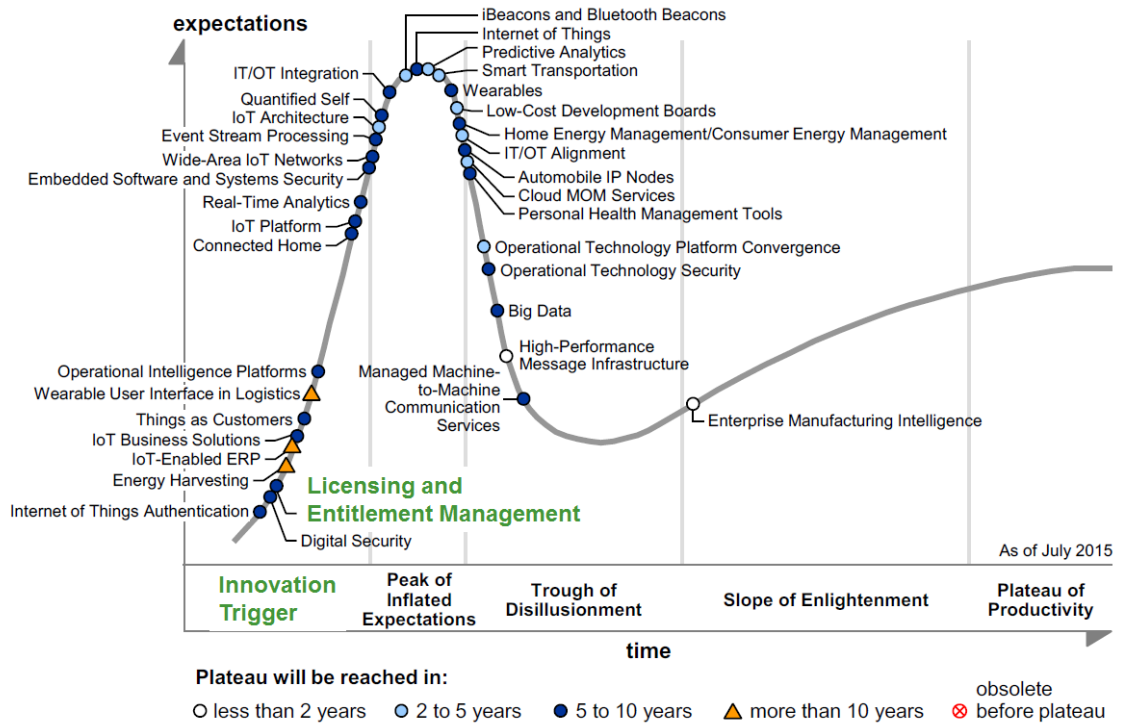
- Järjestelmiä varten pitää luoda uusi, kyvykkäämpi konnektiviteettiratkaisu, mikä mahdollistaa tiedonkeruuseen tarvittavat yhteydet myös haastavissa kohteissa ja puutteellisissa verkkoympäristöissä.
- Käytettyjen ratkaisujen pitää olla todistetusti tietoturvallisia käyttäen riittäviä suojausmenetelmiä, kuten salaustekniikoita ja laajoja käyttöoikeuksien hallintaominaisuuksia.
- Tietoa pitää pystyä käsittelemään automaattisesti suuriakin määriä ja tehdä erilaisia toimia myös reaaliajassa tiedon saapuessa järjestelmään, esimerkiksi erilaisten hälytyksien muodossa.
- Tiedon esittämiseen ja visualisoimiseen pitää olla helposti käytettäviä työkaluja siten, että niitä voidaan mahdollisesti käyttää myös loppukäyttäjäsovelluksien tarjoamiseen asiakkaalle. Myös ulkoiset rajapinnat tiedon hakemiseen oli ehdoton vaatimus.
- Ratkaisun pitää olla standardi siten, ettei se muutu järjestelmien välillä oleellisesti muun muassa toimituksen yksinkertaistamiseksi, ylläpidon minimoimiseksi sekä tuen antamisen helpottamiseksi.
- Ratkaisun pitää olla kuitenkin joustava siten, että muutoksien tekeminen ja uusien asioiden toteutus ja julkaisu ovat helppoa ja nopeaa. Mahdollisesti standardoidusta ratkaisusta pitää voida poiketa ja muuttaa toimintalogiikkaa normaalista ilman, että ratkaisun muu toiminta häiriintyy.

Teknisiä vaatimuksia muodostaessa pohdittiin jo alustavia teknisiä ratkaisuita ja yleisesitkin vaadittuja tai määriteltyjä teknologioita. Oli tiedossa, että liiketoiminnan tavoitteiden määrittelyssä oli käytetty niin sanottuja hype-käyriä, joilla voidaan esittää teknologian kehityskaari ja -nopeus tietyllä ajanhetkellä. Koska näitä teknologioita oli listattuna tavoitteissa, niin päätettiin myös käyttää hype-käyriä teknologioiden ja teknisten ratkaisuiden pidemmän aikavälin arvioitiin. Käytännössä katsottiin käyrien avulla, kannattaako tiettyjä, mahdollisesti liiketoiminnan tavoittelemia teknologioita vielä ottaa käyttöön vai onko parempi odottaa vielä sen kypsymistä sekä vakiintumista. Esimerkki hype-käyrästä on nähtävissä kuvassa 7, missä on esitetty käyrän perusrakenne ja sen komponentit.



Kuva 7. Hype-käyrän perusrakenne (Gartner n.d.)

Hype-käyrä on kaavio missä y-akselilla on teknologialle asetetut vaatimukset ja x-akselilla aika. Itse käyrä on aina saman mallinen ja on jaettu viiteen osaan: innovaation alkuaika, jolloin teknologian odotukset nousevat todella korkealle, kunnes saavutetaan odotuksien huippupiste. Tämän jälkeen odotukset laskevat, kun teknologia alkaa osoittaa saavutuksiaan todellisuudessa. Harvoin teknologia vastaa suurimpia odotuksia mutta myöhemmin teknologian kehittyessä ja uusien oivalluksien ilmetessä alkaa niin sanottu valaistumisen aika. Tämän osion aikana teknologiasta saattaa ilmestyä useampikin sukupolvi, kunnes saavutetaan hetki, jolloin teknologia on tarpeeksi kehittynyt käytettäväksi tuotantotarkoituksiin ja teknologiaa omaksutaan laajemmin. Hype-käyrät perustuvat asiantuntijoiden parhaisiin arvioihin teknologian kehittymisestä ja tarjoaakin helpon tavan nähdä oletettu teknologian kehityskaari. Käyriä voidaan muodostaa monenlaisista teknologioista ja kuvassa 8 on esitetty yksi projektin aikana käytetyistä hype-käyristä yleisien IoT-teknologioiden ja ajatusmallien arviointiin. (Gartner n.d.).



Kuva 8. Hype-käyrä yleisien IoT-tekniologioiden maturiteettien arviointiin (Gartner, 2015)

Yksi esimerkki tuloksista, joita hype-käyriä hyödyntäessä saatiin, oli päätös siirtää koneälyn hyödyntämisen vaatimus toiveeksi siten, että on etu, jos alustassa on mahdollisuus tulevaisuudessa hyödyntää koneoppimisen sovellutuksia. Päätöksen teon aikana koneoppiminen oli vielä hype-käyrän aivan alkupäässä, mutta sen ennustettiin jo silloin siirtyvän käyrällä eteenpäin todella nopeasti. Tähän haluttiin varautua, että tulevaisuudessa käyttöönotto olisi helpompaa, mutta alusta asti ominaisuuksien ei tarvitsisi olla saatavilla.

4. IOT-ALUSTAT

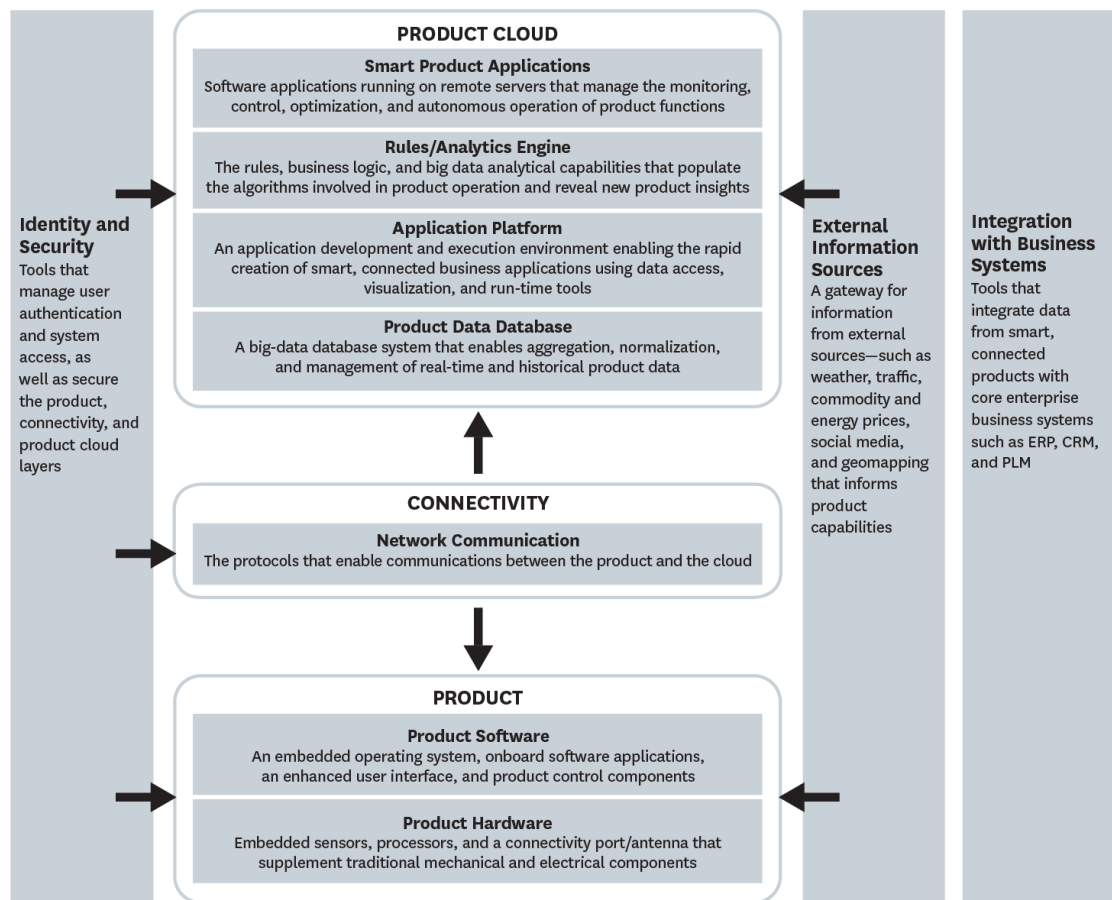
4.1 IoT-alustan määritelmä

IoT-alusta on tietotekninen järjestelmä mikä sisältää toiminnallisuudet kerätä, siirtää, hallita, käsitellä ja esittää tietoa. Alustan voi mieltää myös integroiduksi sovelluskehikseksi, mikä mahdollistaa IoT-sovelluksien kehittämisen alustan sisällä hyödyntäen alustan tarjoamia ominaisuuksia. Tarkoituksena on kerätä mahdollisimman paljon IoT-kehityksessä tarvittavia työkaluja yhden järjestelmän alle, mikä vähentää tarvetta toteuttaa toiminnallisuutta monessa eri järjestelmässä ja poistaa arvoa lisäämätöntä työtä. (Matthew, 2016).

IoT-alustat mahdollistavat nopeamman ja joustavamman sovelluskehityksen kuin perinteiset, erillisistä toiminnallisuuksista kasatut IoT-sovellukset. Alustan itsessään ollessa integroitu kokonaisuus, integroituu se myös saumattomasti ulkoisiin tietolähteisiin kuten tiedonkeruujärjestelmiin sekä yrityksiin ERP- ja CRM-järjestelmiin. Alustoissa useimmiten on myös panostettu yhteistyöhön niin sitä käyttävän yrityksen sisällä sekä sen ulkopuolella esimerkiksi erilaisten ekosysteemien kautta. (Porter & Heppelmann 2014).

4.2 Ominaisuudet ja erityispiirteet

IoT:n soveltamisalan ollessa todella laaja, tuottaa se haasteita myös IoT-alustoille niin toiminnallisuuden kuin teknologioiden suhteen. Yleisluontoisen sovelluksen luominen on käytännössä mahdotonta niin, että se kattaisi ja olisi käytettävissä kaikilla soveltamisalueilla, kun internetiin kytketään kaikkea kokonaisista kaupungeista siellä kulkeviin autoihin. Kuitenkin kaikkien sovelluksien pohjalla tarvitaan samoja perusasioita kuten tiedon siirtoa, tallennusta ja oikeushallintaa. Nämä toiminnallisuudet voidaan esittää niin sanotussa pinomuodossa missä jokaisen ylemmän osion toteuttamiseen vaaditaan alempien osioiden toiminnallisuus. (Porter & Heppelmann 2014). Yleinen teknologiapino IoT-sovellukselle on nähtävissä kuvassa 9. IoT-sovelluksen pitäisi täyttää ainakin nämä perusvaatimukset, jotta ylipäätään tietoa saadaan kerättyä ja hyödynnettyä IoT:n perusperiaatteiden mukaisesti.



Kuva 9. IoT-sovelluksen toiminnallisuus- ja teknologiavaatimuksia pinomuodossa (Porter & Heppelmann 2014).

Teknologiapinon mukaisesti itse tuotteessa, eli lähdejärjestelmässä, tarvitaan jotain fyysistä ja ohjelmallista toiminnallisuutta tiedon tuottamiseen ja sen keräämiseen. Tietoa voidaan tuottaa esimerkiksi erilaisten sensorien avulla. Tämä kerätty tieto pitää myös saada siirrettyä keskitettyyn tietovarastoon eli tarvitaan jonkinlainen yhteystapa eli konnektiiviteetti mikä useimmiten on jonkinlainen verkkoyhteys internetin ylitse. Tietokantaan tallennetun tiedon käsittelyä varten alustoissa on erilaisia työkaluja kuten analysointityökaluja ja sääntömoottoreita, jotka voivat esimerkiksi reagoida normaalista poikkeavaan tietoon. Tietoa myös pystytään jalostamaan informaatioksi muokkaamalla, yhdistelemällä siihen muuta tietoa myös muista tietojärjestelmistä. Itse verkotettu tuote ei useimmiten riitä ainoaksi tiedon lähteeksi, joten tietoa tarvitaan myös ulkoisista lähteistä kuten ERP-järjestelmistä tai toisista laitteista. Alustassa tarvitsee siis olla keinoja ulkoisten tietolähteiden liittämiseen ja integroimiseen osaksi järjestelmää. Alustoissa on yleensä myös kyvykkyydet esittää informaatiota ja visualisoida tietoa erilaisten käyttöliittymien ja raporttien avulla tai jakaa tietoa toisille järjestelmille ulkoisien rajapintojen kautta. Tietoturva on ehdoton vaatimus ja siihen pitääkin kiinnittää erityistä huomiota, sillä nyt jo IoT:n alkuaskeleilla on saatu ymmärrystä, millaista tuhoa internettiin kytketyillä, suojaamattomilla laitteilla voidaan tehdä esimerkiksi palvelunestohyökkäyksen muodossa. Tästä

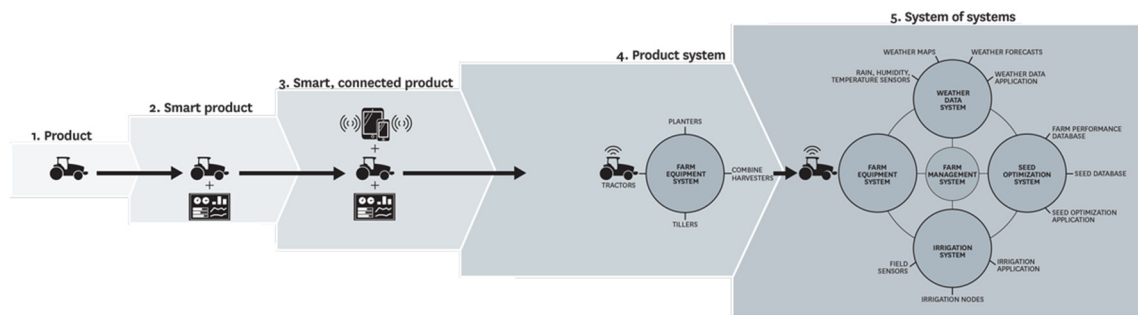
syystä tietoturva, pääsyn- ja oikeuksienhallinta onkin kuvassa 9 nähtävässä teknologiapi-
nossa esitettynä siten, että se kattaa koko pinon ja liittyy jokaisen pinon tasoon. (Porter &
Heppelmann 2014).

IoT-alustat usein eroavat perinteisestä sovelluskehityksestä, jossa sovellus tai osa siitä,
ohjelmoidaan lähes alusta asti. Alustoissa ohjelmoinnin määrä pyritään minimoimaan tar-
joamalla valmiiksi toteutetut perustoiminnallisuudet esimerkiksi tiedon siirtoon, tallen-
nukseen ja sen oikeus- ja pääsynhallintaan. Loppukäyttäjän, tässä tapauksessa alustaa
käyttävän yrityksen, liiketoimintalogiikka voidaan useimmiten konfiguroida järjestel-
mään ohjelmoinnin sijasta. Joissain alustoissa konfigurointiin voi olla myös graafisia työ-
kaluja esimerkiksi vuokaavion muodossa. Näin muutoksien tekeminen on helpompaa,
nopeampaa ja joustavampaa ja mahdollista kevyemmin resurssein kuin perinteisen ohjel-
mistokehityksen keinoin tehdyt muutokset. Usein myöskään IoT-alustoja ei suoranaisesti
räätälöidä asiakkaiden tarpeiden mukaan vaan on myös alustan kehittäjälle helpompaa ja
kannattavampaa tehdä konfiguroimisesta mahdollisimman joustavaa ja pitää tuote mah-
dollisimman standardina. Kun sovellusta ei tehdä asiakkaan tilauksesta, voidaan omaa
toimintaa skaalata tehokkaasti ja tarjota IoT-alustaa esimerkiksi palveluna, jolla sovellus-
toimittaja saa lisätulonlähteitä. (Mineraud et al. 2016).

IoT:ssä nimensä mukaisesti puhutaan esineiden internetistä missä kaikki laitteet verkot-
tuvat, joten järjestelmien skaalautuvuus on tärkeässä roolissa. Kytettävien laitteiden ja
tietovarastojen ja -lähteiden määrän kasvaessa ja näiden kytkeytyessä järjestelmiin näitä
pitää pystyä hallitsemaan niin tietoturvan kuin tietovirtojen suhteen. Myös teknologiat ja
algoritmit kehittyvät kiihtyvällä vauhdilla, joten laitteita pitää pystyä esimerkiksi päivittä-
mään ajan myötä. IoT-alustoissa on usein jo valmiina erilaisia laitteiden hallintaan liitty-
viä toiminnallisuuksia, joilla varmistetaan helppo skaalautuvuus laitteiden lisääntyessä.
Skaalautumiseen voi liittyä myös loppukäyttäjäsovellukset ja niiden käytön aiheuttama
kuorma tai tiedon säilytykseen käytettävä tallennustila. Tästä syystä, IoT-alustat hanki-
taan usein pilvipalveluina, joissa tehon ja tallennustilan skaalautuminen on käytännössä
rajatonta ja poistaa teho- ja tilarajoitukset sovelluksilta. (Matthew, 2016).

IoT-alustojen yhteydessä puhutaan usein ekosysteemeistä eli kyseessä olevan alustaan,
sovellustoimittajaan tai loppukäyttäjän sovellusalueeseen liittyvistä kolmansista osapuol-
ista, jotka muodostavat yhdessä uutta liiketoimintaa. (Mineraud et al. 2016). Ekosys-
teemi voi muodostua ja toimia usealla eri tasolla tai liiketoiminta-alueella mutta kaikilla
on sama päämäärä eli saada hyötyä toisistaan. Esimerkiksi joku kolmannen osapuolen
yritys voi tarjota tuki-, ylläpito tai konsultointipalveluita alustaan liittyen tai sovellustoi-
mittaja voi tarjota alustalle sovelluskaupan, jossa kuka tahansa voi myydä esimerkiksi
tekemiään laajennuksia sovellustoimittajan ottaessa osuutensa jokaisesta kaupasta. Näin
hyötyvät sekä sovellustoimittaja että kolmannet osapuolet. (Porter & Heppelmann 2014).
Toinen taso, jolle ekosysteemejä muodostuu, on toimitusketjut ja laitteistoja valmistavat
yritykset. Erilaiset sensori- ja tiedonkeruulaitteistovalmistajat voivat integroida alustan
toiminnallisuudet omaan laitteeseensa ja näin saada alustan käytön avulla lisättyä omaa

laitemyyntiään. Myös konnektiviteettiä voidaan tarjota kumppanina, esimerkiksi teleoperaattori voi tarjota mobiiliyhteydet IoT-sovelluksen tiedonsiirtoon. (Mazhelis et al. 2012). Myös tiedon jakamisen ja hyödyntämisen ympärille muodostuu ekoympäristöjä. Yksinkertainen esimerkki tiedon jakamisesta voisi olla yritys, joka tarjoaa lämpötilojen historiatietoa maanviljelyn optimointiin. Nämä tiedot voitaisiin kerätä traktorivalmistajan älykkäistä traktoreista, jotka mittaavat lämpötilaa maataloilla. Näitä kaikkia hyödyttäviä tiedonjakoverkostoja kutsutaan järjestelmien järjestelmiksi (engl. system of systems). Kuvassa 10 on havainnollistettu yhdenlaista järjestelmien järjestelmää. Kaaviossa vasemmalla on tuote, tässä tapauksessa traktori, johon lisätään sensoreita, jotka keräävät tietoa ja tuotteesta tulee älykäs. Kun älykkääseen tuotteeseen lisätään konnektiviteetti, saadaan älykäs ja yhteyksissä oleva tuote, jonka keräämä tieto lähetetään eteenpäin. Kun traktorin keräämää hyödynnetään esimerkiksi maatilan hallintajärjestelmässä, mikä hyvin voisi olla IoT-alusta, saadaan yksittäinen tuotesysteemi. Kun hallintasynteen tietoa viedään ja sinne lisätään sekä yhdistellään tietoa muista lähteistä, mahdollisesti ekosysteemin jäseniltä tai toisista IoT-alustoista, saadaan järjestelmien järjestelmä. (Porter & Heppelmann 2014).



Kuva 10. Esimerkki systeemien systeemistä (Porter & Heppelmann 2014).

4.3 IoT-alustat teollisessa ympäristössä

IoT on käsitteenä ja ilmiönä hyvin kattava, joten IoT-alustojen tarjonta kohdistuu myös useaan eri kohdesegmenttiin. Teollisen internetin ollessa vain pieni osa digitalisaatiota mikä kohdistuu yrityksiin, iso osa markkinoilla olevista IoT-alustoista onkin tarkoitettu muun muassa kuluttajien käyttöön. Tämä näkyy tiettyinä näkökulmina, miten alustaa käytetään ja mitä sillä on tarkoitus tehdä. (Perry 2016). Kuvassa 11 on nähtävä vertailutaulukko kuluttajatasen sekä teollisen tason IoT-alustojen eroista.

	Consumer IoT	Industrial IoT
Examples	Wearables, devices, B2C (business-to-consumer) products	Automotive, agriculture, aerospace, military, manufacturing, automated factory
Primary value	Contained to the physical product	Embedded software, apps, generated data, and responsiveness
Availability and scalability	Low concern	High concern
Security and reliability	Moderate concern	High concern
Connectivity	Unidirectional, data to cloud; low frequency	Bidirectional for data transmission/remote access control/automation; high frequency

Kuva 11. Kuluttajille tarkoitetun asioiden internetin sekä yritystason teollisen internetin vertailua (Perry 2016)

Kuluttajille tarkoitetussa IoT-alustassa ja myös IoT:ssä ylipäättään lisäarvo yleensä rajoittuu itse fyysiseen tuotteeseen, johon tuodaan lisäarvoa esimerkiksi etäpäivityksien muodossa. Teollisen internetin puolella lisäarvo muodostuu sovelluksista, kerätyn datan analytiikasta ja käytöstä päätöksenteon apuna sekä reagointikyvyn paranemisesta esimerkiksi ongelmien ilmetessä. Myös saavutettavuus, laajennettavuus, tietoturva sekä luotettavuus ovat usein yrityksille suuria haasteita mutta kuluttajapuolella nämä eivät ole ensimmäisiä huolenaiheita. (Perry 2016). Valitettavasti tähän mennessä jo on nähty esimerkkejä siitä, mitä vaikutuksia kuluttajapuolen tietoturva-asiat sivuuttaessa voi ilmetä. Maailmalla on ollut suuriakin yrityksiä haitanneita verkkohyökkäyksiä, joissa apuna on käytetty huonosti turvattuja IoT-laitteita. Yrityksillä tietoturva on onneksi lähtökohtaisesti korkealla prioriteetilla, sillä siirrettävä tieto saattaa sisältää yrityssalaisuuden alaista tietoa.

Kuluttajapuolella kerättävä tieto usein on yksisuuntaista tiedon lähettämistä pilveen. Esimerkkinä älykkään tuotteen sensoreista kerättyjä raaka-arvoja lähetetään pilveen käsiteltäväksi. Yleensä tietoa lähetetään hyvin harvoin, jossain tapauksissa puhutaan minuuteista tai tunneista, useimmiten tiedonsiirtokustannuksien minimoimiseksi sekä virrankulutuksen alentamiseksi. Yrityspuolella ja etenkin teollisuudessa tietoa siirretään molempiin suuntiin, esimerkiksi etäohjauksen muodossa, jolloin myös kerätylle tiedolle aset-

taan vaatimuksia muun muassa reaaliaikaisuuden suhteen. Jos tietoa saadaan liian harvoin, menettää se nopeasti merkityksensä, jos sen perusteella ei voida tehdä päätelmiä nykyisestä tilasta. (Perry 2016).

Esitellyistä syistä teollisuudessa ja yleisestikin yrityksissä IoT-alustat ovat suosittuja digitalisaation mahdollistajia, sillä nämä mahdollistavat nopean kehityksen, valmiit teknologiat ja yrityksiä arvostamat tuki- ja konsultointimallit. IoT-alustat myös erikoistuvat näihin nopeasti kasvaviin teollisuudenaloihin näiden tarjoaman todella suuren potentiaalin takia, kun aikaisemmin eristetyt teollisuuslaitokset kaikkine kriittisine laitteineen ovat IoT:n hyödyntämisen mahdollisuuksia täynnä. Uudet ideologiat kuten älykäs valmistus (engl. smart manufacturing), missä pyritään lisäämään tai parantamaan tuotantoa tiedon avulla, sekä lisäävät työstömenetelmät (engl. additive manufacturing), kuten 3D-tulostus houkuttelevat alustatoimittajia kilpailemaan uusista markkinoista.

5. IOT-ALUSTOJEN VERTAILU

5.1 Vertailukriteerien muodostus

Vaatusmäärittelyssä löydetyistä teknisistä vaatimuksista muodostettiin alustojen vertailussa käytettävät kriteerit. Teknisiä vaatimuksia muutettaessa vertailukriteereiksi pohdittiin, mitkä asiat vaikuttavat vaatimuksia täyttäessä ja mitä yhteisiä ominaisuuksia alustoilla on, joita tarvittaessa voidaan yleistää vertailun mahdollistamiseksi. Monessa vertailukriteerissä mennään jo toteutustasolle, jossa haetaan teknisiä ratkaisuita, joilla jokainen alusta on toteuttanut vaatimuksenmukaiset ominaisuudet. Teknisten vaatimuksien lisäksi vertailukriteereihin lisättiin alustan tekniseen ylläpitoon ja myös kaupalliseen puoleen liittyviä asioita kuten eri ylläpitomallit ja kustannusarviot sekä -mallit. Kuvassa 12 on esitetty, miten vertailukriteerit ovat muodostettu teknisten vaatimuksien avulla ja ryhmitelty yleisellä tasolla.

Tekniset vaatimukset

Uudet tietoliikennetyhteudet järjestelmiin

- Aikasemmin käytetty on-demand-ratkaisu ei toiminut datan keräykseen
- Tekniikkapäivitys vanhentuneen tekniikan tilalle
- Mahdollisuus muodostaa yhteydet usealla tavalla (Ethernet, WLAN, 3G/4G)

Tietoturva

- Kaikki tietoliikenne ja tieto pitää olla salattua riittävän menetelmän
- Järjestelmään pitää olla mahdollista päästää asiakkaita ja muita kolmansia osapuolia turvallisesti ilman mahdollisuutta että käyttäjät pääsevät käsittelemään tietoa johon heillä ei ole oikeuksia

Tiedon automaattinen käsittely

- Mahdollisuus koneoppimiseen/tekoälyyn
- Automaattinen tiedon seuranta
- Mahdollisuus luoda hälytysrajoja, tapahtumia ja sääntöjä
- Suurien tietomäärien tallennus ja hallinta
- Riittävä laskenta- ja tallennuskapasiteetti

Tiedon visualisointi

- Helppo tapa luoda näkymiä ja käyttöliittymiä tietoon ja sen hallintaan
- Mahdollisesti loppukäyttäjille/asiakkaille
- Rajapinnat ulkoisiin visualisointi-sovelluksiin

Standardoitu ratkaisu

- Ratkaisu pitää olla sovellettavissa suurimpaan osaan tuotteista ja projekteista sellaisenaan tai mahdollisimman vähin muutoksin
- Toimituksen, ylläpidon tuen helpottaminen
- Asiakkaan vaatimusten huomioinnin ottaminen

Joustava ratkaisu

- Ratkaisun pitää olla joustava ja helppo kehittää
- Ratkaisussa pitää olla mahdollisuus poiketa standardoidusta ratkaisusta ja muuttaa toimintalogiikkaa joltain osin
- Ratkaisulla pitää olla mahdollista tehdä jatkossa vielä vielä tuntemattomia asioita joten tässä vaiheessa määritelty sovellus ei voi olla ainoa mahdollinen toteutus

Vertailukriteerit

Ekosysteemi

- Ohjelmistojen toimittajat
- Ekosysteemin laajuus
- Tukipalvelut
- Sovelluskauppa
- Referenssiasiakkaat
- Kokonaisratkaisu saatavilla

Tiedon keräys - Laitteisto

- Laitteisto saatavilla toimittajalta
- Konnektiveettivaihtoehdot
- Kenttäväyliiitynnät
- Muut liitynnät
- Muut sensorit
- Laitehallinta
- Etäyhteyksien muodostus
- Look-and-feel
- DIN-kiskoasennus

Tiedon keräys - Sovellus

- Valmiit sovellukset ja ohjelmointikirjastot
- OPC-protokollatuki
- Viestintäprotokollat
- Liitynnät ulkoisiin palveluihin
- Yhteyksien hallinta
- Tiedon tallennussijainti
- Turvallisuus

Tiedon käsittely ja analysointi

- Tiedon automaattinen käsittely
- Analytiikkatyökalut
- Koneoppiminen
- Analysoidun tiedon hyödyntäminen
- Tiedon siirtäminen alustan ulkopuolelle (migraatio)

Visualisointi

- Näkymien luonti
- Mobiililaitteet
- Staatittiset raportit

Palvelukehityksen nopeus ja toimintamallit

- Asennus ja konfigurointi
- Ohjelmoinnin määrä (vs. konfigurointi)
- Itsenäinen käyttö
- Tietomallien hallinta

Tietoturva ja oikeushallinta

- Multitenanttisuus
- Käyttäjä- ja oikeushallinta
- Fastestins pääsy tietoon yli asiakkaiden

Skaalautuvuus

- Sama ratkaisu sisäiseen ja sekä ulkoiseen käyttöön
- Usean asian tekeminen samalla ratkaisulla
- Suuren/kasvavan laitemäärän hallinta
- Pilvipalvelu vai lokaaliasennus
- Yleinen skaalautuvuus

Joustavuus

- Onnistuuko nyt määriteltyjen konseptien toteuttaminen
- Muutoksien tekeminen helppous
- Määritellyistä toteutuksista poikkeaminen
- Yleinen joustavuus

Kustannukset

- Kustannusmalli
- Aloituskustannukset
- Ylläpitokustannukset
- Käyttökustannukset/instanssi
- Käyttökustannukset/asiakas
- Käyttökustannukset/laitte
- Käyttökustannukset/käyttäjä

Kuva 12. Teknisten vaatimusten ja IoT-alustojen vertailukriteereiden riippuvuussuhteet

Vertailukriteereissä arvioitiin muun muassa sovelluksien toimittajia ja yksi Fastemsin vaatimuksia alustalle oli, että alustalla pitäisi olla mahdollisimman laaja ja aktiivinen ekosysteemi, tarkoittaen muun muassa laitevalmistajien ja muiden sovellustoimittajien kanssa tehtyä yhteistyötä, jonka lopputuloksena alusta on suoraan käytettävissä muiden valmistajien ratkaisujen kanssa. Yhteisön rakentamat laajennukset, avoin sovelluskauppa ja muut alustatalouden mahdollisuutta tarjoavat ominaisuudet olivat myös vertailun kohteena. Vertailussa kriteerinä oli myös, jos alustalla on jälleenmyyjä tai jakelija Suomessa. Tällöin tukipalveluilta pystyttäisiin käyttämään nopeammin ja helpommin paikallisesti kuin ulkomaisia toimijoita, joiden kanssa työskennellessä ongelmia saattaisi aiheuttaa muun muassa fyysinen etäisyys, kieliongelmat tai jopa aikaero. Myös alustaa käyttäviä referenssiyrityksiä vertailtiin ja katsottiin, onko mukana Fastemsiä vastaavia yrityksiä.

Tiedon keräyksen ominaisuuksia vertailtiin erikseen laitteiston ja konnektiviteetin sekä sovellustoimintojen osalta. Laitteiston saatavuuden lisäksi vertailtiin liitännäismahdollisuuksia eri järjestelmiin, kuten teollisuuden kenttäväyliin sekä mahdollisuutta muun tiedon keräämiseen esimerkiksi lisäämällä mittalaitteita ja muita antureita järjestelmään. Myös laitteiden hallintamahdollisuudet alustasta käsin olivat mielenkiintoinen kriteeri skaalautumisen hallinnan näkökulmasta. Tiedon siirtämisen osalta vertailtiin mahdollisuuksia, teknisiä ratkaisuita sekä tietoturva.

Tiedon käsittelystä vertailtiin kerätyn tiedon käsittelyn automaation astetta ja sitä, minäkalaiset analysointityökalut alustassa on. Myös jalostetun tiedon jatkokäsittelymahdollisuuksia vertailtiin kuten myös sitä, kuinka helposti tieto on vietävissä järjestelmästä ulos esimerkiksi alustan vaihdon yhteydessä. Pahimmassa tapauksessa jos migraatio järjestelmien välillä ei onnistu, voidaan kaikki kerätty tieto menettää, kun se joudutaan hylkäämään edelliseen järjestelmään. Myös visualisointimahdollisuuksia, kuten erilaisten näkymien tekemisen mahdollisuutta sekä helppoutta arviointiin ja erityistä huomiota kiinnitettiin mobiililaitteille suunnattuihin näkymiin.

Palvelukehityksen nopeus ja toimintamallit olivat hyvin tärkeät vertailukriteerit, sillä Fastemsin tavoitteena oli tehdä palvelukehitystä itsenäisesti mahdollisimman pitkälle ilman ulkopuolista apua. Tämän takia alustoja vertailtiin asennuksen ja konfiguroinnin sekä yleisen käytön haastavuuden suhteen kuten myös ohjelmoinnin määrää etenkin tietomallien hallinnassa. Nopean palvelukehityksen mahdollistamiseksi ulkopuolisen konsultoinnin tai ohjelmoinnin hankkimisen nähtiin olevan liian hidasta ja siksi yritykselle haluttiinkin kyvykkyydet tehdä nopeaa prototyypausta sekä ideoiden testausta yhdessä liiketoiminnan kanssa.

Koska järjestelmältä vaadittiin riittävän turvallisia teknologisia ratkaisuja, kyvykkyyttä koventaa järjestelmän tietoturva ja prosesseja siten arvioitiin. Vertailukriteerinä oli myös, että voidaanko samaan järjestelmään tallentaa kaikkien eri asiakkaiden tietoa ja myös antaa näiden asiakkaiden käyttäjille pääsy järjestelmään turvallisesti. Oikeuksien

hallinnan haluttiin olevan vaivatonta ja tästä syystä järjestelmässä pitäisi olla mahdollisimman monella tasolla hallittavat oikeudet. Oikeushallintaa vertailtiin siten, että niitä olisi hyvä pystyä hallitsemaan ainakin yritystasolla, ryhmätasolla sekä käyttäjätasolla, päämääränä siten, että pääsy tietoon säilyisi vain niillä henkilöillä, joilla se pitäisikin olla.

Alun perin Fastemsin IoT-työryhmän vähäisien resurssien johdosta alustan haluttiin skaalautuvan mahdollisimman tehokkaasti ja pienin ponnistuksin. Jos esimerkiksi jokainen lisätty asiakkuus tai kytketty järjestelmä vaatisi huomattavan paljon käsityötä, olisi alustan ylläpito ja päivittäinen käyttö raskasta ja vaatisi resursseja, joita ei ainakaan alkuvaiheessa ole saatavilla. Tämä voi pahimmassa tapauksessa johtaa ratkaisun laajemman käyttöönoton esteeksi tai ainakin hidastaa sitä. Vaarana olisi myös alustan hyödyntämisen vähäisyys, jos sen käyttö nähtäisiin liian monimutkaisena tai raskaana. Tästä syystä erityistä huomiota kiinnitettiin oikeushallinnan ja kytkettyjen laitteiden sisäisten mallien luomisen automatisointimahdollisuuksiin, esimerkiksi toisista järjestelmistä kerätyn tiedon avulla. Automatisointi korvaa puuttuvia resursseja ja mahdollistaa nopeamman toiminnan laajentamisen, kun aikaa vievää ja arvoa lisäämätöntä käsityötä ei tarvitse tehdä.

Yksi tärkeimmistä vertailukriteereistä, joita ratkaisuiden kesken arvioitiin, oli niiden joustavuus. Tässä tapauksessa joustavuudella haettiin mahdollisuutta toteuttaa useita erilaisia asioita, kuten palveluita tai prosesseja. Joissain tapauksissa ratkaisut keskittyvät vain yhden prosessin ja ongelman ratkaisuun, mutta Fastemsin tapauksessa joustavuutta tarvittiin niin yksilöllisten projektitoimituksien hallitsemiseen kuin vielä tuntemattomien konseptien toteuttamiseen. Myös muutoksien tekemisen helppous oli yksi kriteeri, jonka avulla vertailtiin, kuinka työlästä määritelmästä ratkaisusta poikkeaminen on. Joustavuutta arvioitiin myös yleisellä tasolla yksittäisten vertailukriteerien lisäksi, että saataisiin helpommin vertailtavissa oleva käsitys joustavuudesta.

Vertailukriteerit kustannuksista eivät tulleet vaatimusmäärittelyn kautta, mutta ovat kuitenkin olennainen osa kaikkia hankintaprojekteja. Ratkaisuiden kustannuksia vertailtiin monella eri tasolla ja niiden kulurakenteita yritettiin yleistää mahdollisimman tehokkaasti vertailun helpottamiseksi. Ratkaisun käyttöönotossa saattaa tulla aloitusmaksuja, lisenssimaksuja ja asennuskuluja. Jos ratkaisu hankintaan palveluna, voi ratkaisulla olla myös ulkoisia ylläpitokustannuksia sisäisten kustannuksien lisäksi. Käyttökustannuksia yritettiin arvioida skaalautumisen näkökulmasta ja siksi kulut laskettiin yksiköittäin yhdelle instanssille, sillä jossain ratkaisuisissa instansseja tarvittiin useampi esimerkiksi testaus- ja kehitysympäristöiksi tai asiakkaiden eristämiseksi. Käyttökustannuksia arvioitiin yhdelle yksikölle myös asiakkaiden, laitteiden, käyttäjien sekä alustaan tallennetun tiedon määrän suhteen.

5.2 Alustojen vertailuprosessi

Alustojen vertailuprosessissa päämääränä oli muodostaa ylätasoinen ymmärrys tarjonnasta ja miten eri alustat ovat verrattavissa toisiinsa. Suurin osa vertailuun käytetystä ajasta

kuluikin vertailukriteereinä käytetyn tiedon etsimiseen kaikista vaihtoehdoista. Tätä tietoa kerättiin usealla eri menetelmällä. Suuri osa tiedosta löytyi vertailtavien vaihtoehtojen omista materiaaleista kuten toimittajan verkkosivuilta esimerkiksi markkinointimateriaalista tai tukimateriaalista dokumentaation sekä erilaisten ohjeiden muodossa. Kriteereihin, joihin ei löydetty tietoa itsenäisesti, haettiin vastauksia myös keskustelemalla ratkaisuiden toimittajien kanssa. Nämä toimittajat pitivät tuote-esittelyitä, vastailivat kysymyksiin sekä esittivät demoja, mahdollisesti referenssiyrityksiltä esimerkiksi vastaavilta teollisuudenaloilta. Tietoa saatiin myös referenssiyrityksiltä, joita alustoilla tiedettiin olevan ja myös ratkaisuiden toimittajat saattoivat järjestää referenssikeskusteluita, jotka koettiin hyvin hyödyllisiksi tiedon etsinnän aikana.

Tarkemman tietämyksen ja kokemusperäisen arvioinnin mahdollistamiseksi arvioinnin alussa oli päätetty, että prosessin aikana koekäytetään mahdollisimman montaa vaihtoehtoa. Koekäyttöä varten oli suunniteltuna pieni prosessi tilatiedon keräykseen, seuraamiseen ja virhetilanteeseen reagoimiseen esimerkiksi hälytyksen muodossa. Alustoja saatiin koekäyttöön toimittajilta tai avoimista kokeilujaksoista, joita monella alustajatoimittajalla oli tarjota suoraan, jopa ennen yhteydenottoa toimittajan myyntiorganisaatioon. Näistä kokeiluista saatiin tärkeää informaatiota etenkin käytön aloituksen sekä käytön helppouden suhteen. Kun käytössä oli vain alustan oma dokumentaatio tai kokeilujaksoon mahdollisesti sisältyvä itseopiskelutoiminto (engl. tutorial), nähtiin välittömästi, minkälaista alustan käyttö tulee olemaan ja kuinka paljon muun muassa ulkopuolista koulutusta tarvittaisiin alustan tehokkaaseen käyttöön. Kuitenkin, lopulta rajallisen ajan puitteissa ehdittiin koekäyttämään vain viittä eri ratkaisuvaihtoehtoa, jotka olivat PTC Thingworx, Cumulocity, Wapice IoT-Ticket, Microsoft Azure ja IBM Bluemix. Kokonaisuudessaan koekäyttöä varten suunniteltu testi tehtiin kuitenkin vain yhdellä, mutta lupaavimmalla alustavaihtoehdolla, PTC Thingworxillä. Muilla testatuilla alustoilla koekäyttö rajoittui esimerkkien seuraamiseen sekä valmiiden kirjastojen testailuun kytkemällä esimerkiksi Raspberry Pi -minitietokoneita alustoihin, konfiguroimalla ne lähettämään tietoa oikeiden tuotantojärjestelmien sijasta koekäytettyihin alustoihin.

Vaihtoehdoista löydettyä tietoa täytettiin vertailutaulukkoon, mikä sisälsi vertailukriteerit sekä vertailtavat vaihtoehdot matriisimuodossa. Tällä tavoin pystyttiin helposti muodostamaan kokonaiskuva alustojen ominaisuuksista ja pitämään helposti kirjaa mitä asioita on jo tutkittu ja mitä asioita pitää vielä selvittää. Tietoa merkattiin taulukkoon kohtuullisen vapaassa muodossa, joten tieto itsessään ei ollut läheskään aina suoraan vertailtavissa vaihtoehtojen välillä vaan vaihtoehtojen paremmuusjärjestys piti ratkaista jollain muulla tavalla.

Vaihtoehtojen keskinäiseen arviointiin käytettiin kappaleessa 2.5 esiteltyä Pughin matriisia. Pughin matriisin pohjana käytetään aikaisemmin täytettyä vertailutaulukkoa, josta valitaan yksi vaihtoehto vertailukohteeksi. Tämä vertailukohde valitaan empiirisesti sen mukaan, mikä alusta voisi olla lupaavin vaihtoehto valinnaksi tässä vaiheessa vertailua. Vertailukohteen valinnan jälkeen kaikkien muiden vaihtoehtojen sarakkeita verrataan

vertailukohteeseen yksi kerrallaan, ottaen huomioon vertailutaulukon vapaamuotoisen tiedon ilmaisutavan. Pughin matriisin käyttö poistaa ongelmat taulukon solujen vertailussa yksinkertaistamalla ilmaisutavan samanlaiseksi kaikkien solujen kesken.

5.3 Vaihtoehtojen kartoitus ja esivalinta

IoT-alustavaihtoehtojen kartoitus ja löytäminen ei tuottanut ongelmia, sillä alustojen tarjonta ja markkinat ovat nykyään hyvin kattavat. Uusia alustoja syntyy sekä poistuu markkinoilta viikoittain ja useat markkinoille pyrkijät ovatkin uusia startup-yrityksiä, jotka yrittävät läpimurtoa ruuhkaisella toimittajakentällä. Kuitenkin, kuten usein monessa muussakin tapauksessa, tietoteknisistä ratkaisuista halutaan valita vakaa ja pitkäikäinen ratkaisu, joten tässä kartoituksessa keskityttiin kahteen ryhmään: pitkään markkinoilla olleisiin ja jo hieman vakiintuneisiin alustoihin sekä valmistajiin.

Vaihtoehtoja kartoitettiin pääsääntöisesti yleisesti internetistä, alan julkaisuista sekä toimittajien yhteydenottojen perusteella. Tiedossa oli myös referenssiyrityksiä samoilta teollisuudenaloilta, joilla oli yksi tai useampi IoT-alusta käytössä. Kartoitusvaiheessa kuitenkin oli tarkoituksena löytää mahdollisimman paljon alustoja, joista valitaan vielä hallittavissa oleva määrä tarkempaan vertailuun. Kartoituksen tuloksena löydettiin kymmeniä mahdollisia alustoja mutta tarkempaan tarkasteluun valittiin 12 alustaa. Ensimmäiselle vertailukierrokselle valittiin tarkoituksella suuri määrä alustoja, joita tarkasteltiin hyvin yleisellä tasolla, sillä tiedettiin, että lähes jokaisessa alustassa oli hieman erilaiset ajattelumallit ja hyvin todennäköisesti tarkemmin tutkittuna useimpien alustojen ajatusmallit eivät tulisi sopimaan Fastemsin käyttötarkoituksiin.

Tärkeimmät esivalinnassa käytetyt kriteerit olivat, että alusta olisi joko täysin tai välillisesti teolliseen käyttöön sopiva. Kriteerin täyttämiseen riitti, jos esimerkiksi kandidaatin referensseissä oli mainittu teollisuuteen keskittyneitä yrityksiä. Valinnassa myös painotettiin alustoja, joilla oli suomalainen toimittaja, jakelija tai vähintään tukikumppani, jonka kanssa hankintakeskusteluita voitiin käydä. Myös pidempään markkinoilla olleita ja isompia toimijoita suosittiin esivalinnassa.

5.4 Esivalitut vaihtoehdot

5.4.1 PTC Thingworx

Thingworx on yksi pisimpään markkinoilla olleista alustoista. Alun perin se on aloittanut startup-yrityksenä vuonna 2009, joka sulautettiin yhdysvaltalaiseen PTC:hen yritysoston kautta vuonna 2013. Thingworx on ominaisuuksiltaan hyvin kattava ja se sisältääkin täysin integroidun IoT-teknologiapinon. Thingworxin pääsegmentti on eri teollisuuden alat.

Tiedon keräykseen ja siirtoon Thingworx tarjoaa hyvät lähtökohdat. Tarjolla on muun muassa omat sovelluskehityskirjastot (engl. software development kit, SDK) kaikille

yleisimmille ohjelmointikielille ja laaja tuki erilaisille tiedonsiirtoprotokollille esimerkeinä HTTP (Hypertext Transfer Protocol), MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol) ja OPC (Open Platform Communications). Thingworx tarjoaa myös valmiit työkalut alustan ulkopuolella sijaitseviin, jopa palomuurin takana oleviin tietokantoihin. Tällä tavalla voidaan helposti integroida muita yrityksen tietojärjestelmiä alustaan.

Tiedon tallennukseen Thingworxissa on sisäinen tietokanta, mikä ei suoraan ole käsiteltävissä vaan sen sijaan alustassa on tehokkaat toiminnallisuudet tiedon ja siihen liittyvien laitteiden mallinnukseen. Mallinnustapa on olio-ohjelmointia muistuttava, objektipohjainen, monitasoinen, periytyvä ja laajennettava, jolla pystyy hyvin helposti luomaan rakenteita ja niihin liittyvää tietoa sekä uudelleenkäyttämään jo olemassa olevaa toteutusta. Näihin objekteihin, thingeihin, voidaan myös liittää toiminnallisuutta Javascript-kielen avulla, millä pystytään laajentamaan alustan itse tarjoamia toiminnallisuuksia. Käytännössä kaikki alustan toiminnot ovat käytettävissä ohjelmallisesti, joten prosessien ja tehtävien automatisointi on mahdollista. PTC:llä on myös laaja ekoympäristö koostuen tuki- ja konsultointipalveluista, laitetoimittajista sekä sovelluskaupasta alustan toimintaa lisäävistä laajennuksista

Thingworxissa on sisäänrakennettuna työkalu, nimeltään mashup builder, yksinkertaisten käyttöliittymien tekemiseen ilman ohjelmointia. Mashup builder käyttää hyväkseen valmiita käyttöliittymäkomponentteja ja luotuja tietomalleja, jolloin haluttua tietoa voidaan yhdistää näytettäväksi helposti konfiguroimalla. Omia käyttöliittymäkomponentteja on mahdollista luoda ja tähänkin Thingworx tarjoaa valmiit kehitystyökalut. Jopa kevyiden loppukäyttäjäsovelluksien luominen onnistuu mashup builderilla, etenkin kun Thingworxin monipuolinen käyttäjäoikeuksien hallinta mahdollistaa loppukäyttäjille pääsyn antamisen järjestelmään. Pääsyä tietoon, näkyymiin ja yksittäisiin toimintoihin pystytään rajoittamaan ja käyttäjien oikeuksien hallintaan on käytössä monitasoiset työkalut aivan yksittäisestä käyttäjästä, käyttäjäryhmien kautta organisaatiotasolle.

Thingworxin osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukoon 6.

Taulukko 6. Kerätyt vertailutiedot: PTC Thingworx

Da ta O E c o s y s t e m	Provider	PTC
	Distributors (fi)	Elisa
	Partners (fi)	Bilot
	Ecosystem size	Large
	Marketplace	Yes. Applications, extensions, HW
	References (fi)	Wärtsilä, Metso, Valmet
	Other notable references	Volvo, Caterpillar
	End-to-end solution available	Yes
Da ta O	HW available from distributor	Yes, CloudGate

	WAN options	Ethernet, Optional 3G, LTE and WLAN
	Fieldbus connections	None
	Other connections	Optional RS-232 and RS-485
	Other sensors	Optional GPS
	Device management	Yes, own cloud
	Remote connections	Yes, Port tunneling via WS and managed from cloud
	Look-and-feel	Semi-industrial
	DIN installation	Optional. With adapter
Data collection SW	SW available	SDK (multiple languages), binary microserver & source
	OPC supported	AD, UA coming
	Messaging protocols	Multiple (Websocket, MQTT, HTTP, TCP/UDP, CoAP, etc)
	Connections to external services	HTTP (REST APIs), JDBC database connectors
	Data storage location	Finland (Elisa), global cloud or local
	Data security	HTTPS and WSS by default. 128-256bit AES
Handling and analyzing data	Data modelling	Multilevel with inheriting thing templates and extending thing shapes
	Automatic data handling	Some coding with JS scripts in events
	Analytic tools	Yes. With partner extensions and SQUEL (user's search, query and analyze)
	Machine learning	Yes. With partner extensions (e.g. MS Azure) or APIs
	Utilizing analyzed data	Can be directed to streams and handled as any other data input
	Data migration	Ability to export data in different formats or using REST. Uses DataStax on Cassandra.
Visualization	Creating views/dashboards	HTML5 drag and drop. Configurable logic.
	Mobile devices	Mobile mashups (automatic forwarding) Native apps coming
	Static reports	No
Service creation speed and models	Installation and configuration	If local, environment needs some work. If cloud, easy to start
	Need of coding	Minimal to High depending the complexity
	Independent usage	Possible
	Data model management	Manual but can be automated
Data security and rights management	Multitenancy	Yes. Organization level
	User and rights management	Multilevel with organizations. Users can manage their own organization (custom UI/mashup/login can be created)
	Access over all customer data	Easy
Scalability	Handling extending device count	Manual but can be automated
	Cloud or local installation	Cloud or local
	General scalability	Medium

Flexibility	Multiple tasks with one system	Yes
	Implementation of current concepts	Possible
	Effort for making changes	Easy. Inheriting and linked templates
	General flexibility	High
Pricing	Pricing model	License + number of excess top-level thing models + excess users + <u>file</u> transfer + storage
	One time costs	
	Upkeep costs	
	Costs/instance	
	Costs/customer	
	Costs/system	
	Costs/user	

5.4.2 Cumulocity

Cumulocity on vuonna 2010 Nokia Siemens Networkista lähtöisin oleva, saksalainen IoT-alusta ja alustakehittäjä. Alun perin operaattorikäyttöön suunniteltua IoT-alustaa on laajennettu yleisempään käyttöön soveltuvaksi ja siksi tällä hetkellä Cumulocityllä näytetään olevan erityisen selkeää pääsegmenttiä.

Cumulocity on enimmäkseen keskittynyt yksittäisten datapisteiden, kuten sensoreiden tuottaman tiedon keräämiseen ja seuraamiseen. Cumulocityllä onkin vahva ekoympäristö tiedon keräyslaitteistoja valmistavien yritysten kanssa, joissa on valmius tiedon lähettämiseen Cumulocityn alustaan. Tarjolla on Java-pohjainen SDK mutta pääasiallinen tiedonsiirtoprotokolla on http-viestit. Laittehallinta ja uusien laitteiden lisäys eli provisiointi, on tehokasta mutta rajoittuu vain yksinkertaisien tiedonkeruulaitteiden hallintaan.

Tieto on tallennettuna Cumulocityn sisäiseen tietokantaan ja linkattu laiteobjektiin alustan käyttöliittymissä. Laiterakenteiden mallintamiseen on käytössä alilaitte-käsite, jolla pystyy yksittäisiä laitteita ryhmittelemään toisten laitteiden alle. Laitteisiin liitetyn tiedon oletetaan olevan numeraalista tietoa ja tiedon käsittelyyn tarkoitettujen työkalujen ovatkin yksinkertaisia, valmiiksi konfiguroituja sääntöjä esimerkiksi raja-arvojen seurantaan, joiden perusteella voidaan tehdä erinäisiä toimintoja, kuten lähettää sähköpostia.

Cumulocityssä on mahdollisuus luoda näkymiä tietoon valmiista näyttökomponenteista hyvin nopeasti ja helposti. Luotuja sovelluksia voi jakaa eri käyttäjien kesken, mutta esimerkiksi loppuasiakasnäkymien luontiin toiminnallisuus ei sovellu, sillä käyttäjähallinnan ollessa hyvin yksinkertainen, ei järjestelmässä ole mahdollisuutta eristää asiakkaita

toisistaan ilman kokonaan erillisen instanssin luomista. Asiakkaiden käyttöön tarkoitettut tai monimutkaisemmat käyttöliittymät pitäisi siis luoda järjestelmän ulkopuolella.

Cumulocityn osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukoon 7.

Taulukko 7. Kerätyt vertailutiedot: Cumulocity

Ecosystem	Provider	Cumulocity
	Distributors (fi)	TeliaSonera
	Partners (fi)	Symbio, Tieto, Futurice, Avanade, Nortal, Fin- iconic, Microsoft, IBM, Haltian, Elektrobit
	Ecosystem size	Medium
	Marketplace	No
	References (fi)	None
	Other notable references	Deutsche Telekom
	End-to-end solution available	Yes
Data collection HW	HW available from distributor	Yes, M2M-in-a-box
	WAN options	Mobile
	Fieldbus connections	None
	Other connections	None
	Other sensors	GPS, Temperature, humidity, Motion/IR
	Device management	Cloud management
	Remote connections	No
	Look-and-feel	Consumer
	DIN installation	No
Data collection SW	SW available	Java SDK.
	OPC supported	None
	Messaging protocols	HTTP
	Connections to external services	HTTP (REST APIs)
	Data storage location	Finland (TeliaSonera) or global cloud (Az- ure/Softlayer)
	Data security	HTTPS by default.
Handling and analyzing data	Data modelling	Device level with subdevices
	Automatic data handling	Simple rules engine
	Analytic tools	None
	Machine learning	No
	Utilizing analyzed data	Only for display
	Data migration	Can be exported using REST
Visualiza- tion	Creating views/dashboards	HTML5, Angular JS, Cumulocity frame- work/plugin system
	Mobile devices	Responsive: Phone and tablets
	Static reports	No
Service creation speed	Installation and configuration	Easy to start
	Need of coding	Minimal to medium (end application needs to be coded with framework)

	Independent usage	Possible
	Data model management	Automatic
Data security and rights man-	Multitenancy	Upcoming feature. Currently needs separate instances
	User and rights management	Device level rights (owner)
	Access over all customer data	Hard
Scalability	Handling extending device count	Automatic
	Cloud or local installation	Cloud
	General scalability	Medium
Flexibility	Multiple tasks with one system	No
	Implementation of current concepts	Partly possible
	Effort for making changes	Medium. Limited rules and user interfaces needs coding.
	General flexibility	Medium
Pricing	Pricing model	Number of devices + storage + data transfer
	One time costs	
	Upkeep costs	
	Costs/instance	
	Costs/customer	
	Costs/system	
	Costs/user	

5.4.3 Wapice IoT-Ticket

IoT-Ticket on suomalaisen Wapice-yrityksen IoT-alusta, mikä on julkaistu vuonna 2015. IoT-Ticket on rakennettu yrityksen toisen tuotteen, tiedonkeräyslaitteiston tueksi mutta sitä on myös mahdollista käyttää myös ilman keräyslaitetta. IoT-Ticketin pääsegmentti on energiateollisuudessa sekä liikkuvissa koneissa, mihin keräyslaitteistokin on optimoitu. Alustan ollessa hyvin uusi, on sen kehitystahti yksi vertailun nopeimmista, mutta se tarkoittaa myös, että vertailun aikana moni ominaisuus oli vielä kehityksen alla.

IoT-Ticket keskittyy tietovirtojen hallintaan sekä niiden analysointiin. Lähtökohtaisesti tiedon oletetaan tulevan Wapicen omalta tiedonkeräyslaitteistolta, jonka hallintaan löytyy myös ominaisuuksia alustasta. Tiedonkeruulaitteistoa voidaan integroida projektina asiakkaan haluamiin kohteisiin ja tietolähteisiin. Tietoa on myös mahdollista siirtää alustaan käyttäen HTTP-protokollaa tai OPC:n avulla. Tiedon kuitenkin oletetaan olevan numeerista tai binääristä, kuten yksinkertaisista sensoreista yleensä tulee. Myös yksinkertainen ohjaus alustasta laitteeseen päin on mahdollista, jos käytetyssä laitteistossa on siihen tarvittavat ominaisuudet.

Tieto tallennetaan IoT-Ticketin tietokantaan mutta sen käsittelyyn ei tarjota kovin kattavia työkaluja. Sisään tulevaan tietoon voidaan liittää kevyttä raja-arvo tai poikkeavuus-analyysiä ja luoda hälytyksiä, mutta tiedon jalostamiseen ja yhdistelyyn ei ole mahdollisuutta. IoT-Ticketissä on graafinen konfigurointikäyttöliittymä, dataflow editor, tiedon kevyeen tarkasteluun, kuten minimi- tai maksimiarvojen etsintään, vertailuun ja tehtävien suorittamiseen, kuten ilmoitusten lähettämiseen. Alustassa on myös tehokkaammat analysointityökalut, joissa voidaan käyttää esimerkiksi R-kieltä analyysien ja graafien tekemiseen, mutta analysoinnin jälkeen tuloksia ei pysty tallentamaan takaisin järjestelmään. Tästä syystä analyysityökalut sopivat vain käsin käytettäviksi datan tutkiskelua varten. Laitteiden mallintamiseen tai hierarkkiseen rakenteeseen ei ole mahdollisuutta, vaan kaikki laitteet ovat samalla tasolla jokaisessa instanssissa.

Tiedon esittämiseen IoT-Ticketissä on mahdollisuus luoda dashboard-näkymiä tietovirtoihin. Tyylikkäitä ja moderneja dashboard-näkymiä on helppo luoda graafisella konfigurointityökalulla, jolla on mahdollista luoda esimerkiksi graafeja, numeronäyttöjä ja mittareita. Myös grafiikan, esimerkiksi prosessikaavion luonti on mahdollista. Ominaisuudet kuitenkin painottuvat valvomonäkymien tekemiseen eivätkä niin toimi käyttöliittymänä monimutkaisempien loppukäyttäjäsovelluksien tekemiseen. Näkymiin on rakennettu kevyttä oikeushallintaa, mutta koska jokaiselle asiakkaalle pitäisi kopioida oma, itsenäinen näkymänsä, on skaalautuvuus ja sen hallinta hankalaa.

IoT-Ticketin osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukoon 8.

Taulukko 8. Kerätyt vertailutiedot: Wapice IoT-Ticket

Ecosystem	Provider	Wapice
	Distributors (fi)	Wapice
	Partners (fi)	Analyticscloud
	Ecosystem size	None
	Marketplace	None
	References (fi)	TKL. Fortum.
	Other notable references	-
	End-to-end solution available	Yes, customized product
Data collection HW	HW available from distributor	Yes, WRM
	WAN options	3G, WLAN, Ethernet
	Fieldbus connections	CANopen, Modbus, Modbus TCP
	Other connections	1-wire protocol, Analog and digital inputs and outputs, ISO 11783, IEC61850, USB, OBD
	Other sensors	Accelerometer, GPS, Temperature
	Device management	Yes, own cloud
	Remote connections	Yes. Managed from Cloud
	Look-and-feel	Semi-industrial
	DIN installation	Optional. With DIN-backplate

Data collection SW	SW available	Java SDK
	OPC supported	AD, UA
	Messaging protocols	Websockets
	Connections to external services	HTTP (REST APIs)
	Data storage location	Finland cloud
	Data security	Transferred via VPN from WRM
Handling and analyzing data	Data modelling	Device level with templates
	Automatic data handling	Configured logic with Node-RED
	Analytic tools	R-statistical user interface (user's search, query and analyze)
	Machine learning	Not yet. Partnership with AnalyticsCloud
	Utilizing analyzed data	Only for display
	Data migration	Can be exported by provider
Visualization	Creating views/dashboards	HTML5 drag and drop. Configurable logic.
	Mobile devices	Responsive user interfaces. Mostly for tablets
	Static reports	Yes. With automatic sending (timed or triggered).
Service creation speed and models	Installation and configuration	Easy to start
	Need of coding	Minimal
	Independent usage	Possible
	Data model management	Automatic
Data security and rights management	Multitenancy	Yes. With own instances (handled automatically with scripts) or in one instance handled organization level.
	User and rights management	Multilevel with organizations. Users can manage their own organization (from admin view with restricted viewing rights)
	Access over all customer data	Hard to easy
Scalability	Handling extending device count	Automatic
	Cloud or local installation	Cloud or local
	General scalability	Medium
Flexibility	Multiple tasks with one system	Yes
	Implementation of current concepts	Mostly possible
	Effort for making changes	Hard. Manual work one by one
	General flexibility	Medium
Pricing	Pricing model	Number of devices + users + data transfer
	One time costs	
	Upkeep costs	
	Costs/instance	
	Costs/customer	

	Costs/system	
	Costs/user	

5.4.4 Microsoft Azure

Azure ei itsessään ole IoT-alusta, vaan sen sijaan monipuolinen yhdysvaltalaisen Microsoftin pilvipalvelualusta ja se on julkaistu vuonna 2010. Azure mahdollistaa erilaisten tietoteknisten ratkaisuiden toteuttamisen, ylläpidon ja julkaisun, oli kyseessä sitten virtuaalipalvelimet, pilvitalennustila tai koneoppiminen. Palveluvalikoima on todella laaja ja se kattaakin monta palvelutasoa kuten infrastruktuurin palveluna (infrastructure as a service, IaaS), alustan palveluna (platform as a service, PaaS) tai sovelluksen palveluna (software as a service, SaaS). Käytännössä kaikki IoT-teknologiapinon toiminnallisuudet löytyvät Azuresta, mutta ne eivät ole saumattomasti tai valmiiksi integroitu toisiinsa.

Azuren IoT-ratkaisu koostuu useasta eri Azuren palvelusta, joita sisäisiä rajapintoja yhdistämällä saadaan kasattua haluttu kokonaisuus. Tiedon keruuta varten Azuren palveluvalikoimasta löytyy Sensor Hub -toiminnallisuus, jolla tietoa voidaan lähettää Azureen Event Hub -palvelun käsiteltäväksi. Sensor Hub tukee todella montaa protokollaa ja myös valmiit SDK:t ovat saataville yleisimmille ohjelmointikielille.

Event Hub pystyy ohjaamaan tiedon jatkokäsiteltäväksi muihin Azuren palveluihin kuten Stream Analytics -palvelulle tarkastelua ja hälytyksiä varten tai erilliseen tietokantapalveluun tallennettavaksi, mutta koska ratkaisu koostetaan saatavilla olevista palveluista, voi logiikka ja arkkitehtuuri olla mitä tahansa. Azuressa on kattava valikoima erilaisia analytiikkapalveluita, kuten koneoppimista tai Python- sekä R-kielen työkaluja.

Valmiita käyttöliittymiä varten Azuresta löytyy Datazen-palvelu, minkä avulla tietoa voidaan esittää graafisesti konfiguroimalla. Kuitenkin, myös Datazenin tapauksessa, pääasiainen käyttöliittymien kehitys on tarkoitus tapahtua sisäisiä rajapintoja käyttämällä, joten käyttöliittymien tekeminen ei ole integroitu tietoon millään tavalla. Myös oikeushallintaa ei ole integroitu kokonaisuuteen, vaan eri loppukäyttäjää varten pitäisi pystyttää omia instansseja toiminnoista, joiden käyttöoikeuksia voidaan hallita vain ylätasolla. Toimintoja voidaan kuitenkin automatisoida erillisillä skripteillä, joiden hallintaan ei ole tarjolla mitään käyttöliittymää.

Azuren osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukkoon 9.

Taulukko 9. Kerätyt vertailutiedot: Microsoft Azure

Ecosystem	Provider	Microsoft
	Distributors (fi)	MS Finland
	Partners (fi)	Symbio, Cybercom and others
	Ecosystem size	Large
	Marketplace	None

	References (fi)	Valmet
	Other notable references	-
	End-to-end solution available	Not yet. Windows 10 IoT solution coming.
Data collection HW	HW available from distributor	No HW. Windows 10 IoT-core for e.g. Raspberry Pi 2
	WAN options	-
	Fieldbus connections	-
	Other connections	-
	Other sensors	-
	Device management	Yes, from Azure with Win10-IoT
	Remote connections	-
	Look-and-feel	-
	DIN installation	-
	Data collection SW	SW available
OPC supported		None
Messaging protocols		HTTP, AMQP
Connections to external services		HTTP (REST APIs)
Data storage location		Global cloud but can be locked to one location
Data security		VPN as option
Handling and analyzing data	Data modelling	Device level
	Automatic data handling	Yes. With Stream analytics.
	Analytic tools	Multiple, e.g. R-statistical language.
	Machine learning	Yes. Built-in
	Utilizing analyzed data	Can be directed to event hub and handled as any other data input
	Data migration	Application can be written to export the data
Visualization	Creating views/dashboards	Customized web pages or with DataZen drag and drop in desktop designer.
	Mobile devices	DataZen dashboards in native apps or embedded in other apps.
	Static reports	No
Service creation speed and models	Installation and configuration	Complex. Features need to be connected manually.
	Need of coding	Managing everything (customers, settings, properties) needs PowerShell scripting
	Independent usage	Hard
	Data model management	Manual
Data security and rights management	Multitenancy	No. Need separate instances of all components (storage, visualization)
	User and rights management	Multilevel with organizations. Users can manage their own organization (with custom built web service)
	Access over all customer data	Hard
Scalability	Handling extending device count	Manual but can be automated
	Cloud or local installation	Cloud

	General scalability	High
Flexibility	Multiple tasks with one system	Yes
	Implementation of current concepts	Possible but hard.
	Effort for making changes	Hard. Managing multiple instances needs PowerShell scripting.
	General flexibility	High
Pricing	Pricing model	Number of customers/ sets of components needed + number of messages + storage. Lot of consulting needed.
	One time costs	
	Upkeep costs	
	Costs/instance	
	Costs/customer	
	Costs/system	
	Costs/user	

5.4.5 IBM Bluemix

Kuten Microsoftin Azuren tapauksessa Bluemix ei ole tarkalleen ottaen IoT-alusta, mutta sisältää kaikki teknologiapinon toiminnallisuudet, joten IoT-ratkaisun toteuttaminen Bluemixillä on mahdollista. Bluemix on yhdysvaltalaisen IBM:n vuonna 2014 julkaisema pilviympäristö, joka on keskittynyt enemmän alustan toimittamiseen palveluna (PaaS). Bluemix sisältääkin enemmän toiminnallisuuksia sovelluksien kehittämiseen infrastruktuurista välittämättä ja erillisten sovellusten ajatusmalli vaatii, että ratkaisussa on enemmän ohjelmointia tarjottujen palveluiden integroimiseksi. Bluemixillä ei ole tarkkaa pääsegmenttiä, mutta palveluissa on nähtävissä hyvin laaja kirjo potentiaalisia käyttäjiä aivan kuluttajista, joita houkutellessa ilmaisilla palveluilla tiettyyn rajaan asti, kuin teollisuusyrityksiä valmiiksi saatavalla ennakoivan kunnossapidon modulilla.

Tiedon siirtämiseen löytyy oma IoT-toiminnallisuus, jolla tietoa voidaan lukea erikseen integroitavaan tietokantaan pääasiallisesti käyttäen MQTT-protokollaa. Tuetuille laitteille on olemassa laitehallinta, mistä kytkettyjä laitteita voidaan hallita. Tällä hetkellä valmis tuki löytyy enimmäkseen sulautetun elektroniikan kehitysalustoille, ja tarkoituksena onkin, että Bluemixin kumppanit toteuttavat tiedonkeruuratkaisut asiakkaalle tarpeen mukaan, painopisteen ollessa sulautetussa, yksinkertaisessa sensoritiedossa. Tiedon käsittelyyn on mahdollista ohjelmoimalla tehdä mitä tahansa ja käyttää valmiita, alustassa olevia analytiikkakirjastoja, mutta helppoa ja yksinkertaista, konfigurointitoiminnallisuutta ei valmiina Bluemixistä löydy. IBM:n Watson tekoäly- ja koneoppimispalvelu on mahdollista kohtuullisen helposti integroida ohjelmoimalla ratkaisuun.

Käyttöliittymien tai näkymien tekemiseen Bluemixistä ei löydy valmista toiminnallisuutta, vaan sen sijasta käyttöliittymät pitäisi kehittää itse Bluemixin sovelluskehitysoimainaisuuksilla tai ulkoisten rajapintojen kautta. Käyttäjähallinta pohjautuu myös sovelluksiin, joihin käyttäjille voidaan antaa oikeuksia tai sovelluksissa itsessään on erikseen toteutettu käyttäjien- ja oikeuksienhallinta.

Bluemixin osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukkoon 10.

Taulukko 10. Kerätyt vertailutiedot: IBM Bluemix

Ecosystem	Provider	IBM
	Distributors (fi)	Not needed
	Partners (fi)	Espotel and others
	Ecosystem size	Medium
	Marketplace	No
	References (fi)	None
	Other notable references	-
	End-to-end solution available	Yes, customized embedded device
Data collection HW	HW available from distributor	No. Partners have some sensor demos
	WAN options	-
	Fieldbus connections	-
	Other connections	-
	Other sensors	-
	Device management	-
	Remote connections	-
	Look-and-feel	-
	DIN installation	-
Data collection SW	SW available	Readymade boilerplate programs available
	OPC supported	None
	Messaging protocols	MQTT
	Connections to external services	HTTP (REST APIs)
	Data storage location	Global cloud
	Data security	Non-crypted
Handling and analyzing data	Data modelling	Device level
	Automatic data handling	Yes. Can be coded with multiple languages.
	Analytic tools	Multiple
	Machine learning	Yes. Built-in (Watson)
	Utilizing analyzed data	Can be sent to other functionalities and used as any other data input
	Data migration	Uses Hadoop
Visualization	Creating views/dashboards	No
	Mobile devices	-
	Static reports	-

Service creation speed and models	Installation and configuration	Complex. Features need to be connected manually.
	Need of coding	Medium
	Independent usage	Hard
	Data model management	Manual
Data security and rights management	Multitenancy	No. Need separate instances of all components (storage, visualization)
	User and rights management	Singlelevel Bluemix users or custom user management
	Access over all customer data	Hard
Scalability	Handling extending device count	Manual
	Cloud or local installation	Cloud
	General scalability	High
Flexibility	Multiple tasks with one system	Yes
	Implementation of current concepts	Possible but hard.
	Effort for making changes	Hard.
	General flexibility	High
Pricing	Pricing model	Number of instances + users + used memory. Lot of consulting needed.
	One time costs	
	Upkeep costs	
	Costs/instance	
	Costs/customer	
	Costs/system	
	Costs/user	

5.4.6 Amazon AWS

Yhdysvaltaisen Amazonin AWS (Amazon Web Services) on vuonna 2006 julkaistu pilvipalvelualusta, mikä vastaa täysin kovinta kilpailijaansa, Microsoftin Azurea, eikä siten ole itsessään IoT-alusta. AWS on kuitenkin keskittynyt vielä enemmän laskentakapasiteetin ja infrastruktuurin tarjontaan palveluna (IaaS) kuin IoT-ratkaisuihin.

Alustasta kuitenkin löytyy toiminnallisuudet tiedon siirtämiseen alustan eri palveluihin käyttämällä Amazon Simple Queue Serviceä (SQS) HTTP-protokollalla, jonka hyödyntämiseen on tarjolla SDK-kirjastot usealle ohjelmointikielelle. AWS:stä löytyy palvelut tietokantoja varten, mutta edes yksinkertaiseen tiedon käsittelyyn tai esittämiseen ei löydy valmista toiminnallisuutta.

AWS:n osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukkoon 11.

Taulukko 11. Kerätyt vertailutiedot: Amazon AWS

Ecosystem	Provider	Amazon
	Distributors (fi)	Not needed
	Partners (fi)	Cybercom and others
	Ecosystem size	Large
	Marketplace	No
	References (fi)	None
	Other notable references	-
	End-to-end solution available	No HW
Data collection HW	HW available from distributor	No
	WAN options	-
	Fieldbus connections	-
	Other connections	-
	Other sensors	-
	Device management	-
	Remote connections	-
	Look-and-feel	-
	DIN installation	-
Data collection SW	SW available	No
	OPC supported	None
	Messaging protocols	MQTT
	Connections to external services	HTTP (REST APIs)
	Data storage location	Global cloud but can be locked to one location
	Data security	Non-crypted
Handling and analyzing data	Data modelling	None, only data
	Automatic data handling	Yes. Coded with AWS Kinesis & Lambda.
	Analytic tools	Realtime statistics
	Machine learning	Yes
	Utilizing analyzed data	Can be sent to other functionalities and used as any other data input
	Data migration	Application can be written to export the data
Visualization	Creating views/dashboards	No
	Mobile devices	-
	Static reports	-
Service creation speed and models	Installation and configuration	Complex. Features need to be connected manually.
	Need of coding	High
	Independent usage	Hard
	Data model management	-
Data security and rights management	Multitenancy	No. Need separate instances of all components (storage, visualization)
	User and rights management	AWS users or custom user management
	Access over all customer data	Hard
Scalability	Handling extending device count	Manual but can be automated

	Cloud or local installation	Cloud
	General scalability	High
Flexibility	Multiple tasks with one system	Yes
	Implementation of current concepts	Possible but hard.
	Effort for making changes	Hard.
	General flexibility	High
Pricing	Pricing model	Number of customers/ sets of components needed + number of messages + storage. Lot of consulting needed.
	One time costs	
	Upkeep costs	
	Costs/instance	
	Costs/customer	
	Costs/system	
	Costs/user	

5.4.7 2lemetry

2lemetry on yhdysvaltalainen IoT-alusta, mikä on perustettu vuonna 2011 alun perin startup-yrityksenä. Hankintaprojektin aikana Amazon osti 2lemetryn aikeinaan integroida 2lemetryn toiminnallisuudet omaan AWS-palveluunsa. 2lemetry sisältää kaikki IoT-alustalle tarvittavat ominaisuudet, mutta tuntuu olevan tarkoitettu kevyempään käyttöön maksimiviestimäärien ja säilytysaikojen ollessa joihinkin käyttötarkoituksiin alimitoitettu. Alusta soveltuu myös kuluttajakäyttöön ilmaisien käyttötasojen puolesta, mutta myös enterprise-taso on olemassa. Alustalla ei näytä olevan erityistä pääsegmenttiä.

Tieto siirretään järjestelmään MQTT-protokollalla ja tallennetaan alustan sisäiseen tietokantaan. 2lemetry sisältää myös toiminnallisuudet tietoa lähettävien laitteiden ja niiden lähettämien tietojen hallintaan. Viestien lukumäärä on rajoitettu ajan suhteen, jopa ylimällä käyttötasolla ja tallennettua tietoa säilötään alustassa vain rajatun ajan verran, kuukaudesta vuoteen riippuen käyttötasosta. Tietoa voi analysoida jo sen saapuessa alustaan ja reagoida tarvittaessa esimerkiksi hälytyksien muodossa sekä siihen voi luoda erilaisia sääntöjä.

Näkymien tai käyttöliittymien tekemiseen 2lemetry ei suoraan anna mahdollisuutta, mutta se tarjoaa valmiita kirjastoja käyttöliittymien ja muiden integraatioiden tekemiseen alustaan. Tarjolla on muun muassa sovelluskirjasto reaaliaikaisen tiedonsiirron integrointiin omiin käyttöliittymiin. Alusta mainostaa itseänsä multitenanttiseksi eli järjestelmäksi,

jossa useampi asiakas voi sijaita yhtä aikaa, mutta loppukäyttäjien käytössä jokainen asiakas tarvitsisi oman asiakkuuden ja tällöin näkymä yli kaikkien asiakkaiden ei ole mahdollista.

2lemetryn osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukkoon 9.

Taulukko 12. Kerätyt vertailutiedot: 2lemetry

Ecosystem	Provider	Amazon
	Distributors (fi)	None
	Partners (fi)	Intel, Amazon, Heroku, Datastax, Sprint, Salesforce
	Ecosystem size	Small
	Marketplace	Yes. HW and clothes.
	References (fi)	None
	Other notable references	Honeywell
	End-to-end solution available	No. HW and visualization missing.
Data collection HW	HW available from distributor	No
	WAN options	-
	Fieldbus connections	-
	Other connections	-
	Other sensors	-
	Device management	-
	Remote connections	-
	Look-and-feel	-
Data collection SW	DIN installation	-
	SW available	No
	OPC supported	None
	Messaging protocols	MQTT, CoAP, STOMP
	Connections to external services	HTTP (REST APIs)
	Data storage location	US(?) cloud
Handling and analyzing data	Data security	Non-crypted
	Data modelling	Device level
	Automatic data handling	Simple rules engine
	Analytic tools	Real time analytics and statistical tools.
	Machine learning	With AWS Lambda
	Utilizing analyzed data	Can't be imported back but can be used elsewhere
Visualization	Data migration	Can be exported using REST
	Creating views/dashboards	No
	Mobile devices	-
Service creation	Static reports	-
	Installation and configuration	Easy to start
	Need of coding	None

	Independent usage	Possible
	Data model management	Manual
Data security and	Multitenancy	No. All data in one instance
	User and rights management	Singlelevel only
	Access over all customer data	Impossible
Scalability	Handling extending device count	Automatic
	Cloud or local installation	Cloud
	General scalability	High
Flexibility	Multiple tasks with one system	No
	Implementation of current concepts	No
	Effort for making changes	Easy.
	General flexibility	Medium
Pricing	Pricing model	Montly fee + number of devices. Multiple package sizes available (max. Devices, price/device, data rentention time, real time analytics, skinning, support)
	One time costs	
	Upkeep costs	
	Costs/instance	
	Costs/customer	
	Costs/system	
	Costs/user	

5.4.8 ELK Stack

ELK Stack on avoimen lähdekoodin ratkaisu tiedon jalostamiseen ja esittämiseen, mutta ei itsessään ole IoT-alusta. ELK Stack on aluperin julkaistu vuonna 2010. Ratkaisussa on integroitu useampi avoimen lähdekoodin sovellus toimimaan saumattomasti yhdessä, mutta ELK Stackin alkuperäinen käyttötarkoitus, lokiviestien analysointi on hyvin vahvasti esillä.

Tietoa voidaan siirtää, esiprosessoida ja tallentaa ELK Stackissa olevalla Logstash-sovelluksella. Sisään siirretty tieto tallennetaan ElasticSearch-sovelluksen tietokantaan, jossa tietoa analysoidaan ja indeksoidaan automaattisesti. Käsiteltyä tietoa voidaan esittää ratkaisuun integroidulla Kibana-sovelluksella, minkä avulla voidaan tehdä erilaisia näkymiä Elasticsearchin jalostamaan informaatioon ja löydöksiin sekä raakatietoon. Kibana-sovelluksella voidaan myös konfiguroida ja ohjata ELK Stackkiin integroituja sovelluksia.

ELK Stackin osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukkoon 13.

Taulukko 13. Kerätyt vertailutiedot: ELK Stack

Ecosystem	Provider	Elastic
	Distributors (fi)	Cybercom
	Partners (fi)	None
	Ecosystem size	None
	Marketplace	No
	References (fi)	-
	Other notable references	None
	End-to-end solution available	No, only data collection, storage and visualization
Data collection HW	HW available from distributor	No
	WAN options	-
	Fieldbus connections	-
	Other connections	-
	Other sensors	-
	Device management	-
	Remote connections	-
	Look-and-feel	-
	DIN installation	-
Data collection SW	SW available	Yes. Logstac
	OPC supported	None
	Messaging protocols	HTTP
	Connections to external services	None
	Data storage location	Finland cloud or local
	Data security	VPN by default
Handling and analyzing data	Data modelling	None, only data
	Automatic data handling	Need to be configured/coded by partner
	Analytic tools	No
	Machine learning	With tailored solution
	Utilizing analyzed data	Only for display
	Data migration	Uses Hadoop
Visualization	Creating views/dashboards	HTML5 widget selection
	Mobile devices	No
	Static reports	No
Service creation speed and models	Installation and configuration	Complex. Features are different programs.
	Need of coding	High
	Independent usage	Hard
	Data model management	-
Data security and rights management	Multitenancy	No. Every client needs own instance with 3-5 virtual servers.
	User and rights management	Multilevel
	Access over all customer data	Hard
Scalability	Handling extending device count	Manual
	Cloud or local installation	Private cloud or local

	General scalability	Low
Flexibility	Multiple tasks with one system	No
	Implementation of current concepts	No
	Effort for making changes	Hard. Manual work one by one
	General flexibility	Low
Pricing	Pricing model	Consulting + subscription + setup costs + virtual server upkeeping.
	One time costs	
	Upkeep costs	
	Costs/instance	
	Costs/customer	
	Costs/system	
	Costs/user	

5.4.9 Ixonos Industrial Internet Suite

Industrial Internet Suite on suomalaisen Ixonos-yrityksen IoT-alusta. Alusta on erikoistunut teollisuuden PLC-sovellutuksiin ja etenkin prosessiteollisuuteen. Industrial Internet Suite on pohjimmiltaan tiedon keräykseen ja esittämiseen keskittynyt ratkaisu, sillä yrityksen tarjonta keskittyy paljolti sulautettujen sensoriratkaisuiden kehittämiseen sekä käyttöliittymä ja -kokemuksen suunnitteluun. Alusta on myös vahvasti kaksisuuntainen tarjoten teollisten laitteiden ohjauksen sekä arvojen säätämisen käyttöliittymistä käsin. Yritys tarjoaa myös valmista, sulautetulla elektroniikalla toteutettua käyttöpaneeliratkaisua asiakkaan tarpeisiin muokattuna.

Industrial Internet Suiten osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukkoon 9.

Taulukko 14. Kerätyt vertailutiedot: Ixonos Industrial Internet Suite

Ecosystem	Provider	Ixonos
	Distributors (fi)	Ixonos
	Partners (fi)	TI, Intel and Freescale
	Ecosystem size	None
	Marketplace	No
	References (fi)	Abloy, Cargotec, Grundfos, Kone, Metso, NSN, Outotec, Viconsys
	Other notable references	-
	End-to-end solution available	Yes
Data col- lec-	HW available from distributor	Yes, Ixonos Sensor Hub
	WAN options	Ethernet, WLAN

	Fieldbus connections	Multiple
	Other connections	Bluetooth LE, USB, Wi-Fi
	Other sensors	Via partner customization
	Device management	No.
	Remote connections	No
	Look-and-feel	Semi-industrial
	DIN installation	Possible
Data collection SW	SW available	Custom SW
	OPC supported	Yes
	Messaging protocols	HTTP
	Connections to external services	Custom work
	Data storage location	Finland cloud or local
	Data security	Custom solutions
Handling and analyzing data	Data modelling	Need coding
	Automatic data handling	Need coding
	Analytic tools	Possible to implement
	Machine learning	Possible to implement
	Utilizing analyzed data	Possible to implement
	Data migration	Application can be written to export the data
Visualization	Creating views/dashboards	With customized solution
	Mobile devices	Possible
	Static reports	Possible
Service creation speed and models	Installation and configuration	Needs development work
	Need of coding	High
	Independent usage	Hard to impossible
	Data model management	-
Data security and	Multitenancy	No
	User and rights management	Custom
	Access over all customer data	Hard
Scalability	Handling extending device count	Manual but can be automated
	Cloud or local installation	Private cloud or local
	General scalability	Medium
Flexibility	Multiple tasks with one system	Yes
	Implementation of current concepts	No
	Effort for making changes	Hard
	General flexibility	Medium
Pricing	Pricing model	Number of devices + virtual server upkeep + consulting
	One time costs	
	Upkeep costs	

	Costs/instance	
	Costs/customer	
	Costs/system	
	Costs/user	

5.4.10 Tieto I&I Connect

I&I Connect on IoT-alusta, jota tarjoaa suomalainen Tieto-yritys. Ratkaisu itsessään koostuu avoimen lähdekoodin ratkaisusta, joita Tieto yhdistelee ja integroi asiakkaan tarpeiden mukaan. Alustaan on mahdollista liittää erilaisia tietolähteitä ja -virtoja, mutta sulautetun kehitysvalmiuden lisäksi valmiita laiteratkaisuita ei ole tarjolla. Tiedon käsittely ja analysointi tehdään projektina asiakkaan kanssa ja tarjolla onkin toiminallisuutta koneoppimisesta tuotantoanalyysiin. I&I Connectilla ei ole erityistä pääsegmenttiä ja listaakin sovelluskohteissaan lähes kaikki teollisuuden alat.

I&I Connectin osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukoon 15.

Taulukko 15. Kerätyt vertailutiedot: Tieto I&I Connect

Ecosystem	Provider	Tieto
	Distributors (fi)	Tieto
	Partners (fi)	Solteq, Roima, Fouga IT, Trelab, Livion, Data Rangers, Oliotalo, Kaltio Technologies ja Quuppa
	Ecosystem size	None
	Marketplace	No
	References (fi)	Kempfi
	Other notable references	-
	End-to-end solution available	No HW
Data collection HW	HW available from distributor	No
	WAN options	-
	Fieldbus connections	-
	Other connections	-
	Other sensors	-
	Device management	-
	Remote connections	-
	Look-and-feel	-
Data collection SW	DIN installation	-
	SW available	Custom SW
	OPC supported	No
	Messaging protocols	HTTP
	Connections to external services	Custom work
	Data storage location	Finland cloud

	Data security	Custom solutions
Handling and analyzing data	Data modelling	Need coding
	Automatic data handling	Need coding
	Analytic tools	Possible with partner technologies
	Machine learning	No
	Utilizing analyzed data	Possible to implement
	Data migration	Application can be written to export the data
Visualization	Creating views/dashboards	With customized solution
	Mobile devices	Possible
	Static reports	Possible
Service creation speed and models	Installation and configuration	Needs development work
	Need of coding	High
	Independent usage	Hard to impossible
	Data model management	-
Data security and	Multitenancy	Possible
	User and rights management	Custom
	Access over all customer data	Hard
Scalability	Handling extending device count	Manual but can be automated
	Cloud or local installation	Private cloud or local
	General scalability	Medium
Flexibility	Multiple tasks with one system	Yes
	Implementation of current concepts	No
	Effort for making changes	Hard
	General flexibility	Medium
Pricing	Pricing model	Number of devices + virtual server upkeep + consulting + development work
	One time costs	
	Upkeep costs	
	Costs/instance	
	Costs/customer	
	Costs/system	
	Costs/user	

5.4.11 Keskustekniikka Cosmos

Cosmos on tiedonkeruu- ja analysointijärjestelmä suomalaiselta yritykseltä Keskustekniikalta. Keskustekniikka toimittaa yleisesti sähkö- ja automaattioratkaisuita mutta listaa Cosmos-järjestelmänsä erikoisalueeksi kone- ja laitetuottajat raskaissa teollisuudenaloissa kuten prosessi-, kaivos-, energia- ja meriteollisuudessa.

Cosmos keskittyy tiedon keräämiseen erilaisilla antureilla ja tarjoaakin valmiita sulautettua elektroniikan keräyslaitteistoja koneteollisuuden sekä logistiikka-alan tarpeisiin. Myös integroinnit teollisuuden ohjausjärjestelmiin ja ohjelmoitaviin logiikoihin on mahdollista valmistajien omilla rajapinnoilla tai OPC-protokollan avulla. Cosmos-järjestelmän sulautettujen keräyslaitteistoiden ja antureiden keräämää tietoa siirretään Cosmos Enabled -pilvipalveluun, mikä täyttää osan IoT-teknologiapinon toiminnallisuuksista. Cosmos Enabled sisältää sisäisen tietokannan mihin tieto tallennetaan. Tiedon analysointiin ja käsittelyyn ei ole erillisiä työkaluja, vaan tieto on tarkoitus siirtää eteenpäin alustasta jatkojalostusta varten. Alusta kuitenkin tarjoaa yksinkertaiset näkymät kerättyyn tietoon valvontatarkoituksiin.

Cosmoksen osalta kerätyt, vertailukriteereitä vastaavat tiedot ovat koostettuna taulukkoon 16.

Taulukko 16. Kerätyt vertailutiedot: Keskustekniikka Cosmos

Ecosystem	Provider	Keskustekniikka/CLS-Engineering
	Distributors (fi)	Keskustekniikka/CLS-Engineering
	Partners (fi)	None
	Ecosystem size	None
	Marketplace	No
	References (fi)	-
	Other notable references	-
	End-to-end solution available	Yes
Data collection HW	HW available from distributor	Yes, X10
	WAN options	Ethernet
	Fieldbus connections	Modbus/TCP, S7
	Other connections	1-wire protocol, digital inputs
	Other sensors	Temperature, vibration
	Device management	No
	Remote connections	No
	Look-and-feel	Industrial
	DIN installation	Yes
Data collection SW	SW available	No
	OPC supported	None
	Messaging protocols	SMTP (email) or HTTPS
	Connections to external services	Custom work
	Data storage location	Local storage or global cloud
	Data security	Possible to use HTTPS
Handling and analyzing data	Data modelling	Device level
	Automatic data handling	No
	Analytic tools	Trend views
	Machine learning	No
	Utilizing analyzed data	Only for display

	Data migration	MySql database
Visuali- zation	Creating views/dashboards	Yes. Simple data dashboards
	Mobile devices	Yes
	Static reports	Yes
Service cre- ation speed and models	Installation and configuration	Medium work needed to start
	Need of coding	Low
	Independet usage	Possible
	Data model management	Manual
Data se- curity and	Multitenancy	No. Own instance for every customer
	User and rights management	Singlelevel AD-authentication
	Access over all customer data	Hard
Scalability	Handling extending device count	Automatic
	Cloud or local installation	Cloud or local
	General scalability	Low
Flexibility	Multiple tasks with one system	No
	Implementation of current con- cepts	No
	Effort for making changes	Hard to impossible
	General flexibility	Low
Pricing	Pricing model	License + number of sensors and gateways
	One time costs	
	Upkeep costs	
	Costs/instance	
	Costs/customer	
	Costs/system	
	Costs/user	

6. VERTAILUN TULOKSET

6.1 Vertailutaulukko

Vertailutaulukko saatiin täytettyä täysin valmiiksi, eli jokaisesta yhdestätoista alustavaihtoehdosta löydettiin kaikkien vertailukriteerien vaatimat asiat. Vertailukriteereitä oli 53 kappaletta kymmenessä eri kategoriassa. Yhteensä vertailutaulukon soluja oli siis 583 kappaletta. Kuudessakymmenessäkuudessa solussa vertailukriteeri ei ollut merkitsevä johtuen aikaisemman kriteerin tuloksesta ja jäi siten määrittelemättömäksi. Nämä solut merkittiin täyttämällä soluihin tavuviiva. Esimerkiksi, jos alustavaihtoehdon toimittajalta ei ollut ratkaisua saatavilla, merkittiin kaikki ratkaisua koskevat vertailukriteerien solut tavuviivalla esittämään niiden määrittelemättömyyttä.

Vertailutaulukkoa oli helppo täyttää siitä huolimatta, että vertailukriteereitä sekä alustavaihtoehtoja oli paljon. Tietoa oli yksinkertaista täyttää yksi sarake tai vertailukriteeri kerrallaan, mikä poisti kompleksisuutta, kun tietoa oli etsittäväenä paljon useasta eri lähteestä. Haettu tieto on kirjattu kohtuullisen vapaamuotoisesti taulukkoon, mutta kuitenkin siten, että tiedosta on mahdollista muodostaa käsitys alustan kyvykkyyksistä sekä ominaisuuksista. Vertailukriteerien ryhmittely auttaa ymmärtämään toisiinsa kuuluvat vertailukriteerit sekä muodostamaan käsityksen jokaisesta osa-alueesta niin yhden kuin useanakin alustan välillä.

Täytetty vertailutaulukko on nähtävissä kuvassa 13. Vertailutaulukko on isompana nähtävissä työn liitteessä B. Vertailutaulukosta on poistettu yrityssalaisuuden alaisuuteen kuuluvat tarkemmat kustannuksiin liittyvät tiedot.

	ThingWorx	IoT-Ticket	Azure	Bluemix	2lemetry	AWS	ELK Stack	Industrial Internet Suite	I&I Connect	Cosmos	Cumulocity	
Ecosystem	Provider	PTC	Mapice	Microsoft	IBM	Amazon	Amazon	Elastic	Ionos	Tieto	Keskusmekanikka, S-Engineering, Keskusmekanikka, S-Engineering	Cumobility
	Distributors (I)	Elta	Mapice	MS Finland	Not needed	None	Not needed	Cybercom	Ionos	Tieto	TakaSonera	Symbio, Tieto, Finnicor, BMT, Halian, Elektrobit
	Partners (I)	Blitz	Analyticscloud	Symbio, Cybercom and others	Esopel and others	Intel, Amazon, Heroku, Datastax, Spinn, Salesforce	Cybercom and others	None	TI, Intel and Freescale	Sokeq, Roma, FougIT, Tealab, Livion, Data Rangers, Clotals, Kalko Technologies ja Quappa	None	None
	Ecosystem size	Large	None	Large	Medium	Small	Large	None	None	None	None	None
Data collection HW	Marketplace	Yes. Applications, extensions, Väikisa, Matso, Valmet	None	None	None	None	None	None	None	None	None	None
	References (I)	Volvo, Caterpillar	-	-	-	-	-	-	-	-	-	-
	Other notable references	Yes, customized product	-	-	-	-	-	-	-	-	-	-
	End-to-end solution available	Yes	Yes, customized product	No/Yes. Windows 10 IoT Core for e.g. Raspberry Pi 2	Yes, customized embedded device	None	No/Yes	No	Yes	No	Yes	Yes
Data collection SW	HW available from distributor	Yes, CloudGate	Yes, vFRM	No HW. Windows 10 IoT Core for e.g. Raspberry Pi 2	No. Partners have some server demos	No	No	No	Yes, Ionos Sensor Hub	No	Yes, XIO	Yes, M2M-in-a-box
	WAN options	Ethernet, Optional 3G, LTE and WLAN	3G, VLAN, Ethernet	-	-	-	-	-	Ethernet, VLAN	-	Ethernet	Mobile
	Fieldbus connections	None	CANopen, Modbus, TCP	-	-	-	-	-	Multiple	-	Modbus/TCP, S7	None
	Other connections	Optional RS-232 and RS-485	-	-	-	-	-	-	-	-	-	-
Handling and analyzing data	Other sensors	Optional GPS	Optional: USB, OSD Accelerometer, GPS, Temperature	-	-	-	-	-	Bluetooth LE, USB, Wi-Fi	-	-	-
	Device management	Yes, own cloud	Yes, own cloud	Yes, from Azure with Win10/IoT	-	-	-	-	No.	-	-	-
	Remote connections	Yes, Port tunneling via WS and managed from cloud	Yes, Managed from Cloud	-	-	-	-	-	No	-	-	-
	Look-and-feel	Optional, With adapter	Optional, With DIN-backplate	-	-	-	-	-	-	-	-	-
Visualization	DIN installation	Optional, With adapter	Optional, With DIN-backplate	-	-	-	-	-	-	-	-	-
	SW available	SDK (multiple languages), binary microservices to source	Java SDK	Libraries for multiple languages available	Ready-made boilerplate programs available	No	No	Yes, Logstash	Custom SW	Custom SW	No	Java SDK
	DPC supported	AD, UA coming	AD, UA	None	None	None	None	None	Yes	No	None	None
	Messaging protocols	Multiple (Websocket, MQTT, HTTP, TOPIC, CoAP, etc)	Websockets	HTTP, AMQP	MQTT, CoAP, STOMP	None	None	None	HTTP	HTTP	None	None
Services require period and models	Connections to external services	HTTP (REST APIs), JDBC database connectors	HTTP (REST APIs)	HTTP (REST APIs)	HTTP (REST APIs)	HTTP (REST APIs)	None	Custom work	Custom work	Custom work	Custom work	HTTP (REST APIs)
	Data storage location	Finland (EU), global cloud or local	Finland cloud	Global cloud but can be locked to one location	Global cloud	US/Ireland cloud	Global cloud but can be locked to one location	Finland cloud or local	Finland cloud or local	Finland cloud	Local storage or global cloud	Local storage or global cloud (Azure/Citrix/Net)
	Data security	HTTPS and WSS by default, 128-256bit AES	Transferred via VPN from vFRM	VPN as option	Non-encrypted	Non-encrypted	Non-encrypted	VPN by default	Custom solutions	Custom solutions	Possible to use HTTPS	HTTPS by default
	Data modelling	Multiple with inheriting thing templates and extending thing shapes	Device level with templates	Device level	Device level	Device level	None, only data	None, only data	Need coding	Need coding	Need coding	Device level with sub-devices
Data security and rights management	Automatic data handling	Some coding with JS scripts in events	Configured logic with Node-RED	Yes, With Stream analytics	Yes, Can be coded with multiple languages	Simple rules engine	Yes, Coded with AWS Kinesis & Lambda	Needs to be configured/coded by partner	Need coding	Need coding	No	Simple rules engine
	Analysis tools	Yes, With partner extensions and SQL (user's search, query and analyze)	Practical user interface (user's search, query and analyze)	Multiple, e.g. R, statistical language	Multiple	Realtime user interface and statistical tools	Realtime statistics	No	Possible to implement	Possible with partner technologies	Trend views	None
	Machine learning	Yes, With partner extensions (e.g. MS Azure) or APIs	Not used. Partnership with AnalyticsCloud	Yes, Built-in	Yes, Built-in (Watson)	With AWS Lambda	With AWS Lambda	With tailored solution	Possible to implement	No	No	No
	Utilizing analyzed data	Can be directed to streams and handled as any other data input	Only for display	Can be directed to event hub and handled as any other data input	Can be sent to other functionalies and used as any other data input	Can be imported back but can be used elsewhere	Can be sent to other functionalies and used as any other data input	Only for display	Possible to implement	Possible to implement	Only for display	Only for display
Scalability	Data migration	Ability to export data in different formats or using REST. Uses DataStax/Cassandra	Can be exported by provider	Application can be written to export the data	User Hadoop	Can be exported using REST	Application can be written to export the data	Uses Hadoop	Application can be written to export the data	Application can be written to export the data	MySQL database	Can be exported using REST
	Creating view/dashboards	HTML5 drag and drop. Configurable logic	HTML5 drag and drop. Configurable logic	Customized web pages or with Dashboard drag and drop in desktop designer	No	No	No	HTML5 widget selection	With customized solution	With customized solution	Yes, Simple data dashboards	HTML5, Angular JS, Cumulocity framework/widget system
	Mobile devices	Mobile markup (automatic forwarding) Native apps coming	Responsive user interfaces. Mostly for tablets	-	-	-	-	No	Possible	Possible	Yes	Responsive Phone and tablets
	Static reports	No	Yes, With automatic sending (time or triggered)	-	-	-	-	No	Possible	Possible	Yes	No
Flexibility	Installation and configuration	If local, environment needs some work. If cloud, easy to start	Easy to start	Complex. Features need to be connected manually	Complex. Features need to be connected manually	Easy to start	Complex. Features need to be connected manually	Complex. Features are different programs	Needs development work	Needs development work	Medium work needed to start	Easy to start
	Need of coding	Minimal to High depending the complexity	Minimal	Managing everything (customers, settings, properties) needs PowerShell scripting	Medium	None	High	High	High	High	Low	Minimal to medium (end application needs to be coded with framework)
	Independent usage	Possible	Possible	Hard	Hard	Possible	Hard	Hard	Hard to impossible	Hard to impossible	Possible	Possible
	Data model management	Manual but can be automated	Automatic	Manual	Manual	Manual	Manual	Manual	Manual	Manual	Manual	Manual
Pricing	Multitenancy	Yes, Organization level	Yes, With own instances (handled automatically with config) or on one instance handled organization level	No. Need separate instances of all components (storage, visualization)	No. Need separate instances of all components (storage, visualization)	No. All data in one instance	No. Need separate instances of all components (storage, visualization)	No. Every client needs own instance with 3-5 virtual servers.	No	Possible	No. Own instance for every customer	Upcoming feature. Currently needs separate coding
	User and rights management	Multiple with organizations. Users can manage their own organization (custom UIMashup/login can be created)	Multiple with organizations. Users can manage their own organization from admin view with restricted viewing rights)	Single level Bluemix users or custom user management	Single level only	Single level only	Multiple with organizations (with custom built web service)	Multiple	Custom	Custom	Single level AD-authentication	Device level rights (owner)
	Access over all customer	Easy	Hard to easy	Hard	Hard	Impossible	Hard	Hard	Hard	Hard	Hard	Hard
	Handling extending device count	Manual but can be automated	Automatic	Manual but can be automated	Manual	Automatic	Manual but can be automated	Manual	Manual but can be automated	Manual but can be automated	Manual but can be automated	Automatic
Costs	Cloud or local installation	Cloud or local	Cloud or local	Cloud	Cloud	Cloud	Private cloud or local	Private cloud or local	Private cloud or local	Private cloud or local	Cloud or local	Cloud
	General scalability	Medium	Medium	High	High	High	High	Medium	Medium	Medium	Low	Medium
	Multiple tasks with one implementation of customer	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	No
	Effort for making changes	Possible	Mostly possible	Hard	Hard	Hard	Easy	Hard	Hard	Hard	Hard	Hard/impossible
Other	General flexibility	High	Medium	High	High	Medium	High	Low	Medium	Medium	Low	Medium
	Pricing model	License + number of access points + number of sensors + access users + data transfer + storage	Number of devices + users + data transfer	Number of customer sets of components needed + number of messages + storage. Lot of consulting needed.	Number of instances + users + used memory + number of consulting needed.	Monthly fee + number of devices. Multiple package sizes available (max. devices, price/device, data retention time, real time analytics, skinning, support)	Number of customer sets of components needed + number of messages + storage. Lot of consulting	Consulting + subscription + setup costs + virtual server upkeep	Number of devices + virtual server upkeep + consulting + development work	Number of devices + virtual server upkeep + consulting + development work	License + number of sensors and gateways	Number of devices + storage + data transfer
	One time costs											
	Upkeep costs											
Instance costs												
Customer costs												
System costs												
User costs												

Kuva 13. Täytetty vertailutaulukko alustavaihtoehdoista

6.2 Pughin matriisi

Täytetyn vertailutaulukon pohjalta laadittiin kappaleessa 5.2 esitelty Pughin matriisi alus-
tojen keskinäistä vertailua varten. Pughin matriisin periaatteiden mukaisesti kaikkia ver-
tailuvaihtoehtoja vertaillaan yhteen vertailukohteeseen, mikä voi olla valittu satunnaisesti
tai se voi olla kehityshankkeissa nykyisin käytössä oleva ratkaisu. Tässä tapauksessa ver-
tailukohteeksi valittiin lupaavin alustavaihtoehto PTC Thingworx, jonka joustavuus sekä
ominaisuudet olivat vakuuttaneet jo käyttötestien aikana ja jota pidettiin hyvin vahvana
ehdokkaana.

Pughin matriisin täyttäminen oli hyvin helppoa, kun vertailutaulukkoa käytettiin Pughin
matriisin pohjana ja soluihin vapaasti kirjatut arvot korvattiin Pughin matriisin määrittä-
millä +, - ja S -merkinnöillä. Täytetty Pughin matriisi on nähtävissä kuvassa 14. Matriisi
löytyy myös isompana työn liitteestä C.

		ThingWorx	IoT-Ticket	Azure	Bluemix	2lemetry	AWS	ELK Stack	Industrial Internet Suite	I&I Connect	Cosmos	Cumulocity
Ecosystem	Provider	S	S	S	S	S	S	S	S	S	S	S
	Distributors (fi)	S	S	S	S	S	S	S	S	S	S	S
	Partners (fi)	S	S	+	+	-	+	-	-	+	-	+
	Ecosystem size	S	-	+	-	-	+	-	-	-	-	-
	Marketplace	S	-	-	-	S	-	-	-	-	-	-
	References (fi)	S	S	S	-	-	-	-	+	S	-	-
	Other notable references	S	-	-	-	S	-	-	-	-	-	S
	End-to-end solution available	S	S	-	S	-	-	-	S	-	S	S
Data collection HW	HW available from distributor	S	+	-	-	-	-	-	S	-	S	S
	WAN options	S	S	-	-	-	-	-	-	-	-	-
	Fieldbus connections	S	+	S	S	S	S	S	+	S	+	S
	Other connections	S	+	-	-	-	-	-	S	-	S	-
	Other sensors	S	+	-	-	-	-	-	-	-	S	+
	Device management	S	S	S	-	-	-	-	-	-	-	S
	Remote connections	S	S	-	-	-	-	-	-	-	-	-
	Look-and-feel	S	S	-	-	-	-	-	S	-	+	-
	DIN Installation	S	S	-	-	-	-	-	S	-	S	-
Data collection SW	SW available	S	-	S	-	-	-	-	-	-	-	-
	OPC supported	S	S	-	-	-	-	-	-	-	-	-
	Messaging protocols	S	-	-	-	-	-	-	-	-	-	-
	Connections to external services	S	-	-	-	-	-	-	-	-	-	-
	Data storage location	S	-	-	-	-	-	S	S	-	S	-
	Data security	S	S	S	-	-	-	S	-	-	S	S
Handling and analyzing data	Data modelling	S	-	-	-	-	-	-	-	-	-	-
	Automatic data handling	S	+	S	S	-	S	-	-	-	-	-
	Analytic tools	S	+	+	+	+	+	-	S	S	-	-
	Machine learning	S	S	+	+	+	+	S	S	-	-	-
	Utilizing analyzed data	S	-	S	S	S	S	-	S	S	-	-
	Data migration	S	-	-	-	-	S	-	-	-	-	S
Visualization	Creating views/dashboards	S	+	S	-	-	-	-	-	-	-	-
	Mobile devices	S	+	+	-	-	-	S	-	S	+	+
	Static reports	S	+	S	S	S	S	S	+	+	+	S
Service creation speed and models	Installation and configuration	S	S	-	-	S	-	-	-	-	-	S
	Need of coding	S	+	-	S	+	-	-	-	-	S	S
	Independent usage	S	S	-	-	S	-	-	-	-	S	S
	Data model management	S	+	-	-	-	-	-	-	-	-	+
Data security and rights management	Multitenancy	S	S	-	-	-	-	-	-	S	-	-
	User and rights management	S	S	S	-	-	-	-	-	-	-	-
	Access over all customer data	S	-	-	-	-	-	-	-	-	-	-
Scalability	Handling extending device count	S	+	S	S	+	S	-	S	S	+	+
	Cloud or local installation	S	S	-	-	+	-	-	-	-	S	-
	General scalability	S	S	+	+	+	+	-	S	S	-	S
Flexibility	Multiple tasks with one system	S	S	S	S	-	S	-	S	S	-	-
	Implementation of current concepts	S	-	S	S	-	S	-	-	-	-	S
	Effort for making changes	S	-	-	-	S	-	-	-	-	-	S
	General flexibility	S	-	S	S	-	S	-	-	-	-	-
Pricing	Pricing model	S	+	+	+	+	+	-	-	-	+	+
	One time costs	S	-	+	+	+	+	-	S	S	S	S
	Upkeep costs	S	S	S	S	+	S	-	-	-	S	S
	Costs/instance	S	-	-	-	-	-	-	-	-	-	-
	Costs/customer	S	+	-	-	-	-	-	-	S	-	-
	Costs/system	S	+	+	+	+	+	-	-	-	-	-
	Costs/user	S	+	+	+	+	+	+	S	S	S	S
Totals	Total +	0	16	10	8	8	9	1	3	2	6	6
	Total -	0	16	26	32	33	33	45	33	37	32	28
	Grand total	0	0	-16	-24	-25	-24	-44	-30	-35	-26	-22
	Total + w/o HW	0	12	10	8	8	9	1	2	2	4	5
	Total - w/o HW	0	16	19	24	25	25	37	29	29	29	23
Grand total w/o HW	0	-4	-9	-16	-17	-16	-36	-27	-27	-25	-18	

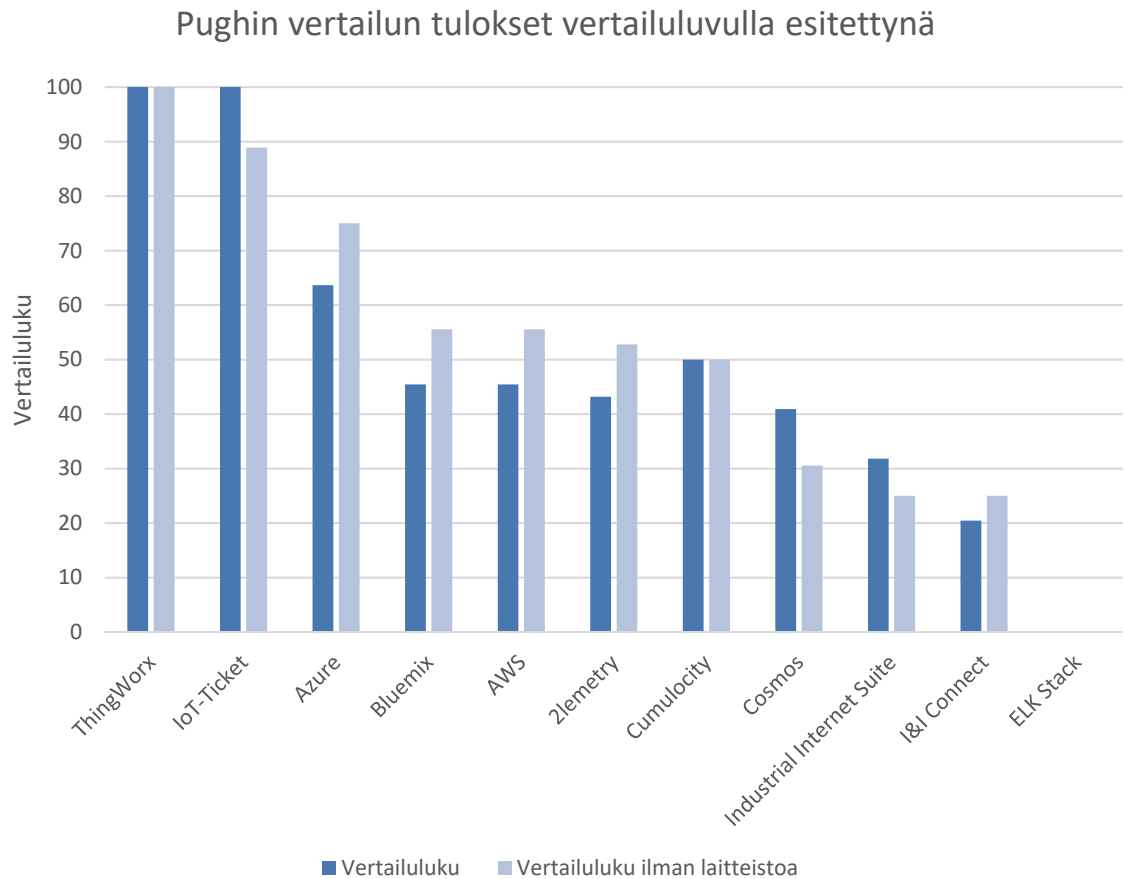
Kuva 14. Täytetty Pughin matriisi vertailuvaihtoehtoista

Kun keskinäinen vertailu oli tehty, laskettiin jokaisen vaihtoehdon saamat plus- ja miinusmerkit sekä niiden summa. Vertailua tehdessä nähtiin, että yksi kategorioista, tiedonkeruulaitteisto sekä sen saatavuus toimittajalta, on hyvin hankala vertailla. Suurimmalla osalla toimittajista ei ollut kokonaisratkaisua tarjottavana tai edes erillistä laitteistoa, jollaista Fastems olisi tarvinnut. Täten päätettiin erottaa laitteiston hankinta erilliseksi projektiksi ja suorittaa keskinäinen vertailu myös jättämällä tiedonkeruulaitteistoista annetut pisteet huomioimatta. Vaihtoehtojen saamat yhteispisteet, tiedonkeruulaitteistolla sekä ilman, ovat esitettyinä taulukossa 17 järjestettynä kokonaispisteiden mukaan, ilman laitteistovertailua, suurimmasta arvosta pienimpään.

Taulukko 17. Pughin vertailun kokonaispisteet

	Thingworx	IoT-Ticket	Azure	Bluemix	AWS	2lemetry	Cumulocity	Cosmos	Industrial Internet Suite	I&I Connect	ELK Stack
Yhteensä +	0	16	10	8	9	8	6	6	3	2	1
Yhteensä -	0	16	26	32	33	33	28	32	33	37	45
Kokonaispisteet	0	0	-16	-24	-24	-25	-22	-26	-30	-35	-44
Yhteensä + ilman laitteistovertailua	0	12	10	8	9	8	5	4	2	2	1
Yhteensä – ilman laitteistovertailua	0	16	19	24	25	25	23	29	29	29	37
Kokonaispisteet ilman laitteistovertailua	0	-4	-9	-16	-16	-17	-18	-25	-27	-27	-36

Kuten taulukosta 17 nähdään, vertailukohteeksi valittu Thingworx oli vertailun mukaan paras vaihtoehtoista. Jos vertailussa on mukana myös tiedonkeräyslaitteiston vertailu, ovat vaihtoehtojen keskinäiset erot suurempia johtuen hyvin vaihtelevasta laitetarjonnasta. Kun tiedonkeräyslaitteiden vertailu jätettiin pois, olivat keskinäiset erot huomattavasti pienempiä. Kuvassa 15 on esitetty tulokset palkkikaaviossa vertailuluvun avulla siten, että vertailun paras vaihtoehto Thingworx sai 100 pistettä ja vertailun huonoin vaihtoehto ELK Stack sai 0 pistettä ja välille jäävät tulokset ovat sovitettu vastaavalle välille. Tulokset ovat järjestetty paremmasta huonompaan vertailuluvun mukaan ilman tiedonkeruulaitteiston vertailua.



Kuva 15. Pughin vertailun tulokset kaaviona

Kun katsotaan vertailulukua ilman tiedonkeruulaitteistoa, kuvasta 15 nähdään, että vertailuvaihtoehtoista muodostuu kolme selkeää ryhmää. Parhaimmat kolme vertailuvaihtoehtoa (Thingworx, IoT-Ticket ja Azure) ovat selkeästi muita parempia ja seuraavat neljä vaihtoehtoa (Bluemix, AWS, 2lemetry ja Cumulocity) olivat tasaisesti noin puolessa välissä parhaan ja huonoimman vaihtoehdon välissä. Huonoimman vaihtoehdon (ELK Stack) ollessa suhteessa hyvin kaukana muista, tätä ennen olevat vaihtoehdot (Cosmos, Industrial Internet Suite ja I&I Connect) muodostivat myös selkeän tasavertaisen ryhmän.

Kun kuvan 15 kaaviosta katsotaan vertailulukua siten, että tiedonkeruunlaitteisto on mukana, on nähtävissä, että vaihtoehtojen keskinäinen järjestys muuttuu hieman, mutta ei radikaalisti. Tuloksista on kuitenkin nähtävissä, kuinka suuria eroja vertailuun tuli, jos toimittajalla oli ylipäättään tarjota laitteistoa, siitä huolimatta, ettei yksikään vertailussa olevista laitteista olisi soveltunut Fastemsin käyttöön. Tulokset ovat hyödyllisempiä ja paremmin soveltuvia valinnan tukena, kun tulokset esittävät vain asioita, joilla on merkitystä.

6.3 Muut tulokset

Vertailun aikana, niin vaihtoehtoja etsiessä kuin vertailua tehdessä, alustoja käsiteltiin kohtuullisen syvällisesti. Tämän ansiosta saatiin muodostettua hyvin kattava käsitys tutkimuksen aikaisesta IoT-alustatarjonnasta. Tarjonnan ymmärtämisen lisäksi ymmärrettiin alustojen maturiteettien erot ja miten pitkään markkinoilla oleminen vaikuttaa alustan ominaisuuksiin, ajatusmalleihin ja kehitykseen.

Alustoihin ja niiden tarjontaan tutustuminen auttoi myös ymmärtämään paremmin omia tarpeita. Samalla kun tutustui monen alustan ajatusmalleihin, suunniteltuihin käyttötaroituksiin ja jo toteutettuihin ratkaisuihin, auttoi se hahmottamaan omia tarpeita, joita oli helppo verrata alussa tehtyyn vaatimusmäärittelyyn. Referenssicaset- ja haastattelut olivat todella hyödyllisiä, sillä niiden avulla saatiin tietää, minkälaisia ratkaisuita vastaavat yritykset ja teollisuudenalat ovat tehneet ja millä alustoilla sekä teknologioilla. Myöskin käyttökokeiden tekeminen auttoi ymmärtämään eri alustojen ajatusmalleja paremmin. Käyttökokeiden aikana joutui miettimään, onko kyseisen alustan ajatusmalli sovellettavissa omien vaatimuksien mukaisten ratkaisuiden toteuttamiseen. Myöskin lyhyistä koe-käytöistä sekä toimittajien pitämistä demotilaisuuksista oli hyötyä kokonaiskuvan muodostamisessa.

7. JOHTOPÄÄTÖKSET

7.1 Löydökset vertailun tuloksista

Vertailussa ja sen tuloksia käsiteltäessä suurimpia haasteita tuotti vertailussa mukana olleiden IoT-alustojen erilaisuus. Koska ei ole muodostunut yleistä ja yhteistä käsitystä mitä IoT on ja tarkoittaa, yrittää jokainen alusta toteuttaa oman näkemyksensä. Tämä tarkoittaa sitä, että IoT-alustat voivat segmentoitua tahtomattaankin eri kategorioihin erilaisten ajatusmallien takia kuin esimerkiksi itse haluavat. Alustoilla oli erilaisia käyttötarkoituksia ja painotuksia kuten tiedon keräämiseen, sen tallentamiseen, analytiikkaan tai sen visualisointiin. Usein alustat mainostivat kokonais kattavaa IoT-ratkaisua, mutta kykenivät käsittelemään vain numeerista, sensoreiden lähettämään informaatiota. Kuitenkin IoT:n ollessa paljon muutakin kuin pelkkiä lukuarvoja, moni alusta rajasi paljon käyttötapauksia ulkopuolelleen, etenkin jos alustalla haluttaisiin käsitellä muutakin informaatiota esimerkiksi tekstimuodossa. Fastemsin tapauksessa tämä tarkoitti järjestelmästä kerättyjen virheilmoitusten ja hälytyksien käsittelyä. Juurikin tässä erilaisen informaation käsitelyssä alustan maturiteetti oli parhaiten nähtävissä. Pitkään markkinoilla olleet alustat olivat selkeästi tietoisia esimerkiksi teollisuuden tarpeista ja usein näihin selkeästi segmentoituivatkin, kun taas uudet, lyhyen aikaa olemassa olleet alustat markkinoivat ratkaisuun soveltuviksi kaikille ja useimmiten sisälsivät vain numeerisen tiedon käsittelyn ominaisuudet. Osalla alustoilla oli myös panostettu erityisen paljon keräyslaitteiden hallintaan ja usein näiden tapauksessa itse tiedon käsittely, etenkin analysointi ja visualisointi oli jätetty vähemmälle huomiolle.

Päänvaivaa aiheuttivat myös hyvin erilaiset hinnoittelumallit, jotka saattoivat olla hyvin monimutkaisia ja monesta asiasta riippuvia. Kuitenkin alustan tuottamia kustannuksia haluttiin arvioida kokonaisuudessaan, joten joidenkin alustojen kohdalla jouduttiin miettimään todella yksityiskohtaisia sekä pieniä yksityiskohtia, kuten tiedonsiirtomääriä kuukaudessa tai kerättävien datapisteidän määriä. Kuitenkaan vielä tässä kohdassa projektia ei tiedetty mitä kaikkea projektin toteutusvaiheen alkaessa tullaan tekemään tai mitä tietoa silloin kerätään. Tästä syystä hinta-arvioita jouduttiin tekemään parhaan tietämyksen mukaisesti sekä esimerkiksi käyttöttestien aikana kerättyjen tietojen pohjalta. Laskelmia tehtiin myös pahimman skenaarion pohjalta siten, että esimerkiksi jokaiselle Fastemsin toimitetulle järjestelmälle laskettiin arvoitu tallennustila ja tiedonsiirtomäärä, jonka mukaan saatiin luvut, jotka vastaisivat kustannuksia, kun järjestelmä on skaalattu lähelle teoreettista suurinta Fastemsin käyttötapauksia. Nämä laskennat kuitenkin antoivat hyvää suunta-arviota sille, miten kustannukset tulisivat jokaisella alustalla skaalautumaan, sillä skaalautumisen mallit olivat alustojen välillä hyvin erilaiset. Joillain alustoilla hinnoittelu ei skaalautunut lainkaan käytön kasvaessa, joillain skaalautuminen tapahtui hyvin korkei-

den kertamaksujen kautta esimerkiksi lisenssien muodossa ja joillakin kustannukset skaalautuivat yksinkertaisesti käytön määrän mukaan, esimerkiksi tallennetun datan tai siirretyn tiedon määrän mukaan.

Vertailua myös vaikeutti se, että joitakin vertailussa käytettyjä tietoja oli vaikeuksia saada kerättyä alustoista tai niiden toimittajilta. Useassa tapauksessa vastaus oli, että kysytty ominaisuus oli tulossa tai työ alla tai että ominaisuus olisi räätälöitävissä toimittajan tekemänä. Kuitenkin, koska alustan valinta ja käyttöönotto oli jo aikataulutettu, tarkoittaen, että valinta oli tehtävä lyhyellä aikavälillä, tulkittiin nämä vastaukset ominaisuuden puuttumiseksi, etenkin jos toimittajalla ei ollut antaa tarkkaa aikataulua, koska kyseinen ominaisuus olisi saatavilla tai jos sen julkaisu oli liian kaukana tulevaisuudessa. Toimittajan räätälöintityö ei ollut vaihtoehto, sillä jo vaatimusmäärittelyssä oli vaatimus siitä että, alustaa ei räätälöidä vaan ominaisuudet pitää olla korkeintaan konfiguroitavissa ohjelmoinnin sijasta. Myös tiedon saantiin vaikutti alustan maturiteetti siten, että pidempään markkinoilla olleilla alustoilla oli esittää tarkemmat kehityssuunnitelmat ja ne olivat myös paremmin konfiguroitavissa. Näistä pidempään kehittyneistä alustoista oli myös enemmän tietoa saatavilla, kun taas uusista alustoista joitakin tietoja ei vain yksinkertaisesti ollut edes toimittajan tiedossa, vaan tietoa jouduttiin kyselemään myös paljon alustojen tuottajilta.

Suurimmassa osassa alustoja ei ollut minkäänlaisia työkaluja tiedon visualisointiin tai raportointiin. Ajatusmallina alustoissa oli useimmiten, että tieto viedään alustoista eteenpäin toisiin järjestelmiin hyödynnettäväksi sekä esitettäväksi. Kuitenkin Fastems halusi tehdä kevyttä sekä nopeaa käyttöliittymäkehitystä sekä kevyttä tiedon visualisointia sen ymmärtämisen helpottamiseksi. Kuitenkin näiden työkalujen integroinnin puute aiheutti sen, että moni alusta ei itsessään riitä vaan lisäksi tarvitaan muita komponentteja, sovelluksia tai palveluita. Näiden ulkoisten työkalujen integrointi ja yhteensovittaminen esimerkiksi ulkoisten rajapintojen kautta tuottaa välittömästi lisätyötä ja näin vaikeuttaa nopean prototypoinnin ja loppusovelluksien kehittämistä.

Yksi yllättävä löydös tutkimuksen aikana oli, että moni alustatoimittajien tai -kehittäjien listaamista yritysreferensseistä olivat loppujen lopuksi merkityksettömiä. Lähes kaikilla alustoilla oli asiakasreferensseissään isoja ja tunnettuja yrityksiä, mutta vertailua tehdessä kävi ilmi, että suurin osa näistä yrityksistä oli vain kokeillut alustaa esimerkiksi käyttökokeilun tai Proof of Concept -projektin (PoC) muodossa. Vain pieni osa mainituista referensseistä oikeasti oli ottanut alustan käyttöön. Tämä todennäköisesti johtuu isompien yritysten suurista resursseista ja kyvykkyydestä tehdä laajempi vertailu alustojen välillä sisältäen kaikkien alustojen käyttökokeilut, toisin kuin Fastemsilla, jolla oli mahdollista käyttää vähemmän resursseja alustavertailuun.

Suurin yksittäinen ongelma vertailun aikana oli tiedonkeruulaitteiston hankinta. Alkuperäisessä ideassa valitulta IoT-alustatoimittajalta hankittaisiin alustan lisäksi myös alus-

taan täysin integroitava tiedon keruulaitteisto, mikä tarjoisi myös yhteyksien muodostamiseen tarvittavat konnektiviteetti-ominaisuudet, kuten WLAN ja mobiiliverkot. Kuitenkin vain yhdellä toimittajalla oli tarjottavana Fastemsin tarpeisiin sopiva kokonaisratkaisu ja vain muutamalla vaihtoehdolla oli ylipäättään laitteistoa tarjottavana. Tästä syystä vaatimuksia muokattiin siten, että alkuvaiheessa ei hankittaisi erillistä tiedonkeruulaitteistoa ja luovutettiin suorasta integraatiosta järjestelmän käyttämään kenttäväylään. Tämän sijasta päätettiin aloittaa tiedon keräys hyödyntäen järjestelmässä olevaa Microsoft Windows -palvelinympäristöä, jonka sovelluskehityksestä oli Fastemsiilla jo valmiiksi osamista, ohjausjärjestelmän ollessa kehitetty vastaavalle ympäristölle. Fastemsin toimittamissa automaatiojärjestelmissä on useampikin tiedonkeräyskäyttöön soveltuvaa Windows-palvelinta, joita voidaan ainakin alkuvaiheessa hyödyntää tiedon keräyksessä. Tiedon siirtoa IoT-alustaan varten tarvittavan konnektiviteettiratkaisun hankinta ja arviointi päätettiin eriyttää IoT-alustan hankinnasta, sillä konnektiviteettiratkaisua tulisi hyödyntämään myös muut käyttötapaukset kuin pelkkä tiedonkeruu, kuten esimerkiksi käyttötuen antamiseen käytetty etätyöpöytäyhteyssovellus. Erillinen ratkaisu tulee olemaan myös pitkäikäisempi ja joustavampi kuin alusta- ja toimittajakohtainen laiteratkaisu. Koska molemmat ratkaisut, IoT-alusta ja konnektiviteettiratkaisu riippuvat toisistaan, luoden vaatimuksia ja rajoitteita toisilleen, tehtiin konnektiviteettiratkaisun valinta rinnakkaisena, yhtä aikaa tapahtuvana prosessina. Konnektiviteettiratkaisun valintaprosessi on kuitenkin rajattu tämän työn ulkopuolelle. Kun konnektiviteettiratkaisua ei arvioitu yhdessä IoT-alustan kanssa, muutti tämä hieman myös vertailun tuloksia. Kuvassa 15 on nähtävissä, että kun tiedonkeruulaitteistosta lasketut pisteet poistettiin vertailusta, lähes puolen vertailussa mukana olleiden alustojen vertailuluvut nousivat. Tämä siitä syystä, että laitteiston saatavuuden puute laski näiden alustojen vertailulukua. Vastaavasti niillä alustoilla, joilla oli tarjota tiedonkeruulaitteistoa, vertailuluku laitteisto huomioon ottaen laskivat. Vertailusta ja siten myös tuloksista tuli merkityksellisemmät Fastemsin kannalta ja siten hyödyllisemmät alustan valintapäätöstä tehdessä.

7.2 Vaihtoehtojen karsinta

Vertailuun valittiin mukaan 11 IoT-alustaa ja vertailun päätteeksi valittiin näistä alustoista kolme lupaavinta alustaa. Nämä kolme alustaa olivat Thingworx, Cumulocity ja IoT-Ticket. Valinta tehtiin Pughin vertailumatriisin tuloksia apuna käyttäen, mutta ei suoraan vertailussa saatujen pisteiden perusteella. Kuten kuvasta 15 on nähtävissä, Thingworx sekä IoT-ticket olivat kaksi parasta alustaa, mutta Cumulocity oli vertailussa keskitasoa. Kuitenkin Cumulocity oli todennäköisemmin parempi vaihtoehto Fastemsin käyttöön kuin vertailussa moni edellä ollut alustavaihtoehto. Taulukossa 18 on nähtävissä näiden kolmen viimeisessä karsinnassa mukaan otettujen alustojen vertailu jokaisen pääkategorian yhteen laskettujen pisteiden avulla. Taulukkoon 18 on myös merkitty jokaisen kategorian paras tulos vihreällä värillä ja huonoin tulos punaisella värillä.

Taulukko 18. Kolmen parhaan alustavaihtoehdon vertailutaulukko

	Thingworx	IoT-Ticket	Cumulocity
Ekosysteemi	0	-3	-2
Tiedonkeruusovellukset	0	-4	-5
Tiedon käsittely ja analysointi	0	-1	-5
Tiedon visualisointi	0	3	0
Palvelukehityksen nopeus ja kehitysmallit	0	2	1
Tietoturva ja pääsynhallinta	0	-1	-3
Skaalautuvuus	0	1	0
Joustavuus	0	-3	-2
Hinnoittelu	0	2	-2
Kokonaispisteet ilman keräyslaitteistoa	0	-4	-18

Vertailussa oli mukana useampi vaihtoehto, jonka IoT-tarjoama keskittyi enemmän infrastruktuurin tarjoamiseen palveluna, johon sisältyi IoT-toiminnallisuuksia. Näissä vaihtoehtoisissa toiminnallisuuksissa piti itse rakentaa sekä yhdistellä kokonaisuudeksi. Useimmissa tapauksissa tähän infrastruktuurin rakentamiseen olisi tarvittu ulkopuolista apua esimerkiksi konsultaatiotyön muodossa, jolloin vaatimus itsenäisestä käytöstä ei täyttyisi. Nämä alustat olivat, Microsoft Azure, IBM Bluemix sekä Amazon AWS. Vertailussa nämä infrastruktuuriin keskittyneet alustat olivat hyvin korkealla sijalla kahdesta syystä. Näiden alustojen toiminnallisuuksien sisälsivät todella paljon toiminnallisuutta ja mahdollistivat hyvin joustavan ja kattavan järjestelmän kasaamisen. Toinen syy korkeisiin pisteisiin olivat näiden alustojen skaalautumiskyvykkyys. Microsoft, IBM sekä Amazon ovat suurimmat markkinoilla olevat kapasiteetti- sekä laskentapalveluiden tarjoajat, joten skaalautuminen on käytännössä rajatonta näiden alustojen osalta. Vastaavasti, jokaisen vaihtoehdon kohdalla vertailun pisteitä laskivat erityisesti huono käyttäjäystävällisyys sekä korkea aloituskynnys. Kuitenkin kaikki näistä infrastruktuurialustoista olivat hyvin samankaltaisia ja se on myös nähtävissä taulukosta 17, missä näiden alustojen saamat pisteet eroavat viidellä pisteellä Azuren hyväksi, AWS:n ja Bluemixin saadessa saman verran pisteitä.

Cumulocityä vielä edellä ollut 2lemetry jätettiin karsinnassa huomioimatta, sillä vertailun tekemisen aikana Amazon ilmoitti ostavansa 2lemetryn kokonaisuudessaan. Amazonin ilmoituksen mukaan 2lemetryn toiminnallisuuksien tullaan integroimaan nopealla aikataululla osaksi AWS:n IoT-tarjontaa. Amazonin ja 2lemetryn muutoksen aikataulun ollessa kuitenkin avoin, laskettiin tämän olevan tämän vertailun ajan ulkopuolella. Molemmat alustat pärjäsivät hyvin vertailussa erillisinä alustoinakin, joten lopputulos olisi voinut olla hyvin erilainen näiden kahden yhdistyessä, oletettavasti korostaen molempien alustojen hyviä puolia.

Viimeiseen arviointiin valittiin vertailun ensimmäiseltä sijalta PTC Thingworx. Thingworxin todettiin olevan hyvin joustava ja helposti laajennettavissa jokaiselta IoT-tekniologiapinon kohdalta. Esimerkiksi visualisointipuolelle oli helppo tehdä omia esityskomponentteja, visualisointipuolen ollessa yksi parhaista. Näkymien ja käyttöliittymien tekeminen onnistui ohjelmoimatta graafisella Mashup builder -ominaisuudella. Itse alustan logiikan ja tietomallien hallinta onnistui helposti konfiguroimalla, mutta alustassa on myös mahdollisuus ohjelmoida toiminnallisuutta Javascript-kielellä, mikä tarvittaessa lisää alustan joustavuutta huomattavasti. Ylipäättään Thingworxissä oli vertailun parhaat mallinnustyökalut tiedon, tietomallien ja laitteiden hallintaan. Alustan laajan ekoympäristön ansiosta tuki- ja konsultointipalveluita on helppo hankkia ympäri maailmaa, myös Suomesta. Hinnoittelun osalta Thingworx oli vertailun kalleimpia alustoja, mutta hinnoittelumalli ja sen skaalautuminen olivat erityisen sopivat Fastemsin käyttöön.

Toiseksi vaihtoehdoksi karsinnasta valittiin vertailussa toiseksi tullut Wapice IoT-Ticket. IoT-Ticketin vahvuuksia olivat helppokäyttöinen sekä näyttävä käyttöliittymätoiminnallisuus, kyvykäs tiedon keräyslaitteisto sekä alustan kotimaisuus. Laitteisto oli vertailun ainoa, joka täytti täysin määritellyt vaatimukset, mutta olisi vaatinut hieman räätälöintityötä. Alustan keskittyminen yksittäisten, numeeristen tietopisteiden käsittelyyn sekä analysointiin oli hyvin toteutettu, mutta kaikki Fastemsin vaatimukset, kuten tekstimuotoisen tiedon käsittely ei täytynyt. Analysointityökalut olivat vertailun aikana hyvin kyvykkäät numeerisen tiedon käsittelyyn ja esittämiseen, mutta analysoidun tiedon ja informaation hyödyntäminen ei ollut lainkaan mahdollista. Kokonaisuudessaan IoT-Ticketissä oli paljon potentiaalia, mutta alusta vielä vertailun aikaan liian uusi ja siten kehitysvaiheessa, usean ominaisuuden vielä tuntuen raakileelta ja alustalla käsiteltävä tieto oli hieman liian yksinkertaista.

Kolmas lopulliseen karsintaan valittu alusta oli Software AG Cumulocity. Cumulocityssä oli panostettu erityisesti keräyslaitteistojen hallintaan sekä näyttävään esitys- ja loppusovelluspuoleen. Visualisointi ja käyttöliittymät pitää kuitenkin itse ohjelmoida web-tekniologioilla kokonaisuudessaan, joten joustavasta ratkaisusta ei voida puhua. Kuten IoT-Ticketin tapauksessa, Cumulocityllä ei pysty käsittelemään kuin numeerista tietoa, jolloin puhutaan vain yksinkertaisista toteutuksista. Tiedon käsittelyyn ei ole tarjolla analysointityökaluja mutta sisään tulevaan tietoon voidaan liittää yksinkertaisia sääntöjä. Sääntöjä on kuitenkin vain hyvin rajallinen määrä, eikä niiden itse tekeminen ollut mahdollista lainkaan. Tiedon mallinnukseen Cumulocity tarjosi alilaite-käsitteen, mutta Fastemsin käyttöön se on liian yksinkertainen. Myöskään kustannuksien puolesta alilaite-käsite oli ongelmallinen, sillä jokainen alilaite laskettiin omaksi, erilliseksi laitteekseen, josta olisi pitänyt maksaa erillinen laitemaksu. Toisin kuin IoT-Ticket, Cumulocity oli ollut pitkään markkinoilla, mutta tuntui IoT-Ticketin mukaisesti keskeneräiseltä tuotteelta, jolla oli huomattavan pitkä lista kehityskohteita. Koska näillä tulevaisuudessa luvatuille ominaisuuksille ei onnistuttu saamaan edes aikatauluarvioita, jätettiin ne pitkälti huomioimatta vertailun aikana.

8. IOT-ALUSTAN VALINTA JA VALIDOINTI

8.1 Valittu IoT-alusta

Mukana olleista vaihtoehtoista parhaiten Fastemsin käyttöön soveltuvaksi alustaksi todettiin PTC Thingworx. Lopputulokseen päädyttiin osittain vertailun tuloksien perusteella, mutta koekäyttöjen aikana tehty alustan käytännön testaus erotti käytettävyydeltään ja joustavuudeltaan muista vaihtoehtoista. Fastemsin käyttötarkoituksen mukaisen joustavan valmistusjärjestelmän mallintaminen onnistui jo koekäytön aikana erityisen helposti ja nopeasti. Myös suuren, teollisuuden keskittyvän kehittäjän sekä laajan toimittajaverkoston takia Thingworx on turvallinen ja pitkäikäinen valinta, etenkin verrattuna nuoriin start-up -yritykseen, jotka vielä yrittävät vakiinnuttaa asemaansa IoT-alustamarkkinoilla.

Thingworxin arkkitehtuuri ja sen mahdollistama kattava alustan laajentaminen ovat edellytykset sille, että alustalla voidaan toteuttaa myös kompleksisia projekteja ja kehityshankkeita. Parhaimmillaan näiden toteuttamiseen ei tarvita lainkaan ulkoista tukea, vaan jopa loppukäyttäjäsovelluksien tuottaminen onnistuu suoraan alustasta poistumatta. Valintaan vaikutti myös se, että Thingworxissa oli vertailun kattavimmat kommunikaatio-protokollat niin tiedon keräämiseen tai sen jakeluun erilaisten rajapintojen kautta. Myös valmiita ohjelmakirjastoja löytyi suoraan Fastemsin hyödynnettäväksi esimerkiksi tiedonkeruusovelluksen toteuttamiseen.

Vaikka vertailun ja valintaprosessin lopputuloksena valittiin yksi, oletettavasti parhaiten käyttöön sopiva alusta, ei voida olla varmoja, että valinta oli millään kriteereillä paras mahdollinen. Tästä syystä kappaleessa 8.2 validoidaan valittu alusta, eli varmistustaan siitä, että Thingworx täyttää sille määritellyt vaatimukset. Näin saadaan varmistus sille, että oliko kyseinen valinta ylipäätään sopiva käyttötarkoitukseensa. Koska vertailussa oli mukana vain määrällinen otos alustoista, on mahdollista, että joku löydetyistä mutta esikarsintavaiheessa vertailun ulkopuolelle jätetyistä alustoista olisi kuitenkin pärjännyt vertailussa paremmin. On myös mahdollista, ettei sopivinta tai parasta alustaa löydetty kartoitusvaiheessa lainkaan. Myös vaatimusmäärittelyn aikana on voitu tehdä virheitä tai vaatimuksia ei olla ylipäätään saatu kerättyä riittävällä tasolla. Valitun alustan käyttöön otto tämän työn ulkopuolella, joten nähtäväksi jää, kuinka valittu Thingworx-alusta oikeassa käytössä pärjää ja minkälaisia haasteita todellisten palveluiden toteuttaminen ja kehittäminen tuottavat.

8.2 Valinnan validointi

8.2.1 Validointiprosessi

Sovellun V-mallin prosessin mukaisesti vertailun ja sen perusteella tehdyn valinnan jälkeen suoritetaan valitun ratkaisun validointi, jossa kuvan 4 mukaisesti verrataan teknisiä vaatimuksia vaatimusmäärittelyn tuloksiin sekä liiketoiminnan tavoitteita yleisiin konsepteihin ja tavoitteisiin. Näin voidaan varmistua siitä, että valittu vaihtoehto täyttää sille määritellyt vaatimukset ja on siten sopiva valinta.

Validointimenetelmäksi valittiin ohjelmistokehityksessä laajasti käytetty testitapaukset (engl. test case). Testitapauksissa käydään läpi yksittäinen skenaario tai tilanne, joka yleensä on joko suoraan vaatimuksista tai niistä johdettu. Testitapauksella on määritelty jonkun tapahtuman syöte, oletettu lopputulos ja realisoitunut lopputulos. Vertaamalla onko realisoitunut lopputulos oletettu, voidaan tehdä johtopäätöksiä, toimiiko testattava järjestelmä oikein eli vaatimuksissa määritellyllä tavalla. (Kaner 2003). Tässä validointiprosessissa tehdään kaksi erillistä validointia. Ensin kirjoitetaan testitapaukset ylätason teknisiä vaatimuksia varten ja varmistutaan, että alusta täyttää teknisesti ylätason vaatimukset. Vastaavat testitapaukset kirjoitetaan myös liiketoiminnan tavoitteita vasten, jolloin nähdään täyttääkö valinta liiketoiminnan alkuperäiset tavoitteet.

Testitilanteet voidaan dokumentoida hyvin vapaasti käyttötarkoituksen mukaan. Jokaista testitilannetta voidaan kuvastaa korttien avulla, jossa on kuvattu vähintään jokaisen testitilanteen tunniste, syöte, oletettu lopputulos ja realisoitunut lopputulos. Usein testitilanteet esitetään myös taulukkorakenteessa, jolloin voidaan esittää enemmän informaatiota pienemmässä tilassa. Tässä validointiprosessissa käytetään taulukon 19 mukaista yksinkertaistettua taulukkorakennetta testitapauksien esittämiseen.

Taulukko 19. *Esitys validointiprosessissa käytetystä testitapaustaulukosta*

	Oletettu lopputulos	Realisoitunut lopputulos	Vaatus täyttynyt
Testitilanne 1			
Testitilanne 2			
Testitilanne 3			

Hyvin yksinkertaistetussa taulukossa on kuvattu testitilanne, tässä tapauksessa vaatimus tai tavoite. Taulukon seuraavassa sarakkeessa on lyhyesti kirjoitettu oletettu lopputulos eli mitä valitun alustan pitäisi tehdä täyttääkseen vaatimuksen tai tavoitteen. Realisoituneessa lopputuloksessa on kuvaus siitä, miten valittu alusta toimii kyseisen tilanteen kohdalla. Viimeisessä sarakkeessa on päätelmä siitä, lasketaanko vaatimus täyttyneeksi.

8.2.2 Validointi teknisiä vaatimuksia vasten

Valitun alustan validointia teknisiä vaatimuksia varten, kappaleessa 3.3.3 esitetyt teknisten vaatimusten ylätasoinen otsikot kerättiin testitapaustaulukkoon ja kirjoitettiin lyhyesti auki vaatimuksen kuvaileman tehtävän oletettu lopputulos. Realisoitunut lopputulos kirjoitettiin vastaamaan valitun Thingworx-alustan avulla saavutettua toteutuksen tasoa. Täytetty teknisten vaatimusten validointi ja testitapaustaulukko on nähtävissä taulukossa 20.

Taulukko 20. Teknisten vaatimusten validointi- ja testitapaustaulukko

	Oletettu lopputulos	Realisoitunut lopputulos	Vaatus täyttynyt
Uudet tietoliikenneyhteydet järjestelmiin	Alustatoimittajalta on saatavilla laiteistoratkaisu tiedon keräämiseen ja konnektiviteettiin.	Soveltuvaa laitteistoa ei ole saatavilla toimittajalta.	Ei
Tietoturva	Käytettävät tietoturvamenetelmät ovat riittäviä ja oikeushallinta mahdollistaa asiakkaiden pääsyn järjestelmään turvallisesti.	Kaikki tietoliikenne on salattua ja tieto on suojattu asianmukaisesti. Käyttäjien hallinta on monitasoinen mahdollistaen tarkan ja tehokkaan oikeus- ja pääsynhallinnan.	Kyllä
Tiedon automaattinen käsittely	Tietoa on mahdollista käsitellä ja jalostaa automaattisesti siten, että informaatiota voidaan hyödyntää automaattisissa reagoineissa ja hälytyksissä.	Tiedon käsittelyyn voidaan tehdä ohjelmallista logiikkaa sen heti sen saapuessa alustaan tai sitä hakea myöhemmin hyvin joustavasti automaattisia mekanismeja varten.	Kyllä
Tiedon visualisointi	Alustalla on helppo luoda näkymiä, käyttöliittymiä ja loppukäyttäjäsovelluksia tiedon esittämiseen sekä toiminnallisuuksien hallintaan.	Alustassa on helppo konfigurointiin ja graafiseen mallinnukseen perustuva työkalu loppukäyttäjäsovelluksien nopeaan luontiin ja kehittämiseen.	Kyllä
Standardoitu ratkaisu	Alustalla toteutettu ratkaisu soveltuu suurimpaan osaan Fastemsin toimitusprojekteista sellaisenaan tai mahdollisimman pienin muutoksin.	Toimitettujen järjestelmien mallinnus onnistuu monella tasolla vakiokomponentein. Toimintalogiikan uudelleen käyttäminen ja yhdistely on mahdollista oliomaisella luokkien perinnällä ja laajentamisella. Malien muokkaaminen kaikille järjestelmille yhtä aikaa on mahdollista.	Kyllä
Joustava ratkaisu	Alustalla pystyy tarvittaessa poikkeamaan standardoidusta ratkaisusta sekä tekemään täysin erilaisia ratkaisuita ja sovelluksia.	Alustan toiminnallisuus on eriytetty toisistaan eli usean eri ratkaisun ja sovelluksen tekeminen on mahdollista. Standardoitu	Kyllä

		doitujen ratkaisuiden toimintalogiikkaa voi korvata tapauskohtaisesti mahdollistaen yksilöinnin.	
--	--	--	--

Taulukosta 20 on nähtävissä, että yhtä kohtaa lukuun ottamatta, valittu alusta täyttää sille määritellyt tekniset vaatimukset. Vaatimus uusista tietoliikenneyhteyksistä ei toteutunut, eli valitulta alustatoimittajalta ei ollut saatavilla konnektiviteetin luomiseen tai tiedon keruuseen fyysistä laitteistoa. Koska vertailussa saatujen tuloksien perusteella tehtiin päätös, että huonon laitteistotarjonnan takia ainakaan aluksi ei erillistä tiedonkeruulaitteistoa hankita, vaan sen sijasta tiedon keräys tehdään sovelluspohjaisesti järjestelmän ohjaus-tietokoneella. Koska tehtiin myös päätös eriyttää konnektiviteettiratkaisun hankinta IoT-alustan hankinnasta, joten käytännössä tämä ainoa täyttymätön vaatimus on merkitsemätön. Täten voidaan todeta, että valittu ratkaisu täyttää tekniset vaatimukset erinomaisesti ja oli siten sopiva valinta Fastemsin käyttöön teknisestä näkökulmasta.

8.2.3 Validointi liiketoiminnan tavoitteita vasten

Valitun alustan validointi Fastemsin liiketoiminnan tavoitteita vasten suoritettiin vastaavasti teknisten vaatimuksien validoinnin kanssa. Kappaleessa 3.1.2 esitetyt liiketoiminnan tavoitteet ylätasolla ja niiden oletetut lopputulokset kirjoitettiin validointi- ja testitapaustaulukkaan, joka on esitetty taulukossa 21. Vastaavasti lopputulos käyttäen Thingworx-alustaa kirjattiin realisoituneen lopputuloksen sarakkeeseen.

Taulukko 21. *Liiketoiminnan tavoitteiden validointi- ja testitapaustaulukko*

	Oletettu lopputulos	Realisoitunut lopputulos	Vaatimus täyttynyt
Yrityksen strategian toteuttaminen	Digitalisaation hyödyntäminen ja uuden liiketoiminnan mahdollistaminen. Toiminnan laajentuminen oman, toimitetun tuotantojärjestelmän ulkopuolelle.	IoT-alustan käyttöönotto on hyödyllinen niin markkinointitarkoituksissa kuin uudenlaisen, tietoon perustuvan liiketoiminnan mahdollistamisessa. Valittu alusta mahdollistaa myös Fastemsin järjestelmän ulkopuolisten laitteiden tiedon keräämisen ja käsittelyn.	Kyllä
Järjestelmissä olevan tiedon hyödyntäminen	Yhteyksien luominen järjestelmiin ja tiedon kerääminen keskitettyyn tietovarastoon järjestelmän ulkopuolelle. Tiedon merkityksellinen jalostaminen informaatioksi ja sen hyödyntäminen Fastemsin liiketoiminnan tarkoituksiin.	Tieto saadaan kerättyä järjestelmistä pilvipalveluun ja käsiteltyä automaattisesti eri tavoin. Alustassa on mahdollista toteuttaa useita sovelluksia vastaamaan liiketoiminnan tarpeisiin.	Kyllä

Nopea ja ketterä palvelukehitys	Loppukäyttäjäsovelluksien ja uusien palveluiden kehitys voidaan tehdä nopeasti ja ketterin menetelmin pienin kehitysaskelein.	Valittu IoT-alusta mahdollistaa erilaisten sovelluksien ja ratkaisuiden toteuttamisen helposti ilman raskaita sovelluskehitysprojekteja. Uusien konseptien kehitys ja testaus asiakkaiden kanssa on helppoa asiakkaiden päästessä alustaan ja uusien kehitysversioiden julkaisu on helppoa.	Kyllä
--	---	---	-------

Taulukosta 21 nähdään, että Thingworx-alusta täyttää kaikki liiketoiminnalta kerätyt tavoitteet. Koska liiketoiminnan tavoitteet on kerätty hyvin korkealla tasolla, täyttyvät nämä vaatimukset hyvin helposti. Koska tavoitteet koskevat yleisesti IoT-aktiviteetteja ja digitalisaatiota, noin puolet vertailussa mukana olleista alustoista olisi täyttänyt kaikki nämä vaatimukset. Loput alustoista ei olisi täyttänyt viimeistä taulukon 21 kohtaa nopeasta ja ketterästä palvelukehityksestä, sillä näissä alustoissa jokainen muutos olisi todennäköisesti ollut oma ohjelmointiprojektinsa, mikä pahimmassa tapauksessa olisi vaatinut kolmannen osapuolen apua tai kehitystyötä. Kuitenkin validoinnin tuloksena voidaan todeta, että kun valittu Thingworx-alusta täyttää kaikki liiketoiminnan määrittämät tavoitteet, on se siten sopiva valinta Fastemsin käyttöön liiketoiminnan näkökulmasta.

9. VALINTAPROSESSIN JA TUTKIMUKSEN ARVIOINTI

9.1 Valintaprosessin arviointi

Siitä huolimatta, että kehitetty valinta- ja vertailuprosessi käytiin läpi vain yhden kerran, validointivaiheessa nähdään, että kehitetty prosessi täyttää alkuperäiset ja määritellyt tarpeet. Tämä tietenkin edellyttää, että alusta asti tarpeet sekä vaatimukset ovat kerätty oikein, oikeilta sidosryhmiltä ja ovat merkitseviä siten, ettei vaatimusmäärittelyprosessissa keskitytä epäolennaisiin vaatimuksiin. Pahimmassa tapauksessa vaatimusmäärittelyssä tehdyt virheet ilmenevät vasta valitun ratkaisun käyttöönotossa, kun oikeat tarpeet tulevat toteutusvaiheessa esille.

Myös alustojen esivalinnassa on mahdollisuus valintaprosessin epäonnistumiseen siten, että vertailuvaihtoehtoiksi ei löydetä sopivia vaihtoehtoja tai valitaan kokonaan vääränlaisia vaihtoehtoja. Tästä syystä kartoitusvaiheessa käytettävät esivalintakriteerit pitää kohdistaa tarkasti vaatimusmäärittelyssä tehdyille pohjalle. Esimerkkinä liian laajan sovelluskohteen valinta voi aiheuttaa sen, että vaihtoehdot ovat liian erilaisia järkevää vertailua varten. Toisaalta myös liian tiukan sovelluskohteen rajaamisen tuloksena voidaan jättää vertailun ulkopuolella hyviä vaihtoehtoja, sillä IoT-alustojen ollessa hyvin joustavia ja suunnitellut ratkaisut vaihtelevat hyvinkin paljon IoT-alustaa etsivien tahojen välillä. Tästä syystä vertailukohteita ja hyvin samanlaisia sovelluksia voi löytyä hyvinkin erilaiselta teollisuudenalalta kuin missä valintaprosessia tekevä taho toimii.

9.2 Tutkimuksen arviointi

Tutkimuksessa haettiin vastauksia seuraaviin tutkimuskysymyksiin:

- Miten valitaan mahdollisimman hyvä IoT-alusta FMS-järjestelmän digitalisoinnin työkaluksi?
- Miten IoT-alustan vaatimusmäärittely, vertailu ja valitun ratkaisun validointi voidaan toteuttaa?
- Mitä erityisvaatimuksia joustava valmistusjärjestelmä tuottaa valintaprosessiin?

Tutkimuskysymykseen ”Miten valitaan mahdollisimman paras IoT-alusta FMS-järjestelmän digitalisoinnin työkaluksi?” löydettiin ratkaisuksi seurata ohjelmistotuotannossa yleisesti käytettyjä menetelmiä ohjelmistojen hankintaan. IoT-alusta kuitenkin on ohjelmistotuote, mutta jonka käyttötarkoitusta eikä sisältöä ei kuitenkaan ole kovin tarkasti määriteltä. Ohjelmiston hankinnassa keskitytään yleensä enemmän hankkijan vaatimuk-

siin sekä toiminnallisuuteen kuin teknisiin yksityiskohtiin. Tässä tapauksessa ei vielä ollut tiedossa tarkkoja loppusovelluksia, joita alustalla oltiin tekemässä, keskityttiin enemmän teknisiin toimintoihin tekemisen mahdollistajina.

Toiseen tutkimuskysymykseen ”*Miten IoT-alustan vaatimusmäärittely, vertailu ja valitun ratkaisun validointi voidaan toteuttaa?*” vastaus oli jälleen seurata tarkemmin ohjelmistotuotannossa yleisesti käytettyjä prosesseja. Esimerkiksi vaatimusmäärittelystä sekä validoinnista on tehty todella paljon aikaisempaa tutkimusta ja tästä syystä menetelmät ovat hyvin pitkälle kehitettyjä ja toimiviksi osoitettuja. Vertailevan tutkimuksen peruseriaatteiden hyödyntäminen ja V-mallin soveltaminen vertailuprosessiin luovat selkeän ja helposti seurattavan viitekehyksen sen sijaan, että vertailua tehtäisiin perinteisesti eli ainoastaan kokemus- tai suosittelijapohjaisesti.

Kolmas tutkimuskysymys oli ”*Mitä erityisvaatimuksia joustava valmistusjärjestelmä tuottaa valintaprosessiin?*”. Vastaus tutkimuskysymykseen saatiin tutkimalla joustavien valmistusjärjestelmien ja niiden hyödyntävän teollisuuden historiaa ja ominaispiirteitä. Havaintoina olivat muun muassa teollisuudenalan hidas muutosnopeus sekä haasteet, jotka ilmenevät joustavien valmistusjärjestelmien räätälöinnistä asiakaskohtaisiin tuotantoprosesseihin. Tärkeimmät löydetyt erityisvaatimukset olivat tietoliikenneyhteyksien muodostaminen vanhoihin, eristyneisiin tehtaisiin, joissa järjestelmät yleensä sijaitsevat sekä tarve mallintaa erilaisia järjestelmiä tasolla, joka mahdollistaa IoT-sovelluksien vakioinnin ja järjestelmien vertailun asiakaskohtaisien toimituksien välillä.

Koska kaikkiin tutkimuskysymyksiin löydettiin vastaus sekä ratkaistiin yrityksen tutkimukselle asettamat vaatimukset, saavutettiin tutkimuksella tavoiteltu lopputulos. Siitä huolimatta, että tutkimuksen tuloksena kehitettiin IoT-alustan valintaprosessi joustavan valmistusjärjestelmän digitalisointitarpeisiin, on lopputulos ja tutkimuksessa käytetyt menetelmät hyödynnettävissä myös muunlaisille sovelluskohteille. Esimerkiksi tutkimuksen aikana selvitetty erityisvaatimukset voivat olla mistä tahansa teknisestä järjestelmästä, miltä tahansa teollisuuden alalta. Itse IoT-alustojen suorasta vertailusta löytyy jonkin verran olemassa olevaa tutkimusta, joten työn uutuusarvo muodostuu lähinnä siitä, ettei IoT-alustoihin liittyvää tutkimusta ole käytännössä lainkaan tehty joustavan valmistusjärjestelmän näkökulmasta.

Joustavien valmistusjärjestelmien ollessa hyvin kapea tutkimuksen kohde, sen etuna on tarkkuus, jolla tutkimus voidaan rajata hyvinkin tarkasti. Tämä parantaa tutkimuksen tarkkuutta ja siten luotettavuutta, kun tutkimuksessa päästään yleistä tasoa syvemmälle. IoT-alustoja käsitellessä tilanne on vastakohtainen, sillä puhuttaessa kaupallisista tuotteista ja niiden ominaisuuksista, tiedon luotettavuus voi olla hyvinkin kyseenalainen. Tutkimuksen aineistoon voi päätyä virheellistä tietoa tahallisesti tai tahattomasti, kun aineistoa kerätään tuotetta valmistavan tai myyvän yrityksen markkinointimateriaalista, myyntitilanteesta tai asiantuntijakommenteista. Esimerkkinä aineiston luotettavuus saat-

taa kärsiä, jos tuotteen myyjän edustajan tarkoituksena on luoda mahdollisimman houkutteleva kuva tuotteesta vääristämällä faktoja. Voi myös olla tilanteita, jossa asiantuntijaksi määritelty henkilö ei tiedä vastausta kysyttävään aineistoon liittyvään kysymykseen antaen tahattomasti väärää tietoa. Paras lopputulos tässä tutkimuksessa olisi saavutettu, jos suunnittelut koekäytöt kaikilla alustoilla olisivat toteutuneet. Tässä tapauksessa aineistoksi kerätty tieto olisi ollut kaikissa tapauksissa kokeellisen tutkimuksen avulla hankittua tutkimuksen myötä. Tässäkin tapauksessa testaajalla on suuri rooli tiedon luotettavuuden kannalta, mutta silti tutkimuksen toistettavuus olisi ollut huomattavasti suurempi. Nyt suurin osa vertailun tiedosta on hankittu kartoittavan tutkimuksen menetelmillä erilaisista lähteistä, mikä ei ole optimiratkaisu tiedon luotettavuuden näkökulmasta.

10. YHTEENVETO

Digitalisaatio muuttaa maailmaa, teollisuutta ja yrityksiä eikä sen vaikutuksilta voi välttyä. Yrityksien on toimittava selviytyäkseen murroksesta hyödyntäen asioiden ja teollisen internetin tuomat uudet liiketoimintamahdollisuudet. Yksi konkreettinen tapa on kerätä ja hyödyntää erilaista tietoa yrityksen tuotteista sekä toimintaympäristöstä ja jalostaa siitä hyödyllistä informaatiota, jonka avulla voidaan ohjata toimintaa tai kehittää esimerkiksi uudenlaisia palveluita eri sidosryhmille.

Joustavat valmistusjärjestelmät ovat automatisoineet valmistavaa teollisuutta jo vuosikymmenien ajan, mutta perinteisesti ne ovat olleet eritettynä tehtaiden sisälle ilman kunnollisia verkkoyhteyksiä. Teollisen internetin ideologian mukaisesti FMS-järjestelmätkin tulevat verkottumaan ja yhdistymään muihin systeemeihin ja ekoympäristöihin. Näin järjestelmien tuottama, huomattava tietomäärä saadaan hyödynnettäväksi järjestelmän ulkopuolella.

Tutkimuksen tavoitteena oli selvittää, miten yrityksen tarpeisiin valitaan IoT-alusta ja mitä asioita kannattaa ottaa huomioon prosessin eri vaiheissa. Tutkimuksen lopputuloksena kehitettiin valintaprosessi, jossa yrityksen sekä digitalisaation ja teollisen internetin tavoitteista saadaan vaatimusmäärittelyn avulla teknisiä vaatimuksia. FMS-järjestelmän erityisvaatimuksista sekä teknisistä vaatimuksista muodostetaan vertailukriteereitä, joiden avulla kaupallisen IoT-alustatarjonnan kartoituksen löydöksiä vertaillaan keskenään. Vertailun avulla löydetään paras IoT-alusta vertailujoukosta ja sen soveltuvuus yrityksen käyttöön varmistetaan validoimalla valittua alustaa teknisiin vaatimuksiin ja liiketoiminnan tavoitteisiin varmistuen niiden täyttymisestä. Vertailuprosessin pohjana käytettiin tuotekehityksessä käytettyä V-mallia, missä suuressa osassa itse suorittamisosion jälkeen, kaavion oikeassa haarassa, on tehdyn työn validointi vasemman haaran määrittelyvaiheessa tehtyä työtä vasten. Sovelletun V-mallin kaaviokuva on nähtävissä kuvassa 4.

Määrittely- ja suunnitteluvaiheessa tehdyssä vaatimusmäärittelyssä hyödynnettiin käyttäjätarinoita kerättyjen tarpeiden purkamiseen ongelmiksi ja mahdollisuuksiksi. Näistä muodostettiin yleisiä vaatimuksia, jotka muunnettiin vielä teknisiksi vaatimuksiksi. Teknisten vaatimuksien priorisointiin hyödynnettiin hype-käyriä. Käyrien avulla tutkittiin vaatimuksien toteuttamiseen käytettävien teknologisten ratkaisuiden maturiteettia ja arvioitiin, kannattaako teknologiaa implementoida heti vai vasta myöhemmin, kun teknologia on kehittynyt paremmaksi.

V-mallin kaavion pohjalla olevassa toteutusvaiheessa kartoitettiin ja valittiin vertailun vaihtoehdot kaupallisesta tarjonnasta. Näiden vaihtoehtojen vertailukriteerien määrittelemät tiedot etsittiin eri lähteistä ja kerättiin yhteen vertailutaulukkoon kohtuullisen vaapassa ilmaisumuodossa. Ilmaisumuoto johtui alustojen suurista toiminnallisista eroista

ja erilaisista toimintatavoista, joiden yleistäminen samalle tasolle olisi voinut vääristää tuloksia. Tästä syystä alustojen keskinäiseen vertailuun käytettiin Pughin matriisimenetelmää, minkä avulla alustat pystyttiin pisteyttämään ja laittamaan paremmuusjärjestykseen. Vertailun aikana ymmärrettiin myös omia tarpeita paremmin ja saatiin hyvä ymmärrys IoT-alustojen sen hetkisestä tarjoamasta, niiden eroista ja ominaisuuksista. Alustojen vertailua hankaloitti niiden erilaisuus sekä tiedon puute ja keskeneräisyys, sillä IoT-alustat olivat vasta nousussa olevaa teknologiaa ja kehitysnopeus oli todella suuri. Vain osa vertailussakin mukana olleista alustoista saavutti maturiteetin muiden poistuessa markkinoilta yhtä nopeasti kuin oli tullutkin, yritystalon tai tappion myötä. IoT-alustoille kuitenkin oli markkinat, sillä niiden kokeilu oli todella suosittua teollisuusyrityksen keskuudessa ja käytännössä kaikkia vertailussa mukana olleita alustoja oli kokeiltu suurten ja tunnettujen teollisuusyrityksien toimesta.

V-mallin oikean haaran mukaisesti sen jälkeen, kun IoT-alustat oli saatu soveltavuusjärjestykseen, valittiin kolme lupaavinta alustaa tarkempaan tutkimukseen. Alustoja vertailtiin vielä uudelleen keskenään ja sen jälkeen tehtiin parhaan IoT-alustan valinta. Valittu alusta validoitiin laajasti ohjelmistotuotannossa käytettyjen testitapauksien avulla. Teknisiä vaatimuksia ja liiketoiminnan tavoitteita vasten kirjoitettiin testitapauksia, joissa vaatimuksilla ja tavoitteilla on jokin oletettu lopputulos. Rinnalle kirjoitettiin realisoitunut lopputulos ja arvioitiin, täyttyykö määritelty vaatimus tai tavoite.

Validoinnin tuloksena saatiin, että valittu alusta täyttää yhtä teknistä vaatimusta lukuun ottamatta kaikki määritellyt vaatimukset ja tavoitteet. Kuitenkin jo vertailuvaiheessa oli päätetty, että kyseisen tietoliikenneyhteyksien toteuttamiseen liittyvät asiat erotetaan IoT-alustan hankinnasta, kun nähtiin, ettei yhdelläkään alustatoimittajalla ollut tarjota integroitua laitteistoratkaisua, jollaista aluksi etsittiin. Näin ollen valittu IoT-alusta oli validoinnin mukaan oikea yrityksen tarpeisiin ja osoitettiin, että kehitetty valinta- ja vertailuprosessi on toimiva työkalu IoT-alustan hankintaan ja näin ollen yrityksen vaatimukset työlle täyttyivät.

Koska tutkimuksessa saavutettiin tavoiteltu lopputulos ja työn aikana löydettiin vastaukset tutkimuskysymyksiin, jotka ohjasivat etsimään parhaita ratkaisuita yleisesti ohjelmistotuotannossa käytetyistä menetelmistä esimerkiksi vaatimusmäärittelyn ja validoinnin suhteen, voidaan todeta tutkimuksen onnistuneen ja olleen hyödyllinen. Hyödyllisyys korostuu etenkin siinä, että työllä on suuri uutuusarvo joustavien valmistusjärjestelmien digitalisoinnista, josta ei ole tehty paljoa aikaisempaa tutkimusta. Kuitenkin, tehtyä tutkimusta ja valintaprosessia on mahdollista soveltaa mihin tahansa muuhun korkean teknologian laitteistoon, kunhan erityisvaatimukset selvitetään kyseisen laitteiston näkökulmasta.

LÄHTEET

Baumeister, H. Knapp, A., Wirsing, M. (2004). Property-driven development. Proceedings of the Second International Conference on Software Engineering and Formal Methods. SEFM 2004, s. 96-102.

Buede, D. & Miller, W. (2016). The Engineering Design of Systems: Models and Methods (Third Edition). John Wiley & Sons, 583 s.

Burge, S. (2009). The Systems Engineering Tool Box - Pugh Matrix (PM). Saatavissa (viitattu 22.11.2018): <https://www.burgehugheswalsh.co.uk/uploaded/1/documents/pugh-matrix-v1.1.pdf>, 15 s.

Das V-Modell (n.d.) Das V-Modell® - V-Model 97. Verkkosivu. Saatavissa (viitattu 27.7.2018): <https://v-modell.iabg.de/index.php/vm97-uebersicht>

Evans, P.C. & Annunziata, M. (2012). Industrial Internet: Pushing the boundaries of minds and machines. General Electric, 37 s.

Gartner. (2015). Saatavilla (24.8.2017): <http://blogs.flexerasoftware.com/ecm/2015/07/licensing-and-entitlement-management-cited-as-key-trigger-in-iiot-in-recent-report.html>

Gartner. (n.d.). Saatavilla (24.8.2017): <http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp>

Hughes, A. (2017). Smart Manufacturing: Smart Companies Have Made Smart Manufacturing the Center of the Enterprise. LNS research, 30 s.

Hwaiyu, G. (2015). Manufacturing Engineering Handbook, Second Edition. McGraw-Hill Education, 624 s.

Kaner, C. (2003). What Is a Good Test Case? Florida Institute of Technology Department of Computer Sciences, 16 s.

Kokeellinen tutkimus. (2015). Verkkosivu. Saatavissa (viitattu 10.12.2018): <https://koppa.jyu.fi/avoimet/hum/menetelmapolkuja/menetelmapolku/tutkimusstrategiat/kokeellinen-tutkimus>

Kossiakoff, A., Sweet, W., Seymour, S., Biemer, S. (2011). Systems Engineering Principles and Practice (2nd Edition), John Wiley & Sons, 560 s.

Kujala, S., Kauppinen, M., Rekola, S. (2001). Bridging the Gap between User Needs and User Requirements. Teoksessa Avouris, N. & Fakotakis, N. (toim), Panhellenic Conference with International Participation in Human-Computer Interaction (PC-HCI2001). Software Usability Network, University of Patras, Patras, Kreikka, s. 45-50.

Lamsweerde, A. & Letier, E. (2000). Handling obstacles in goal-oriented requirements engineering. IEEE Transactions on Software Engineering. Volume 26, Issue 10, October 2000, s. 978-1005.

Lamsweerde, A. (2001). Goal-Oriented Requirements Engineering: A Guided Tour. Proceedings of the IEEE International Conference on Requirements Engineering, s. 249-262.

Mahajan, A. & Dixit, A. (2013). Various approaches of requirement engineering and comparisons among these approaches. International Journal of Advanced Research in Computer Science. Vol. 4, Issue 4, March-April 2013, s. 184-186.

Martinsuo, M. & Kärri, T. (2017). Teollinen internet uudistaa palveluliiketoimintaa ja kunnossapitoa. Kunnossapitoyhdistys Promaint ry, 238 s.

Mazhelis O., Luoma E., Warma H. (2012). Defining an Internet-of-Things Ecosystem. Teoksessa: Andreev S., Balandin S., Koucheryavy Y. (toim) Internet of Things, Smart Spaces, and Next Generation Networking. ruSMART 2012, NEW2AN 2012. Lecture Notes in Computer Science, vol 7469. Springer, Berlin, Heidelberg. s. 1-14.

Mineraud, J., Mazhelis, O., Su, X., Tarkoma, S. (2016). A gap analysis of Internet-of-Things platforms. Computer Communications Volumes 89–90, September 2016, s. 5-16.

Osborne, L., Brummond, J., Hart, R., Zarean, M. & Conger, S. (2005). Clarus: Concept of Operations. US Department of Transportation, Federal Highway Administration FHWA-JPO-05-072.

Perry, M. (2016). Evaluating and choosing IoT platform. O'Reilly Media, 20 s.

Porter, M. & Heppelmann, J. (2014). How Smart, Connected Products are Transforming Competition. Harvard Business Review, November 2014, s. 64-88.

Routio, P. (2007). Mallien käyttö tutkimushankkeessa. Verkkosivu. Saatavissa (viitattu 10.12.2018): <http://www2.uiah.fi/projekti/metodi/077.htm>

Routio, P. (2007). Vertailu. Verkkosivu. Saatavissa (viitattu 10.12.2018): <http://www2.uiah.fi/projects/metodi/072.htm>

Sage, A. & Armstrong, J. (2000). Introduction to Systems Engineering. John Wiley & Sons, 568 s.

Sage, A. & Rouse, W. (2009). Handbook of Systems Engineering and Management (2nd Edition). John Wiley & Sons, 1504 s.

Vilkkä, H. (2007). Tutki ja mittaa: Määrällisen tutkimuksen perusteet. Tammi, 189 s.

Woldeamlak, S., Diabat, A., Svetinovic, D. (2016). Goal-Oriented Requirements Engineering for Research-Intensive Complex Systems: A Case Study: GOAL-ORIENTED REQUIREMENTS ENGINEERING FOR RICS. Systems Engineering. Volume 19, Issue 4, s. 322-333.

World Economic Forum. (2015). Industrial Internet of Things: Unleashing the Potential of Connected Products and Services. Saatavissa (viitattu 14.11.2018):
http://www3.weforum.org/docs/WEFUSA_IndustrialInternet_Report2015.pdf

Young, R. (2004). Requirements Engineering Handbook. Artech House, 275 s.

LIITE C: PUGHIN VERTAILUMATRIISI IOT-ALUSTOISTA

		ThingWorx	IoT-Ticket	Azure	Bluemix	2lemetry	AWS	ELK Stack	Industrial Internet Suite	I&I Connect	Cosmos	Cumulocity
Ecosystem	Provider	S	S	S	S	S	S	S	S	S	S	S
	Distributors (fi)	S	S	S	S	S	S	S	S	S	S	S
	Partners (fi)	S	S	+	+	-	+	-	-	+	-	+
	Ecosystem size	S	-	+	-	-	+	-	-	-	-	-
	Marketplace	S	-	-	-	S	-	-	-	-	-	-
	References (fi)	S	S	S	-	-	-	-	+	S	-	-
	Other notable references	S	-	-	-	S	-	-	-	-	-	S
	End-to-end solution available	S	S	-	S	-	-	-	S	-	S	S
Data collection HW	HW available from distributor	S	+	-	-	-	-	-	S	-	S	S
	WAN options	S	S	-	-	-	-	-	-	-	-	-
	Fieldbus connections	S	+	S	S	S	S	S	+	S	+	S
	Other connections	S	+	-	-	-	-	-	S	-	S	-
	Other sensors	S	+	-	-	-	-	-	-	-	S	+
	Device management	S	S	S	-	-	-	-	-	-	-	S
	Remote connections	S	S	-	-	-	-	-	-	-	-	-
	Look-and-feel	S	S	-	-	-	-	-	S	-	+	-
DIN installation	S	S	-	-	-	-	-	S	-	S	-	
Data collection SW	SW available	S	-	S	-	-	-	-	-	-	-	-
	OPC supported	S	S	-	-	-	-	-	-	-	-	-
	Messaging protocols	S	-	-	-	-	-	-	-	-	-	-
	Connections to external services	S	-	-	-	-	-	-	-	-	-	-
	Data storage location	S	-	-	-	-	-	S	S	-	S	-
	Data security	S	S	S	-	-	-	S	-	-	S	S
Handling and analyzing data	Data modelling	S	-	-	-	-	-	-	-	-	-	-
	Automatic data handling	S	+	S	S	-	S	-	-	-	-	-
	Analytic tools	S	+	+	+	+	+	-	S	S	-	-
	Machine learning	S	S	+	+	+	+	S	S	-	-	-
	Utilizing analyzed data	S	-	S	S	S	S	-	S	S	-	-
	Data migration	S	-	-	-	S	-	-	-	-	-	S
Visualization	Creating views/dashboards	S	+	S	-	-	-	-	-	-	-	-
	Mobile devices	S	+	+	-	-	-	-	S	S	+	+
	Static reports	S	+	S	S	S	S	S	+	+	+	S
Service creation speed and models	Installation and configuration	S	S	-	-	S	-	-	-	-	-	-
	Need of coding	S	+	-	S	+	-	-	-	-	S	S
	Independent usage	S	S	-	-	S	-	-	-	-	S	S
	Data model management	S	+	-	-	-	-	-	-	-	-	+
Data security and rights management	Multitenancy	S	S	-	-	-	-	-	-	S	-	-
	User and rights management	S	S	S	-	-	-	-	-	-	-	-
	Access over all customer data	S	-	-	-	-	-	-	-	-	-	-
Scalability	Handling extending device count	S	+	S	S	+	S	-	S	S	+	+
	Cloud or local installation	S	S	-	-	-	-	-	-	-	S	-
	General scalability	S	S	+	+	+	+	-	S	S	-	S
Flexibility	Multiple tasks with one system	S	S	S	S	-	S	-	S	S	-	-
	Implementation of current concepts	S	-	S	S	-	S	-	-	-	-	S
	Effort for making changes	S	-	-	-	S	-	-	-	-	-	S
	General flexibility	S	-	S	S	-	S	-	-	-	-	-
Pricing	Pricing model	S	+	+	+	-	+	-	-	-	+	+
	One time costs	S	-	+	+	+	+	-	S	S	S	S
	Upkeep costs	S	S	S	S	+	S	-	-	-	S	S
	Costs/instance	S	-	-	-	-	-	-	-	-	-	-
	Costs/customer	S	+	-	-	-	-	-	-	S	-	-
	Costs/system	S	+	+	+	+	+	-	-	-	-	-
	Costs/user	S	+	+	+	S	+	+	S	S	S	S
Totals	Total +	0	16	10	8	8	9	1	3	2	6	6
	Total -	0	16	26	32	33	33	45	33	37	32	28
	Grand total	0	0	-16	-24	-25	-24	-44	-30	-35	-26	-22
	Total + w/o HW	0	12	10	8	8	9	1	2	2	4	5
	Total - w/o HW	0	16	19	24	25	25	37	29	29	29	23
Grand total w/o HW	0	-4	-9	-16	-17	-16	-36	-27	-27	-25	-18	