

Janne Paasimaa

**DYNAAMISEN KESKITETYN LABORATORION RAPORTOINTI-  
JA MONITOROINTIJÄRJESTELMÄ**

# **DYNAAMISEN KESKITETYN LABORATORION RAPORTOINTI- JA MONITOROINTIJÄRJESTELMÄ**

Janne Paasimaa  
Opinnäytetyö  
Kevät 2018  
Tietotekniikan tutkinto-ohjelma  
Oulun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Tietotekniikan tutkinto-ohjelma, laite- ja tuotesuunnittelun suuntautumisvaihtoehto

---

Tekijä: Janne Paasimaa

Opinnäytetyön nimi: Dynaamisen keskitetyn laboratorion raportointi- ja monitorointijärjestelmä

Työn ohjaajat: Kari Jyrkkä, Antti Karppinen

Työn valmistumislukukausi ja -vuosi: Kevät, 2018

Sivumäärä: 37

---

Työn aiheena oli raportointi- ja monitorointijärjestelmän suunnittelu dynaamiseen keskitettyyn laboratorioon. Tavoitteena oli laatia vaatimusmäärittely dynaamisen LaaS-poolin (Laboratory as A Service) resurssien ja testaustopologioiden raportointi- ja monitorointitarpeista, määrittellä tavoitearkkitehtuuri ja toteuttaa järjestelmästä prototyypisovellus. Prototyypisovelluksessa oli tavoitteena esittää muutamien kuvaajaesimerkin avulla idean toimivuus. Työn toimeksiantajana toimi Nokia Oyj.

Työ aloitettiin tutustumalla olemassa oleviin vastaavassa käytössä oleviin työkaluihin ja perehdyttiin laboratorioympäristön teknisiin vaatimuksiin, minkä perusteella päädyttiin työssä esitettyyn järjestelmäarkkitehtuuriin. Työn keskeinen tavoite oli vaatimusmäärittely raportointi- ja monitorointijärjestelmälle. Toimintoanalyysin avulla määriteltiin tieto- ja raportointitarpeet, josta lopulta muodostui työkalun vaatimusmäärittely. Dokumentin tiedoilla oli tarkoitus määrittellä työkalun vaatimukset järjestelmän toteuttavalle yritykselle. Oleellimmat kuvaajatarpeet määriteltiin dokumentissa laboratorioympäristöön liittyvien funktioiden ja haastatteluiden avulla. Lisäksi dokumentissa oli oleellista määrittellä resurssitasoja ja niiden suodatustapoja, jotta voitiin tehdä suunnitelma tietokantarakenteesta.

Lopputuloksena toimintoanalyysin perusteella voitiin esittää tarkat suunnitelmat laboratorion hallintatyökaluun tarvittujen ominaisuuksien osalta, jonka tekninen toteutus jäi suunnitellusti järjestelmän toteuttaneelle yritykselle. Työn aikana selvitettiin perusteellisesti järjestelmässä tarvittavat kuvaajatarpeet ja niihin liittyvät tietotarpeet, jotta operatiivisesta tietokannasta haettu data olisi sisältänyt kaiken tarvittavan. Integroitavan ominaisuuden toteutuksen viivästymisen ja ajanpuutteen johdosta prototyypisovellusta ei voitu tehdä. Kun datahaku on integroitu järjestelmään, kuvaajia voidaan alkaa tehdä Kibana-työkalun avulla.

---

Asiasanat: LaaS, Kibana, monitorointi, tilastotieteen visualisointi, testausjärjestelmä

# ABSTRACT

Oulu University of Applied Sciences  
Bachelor's degree programme, Information Technology

---

Author: Janne Paasimaa

Title of thesis: Reporting and monitoring system of a dynamic consolidated laboratory

Supervisors: Kari Jyrkkä, Antti Karppinen

Term and year when the thesis was submitted: Spring 2018

Pages: 37

---

The subject of this thesis was to design reporting and monitoring system to dynamic consolidated laboratory. The target was to form a requirements specification document regarding the needs of reporting and monitoring system of the dynamic LaaS (Laboratory as A Service) pool's resources and the topologies. Additionally, there was a need for planning the system architecture and later on create a proof of concept by implementing a couple of planned visualized charts. The work was commissioned by Nokia Oyj.

The work was started by studying the current reporting and monitoring tools. Also, technical requirements of the current laboratory were in important role when the necessities of the advanced system were determined. Eventually the system architecture was planned as a result. Function analysis documentation aimed to determine the reporting needs. The company who developed the current reservation and management system had a need for the specifications about the structure of the planned tool at Nokia and the document aimed to fulfill that need. Its purpose was to determine the resource levels and their filtration practices. Moreover, those determinations were used to design the database structure.

The function analysis document was used as a requirement specification for the reporting and monitoring tool. The company who made the LaaS service was cooperating regarding the technical possibilities. Graph needs and the required data for them had to be determined accurately in that document because the solution needed to include all the important data to be able to use the statistics tool powerfully. Unfortunately, the delay in integrating the feature and lack of time to create own simplified solution were the major reasons to finish the work without the actual working Kibana tool. However, the tool will be utilized when the integrated feature is implemented.

---

Keywords: LaaS, Kibana, monitoring, visualized statistics, testing system

## **ALKULAUSE**

Haluan kiittää Nokia Oyj:tä opinnäytetyön mahdollistamisesta ja erityisesti Nokian projektipäällikköä Antti Karppista uusien näkökulmien tuomisesta työhön. Lisäksi haluan kiittää työkavereita ja perhettä kannustamisesta opinnäytetyön teossa.

Opinnäytetyön ohjaajana toimi OAMK:n puolesta lehtori Kari Jyrkkä. Hänelle kiitokset opinnäytetyön ammattitaitoisesta ohjauksesta.

Oulussa 17.5.2018

Janne Paasimaa

# SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
1 JOHDANTO	7
2 ETÄKÄYTTÖINEN LABORATORIO JA STATISTIIKKATARPEET	9
2.1 Velocity etäkäytettävän laboratorion ytimenä	9
2.2 Laitteisiin liittyvien tietojen esittäminen tehokkaammin	11
2.3 Nykyiset ratkaisut	12
3 TYÖKALUTARPEIDEN MÄÄRITTELY TOIMINTOJEN KAUTTA	15
3.1 Vaatimusmäärittely	15
3.2 Tarkempien kuvaajatarpeiden selvittäminen haastattelujen avulla	16
3.3 Toimintoanalyysin lähtökohdat	16
3.4 Toimintoanalyysissa määritellyt resurssiluokat	17
3.5 Toimintoanalyysissa määritellyt funktiot	19
3.5.1 Pitkän aikavälin suunnittelufunktio	20
3.5.2 Lyhyen aikavälin suunnittelufunktio	20
3.5.3 Laboratorion ylläpitofunktio	21
3.5.4 Testisuunnittelufunktio	21
3.5.5 Testifunktio	22
3.5.6 Tukifunktio	23
4 TYÖKALUT JA TEKNIIKAT	24
4.1 Järjestelmäarkkitehtuurikuvaus	24
4.2 Dataprosessointi	25
4.3 Tietovarastointi	26
4.4 Statiikan esittäminen visuaalisesti	27
5 TYÖN TOTEUTUS	29
6 JATKOKEHITYSMAHDOLLISUUDET	32
7 YHTEENVETO	35
LÄHTEET	37

# 1 JOHDANTO

Tuotekehitys on lähtökohtaisesti aina todella kallista, joten kuluja on jollain tapaa minimoitava voittojen maksimoimiseksi. Tukiasemien kehitystyössä testauksen suurimmat menoerät syntyvät tukiasemista, mittalaitteiden hankinnoista ja kuormalaitteista eli dataa ja puheluita tuottavista laitteista. Uudet taajuusalueet ja tekniikat vaativat uusia ominaisuuksia erityisesti testauksessa käytettävältä laitteistolta, ja tekniikan ollessa uutta myös hinnat nousevat huomattavan korkeiksi. Kun manuaalisen testauksen suorittaa työntekijä rajallisella työajalla, laitteiston hyödyntäminen ei nouse lähellekään huippuarvoa. Erityisesti mittalaitteet ovat erittäin kalliita, joten niitä pyritään jakamaan mahdollisimman monen käyttäjän kanssa käyttöasteen jäädessä silti kauas optimaalisesta. Lisäksi testilinjan rakennustyöt voivat viedä kallista testausaikaa, jonka aikana laitteisto jää täysin hyödyntämättä.

Etäkäyttöisyys on ollut jo pidemmän aikaa tavoiteltava asia monella alalla. Nokian tukiasemien testaus on pääasiassa testipaikan vieressä suoritettavaa manuaalista testausta, vaikka monessa tapauksessa testiympäristö olisi ollut helposti valjastettavissa etäkäyttöiseksi. Etäkäyttöisyydestä saadaan aidosti toimiva ratkaisu vasta laitteiston ollessa keskitettynä yhteen tilaan, jolloin kytkentämuutokset eivät rajoitu vain yhteen vaihtoehtoon. Etäkäyttöön vievät testitilanteet pitää suunnitella tapauskohtaisesti, koska monesti testiympäristö vaatii tarkkailua ja puuttamista tilanteeseen fyysisesti. Näin ollen prototyypin liian varhainen versio tai muu tekninen ominaisuus voi olla rajoitteena testitilanteen viemiseksi etäkäyttöön.

Säästöjä ja työn tehostamista on suotavaa pyrkiä parantamaan etäkäytön lisäämisellä, koska mm. mittalaitteet ovat suuri sijoitus ja niiden käyttöaste on manuaalisissa testiympäristöissä todella alhainen. Käyttöastetta nostaa huomattavasti laitteiston tarjoaminen globaalisti saataville, jolla voi olla suora positiivinen vaikutus investointien määrään. Operatiivisista kuluista saadaan säästettyä vähentämällä toistuvaa testilinjojen fyysistä rakentamista, jolloin testihenkilöt voivat keskittyä testaamiseen, mikä nopeuttaa koko testiprosessia. Samalla jäähdytys- ja vuokrakuluissa säästetään, kun laboratorio voidaan keskittää yhteen fyysiseen tilaan. Ihmisten tekemien virheiden määrä on suurempi manuaalisten kytkentöjen

kanssa, jolloin automatisoitua testilinjan rakentamista voidaan pitää toimintavarmempana. (1, s. 29.)

Etäkäytettävä ympäristö on pääpiirteittäin jo toimiva ja valmis päivittäiseen käyttöön. Ympäristö vaatii kuitenkin jatkokehitystä monilta osin, jotta kokonaisuus täyttäisi kaikki vaaditut tarpeet ja toimisi riittävän luotettavasti myös silloin, kun käyttäjämäärä alkaa kasvaa huomattavasti enemmän. Laboratorion varaus- ja hallintajärjestelmä organisoii laitteistoa varausjärjestelmänsä avulla ja ylläpitää inventaariota. Käyttöasteen muutoksia ja muuta dataa on kuitenkin nykyisillä mittareilla hyvin vaikeaa esittää varsinkin, kun kyseessä on mikään vähänkään erikoisempi rajaustarve. Nykyinen raportointinäköymä näyttää vain yksittäisten resurssien ja resurssipohjien perusteella ryhmiteltyjen resurssien käyttöastetta ilman visuaalisesti koottua näköymää puhumattakaan muista rajauksista.

Kehitettävän raportointi- ja monitorointityökalun yhtenä päätavoitteena on tarjota käyttäjäystävällistä statistiikkaa, koska tämänhetkinen toteutus ei anna riittävän helposti ja selkeästi haluttua tietoa. Nykytoteutuksella järjestelmä näyttää varaus-tietoihin pohjautuvaa käyttöastedataa yksittäisten tai ryhmiteltyjen resurssien tasolla. Nämä eri aikavälillä saatavat prosenttitiedot eivät ole helposti hyödynnettävissä formaatissa, eikä järjestelmä tarjoa juurikaan apuvälineitä datan suodatukseen. Jo olemassa olevien ratkaisujen hyödyntäminen ei toimi tässä käyttötarkoituksessa, koska siirrytään staattisista testilinjoista datakeskuksissa sijaitseviin dynaamisiin ympäristöihin. Paranneltu raportointi- ja monitorointijärjestelmä on tärkeä vaihe ennen palvelun ottamista päivittäiseen käyttöön, koska inventaarion esittäminen informatiivisemmin on oleellista, jotta kapasiteetin hallinta olisi sujuvaa käyttäjä- ja laitemäärän kasvaessa.

Työn yhtenä tavoitteena on toteuttaa vaatimusmäärittely raportointi- ja monitorointiratkaisulle, jonka pohjalta toteutetaan Proof of Concept -tyylisesti osa toiminnallisuudesta. Vaatimusmäärittelyn yhtenä tärkeänä osana on tuottaa tavoitetearkkitehtuuri monitorointijärjestelmälle, josta hahmottuu järjestelmän kokonaiskuva. Prototyypin toteutus sisältää raportointitietokannan määrittelyn ja luonnin sekä datan visualisointia työkaluun.

## 2 ETÄKÄYTTÖINEN LABORATORIO JA STATISTIKKATARPEET

Etäkäyttöön siirryttäessä laitekanta keskitetään yhteen palvelinhuoneen tyyliiseen tilaan, jolloin melu- ja lämpöhaitat eliminoiduvat. Perinteisesti testaajan vastuulla on ollut huolehtia testilaitteiston kytkennöistä, mutta keskitetyssä laboratoriossa tarvittavat kytkennät voidaan määritellä virtuaalisesti, jonka jälkeen varaus- ja hallintajärjestelmä suorittaa muutokset kytkimille. Fyysisille kytkentämuutoksille on tällaisessa laboratoriossa tarvetta vain tilanteissa, joissa joidenkin laiteresursien välillä ei olekaan suoraa kytkentää tai mahdollisuutta ohjata sitä kytkimen läpi. Ympäristön hallinta ja ohjaus tapahtuvat pääasiassa virtuaalikoneiden avulla, joten muutos on kokonaisuudessaan huomattava perinteiseen manuaaliseen testaustapaan verrattuna.

Laitteiston siirtyessä keskitettyyn laboratoriotilaan kytkentätapa muuttuu. Kaapelointi ohjataan erilaisiin kytkimiin ja virranjakoyksiköihin, jotta koko infrastruktuurin hallinta ja käyttö on mahdollista suorittaa etänä. Muutokset kytkentöihin onnistuvat muuttamalla virtuaalista testilinjaa, joka käyttöönottoaiheessa ohjaa kytkinten ja niitä varten tehtyjen ajureiden avulla oikeat reitit kaikille tarvittaville laitteille. Testaukseen käytettyä aikaa ja virheiden määrää saadaan vähennettyä huomattavasti.

Perinteinen tukiaseman testausympäristö voi yksinkertaisimmillaan koostua radiomoduulista, systeemimoduulista, tietokoneesta, RF-kytkimestä ja mittalaitteesta. Kun siirrytään etäkäytettävään ympäristöön, verkkoliikennettä hallinnoimaan tarvitaan IP-kytkin ja laitteiston virransyötön ja -katkaisun suorittamista varten virranjakoyksikkö. Laboratorion inventaariota pitää hallinnoida, koska käyttäjä ei voi perinteiseen tapaan ottaa haluttua laitteistoa haltuun menemällä laboratorioon. Lisäksi kytkimien monimutkaistaessa kokonaisuutta tarvitaan varmatoiminen varausjärjestelmä, jotta laitteiston käyttö olisi hallittavissa useamman käyttäjän välillä.

### 2.1 Velocity etäkäytettävän laboratorion ytimenä

Spirentin kehittämä kaupallinen varaus- ja hallintaohjelmisto nimeltään Velocity toteuttaa ympäristön hallittavuudesta suuren osan. Laboratory as a Service

(LaaS) tarkoittaa laboratorion käyttämistä mistä tahansa hallitusti jonkin palvelun avulla, joka on tässä tapauksessa Velocity. Jotta palvelu täyttäisi vaatimukset testien suorittamiseksi dynaamisessa testiympäristössä globaalisti monen käyttäjän välillä, Nokia on tehnyt yhteistyötä Spirentin kanssa saadakseen tarvitsemansa ominaisuudet Velocityyn. Velocity on suunniteltu riippumattomaksi käyttöjärjestelmästä. Sitä voidaan käyttää verkkoselaimella ja etäyhteyden muodostaminen esimerkiksi virtuaalikoneelle onnistuu varauksen yhteydessä suoraan selainikkunassa, joten se sopii hyvin laboratorioympäristöön.

Yhteistyön tuloksena käytettävissä on keskitetty laboratorio, jossa topologia eli laitteista ja niiden välillä olevista kytkennöistä koostuva testilinja voidaan yksinkertaisimmillaan ottaa käyttöön valitsemalla esimerkiksi kuvassa 1 nähtävä topologia. Sen jälkeen asetetaan sopivan pituinen varausaika, jolloin varaus käynnistyy joko heti tai käyttäjän määrittämän ajan kuluttua. VLAN-pilvellä kuvataan IP-kytkimen kautta reitittyvää liikennettä tässä tapauksessa virtuaalikoneen ja systeemimoduulin välillä. Jokaiselle virtuaalikoneelle on määritetty yleensä kaksi verkkoporttia, joista toiseen on asetettu staattinen IP-osoite pelkästään suorille kytkennöille radiomoduuliin. Toiseen porttiin on määritetty staattinen IP-osoite systeemimoduuleita varten. Systeemi- ja radiomoduulin välinen yhteys muodostetaan niin, että molempiin laitteisiin on kytketty SFP-moduuli, josta lähtee kuitukaapeli laitteet toisiinsa yhdistävälle optiselle kuitukytkimelle. Radiomoduulin ja mittalaitteen välillä on RF-signaalia reitittävä matriisikytkin, jolla voidaan vaihtaa varausta keskeyttämättä esimerkiksi mitattavaa liitäntää radiomoduulista tai mittalaitteesta sisääntulo ulostuloksi.



*KUVA 1. Topologia Velocityssa, jolla mitataan saapuvaa tai lähtevää dataa.*

Suurena muutoksena käyttäjille näkyy tiimikohtaisten laitteiden siirtyminen yhteiseen käyttöön, jolloin varataan uuden mallin mukaisesti inventaariosta käyttäjän tarvitsema laitteisto halutuksi ajaksi. Ajattelumalli on aivan erilainen perinteiseen tilanteeseen verrattuna, mutta paikoittain riittämättömien laiteresurssien takia ympäristöä ei voida toistaiseksi valjastaa isomman käyttäjäryhmän käytettäväksi.

Käyttäjiltä tulee nykytilanteessa satunnaisesti pyyntöjä heidän omien laitteidensa saamisesta takaisin testeihin, joka ei tue täysin ympäristön ideaalista käyttötappaa. Keskustelujen ja yleisen asenteen pohjalta tiedostetaan jo alkuvaiheessa se, ettei täysin avoimeen käyttöön tulla pääsemään ainakaan toistaiseksi, koska tilanteeseen vaikuttaa tiimikohtaiset budjetit ja testit täytyy saada suoritettua jo olemassa olevilla hankinnoilla.

Vaihtoehtona voi käytön kasvaessa olla esimerkiksi Velocityn mahdollistama resurssien lainaaminen tiimien välillä, jolloin käytössä oleva laina-aika kuluisi varauksen yhteydessä. Ohjelmistopäivitysten automatisointia voidaan pitää korkealla prioriteetilla siinä vaiheessa, jos laitteiston lainaamista tiimien välillä aletaan ottaa käyttöön. Tiettyjen laitteiden yhteensopivuutta voidaan parantaa esimerkiksi lisäämällä tuki automaattisille ohjelmistopäivityksille, jolloin laite olisi varausvaiheessa alustettu testin vaatimien olosuhteiden mukaisesti ja toiminnon avulla voisi parantaa käyttöastetta vähentäen samalla prototyyppien tarvetta.

## **2.2 Laitteisiin liittyvien tietojen esittäminen tehokkaammin**

Ylläpitotehtävillä on suuri vaikutus ympäristön toimivuuteen, koska se vastaa laboratoriotilojen kapasiteetista mm. lisäämällä tai vähentämällä tarvittaessa tietynlaisten laitteiden poolia ja ratkoo Velocityyn liittyviä ongelmatilanteita. Nykyisessä mittakaavassa ei ole juurikaan ilmennyt suuria ongelmia vapaiden resurssien suhteen, mutta ongelma on jo paikoittain havaittavissa ja tulee esille yleensä vasta käyttäjien ottaessa yhteyttä esimerkiksi mittalaitteiden loppuessa kesken. Nykyisillä Velocityssa olevilla ominaisuuksilla on haastavaa nähdä monenlaiset ongelmatilanteet ennen niiden eskaloitumista ja sitä voisi pitää ongelmana, jos tällaisessa tilassa oleva ympäristö pitäisi ottaa käyttöön paljon laajemmalle käyttäjämäärälle.

Monitorointi- ja raportointityökalun kehittämistarvetta puoltavat myös hyvin rajoittuneet mahdollisuudet tarkastella resurssien käyttöastetta historiallisella ja lähes reaaliaikaisella tasolla. Käyttöasteen seurannasta nähtävät prosenttiarvot kertovat yksinkertaisella tavalla riittävästi tietoa, mutta oikean laitteen tai ryhmän tietojen haku ei operatiivisen tietokannan takia toimi optimaalisella nopeudella. Lisäksi Velocityn tapa koostaa dataa on hyvin rajoittunut, koska organisaatiotasolla

olisi tarvetta nähdä ja hyödyntää paljon kattavampia statistiikkanäkymiä. Käytännössä kaikki tieto, mitä ikinä halutaankaan raportoida ja monitoroida, on jo saatavilla Velocityssa, mutta käyttäjän pitää tietää etsimänsä tiedon sijainti ja osata myös tulkita tietyllä tavalla sitä. Saatavilla oleva data ei myöskään ole monelta osin lähellekään sitä tilannetta, että sitä voitaisiin hyödyntää järkevästi ja yksinkertaisesti.

### **2.3 Nykyiset ratkaisut**

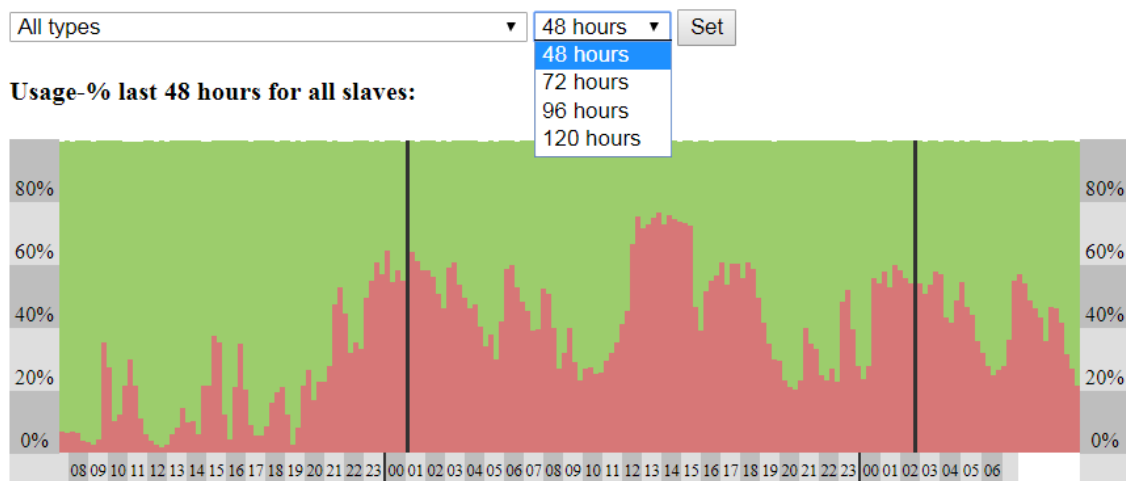
Yrityksen sisältä löytyy useampia toteutuksia niin reaaliaikaisesta kuin historiallisestakin statistiikan tarkastelusta, mutta yksikään työkalu ei sovellu etäkäyttöisen dynaamisen ympäristön kanssa käytettäväksi, koska erot testilinjojen ja käyttötapojen suhteen ovat huomattavia. Suurimpana erona muihin toteutuksiin on testilinjojen dynaamisuus, jota ei ole aiemmin tarvinnut ajatella samassa mittakaavassa. Perinteisessä testitilanteessa suurin osa kytkennöistä on toteutettu ilman ylimääräistä laitteistoa välissä. Etäkäyttöisissä testilinjoissa dynaamisuus toteutetaan lisäämällä kytkin verkko- ja kuitukaapeleita varten, jolloin mikä tahansa osa testilinjasta on vaihdettavissa toiseen.

Velocityssa on mahdollisuutena luoda joko loogisista tai abstrakteista resursseista koostuva topologia, mutta abstraktius aiheuttaa paljon ylimääräistä huomiointavaa raportoinnin ja monitoroinnin suhteen. Varausvaiheessa voidaan määrittellä käyttöönotettavat laitteet pelkkiä parametreja käyttäen, jolloin voidaan varata esimerkiksi mikä tahansa vapaana oleva radiomoduuli tai vaikkapa sellainen tuotevariantti, jossa on kuusi antenniliitäntää. Abstraktissa topologiassa käyttöönotettava laitteisto siis konkretisoituu vasta varausvaiheessa, jolloin Velocityn varausjärjestelmä valitsee laitteet parametrien mukaan. Lisäksi tilanteesta tekee monimutkaisemman mahdollisuus muuttaa testilinjaa kesken varauksen, jonka on todella oleellista olla havaittavissa lopullisessa toteutuksessa. Nykyisistä raportointi- ja monitorointiratkaisuista voidaan kuitenkin havaita monia seikkoja, jotka täytyy huomioida suunnitellessa Velocitysta saatavan statistiikan esittämistapoja.

Työkalu A:n yleisnäkyvä ilmoittaa käyttäjälle testilinjojen tilan värikoodattuna kolmen vaihtoehdon mukaan, jotka ovat saatavilla, varattu ja poiskytketty. Tämän

lisäksi testilinjat on jaettu kategorioihin mm. systeemimoduulin version ja robottiajon mukaan. Testilinjän nimeä klikkaamalla päästään tarkastelemaan varaustilaa, jossa nähdään mm. ohjelmistoversio, varaajan nimi ja varauksen kesto. Työkalussa on mahdollisuus tarkastella статистиikkaa kuvan 2 mukaisesti, missä nähdään mm. käyttöasteita tuntitasolla joko koko työkaluun lisätyn laitekannan mukaan tai vaihtoehtoisesti aiemmin mainittuun kategorisointiin pohjautuen. Prosentuaalinen käyttöasteen aikaväli voidaan valita 48, 72, 96 tai 120 tunnin välillä ja tieto pylväsdiagrammissa nähdään tunnikohtaisesti. Varausaika, varausmäärä, erillisten käyttäjien määrä ja jonotusaika näytetään muiden välilehtien alla joko tunti-, päivä tai kuukausikohtaisesti.

[ back ] [ overview ] [ hourly ] [ daily ] [ monthly ] [ top 10 ] [ usage-% ]



KUVA 2. Näkymä käyttöasteelle työkalussa A.

Toteutuksesta jää kokonaisuudessaan ristiriitainen olo dynaamisen keskitetyn laboratorion kannalta, koska näkymä on hyvin pelkistetty, mutta tuo samalla esille oleellimmän tiedon. Kuvaajien latautuminen ei nopeudeltaan kuitenkaan poikkea merkittävästi Velocityyn tehdystä toteutuksesta ja sitä olisi haastavaa soveltaa aiottuun käyttötarkoitukseen. Statistiikkaa on hyvin tarjolla juuri siihen käyttöön kuin työkalu on tarkoitettu, mutta työkalua ei voi ajatella käytettäväksi dynaamisen testiympäristön kanssa erilaisen datakeruutavan vuoksi.

Työkalu B on raportointiportaali, josta näkee kuvaajia, testituloksia ja muuta niihin liittyvää. Oleellisin osa työkalussa työn kannalta on kuvaajat, joissa nähdään automatisoitujen testilinjojen käyttöaste joko tunti-, päivä-, viikko-, kuukausi- tai vuo-

sitasolla. Vaihtoehtoisesti käyttöaste voidaan jaotella pylväsdiagrammissa testilinjojen tai testilinjatyyppien mukaan. Testiin käytetty aika on jaoteltu monipuolisesti ja koostuu mm. varauksesta, saatavuudesta, poiskytketyymisestä ja regressiosta, mutta ajan yksikköä ei kerrota missään. Tämänkin työkalun yksi isoimmista ongelmista on sen hitaus. Työkalu ei näytä millään tapaa staattisten testilinjojen laitteistoa eikä kerro testilinjoista muuta kuin lyhenteistä ja numerosarjoista koostuvan nimen.

## 3 TYÖKALUTARPEIDEN MÄÄRITTELY TOIMINTOJEN KAUTTA

### 3.1 Vaatimusmäärittely

Vaatimusmäärittelydokumentin toteutus oli yksi tärkeimmistä työn konkreettisista tuloksista. Sen avulla määriteltiin nimensä mukaisesti vaatimukset koskien raportointi- ja monitorointitarpeita, jotta järjestelmän toteutuksessa tehdyt ratkaisut olisivat perusteltuja. Lisäksi vaatimusmäärittelyn avulla pyrittiin selvittämään lähtökohdat ja tavoitteet hyvin huolellisesti, jolloin lopputulosta haittaavat suunnitteluvirheet saatiin minimoitua. Työssä päätettiin tehdä ensiksi toimintoanalyysi, josta lopulta muodostui tieto- ja raporttitarpeet. Toimintoanalyysissä määriteltiin toimintojen kautta kuvaajatarpeet ja resurssiluokat. Näiden tietojen pohjalta oli lopulta määriteltävä tietovaraston rakennetta, jotta Spirentilla voitiin laskea tilatarpeet dataa varten. Lisäksi dokumentti sisälsi useita kuvaajaesimerkkejä ja niiden hakukriteerit, jotta suunnitelma olisi ollut selkeä kaikille osapuolille. Lopputuloksena muodostui vaatimusmäärittely työkalulle.

Raportointi- ja monitorointijärjestelmän avulla ei pyritä pelkästään täyttämään testaa- jien tarpeita, vaan sillä on paljon laajemmat käyttömahdollisuudet. Jo pelkästään ympäristön ylläpito- ja tukitehtävien nähdään tehostuvan monella eri organisaatiotasolla varsinkin inventaarion ja käyttäjämäärän kasvaessa. Puuttuminen kapasiteetin loppumiseen voi parhaimmassa tapauksessa olla ennakoivaa ja muitakin ongelmatilanteita voidaan pystyä välttämään hyvissä ajoin ennen niiden eskaloitumista.

Tukihenkilöiden on laboratorion kapasiteetin kasvaessa oleellista puuttua varauksiin, testilinjoihin ja laitteistoon liittyviin ongelmiin proaktiivisesti ja reaktiivisesti. Proaktiivisessa statistiikan tulkitsemisessä olisi oleellista reagoida muuttuvien trendien mukaan, ja reaktiivisesti reagoidessa statistiikasta olisi nähtävissä esimerkiksi virhemäärän kasvua toimimattomien resurssien kohdalla. Mittalaite voisi alkaa aiheuttaa kasvavassa määrin varausvirheitä, mikä ilmenisi helposti vikatilanteisiin painottuvasta kuvaajasta.

Tärkeänä erillisenä vastuualueena tukitoimiin liittyen on verkkopuoleen erikoistuneet henkilöt, jotka vastaavat mm. kytkimien konfiguraatioista ja niihin liittyvistä

vikatilanteista. Erillistä jakoa näiden tukitoimien välillä ei ollut oleellista tehdä, koska funktiona ajattelen tarpeet, konkreettiset tehtävät ja tehtävien lopputulokset ovat hyvin samankaltaisia. Kuitenkin yleensä organisaation verkkotoimista vastaavat henkilöt työskentelevät eri tehtävissä kuin varauksiin ja laboratoriolaitteisiin liittyviin tukihenkilöihin.

### **3.2 Tarkempien kuvaajatarpeiden selvittäminen haastattelujen avulla**

Ennen vaatimusten määrittelyä suoritettiin haastatteluja, joilla tavoiteltiin uutta näkökulmaa tiimin ulkopuolelta. Omaa tai tiimiä laaja-alaisemman näkökulman saavuttamiseksi oli tarpeellista haastatella eri tehtävissä työskenteleviä henkilöitä, jotta suunniteltuihin funktioihin olisi saatu sisällytettyä kaikki tärkeät tehtävät ja niiden suorittamiseen tarvittavat kuvaajatarpeet jo alkuvaiheessa. Erilaisten kuvaajanäkymien luominen olisi helpompaa, jos tarvittava data on jo tietokannassa. Vastaavasti datavaraston koko pysyisi hallinnassa helpommin, kun haastatteluiden perusteella tarpeettomaksi todetut näkymät karsittaisiin jo suunnitteluvaiheessa pois.

Haastatteluja varten pyrittiin valitsemaan henkilöitä, joilla olisi ollut perustiedot ympäristöstä tiedossa ja mahdollisesti uutta innovatiivista näkökulmaa liittyen etäkäytettävän laboratorion hallintatarpeisiin. Paikalle pääsi lopulta yhteensä 13 henkilöä, jotka hajautettiin neljään eri haastattelukertaan. Useimpiin haastattelu-palavereihin osallistui yleensä joku muukin raportointi- ja monitorointijärjestelmän parissa työskentelevä henkilö. Haastateltavia oli monelta eri osa-alueelta, jotta anti ei olisi jäänyt kapeakatseiseksi. Haastattelut jaettiin sopivan kokoisiin henkilömääriin kertaa kohden, jotta keskustelua olisi herännyt muiden ehdotuksista. Vastaavasti liian suuri osallistujamäärä olisi voinut vaikuttaa negatiivisesti aktiivisuuden suhteessa osallistujamäärään. Haastatteluissa keskustelua herättäneet aiheet huomioitiin kuvaajatarpeiden suunnitteluvaiheessa.

### **3.3 Toimintoanalyysin lähtökohdat**

Toimintoanalyysillä määriteltiin työkalun vaatimukset, josta tuli selvittää Nokian tarpeet järjestelmää varten. Velocityn tietokannasta olisi ollut mahdollista noutaa tarvittava tieto esimerkiksi Python-komentosarjoilla eli REST-ohjelmointirajapin-

taa käyttäen, mutta Spirentilla oli tavoitteena integroida tietokannasta haku palvelun sisälle eli toiminnon toteutuminen mahdollisimman nopealla aikataululla oli tärkeää. Kuvaajatarpeet tulevaa järjestelmää varten olivat pääpiirteittäin selvillä, koska tarve järjestelmän kehittämiseksi oli olemassa ja kehitystarpeita oli jo mietitty ainakin ajatustasolla. Kuvaajia ei voitu kuitenkaan alkaa spesifioida ilman tarkemmin laadittuja perusteita mm. kuvaajissa käytetyistä aikaväleistä, minkä takia laadittiin toimintoanalyysi.

Toimintoanalyysin lähtökohtana oli suunnitella perusteltuja kuvaajatarpeita etäkäyttöiseen laboratorioon liittyvien toimintojen ja niiden suorittamiseen tarvittavien alkutietojen pohjalta. Ympäristöön liittyvät toiminnot pyrittiin jakamaan omiksi järkeviksi kokonaisuuksiksi välittämättä organisaatiotasojen todellisista tehtäväalueista, koska tehtävät voisivat vaihdella organisaatiomuutosten myötä ja funktioiden voi olettaa pysyvän tarpeellisina laboratorion hallinnan suhteen jatkossakin.

Laaditussa toimintoanalyysidokumentissa oli myös oleellista ottaa kantaa siihen, halutaanko nähdä esimerkiksi vain yhden tai vaikkapa kaikkien virtuaalikoneiden käyttöaste tai jokin muu tarkasteltava asia ja millä muilla tavoin tuloksia suodatetaan useammalle käyttäjälle hyödylliseksi. Lopullisesta dokumentista tuli ilmetä riittävän selkeästi vaatimukset, joiden perusteella Spirentin oli mahdollista toteuttaa suunniteltu integrointiominaisuus, joka myös määritteli hyvin pitkälle perinteisestä relaatiomallista poikkeavan tietokannan sisällön.

### **3.4 Toimintoanalyysissa määritellyt resurssiluokat**

Laitekannan suodattamiseksi täytyi määritellä resurssikategoriat, jotta tietovarastosta tehtävän datahaun integroiminen osaksi Velocitya olisi onnistunut. Lisäksi täytyi tuoda ilmi, mistä kentästä esimerkiksi resurssikategoriat määritellään ja suodatetaan. Resurssien luontivaihetta koskien oli sovittu Velocityn käyttöön-otosta alkaen käytännöksi resurssipohjien hyödyntäminen, joka esimerkiksi tehosti laboratoriolaitteiston organisointia. Organisoidumman inventaarion lisäksi resurssipohjia voitiin hyödyntää tehokkaasti raportointi- ja monitorointijärjestelmän kanssa. Yksi pohja voi sisältää esimerkiksi kaikki virtuaalikoneet, jolloin se

niputtaa merkittävän määrän samankaltaisia laitteita helposti tarkasteltavaksi yhden ryhmän alle.

Määrittelyssä tuli kiinnittää huomiota erityisesti logiikan yhteneväisyyteen kaikkien laitteiden kanssa ja sitä varten piti määritellä kentät, joiden mukaan luokittelu tehtäisiin. Kuitenkin esimerkiksi paikkatietojen merkitsemistapa vaihteli riippuen resurssipohjissa määritellyistä kentistä, jolloin esimerkiksi laitetelinetietojen hyödyntäminen tilatarpeiden kuvaamisessa piti suunnitella tarkasti.

Laboratoriossa olevat laitteet voidaan jakaa kolmeen erilaiseen luokkaan, jotka ovat Nokian tuotteet, testilinjalaitteisto ja infrastruktuuri. Laitteiden luokittelun ajateltiin onnistuvan yksinkertaisesti, jolloin käyttäjä olisi voinut suodattaa itse kuvaajanäkymään esimerkiksi pelkät kytkimet. Tämän mahdollistaa neljään tasoon jaetut resurssiluokat eli esimerkiksi tietynmalliset kytkimet voidaan yhdistää samaan näkymään hyödyntäen Velocityn kenttää, jossa listataan malli. Vastaavasti kaikki saman valmistajan mittalaitteet voidaan niputtaa samaan näkymään resurssipohjan perusteella.

Alimmalla tasolla eli resursseilla kuvataan mitä tahansa yksittäistä laitetta, jonka luokittelemiskäytäntö raportoinnin suhteen perustuu käytännössä sillä nähtävään nimeen. Resurssipohjasta luotu resurssi nimetään useissa tapauksissa laitetta kuvaavan nimen mukaan, kuten Keysightin tai Nokian tuotteissa, sille annettun lyhyen tuotenimen ja laitteen indeksointia varten määritellyn ID:n avulla.

Nokian kehitysvaiheessa olevista tuotteista on monia variantteja pienin muutoksin, joita kutsutaan Velocityn yhteydessä tuotevarianteiksi. Tuotevarianttien erot merkitsevät esimerkiksi testejä suunnitellessa, jolloin kyseistä luokittelutasoa voidaan pitää erittäin merkittävänä. Jokaisen tuotteen tiedoissa kirjataan laite tuotekoodien mukaan, jonka perässä merkitään variantin koodi. Jonkun muun kuin Nokian valmistamien laitteiden kohdalla tällaista tasoa ei ole käytössä.

Seuraavaa tasoa laboriolaitteiston kuvauksessa kutsutaan tuotenimeksi. Se muodostuu joko Nokian tuotteista puhuttaessa niiden lyhyistä tuotenimistä tai muissa tapauksissa laitteiden mallinimestä. Tuotenimeksi kutsuttu taso on syytä erottaa omakseen, koska pohjien perusteella tehty luokittelu sisältäisi tilanteesta riippuen liian suuren määrän erilaisia laitteita. Esimerkiksi mittalaitteita voi olla

samalta valmistajalta useita, mutta mallinimen perusteella erot voivat olla toisistaan merkittävästi poikkeavia.

Ylintä tasoa kutsutaan resurssikategoriaksi, jolla tarkoitetaan resurssien kategorisointia resurssipohjien perusteella. Yksi resurssikategoria voi sisältää esimerkiksi kaikki laboratorioissa olevat tietynmerkkiset IP-kytkimet. Nokian tuotteista puhuttaessa yksi pohja voi sisältää esimerkiksi kaikki 4G-tekniikkaan perustuvat laitteet yhden tuoteperheen sisällä.

### 3.5 Toimintoanalyysissä määritellyt funktiot

Toimintoanalyysissä suunnitellut funktiot muodostuvat taulukon 1 mukaisesti lyhyen ja pitkän aikavälin budjetoinnista, infrastruktuurin muutostöistä, testisuunnittelusta, testauksesta ja tukitoiminnoista. Jokaisella funktiolla on omanlaiset tarpeet koskien mm. käytettyä aikaväliä kuvaajien suhteen. Työssä haluttiin pohtia kuvaajatarpeita funktioiden kautta, koska todellisten vastualueiden voidaan ajatella muuttuvan paljon herkemmin.

TAULUKKO 1. Funktioiden määrittely.

Input	CAPEX/OPEX planning function (long term)	Output
Future SW/HW test capacity need Utilization rate and capacity of Nokia products (core/radio cloud, System/Radio module) Utilization rate and capacity of testline devices (traffic generators, instruments etc) Utilization rate and capacity of lab infrastructure (L1/L2 switches, PDU etc)	CAPEX/OPEX planning Planning the need for sharing lab resources between teams	CAPEX/OPEX budget and purchase planning Amount of Nokia products budgeted (core/radio cloud, System/Radio module) Amount of testline devices budgeted (traffic generators, instruments etc) Amount of lab infrastructure budgeted (L1/L2 switches, PDU etc) Amount of lab space budgeted
Input	Lab resource planning function (short term)	Output
Test capacity need Utilization rate and capacity of Nokia products Utilization rate and capacity of testline devices Utilization rate and capacity of lab infrastructure Rack space availability	Planning for increasing/decreasing capacity of a lab resource category Planning for removal of unused lab resources Space and connections for new lab resource is allocated	Pool capacity and connections plan with lab layout design Removal plan of unused lab resources
Input	Lab maintenance function	Output
Planned changes of lab resources Rack space in consolidated lab Utilization rate of Nokia products increases/decreases Utilization rate of testline devices increases/decreases Utilization rate of lab infrastructure increases/decreases	New lab resource and cabling are added to pool physically and virtually Removal of unused lab resources and connections Perform the requested changes regarding the lab inventory (e.g. adding a missing connection)	Optimized lab capacity and connectivity Completed the change requests
Input	Test planning function	Output
Performed reservations Utilization rate and capacity of Nokia products Utilization rate and capacity of testline devices Utilization rate and capacity of lab infrastructure Testline/reservation conflicts	Test planning Planning capacity of lab resources and connectivities	Optimized test capacity plan Resource utilization plan Plan to increase/decrease lab resources
Input	Testing function	Output
Availability of resources	Testline creation Automatic reservation execution Manual reservation execution	Executed test cases Reporting of testline/reservation conflicts
Input	Support function	Output
Reservation not working, error data Topology not working, error data Resource out of service	Reservation error troubleshooting Testline error troubleshooting Resource error troubleshooting	Root cause analyzed Problem solved

Kuvaajien vaatimuksia suunnitellessa oli oleellista määrittellä datan tarkastelu- ja -väli, jotta Spirentilla voitiin laskea karkea arvio datavaraston tilatarpeista

ja arvioida tavoitteemme realistisuus suhteutettuna laitekantaan ja siitä syntyvään datamäärään. Ennen tietokannan määrittelyä täytyi kuitenkin pohtia ympäristöön liittyvät funktiot valmiiksi, jotta mm. ajan suhteen rajaaminen olisi optimaalinen. Oleellisimpien kuvaajatarpeiden havaitsemiseksi tuli miettiä toimintojen kautta, mitä edellytyksiä sen toteutumiseksi oli.

### **3.5.1 Pitkän aikavälin suunnittelufunktio**

Pitkäaikainen suunnittelu keskittyy pääasiassa uusiin laitteisiin, jotka ovat vasta tulossa lähitulevaisuudessa, ja niihin liittyviin mahdollisiin muihin hankintoihin, kuten laitetelinetilän kasvattamiseen. Budjetoinnin yhteydessä käytetään yleensä termejä CAPEX ja OPEX, joilla kuvataan kustannuksia. Capital Expenditure (CAPEX) sisältää pääomakustannukset eli esimerkiksi laitehankinnat ja muut investoinnit, kun taas Operating Expense (OPEX) tarkoittaa käyttökustannuksia eli mm. laboratoriotilojen ja joidenkin laitteiden vuokratkustannuksia (2).

Pitkäaikaisen suunnittelun kannalta koettiin järkeväksi nähdä varausasteeseen ja resurssimäärään liittyvää kuukausitason dataa korkeintaan vuosi taaksepäin, koska budjetointi tehdään aina arviolta vuodeksi eteenpäin, jolloin tarkasteluvälin pitäisi riittää tukemaan päätöksiä. Yksittäisten resurssien tasolla ei koettu olevan tarvetta päästä ainakaan käyttöasteen seurannassa vuoden pituisella aikavälillä, koska ei nähty realistisia käyttötilanteita niin yksityiskohtaisille tiedoille. Voisi olla kuitenkin järkevää nähdä kaikkien virtuaalikoneiden käyttöaste vuoden ajalta, koska se kertoisi mm. kapasiteetista ja testaustarpeiden vaihtelusta. Tuotevarianttitasolla nähtävillä määrillä ja käyttöasteilla ajateltiin olevan merkitystä, koska vanhoilla varianteilla voisi olla vääristävä vaikutus todelliseen kapasiteettiin, koska ne ovat hyödyttömiä tietynlaisten testien suhteen.

### **3.5.2 Lyhyen aikavälin suunnittelufunktio**

Lyhyen aikavälinkään suunnittelussa ei ollut oleellista nähdä yksittäisten resurssien statistiikkaa, koska niin pitkän kuin lyhyenkin aikavälin suunnittelussa päälimmäisenä tarkoituksena ajateltiin olevan laitekannan budjetointi testejä varten. Tuotevarianttien mukaan tehdyn jaon ajateltiin olevan järkevää ainakin lyhyen aikavälin budjetoinnissa, koska testien kannalta varianteilla voisi olla suuri merki-

tys. Lisäksi joillain tiimeillä voisi ajatella olevan tarvetta päivittää uudempaan varianttiin, jolloin käytössä olleilla laitteilla voisi paikata puutteita muunlaisissa testeissä, joissa varianttierot eivät ole niin merkitseviä.

Lyhytaikaiseen suunnitteluun vaikuttaa vahvasti testaustarpeet, jotka voivat tulla yllättäen esille ja loppua yhtä nopeasti. Vaikka jonkin laitevariantin suurin tarve loppuisikin, sitä on hyvä pitää jonkinlainen minimimäärä käyttövalmiina. Näitä laitteita voi hyödyntää testilinjossa, joissa testauksen pääkohteena ei ole kyseisen radiomoduulin variantti, vaan jokin muu testilinjassa oleva laite. Lyhyen aikavälin suunnittelussa on huolehdittava riittävästä niin uusien kuin vanhojenkin laitteiden määrästä, jotta laitemäärä ei paisu yli laboratorion vapaan tilan täyttämällä samalla testeihin vaadittujen laitteiden määrän.

### **3.5.3 Laboratorion ylläpitofunktio**

Laboratorion ylläpitofunktion oleellisimpana vastuualueena ajateltiin olevan kytkentöjen ja laitekannan fyysiset ja virtuaaliset muutokset. Vaikka lyhyen aikavälin suunnittelufunktion olisi oleellista huomioida myös tilatarpeet suunnitteluissaan, niin samaa tietoa tarvitaan myös ylläpitofunktiossa hieman eri perspektiivistä. Lyhyen aikavälin suunnittelussa laboratorion tarkastelutarve kohdistuu enemmän siihen, että joka tiimillä on tarpeeksi laitteita käytettävissä testeihin, kun taas ylläpidon tehtävänä on suunnitella laitteiden järkevät konkreettiset sijainnit ja muutostyöt laboratorioon. Liian pitkän matkan päähän tehtävät laitteiden väliset kytkennät voivat olla paikoittain mahdottomia eli vapaa tila jossain tietyssä laitetelineessä tai resurssien vapaus eivät suoraan kerro topologian olevan mahdollinen, jolloin voi olla tarpeellista esimerkiksi siirtää laitteita lähemmäs toisiaan.

### **3.5.4 Testisuunnittelufunktio**

Testisuunnittelussa oleellisinta ajateltiin olevan käyttöastedatan tarkastelu, jolloin mm. testiajankohdan määrittelyä voitaisiin tehostaa huomattavasti. Lisäksi menneiden varausten ja päällekkäisistä laitevarausyrityksistä aiheutuvien konfliktien tiedosta voidaan saada apua koko kapasiteetin hyödyntämisessä verraten tilanteeseen, jossa käytettävissä olisi pelkästään käyttöastedata.

### 3.5.5 Testifunktio

Ympäristöä käyttää eniten testaajat, joten käyttäjäryhmää voidaan pitää tärkeänä kuvaajien suunnittelun kannalta. Testausta voidaan suorittaa automatisoidusti tai manuaalisesti, ja käyttötavan suurimmat erot koskevat topologioiden luontitapaa ja varaustekniikkaa. Automatisoituun testaukseen luodut topologiat toimivat tällä hetkellä pääasiassa abstraktisti. Vaihtoehtoisena tapana voidaan pitää loogisia resursseja, joissa resurssit dedikoidaan pelkästään tietyn robotin käyttöön määrittelemällä varattavat yksilölliset resurssit manuaalisesti.

Abstrakteihin topologioihin siirtyminen tulee lisääntymään tulevaisuudessa, koska abstraktius vähentää päällekkäisten resurssivarausten mahdollisuutta. Tästä syystä manuaalisia testauksia suoritettavia pyritään kouluttamaan tarvittaessa myös tällä osa-alueella. Inventaario ei kuitenkaan ole riittävän laaja nykytilan kaltaiselle käyttäjämäärälle, ja ympäristön käyttäminen on vielä pääasiassa koekäyttöä, joten täysin abstrakteihin resursseihin siirtyminen vie vielä aikaa. Robotin käynnistämät automatisoidut varaukset hyödyntävät REST-ohjelmointirajapintaa, jolloin manuaalisten ja automaattisten varausten erottelu esimerkiksi statistiikan luontia varten helpottuu.

Testaustoiminnon suurin tarve kohdistuu yksittäisissä testilinjoihin käytettyihin laitteisiin ja niiden tilaan ajankohtana, jolloin testejä on tarkoitus suorittaa. Ideallisinta funktion kannalta olisi, kun päätös testilinjain konkreettisesta laitteistosta tai parametrit abstraktien resurssien kohdalla tulisi testisuunnittelufunktiolta. Tällöin testaajalla olisi sopiva määrä erilaisia vaihtoehtoja tarvittavasta testilinjasta, jolloin testaajan aika jäisi automaation tehostamiseen tai muuhun testaukseen liittyvään tehtävään. Testaustavan muuttuessa yhä enemmän kohti automaatiotestausta Velocity ja sen abstraktit testilinjat tulevat parantamaan laitekannan käyttöastetta helpottaen samalla testiajankohtien suunnittelua.

Velocity on tehokkaimmillaan yli maarajojen tai aikavyöhykkeiden tapahtuvassa testauksessa. Tiimikohtaisten budjettien takia koko laitekantaa ei kuitenkaan voida jakaa avoimesti kaikkien käyttöön, ja siksi aluksi olisi syytä pyrkiä tiimien välisiin sopimuksiin, jota varten Velocityssa on integroituna lainausjärjestelmä. Kun kyseessä on rajatumpi määrä laitteita ja testaajia, kuten jokin testaustiimi,

voidaan automaation avulla saada hyödynnettyä normaalisti yöaikana käyttämättä jääneet laitteet tehokkaasti. Laitekannan käyttöaste tasoittuu automaation avulla paremmin vuorokausitasolla. Automaatio siirtää testauskuorman vaivattomasti päiväajan ulkopuolelle, jolloin aikaa jää enemmän testitulosten analysointiin ja pakollisten manuaalisten testien suorittamiseen.

### **3.5.6 Tukifunktio**

Tukitoimintojen tarkoituksena on auttaa testaajaa, kun testilinja, varaus tai yksittäinen resurssi aiheuttaa testausta haittaavan ongelman. Virhetietojen tarkastelu on funktion tärkein apukeino ongelmien ratkaisemiseksi. Virheiden kokoaminen kuvaajanäkymään voi nopeuttaa ongelmanratkaisua ja helpottaa kokonaiskuvan hahmottamista. Kokonaiskuvasta voisi parhaimmillaan paljastua ongelman lähteeksi vikaantunut kytkin, kun alkuperäinen vika olisi voinut näkyä vain tietyn yksittäisen laitteen kohdalla. Virheitä ei suunnittelun aikana luokiteltu millään tapaa, joten ennen ominaisuuden integrointia oli oleellista keskustella aiheesta Spirentin kanssa. Virheilmoituksista ehdotettiin tehtäväksi kategorisointia perustuen viesteissä esiintyviin avainsanoihin, joiden perusteella oli mahdollista luoda selkeä ID-luokittelu.

## 4 TYÖKALUT JA TEKNIIKAT

Datan tuominen Velocitysta onnistuu vain REST-ohjelmointirajapintaa hyödyntäen, joten raportointi- ja monitorointityökalun valinta oli jo melko selvää projektin alkuvaiheessa. Lisäksi valintaa tuki Spirentin aiempi kokemus valittujen työkalujen kanssa, jolloin soveltuvuudesta voitiin olla varmoja. Velocityssa olevaa dataa voidaan tarjota visualisointityökalun saataville valmiiksi oikeassa muodossa integroidun ominaisuuden avulla. Työn kannalta oli kuitenkin oleellista tutustua muihin toteutuksiin yrityksen sisällä, jotta pääpiirteinen tarve halutuista kuvaajista olisi selvinnyt. Raportointi- ja monitorointitarpeet olivat paljon laajemmat kuin missään muussa aiemmin sisäiseen tuotekehitykseen toteutetussa työkalussa, joten aiempien toteutusten soveltaminen ei ollut kaikilta osin järkevää.

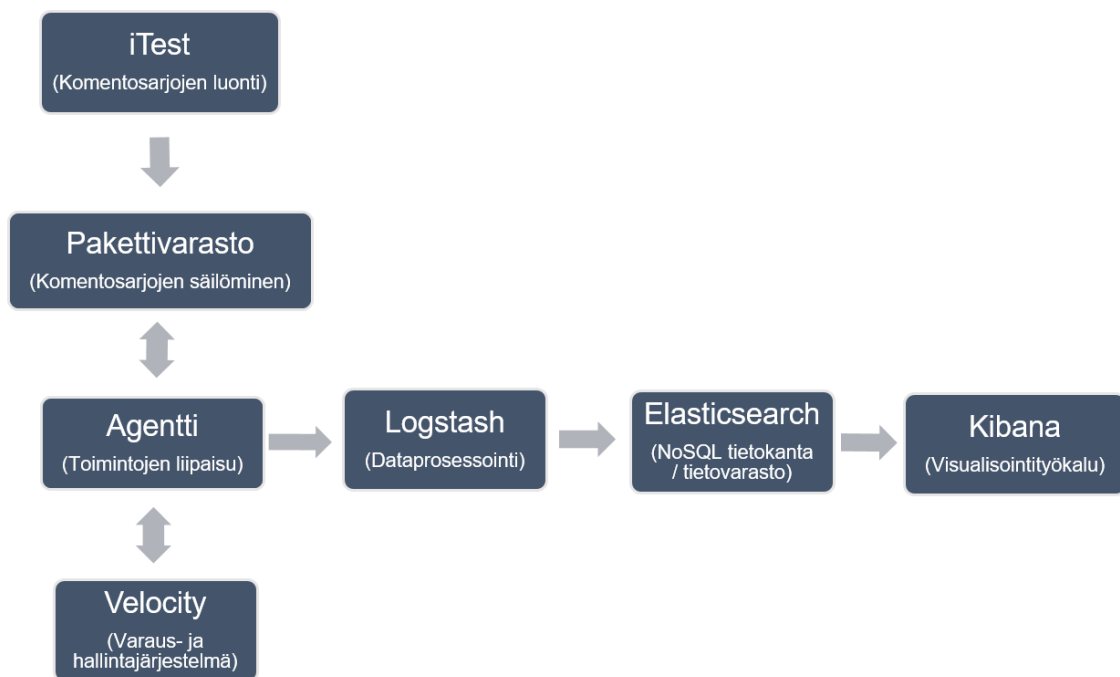
Työn yhdeksi tärkeimmistä työkaluista valikoitui jo alkuvaiheessa Elasticin avoimen lähdekoodin työkalupaketti, johon kuuluu Logstash, Elasticsearch ja Kibana. Logstash on palvelinpuolen putki dataprosessointiin, joka kykenee vastaanottamaan tietoa useasta lähteestä samanaikaisesti ja suodattaa sen tietokantaa varten. Elasticsearch on haku- ja analysointikone, jolla toteutetaan perinteisestä relaatiomallista poikkeava tietokanta (NoSQL). Kibana on visualisointityökalu, jolla voi muuttaa datan monenlaiseksi eri kuvaajaksi ja viedä työkalusta helposti esimerkiksi omalle nettisivulle. (3.)

### 4.1 Järjestelmäarkkitehtuurikuvaus

Työn alkuvaiheessa työkalujen selvittyä päätettiin laatia alustava suunnitelma, jossa summautuu järjestelmän rakenne. Järjestelmän rakenteesta täytyi olla aikaisessa vaiheessa suunnitelma, jotta voitiin suunnitella mm. datan esittämistapoja huomioimalla mahdolliset rajoitteet työkalujen suhteen. Samalla ongelma-kohtiin reagoimista olisi voinut nopeuttaa, jos sellaisia olisi ollut havaittavissa jo ennen työn toteuttamista.

Arkkitehtuurissa yhdistyvät Spirentin ja Elasticin työkalut, joiden välillä kulkevien käskyjen tai tietojen suunta voidaan nähdä kuvassa 3. Automaatio toimintojen tekemiseen yhtenä vaihtoehtona voidaan ajatella olevan iTest-työkalulla tehdyt komentosarjat. Spirentin iTest on erillinen tuote, joka on integroitu osaksi Velocitya.

Sen avulla voidaan luoda vaivattomasti esimerkiksi uudelleenhyödynnettäviä automaatioastetta nostavia ja testausta tukevia toimintoja. Lisäksi iTestin avulla voidaan kontrolloida interaktiivisesti yli 30:tä erilaista sessiotyyppiä, kuten REST-ohjelmointirajapintaa, HTTP-protokollaa ja SSH-protokollan yli hallittavaa komentoriviä. (4.) Työkalulla luotujen komentosarjojen sijasta voidaan hyödyntää esimerkiksi Pythonilla tai Bashilla tuotettua ohjelmakoodia. Millä tahansa tavalla komentosarjat laaditaan, niin tiedostot pitää siirtää Velocityn käyttöliittymän avulla pakettivarastoon (repository). Agentin tehtävänä on huolehtia ajoitettujen toimintojen suorittamisesta ja suorittaa niitä koskevien komentosarjojen ajaminen oikealla hetkellä.



KUVA 3. Järjestelmäarkkitehtuurin alustava määrittely.

## 4.2 Dataprosessointi

Logstash on avoimen lähdekoodin datankäsittelykone olemalla samalla kyvykäs hyödyntämään reaaliaikaista tiedonsiirtokanavaa. Sillä voidaan yhtenäistää dynaamisesti dataa erillisistä lähteistä ja normalisoida se käyttäjän haluamaan sijaan. Toimipisteissä on yleensä oma palvelimensa Velocitylle, jolloin Logstashin avulla voisi prosessoida dataa useasta sijainnista samanaikaisesti. Mikä tahansa tapahtuma voidaan rikastaa ja muuntaa erilaisten tulo-, suodatus- ja lähtöliitännäisten avulla. Sisäkkäisten avaimen ja arvon pareja voidaan yksinkertaistaa ja työkalussa löytyy tuki JSON-tiedostoformaatile. (5.)

Logstashin tärkeimpänä tehtävänä työn kannalta on datan suodatus, jonka jälkeen data on hyödynnettävissä analytiikka- ja visualisointityökalujen kanssa. Velocityn tietokannasta haetaan tarvittava data, jota voidaan suodattaa JSON-tiedostomuotoa varten tarkoitetun filterin avulla. Velocitysta saatua dataa pitää muotoilla muillakin tavoilla, jotta se on yhteensopiva Kibanan kanssa. Suodatuksen jälkeen data tallennetaan Elasticsearchiin. Datakäsittelyltä tai Logstashin suodattimien määrittelyltä voidaan välttyä siinä tapauksessa, jos datahaut suoritetaan Velocityn kautta, jolloin data saadaan muotoiltua automaattisesti Kibanan hyväksymään muotoon.

### **4.3 Tietovarastointi**

Velocityssa sijaitsevan operatiivisen tietokannan yhteyteen oli tarvetta ottaa käyttöön erillinen tietovarasto, johon kopioidaan kaikki hyödynnettävissä oleva historiallinen data. Operatiivinen tietokanta on osana Velocitya samalla palvelimella ja se sisältää kaiken datan koskien mm. resursseja, varauksia ja niistä kerättyä statistiikkaa. Operatiivisen tietokannan kuormituksen minimointi täytyi huomioida jo raportointi- ja monitorointijärjestelmän suunnitteluvaiheessa, koska tietokantarakenne oli muodostunut olemassa olevien tarpeiden ja suunniteltujen työmäärien mukaan, kuten tietynlaisten yksinkertaisten hakutoimintojen hyödyntäminen. Tästä syystä tietovaraston hyödyntäminen raportointi- ja monitorointijärjestelmän kanssa oli toimivuuden kannalta paras ratkaisu.

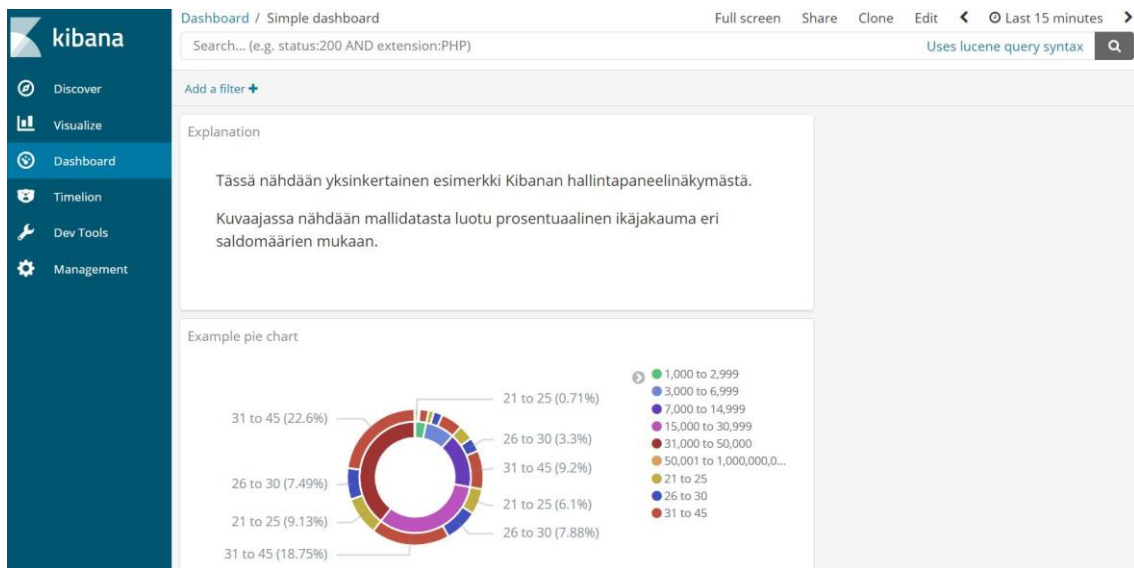
Tietovarastoissa olevasta datasta suoritettavat kyselyt voivat olla paljon moniulotteisempia aiheuttamatta hitautta tai muita ongelmia erillisellä palvelimella toimivaan laboratorion hallinta- ja varausjärjestelmään. Eri toimipaikkojen välillä on erilliset paikalliset palvelimet Velocitylle, jotka suorittavat vain paikallisten laboratoriotilojen toimintoja. Resursseihin ja statistiikkaan liittyvien tietojen hyödyntämisen kannalta jaosta ei kuitenkaan tarvitse välittää. Tietovarastot suunnitellaan lähtökohtaisesti aina tukemaan monista eri lähteistä noudettua dataa, jonka avulla voidaan koota esimerkiksi useasta eri laboratorioympäristöstä dataa ja tarkastella sitä moniulotteisesti. Data vie enemmän tilaa tietovarastossa kuin pelkkää operatiivista tietokantaa hyödyntäen, mutta datasta saadaan huomattavasti mukautuvampi haluttujen asioiden esittämistä varten. Datan voidaan ajatella olevan tietovarastossa kuutio, jossa voidaan tarkentaa esimerkiksi kvartaalitasolta

kuukausinäkymään eli datasta saadaan mukautuva erilaisiin käyttötarkoituksiin. Datan säilymisen kannalta se voidaan suojata vain lukemiseen tarkoitettuun tilaan, jolloin dataa voidaan lisätä vaarantamatta jo olemassa olevan dataa. (6.)

Elasticsearch on REST-ohjelmointirajapintaa käyttävä haku- ja analytiikkakone, joka kykenee hallitsemaan monenlaisia yhtäaikaista toimintoja. Työkalu on erittäin nopea isojenkin datamäärien kanssa, koska kaikki on indeksoitu. Elasticsearch skaalautuu saumattomasti yhdessä tietokoneessa ajettavasta solmusta klusteroituun isompaan kokonaisuuteen halliten automaattisesti indeksien ja kyselyjen eli kuorman jakamista eri solmujen välillä. Lisäksi työkalun kanssa on mahdollista hyödyntää Elasticin ja yhteisön tuottamia luonti- tai ylläpitoasiakasohjelmia, joita on toteutettu kattavasti monella ohjelmointikielellä. (7.) Elasticsearchin käyttäminen ensisijaisena datavarastona on mahdollista, mutta tärkeiden tietohäviöiden takia olisi oleellista varautua varmuuskopioiden avulla, jos raakadata ei säily Velocityn omassa tietokannassa.

#### **4.4 Statistiikan esittäminen visuaalisesti**

Kibana hyödyntää Elasticsearchiin indeksoitua dataa kehittyneen analysoinnin ja visualisoinnin avulla tarjoten mahdollisuuden visualisoida esimerkiksi aikajaksollista ja maantieteellistä dataa sekä havaita poikkeavuuksia datassa koneoppimisen avulla. Kibanan avulla suurtenkin datamäärien sisällön ymmärtäminen ja tarkastelu tapahtuvat käyttäjälle yksinkertaisella tavalla. Työkalun käyttöliittymää päästään tarkastelemaan selaimesta syöttämällä palvelimen IP ja portti, jossa Kibana-resurssi on aktiivisena. Kuvassa 4 nähtävää esimerkinäkymää hallintapaneelista voidaan jakaa nopeasti ja helposti eteenpäin. Hallintapaneelin näkymä voidaan jakaa nähtäväksi esimerkiksi verkkosivulla, jakamalla näkymää URL-osoitteen perusteella tai tuomalla kuvaajanäkymä PDF-tiedostomuodossa. (8.)



*KUVA 4. Esimerkinäkymä hallintapaneelistä.*

Kun data on Elasticsearchissa valmiina, niin Kibanassa täytyy määritellä indeksimalli, jotta dataa voidaan hyödyntää visualisointityökalussa. Management-välilehdellä etsitään ja valitaan indeksoitavat datajoukot sekä datan sisällöstä riippuen kerrotaan sen sisältävän aikataason tietoja. Indeksiä lisätessä voidaan käyttää asteriskia jokerimerkkinä, jolloin esimerkiksi indeksin sisältäessä päivämäärän voidaan rajata halutun datan sisältävän vain tietyn kuukauden. (9.)

## 5 TYÖN TOTEUTUS

Työn yhdeksi tavoitteeksi asetettiin prototyypisovelluksen toteuttaminen, josta olisi ilmennyt muutaman kuvaajaesimerkin avulla visualisoinnin tuomat edut verrattuna nykyiseen Velocityssa olevaan статистиikkänäkymään. Prototyypisovelluksen avulla oli tarkoituksena todistaa idean toimivuus ja hyödyllisyys käytännössä. Tavoitetta ei kuitenkaan päästy toteuttamaan loppuun asti johtuen mm. tehdyistä päätöksistä, joiden takia prototyypisovelluksen toteutus menetti siitä saatavaa hyötyä suhteessa jäljellä olevaan aikaan ja jätettiin lopulta toteuttamatta. Datahakua ja datan hyödyntämistä Kibanassa olisi voinut havainnollistaa oman yksinkertaistetun Python-komentosarjoja hyödyntävän esimerkin avulla. Sillä ei kuitenkaan nähty olevan juurikaan lisäarvoa työn tai ominaisuuden toteutuksen kannalta ja lisäksi ongelmana oli ajanpuute.

Työn edetessä tuotettiin Spirentille toimintoanalyysi, koska sen perusteella oli tarkoitus integroida ominaisuus osaksi Velocitya. Dokumentin tavoitteena oli tuoda esille tarpeet Velocityyn integroitavan ominaisuuden osalta, jonka avulla tarvittava data olisi voitu hakea operatiivisesta tietokannasta jatkohyödyntämistä varten. Toimintoanalyysissä pyrittiin määrittelemään osia tulevasta tietovarastosta ja muuta siihen liittyvää, jotta Spirentin tuottama toteutus vastaisi suunnitelmia. Määrittely saatiin suoritettua työn loppuun mennessä ja sen avulla Spirentilla voitiin aloittaa suunnittelu- ja toteutustyö.

Tietovarastorakennetta pyrittiin havainnollistamaan taulukon 2 avulla. Järjestelmässä suunniteltiin olevan neljä pääkohtaa, jotka olivat edellytyksenä haluttujen kuvaajien luonnin kannalta. Käyttöaste, resurssi-, konflikti- ja virhemäärä olivat haastattelujen, muiden keskustelujen ja pohdinnan perusteella oleellimmat datatarpeet. Työn aikana resurssit päätettiin jakaa neljään eri tasoon, jotta esimerkiksi yhden resurssin sijasta olisi voitu tarkastella koko tuotenimeen kuuluvaa laitekantaa.

TAULUKKO 2. Suunnitelma tietovaraston datatarpeista ja resurssitasoista.

Data	Resource level	Grouping					Location			Storing time (granularity)			
		Grouping by test area(resource group)	All resources without grouping	Resource grouping by reservations filtered by topology	Resource grouping by reservations	Grouping by racks (room + rack ID)	Grouping by error IDs	Site	Room	No location grouping (=all)	Hour	Day	Week
Utilization rate	Resource			X	X					14d	60d		
	Product variant	X	X				X	X	X	14d	60d	3m	13m
	Product name	X	X				X	X	X	14d	60d	3m	13m
	Resource category	X	X				X	X	X	14d	60d	3m	13m
Amount of resources	Resource	X	X							14d	60d		
	Product variant	X	X				X	X	X	14d	60d	3m	13m
	Product name	X	X			X	X	X	X	14d	60d	3m	13m
	Resource category	X	X			X	X	X	X	14d	60d	3m	13m
Amount of conflicts	Resource	X	X	X	X					14d	60d	13m	13m
	Product variant	X	X				X	X	X	14d	60d	13m	13m
	Product name	X	X				X	X	X	14d	60d	13m	13m
	Resource category	X	X				X	X	X	14d	60d	13m	13m
Amount of errors	Resource	X	X	X	X	X	X			14d	60d	13m	13m
	Product variant	X	X				X	X	X	14d	60d	13m	13m
	Product name	X	X				X	X	X	14d	60d	13m	13m

Resursseja ja niihin liittyviä datatarpeita oli oleellista suodattaa tai summata muillakin tavoilla. Resurssiryhmillä Velocityssa tarkoitetaan tiimejä ja niiden nähtävillä olevaa laitekantaa. Tiimitasolla nähtävän suodatuksen koettiin olevan erittäin oleellista tilastoinnin hyödyntämisen kannalta, ja vastaavasti suodatuksessa täytyi olla vaihtoehto jättää huomioimatta tiimien väliset rajaukset.

Topologioiden ja varausten perusteella tehtävä suodatus oli tärkeää huomioida, koska tilastoinnin voitiin ajatella tukevan esimerkiksi testausta. Topologiat eivät lähtökohtaisesti sisällä mitään resursseihin viittaavaa dataa, mutta varaukset tehdään aina topologioiden pohjalta. Tästä syystä voitiin ajatella olevan mahdollista yhdistää monta samasta topologiasta tehtyä varausta yhteen kuvaajanäkymään.

Tällaisella näkymällä voisi vertailla toisistaan poikkeavia testilinjoja yksittäisiä varauksia laajemmalla tarkasteluvälillä.

Virheiden tarkastelun kannalta suunniteltiin pareto-diagrammin kaltaista näkymää, koska selkeä virheluokittelu puuttui työn aikana ja sen ajateltiin kuitenkin olevan hyvin tärkeää. Tunnisteiden avulla tehtävää luokittelua ja yksityiskohtia ei työn aikana tarvinnut määrittellä tarkemmin, mutta idean toteuttamiskelpoisuutta oli syytä pohtia. Virheviesteistä oli teoriassa mahdollista tehdä luokittelua hyödyntäen esimerkiksi toistuvia avainsanoja. Yleisimpiä virheitä aiheuttavia vikoja olivat mm. päällekkäiset varausyritykset ja ajuri- tai yhteysongelmista aiheutuvat aikakatkaisut.

Resursseista oli kirjattuna tilatieto laiteteline-, huone- ja toimipaikkatasolla, joista voitiin ajatella olevan apua laboratoriokapasiteetin määrittelemisessä. Merkitsemistavat vaativat kuitenkin yhtenäistämistä ja tarkentamista, jotta tilatiedoista saadaan irti kaikki potentiaali. Nykyisellä toteutuksella voisi kuitenkin havainnollistaa kuvaajanäkymän tarjoamia mahdollisuuksia kapasiteetin hallinnan kannalta, joten datatarpeet olivat oleellisena osana suunnittelua.

Datan säilömisäika ja datatiheys olivat mukana suunnittelussa, koska vaatimusten perusteella voitiin arvioida Spirentilla mahdollisia tilatarpeita tietovaraston sisältämällä palvelimella. Tuntitason dataa oli oleellista summata kahden viikon ajalta, koska viikon ajalta ei olisi välttämättä saanut näkymään selkeitä trendejä. Lisäksi yö- ja viikonloppuajan voidaan ajatella olevan vähemmän suosittua, joten pelkkä päivätason näyttävä data ei ole kaikissa käyttötarkoituksissa yhtä hyödyllistä. Viikko ja kuukausitason datasta on varsinkin konfliktien ja virheiden osalta tärkeää nähdä tietoa vuoden ajalta, koska sen avulla voidaan tehdä johtopäätöksiä mm. laboratorion tehokkuudesta ja kapasiteetin riittävydestä.

## 6 JATKOKEHITYSMAHDOLLISUUDET

Raportointi- ja monitorointijärjestelmää varten oli mahdotonta suunnitella toteutettavaksi kaikkia hyödyllisiä ominaisuuksia määrätyn ajan puitteissa, mutta työn edetessä oli silti oleellista pohtia kaikkien potentiaalisten ideoiden toteuttamisarvoa. Velocitysta on saatavilla todella paljon dataa ja siitä on mahdollista luoda erittäin kattavasti näkymiä niin Kibanan kuin muidenkin työkalujen avulla. Oleellista on kuitenkin pohtia, millaista tietoa käyttäjät voisivat oikeasti tarvita Velocitysta ja kuinka hyvin toteutettavissa nämä ideat voisivat olla.

Toivotut näkymät voivat vaatia muutoksia esimerkiksi resurssitasolla oleviin tietoihin, jolloin on oleellista ottaa huomioon mahdollinen idean toteutettavuus ja siitä saatavat hyödyt. Esimerkiksi aiemmin mainittu laboratoriotilan hallinta kiinnostaa monia, mutta sen toteuttamista työn aikana ei pidetty realistisena tavoitteena, koska resurssitiedoissa olevat paikkatiedot eivät ole yhdenmukaisia. Tilatietojen kirjaamistavasta nähtiin ainakin kahta erilaista tyyliä, joissa kenttien nimeämiskäytännöt ja niihin pyydetty sisältö eivät olleet yhdenmukaisia. Esimerkiksi joissain resursseissa yhteen kenttään oli syötetty laitetelineen koordinaatti-, ID- ja hyllytiedot ja toisessa tapauksessa kaikki tiedot olivat omina kenttinään.

Resurssit luodaan aina pohjien avulla ja niissä olevia kenttiä muokkaamalla saadaan muutos vaikuttamaan myös jo olemassa oleviin resursseihin. Siltikin jokaiselle resurssille pitää korjata arvot uusien kenttien mukaisesti, joten tilatarpeiden esittämistapaa koskevaa ongelmakohtaa ei lähdetty korjaamaan työn aikana. Tilatietoja ei voitu harkita hyödynnettävän optimaalisesti työkalun suunnittelussa ensimmäisessä versiossa ennen tietojen yhtenäistämistä.

Laboratorioympäristön toimivuuden kannalta suuressa roolissa voisi pitää tukitoimien sujuvuutta. Velocitysta saatava statistiikka on ilman integroitujen ominaisuuksien tehostamista tai ulkopuolisten työkalujen hyödyntämistä hyvin pelkistetty. Vikatilanteissa aloitetaan vika-analyysi ja ongelmakohtan alustava rajaus tarkastelemalla Velocityn tarjoamia vikailmoituksia. Käyttäjä voi huomata konkreettisen vian ilmaantuessa samaisen virheilmoituksen Velocityssa, jos vika

on niin kriittinen, että siitä jää merkintä palveluun. Näin tapahtuessa vika aiheuttaa varauksen keskeytymisen. Virheilmoitusten luokittelua ja sitä kautta virhemäärien hyödyntämistä voisi käyttää lähinnä tilastotieteen ja nykytilan ilmaisimena, mutta niillä ei juurikaan ole merkitystä yksittäisen vian korjausnopeuteen. Vikatietojen suurin hyöty ympäristön kannalta voidaan ajatella olevan ongelmakohtien havaitsemisessa. Esimerkiksi verkkolaitteet voivat toistuvasti aiheuttaa uudelleenkäynnistyksellä korjaantuvia yhteysongelmia, mutta vikojen kohdistuessa aina samoihin laitteisiin voidaan paljastaa isompia ongelmakohtia.

Tukipyynnöjä aiheuttavien vikojen hyödyntämisessä havaittiin olevan ongelma, koska tällä hetkellä pyynnöjä voi jättää useamman työkalun kautta. Velocityn ulkopuolella jätetyt tukipyynnot eivät ole hyödynnettävissä tulevassa järjestelmässä. Tukipyynnöjä olisi syytä tehdä enemmän Velocityn kautta, jotta tilastotieteen tulisi tarkempaa. Tarkemman tilastotieteen avulla voidaan tarkkailla tehokkaammin esimerkiksi työpyynnöiden suoritusajaa ja ongelmien kokonaismäärää ympäristön kanssa. Lisäksi tukipyynnöiden suoritusajoilla voi olla vaikutusta suoraan laitteiden käyttöasteeseen, koska testejä ei välttämättä voida suorittaa testitiloin ollessa vikatilassa.

Velocitysta laadittavissa kuvaajanäkymissä ei ole järkevää tarkastella ainoastaan laitekantaa yksittäisten resurssien tasolla, koska isommalla kokonaisuudella on monessa tapauksessa enemmän merkitystä. Abstraktien resurssien yleistynyt käyttö tulee lisäämään entisestään tarvetta tarkkailla ylempiä resurssitasoja, koska järjestelmän päätäessä varattavista laitteista yksittäisten laitteiden käyttöasteella ei ole juurikaan merkitystä. Kun automaatiota aletaan hyödyntää testauksen tukena isommassa mittakaavassa, abstrakteista resursseista tulee enenevässä määrin välttämättömyys.

Kapasiteettia voidaan tulevaisuudessa hallita ennakoivasti varsinkin tilanteissa, joissa manuaali- tai automaatiotestaus ilmoittaa järjestelmälle riittävän ajoissa tarpeesta tehdä tietynlainen varaus. Lisäksi aiempien viikkojen varaus- ja käyttöastetiedoista voisi päätellä hetket, joina kapasiteetti tulee olemaan huipussaan ja milloin olisi tarvetta kasvattaa laitemäärää.

Välillä voi olla oleellista tarkentaa tarkastelunäkymää yksittäiselle tasolle, jos esimerkiksi yksittäinen resurssi aiheuttaa ongelmia useassa erilaisessa testilinjassa pidemmän aikaa. Konfliktit eli yhtäaikaiset saman resurssin varausyritykset voivat aiheuttaa enenevässä määrin ongelmia laitekannan kasvaessa. Ongelmaa voidaan torjua vaihtamalla varattavaa resurssia automaatiolla ja manuaalisten testien tehostamisella statistiikan avulla.

Keskusteluissa on käsitelty myös sähkönkulutuksen huomioimista statistiikanäkymässä. Virranjakoyksiköt näyttävät paljon tietoa, kuten reaaliaikaista virrankulutusta. Virrankulutusta voitaisiin hyödyntää esimerkiksi mitoittaessa virransyöttöä. Eri laitteiden välillä voi olla hyvinkin paljon vaihtelua virrankulutuksessa, jolloin useimmissa tapauksissa laitetelinekohtaisesti asennettujen virranjakoyksiköiden ja statistiikan tuella voisi tehostaa laboratorioympäristön mitoitus töitä. Lisäksi virrankulutustiedolla voisi teoriassa pyrkiä vähentämään turhaa sähkönkulutusta ja näin ollen parantamaan yhtiön ekologisuutta, mutta realistisesti ajatellen tällaiselle datalle ei juurikaan taida olla kysyntää ja voisi olla haastavaa erottaa tarpeeton ja perusteltu kulutusdata.

## 7 YHTEENVETO

Työn tavoitteena oli määrittellä toimintoanalyysidokumentin avulla dynaamisista testilinjoista koostuvan laboratorion raportointi- ja monitorointitarpeet. Suunnitelmien pohjalta aiottiin toteuttaa Proof of Concept -tyylisesti muutaman kuvaajanäkymän havainnollistama järjestelmä, mutta ajanpuutteen vuoksi sitä ei tässä opinnäytetyössä toteutettu loppuun asti. Lisäksi työn alkuvaiheessa sovittiin tavoitearkkitehtuurin määrittelystä raportointijärjestelmälle.

Työn alkuvaiheessa vertailtiin eri vaihtoehtoja, joilla raportointijärjestelmä voitaisiin toteuttaa, mutta jo hyvin alkuvaiheessa käyttöön valikoituivat Elasticin avoimen lähdekoodin työkalut. Työkalut olivat jo esiasennettuna eräälle palvelimelle ja lisäetuna oli Spirentin aiempi kokemus työkaluista Velocityn kanssa, joten yhteensopivuusongelmien välttämiseksi valinta oli perusteltu.

Vaatimusmäärittelyn laatiminen oli yksi työn sovituista tavoitteista, ja siinä oli tarkoituksena määrittellä nykytila, tavoitteet, käyttöympäristö ja käyttäjäryhmät. Toimintoanalyysin avulla määriteltiin työkalun tieto- ja raportointitarpeet, josta lopulta selvisi työkalun vaatimukset. Dokumentti sisälsi alustavien kuvaajatarpeiden määrittelyn toimintojen kautta, resurssien kategorisoinnin eri tasojen mukaan, tietokantarakenteen määrittelyn, konkreettisia esimerkkejä eri toimintoihin tarvittavien kuvaajien sisällöstä ja jokaiselle toiminnolle tietokyselyiden ja niiden näyttämien kuvaajanäkymien havainnollistavat esimerkit.

Tiedonsiirto operatiivisesta tietokannasta raportointitietokantaan päätettiin integroida osaksi Velocitya. Tällöin välttyttiin alun perin suunnitelmassa olleen erillisen työkalun kautta ajettavien Python-komentosarjojen luonnilta, mikä yksinkertaisti ja yhtenäisti palvelua. Tätä integrointiominaisuutta varten täytyi määrittellä Nokian tarpeet yksityiskohtaisesti, jotta raportointitietokantaan olisi saatu hyödyllisin data viipymättä ja voitu samalla hahmottaa suunnitelmien realistisuus huomioiden mm. tallennustilan vaatimukset.

Työn suurimmaksi konkreettiseksi tulokseksi muodostui lopulta toimintoanalyysidokumentti. Sen tavoitteena oli määrittellä mm. tietokannan sisältötarpeita

Spirentille ominaisuuden integroimista varten, jotta Velocityn operatiivisesta tietokannasta olisi saatu siirrettyä tarvittava data Elasticsearchiin ja sitä kautta visualisointityökaluna käytettyyn Kibanaan. Tavoitearkkitehtuurin määrittely oli suoritettu jo alkuvaiheessa, mutta sitä jouduttiin vielä myöhemmässä vaiheessa muokkaaman johtuen muuttuvista suunnitelmista koskien integroitavaa ominaisuutta. Alkuperäisessä tavoitearkkitehtuurissa suunniteltiin datahaun toteuttamista Python-komentosarjoilla, mutta lopulta koettiin hyödyllisemmäksi sulauttaa ominaisuus osaksi Velocitya. Prototyypisovelluksen toteuttamisesta jouduttiin luopumaan, koska Velocityyn ei saatu integroitua ominaisuutta riittävän ajoissa ja ajanpuutteen vuoksi ei voitu toteuttaa Pythonin avulla vastaavaa ominaisuutta.

## LÄHTEET

1. Särkiniemi, Markus 2018. Automatic Generation of a Dynamic Test Environment for C-RAN. Master's thesis. University of Oulu, Faculty of Information Technology and Electrical Engineering. Luettavissa Oulun yliopiston sisäisessä verkossa.
2. Maverick, J.B. 2017. What is the difference between CAPEX and OPEX? Investopedia. Saatavissa: <https://www.investopedia.com/ask/answers/020915/what-difference-between-capex-and-opex.asp>. Hakupäivä 28.3.2018.
3. What is the ELK Stack? 2018. Elastic. Saatavissa: <https://www.elastic.co/elk-stack>. Hakupäivä 20.3.2018.
4. Spirent iTest, Test as a Service Platform. 2018. Spirent. Saatavissa: [https://www.spirent.com/-/media/Datasheets/Broadband/PAB/CLEAR/Spirent\\_iTest\\_datasheet.pdf](https://www.spirent.com/-/media/Datasheets/Broadband/PAB/CLEAR/Spirent_iTest_datasheet.pdf). Hakupäivä 20.4.2018.
5. Logstash Introduction. 2018. Elastic. Saatavissa: <https://www.elastic.co/guide/en/logstash/current/introduction.html>. Hakupäivä 28.3.2018.
6. Data Warehousing - Quick Guide. 2018. Tutorials Point. Saatavissa: [https://www.tutorialspoint.com/dwh/dwh\\_quick\\_guide.htm](https://www.tutorialspoint.com/dwh/dwh_quick_guide.htm). Hakupäivä 7.5.2018.
7. The Heart of the Elastic Stack. 2018. Elastic. Saatavissa: <https://www.elastic.co/products/elasticsearch>. Hakupäivä 28.3.2018.
8. Your Window into the Elastic Stack. 2018. Elastic. Saatavissa: <https://www.elastic.co/products/kibana>. Hakupäivä 28.3.2018.
9. Defining Your Index Patterns. 2018. Elastic. Saatavissa: <https://www.elastic.co/guide/en/kibana/current/tutorial-define-index.html>. Hakupäivä 7.5.2018.