

Ere Viitanen

METSÄKONEDATAN HYÖDYNTÄMINEN PILVIPALVELUYMPÄRISTÖSSÄ

TIIVISTELMÄ

Ere Viitanen: Metsäkonedatan hyödyntäminen pilvipalvelu ympäristössä
Pro gradu -tutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Toukokuu 2023

Metsänhoitotyön koneellistumisen myötä hakkuuyrityksiltä vaaditaan yhä tehokkaampaa puuntuotantoa ja työn jälkeä seurataan maanlaajuisesti. Yritykset ovat velvollisia raportoimaan tuotantonsa valvoville tahoille, mutta järjestelmät eivät ole tarkoitukseen järin tehokkaita. Pilvipalveluiden avulla prosessia voidaan tehostaa ja informaation saatavuus sekä koostaminen helpottuu. Pilvipalvelut mahdollistavat järjestelmien kehitykseen sujuvuutta verrattaen vanhoihin menetelmiin. Kehitystyössä ei olla enää sidottuina yhteen monimutkaiseen ratkaisuun, vaan työkaluja voidaan käyttää monipuolisemmin käyttötapauksesta riippuen. Oleellisia osia pilvipalveluissa on yhtenäinen tiedonlähde ja resurssien ehtymättömyys. Ohjelmistoversioiden julkaisut ovat saumattomia niin kehittäjille kuin käyttäjillekin.

Tämän tutkielman tarkoituksena on kehittää John Deere Forestry Oy:n tarjoamia ohjelmistoja ja antaa katsaus metsäkonetyöhön liittyvään standardiin sekä pilvipalveluiden arkkitehtuuriin. Keskeisenä aiheena on siirtää TimberOffice 5 -työpöytäsovelluksen ominaisuuksia pilvipalveluihin ja kehittää ratkaisuja ongelmakohtiin, kuten kuljettajien erottaminen toisistaan hakkuuorganisaation laajuisesti sekä tarjota standardin mukaiset raportit käytettäväksi helpommin. Tutkielman laajuus on rajattu pohjoismaissa käytettävien puunkatkontamien tuottamaan dataan ja pilvipalveluarkkitehtuuriin. Aluksi käsitellään kansainvälisen standardin muutoksia sekä ongelmakohtia, jonka jälkeen aihetta tarkennetaan metsäkonedatan käsittelyyn pilvipalveluympäristöissä. Lopuksi käydään läpi mahdollisia ratkaisuja, joista on valikoitu mahdollisimman hyvin käyttötarkoitukseen sopivat sovellutukset.

Avainsanat: Metsäkone, StanForD-standardi, XML, AWS, tietokannat, pilvipalvelut

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

Sisällysluettelo

1	Johdanto	1
2	Metsätalouden data	3
2.1	Metsänhoitotyön asettamat vaatimukset	3
2.2	Standardit	6
2.3	Pakkaaminen	10
2.4	Tietomallit	12
2.5	Tiedon käsittely	15
3	Datan ympäristö	17
3.1	Pilvipalvelut	17
3.2	Reuna- ja sumulaskenta	19
3.3	Luottamus ja turvallisuus	21
4	Ohjelmistoprojekti John Deere Forestry Oy:llä.....	24
4.1	Datan lähteet	24
4.1.1	Vanhat olemassa olevat järjestelmät	24
4.1.2	Uusi järjestelmä	26
4.1.3	Kuljettajien hallinta	27
4.1.4	Hakkuuohjeiden liittäminen leimikkoon	29
4.2	Statistiikka ja analysointi	30
4.3	Tietokannan vaatimukset	30
4.4	StanForD-tietokannan arkkitehtuuri ja datan käsittely pilvessä	33
5	Jatkokehitys	39
6	Yhteenveto.....	41
7	Viiteluettelo	43

1 Johdanto

Metsätöiden kannattavuuden parantaminen on ollut pitkään kehityksen alla ja on edelleen. Metsiä on hakattu aikaisemmin käsin ja nykyisin metsäkonein. Jo ennen langattomia yhteyksiä metsistä on haluttu tietää sen rahallinen arvo ennen hakkuutyötä ja sen jälkeen. Nykyisin metsien hoitotyötä seurataan tarkoin ja yritysten tuotantotieto päätyy lopulta myös EU:n tilastoihin.

Metsäkoneilla tuotetaan dataa erilaisien mittarien ja sensorien avulla työnlaadun takaimiseksi. Hakkuuprosessin yhteydessä koneilta kerätty data tallennetaan eteenpäin lähetettäviin raportteihin. Raportit ja hakkuutyön moni muu osa-alue on sidottu Ruotsin ja Suomen yhteistyönä kehitettyyn maailmanlaajuiseen standardiin. Standardia on kehitetty jo pitkään. Aluksi se oli metsäkoneille helposti luettavissa olevaa dataa, mutta ihmisille ja jälkitarkasteluun se ei soveltunut. Siksi StanForD-standardia alettiin jalostaa XML-pohjaiseksi ja loogisesti jäsennellyksi. Saman standardin alla rinnakkaiselossa on vanha StanForD sekä uusi StanForD2010.

Pilvipalvelut ovat uusi verkkosivujen kehittämisen pohja, sillä yritysten ei tarvitse huoltaa ja hallita omia palvelimiaan. Asentaminen on helppoa, verkkosivuille saa näkyvyyttä, viive vähenee ja pilvessä laskentatehon skaalaaminen on haluttaessa loputonta. Pilvi tarjoaa paljon erilaisia palveluita, joita voidaan käyttää ratkaisuin moniin erilaisiin käyttötapauksiin niiden laajuudesta huolimatta.

Tämän työn tarkoituksena on kehittää John Deere Forestry Oy:n tarjoamia ohjelmistoja ja antaa katsaus metsäkonetyöhön liittyvään standardiin, sekä pilvipalveluiden arkkitehtuuriin. Keskiössä on tarjota StanForD-standardeja mukailien vanhan TimberOffice 5 -työpöytäsovelluksen ominaisuuksia pilvipalvelualustalla.

Tutkielman luvussa 2 käydään läpi koneellisen metsänhoidon asettamia vaatimuksia. Lisäksi luvussa esitellään StanForD-standardia, datan pakkaamista sekä erilaisia tietomalleja. Kolmannessa luvussa siirrytään pilvipalveluiden esittelyyn. Esittelyssä on pilviarkkitehtuurin tasoja, yleisimpiä palveluita, XML-datan käsittelyn työkaluja ja muutamia mahdollisia tietouhkia sekä turvallisuutta. Luvussa 4 keskitytään tarkemmin metsäkonedatan käsittelyn ongelmiin, niiden ratkaisuihin ja datanlähteisiin. Tietokantojen tyypit asettavat rajoituksia ja niiden käyttöä tutkitaan tapauskohtaisesti. Läpi käydään myös rat-

kaisu arkkitehtuurillisesti esitellen tietokantatyyppejä vastineet pilvipalveluina ja monihenkisen tiimin kehittämistyöhön vaadittuja työkaluja. Projektin kehitysehdotuksena esitetään datan tarjoamista asiakkaille, jotta he voivat tarkastella sitä tarkemmin kolmannen osapuolen järjestelmillä. Palveluiden ja selaimen välistä viivettä voidaan vähentää Apache Parquet:in avulla. Yhteenvedossa on koottu yhteen ongelmien ratkaisuja ja niihin johtaneita päätelmiä.

2 Metsätalouden data

Luku sisältää metsänhoitoon liittyvän esittelyn eri tahojen toimista ja heidän vaatimastaan datasta. Data on sidottu StanForD-standardiin, jonka käyttötapaukset ja tekninen viitekehys tarkentuvat luvun loppua kohti.

2.1 Metsänhoitotyön asettamat vaatimukset

Metsiä on hakattu ja hoidettu Suomessa vuodesta toiseen enenevässä määrin. Luonnonvarakeskus Luken tilaston mukaan tukkipuuta oli hakattu ennätys määrä 30 miljoonaa kuutiometriä vuonna 2021. [Luonnonvarakeskus, 2022c] Luken aineistot perustuvat Suomen virallisiin tilastoihin ja EU:n asetuksissa määrättyihin tilastointeihin. Tilastot kootaan kuukausittain ja vuosittain puunostajilta saatujen tietojen perusteella. [Luonnonvarakeskus, 2022a; Luonnonvarakeskus, 2022b]

Metsien kuntoa ja tilaa on pyritty tarkkailemaan jo pitkään. Metsäyhtiöt ovat erittäin tarkkoja datasta ja pyrkivät saamaan aina yhtenäistä eheää tietoa hakatusta ja jäljellä olevasta puustosta sekä metsien kunnosta. Suomessa yhtiöt kuten Stora Enso ja UPM-Kymmene ovat suurimpia toimijoita ja toimivat laajasti koko alalla. Yhtiöt omistavat metsää, järjestävät hoito-, sekä hakkuutyön ja jatkojalostavat puutavaran. Molemmat esimerkkiyhtiöt ovat monikansallisia ja niillä on toimintaa usealla mantereella [Stora Enso, 2022; UPM-Kymmene, 2023]. Isojen toimijoiden puutavarantuotantoprosessi on järjestelmällistä. Kun datan välikädeksi tulee eri mantereelta peräisin oleva hakkuu-urakoitsija, asiat mutkistuvat johtuen esimerkiksi paikallisista mittayksiköistä ja tavoista mitata puutavara.

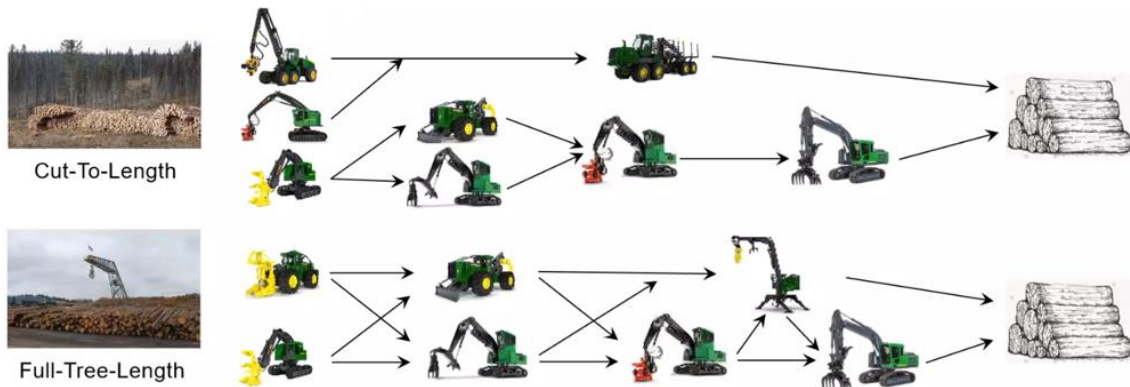
Maailmanlaajuisesti StanForD-standardin avulla ei olla päästy yhteisymmärrykseen siitä, miten katkottu puu kuuluisi mitata. Puutavaran ostava taho haluaa maksaa mahdollisimman vähän puusta, joten metsäkoneissa joudutaan tukemaan erilaisia mittaustapoja ja yhtälöitä. Toimintatavat määräytyvät usein maan yleisten toimintatapojen mukaan. Mittayksiköt ja mittaustavat saattavat vaihdella. Kuoren alta mittaaminen tarkoittaa puutavaran kuoren paksuuden arviointia ja sen poistamista laskennallisesti metsäkoneella. Kuoren alta mittaamisella ja tyviyhtälöillä pyritään määrittelemään sahalla hyödyksi käytettävän puun määrä. Tällaisia käytetään etenkin paikoissa, joissa sahoilla ei pystytä mittaamaan puutavaran määrää erikseen vaan se perustuu koneen antamaan tuotantodataan. Moder-

neimmilla sahoilla on otettu käyttöön laserkeilaus, jolla puutavarasta saadaan eksaktit mitat. Tukin sahauksen suunnittelu helpottuu, joka tarkoittaa sahalle tuottoisampaa liiketoimintaa.

Vaikka vanha StanForD- ja uusi StanForD2010-standardi on kehitetty lähinnä Ruotsin ja Suomen yhteistyönä. Käyttötavat eriävät hieman niiden välillä etenkin puiden kuljetuksessa. Suomessa puiden kuljetus on sidottu tavaralajeihin, jotka määrittävät pitkälti niiden toimituspaikan. Ruotsissa puolestaan puiden keräilyvaiheessa ne kerätään kuljetuksiin, jotka voivat sisältää useita tavaralajeja. Metsäkoneilla tällaisten tapausten hallinta ja tukeminen luotettavasti on yksi keskeisiä tehtäviä mitä tietotekniikalta vaaditaan metsäkoneissa.

Metsänomistajuus on Suomessa muuhun maailmaan verraten harvinaista, sillä suurin osa metsistä on yksityisessä omistuksessa. Yksityinen omistajuus antaa enemmän vapauksia tai vastuuta päättää metsän kasvatustavoista. Suomessa on käytössä kaksi kasvatustapaa: jaksollinen- ja jatkuva kasvatusta. Jatkuva kasvatusta tarkoittaa eri ikäisen puuston kasvattamista samalla alueella. Suurimmat puut kerätään pois ja luodaan tilaa uusille pienemmille puille kasvaa. Jatkuva kasvatusta pyrkii käyttämään hyväkseen puuston luonnollista uusiutumista. Jaksollisessa kasvatuksessa puustoa harvennetaan aina siihen pisteeseen asti, kunnes voidaan suorittaa avohakkuu. Tämä menetelmä vaatii usein maanmuokkausta tai taimien istuttamista avohakkuun jälkeen uusiutumisen varmistamiseksi. Jaksollinen kasvatusta on huomattavasti yleisempi tapa, mutta sitä säätelee vuoden 2014 metsälaki muutos, joka vasta mahdollisti jatkuvan kasvatustavan [Metsäkeskus, 2023; Finex, 2013]

Metsänomistaja haluaa metsästä parhaan mahdollisen hinnan, joka edellyttää sen hoitamista. Hoitotoimet määräytyvät maaston, kasvatustavan ja sertifikaattien mukaan. Tällaisia sertifikaatteja ovat PEFC ja FSC, joista tiukemmin määritellyt hoitovaatimukset ovat FSC:ssä. Suomen metsistä noin 90% on PEFC:n piirissä ja 10% kuuluu FSC sertifikaatin alle. Sertifikaatit ovat suuressa roolissa, sillä ohjeiden noudattamatta jättäminen johtaa sertifioimattomiin puihin, jotka menevät huonommin kaupaksi pienemmällä hinnalla. Sertifikaateissa otetaan kantaa luonnon monimuotoisuuteen, virkistyskäyttöön ja puutavaran jäljittämiseen. Huonekaluihin voidaan laittaa PEFC-merkintä, joka tarkoittaa puun tulevan kestävästi hoidetusta metsästä. [Metsähallitus, 2023]



Kuva 1. Erityyppisiä FT- ja CTL-koneketjuja

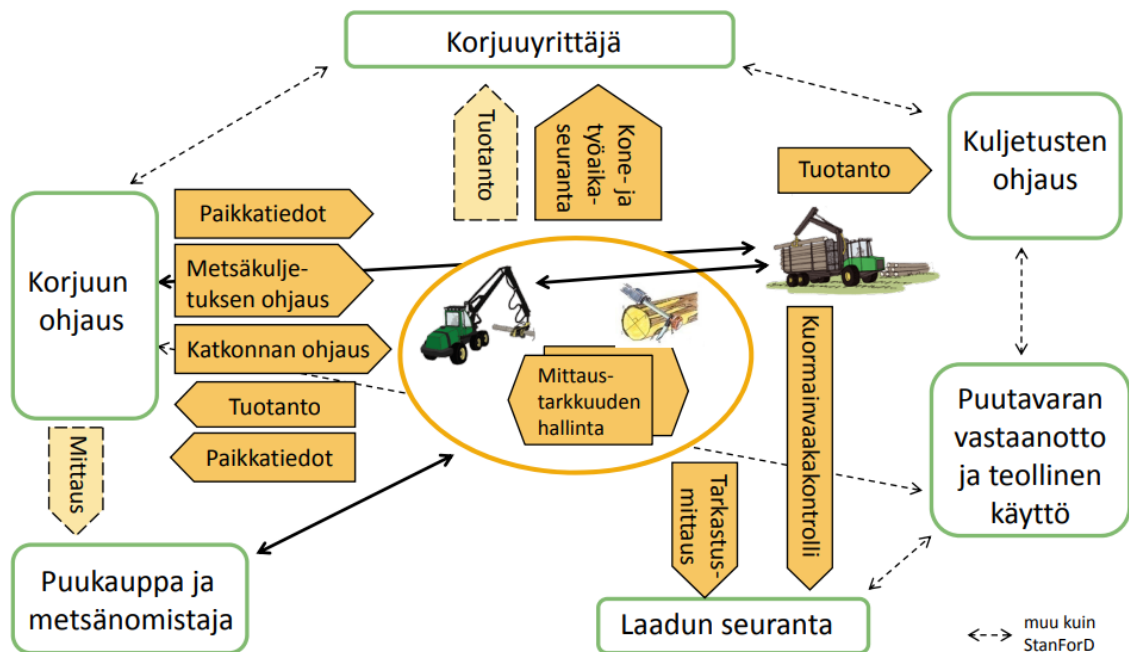
Hoito- ja hakkuutyöt tehdään usein palkatun urakoitsijan toimesta, joka ammattiin kuuluvan koulutuksen myötä omaa tiedot ja taidot kuhunkin tarkoitukseen. Leimikkoon liittyvät koneen tarvitsemat hakkuuohjeet tulevat useimmiten metsäyhtiöltä, jolle puutavara on myyty. Kaadettavat puut on mahdollista sopia ja merkata metsänomistajan toimesta, mutta se on usein tarpeetonta. Puut voidaan merkitä maalilla tai nauhalla. Merkkaustapaa kutsutaan leimaamiseksi. Tästä merkkaustavasta tulee metsätalouden sana ”leimikko”, jolla tarkoitetaan hakattavaksi suunniteltua metsäkuviota. Nykyisin yleisessä käytössä leimikko nimityksen rinnalla kulkee termi ”hakkuualue”.

Urakoitsijalla metsäkonevalmistajan termein on ”laivasto”, joka käsittää kaikki organisaatiolle kuuluvat metsänhoitokoneet. Laivastoon voi kuulua useampi koneketju, jolla määritellään puun metsästä pinoon saattamiseen tarvittavat koneet. Koneketjut eroavat katkontamenetelmästä riippuen. Kuvassa 1 koneet voidaan jakaa kokorunko- (engl. full tree, FT) ja katkotun puutavaralajin (engl. Cut-to-length, CTL) koneisiin, joista yleisessä käytössä Euroopassa on jälkimmäinen. Kokonaisten puiden kaataminen on tavallista Pohjois- ja Etelä-Amerikassa. Amerikassa puut kaadetaan kokonaisina ja kuljetetaan oksineen karsinta- ja pinoamispaikoille. Puu kulkee jokaisen koneketjun koneen kautta. Jokaiseen toimintoon käytettäessä eri tavalla varusteltua konetta, puu on saattanut käydä jopa 4 eri koneen käsittelyssä, ennen kuin kuorma-auto voi noutaa sen tien vierestä. Tutkielmassa käsitellään lähinnä katkotun puutavaralajin koneita, jotka jaetaan hakkuukoneisiin ja kuormakoneisiin. Tällä kahden koneen koneketjulla saavutetaan sama lopputulos, kun Amerikan toimintamallilla. [Forest.fi, 2023]

2.2 Standardit

CTL-työmaalle tuodaan ensin hakkuukone, joka aloittaa leimikon työstämisen metsäyhtiöltä saaduilla katkontaohjetiedoilla. Hakkuukoneen katkontaohjeen tiedot voidaan jakaa kolmeen osaan: leimikon-, puulajien- tai puutavaralajien tiedot. Leimikon tietoihin sisältyy muun muassa prosessiin osallistuvien eri tahojen yhteystietoja ja leimikon ja pinojen sijainti. Puulajit ovat Suomessa ja Ruotsissa standardoitu. Kaikissa maissa standardeoituja puulajeja ei ole käytettävissä, joten niiden määritelmät vaihtelevat. Puut katkotaan metsässä valmiiksi tiettyyn mittaan, joten ohjeiden tärkein elementti on eri puutavaralajeilta vaaditut mitat ja määrät. Katkontaohjeet ovat yksi osa metsäkonealan kansainvälisesti yhtenäistä StanForD-standardia.

Yrityksillä on tarve ja velvollisuus tarkkailla omaa puuntuotantoa sekä välittää informaatiota eteenpäin. StanForD oli alun perin kehitetty kommunikointiin metsäkoneiden välillä. Nykyään StanForD:ia käytetään liikekumppaneiden välisessä tiedon välityksessä. Puunkorjuutyö tuottaa metsäkoneilla luotua dataa, jossa on määritelty puiden mitat, koneen kulkema matka, polttoaineen kulutus ja muuta tarpeellista informaatiota hakkuutyöstä. Standardia käytettiin muuhunkin. Hakkuudatan avulla voitiin ilmoittaa tavarantajalle puutavaranmäärä ja sen perusteella veloittaa työstä. Koneen tuotantodatassa todettiin siis olevan tarpeita koko metsäalan laajuiseen raportointiin. [Arlinger et al., 2008] Data muodostuu metsässä ja heikkojen tai olemattomien verkkoyhteyksien vuoksi siirretään huonoimmassa tapauksessa manuaalisesti ulkoisilla muisteilla. Datan käsittelyä varten lähes kaikille tahoille on kehitetty omia sovelluksia tukemaan omintakeista tiedostomuotoa, joissa se saadaan helppolukuisen ja järkevöitettyyn muotoon.

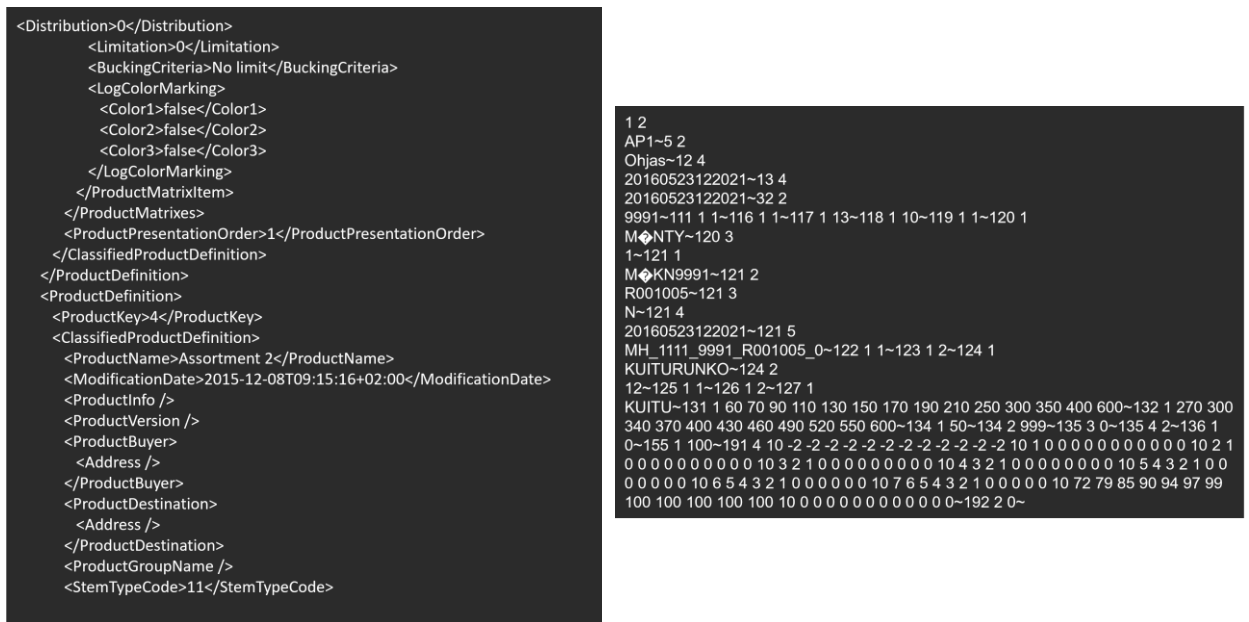


Kuva 2. StanForD-data ja sitä vaativat tahot [Räsänen & Sorsa, 2010]

StanForD-standardilla on pyritty saamaan kaikkien tahojen välille samanmuotoiset ja saman sisällön omaavat viestit. Kuvassa 2 on esiteltyä eri tahot ja niiden tarvitsema informaatio. Standardi ei ota kantaa tiedostojen käsittelyyn tai raporttien lähettämiseen. Standardin isoimpina osa-alueina on jo aiemmin mainitut leimikon aloitukseen vaaditut ohjeet sekä leimikon lopetuksen myötä muodostettavat raportit. Kaikkien etua ja näitä toimintoja tukevia toimintoja standardin piirissä on myös työajan sekä työnlaadun seuranta. Työajanseurantaan merkitään koneen eri toimintoihin kulutettu aika kuten esimerkiksi: puun sahaus, syöttö, passiivinen aika ja kuljettajan tauot. Laadun seuranta sisältää puutavaran mittauksen, kalibroinnit ja ajourien seurannan. Kaikkiin tiedostoihin tai standardin mukaan kaikkiin viesteihin on liitetty paljon samoja määrittäjiä ja tietoja, jotta viestien jäljitys takaisin niitä tuottavalle koneelle olisi helppoa varsinkin mittausvirheiden tai suorituskykyongelmien ilmetessä.

Alkuperäinen StanForD-standardi on kehitetty Ruotsissa 1987. StanForD on lyhenne sanoista ”Standard for Forest Data and Communications”, eikä se ole liitännäinen Kalifornian osavaltiossa sijaitsevaan Stanfordin alueeseen [Sorsa, 2002, s. 4]. Vanhassa standardissa tiedostoilla on tietynlainen riveittäin määritelty rakenne, eivätkä tiedostot sisällä mitään selittäviä elementtejä, ainoastaan numero- ja nimimuuttujia sekä niiden koodeja. Muuttujaryhmät voivat siis olla millä tahansa rivillä tiedostoissa [Metsäkeskus, 2023]. StanForD:in uudemman version kehittäminen aloitettiin 2000 luvun alussa, kun vanhaa

haluttiin modernisoida sen teknisen toteutuksen ollessa jo hieman vanhentunutta. On todettu, että XML:n pohjalte olisi hyvä rakentaa uutta standardia. XML on 20 vuotta aiemmin ollut käytössä laajasti useiden standardien luomisessa. [Sorsa, 2002, s. 4]



Kuva 3. StanForD hpr-tiedoston sisältöä ja vanhan standardin vastine

XML on puolirakenteista dataa, eli ei varsinaisesti rakenteista tai rakenteetonta [Snowflake, 2023]. Se on itsestään määrittävää dataa, eli rakenne määräytyy tiedostoissa lisäämällä ja sientämällä XML-elementtejä. Tämä muodostaa tietomallillisen rakenteen, jonka mukaan dokumenttirakenne pystytään kuvaamaan koko standardin kattavalla skeemalla. Skeemat ovat tutumpia relaatiotietokantamalleista, joissa luodaan toisiinsa suhteilla liitettyjä tauluja, jotka sisältävät attribuutteja. XML-skeema voidaan esittää vastaavanlaisesti, ihmisluettavana rakenteena XML-dokumenteista. [Sorsa, 2002, s. 5–8]

Kuvassa 3 vasemmalla puolella on esitetty uuden standardin muoto ja oikealla vertailun vuoksi vanha. Kuvien tiedostot sisältävät samat tiedot. Uudessa standardissa data esitellään sisennyksin ja tagein. Elementti koostuu kahdesta saman nimisestä tagista. Elementin sisältö on alku- ja lopputagin välinen informaatio, joka voi sisältää rajattoman määrän eri elementtejä. Määritelmä on yksi elementti, kuten kuvassa 3 puutavaralajin määritelmä (ProductDefinition). Määritelmät sisältävät StanForD:in vaatiman informaation kyseisestä tietojoukosta, tässä tapauksessa puutavaralajista. Lopputagi merkitään nimen jälkeisellä /-merkillä.

StanForD XML:ää suunniteltaessa haluttiin myös pitää kiinni datan eheydestä ja paikansapitävyydestä. Uuden StanForD XML:n tuoma datatyyppien rajoittamisen mahdollisuus estää käyttäjiä tekemästä virheitä, sekä tiedostot voidaan tarkistaa ohjelmallisesti. Koko standardin ylläpito ja jatkokehitys helpottuu merkittävästi, kun muuttujien lisääminen tapahtuu muokkaamalla tiedoston sisältämää StanForD-määritelmää. Määritelmiä voidaan käyttää jatkossa uusien luomiseen kirjaston tapaisesti. Skeemaa muokattaessa julkaistaan periaatteessa uusi versio standardista ja tämä tarkoittaa, että tarvitaan versionhallintajärjestelmä ja koneiden eri versioiden tukemisessa tulee ongelmia. Uuden version ilmestyessä sitä on tuettava käytössä olevilla metsäkoneilla, mutta kaikki ei ole mahdollista koneiden käyttäessä jopa vanhentuneita Windows XP-käyttöjärjestelmiä. [Sorsa, 2002, s. 5–8]

Uuden standardin käyttöönottamiseen eri tahoilla kuviteltiin sopivaksi muutaman vuoden aikaikkuna, jolloin kaikki järjestelmät pystyisivät luomaan ja hyödyntämään uuden standardin mukaista dataa [Arlinger et al., 2008]. Standardi oli määrä julkaista valmiina vuonna 2010, mutta aikamääreeseen ei päästy. Uudempi standardi on saanut kyseisestä vuosiluvusta nimensä StanForD2010.

Uuteen standardiin siirryttäessä suuremmissa yrityksissä todettiin joidenkin kenttien muokkaamiseen olevan tarvetta mahdollisten kuljettajan- tai kirjoitusvirheiden vuoksi, jotta olemassa olevissa yhtiöiden järjestelmissä data saataisiin yhtenäiseksi. Tällainen esimerkki on SF10:n (StanForD2010) kuljettajantunniste (OperatorID), joka on metsäkoneella oltava vapaasti muokattavissa oleva tekstikenttä, vaikka sitä käytetään kuljettajan yksilölliseen tunnistamiseen. Kuljettajan nimeä (OperatorName) voidaan käyttää yksilöllisenä tunnisteena järjestelmissä, joka tuo lisää haasteita. Kuljettaja luo nimellä itselleen profiilin jokaisella eri koneella. Kuljettajan työskennellessä data tallentuu profiilin taakse. Jos metsäkoneenkuljettaja näppäilee nimensä väärin, se pitää olla korjattavissa, jotta kahden koneen eriävät tunnisteet voitaisiin yhdistää samaksi tietolähteeksi.

Tekstikenttien vapaamuotoisuus aiheuttaa ongelmia logiikassa ja konepäätoteutuksissa pidemmän päälle, koska samalle kuljettajanimelle saattaa tulla useampi kuljettaja. Vaikka kyseessä olisikin sama kuljettaja, hänellä voi olla toinen profiili, jonka data ei ole yhdistettynä metsäyhtiön sovelluksessa ja se jää tarkastelun ulkopuolelle. Mikäli järjestelmässä on toteutettu kuljettajiin kohdennettua seurantaa, se häiriintyy, jos kuljettajatiedot eivät ole oikein.

Uuden standardin mukana joillekin elementeille tulee avainelementtejä, jotka ovat koneen asettamia. Esimerkiksi kuljettaja-avain (OperatorKey), jonka on koneen sisäisessä järjestelmässä oltava uniikki. Uniikkius määräytyy koneavaimen (MachineKey) mukaan. Avaimilla ratkaistaan saman nimisten kuljettajien hallinta metsäkoneiden tietokannassa. Ongelma säilyy datan siirtyessä toiseen järjestelmään, sillä avaimet ovat juoksevia lukuja ja voivat olla samat, vaikka kuljettajan tunniste on eriävä. Tarvitaan siis täysin uniikki ja organisaation laajuinen yksilöivä tunniste GUID (engl. Globally Unique Identifier). [Arlinger et al., 2020]

2.3 Pakkaaminen

XML lisää luettavuuden myötä tiedostoihin kokoa, koska niihin lisätään tageja, rivinvaihtoja ja sisennyksiä. Uudessa StanForD2010-hakkuutietoraportissa rungot eritellään tarkemmin ja tiedot jäsenellään useammalle riville. Useammassa lähteessä StanForD-kehitystyön alussa on todettu XML-tiedostojen olevan suurempia, noin kaksinkertaisia. Pakkautuvuudesta [Sorsa, 2002, s. 20–21]. Metsäkoneilla luodaan normaalin työpäivän aikana tuhansia rivejä dataa ja siitä koostetaan useita tiedostoja, jotka on jatkokäsiteltävä. Jatkokäsittelyn ja lähettämisen kannalta tiedostojen tulee olla tallennettuina mahdollisimman pieninä. Suuri tiedoston koko ja määrä aiheuttaa lisäkuluja tarvittavina resursseina. Tällaisia resursseja ovat tiedonsiirtonopeus, aika, tallennustila ja prosessointi.

Pakkaamista on kahdenlaista joko häviöllistä tai häviötöntä. Häviöllisessä pakkaamisessa pakkaaminen perustuu turhan tiedon poistamiseen tiedostosta. Metsäkoneen tuottamatiieto on kokonaisuudessaan liiketoiminnan kannalta tärkeää, joten häviöllinen pakkaaminen ei ole hyväksyttävää. Hyväksi todettuja häviöttömiä pakkaustapoja on muutamia. Tiedostojen pakkautuvuuteen vaikuttaa niiden sisältämä turha toisto eli redundanttisuus. Yleisesti käytettävät häviöttömät pakkaustavat ovat ZIP ja LZW (engl. Lempel–Ziv–Welch compression).

LZW-metodia käytetään esimerkiksi pdf-tiedostojen muodostamiseen ja useamman koodikielen kirjastoissa ”compress”-metodissa. Metodilla luodaan sanalista, joka sisältää tiedoston jokaisen sanan numeroituna indekseihin. Indeksit toimii sanalle osoitettuna viiteavaimena. [Badshah et al., 2016] Pakkauminen tapahtuu löydettyä tiedostosta sama sana useampaan kertaan. Tiedostoon kirjataan sanan kohdalle sitä vastaava viiteavain.

Teksti voidaan kirjoittaa uudelleen viiteavaimilla ja siten pienentää tiedostossa käytettyjen bittien määrää. Tiedosto voidaan palauttaa alkuperäiseen muotoonsa viiteavainlistan avulla.

LZW-metodissa pakkauksen purkaminen on huonoa verrattaessa ZIP-tiedoston purkunopeuteen. Useimmissa tapauksissa purkunopeus on syynä LZW-metodin pois jäämiselle. ZIP-pakkaustapa perustuu ”deflate”-metodiin, jonka nimitys antaa osviittaa ilmojen pois päästämisestä. ZIP-pakkaustapa on johdannainen LZW-metodista ja sitä on optimoitu Hoffman coding:lla, eli siihen on lisätty binääripuu. Binääripuu toimii sanakirjan tapaan poistaen toistoa. Metodi käännettäessä vastakkaiseen suuntaan on ”inflate”. Tällä ZIP-paketti saadaan alkuperäiseen muotoon. [Deutsch, 1996]

Vanhassa standardissa on ennakkoon luotu avainsanahakemisto, josta tiedoston koodiyhdistelmiin saadaan sen oikea sisältö samaan tapaan kuin LZW:ssä. Vanhassa standardissa käsitellään käytännössä pakattua tiedostoa, joka on kooltaan pieni, mutta ihmiselle lähes mahdoton lukea. ”Deflate”-metodi soveltuu hyvin XML-tiedostoihin. XML-tiedoston elementissä on sama tagi kahdesti, mutta elementin muuttujat pitää taltiota erikseen. Tiedostoissa tagi saattaa toistua useamman kerran, esimerkiksi jokaisen sahatun puun kohdalla on aina samat vaaditut tiedot.

Tiedostojen koon kasvu uuden XML-muodon myötä ei ole ainoa tiedostojen hallintaan liittyvä ongelma. Versioiden hallinta on osoittautunut haasteelliseksi. Skeeman määrittelyä muutettaessa julkaistaan uusi versio StanForD-tiedostoista. Versiot merkataan kuten ohjelmistopäivityksetkin. Hiljattain version v3.6 jälkeen julkaistiin v4.0 [Skogforsk, 2023]. Version sisäisissä pienemmissä päivityksissä, esimerkiksi v3.3 ja v3.6 välillä ei ole vaihdettu muuttujien nimiä tai paikkoja, mutta niitä on lisätty. Lisäykset eivät aiheuta ongelmia, koska tietokentät voidaan jättää tyhjäksi, jos eri versioita käytetään yhtiöiden välillä ristiin. Tiedoista on tunnistettava puuttuvat osat ja niitä on käsiteltävä jatkojalostuksessa eri tavalla, sillä ne eivät vertaudu täysin keskenään. Versiosta v3.x vaihdettaessa versioon v4.x on esitelty käyttötapauksia geologiselle tiedostolle standardin piirissä ja toistaiseksi siihen sopeutuminen on epäselvää metsäkonevalmistajien keskuudessa. Uusimman v4.0 version käyttöönottoon ja sen vaatimiin muutoksiin metsäalalla menee useampia vuosia.

2.4 Tietomallit

XML on yleisesti käsitetty tiedonsiirtoformaattina. Sitä käytetään sivustojen ja applikaatioiden välisessä kommunikoinnissa. [Wiese, 2015; Banxia software, 2023] Samaan tarkoitukseen voidaan käyttää muitakin muotoja, mutta yleisimpinä sekä koneelle että ihmiselle helppolukuisina muotoina XML:n lisäksi on JSON. Kuvasta 4 huomataan JSON:in olevan rakenteellisesti lähellä XML-tiedostomuotoa, mutta sen merkintätapa eroaa tageista. JSON on XML:n tavoin puolirakenteista. JSON:lle on omat standardinsa, esimerkiksi GeoJSON. [Butler et al., 2016]. JSON muodostuu sisennetyistä objekteista ja avainarvolistoista. Tiedoston objektit on merkitty erilaisilla suluilla, eroteltu pilkuilla ja sisennetty oikeaan hierarkkiseen järjestykseen. [Wiese, 2015]

```
{
  "StemTypeDefinition": [
    {
      "StemTypeCode": 11,
      "StemTypeName": "Mäntytukki"
    },
    {
      "StemTypeCode": 12,
      "StemTypeName": "Mäntykuitu"
    },
    {
      "StemTypeCode": 13,
      "StemTypeName": "Mänty muu runkolaji"
    }
  ],
  "SpeciesGroupPresentationOrder": 1,
  "DBHHeight": 120,
  "Grades": {
    "StartGrade": 1,
    "Grade": {
      "GradeNumber": 1,
      "GradeName": "Laatu 1"
    }
  }
}
```

Kuva 4. StanForD:in hpr-tiedostoa konvertoituna JSON-muotoon

Verkossa selaimet ja palvelimet keskustelevat keskenään hypertekstin siirto-protokollalla (engl. Hypertext Transfer Protocol, HTTP). Siihen voidaan liittää kyselyitä ja tietoa, jotta käyttäjä pystyy navigoimaan seuraavalle sivulle. Viestintä on molemmin suuntaista ja lähettäjälle saa palvelimelta aina vastauksen viestin vastaanottamisesta. HTTP-kyselyihin liitetty data voi olla monenlaisessa muodossa. Isomman listan tai useampaan aiheeseen rinnastettavan tiedon siirrossa käytetään yleensä XML- tai JSON-muotoisia hyötykuormia (engl. payload). Hyötykuormien sisältö on helppo muuttaa toiseen muotoon, jos se ei sisällä liian sisennettyä dataa. Esimerkkinä hyötykuormasta HTTP-kyselyissä on verkkokauppojen tilaukset. Verkkosivulla valitaan ostettavat tuotteet ja annetaan maksupalvelulle sekä kauppiaan että asiakkaan tiedot maksusta. Nämä tiedot on sidottu hyötykuormaan HTTP-kyselyn alle. Sisennyksillä määritellään mikä tieto kuuluu kullekin taholle.

Nimi tietoa voidaan käyttää kauppiaille, asiakkaalle ja tuotteelle, mutta sisennys ja sen sijainti määrittää osapuolen.

HTTP:llä välitetty tieto voidaan käsittelyn jälkeen taltioida tietokantaan. Tietokantoja on paljon erilaisia, mutta yleisimpiä niistä ovat pitkään olleet relaatiotietokannat, joiden kyselykielenä on SQL. Toinen tietokantojen jaottelu kategoria on NoSQL-tietokannat. Nämä tietokannat eivät pohjautu SQL-kieleen. NoSQL-tietokannoilla on usein oma tarkoitusta varten luotu kyselykieli. Ne eivät ole relaatiotietokantoja, joka mahdollistaa erilaiset datarajapintojen sovellutukset.

Perinteinen relaatiotietokanta rakennetaan datan mukaan ja jäsennellään haluttuihin ”tauluihin” jotka viittaavat toisiinsa viiteavaimilla. Tiedot ovat rinnastettavissa, eli relaatiossa toisiinsa. Luotaessa relaatiotietokantaa ollaan riippuvaisia datan muodosta ja tauluihin varastoitava data on oltava ennalta määritellyssä muodossa. Taulut tukevat saman datan taltioimista useampaan kertaan, mutta se ei ole optimaalista. SQL-kyselykielin haettava tieto on tarkoin strukturoitu ja kyselykielen yksi toiminnan edellytys. Rakenteellisen datan vuoksi relaatiotietokannat skaalautuvat vertikaalisesti. Vertikaalinen skaalaaminen tarkoittaa yhden palvelimen resurssien kasvattamista. [Chao, 2013]

NoSQL-tietokantoja on useita erilaisia ja niille on toteutettu omat toimintatavat ja kyselykielet täysin riippuen valitusta alustasta. Esimerkkeinä NoSQL-tietokannoista: avain-arvopari-tietokanta, dokumenttitietokanta, graafitietokanta ja aikasarjatietokanta. [Alam et al., 2013; Sullivan, 2015]

Avain-arvopari-tietokannassa taltioidaan viiteavain ja sen viittaama tieto. Avaimen taakse talletettu data on tietokoneelle nopeasti löydettävissä. Tietojoukossijainti löytyy nopeasti, mutta sen arvoon ei päästä suoraan kiinni ennen käsittelyä. Avain-arvopari-tietokannalle ei ole standardoitua kyselykieltä ja avaimien hallinta menee monimutkaisella datalla haastavaksi. Arvojen sisältöä ei ole ennalta määritelty, joten se soveltuu kaikille tietojoukoille. Avain-arvoparit eivät ole relaatiossa toisen samassa ympäristössä olevan tietokannan tietojoukkoihin. Relaation puuttuminen mahdollistaa horisontaalisen skaalaamisen. Horisontaalinen skaalaaminen tarkoittaa tietokannan hajauttamista useammalle palvelimelle. [Alam et al., 2013; Sullivan, 2015]

Dokumenttitietokannan taustalla toimii avain-arvopari-tietokanta, mutta se eroaa toiminnaltaan datankäsittelyssä. Dokumentti sisältää taulukon, joka halutaan tallentaa. Taulukko pilkotaan avain-arvopareiksi isompina kokonaisuuksina ja taltioidaan sen jälkeen

tietokantaan. Dokumenttitietokannassa on erona avain-arvoparitietokantaan koko dokumentin tai sen suurempien osien taltioiminen. Rakenteen määrittelee itse dokumenttien rakenne. Tietokantaan voidaan taltioida XML- ja JSON-tyyppisiä tiedostoja ja tietokannasta kysely onnistuu niiden puolirakenteisuuden ansiosta. Toiminta hidastuu verrattuna normaaliin avain-arvoparitietokantaan, mutta samalla mahdollistaa rikkaamman datan taltioimisen. [Alam et al., 2013; Sullivan, 2015]

Graafitietokanta on dokumenttitietokannan tapainen ratkaisu, mutta toteutuksessa on lisäksi solmukohtien (engl. node) väliset suhteet. Tiedostot voivat toimia solmukohtina ja ne ovat linkitettyinä toisiinsa. Linkit eli yhteydet tallennetaan metadatanä, eli liitännäisenä ja kuvaavana tietona. Linkit sisältävät tietoja kuten nimen ja suunnan. Tietokannan moottorista riippuen, linkkien tiedot tallennetaan relaatio- tai avain-arvotietokantoihin. Tietokantamoottoreista esimerkkinä yleisesti tunnetuimpia ovat: MySQL, PostgreSQL ja MongoDB. Graafitietokannoilla ei ole omaa vakiintunutta hakukieltä. SQL- ja yksinkertaiset NoSQL-tietokannat ovat kehittäjille tutumpia, joka tarkoittaa nopeampaa kehitystä ja ongelmien ilmetessä nopeampaa virheiden korjausta. [Sullivan, 2015]

Aikasarjatietokanta on ollut esineiden internetissä (engl. Internet of Things, IoT) mullistuksellinen. Aikasarjatietokannat organisoivat tietoa, joka perustuu aikamäärein määriteltyjen mittauksien arvoihin. [Sychev et al., 2020] Tietokannat perustuvat avain-arvopareihin, joissa avaimena toimii aikaleima ja arvona on mittarin arvo. Oleellista aikasarjatietokannoille on, että tiedon mukana on aina oltava aikaleima. Tietokannassa mittarindataa ei voi muokata jälkikäteen. Aikajana toimii kuin jono, jonka toisesta päästä poistetaan dataa, kun se vanhenee ja toisesta päästä syötetään lisää. Aikajanelle on oltava mahdollista taltioida vanhempaakin dataa, jos yhteysongelmien vuoksi tietoa ei ole ollut mahdollista saada reaaliaikaisesti tai tietyn iteraation. Koska datalla on tietty piste, jossa se on luotu, sitä on helppo tarkastella ja siitä on helppo piirtää diagrammeja. Sen pohjalta voidaan tarkastella mittauksien kehitystä. Tietokannoissa on usein mahdollista asettaa jokin metriikka tarkastelun kohteeksi ja arvon ylittäessä asetetun rajan, järjestelmä lähettää hälytyksen.

Nämä kaikki yllä luetellut ovat kehittyneet tehokkaammiksi pilvipalvelujen tullessa yleiseen käyttöön lähivuosien aikana. Kaikki ylimääräinen laskenta-aika maksaa käyttäjälle ja siksi ne on pyritty optimoimaan työkaluina ja sovelletuissa ympäristöissä. Innovaatiot ja muutokset ovat vaikuttaneet tapoihin käsitellä esimerkiksi avain-arvoparitietokannan avaimia.

2.5 Tiedon käsittely

XML-tiedoston muuttaminen muuhun muotoon on haasteellista, koska tiedot ovat relaatiotissa toisiinsa ja ylimääräistä tekstiä on joka elementissä. Tieto voidaan erottaa elementeistä tietokantoihin tai muuttaa erittäin yleiseen JSON-muotoon. XML:llä on oma merkintäkieli XSLT (the eXtensible Stylesheet Language), jolla tiedostot voidaan muuttaa haluttuun muotoon. Kuvassa 5 on kaksoispisteiden jälkeen XSLT:n ennalta määriteltyjä toimintoja, joilla saadaan tieto pilkottua elementeittäin. XSLT luotiin alun perin muuttamaan XML-tiedostojen ulkomuotoa HTML-, CSV- tai muihin tiedostomuotoihin. XSLT-koodilla voidaan vastaanottaa yksi tai useampi XML-tiedosto ja muuttaa elementeistä yksi tai useampi tiedosto. Normaaleihin koodikieliin verrattaen XSLT ei muuta arvoja tai ohjelman tilaa, vaan kertoo mitä kullekin määritellylle elementille tulisi tehdä. XSD eli XML-skeemamääritelmä (engl. The XML Schema Definition) on XML-dokumentti, jolla voidaan määrittellä XML-tiedostojen sisältö sekä tarkistaa niiden oikeellisuus [W3C, 2001]. Skeemassa on mahdollista määrittellä esimerkiksi dokumentin rakenne ja arvojen tyypit. Skeema mahdollistaa XML-jäsentelytyökalujen käytön ja ilman sitä ei voida olla varmoja XML-tiedoston rakenteesta. [Wiese, 2015; W3C, 2017a]

```
<!--match all and apply correct templates-->
<xsl:template match="/">

  <xsl:apply-templates select="*/*/x:ObjectDefinition"/>

  <xsl:apply-templates select="*/*/x:SpeciesGroupDefinition"/>

  <xsl:apply-templates select="*/*/x:ProductDefinition"/>

  <xsl:apply-templates select="*/*/x:DeliveryDefinition"/>

  <xsl:apply-templates select="*/*/x:LocationDefinition"/>

  <xsl:apply-templates select="*/*/x:ScaleDefinition"/>

  --/--
  [
  <xsl:apply-templates select="*/*/x:Stem" mode="stem"/>

  <xsl:apply-templates select="*/*/*/x:Log" mode="log"/>

  <xsl:apply-templates select="*/*/x:Load" mode="load"/>


```

```
<xsl:template match="x:DeliveryDefinition">
  INSERT INTO sf_delivery VALUES (
    '<xsl:value-of select="x:DeliveryUserID"/>',
    '<xsl:value-of select="x:ModificationDate"/>',
    '<DeliveryDefinition>
      <xsl:apply-templates mode="sql"/>
    </DeliveryDefinition>'
  ) ON CONFLICT DO NOTHING;

  INSERT INTO machine_has_delivery VALUES (
    '<xsl:value-of select="x:DeliveryUserID"/>',
    '<xsl:value-of select="x:ModificationDate"/>',
    '<xsl:value-of select="../x:MachineKey"/>',
    '<xsl:value-of select="x:DeliveryKey"/>'
  ) ON CONFLICT DO NOTHING;
</xsl:template>
```

Kuva 5. hpr-tiedoston jäsentely XSLT:llä tietokantaan

XSLT voi muuttaa elementtejä ja niiden löytämiseen se käyttää ei-XML-pohjaista XPath-kieltä. XPath eli XML-polkukielen (engl. XML Path Language) yleisimpiä lausekkeita

ovat polkulausekkeet. Lausekkeet ovat samalaisia kuin Windowsin kansiorakenteissakin, mutta kansioiden sijaan navigoidaan XML elementtien ja niiden sisennyksien mukaan. Jokainen ”/”-merkki tarkoittaa yhtä sisennystä ja kirjainjonot niiden välissä ovat elementtien nimiä. [Wiese, 2015; W3C, 2017b]

XML-kyselykieli (engl. XML Query, XQuery) on funktionaalinen kieli, jolla informaatio erotellaan elementeistä. Kielessä on muuttujia, funktioita, silmukoita ja ehtolausekkeitä. Kielen avulla voidaan hakea tietoa suuresta määrästä tiedostoja SQL-tietokannan tapaan. XQuery pohjautuu XPath:iin, joka erittäin paljon sisennetyissä XML-tiedostoissa vaikuttaa suorituskykyyn negatiivisesti. [Wiese, 2015; W3C, 2017c]

3 Datan ympäristö

Luvussa esitellään pilvipalveluiden pääperiaatteita ja eri arkkitehtuurillisia tasoja. Reuna- ja sumulaskenta ovat pilvilaskennan alitasoja, joilla suoriutumista voidaan optimoida. Lopuksi käsitellään huomioon otettavia tietoturvaohjeita ja käyttöoikeustasoa eri järjestelmissä.

3.1 Pilvipalvelut

Totuttu tapa nettisivujen rakentamiseen on pitkään ollut joko yrityksen itse omistama ja hallinnoima palvelin tai vuokrattu palvelin verkkopalveluntarjoajalta. Tämän kaiken on pitkälti mullistanut ja nyt jo vakiintunut pilvikehittäminen. Vanhoja menetelmiä käytetään edelleen pienemmissä yrityksissä, joissa pilviympäristö aiheuttaa ainoastaan lisävai-
vaa.

Pilvipalvelut alkoivat yleistyä noin 2010-luvulla, hieman ensimmäisten palveluiden julkaisun jälkeen. Aluksi palvelut olivat yksinkertaisia prosessointitehoa tarjoavia palvelimia, mutta kehittyivät kuitenkin verkkokehitystä paremmin tukeviksi alustoiksi.

Arkisissakin keskusteluissa kuulee usein puhuttavan pilvestä, pilvilaskennasta ja pilvipalveluista. Nämä voidaan selittää yksinkertaisesti ajatellen, että dataa tallennetaan pilveen ja siellä sille suoritetaan pilvilaskentaa eri pilvipalveluilla tai palveluissa.

Tietotekniikan alalla yleisimmät IaaS, PaaS ja SaaS (infrastructure-, platform- and software as a service) lyhenteet kertovat eri tasoista, joihin palvelut voidaan jaotella sen tarjonnan perusteella. Infrastruktuuripalvelut (IaaS) tarjoavat pilvessä pääsyä fyysisiin ja virtuaalisiin resursseihin, kuten palvelimiin ja tallennustilaan. Alustapalveluna (PaaS) tarkoittaa esimerkiksi valmiiksi asennettua virtuaalikonetta, jossa on määriteltynä käyttöjärjestelmä ja muistin määrä. Valmiiksi asennetulla virtuaalikoneella sovelluksen kehittämisen voi aloittaa lähes välittömästi. Ohjelmistopalvelulla (SaaS) tarjotaan ohjelmistoja. Niistä esimerkkinä pilvipalvelimille asennettavat ERP-järjestelmät. ERP- ja verkko-kauppasovelluksen asennusvaiheessa käyttäjän on mahdollista valita ja laskea kustanteet vaadittavan laskentatehon mukaan. Asiakkaan ottaessa käyttöönsä SaaS-tasoisien palvelun, sen taustalla on aina IaaS- ja PaaS-tasoiset palvelut valmiiksi. Käyttäjän on tarkoitus tehdä muutoksia lähinnä vain ohjelmistoon, jonka sai asennuksen yhteydessä. Yrityksillä on usein hallinnassaan useampi taso. Pilvialustoilla ratkaisuihin on mahdollista vaikuttaa jälkikäteen. [Chao, 2013]

Pilvipalveluiden tärkeimpiä kulmakiviä ovat helppo pääsy kaikkialta, loputon skaalautuvuus, uusien innovaatioiden luominen nopeasti ja kriisitilanteista toipuminen [Chao, 2013]. Pilvipalveluissa voidaan käyttää kaikkea mahdollista dataa ja uusien verkkosovellusversioiden luominen on saumatonta. Kun uusi ominaisuus julkaistaan, se on kaikilla käyttäjillä toimintavalmiina heti sivun päivittyessä. Loputon skaalautuvuus ja sen rajoitusten poistaminen voi olla hintavaa. [Alapati et al., 2019]

Tavanomaiset palvelimet ovat saatavilla internetin kautta mistä tahansa sijainnista, mutta otettaessa yhteys toiselle puolelle maailmaa, yhteydessä ilmenee latenssia ja käyttäjäkokemus kärsii. Pilvipalveluidentarjoajat omistavat useampia datakeskuksia ympäri maailmaa. Useampaan sijaintiin hajautetulla järjestelmällä päästään lähemmäs loppukäyttäjää ja lievennetään ongelmia. Datakeskuksissa on eroja teknologian tasossa, määrässä ja hinnassa. Palveluiden sijainneilla voidaan keinotella halvempaa hintaa, mutta samalla joudutaan tinkimään viiveestä tai teknologioiden yleisestä tasosta. Palveluiden ripottelu useampaan sijaintiin ei ole huono ajatus loppukäyttäjien tyytyväisyyden ja kriisitilanteista toipumisen suhteen.

Maanjäristyksien tai muiden luonnonilmiöiden sattuessa datakeskuksen kohdalle on hyvä luoda palautumissuunnitelma. Palautumissuunnitelmaan määritellään toiminnot datan tai laitteiston vaurioitumisen varalta. Suunnitelmassa keskeistä on datan palauttaminen ja palvelimien ylös ajaminen. Järjestelmiä luotaessa kehittäjiä on suunniteltava työpohjat, eli templaatit. Templaatteja käytetään pilvipalvelujen arkkitehtuurin ja infrastruktuurin yhtenäistämiseen ja pystyttämiseen uusissa pilviympäristöissä. Infrastruktuuri luodaan yaml-merkintäkielisinä konfigurointitiedostoilla (engl. YAML Ain't Markup Language), jotka ovat tekstipohjaisia. Määrittelytiedostoissa voidaan määrittää instanssille käyttöjärjestelmä, järjestelmän kirjastot, riippuvuudet ja asetukset.

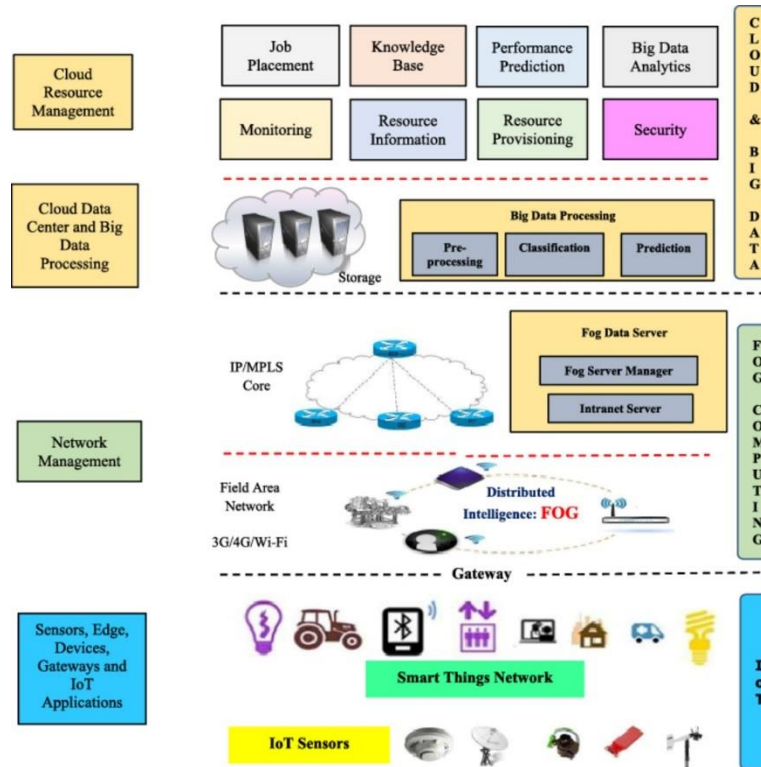
Pilvi-instanssilla tarkoitetaan yksittäistä fyysisellä tai virtuaalisella palvelimella suoritettavaa järjestelmää. AWS-alustalla instanssi on virtuaalikone (engl. virtual machine) EC2, johon määrittelytiedostoilla määritellään resurssit kuten prosessorin teho ja muistien määrät. Virtuaalikone on ohjelmallisesti toteutettu tietokone, jota voidaan käyttää oikean koneen tavoin. Aidon tietokoneen tai palvelimen resurssit voidaan jakaa virtualisoinnin avulla useiksi virtuaalikoneiksi. Instansseista voidaan luoda niin sanottuja järjestelmäkuvia (engl. system image) tiedosto muotoon, jolla kyseisen koneen asetukset ja tila voi-

daan monistaa helposti, jos toimintaa haluttaan laajentaa uuteen ympäristöön. Myös tietokannoille voidaan luoda saman tapaisia kuvia (engl. snapshot), mutta kummassakaan tapauksessa kuvalla ei voida palauttaa täydellistä kopiota instanssista. [Heller, 2021]

Uusimpia teknologioita pilvikehityksessä on lambda-funktiot. Lambda funktioita sanotaan palvelimettomiksi (engl. serverless) palveluiksi. Ne ovat funktioita palveluina (FaaS), joiden sisäinen koodi ajetaan tarvittaessa. Herätessään (engl. invoke) lambda-funktio varaa resursseja ja skaalaa niitä tarpeen mukaan. Haittapuolena tällä on heräämisen hitaus. Heräämisen hitaus haittaa, kun funktiota kutsutaan harvoin ja herääminen vie suurimman osan ajasta. Hitaus ja instanssin turhan suureksi skaalautuminen maksavat huomattavan summan, joten lambdan käyttäminen ei ole kaikissa tapauksissa järkevää. ”Serverless” ei tarkoita täysin palvelimetonta, vaan ennemminkin ”less-server” eli tarpeen mukaan käytettävää toimintoa. Esimerkki hyvästä käytöstä on datan pureskelu ymmärrettävämpään muotoon, kun toimintoon on jonossa muutamia tiedostoja. Metsäkonedatan tapauksessa toiminto voidaan herättää aina kun pilveen saadaan uusi XML-tiedosto ja se halutaan johonkin muuhun muotoon. Konkreettisesti lambda funktiolla voidaan ajaa XSLT-koodia python moottorilla. [Paul, 2023; Eismann et al., 2021]

3.2 Reuna- ja sumulaskenta

Ennen pilvilaskentaa, datalle voidaan tehdä operaatioita matkalla pilveen. Esimerkkejä tasoista, joilla operaatioita voidaan suorittaa, on kuvassa 6. Laskentaa voidaan suorittaa lähempänä käyttäjää. Käyttäjän päätelaitteessa tehtyä datan käsittelyä kutsutaan reunalaskennaksi (engl. edge computing). Reunalaskennalla pyritään vähentämään kuormitusta verkoston myöhemmissä vaiheissa ja datakeskuksissa. Reunalaskentaa voidaan haluta suorittaa myös tapauksissa, joissa verkkoyhteys on huono ja metsäkoneiden tapauksessa se usein on. Koneelta voidaan haluta valmistajan tai kuljettajan takia tietoa muusakin kuin StanForD-muodossa. Aggregoimalla voidaan vähentää tiedon kohtuutonta määrää. Aggregointi on aikavälille laskettujen keskiarvojen muodostamista. [Yoo et al., 2021]



Kuva 6. Eri laskentatyyppien hierarkia [Gill et al., 2019]

Liiketoiminnalle kriittistä tietoa on polttoaineen kulutus ja metsänhoitotyön eri osa-alueisiin kulunut aika. Kärjistettynä mielivaltaisena esimerkkinä: 8 tunnin työvuoron aikana polttoainetta on kulunut 200 litraa ja mittauspisteitä on otettu jokaista minuuttia kohden 5. Työvuoron aikana pelkkää polttoaineen kulutusta on mitattu 2400 kertaa ja se ilmoitettakoon litroina kahden desimaalin tarkkuudella määriteltynä aikaleimoin. Dataa kertyy useita kilotavuja yhdessä päivässä pelkän polttoaineen kulutuksen mittauksista. Jotta jokaista mittausa ei tarvitsisi välittää sellaisenaan, mittauksista voidaan laskea keskekulutus esimerkiksi kymmenen minuutin välein. Data pakataan ja välitetään eteenpäin. Pakkaaminen ei vaikuta huomattavasti kyseisessä mittausintensiivisessä käytössä.

Päätelaiteella summattu data voidaan välittää välissä oleviin sumusolmuihin (engl. fog-node). Tällaisia solmukohtia ovat kaikki laitteet kuten palvelimet ja reitittimet, joiden kautta tieto kulkee ja joissa datalle suoritetaan laskentaa. Sumulaskennan etuina on tarpeettoman datan siivoaminen ja aggregointi. Sumusolmuissa tietoa voidaan pitää väliaikaisvarastossa, datan perusteella voidaan vastata kyselyihin ja datasta voidaan poistaa tarpeeton informaatio. Hyötynä sumusolmuissa on isomman alueen kattava tietojoukon kerääminen ja kyselyihin vastaaminen pienellä viiveellä. Pilven kautta käytettynä vastauksen saaminen veisi pidempään.

3.3 Luottamus ja turvallisuus

Jos kaikkea reuna-, sumu- ja pilvilaskennassa osallisena olevaa dataa käytetään yhdessä, voidaan sanoa, että tietoa on valtavasti useammasta lähteestä, eikä sitä voida käsitellä tavanomaisin keinoin. Lähteessä [Alam et al., 2013] bigdataa on kuvailtu kolmella sanalla: määrä, moninaisuus ja nopeus. Nämä ovat useimpien lähteiden mukaan edellytyksiä bigdata:lle. NoSQL-tietokannat soveltuvat tämän bigdatan taltiointiin, koska niihin on mahdollista varastoida rakenteetonta, sekä puolirakenteista dataa dynaamisesti [Alam et al., 2013]. Bigdata on määritelty epätarkasti ja sen määrittely elää edelleen, sillä se mikä on bigdataa nyt, ei välttämättä ole sitä enää 10 vuoden päästä. Tulevaisuudessa nykyisiä datamääriä voidaan käsitellä tavanomaisin keinoin teknologioiden kehittyessä. Sama ilmiö on nähtävissä joidenkin salausmenetelmien käytössä.

Jotkin salausmenetelmät menettävät suojaansa laskentatehon lisääntyessä ja teoreettisesti niiden suojaus on liian matala muihin uusiin salausmenetelmiin verrattaen. RSA-SHA-1 avaimet omaavat liian matalan suojaustason GitHubin mukaan, sillä se on estänyt kyseisten SHA-1 salauksien käyttämisen avaimille. Nämä seikat on huomioitava käsiteltäessä metsäkone dataa. [Cooper & Carlson, 2021]

Datan tulisi sisältää mahdollisimman vähän asiakkaiden henkilökohtaisia tietoja. Välttämättömien tietojen rajausta ja ympäristö on määriteltävä mahdollisimman hyvin. EU:n tietosuojalaki (engl. General Data Protection Regulation, GDPR) velvoittaa lakiin nojaten käsittelemään henkilötietoja tietyllä tavalla ja sen seuraamatta jättämisestä voi tulla sanktioita. [Your Europe, 2022; EU, 2016]

Metsäkoneet on rekisteröity tietyille yrityksille ja yritykset työskentelevät työsopimuksilla metsänomistajille. Yksi yritys ei välttämättä tee kaikkea työtä ja leimikoiden tila, sijainti ja edistyminen on leimikolla samaan aikaan työskenteleville kriittistä tietoa. Tärkeimpiä huomioitavia on henkilö-, sekä tuotantotiedon rajoittaminen. Mitään tietoa ei pitäisi jakaa, ellei sille koeta olevan tarvetta. Henkilöiden pääsyoikeuksia rajoitetaan eri sovelluksissa, kuten metsäkoneella, pilvipalvelussa toimivassa työmaanhallinnassa sekä organisaation hallintaan liittyvässä verkkosovelluksessa. Edellä mainittuihin järjestelmiin kirjaututtaessa vaaditaan profiili, jonka kirjautumistunnukset pidetään vain omana tietona.

Alimmalla käyttäjiä rajoittavalla tasolla hakkuuorganisaatioissa on metsäkone. Metsäkoneelle kirjataan käyttäjiä ja niiden pääsyä eri toimintoihin metsäkoneen sisällä voidaan

rajoittaa. Kuljettajille asetetaan rajoitteet usein työkokemuksen mukaan. Osaavampi kuljettaja pääsee käsiksi useampaan ominaisuuteen. Aloittelijan on turvallista ja jopa tehokkaampaa käyttää hitaampaa puomin nopeutta, jotta hän ei tee liikaa korjausliikkeitä tai vahingoita ympäristöään käyttäessään konetta.

Kuljettajista pitää standardin mukaan olla tietty määrä tietoa, jotta voidaan luoda StanForD:in mukaisia raportteja. Henkilökohtaisten tietojen kerääminen on minimoitava, jotta ei syntyisi henkilötietorekisteriä. Organisaatio käyttää työmaiden hallintaan Timber Manager-nimistä pilvitoteutusta. Sen avulla on mahdollista luoda leimikoita ja asettaa niille hakkuualueiden rajat, merkinnät ja muuta hakkuutyölle oleellista tietoa. Koneen tekemä työ nähdään reaaliajassa leimikolla, mikäli yhteydet ovat kunnossa. Leimikko voidaan jakaa usealle organisaatiolle, jotka tässä tapauksessa ovat hakkuuyrityksiä. Pienemmät hakkuuyritykset tekevät usein tällaista yhteistyötä esimerkiksi puutteellisen kaluston vuoksi. Työmaalle saattaa liittyä muitakin tahoja, kuten puutavara-, polttoaine- ja lavettirekkoja. Rekoille oleellinen tieto on tavaran sijainti ja määrä. Leimikoihin voidaan antaa rajatut tai täydet käyttöoikeudet. Rajatuilla oikeuksilla annetaan lupa seurata tuotannon etenemistä ja luoda karttamerkintöjä. Täysillä oikeuksilla lisätään oikeus leimikon tietojen muokkaamiseen sekä datan siirtoon ja arkistointiin.

Kolmas taso tiedon jakamisessa on MyJohnDeere-sivusto (MyJD), jolla hallitaan muun muassa koneiden lisenssejä, anturidataa, tiedostoja ja organisaation henkilöstöä. Henkilöstölle voidaan asettaa käyttöoikeuksia viiteen eri kategoriaan. Laitteisto, organisaatio, sijainnit, työ ja talous. Kategorioilla on vielä neljä tasoa, joista 0 on pienin ja 3 suurin. Taso 1 antaa lukuoikeuden ja kaksi seuraavaa koskevat tietojen muokkausta. Nämä 5 kategoriaa vaikuttavat myös Timber Manager-käyttäjän käyttöoikeuksiin.

Yhtenä vaaratekijänä voi olla sähköpostein lähetettävät tiedostot. Väliintulohyökkäyksen (engl. man-in-the-middle attack) sattuessa kohdalle järjestelmän turvallisuus voi vaarantua. Väliintulohyökkäyksessä viesti kaapataan ennen kuin se on kerennyt perille, sitä mahdollisesti muokataan ja loppukäyttäjälle lähetetään muokattu sisältö.

Uuden StanForD2010-standardin mukana haluttiin järjestelmällisyyttä, mutta samalla otettiin käyttöön uusi muoto esittää vapaamuotoisempaa tietoa tiedostojen sisällä, joka ei varsinaisesti ole osa standardia, eikä jäsennellyä. Extension-niminen XML-elementti on standardissa määritelty kenttä, mutta voi sisältää täysin vapaamuotoista dataa. [Arlinger et al., 2020, s.66] Extension-elementti on yksi syy, jonka vuoksi jokaisen sovelluksen

kohdalla on kehitysvaiheessa tekstisisältö sanitoitava ja huomioitava, ettei sisältöä missään kohtaa suoriteta. Sanitointi on tekstisisällön muokkaamista käsittelyvaiheessa poistaen ei-toivotut merkinnät. XML-tiedostojen sisältö tulisi lukea tietokantoihin ainoastaan tekstimuodossa.

4 Ohjelmistoprojekti John Deere Forestry Oy:llä

Tässä luvussa käydään läpi John Deere Forestry Oy:n pilviympäristön ohjelmistoprojektin teknologioiden valinta- ja kehitysprosessia. Luvussa esitellään projektissa kohdattuja ongelmia, vaatimuksia, datanlähteitä ja valittuja ratkaisuja. Ratkaisuilla on pyritty vastaamaan käyttötarkoituksiin mahdollisimman kattavasti. Pilviympäristön kehitystyössä on huomioitava monihenkisen tiimin käyttämät työkalut, jotta työskentely olisi sujuvaa.

4.1 Datan lähteet

Projekti on lähtenyt liikkeelle parantaakseen käyttämisen helppoutta ja statistisen datan saatavuutta. StanForD-data sisältää mittauksia ja avainlukuja, mutta sen sisältö ei käy yhdellä vilkaisulla ilmi. Tiedostojen lisäksi on otettava huomioon myös standardiin rinnastettava informaatio, jolla mahdollisesta tulevaisuuden ominaisuuksia pilvipalveluissa.

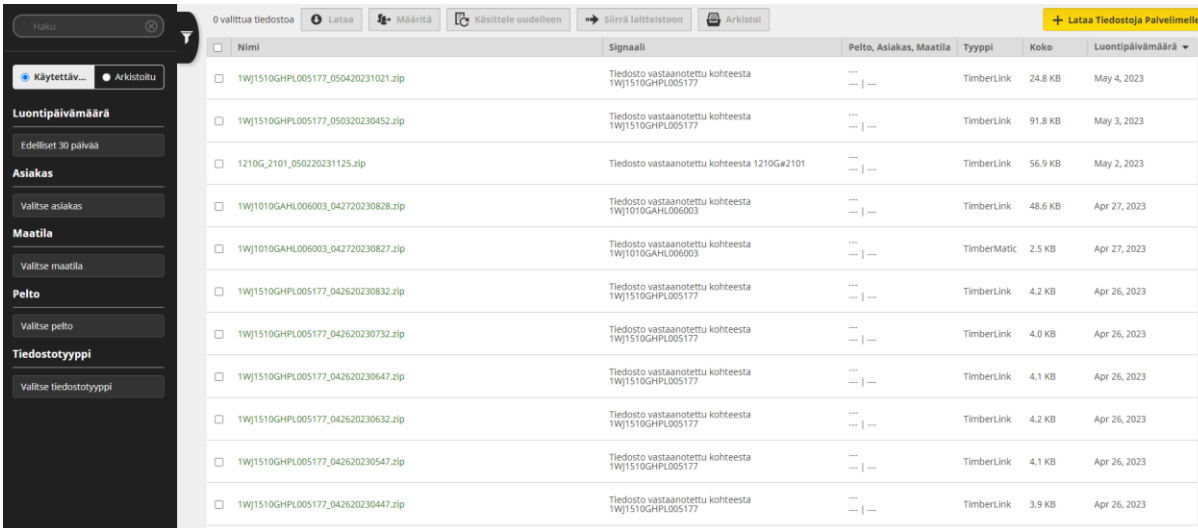
4.1.1 Vanhat olemassa olevat järjestelmät

Perimmäinen ajatus on ollut TimberOffice 5 (TO5) -työpöytäsovelluksen ominaisuuksien tuominen pilvipalveluun, josta ne ovat saatavilla yhä suuremmalle yleisölle. Huonoina puolina TimberOfficessa on pitkään ollut vaikealukuisuus, työpöytäsovelluksen paikallinen tietokanta sekä sen vanhahkon toteutuksen muuttaminen. TO5 saa tiedostot oikein asetettuna suoraan metsäkoneelta automaattisesti. Tiedostot voidaan tarvittaessa ladata sovellukseen manuaalisesti. Automaattinen lataaminen aiheuttaa sovelluksen tietokannan paisumisen huomaamatta liian suureksi. Tietokannan käsitteleminen jokaisella käynnistyskerralla vie paljon aikaa, eikä käyttäjä ole täysin varma käytössä olevista tiedostoista. TO5:ssa on mahdollista asettaa haluttu aikaintervalli, mutta hitaammalla tietokoneella diagrammien muodostamiseen kuluu liikaa aikaa.

TO5 parantaa huomattavasti vanhan ja uuden StanForD:in tekstitiedostojen luettavuutta, mutta ei kykene tarjoamaan kaikkea nykyisen globaalin markkinan ja alati mullistuvan teknologian aikakaudella. Tärkeimpiä toimintoja ovat olleet kuljettajien ja koneiden suoriutumisen tarkastelu sekä ohjetiedostojen luominen. Sovellusta voi käyttää myös uuden standardin tiedostoihin. TimberOfficessa koneilta saadut tiedostot pureskellaan valmiiksi käyttäjää varten erilaisiin näkymiin koneiden, kuljettajien ja leimikoiden perusteella. Näkymissä on enemmän eksaktia dataa sekavassa teksti- ja numeromuodossa. TO5 ei tue

kaikkea dataa ja anturidatan lisääminen paikalliseen työpöytäsovellukseen uudessa versiossa on iso työ kehittäjille.

MyJD on maatalouden tarkoituksiin räätälöity alusta, jossa voi hallita metsäkoneita, mutta sen ajattelutapa pohjautuu peltojen ja muun kuin metsälaitteiston hallintaan. Koneilta lähtevät tiedostot ovat pakattuna ZIP-muotoon tiedonsiirron vuoksi. Koneen ollessa yhteydettä laitevian tai katvealueen vuoksi, tiedostoja ei lähetetä ja kerralla lähetettävä datamäärä voi kasvaa suureksi yhteyden muodostuessa uudelleen.



Nimi	Signaali	Pelto, Asiakas, Maatila	Tyyppi	Koko	Luontipäivämäärä
<input type="checkbox"/> 1WJ1510GHPL005177_050420231021.zip	Tiedosto vastaanotettu kohteesta 1WJ1510GHPL005177	---	TimberLink	24.8 KB	May 4, 2023
<input type="checkbox"/> 1WJ1510GHPL005177_050320230452.zip	Tiedosto vastaanotettu kohteesta 1WJ1510GHPL005177	---	TimberLink	91.8 KB	May 3, 2023
<input type="checkbox"/> 1210G_2101_050220231125.zip	Tiedosto vastaanotettu kohteesta 1210Gw2101	---	TimberLink	56.9 KB	May 2, 2023
<input type="checkbox"/> 1WJ1010GAHL006003_042720230828.zip	Tiedosto vastaanotettu kohteesta 1WJ1010GAHL006003	---	TimberLink	48.6 KB	Apr 27, 2023
<input type="checkbox"/> 1WJ1010GAHL006003_042720230827.zip	Tiedosto vastaanotettu kohteesta 1WJ1010GAHL006003	---	TimberMatic	2.5 KB	Apr 27, 2023
<input type="checkbox"/> 1WJ1510GHPL005177_042620230832.zip	Tiedosto vastaanotettu kohteesta 1WJ1510GHPL005177	---	TimberLink	4.2 KB	Apr 26, 2023
<input type="checkbox"/> 1WJ1510GHPL005177_042620230732.zip	Tiedosto vastaanotettu kohteesta 1WJ1510GHPL005177	---	TimberLink	4.0 KB	Apr 26, 2023
<input type="checkbox"/> 1WJ1510GHPL005177_042620230647.zip	Tiedosto vastaanotettu kohteesta 1WJ1510GHPL005177	---	TimberLink	4.1 KB	Apr 26, 2023
<input type="checkbox"/> 1WJ1510GHPL005177_042620230632.zip	Tiedosto vastaanotettu kohteesta 1WJ1510GHPL005177	---	TimberLink	4.2 KB	Apr 26, 2023
<input type="checkbox"/> 1WJ1510GHPL005177_042620230547.zip	Tiedosto vastaanotettu kohteesta 1WJ1510GHPL005177	---	TimberLink	4.1 KB	Apr 26, 2023
<input type="checkbox"/> 1WJ1510GHPL005177_042620230447.zip	Tiedosto vastaanotettu kohteesta 1WJ1510GHPL005177	---	TimberLink	3.9 KB	Apr 26, 2023

Kuva 7. MyJohnDeere-sivuston ZIP-pakettilistaus

Kuvan 7 tiedostolistauksessa MyJD:ssä kaikki on koneilta tulevassa pakatussa muodossa. Asiakkaille pakattujen tietojen hallinta on haastavaa, koska paketeista ei saa selville sen sisältöä. Käyttäjän täytyy ladata kerralla useampi paketti ja purkuprosessi vie oman aikansa.

Timber Managerin käytetyimpiä toimintoja on leimikoiden hallinta ja leimikkotietojen jakaminen. Karttapohjaa apuna käyttäen voidaan luoda leimikon rajat ja säästettävät kohteet. Nämä eivät sisälly ohjetiedostoihin, vaan kuuluu käsitellä erillisen metsäkoneelle tehdyn ohjelmiston kautta, kuten kuvassa 10. Metsäkoneilla voi käyttää metsäkonevalmistajan tai metsäyhtiön tarjoamaa karttapalvelua. Suomessa metsäyhtiöiden karttasovelluksessa leimikon rajat ovat piirrettynä valmiiksi, mutta ulkomailla tilanne on toinen. Kuljettajan on siis piirrettävä karttapohjalle kaikki tarvittavat merkinnät itse. Merkit ovat pisteitä, viivoja tai alueita. Tällaisia merkintöjä voi olla metsästä riippuen useita, kuten leimikon rajat, laanipaikat, kosteikkoalueet, sähkölinjat ja leimikolle sertifikaatin vaatimat säästettävät puut ja korkeat kannot. Karttapohja auttaa metsätoissa, mutta näyttöä ei

voi töitä tehdessä tarkastella kauaa yhtäjaksoisesti. Viiva- ja aluemerkinntöihin voidaan liittää varoitus, josta kuljettaja huomaa vaaratilanteen koneen ollessa alueen läheisyydessä.



Kuva 10. Karttasovelluksella piirretyt leimikon rajat ja merkinnät

Joissakin tapauksissa, jopa Suomessa, kuormakoneelle ei tarjota ohjetiedostoja, jolloin käytetään pelkästään karttasovelluksen pohjaa. Karttasovelluksen kautta hakkuutyön yhteydessä tukkien sijainnit tulevat näkyviin kaikkialle niin metsäkoneille kuin verkkosivuillekin. Ilman StanForD-tiedostoja ei voida tehdä StanForD-rekisteröintiä, joka tarkoittaa, että kuormakoneelta ei saada standardin mukaisia tuotantoraportteja. Metsäkoneilla on saatavilla puutavaralaji- ja puulajikirjastot, joiden tietoja tarvitaan luomaan standardin vaatimia kuljetuksia. TO5:ssa juuri julkaistun ominaisuuden avulla puutavaralajien tiedot voidaan liittää kuljetuksiin, mutta sen tekeminen on vielä kohtalaisen monimutkaista.

4.1.2 Uusi järjestelmä

Metsäkoneiden lähettämät tiedostot ovat saatavilla metsäyhtiöitä ja analysointia varten sekä vanhan että uuden toteutuksen kautta. Uusi toteutus on tehty Timber Manageriin ja kulkee vanhan MyJD-toteutuksen rinnalla.

Tiedostonimi	Leimikon nimi	Kone	Tunnisteet	Päiväys	Toiminnot
kuoriparam_testi 2.hrv.env	-	ARDHARV01032023	+	04.05.2023	↓ □
Cole Cassidy_20230504_102935_testi.pdf	-	ARDHARV01032023	+	04.05.2023	↓ □
kuoriparam_testi 3.hrv.env	-	ARDHARV01032023	+	04.05.2023	↓ □
Cole Cassidy_20230504_100918.mom	kuoriparam_testi 2	ARDHARV01032023	work time	04.05.2023	↓ □
Cole Cassidy_20230504_102448.mom	kuoriparam_testi 3	ARDHARV01032023	work time	04.05.2023	↓ □
Cole Cassidy_20230504_102448.hpr	kuoriparam_testi 3	ARDHARV01032023	harvester production	04.05.2023	↓ □
Cole Cassidy_20230504_100918.hpr	kuoriparam_testi 2	ARDHARV01032023	harvester production	04.05.2023	↓ □
Cole Cassidy_20230504_092753.mom	kuoriparam_testi 2	ARDHARV01032023	work time	04.05.2023	↓ □
Cole Cassidy_20230504_091345.hpr	kuoriparam_testi 1	ARDHARV01032023	harvester production	04.05.2023	↓ □
Cole Cassidy_20230504_092753.hpr	kuoriparam_testi 2	ARDHARV01032023	harvester production	04.05.2023	↓ □
Cole Cassidy_20230504_091345.mom	kuoriparam_testi 1	ARDHARV01032023	work time	04.05.2023	↓ □
kuoriparam_testi 1.hrv.env	-	ARDHARV01032023	+	04.05.2023	↓ □
Cole Cassidy_20230504_095316_testi.pdf	-	ARDHARV01032023	+	04.05.2023	↓ □

Kuva 8. Timber Manager-sivuston tiedostolistaus

Kuvassa 8 olevassa uudessa toteutuksessa tiedostot on listattu käyttäjälle valmiiksi puretuina. Puretuissa tiedostoissa on tunnisteet tiedoston lähettäneestä koneesta. Toiminto karsii asiakkaalle tarpeettomat intervalein muodostuneet konedata tiedostot. Tiedoston päätteen perusteella listassa näkyy tiedostokohtaisesti ja automaattisesti lisättävä tunniste sen ollessa esimerkiksi työaika- tai tuotantodatatiedosto. Käyttäjä voi halutessaan myös itse lisätä tiedostoille haluamiaan tunnisteita. Tunnisteita, koneiden sarjanumeroita, aikaa ja vapaamuotoista tekstiä voidaan käyttää apuna listan suodattamisessa. Suodatin on pois-sulkeva, eli mitä enemmän ehtoja, sitä vähemmän tiedostoja näkyy.

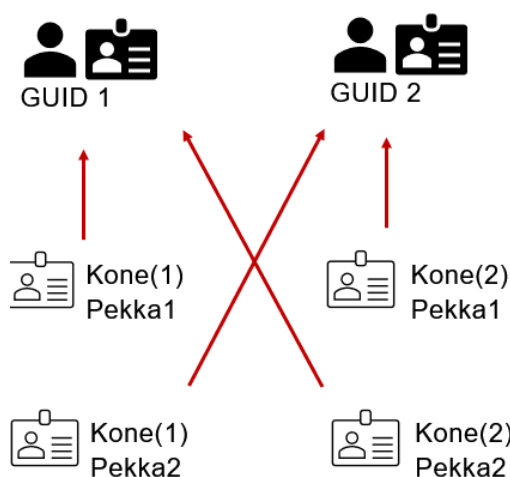
Tiedostonäkymässä on vanhan toteutuksen tapaan massatoimintoja, joilla arkistoida tai ladata tiedostoja haluttu määrä. Massatoimintoihin valittavat tiedostot voidaan määrittää suodattimin. Tiedostojen välittäminen eteenpäin on mahdollista toistaiseksi koneiden sarjanumeroiden perusteella, joko päivittäin, viikoittain tai kuukausittain. Kuukausittaisen datapaketin tilaaminen helpottaa TO5 käyttäjää tietokannan koon hallinnassa. Palataan tiedostojen käsittelyyn myöhemmin tarkastellessamme statistiikkaa, koska tämä ominaisuus on yksi sen mahdollistavista toiminnoista.

4.1.3 Kuljettajien hallinta

Kuljettajienprofiilit luodaan metsäkoneella ja kukin profiili on sidottu yhdelle koneelle. Kuljettajan lataaminen useammalle koneelle on mahdollista muistitikulla, mutta ei verkon ylitse. Kuljettajaprofiilissa kuljettaja voi asettaa kyseiselle koneelle henkilökohtaisia asetuksia ja parametrejä riippuen kone- ja kuljettajajyksilöistä sekä tehtävästä työstä. Ko-

neyksilöt vaativat muutoksia parametreihin tyyppin ja kokoluokan mukaan. Konekuljettajien hallinnan muuttaminen paikallisesta toteutuksesta pilvipalveluun mahdollistaa saman käyttäjän lataamisen useammalle koneelle. Profiilin lataaminen toiselle metsäkoneelle mahdollistaa samojen parametrien käytön. Parametrit ja kuljettajaprofiili eivät ole täysin StanForD:in mukaisia, sillä ne sisältävät lisättyjä tunnistetietoja. Kuljettajien profiileista on kyettävä erottelemaan StanForD:in osuus ja se on rinnastettava muuhun informaatioon.

Kuljettajat luovat uudelle koneelle uuden profiilin. Luomisen yhteydessä käyttäjälle saattaa tulla kirjoitusvirheitä. Kirjoitusvirheet johtavat saman henkilön luomiin useisiin profiileihin. Kuljettaja tunnustetaan tuotantodatatiedoissa StanForD:in tietokenttien, kuten kuljettajatunnuksen ja nimen perusteella. Kuljettajatunnus luodaan kuljettajaprofiiliin asetetun nimen perusteella ja molemmat tietokentät ovat täysin vapaasti muokattavissa olevia tekstikenttiä, mikäli profiilia ei ole suojattu salasanalla. Työmailla yhtä vuoroa tekevissä hakkuuorganisaatioissa on usein nähtävissä, että yhtä konetta kuljettaa vain yksi kuljettaja ja esimerkiksi sairaspöissaolon vuoksi hänen profiiliaan käytetään jonkun toisen toimesta. Työt kirjautuvat käytössä olevalle profiilille, eikä tuotantoa voi siirtää toiselle profiilille.



Kuva 9. Kuljettajien profiilien yhdistäminen pilvessä

Kun yksi kuljettaja käyttää useampaa konetta satunnaisesti ja on luonut oman profiilin jokaiselle nimellä Pekka. Profiilien nimet ja käyttäjätunnukset ovat samat, asiat hoituvat moitteetta. Jos kahden kuljettajaprofiilin nimet tai nimimerkit ovat lähes samat yhdellä tai useammalla koneella, heidän datansa erottaminen toisistaan hankaloituu. Koneen sisäisesti profiilien erottaminen toisistaan on hoidettu kuljettaja-avainmuuttujalla, joka on

juokseva luku. Kuvassa 9 on esitetty ongelma, jossa koneilla 1 ja 2 on profiilit Pekka 1 ja Pekka 2. Kuljettaja 1 on koneella 1 profiililla Pekka 1 ja toisella koneella Pekka 2. Huomataan että kuljettajilla on koneesta riippuen samat arvot, mutta eriävät identiteetit, eli heitä ei voi erottaa ilman ylemmän tason tunnistetta, joihin heidät olisi liitetty.

Pilvipalvelun kautta kuljettajien profiilien luonti ja niiden välittämien eri koneille, oikeine parametreineen poistaisi kaikki näistä ongelmista. Virheiden korjaamisilla on tapana luoda uusia ongelmia, mutta ainakaan ne eivät vääristäisi dataa. Data on arvokasta, mutta puhdas data vielä arvokkaampaa. Hallinnoimaton ja huonosti koottu data ajauttaa järjestelmän ennen pitkää datasuohon, josta ylösnouseminen ja sen puhdistaminen vaati paljon resursseja.

“Without a systematic way to start and keep data clean, bad data will happen.”

— Donato Diorio

4.1.4 Hakkuohjeiden liittäminen leimikkoon

Myös leimikon luomisessa on parannettavaa. Aikaisemmin StanForD:in esittelyn yhteydessä mainittu hakkuuohje on kriittinen osa leimikon luomista ja alustamista metsäkoneella. Hakkuuohjeet saadaan usein metsäyhtiöltä heidän puutavaratarpeisiinsa räätälöityinä ja oikealle hakkuuyhtiölle lähetettyinä. Kuljettaja saa tiedostot sähköpostitse tai erillisen metsäkoneelle asennettu metsäyhtiön sovelluksen kautta. Kuljettaja käy jokaisen leimikon kohdalla läpi usean vaiheen ohjeistuksen, jotta kone ja kuljettaja pystyvät käyttämään annettua ohjeistusta. Tärkeimpinä kaatokoneen tietoina ovat puutavaralajit, joiden mukaan puutavaran arvo lasketaan. Kuljettajan ammattitaitoon kuuluu metsäkoneen käyttö ja harvennuksen tapauksessa oikeiden puiden valinta. Oikeiden puiden valitseminen on tärkeää, jotta metsä jää parhaaseen mahdolliseen tilaan ennakoiden puuston kasvua useampia vuosia eteenpäin.

Hakkuuohjeiden luomisen tai tuomisen automatisointi pilvipalvelussa korjaisi ongelmatilanteen. Liikkeelle voidaan lähteä ensin tuomalla verkkosovellukseen puutavaralajikirjastot, joista valitsemalla oikeat leimikolla käytettävät puutavaralajit liitettäisiin tiedostoihin. Hakkuuohjeet pitäisi liittää leimikon tietoihin karttasovelluksessa, josta metsäkone saisi vaatimansa tiedot.

Kun leimikon luominen on mahdollista yhdestä paikasta koko järjestelmään se säästää koneen kuljettajalta useita minuutteja päivittäin. Karttojen luominen ja ohjetiedostojen

liittäminen leimikkoon tapahtuisi Timber Manager:in kautta. Toiminnon opetteluun vaatiessa aikaa ja vaivaa se jää huomiotta, joten leimikon luonnin tulisi olla mahdollisimman selkeää ja suoraviivaista.

4.2 Statistiikka ja analysointi

Statistiikka ja analysointi on TO5-sovelluksella edistyksellistä, jos sitä osataan tulkita ja yrittäjällä on aikaa perehtyä työntekijöidensä tuottamaan dataan. Dataan perehtyminen vaatii monihenkisessä hakkuuyrityksessä täysin analysointityöhön omistautuneen henkilön. Kuljettajille ei kuulu antaa pääsyä muiden kuljettajien statistiikkaan ja yritykset haluavat rajoittaa pääsyä finanssitietoihin.

Statistiikka on pitkälti tekstipohjaista ja diagrammit eivät ole tarpeeksi kuvaavia. Yritys haluaisi mahdollisimman paljon tuottoa ja metsäkoneenkuljettajat eroavat osaamistasolla toisistaan. Kuukausittainen suoriutumisen seuraaminen on tarpeen yrityksen menestyksen kannalta. Kuljettajien tai koneiden suoriutumisen notkahdukset pitäisi pystyä osoittamaan aikajanalla, sekä korjaamaan kohtalaisessa ajassa esimerkiksi kuljettajan jatkokoulutuksin tai koneen huollolla. Kuljettajien vertailu ja työosaamisen arviointi vaikuttaa siihen, mille leimikoille kuljettajat kannattaa sijoittaa parhaan tuloksen saavuttamiseksi.

Melkein kaikki tarpeellinen data metsäkoneelta on saatavilla, mutta se on tiedostomuodossa. Tiedostoja voidaan käyttää pohjana erilaisille metriikoille, joita kuljettajista ja koneista halutaan kuten TO5:ssa, mutta siihen täytyy liittää metsäkoneelta anturien dataa, jolla saataisiin yhä tarkempaa tietoa suoriutumisesta.

4.3 Tietokannan vaatimukset

Kuten lähteessä Chao [2013] mainitsi, tietokannan rakentamisen perusteina pitää ensin määritellä sen käyttötarkoitus ja tiedonlähteet. Ensimmäinen pilviympäristöön kehitettävä osa-alue oli aiemmin mainittu tiedostojen listaaminen. Se mahdollistaa pilvipalvelulle helpon pääsyn StanForD-dataan, joka voidaan käsitellä haluttuun muotoon ja siirtää talteen tietokantaan, josta data on vielä helpommin ja nopeammin saatavilla muille sovelluksille. Jos statistiikkaa halutaan tutkia, jokaisen tiedoston läpikäyminen eri toimintojen käynnistyessä ei ole tehokasta. Jos data taltioidaan tietokantaan suoraan tiedostoista, eikä tiedostoja hävitetä, datan määrä moninkertaistuu.

Kunnes putket (engl. pipeline) on saatu johdettua koneilta pilvipalveluihin halutulla tavalla, voidaan ensisijaisesti käyttää SF-tiedostopohjaista analysointia. Tiedostot on tarkoitus pilkkoa tietokantaan haluttuihin tietojoukkoihin XSLT:n avulla, jotta ne ovat useamman pilvipalvelun käytettävissä. Ongelmakohtiksi muotoutui sisällöltään toisistaan eroavat standardin mukaiset tiedostot, niiden käyttötapaukset ja muu lisäksi liitettävä data. Tietokannan tulisi olla mahdollisimman tehokas ja kaikkia StanForD-tiedostoja tukevan. Tärkeimpänä ja pidempiaikaista tutkimustyötä tarvitsevana toimintona on statistiikka ja analysointi, jotta TO5-sovellukseen verrattuna saadaan merkittävä parannus ja taustapalvelut (engl. back-end) vastaamaan vaatimuksia.

Tiedon tulisi olla muuttumatonta (engl. immutable) ainakin puiden tuotantotietojen tapauksessa. Puutavaran mittaustuloksien ei tulisi muuttua mittauksen jälkeen. Puutavarojen määrien taltioimiseen mietittiin aluksi aikasarjatietokantaa, johon olisi linkitetty koneiden ja kuljettajien tiedot. Kuljettajien ja puutavaran liittäminen toisiinsa antaisi mittarin tuottavuudesta. Nykyiseen malliin se oli turhan iso muutos, eikä välttämättä pysty yksinään kattamaan kaikkia mahdollisia tapauksia. Tiedon rinnastaminen on haasteellista, koska kaikkea tietoa ei saada vielä luotettavasti samasta paikasta.

StanForD2010 on jäsennelty sisennyksin ja jotkin osat datasta on ripoteltu myös muihin tiedostoihin, kaikkea ei tarvitse tallentaa joka kerralla uudelleen. Tällaisia tiedonjyviä ovat esimerkiksi kuljettajan, leimikon, puulajien ja puutavaralajien määritelmät. Raportointiin liittyvistä mom- ja hpr-tiedostoista löytyy molemmista samat leimikon ja kuljettajan määritykset. Tiedot ovat siis riippuvaisia toisistaan, joten ne voidaan asettaa relaatioon ja suorittaa kuljettajien ja leimikoiden suhteen vertailuja. Tähän käyttötapaukseen voitaisiin suoraan soveltaa relaatiotietokantaa.

Relaatiotietokannoissa huonoina puolina oli skaalautuvuus ja datan ennalta määrittely. Uuden SF10 version julkaisun myötä joitain tauluja joudutaan päivittämään. Päivityksessä sama tietokenttä pitää lisätä jokaiselle riville. Käyttäjälle ilmaantuu hetkellisiä häiriötä siihen asti, kun tiedot on päivitetty. Häiriöajan minimointi on tärkeää ja huoltokoista tulisi aina ilmoittaa asiakkaille, sekä suunnitella virheistä toipuminen huollon ongelmatilanteiden varalle.

Jos tarkastellaan ainoastaan puutavaraa, jota hpr- ja fpr-tiedostot sisältävät, tarvitaan puutavaroiden määritelmät. Katkotut puut ovat määritelty viiteavaimin, jotka XML-rakenteessa on määritelty kuuluvan tietylle rungolle, sitten puutavaralajille ja niin edelleen.

Suurin haluttu tarkasteltava kohde datassa on metsäkone, joka on jo niin sanottua ”rividataa”, joka aina rinnastetaan määritelmiin. Pienin haluttu tiedonjyvä on rungon katkotujen tukkien muuttujat kuten pituus ja paksuus. Avain-arvotietokantojen suorituskyky tämän tyyppisen datan kirjoittamiseen ja siihen suoritettavaan tiedonhakuun on huomattavasti nopeampaa kuin SQL-kyselyin relaatiokannasta. Myös aikaisemmin mainitsemani skaalautuvuus on parempaa NoSQL-tietokannoissa, koska uuden tietokannan voi luoda pienemmällä määrällä tietoa ja hajautetusti. Useamman NoSQL-kannan haut voidaan suorittaa samanaikaisesti, joka nopeuttaa hakua, kun käytössä on useampi prosessori. [Alam et al., 2013]

Rividata on sarja luettelomaista dataa, joka ei sisällä ylimääräistä määrittelevää tietoa sen entiteetistä. Kukin entiteetti voidaan ilmaista tekstitiedoston rivinä ja niiden keskinäisen sisällön eroavaisuutena on vain tietokenttien lukuarvot. Tukki on rividataa, joka koostuu numeraalisista eri mittauksin tuotetuista tiedoista sekä juoksevasta luvusta avainarvona (LogKey). Tukista mitataan paksuus ja pituus, joiden perusteella sille lasketaan tilavuus. Avainarvolla se voidaan rinnastaa kaadettuun runkoon. Runko voi sisältää yhden tai useamman tukin. StanForD2010:ssä tukit ilmoitetaan listattuna peräkkäisinä riveinä runkoelementin sisällä. Runko sisältää enemmän StanForD-määritelmiä, joilla se voidaan rinnastaa leimikkoon ja metsäkonetta käyttäneen kuljettajan profiiliin.

Ongelmana on kahden tyyllisen datan kerääminen isosta tietojoukosta, datan kysely mahdollisimman nopeasti ja kustannustehokkaasti. Pilvipalveluissa on mahdollista skaalata instansseja loputtomasti, mutta tiettyjen rajojen ylittyttyä, hinta nousee huomattavasti. Tieto ei saavu järjestelmään reaaliajassa ja aiheuttaa aina hetkellisesti kuormaa tiedostoja purettaessa. Kyselyiden nopeuden lisääminen on mahdollista, mutta isommalla instanssilla hinta nousee.

Yhtä isoa yhtenäistä sovellusta pilvessä tai levykkeellä sovelluskehittäjien keskuudessa kutsutaan monoliitiksi (engl. monolith), jolla on samanlaisia hyötyjä ja ongelmia kuin SQL-tietokannoilla. Ne voivat olla turhan isoja ja niitä on hidasta kehittää. Kehityksessä on otettava huomioon kaikkien osien vastaavuus uuden ominaisuuden arkkitehtuuriin ja julkaistaessa koko sovellus on koottava ja rakennettava (engl. compiled and built) uudelleen. Sen asentaminen aiheuttaa usein huoltokatkoja palvelimille.

Micro-palvelut (engl. microservice) ovat pieniä osasia järjestelmässä, joista kokoamalla saadaan kehitettyä sovellus. Ne mahdollistavat kehittämisen useammalla koodikielellä ilman huoltokatkoja ja tukevat ketterää kehittämistä. Micro-palveluita voi olla niin monta kuin tarvitaan. Kaikkien osien ei tarvitse odottaa aikaisemman prosessin loppumista. Micro-palvelut tukevat useamman prosessorin käyttöä. Toiminta on ideaalitalanteessa nopeaa ja palveluita voi sammuttaa sitä mukaa kun niitä ei enää tarvita aktiivisesti. Kehityksessä ongelmana verrattaen monoliitteihin on uusien palasten välinen kommunikaatio, palvelujen tarkka määrittely, testaaminen, turvallisuus ja yleisesti monimutkainen rakenne. Micro-palvelut omaavat samanlaisia ominaisuuksia kuin NoSQL-tietokannat. [Balalaie et al., 2016; Alam et al., 2013]

4.4 StanForD-tietokannan arkkitehtuuri ja datan käsittely pilvessä

Timber Manager on kehitetty AWS (Amazon Web Services) -pilvipalvelualustalle. Uusien ominaisuuksien haluttiin olevan mahdollista siirtää kohtalaisessa ajassa myös muiden pilvipalvelujen tarjoajien alustoille, joten niiden tuominen Timber Manageriin oli loogista. Tarjoajilla on usein omat nimensä erilaisille palveluille ja niiden toimintatavat voivat poiketa toisistaan. Virtuaalikoneiden kuvat (engl. image) joilla asennukset alustetaan, ovat yhteensopivia kaikilla alustoilla, samoin tietokantojen luontilauseet.

Amazon Web Service:llä on tarjolla palveluita, joista osa on heidän hallinnoimiaan. AWS:n hallinnoimattomien palveluiden käyttäminen jää yrityksen omalle vastuulle niin turvallisuuden kuin käyttötärpeeseen kehittämisen kannalta. Kumppaneina käyttäjätuen kautta voidaan ongelmatilanteissa olla yhteydessä, mikäli jokin palvelu antaa niin suuren edun, että sen hallinnoiminen on kehittäjien mielestä kannattavaa. Pääsääntöisesti tulisi käyttää ratkaisuja, jotka ovat AWS:n hallinnoimia.

Tietokannan toteutus nimettiin StanForD-tietokannaksi. Tietokantoja päädyttiin käyttämään ratkaisussa micro-palveluiden omaisesti. Micro-palvelut antaisivat tulevaisuudessa mahdollisimman paljon hyötyä mahdollisimman pienin kustannuksin. Timber Manager on rakennettu käyttämällä micro-palveluita, joten jo olemassa oleva kompleksisuus ja uuden opettelu ei ole kehittäjille ongelma. Pilvipalveluissa ei olla sidottuja yhteen ratkaisuun, joten on mahdollista käyttää sekä SQL- että NoSQL-tietokantaratkaisujen ominais-toimintoja hyödyksi niille sopiviin tarpeisiin. [Alam et al., 2013]

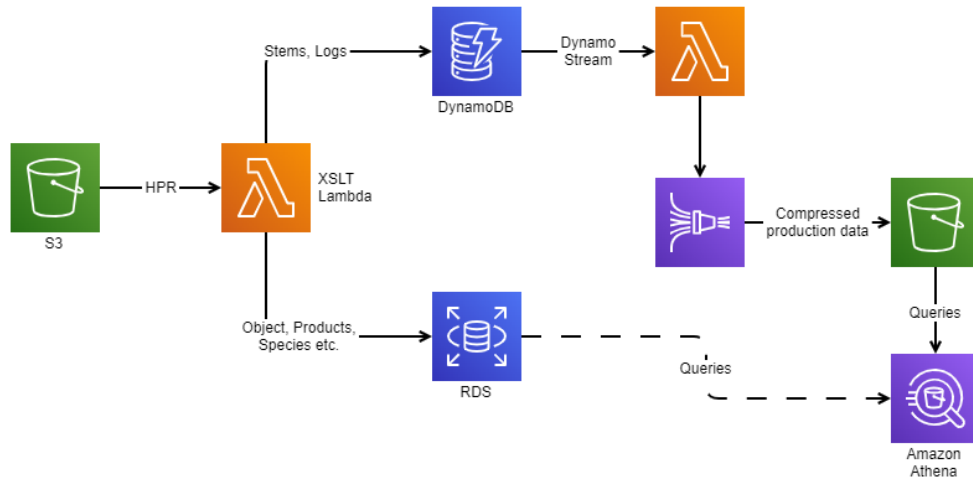
StanForD-tietokannalla mahdollistetaan vaadittujen toimintojen datan taltiointi. Tiedostojen tiedot puretaan XSLT:n avulla tietokantaan, jotta dataan päästään käsiksi micro-

palveluista nopeasti. Tietokannassa kuljettajien profiilit ja niiden parametrit voidaan eriyttää eri tauluihin ja asettaa relaatioon. Tällä ratkaistaan ongelma kuljettajan profiilin StanForD:in mukaisesta ja muusta siihen kuulumattomasta datasta. Kuljettajaprofiiliin lisätään globaalisti uniikki tunniste, jonka alle kuljettajien tunnistetiedot listataan. Globaalin tunnisteen alle kirjataan kuljettajan tunniste, nimi ja koneelta saatuja avaimia. Kun mahdollistavat toiminnot on tehty, eli raportointitiedostot purettu tietokantoihin, kuljettajille luotu organisaation mukaan yksilöivät profiilit ja leimikoiden luonnin ja raportoinnin osalta voidaan luottaa datan olevan oikeellista, voidaan vihdoin pureutua itse datan analysointiin. Statistiikan ja raportoinnin seurannan helpottaminen on ollut pitkään hakkuuyritysten haaveena ja se on näiden ominaisuuksien myötä mahdollista. Kaiken ollessa nivottu yhteen totuudenlähteeseen asiakas voi luoda raportteja mittavammin viikoittaisiin tarpeisiinsa. Asiakkaan on mahdollista myös suorittaa kuljettajien ja koneiden vertailua.

Koska kuljettajat ovat tunnistettavissa, profiiliin voidaan relaatioilla liittää koneiden tuottamaa dataa. Sen perusteella kuljettajat on helppo profiloida ja segmentoida erilaisiin työtehtäviin osaamisen perusteella. Organisaation esimiehet ja kuljettajat saavat luotettavaa dataa, jota voidaan käyttää parantamaan kuljettajan osaamista tai jopa mittareina työhaussa. Tarve on siis luoda jälkeinpäin kattavia StanForD-muotoisia raportteja, jotka perustuvat StanForD-tietoon. Työn päätyttyä leimikolla, voidaan luoda kaikenkattava raportti. Nämä raportit voidaan välittää tietoa tarvitseville tahoille, kuten metsäyhtiöille, mikäli heillä ei jo ole pääsyä dataan sovelluksien kautta. Tietokantaan lisättävä konedata moottorinkäyntitunneista ja sensoreilta tuo asiakkaiden käyttöön uusia mittareita.

Leimikoiden työn arviointi jälkikäteen on myös metsäkonevalmistajille arvokasta, sillä yritysten antama data voidaan hyödyntää myös myynnissä. Kun riittävä määrä leimikoita on tehty tietyn mallisilla ja kokoisilla koneilla, voidaan antaa arvioita esimerkiksi suurempien kokoluokkien koneiden paremmuudesta työssä. Jos pienemmällä koneella tietyllä hakkuupäällä tehty työ vei aikaa määrän X, isommalla koneella ja kenties optimaalisemmalla hakkuupäällä hakkuutyöhön teoreettisesti aikaa olisi kulunut määrä Y. Arvioihin perustuvaa kestoa leimikoista suoritetaan jo nyt yrityksissä, mutta se voisi perustua oikeaan dataan. Karttoihin on rajauksien mukaan laskettavissa leimikon pinta-ala ja sen tuottamat kuutiot. Ainakin Suomessa lähes kaikki, ellei kaikki metsistä on laserkeilattu, joka tarkoittaa lidar-teknologian (*light detection and ranging*) avulla kuvattua metsää. Sen perusteella saadaan tarkka arvio puuston tiheydestä ja runkojen tilavuudesta. Lidar eli valotutka on tavallaan kaukoluotaamista, mutta laservalopulssein. [Admin, 2012]

Dataa on hyvä saada myös kilpailijoiden järjestelmistä. StanForD-datan vajavaisuuden vuoksi perusteellista vertailua kaikesta ei voida suorittaa. Leimikon ohjetiedostojen luominen muun merkkisille koneille ei myöskään ole ongelma, koska se kannustaa ostamaan järjestelmien käyttöä helpottavia metsäkoneita.



Kuva 11. StanForD-tietokannan pilviarkkitehtuuri ja datan käsittely.

Miten pilvipuolen toteutus käytännössä on toteutettu? Kuva 11 luettaessa vasemmalta oikealle kuvaa datan käsittelyn eri AWS:n palveluilla. AWS-pilvessä yleisimpiä tiedon tallointitapoja ovat ”S3 buckets” eli ämpärit, joihin voidaan tallioida tietoa objektimuodossa. Objekti koostuu tiedostosta tai kansioista ja sen metadatatista. Objektit voivat sisältää mitä tahansa dataa kuten kuvia, videoita tai tekstiä. Objektit tallioidaan oletuksena useampaan paikkaan samanaikaisesti, joka vähentää riskejä datan katoamisesta. Vahingoittuneen tiedon ilmetessä kyseinen objekti korvataan sen kopiolla. Jokaisen ämpäriin objektin käyttöoikeuksia voidaan hallita erikseen. [Nakivo, 2022] XML- tiedostot tallioidaan S3-ämpäriin ja jatkokäsittellään sieltä tarkempiin käyttötarkoituksiin. S3-palvelu on graafitietokannan tapainen, mutta tippuu ”pilvi varasto”-kategorian alle. [Sunil, 2017] Tavara- ja puulajikirjastot sisältävät StanForD:in ohjetiedostoihin tarpeelliset määritelmät. Käyttöliittymän kautta voidaan ohjetiedostojen luomiseen asettaa oikeat puutavara-lajit ja luoda koneilla käytettävät tiedostot. Tiedot eivät päivity usein ja ne on rinnastettava aina jonkun tukin tai leimikon tietoihin. Tämän tiedon perusteella sopivin sijoituskohta datalle on relaatiotietokanta.

Attribute name	Value	Type
PK - Partition key	1WJ1470GXD000420_5551e38b-1489-4e7b-8a28-9eb6d1eb0c61	String
SK - Sort key	25600	String
DBH	359	Number
Machine	1WJ1470GXD000420_5551e38b-1489-4e7b-8a28-9eb6d1eb0c61	String
ObjectKey	11	Number
OperatorKey	4	Number
ProcessingCategory	SingleTreeProcessing	String
ReferenceDiameter	Insert a field ▼	Map
SpeciesGroupKey	37	Number
StemGrade	Insert a field ▼	Map
StemKey	25600	Number
StemNumber	37	Number
SubObjectKey	11	Number
Type	STEM	String

Kuva 12. AWS NoSQL-tietokannan rivin data

StanForD sisältää sisennyksin toisissaan relaatioissa olevaa dataa, jonka perusteella kyseinen data voidaan käsitellä parhaiten relaatiotietokannassa. Määritelmät ja viittaukset ovat relaatioissa, joten ne on käsiteltävä SQL-kyselyin ”amazon relational database service”-palvelulla eli lyhyemmin RDS:llä. Runkojen ja tukkien käsittely tapahtuisi avainarvoparitietokannassa nimeltä Dynamo. Lambdassa pyöritetään Python 3:sta ja sen omasta kirjastosta löytyvää ”etree” eli ”element tree parser scriptiä”, joka on nimenomaan XML-muotoisen datan pilkkomiseen tarkoitettu moduuli. XSLT-jäsentelijä herätetään aina kun ämpäriin saadaan uusi tiedosto. Riippuen datasta sille osoitetaan oikea paikka jommastakummasta tietokantatyypistä. Jäsentelyn yhteydessä muuttujien keskuudesta poistetaan turhat kentät, joiden arvona on ”other” tai muu merkityksetön arvo. RDS-tietokantaan laitetaan leimikoiden, puutavaroiden ja puulajien määritelmät jaoteltuna kukin omaksi taulukseen. Ainoana poikkeuksena on työvuorojen määritelmät, jotka lähetetään dynamolle sen sisältämän työvuororividatan vuoksi.

Määritelmästä saadaan selville esimerkiksi työntekijöiden tekemien työvuorojen pituudet. Dynamo oli haku- ja kirjoitustaaajuuden kannalta huomattavasti nopeampi, joten on luonnollista kirjoittaa sinne runkoyksilöiden tiedot. Kuvassa 12 on esimerkki Dynamon yhden rivin sisällöstä. Dynamossa jakoavaimena (engl. partition key) toimii koneen tunniste, jolla saadaan myös datan alkuperä. Järjestysavain (engl. sort key) vaihtelee tiedostotyypin mukaan helpottaen samanlaisen tiedon hakua koko kannasta, mutta on hpr-tapauksessa rungon tunniste. Joissakin Dynamon sarakkeissa on yhdistettynä muuttujia,

jotta haku nopeutuisi. Koska tukit ovat yksilöllisiä ja koneavaimiin sidottuja, niiden kaksoiskappaleita ei pitäisi ilmetä, vaikka tiedosto olisi vahingossa syötetty useampaan kertaan. Dynamossa uuden tiedoston ilmetessä käydään tukit läpi ja luodaan ne kantaan PUT-operaatiolla. PUT-operaatio alustaa kyseisen rivin uudelleen, jos tieto on jo olemassa.

Dynamo virta (engl. stream) tallentaa lokiin muutokset kannassa ja lähettää ne eteenpäin JSON- muodossa, josta ne voidaan päivittää muille dataa käyttäville päätepisteille [AWS, 2023]. Dynamo virran luodessa uutta tietoa, herätetään lambda toiminto, jolla taltioidaan puhtaampi datan uuteen ämpäriin käyttäen AWS:n kineses data firehose-palvelua. Kinesis-palvelulla datavirta voidaan kompressoida sekä muokata haluttuun muotoon, mutta se ei voi olla yhtä megatavua suurempi tietojoukko.

Mitä tästä hyödytään? Amazon Athena voi hakea tietoa SQL-lausekkein ja luoda sen mukaan graafeja nopeina näkyminä (engl. quick sight). Athena kykenee hakemaan dataa tietoltaista, tietokannoista ja bigdata-sovelluskehysistä. Kun tieto saadaan perille ämpäriin, vaikka se on saatu alun perin NoSQL-tietokannasta siihen voidaan suorittaa SQL-kyselyitä käyttäen Amazon Athena-palvelua. Athena:lla luotua grafiikkaa on mahdollista upottaa pilvisovelluksiin käyttäjien nähtäville.

Ennen Athena:a yritettiin käyttää Glue-palvelua, jolla olisi ollut mahdollista lukea useasta resurssista dataa ja liittää ne yhteen sekä lähettää RDS:lle omaksi sarakkeekseen. Sarake olisi sisältänyt aggregoitua, eli summattua dataa. Tukkien mitat oltaisiin voitu laskea yhteen ja tietojen tarkastelu olisi tapahtunut lähinnä vain RDS:n kautta. Mikäli aggregoitua dataa hyödynnettäisiin nykyisessä ratkaisussa, Dynamo virrasta voidaan ottaa tarvittavat arvot sellaisenaan ja laskea ne lambda funktiolla yhteen. Athena:a apuna käyttäen pystytään tekemään kyselyt SQL-muotoisina suoraan Dynamo-tietokannasta, joten datan aggregointi on näillä näkymin turhaa. Glue-palvelua käyttämällä, ei samanaikaisesti olisi käytetty Dynamon datavirtaa ja se olisi aiheuttanut ongelmia kaksoiskappaleisten tukkien kanssa, koska prosessi olisi vaatinut aikaleimat SstanForD-tiedostosta. StanForD-tiedostoista toiminto on usein otettu pois käytöstä, ettei hakkuutyön tuottavuus paljastu metsäyhtiölle.

Arkkitehtuurin ja instanssien pystyttämiseen yaml-tiedostoista tarvitaan välivaiheita. Git on versionhallinta työkalu, jonka avulla isoissa projekteissa voidaan luoda eri sovelluksen

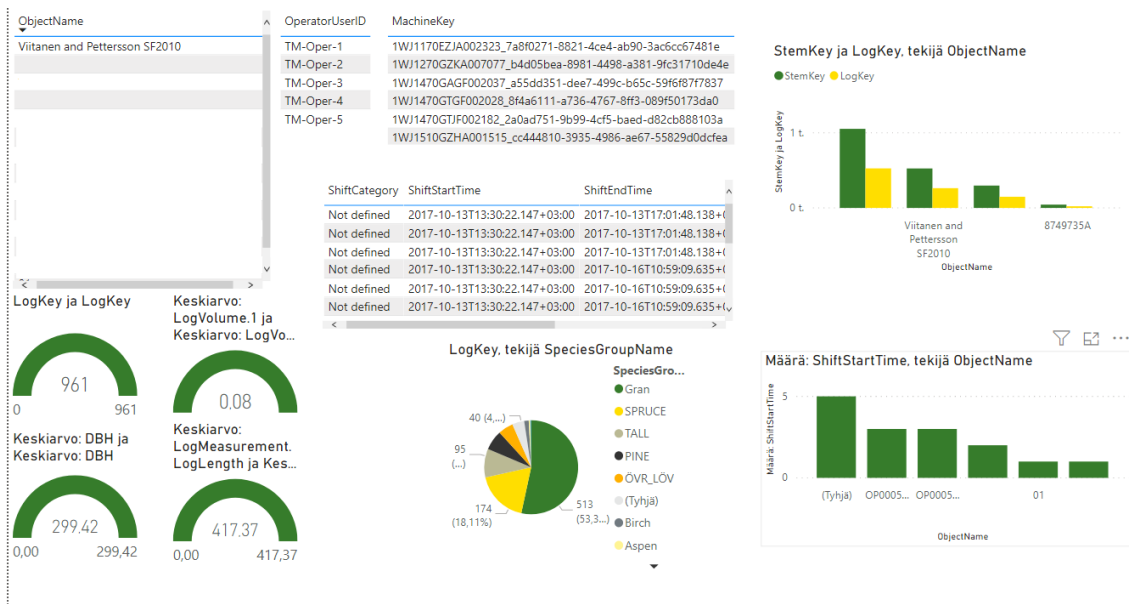
toiminnallisuuksista omat haaransa. Projektin haarat ovat aliversioita ja niitä voidaan kehittää samanaikaisesti. Sivuhaarojen muutokset yhdistetään julkaistavaan päähaaraan, kun haarassa tehty uusi ominaisuus on valmis. Git-versionhallinta voidaan kuvitella aikajanana projektin kehitykselle, jolle taltioidaan vain muutokset tiedostojen sisällöstä.

Useamman kehittäjän tiimissä otetaan usein avuksi jokin verkossa toimivista Git-alustoista kuten GitBucket tai aikaisemmin salausavaimien yhteydessä mainitsemani GitHub. GitHub:ssa on jokaiselle projektille oma säilö ja siihen voi tehdä muutoksia Git-komennoin eri päätelaitteilta samanaikaisesti, kunhan salausavaimet on asetettu oikein säilön ja päätelaitteen välillä. Avoimen lähdekoodin yhteydessä käytetään lähes aina Git:iä, joka itsessäänkin on avointa lähdekoodia. GitHub-sivustolla eri välilehdille on jäsenneily lähdekoodi, viat, tulevat koodimuutokset sekä muita ominaisuuksia, mutta tärkeimpänä tässä kohtaa ovat ”toiminnot”. [Heller, 2021]

GitHub actions, eli toiminnot välilehdeltä voidaan automatisoida työnkulkuja, joista arkkitehtuurin rakentaminen on yksi. Kaikki mikä voidaan automatisoida, halutaan automatisoida, kuten testaaminen ja uusien versioiden oikeellisiksi tarkistaminen. Muutokset projektin sisällä tai kolmannen osapuolen sovelluksesta luovat tapahtumia ja tapahtumien perusteella kuhunkin tapahtumaan voidaan ottaa kantaa halutuilla tavoilla erikseen. [Heller, 2021]

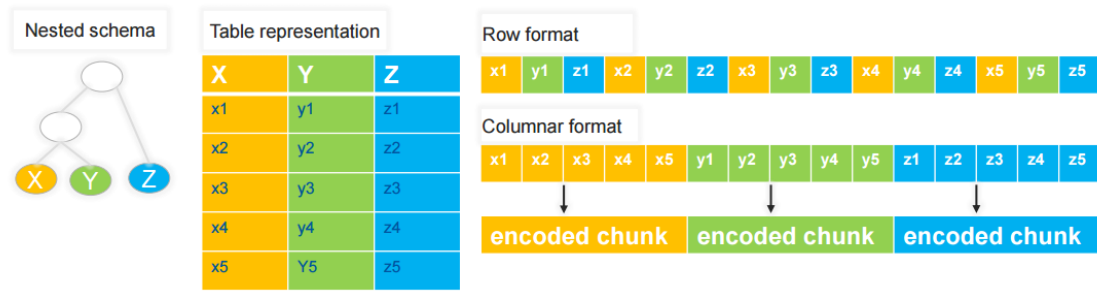
5 Jatkokehitys

Hakkuuyrityksille ongelmallisinta omien koneiden ja kuljettajien seurannassa oli tilastojen hankala seurattavuus, koska työkalut vaativat opettelua, sekä pahimmassa tapauksessa tiedostot on siirrettävä metsäkoneelta muistitikulla työpöytä sovellukseen. Pienemmillä yrityksillä ei välttämättä ole resursseja palkata toimistotyölle omistautunutta henkilöä. Asiakkaille voidaan StanForD-tietokannan myötä tarjota helposti luettavia mittareita erilaisin graafein, jotka on ennalta määritelty. Metsäkoneen datan mittareiden ei pidä olla monimutkaisia, vaan kaiken tarpeellisen tulisi käydä ilmi yhdellä nopealla vilkaisulla, kuten kuvassa 13.



Kuva 13. Power BI:llä StanForD-tiedoista luotu näkymä

Haasteina nopeasti vilkaistavissa olevissa mittareissa on kaikkien asiakkaiden toiveiden toteuttaminen. Kaikkien toiveita harvoin voidaan täyttää, varsinkaan globaalisti, joten joiain on jätettävä toteuttamatta. Ketterässä kehittämisessä oleellista on minimivaatimusten täyttäminen aluksi ja toteutuksen jatkokehittäminen asiakkaiden palautteen perusteella. Eri mittareiden yhdistäminen voidaan toteuttaa, mutta asiakkaalle tulisi uusi opeteltava työkalu. Onneksi tilanteeseen löytyy toinen ratkaisu. API (Application programming interface), eli rajapina, jota voidaan käyttää eri sovellusten väliseen kommunikointiin. Rajapinnoista yleisimpiä ja tutuimpia ovat HTTP-rajapinnat, joilla selaimet ja palvelimet keskustelevat keskenään.



Kuva 14. Saraketietokanta [Bhati, 2016]

Asiakkaille tarjottaessa rajapinnan kautta dataa, siitä saadaan reaaliaikaisempaa, vapaasti muokattavaa ja asiakkaan käyttötarkoitusta kuvaavaa, mutta asiakas joutuu luomaan mitarinsa itse. Tämä mahdollistaa suuremmille hakkuuyrityksille pääsyn tarkempaan data-analytiikkaan. Yleisiä API:hin yhdistettäviä data-analysointityökaluja ovat Tabula ja Power BI. Työkaluilla on mahdollista yhdistää yhteen tai useampaan erityyppiseen tiedonlähteeseen, kuten SQL-palvelimeen, tiedostoihin ja rajapintoihin. Monimutkaisemmissa tapauksissa ratkaisu vaatii relaatiotietomallin rakentamisen työkalun näkymään.

Palveluiden välisen tietoliikenteen viiveen vähentämiseksi projektissa voitaisiin käyttää XML- ja JSON-datan sijaan Apache Parquet-formaattia, joka on huomattavasti kompaktimpi muoto. Se on vanhan StanForD:in tapaan tehty koneille nopeasti luettavaan tarkoitukseen ja on tietoliikenteessä yleistymässä sen nopeuden vuoksi. Parquet on Apache:n kehittämä tiedontaltiointiin tarkoitettu muoto, jonka pääasiallinen käyttötarkoitus on hallinnoida dataa tietokannoissa ja sen kehityksessä on ajateltu erityisesti monimutkaisen sisennetyn datan käsittelyä [Github, 2023]. Formaattilla pyritään vastaamaan tietokantakyselyihin nopeammin ja se eroaa normaalista riveihin perustuvasta hausta, koska data taltioidaan sarakkeisiin rivien sijaan. Kuvassa 14 on erilaisia tapoja esittää sarakedataa. Nopeus perustuu sarakkein haettuun tietoon, jossa vain tarpeellinen osa tietoa haetaan, joten tarvittava muisti prosessiin on vähäisempi. Formaatti soveltuu hyvin datan syöteenä analysointityökaluihin kuten Power BI. Jos tietokantaan tehdään suuremman tietojoukon sisältäviä hakuja etu muistin vähäisestä käytöstä menetetään. [Apache Parquet, 2022]

6 Yhteenveto

Työn tavoitteena oli siirtää TimberOffice 5 -työpöytäsovelluksen toimintoja pilvipalveluympäristöön ja tuoda sen rinnalle uusia toiminnallisuuksia parantaen käyttökokemusta. Uusia Timber Manager pilviympäristöön tuotavia ominaisuuksia löytyi useampia, joista osa oli jo toteutettu vanhoissa järjestelmissä. TimberOffice 5:n statistiikkanäkymän ja ohjetiedostojen luomisen lisäksi haluttiin parannusta MyJohnDeere -sivuston ZIP-tiedostolistaukseen. Uusina toimintoina mahdollistettiin metsäkonekuljettajien hakkuuorganisaation laajuinen hallinta ja leimikon hallinnan kautta metsäkoneille tuotavat ohjetiedostot.

Statistinen data on TimberOffice 5:ssa vaatinut hakkuuorganisaation johdolta perehtymistä, jos työntekijöiden tai metsäkoneiden suoriutumisesta on haluttu tarkempaa tietoa. Kuljettajien hallinnalla mahdollistettiin jokaisen kuljettajayksilön tunnistaminen, vaikka käyttäjänimet eroaisivat koneiden välillä toisistaan. Kuljettajan tunnistaminen mahdollistaa StanForD-datan liittämisen identiteettiin, jotta vertailua kuljettajista voidaan tehdä tarvittaessa useamman koneen ja leimikkotyypin välillä. StanForD-data tuodaan Timber Manager:iin vanhasta MyJohnDeere:n rajapinnasta, mutta ZIP-paketit puretaan uuteen listaukseen, josta niiden etsiminen esimerkiksi nimen tai tiedostopäätteen perusteella on helpompaa.

Hakkuuohjeiden luominen ja niiden lähettäminen leimikon hallinnan kautta nopeuttaa työskentelyä, koska ohjeet leimikon rajoineen ovat ladattavissa leimikon aloituksen yhteydessä. Leimikon aloitukseen ei ominaisuuden myötä vaadita ulkoisen muistin kantamista metsästä toimistolle ja takaisin.

Projekti vaati perehtymistä dataan ja käyttöympäristöissä käytettäviin teknologioihin. Teknologioiden valintaan päädyttiin kokeilemalla niiden soveltuvuutta ensin pienemmissä ympäristöissä. Ominaisuuksien käyttövaatimukset ja käytettävissä olevat teknologiat muovasivat projektinaikaisia määritelmiä. Kokemusten perusteella ja kattavan vertailun jälkeen projektia voitiin ohjata oikeaan suuntaan valitsemalla sopivimmat toteutustekniikat ja ympäristöt. Arkkitehtuurillisiin ratkaisuihin eniten vaikutti jo käytössä olevat järjestelmät ja niiden kanssa yhteensopivuus. Kun yhteen ratkaisuun oli sitouduttu, sen jatkojalostaminen tavoitteita vastaavaksi oli helpompaa lisäämällä ympärille AWS:n tarjoamia monipuolisia pilvipalveluita.

Kaikkiin tässä luvussa mainittuihin ongelmiin vastattiin StanForD-tietokannalla. Tietokannan toteutukseen sisältyy sekä SQL- että NoSQL-tietokanta AWS:n tarjoamana pilvipalveluna. StanForD-tiedostot käsitellään tiedostolistauksen kautta tietokantoihin ja data eritellään sen tyyppin mukaan. Rividata sijoitetaan NoSQL-tietokantaan ja StanForD-määritelmät relaatiotietokantaan. StanForD-tietokanta mahdollistaa rividatan ja määritelmien hakemisen, sekä liittämisen toisiinsa avaimien ja tunnisteiden avulla. Tietokantaratkaisulla voidaan käsitellä myös dataa, joka ei ole StanForD-muotoista.

Projektilla luotiin pohjaa tuleville standardiin perustuville toiminnoille, joilla voidaan helpottaa metsäkonetyöskentelyä monin eri tavoin. Olen ollut projektin alusta asti mukana suunnittelussa, rikastamassa ideaa ja luomassa määriytyksiä sekä vaatimuksia, jotka eivät olleet aluksi itsestään selviä.

7 Viiteluettelo

- Admin. (2012). Laserkeilausaineiston hyödyntäminen metsätaloudessa. Haettu 8.5.2023 osoitteesta <http://www.puuntuottaja.com/laserkeilausaineiston-hyodyntaminen-metsataloudessa/>
- Alam, M., Muley, A., Kadaru, C., & Joshi, A. (2014). *Oracle NoSQL database : real-time big data management for the enterprise (1st edition)*. McGraw-Hill Education.
- Apache Parquet. (2022). Documentation. Haettu 9.5.2023 osoitteesta <https://parquet.apache.org/docs/>
- Alapati, S., Neu, W. A., Oram, A. & Vlasceanu, V. (2019). *An Introduction to Cloud Databases*. O'Reilly Media, Inc.
- Arlinger, J., Möller, J. J., Räsänen, T., & Sorsa, J.-A. (2008). Metsätehon raportti 201: StanForD 2010 Benefits, needs and requirements for users and manufacturers. Haettu 8.5.2023 osoitteesta <https://www.metsateho.fi/stanford-2010-benefits-needs-and-requirements-for-users-and-manufacturers/>
- Arlinger, J., Möller, J. J., Sorsa, J.-A. & Räsänen, T. (2020). Introduction to StanForD 2010. Haettu 8.5.2023 osoitteesta https://www.skogforsk.se/cd_20210111174505/contentassets/1a68cdce4af1462ead048b7a5ef1cc06/stanford-2010-introduction-200820.pdf
- EU (2016). Asetus 2016/679, Luonnollisten henkilöiden suojelusta henkilötietojen käsittelyssä sekä näiden tietojen vapaasta liikkuvuudesta ja direktiivin 95/46/EY kumoamisesta (yleinen tietosuojasetus). Haettu 10.5.2023 osoitteesta <https://eur-lex.europa.eu/legal-content/FI/TXT/HTML/?uri=CELEX:32016R0679&from=FI>
- AWS. (2023). Change data capture for DynamoDB Streams. Haettu 8.5.2023 osoitteesta <https://docs.aws.amazon.com/amazondynamodb/latest/developer-guide/Streams.html>
- Badshah, G., Liew, S.-C., Zain, J. M., & Ali, M. (2016). Watermark Compression in Medical Image Watermarking Using Lempel-Ziv-Welch (LZW) Lossless Compression Technique. *Journal of Digital Imaging*, 29(2), 216–225. <https://doi.org/10.1007/s10278-015-9822-4>
- Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture. *IEEE Software*, 33(3), 42–52. <https://doi.org/10.1109/MS.2016.64>
- Banxia software. (2023). XML data transfer. Haettu 8.5.2023 osoitteesta <https://banxia.com/dexplore/resources/frequent-questions/xml-data-transfer/>

- Bhati, V. (2016). Data Storage and Modelling in Hadoop. <https://techmagie.wordpress.com/category/big-data/data-modelling/>
- Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S. & Schaub, T. (2016). The GeoJSON Format. Haettu 10.5.2023 osoitteesta <https://datatracker.ietf.org/doc/html/rfc7946#page-28>
- Chao, L. (2013). *Cloud Database Development and Management*. <https://doi.org/10.1201/b15264>
- Cooper, M. & Carlson, B. (2021). Improving Git protocol security on GitHub. Haettu 9.5.2023 osoitteesta <https://github.blog/2021-09-01-improving-git-protocol-security-github/>
- Deutsch, P. L. (2023). DEFLATE Compressed Data Format Specification version 1.3. (1996). Haettu 10.5.2023 osoitteesta <https://www.rfc-editor.org/rfc/rfc1951>
- Eismann, S., Scheuner, J., van Eyk, E., Schwinger, M., Grohmann, J., Herbst, N., Abad, C. L. & Iosup, A. (2021). Serverless Applications: Why, When, and How? *IEEE Software*, 38(1), 32–39. <https://doi.org/10.1109/MS.2020.3023302>
- Finlex (2013). Säädös 1085/2013, Laki metsälain muuttamisesta. Haettu 11.5.2023 osoitteesta <https://www.finlex.fi/fi/laki/alkup/2013/20131085>
- Forest.fi. (2023). Puunkorjuu (harvesting). Haettu 9.5.2023 osoitteesta <https://forest.fi/fi/sanasto/puunkorjuu-harvesting/>
- Gill, S. S., Garraghan, P. & Buyya, R. (2019). ROUTER: Fog enabled cloud based intelligent resource management approach for smart home IoT devices. *The Journal of Systems and Software*, 154, 125–138. <https://doi.org/10.1016/j.jss.2019.04.058>
- Github. (2023). Parquet. Haettu 9.5.2023 osoitteesta <https://github.com/apache/parquet-format>
- Gulabani, S. (2017). *Practical Amazon EC2, SQS, Kinesis, and S3: A Hands-On Approach to AWS*. Apress L. P.
- Luonnonvarakeskus. (2022a). Luonnonvarakeskuksen tilasto-ohjelma. Haettu 8.5.2023 osoitteesta <https://www.luke.fi/fi/documents/luonnonvarakeskuksen-tilastoohjelma-20222024>
- Luonnonvarakeskus. (2022b). Tilasto-ohjelma Liite 1. Luonnonvarakeskuksen tilastot. Haettu 9.5.2023 osoitteesta <https://www.luke.fi/fi/documents/luken-tilastoohjelman-liite-1-luken-tilastot>
- Luonnonvarakeskus. (2022c). Hakkuukertymä ja puuston poistuma 2021 (ennakko). Haettu 10.2.2022 osoitteesta <https://www.luke.fi/fi/uutiset/tukkipuun-hakkuissa-ennatykset-rikki-vuonna-2021>
- Metsähallitus. (2023). Metsäsertifiointi ja ympäristöjärjestelmä. Haettu 11.5.2023 osoitteesta <https://www.metsa.fi/vastuullinen-liiketoiminta/metsatalous/sertifiointi/>

- Metsäkeskus. (2023). Metsänkasvatus ja kasvatushakkuut. Haettu 11.5.2023 osoitteesta <https://www.metsakeskus.fi/fi/metsan-kaytto-ja-omistus/metsanhoito-ja-hakkuut/metsankasvatus-ja-kasvatushakkuut>
- Nakivo. (2022). Introduction to Amazon S3: How Object Storage in the Cloud Works. Haettu 8.5.2023 osoitteesta <https://www.nakivo.com/blog/amazon-s3-object-storage-introduction/>
- Paul, J. J. (2023). *Distributed Serverless Architectures on AWS: Design and Implement Serverless Architectures*. Apress L. P.
- Priscila Heller. (2021). *Automating Workflows with GitHub Actions*. Packt Publishing.
- Räsänen, T. & Sorsa, J.-A. (2010). StanForD 2010 – Metsäkoneiden uusi tiedonsiirtostandardi. *Metsätehon tulosalvosarja 11/2010*. Haettu 8.5.2023 osoitteesta <https://www.metsateho.fi/stanford-2010-metsakoneiden-uusi-tiedonsiirtostandardi/>
- Skogforsk. (2023). StanForD. Haettu 8.5.2023 osoitteesta <https://www.skogforsk.se/english/projects/stanford/>
- Snowflake. (2023). Semistructured Data. Haettu 8.5.2023 osoitteesta <https://www.snowflake.com/guides/semi-structured-data-101>
- Sorsa, J.-A. (2002). Metsätehon raportti 131: Towards StanForD-XML. Haettu 8.5.2023 osoitteesta <https://www.metsateho.fi/towards-stanford-xml/>
- Stora Enso. (2022). Strategia. Haettu 10.2.2022 osoitteesta <https://www.storaenso.com/fi-fi/about-stora-enso/our-strategy>
- Sullivan, D. (2015). *NoSQL for Mere Mortals*. Pearson Education, Limited.
- Sychev, I., Abdelkader, A., Kozak, W., Bonett, R. & Filzek, F. H. P. (2020). Closed Loop Benchmark for Timeseries Databases. *IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*.
- UPM-Kymmene. (2023). Kasvun moottori. Haettu 8.5.2023 osoitteesta <https://www.upmpasodelostoros.com/fi/>
- Wiese, L. (2015). *Advanced Data Management : For SQL, NoSQL, Cloud and Distributed Databases*. De Gruyter,. <https://doi.org/10.1515/9783110441413>
- W3C. (2001). XML Schema. Haettu 10.5.2023 osoitteesta <https://www.w3.org/XML/Schema>
- W3C. (2017a). XSL Transformations (XSLT) Version 3.0. Haettu 10.5.2023 osoitteesta <https://www.w3.org/TR/xslt-30/>
- W3C. (2017b). XML Path Language (XPath) 3.1. <https://www.w3.org/TR/xpath-31/>
- W3C. (2017c). XQuery 3.1: An XML Query Language <https://www.w3.org/XML/Query/>
- Yoo, S., Yoo, S., Kim, T., Kim, Y., & Kim, T. (2021). Edge/Fog Computing Technologies for IoT Infrastructure. MDPI - Multidisciplinary Digital Publishing Institute.

Your Europe. (2022). Yleinen tietosuoja-asetus. Haettu 10.5.2023 osoitteesta https://europa.eu/youreurope/business/dealing-with-customers/data-protection/data-protection-gdpr/index_fi.htm