



Vernerri Närhi

# Käyttäjän aktiivisuuden seuraaminen Teams- viestintäsovelluksen kautta

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto -ja viestintäteknikka

Insinöörityö

21.3.2022

# Tiivistelmä

Tekijä:	Vernerinen Närhi
Otsikko:	Käyttäjän aktiivisuuden seuraaminen Teams- viestintäsovelluksen kautta
Sivumäärä:	30 sivua
Aika:	21.3.2022
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Ohjelmistotuotanto
Ohjaajat:	Simo Silander, Lehtori

---

Koronaviruspandemian aikana Metropolia Ammattikorkeakoulu on antanut henkilökuntansa jäsenille useita etätyösuosituksia. Etätyön aikana tapahtuvaan kommunikaatioon käytetään organisaation sisällä laajasti Microsoft Teams - viestintäsovellusta, jonka avulla on mahdollista viestittää, soittaa tai järjestää tapaamisia käyttäjien välillä.

Insinööriyön tarkoitus on kartoittaa Teams-viestintäsovelluksen paljastaman aktiivisuustilan väärinkäytön mahdollisuuksia ohjelmallisen seurannan avulla. Ohjelmallinen seuranta toteutettiin hakemalla käyttäjien aktiivisuustila Teams-viestintäsovelluksen selainversiosta Playwright-testauskirjastoa hyödyntävällä NodeJS-sovelluksella.

Kerätyt tiedot indeksoitiin Elasticsearch-indeksiin Logstash-palvelimen kautta. Elasticsearchin indeksiin kerättyä tietoa visualisoitiin Kibana-visualisointipaneelin avulla. Paneeliin luotiin näkymiä, joilla käyttäjien aktiivisuustietoa voitiin seurata hyvinkin tarkasti.

Insinööriyössä pohditaan projektin aikana tehtyjä havaintoja ja ohjelmallisesti kerätyn tiedon hyödyntämistä mahdollisen hyökkääjän tai haitallisen tahon toimesta.

Avainsanat: Teams, Elasticsearch, Kibana, Logstash, Aktiivisuuden seuranta

## Abstract

Author: Verner Närhi  
Title: Tracking User Activity Through Teams Communications Platform  
Number of Pages: 30 pages  
Date: 21. March 2022  
Degree: Bachelor of Engineering  
Degree Programme: Information and Communication Technology  
Professional Major: Software Engineering  
Supervisors: Simo Silander, Senior Lecturer

---

During the coronavirus pandemic, Metropolia University of Applied Sciences has given several telework recommendations to its staff. Microsoft Teams communication platform is widely used in Metropolia University of Applied Sciences for communication, making calls, or scheduling meetings between users.

The purpose of the thesis is to map the possibilities of the abuse of the activity state revealed by the Teams communication application by means of software tracking. Software tracking was performed by retrieving user activity status from the browser version of the Teams communications platform using the NodeJS application utilizing the Playwright testing library.

The collected data was indexed into the Elasticsearch index through the Logstash server. The data collected for the Elasticsearch index was visualized using the Kibana visualization panel. Views were created in the panel, which allowed very close monitoring of users' activity data.

The thesis considers the observations made during the project and the utilization of the information collected programmatically by a potential attacker or malicious party.

Keywords: Teams, Elasticsearch, Kibana, Logstash, Activity Tracking

## Sisällysluettelo

1	Johdanto	1
2	Käyttäjätietojen kerääminen	2
2.1	People Finder	2
2.2	Web Scraper	4
3	Teams-aktiivisuuden seuranta	8
3.1	Teams	9
3.2	Teams-aktiivisuustilaa lukeva sovellus	13
4	Kerätyn aktiivisuustiedon visualisointi	17
4.1	Logstash ja datan indeksointi Elasticsearchiin	18
4.2	Kibana	20
5	Sovelluksien kontittaminen	25
6	Tulokset ja pohdinta	26
6.1	People Finderin henkilötiedot	26
6.2	Teams-aktiivisuuden kerääminen	27
6.3	Kehityskohteet ja tavoitteiden saavuttaminen	28
7	Yhteenveto	28
	Lähteet	30

## Lyhenteet

- POC: *Proof of concept*. Yksinkertainen toteutus, jolla todennetaan ratkaisun tai toteutuksen toimivuus.
- URL: *Uniform Resource Locator*. Sisältää tiedon hakemiseen tarvittavan protokollan ja palvelimen osoitteen. Esim <http://peoplefinder.metropolia.fi>.
- CSS: *Cascading Style Sheets*. Verkkosivuille kehitetty tyyliohje, jolla voidaan muokata sivun ulkoasua.
- CSV: *Comma-separated values*. Tiedostomuoto, jossa taulukkomuotoisia tietoja tallennetaan tekstitiedostoon.
- ELK-pino: *Elasticsearch, Logstash ja Kibana* muodostavat yhdessä kokonaisuuden, jolla on helppo tallentaa ja analysoida Elasticsearch-hakumootoriin tallennettua tietoa.
- HTTP: *Hypertext Transfer Protocol*. Protokolla, jota selaimet käyttävät tiedonsiirtoon.
- JSON: *JavaScript Object Notation*. Avoimen standardin tiedostomuoto, jota käytetään tiedonvälitykseen.

## 1 Johdanto

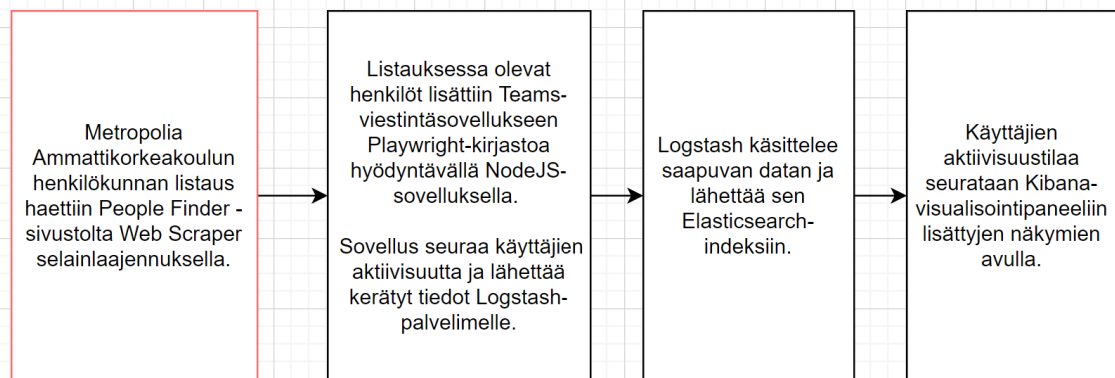
Koronaviruksen aiheuttaman globaalin pandemian aikana työpaikoilla on siirrytty laajalti koko- tai osa-aikaiseen etätöskentelyyn. (Blomqvist, Sivunen and Vartiainen, n.d.) Metropolia Ammattikorkeakoulun organisaation sisäisessä viestinnässä on annettu useita etätösuosituksia pandemian aikana. Etätöiden aikana työntekijöiden välinen kommunikaatio hoidetaan pääasiassa sähköisten viestintäsovellusten kautta.

Microsoft Teams on Metropolia Ammattikorkeakoulussa yleisessä käytössä, ja sovellus on valmiiksi asennettuna kampuksilla sijaitsevilla yleiskäyttöisissä tietokoneissa ja työntekijöille jaettavissa kannettavissa tietokoneissa. Teams-viestintäsovellus käynnistyy automaattisesti tietokoneen käynnistymisen yhteydessä. Teams seuraa päällä ollessaan käyttäjän aktiivisuutta tietokoneella tai muilla laitteilla, joihin se on asennettu ja näyttää aktiivistilan muille käyttäjille organisaation sisällä. Koodaamalla sovelluksen, joka seuraa käyttäjien aktiivisuustilaa on mahdollista tarkkailla henkilön työaika ja aktiivisuutta työpäivän aikana.

Insinööriyö käy läpi POC-toteutuksen, jolla on mahdollista seurata organisaation käyttäjien aktiivisuustietoja. Lisäksi pohditaan projektin aikana tehtyjä havaintoja tiedon hyödyntämisen kannalta. Käyttäjien yksityisyydensuoja on huomioitu insinööriyötä tehdessä niin, että kerättyjä tietoja ei ole mahdollista yksilöidä tiettyyn henkilöön. Tämä on toteutettu muovaamalla käyttäjän yksilöivät henkilötiedot satunnaisiksi merkkijonoiksi ennen tietojen indeksoimista Elasticsearchiin. Lisäksi henkilöiltä, joiden tietoja insinööriyössä on käytetty, on kysytty lupa tietojen hakemiseen ja hyödyntämiseen insinööriyötä varten. Tietoja ei myöskään säilytetä sovelluksen sulkeutumisen jälkeen, vaan ne poistuvat pysyvästi.

## 2 Käyttäjätietojen kerääminen

Tässä luvussa käydään läpi listauksen kerääminen Metropolia Ammattikorkeakoulun henkilökunnan jäsenistä ja heidän perustiedoistaan (kuva 1). Listausta käytetään käyttäjien lisäämiseen Teams-viestintäsovellukseen. Lisäksi käyttäjistä kerättyä tietoa on mahdollista suodattaa ja hakea myöhemmin työssä käytetyn Kibana-visualisointipaneelin kautta. Käyttäjälistauksen kerääminen tapahtui Google Chrome -selaimen Web Scraper -selainlaajennuksella. Selainlaajennuksen avulla kerättiin People Finder -hakusivustolla listatun Metropolia Ammattikorkeakoulun henkilöstöstön tiedot CSV-tiedostoon.



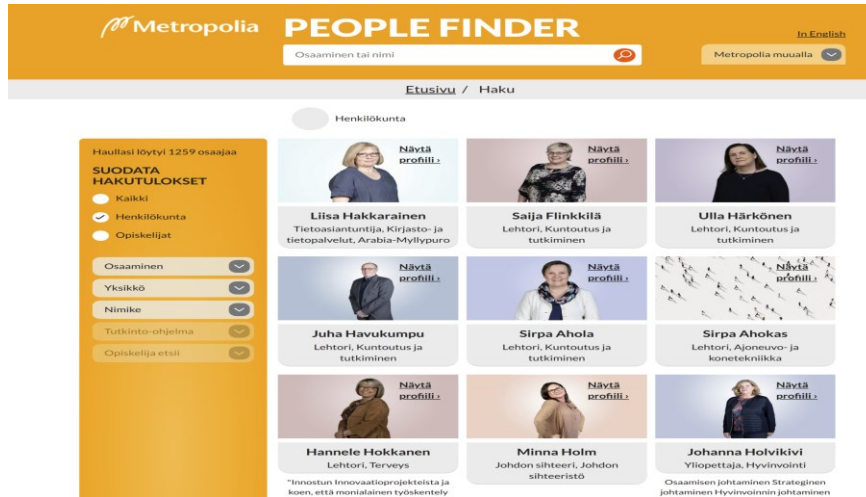
Kuva 1. Työtä kuvaava prosessikaavio. Käyttäjätietojen kerääminen on kuvattu kaaviossa punareunaisessa suorakulmiossa.

### 2.1 People Finder

People Finder (Metropolia People Finder - Henkilöstö ja opiskelijat, 2022) on sivusto, joka näyttää ajantasaisen listan Metropolia Ammattikorkeakoulun työsuhteessa olevasta henkilökunnasta. Opiskelijoiden profiilit eivät näy oletuksena sivustolla, vaan opiskelija joutuu käydä sallimassa erikseen oman

näkyvyytensä sivustolla. Opiskelijoiden tiedot rajattiin pois yksilöinnin estämiseksi.

People Finderin etusivulla valitaan haettavaksi kaikki henkilökunnan jäsenet. Valinnan jälkeen avautuu lista, jossa näkyy kokonaismäärä ja listaus Metropolia Ammattikorkeakoulun henkilökunnan jäsenistä. Käyttäjää on myös mahdollista hakea ja filteröidä osaamisen, yksikön tai nimikkeen perusteella (kuva 2).



Kuva 2. People Finder -sivuston henkilökuntalistaus

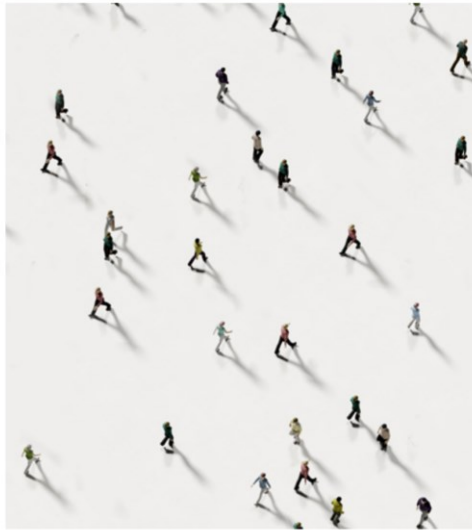
Käyttäjän profiili sisältää oletusarvoisesti henkilön etu- ja sukunimen, tittelin, yksikön, sähköpostin ja kampuksen osoitetiedot (kuva 3). Käyttäjän on mahdollista käydä lisäämässä vapaaehtoisia lisätietoja People Finder -profiiliinsa, mutta insinööriyössä haetaan vain profiileista saatavilla olevat perustiedot yksilöimisen välttämiseksi.



## Vernerin Närhi

Ohjelmistosuunnittelija

Tiedonhallinta- ja järjestelmäpalvelut



Päivitetty 20.11.2020

### Ota yhteyttä

Sähköposti:  
[Vernerin.Narhi@metropolia.fi](mailto:Vernerin.Narhi@metropolia.fi)

Postiosoite:  
PL 4000, 00079 Metropolia

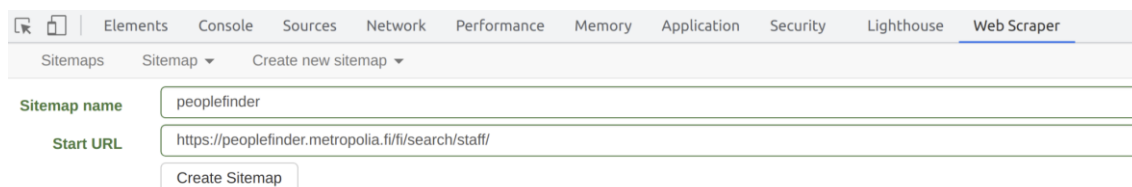
Kampus:  
Myllypurontie 1, 00920  
Helsinki  
[Näytä kartalla](#)

Kuva 3. People Finder -sivustolta avattu henkilökunnan jäsenen profiili.

## 2.2 Web Scraper

Lista organisaation työntekijöistä haettiin People Finder -sivustolta hyödyntämällä Web Scraper -selainlaajennusta Google Chrome -selaimelle. Web Scraper on ilmainen verkkosivustojen sisällön keräämiseen tehty lisäosa, joka tarjoaa selkeän käyttöliittymän ja monipuolisesti ominaisuuksia nopeaan tietojen hakemiseen. Web Scraperiin määritellään haluttu kohdesivusto ja halutut tiedot, jotka sovelluksella haetaan. Web Scraper käy läpi sivuston elementit ja tallentaa näistä kerätyt tiedot CSV-tiedostoon käyttäjän ladattavaksi. Kerätyt tiedot tallennetaan käyttäjän koneelle, joten tiedot eivät päädy kolmannen osapuolen palvelimille.

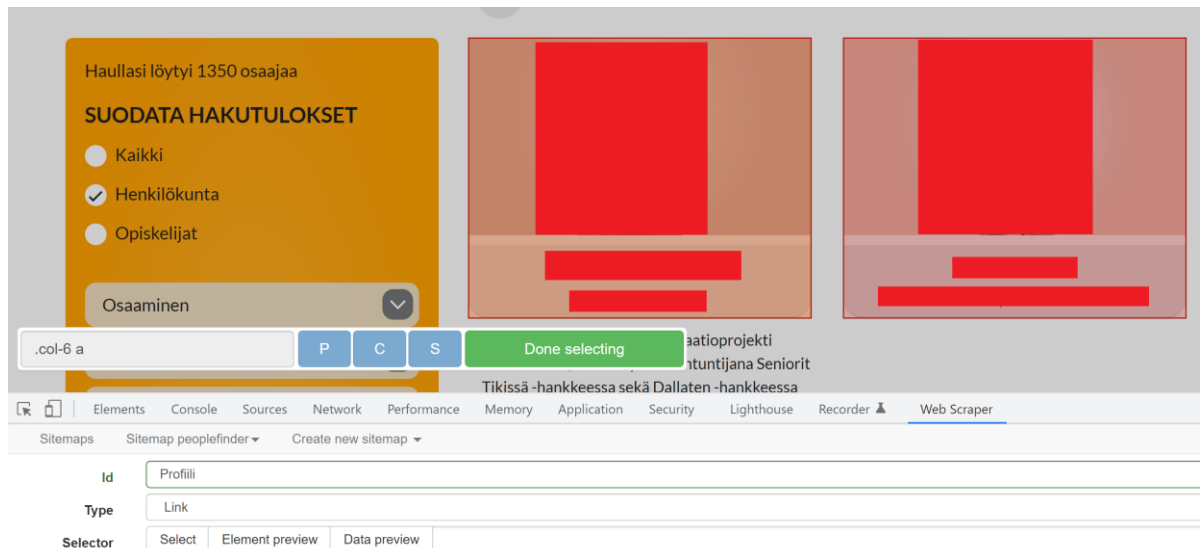
Web Scraper asennetaan Google Chromen sovelluskaupasta. Lisäosan käyttö aloitetaan avaamalla Chrome-verkkoselaimen kehitystyökalut-näkymä ja valitsemalla näkymästä "Web Scraper" -välilehti. Näkymässä luodaan uusi projekti ja syötetään verkkosivuston osoite, jolta tiedon kerääminen aloitetaan (kuva 4). URL-osoite "https://peoplefinder.metropolia.fi/fi/search/staff/" avaa listauksen Metropolian henkilökunnan jäsenistä.



Kuva 4. Web Scraper -selainlaajennuksen näkymä, jossa luodaan projekti ja määritellään URL-osoite, josta tietojen kerääminen aloitetaan.

People Finder -sivuston listauksesta tuli tiedon keräämiseksi avata selainlaajennuksen toimesta jokaisen sivulla näkyvän henkilökunnan jäsenen profiili. Selainlaajennus tallensi jokaiselta avatulta profiilisivulta henkilön julkiset tiedot. Kun sivun kaikki henkilökunnan jäsenten tiedot oli kerätty, selainlaajennus siirtyi listauksen seuraavalle sivulle ja aloitti seuraavan sivun profiilien läpikäymisen. Tätä jatkettiin kunnes kaikki sivuston profiilit on käyty läpi.

Edellä mainitun toiminnallisuuden toteuttamiseksi tietoa keräävät komponentit luotiin Web Scraper -selainlaajennuksen käyttöliittymän "Uusi komponentti" -toiminnallisuudella. Ensimmäisenä luotu Profiili-komponentti avaa listauksesta jokaisen henkilökunnan jäsenen profiilin. Komponentin tyyliksi valittiin "linkki" ja selainlaajennoksen käyttöliittymästä valittiin kursorilla valitsemalla halutut elementit, jonka jälkeen Web Scraper loi elementteihin osoittavan CSS-selektorin (kuva 5).



Kuva 5. Web Scraper-selainlaajennuksen käyttöliittymästä valitaan komponentin luomisen yhteydessä sivuston elementit, joiden tiedot halutaan hakea. Selainlaajennus luo CSS-selektorin, joka osoittaa haluttuun tietoon. Selektori on kuvassa ".col-6 a" ja osoittaa henkilökunnan jäsenen profiilin linkkiin. Kuvasta on peitetty henkilötiedot punaisilla palkeilla.

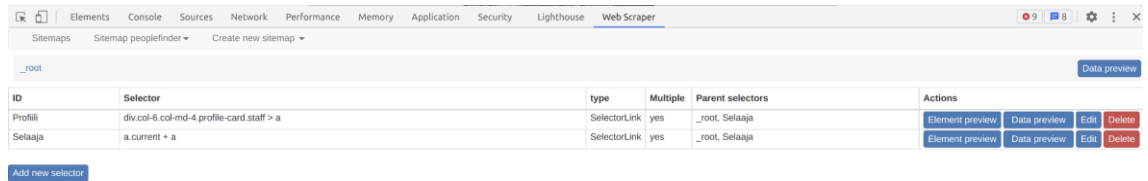
Profiili-komponentille luotiin "Uusi komponentti" -toiminnallisuudella alikomponentteja, jotka hakevat avatusta profiilista käyttäjän nimen, tittelin, yksikön, puhelinnumeron, sähköpostin, kampuksen postiosoitteen ja kampuksen kaupungin ja postinumeron (kuva 6). Komponenttien hakemat tiedot määriteltiin käyttöliittymän kautta ja selainlaajennus loi CSS-selektorit, jotka osoittavat haluttuun tietoon.

ID	Selector	type	Multiple	Parent selectors	Actions
Nimi	.name h2	SelectorText	no	Profiili	Element preview, Data preview, Edit, Delete
Titeli	profile-title strong	SelectorText	no	Profiili	Element preview, Data preview, Edit, Delete
Yksikko	em	SelectorText	no	Profiili	Element preview, Data preview, Edit, Delete
Puhelin	profile-some a	SelectorText	no	Profiili	Element preview, Data preview, Edit, Delete
sähköposti	div.contact-info:nth-of-type(2) a	SelectorText	no	Profiili	Element preview, Data preview, Edit, Delete
postiosoite	div.contact-info:nth-of-type(3) p:nth-of-type(2)	SelectorText	no	Profiili	Element preview, Data preview, Edit, Delete
kampus	div.contact-info:nth-of-type(4) p:nth-of-type(2)	SelectorText	no	Profiili	Element preview, Data preview, Edit, Delete

Kuva 6. Profiili-komponentin alikomponentit, jotka keräävät halutut tiedot profiilista.

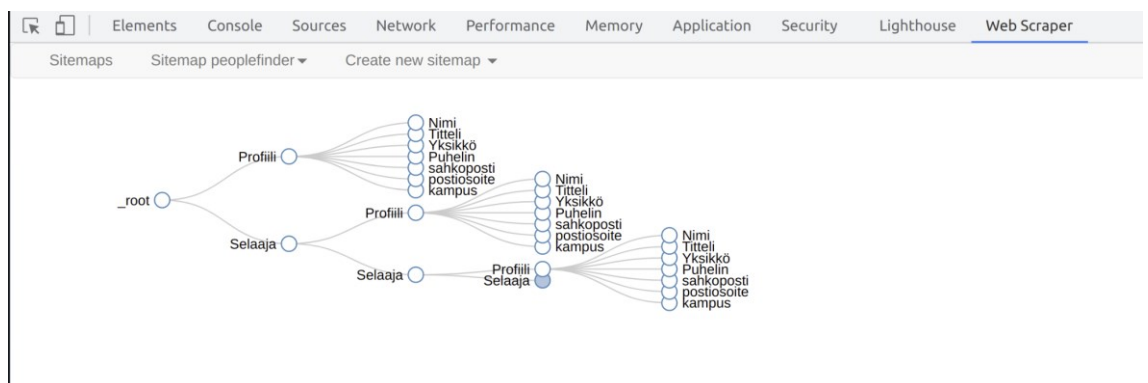
Tämän jälkeen luotiin Selaaja-komponentti, joka avaa People Finder -listauksen profiilien läpikäymisen jälkeen listauksen seuraavan sivun. Komponentin

tyypiksi valittiin ”linkki” ja listauksen seuraavalle sivustolle ohjaavan linkin CSS-selektori haettiin selainlaajennuksen käyttöliittymän avulla (kuva 7).



Kuva 7. Kaksi Webscraper-komponenttia. Profiili- ja Selaaja-komponentit käyvät läpi listauksessa näkyvät profiilit.

Komponenttien luomisen jälkeen valittaessa valikosta ”Browse sitemap” -vaihtoehto nähdään puumuotoinen kaavio luoduista komponenteista ja näiden toiminnasta. Kaavio havainnollistaa komponenttien toiminnan visuaalisessa muodossa. Profiili-komponentti käy läpi jokaisen sivulla olevan profiilin, ja sen alikomponentit hakevat profiilista halutut tiedot. Kun sivun kaikki profiit on käyty läpi, Selaaja-komponentti aukaisee listauksen seuraavan sivun, ja sama toistuu uudelleen, kunnes kaikki People Finder -listauksessa olevat profiilit on käyty läpi (kuva 8).



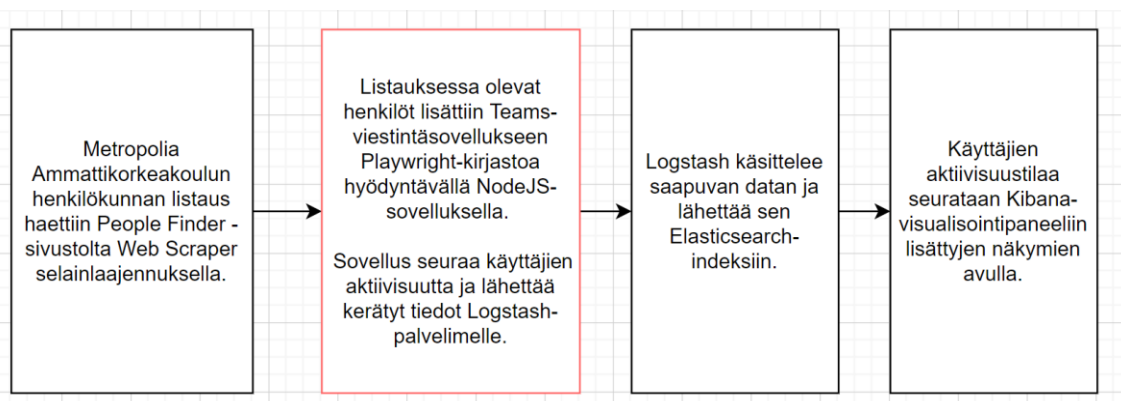
Kuva 8. Web Scraper -selainlaajennuksen ”Browse sitemap” -näkyvä näyttää tietojen keräämisen etenemisen. Profiili-komponentti avaa kaikki listauksen profiilit ja sen alikomponentit hakevat profiilista halutut tiedot. Selaaja-komponentti avaa listauksen seuraavan sivun ja tätä jatketaan, kunnes listauksen kaikki sivut on käyty läpi.

Komponenttien luomisen jälkeen valikosta valitaan ”kerää dataa” -vaihtoehto. Painettaessa ”aloita”-painiketta aloittaa Web Scraper toimintansa ja kerää saadut tulokset taulukkonäkymään. Kun tiedon kerääminen on suoritettu loppuun, saa kerätyn datan ladattua CSV-tiedostomuodossa käyttäjän koneelle.

Ladattu CSV-tiedosto Metropolia Ammattikorkeakoulun henkilökunnan jäsenistä sisältää selainlaajennuksen lisäämiä kenttiä, joita projektissa ei tarvita. Selainlaajennoksen lisäämät kentät sisältävät ajan, jolloin tieto on haettu ja järjestyksen, jossa profiilit on käyty läpi Web Scraper -selainlaajennuksen toimesta. Projektia varten luotu ”csv\_parser”-niminen bash-skripti poistaa edellämainitut ylimääräiset kentät ja säilyttää käyttäjistä kerätyistä tiedoista nimen, tittelin, yksikön, sähköpostin, osoitteen, postinumeron ja linkin henkilön People Finder -profiiliin.

### 3 Teams-aktiivisuuden seuranta

Luvussa esitellään Teams-viestintäsovelluksen tarjoamaa käyttäjän aktiivisuustilaa ja käydään läpi erilaiset aktiivisuustilat, joita Teams käyttäjistään tarjoaa. Lisäksi käydään läpi Teams-aktiivisuustilaa automaattisesti seuraavan Playwright -testauskirjasto hyödyntävän NodeJS-sovelluksen toiminta (kuva 9).



Kuva 9. Työtä kuvaava prosessikaavio. Teams-aktiivisuuden seuranta on kuvattu kaaviossa punareunaisessa suorakulmiossa.

### 3.1 Teams

Teams-viestintäsovellus seuraa aktiivisesti käyttäjän aktiivisuutta laitteilla, joihin se on asennettuna. Oletuksena käyttäjän aktiivisuustieto näkyy organisaation laajuisesti muille käyttäjille. Käyttäjän on mahdollista asettaa itselleen aktiivisuustila manuaalisesti, mutta manuaalisesti asetettu tila raukeaa automaattisesti yhden - seitsemän vuorokauden kuluessa sen asettamisesta (kuva 10). Tilan raukeamisen jälkeen Teams jatkaa käyttäjän aktiivisuuden seuraamista automaattisesti. (User presence in Teams - Microsoft Teams | Microsoft Docs, 2021.) Oletettavaa onkin, että käyttäjät eivät aktiivisesti vaihda tilaansa itse vaan aktiivisuustila heijastaa käyttäjän toimintaa tietokoneella.

User configured state	Default expiration
Busy	1 day
Do not disturb	1 day
Others	7 days

Kuva 10: Manuaalisesti asetetun aktiivisuustilan kesto tilan asettamisen jälkeen. Kun manuaalisesti asetetun tilan kesto on kulunut loppuun, alkaa Teams seurata käyttäjän aktiivisuutta automaattisesti.

Teams-viestintäsovellus näyttää käyttäjästäan 13 erilaista aktiivisuustilaa, joiden lisäksi aktiivisuustilan perään voi tulla liitteeksi "Out of office", mikäli käyttäjä on asettanut itselleen manuaalisesti ajan, jolloin ei ole tavoitettavissa (taulukko 1).

Taulukko 1. Teams-viestintäsovelluksen aktiivisuustilat ja niiden selitteet

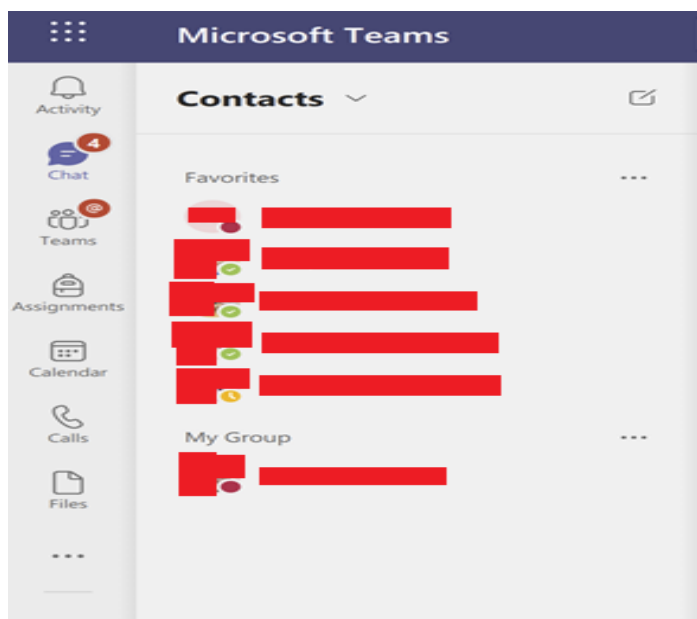
Tila	Selite
Available (tavoitettavissa)	Käyttäjä on aktiivinen koneella ja tavoitettavissa
Busy (kiireinen)	Käyttäjän on mahdollista valita busy-tila manuaalisesti.
In a call (puhelussa)	Käyttäjä on puhelussa toisen Teams-käyttäjän kanssa
In a meeting (tapaamisessa)	Käyttäjä on Teams-tapaamisessa tai Office-kalenterin kautta tulleessa tapaamisessa.
Do not disturb (älä häiritse)	Käyttäjä voi valita tilan halutessaan rauhallista työskentelyaikaa.
Presenting (esittää sisältöä)	Käyttäjä jakaa koneensa ruutua eli esittää sisältöä muille puhelussa tai tapaamisessa.
Focusing (keskittyy)	Käyttäjä on varannut aikaa keskittymiselle MyAnalytics tai Insign-kalentereissaan.
Away (poissa)	Käyttäjä on poissa koneelta tai epäaktiivisena
Away. Last seen {aika} (poissa. Nähty viimeksi {aika})	Käyttäjä on epäaktiivisena. Statuksessa näkyy milloin käyttäjä on viimeksi ollut aktiivisena.

Be right back (palaan kohta takaisin.)	Käyttäjä voi merkata manuaalisesti tilakseen kohta takaisin.
Offline (kirjautunut ulos)	Käyttäjä näkyy kirjautuneena ulos kun käyttäjä ei ole ollut kirjautuneena millään laitteellaan muutaman minuutin ajan.
Status unknown (tila ei tiedossa)	Käyttäjän tila ei ole tiedossa
Out of office (poissa toimistosta)	Käyttäjä voi asettaa poissa toimistosta-tilan manuaalisesti

Käyttäjän tila näkyy Teams-viestintäsovelluksessa aktiivisena, kun käyttäjä antaa syötteitä aktiivisesti tietokoneen näppäimistöllä ja hiirellä. Tila vaihtuu "poissa"-tilaksi, kun käyttäjä on hetken epäaktiivisena tai kun tietokone menee lepotilaan. Käyttäjän tila muuttuu myös sen mukaan, onko käyttäjä puhelussa toisen Teams-käyttäjän kanssa tai onko käyttäjä Outlook-kalenterin kautta tullessa tapaamisessa. Tämän automaattisesti vaihtuvan aktiivisuustilan seuraaminen mahdollistaa käyttäjän aktiivisuuden ja jopa työajan seuraamisen.

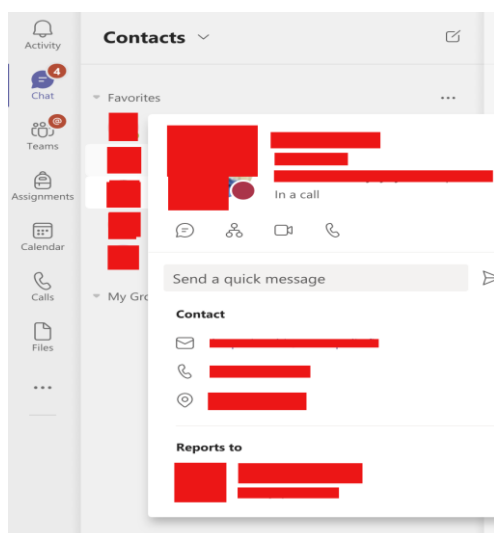
Useamman käyttäjän aktiivisuuden seuraaminen onnistuu käyttöliittymästä "suosikit"-näkyvän kautta. Teams-viestintäsovelluksen selainkäyttöliittymästä avataan viestit-välilehdeltä kontaktit-listaus. Lisäämällä käyttäjiä näkymässä olevaan suosikit-listaan saadaan näkyviin usean käyttäjän aktiivisuustiedot samalla kertaa (kuva 11). Suosikit-listaan mahtuu rajallinen määrä käyttäjiä, mutta rajoituksen pystyy kiertämään lisäämällä ja poistamalla käyttäjiä erissä.





Kuva 11. Teams-viestintäsovelluksen suosikit-näkymä, josta näkyy listattujen käyttäjien aktiivisuustila. Kuvasta on peitetty henkilötiedot punaisilla palkeilla.

Yksittäisen käyttäjän aktiivisuustiedot saa näkyviin, kun kursorin vie listauksessa henkilön profiilin päälle. Aukeavasta ikkunasta näkyy käyttäjän perustietojen lisäksi käyttäjän aktiivisuustila (kuva 12).



Kuva 12. Kursorin vieminen henkilön profiilin päälle avaa ikkunan, joka näyttää henkilön perustietojen lisäksi käyttäjän aktiivisuuden. Kuvasta on peitetty henkilötiedot punaisilla palkeilla.

Käyttäjän aktiivisuuden sisältävä elementti on mahdollista hakea sivulta CSS-selektoria hyödyntämällä. Selektori haettiin manuaalisesti Chrome-selaimen ChroPath-lisäosalla. (ChroPath - Chrome Web Store.) Klikkaamalla hiirellä elementtiä näyttää ChroPath elementin CSS-selektorin (kuva 13). Käyttäjän statukset ovat löydettävissä CSS-selektorilla: "skype-status > span.ts-skype-status.visible". CSS-selektoria hyödynnetään Playwright-testauskirjastoa hyödyntävässä NodeJS-sovelluksessa haettaessa käyttäjien aktiivuuustietoja ohjelmallisesti.



Kuva 13. ChroPath-lisäosa näyttää elementin CSS-selektorin.

### 3.2 Teams-aktiivisuustilaa lukeva sovellus

Playwright on monipuolinen usealla ohjelmointikielellä toimiva end-to-end-testauskirjasto, joka tukee testien automatisointia useilla selaimilla ja sisältää laajan määrän ominaisuuksia testien luomiseksi. (Playwright: Fast and reliable end-to-end testing for modern web apps, 2022.) Playwright-kirjastoa käytettiin projektissa Teams-viestintäsovelluksen selainäkymän komponenttien sisällön lukemiseen ja sivustolla sovelluksen toimesta tehtävien toimintojen automatisointiin. Playwright-testauskirjastoa hyödyntävä Teams-aktiivisuustilaa lukeva sovellus toteutettiin NodeJS-ohjelmointikielellä.

NodeJS-sovellusta käynnistettäessä sovellus lukee ensimmäisenä Web Scaper-selainlaajenuksella haetun CSV-listan People Finderista olevista käyttäjistä ja lukee näiden tiedot taulukkoon rivikohtaisesti (esimerkkikoodi 1).

```
let peoplefinderList = [];

// Hae csv-tiedostossa olevista käyttäjistä
const getPeoplefinderList = () => {
  fs.createReadStream('peoplefinder.csv')
    .pipe(csv({separator:";"}))
    .on('data', (row) => {
      peoplefinderList.push(row);
    })
    .on('end', () => {
      peoplefinderList.sort((a,b) => a.name > b.name ? 1 : -1 )
    });
};
```

Esimerkkikoodi 1. Käyttäjien tuominen NodeJS-sovellukseen Web Scaper-selainlaajenuksella luodusta CSV-tiedostosta. Käyttäjät luetaan riveittäin ja lisätään taulukkoon.

Käyttäjälistauksen hakemisen jälkeen sovellus avaa Teams-viestintäsovelluksen kirjautumissivun Chromium-selaimella ja kirjautuu sisään viestintäsovellukseen käyttäjän määrittelemillä kirjautumistiedoilla (esimerkkikoodi 2).

```
const loginToTeams = async (page) => {

  await page.goto(pageUrl);
  await page.type('input[name="loginfmt"]', teamsUSR);
  await page.click('input[type="submit"]');
  await page.type('input[name="Password"]', teamsPWD);
  await page.click('span[id="submitButton"]');
  await page.click('input[name="DontShowAgain"]');
  await page.click('input[type="submit"]');

}
```

Esimerkkikoodi 2. Ohjelma avaa Teams-viestintäsovelluksen kirjautumissivun ja kirjaa käyttäjän sisään.

Kirjautumistiedot tulevat sovellukseen Docker Composen lukeman .env-konfiguraatitiedoston kautta. Konfiguraatitiedostoon syötetyt tiedot ovat

luettavissa NodeJS-sovelluksessa process.env-muuttujasta (esimerkkikoodi 3). Konfiguraatiossa on myös mahdollista määrittää, kuinka usein sovellus hakee käyttäjien aktiivisuustietoja Teams-viestintäsovelluksesta.

```
const teamsUSR=process.env.teamsUser;  
const teamsPWD=process.env.teamsPassword;  
const logRate=process.env.logRate;  
const delayAfterAction= 10 * 1000;  
const pageUrl = 'https://teams.microsoft.com'  
const backendUrl = 'http://logstash:8787'
```

Esimerkkikoodi 3. Kirjautumistietojen hakeminen process.env-muuttujasta. Samassa yhteydessä ladataan myös määritelmät Logstash-palvelimen URL-osoitteesta ja portista.

Sisäänkirjautumisen jälkeen sovellus poistaa Teams-viestintäsovelluksen suosikit-listauksesta siellä entuudestaan olevat käyttäjät. Suosikit-listausta käytetään käyttäjien aktiivisuuden seurantaan, joten listalla entuudestaan olevien käyttäjien poistamisella käyttäjien varmistetaan, että joukossa ei ole käyttäjiä, joita sovellus ei ole listaukseen lisännyt. Tämän jälkeen CSV-tiedostosta haetut käyttäjät lisätään suosikit-listaan sähköpostiosoitetta hyödyntämällä (esimerkkikoodi 4).

```
// Kirjautuminen  
await loginToTeams(page);  
// Kontaktien poisto  
await removeFriends(page);  
// Kontaktien lisääminen  
await addFriends(page);
```

Esimerkkikoodi 4. Kirjautuminen, käyttäjien poisto suosikit-listalta ja uusien profiilien lisääminen suosikit-listaan.

Kun käyttäjät on lisätty suosikit-listaan sovellus alkaa kerätä lisättyjen käyttäjien läsnäolotietoja Teams-viestintäsovelluksesta ennaltamääritellyn aikamääreen välein. Käyttäjien aktiivisuustilat eivät päivity näkymässä automaattisesti muutaman minuutin jälkeen, joten sovellus avaa näkymän uudestaan joka kerta, kun käyttäjien aktiivisuustietoja haetaan (esimerkkikoodi 5). Käyttäjän aktiivisuustiedot päivittyvät vasta muutaman sekunnin kuluttua näkymän avaamisesta, joten sovellus odottaa aktiivisuustilan päivittymistä ennen tietojen hakemista.

```

while (true) {
  // Mene chat-näkymään
  await goToChat(page);

  // Hae kontaktien läsnäolotiedot
  await crawlData(page);
  // Odota lokitusväli
  await page.waitForTimeout(logRate * 1000);
}

```

**Esimerkkikoodi 5.** Sovelluksen toimintalogiikka on pohjimmiltaan hyvin yksinkertainen. Viestit-näkymä avataan, odotetaan aktiivisuustietojen latautumista, haetaan aktiivisuustilat ja odotetaan ennaltamääritetty aikamääre, jonka jälkeen tiedot haetaan uudestaan.

Sovellus hakee Teams-viestintäsovelluksen suosikit-listasta käyttäjän läsnäolotiedot sisältävät elementit. Elementeistä haetaan käyttäjän nimi ja aktiivisuustila (esimerkkikoodi 6.) Sovellus yhdistää kerätyt tiedot aiemmin kerättyyn People Finder -sivustolta haettuun dataan.

```

const contactList = await page.$$eval('li.group-list-item.left-rail-
item-hover', (favourites) => {
  return favourites.map(person => {

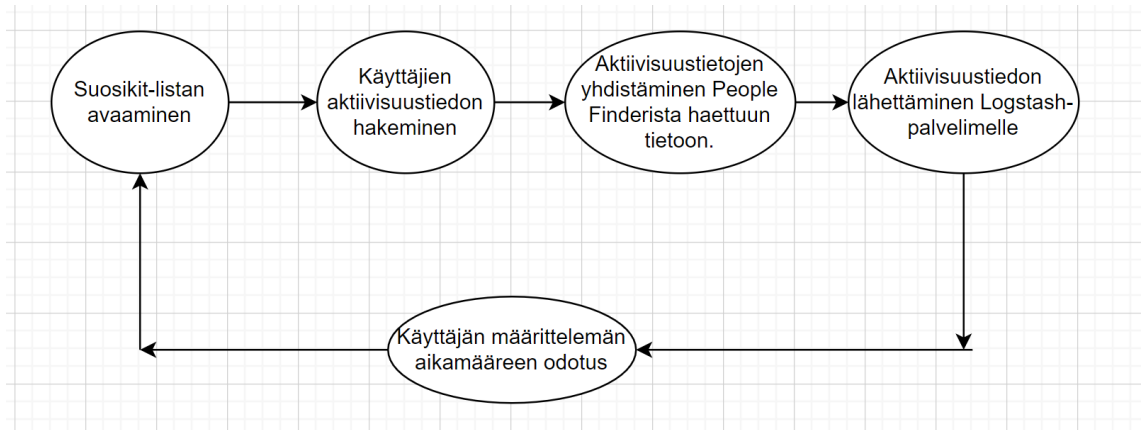
    const name = person.querySelector('span.truncate.truncate-
name.type-ahead-hint');
    const status = person.querySelector('skype-status > span.ts-
skype-status.visible');

    const toName = (element) => element.innerText.trim();
    const toStatus = (element) => element.getAttribute("title");
    return {
      fullName: toName(name),
      status: toStatus(status)
    };
  });
});

```

**Esimerkkikoodi 6.** Suosikit-listauksesta haetaan käyttäjän nimi ja status CSS-selektoreita hyödyntämällä.

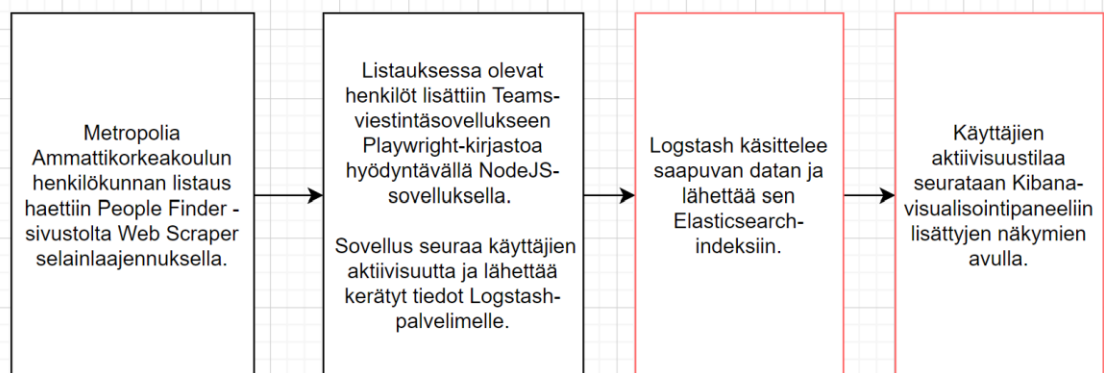
Kerätty data lähetetään sovelluksesta Axios-kirjastoa hyödyntämällä JSON-muodossa POST-requestina ennaltamääritettyyn osoitteeseen Logstash-palvelimelle. Käyttäjän aktiivisuustietojen kerääminen ja lähettäminen Logstash-palvelimelle on havainnollistettu seuraavassa prosessikaaviossa (kuva 14).



Kuva 14. Prosessikaavio, josta ilmenee käyttäjän aktiivisuustiedon kerääminen ja aktiivisuustietojen lähettäminen Logstash-palvelimelle.

#### 4 Kerätyn aktiivisuustiedon visualisointi

Insinöörityössä ohjelmallisesti kerättyä Teams-viestintäsovelluksen käyttäjien aktiivisuustietoa haluttiin esittää graafisessa muodossa mahdollisimman selkeästi (kuva 15). Lisäksi tiedon hakemisen ja rajaamisen piti onnistua käyttöliittymän kautta. Tiedon keräämiseen, hakemiseen ja visualisointiin valittiin edellä mainituista syystä ELK-pino. ELK-pino koostuu kolmesta sovelluksesta: Logstashista, Elasticsearchista ja Kibanasta. ELK-pinoa käytetään yleisesti lokitiedostojen keräämiseen, analysointiin ja graafiseen esittämiseen, mutta se soveltuu erinomaisesti projektissa kerätyn aktiivisuustiedon havainnollistamiseen.



Kuva 15. Työtä kuvaava prosessikaavio. Kerätyn aktiivisuustiedon visualisointi on kuvattu kaaviossa punareunaisissa suorakulmioissa.

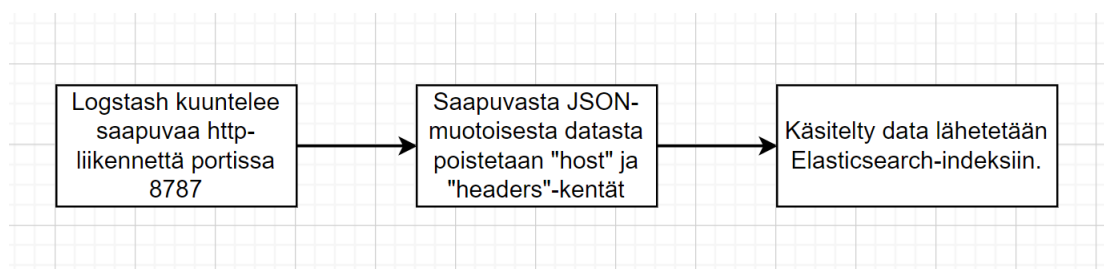
Logstash on lokitiedostojen käsittelyyn tarkoitettu sovellus, joka muokkaa ja suodattaa sille lähetetyt lokitiedostot haluttuun muotoon Elasticsearch-indeksiin viemistä varten.

Logstashin käsittelemä tieto tallennetaan Elasticsearch-indeksiin. Elasticsearch toimii kerätyn tiedon hakumoottorina, jonka avulla indekseihin tallennettua tietoa voi hakea ja analysoida.

Kibana on visualisointipaneelinäkymä Elasticsearchiin. Kibanaa hyödyntämällä on mahdollista luoda dataa visualisoivia näkymiä. Näkymien luominen on nopeaa ja niiden avulla saa havainnollistettua monimutkaistakin dataa helposti omaksuttavassa muodossa.

#### 4.1 Logstash ja datan indeksointi Elasticsearchiin

Logstash käsittelee nodeJS-sovelluksen lähettämän JSON-muotoisen käyttäjien aktiivisuustiedon ja lähettää sen Elasticsearchiin indeksoitavaksi (kuva 16). Käyttäjän tiedot ovat yhdistelmä People Finderista kerättyä tietoa ja Teams- viestintäsovelluksesta kerättyä aktiivisuustietoa (taulukko 1).



Kuva 16. Logstash-kuuntelee saapuvaa HTTP-liikennettä. Saapuva JSON-muotoinen data käsitellään ja lähetetään Elasticsearch-indeksiin.

Taulukko 1: Käyttäjistä kerätyt tiedot

Parametri	Selite
-----------	--------

logTime	aika, jolloin Logstash on vastaanottanut käyttäjän aktiivisuustiedon
fullname	käyttäjän kokonimi
email	käyttäjän sähköposti
statusNumericCode	käyttäjän aktiivisuus numeerisessa muodossa
status	käyttäjän aktiivisuus sanallisessa muodossa
postNumber	käyttäjän kampuksen postinumero ja kaupunki
unit	käyttäjän organisaatioyksikkö
link	linkki käyttäjän peoplefinder-profiiliin
title	käyttäjän titteli organisaatiossa

Logstashin toiminta määritellään konfiguraatitiedostossa, jossa määritellään, mistä data haetaan, miten sitä käsitellään ja mihin valmis data lähetetään. Logstash kuuntelee NodeJS-sovelluksesta saapuvaa HTTP-liikennettä ja käsittelee pyynnön mukana saapuvan JSON-muotoisen datan. Datasta filteröidään Logstashin ”mutate”-ominaisuudella pois ”host” -ja ”headers”-kentät, sillä kenttien sisältämästä datasta ei ole hyötyä aktiivisuuden seurannassa. Ulostuloksi määritellään Elasticsearch-palvelimen osoite, portti, kirjautumistiedot ja indeksin nimi, johon käsitelty data lähetetään (esimerkkikoodi 7).



```

input {
  http {
    id => "HTTP_ID"
    codec => json{}
    port => 8787
  }
}

filter {
  mutate { remove_field => [ "host", "headers" ] }
}

output {
  elasticsearch {
    hosts => "elasticsearch:9200"
    user => "elastic"
    password => "changeme"
    index => "stalklist"
    ecs_compatibility => disabled
  }
  stdout {
    codec => rubydebug
  }
}

```

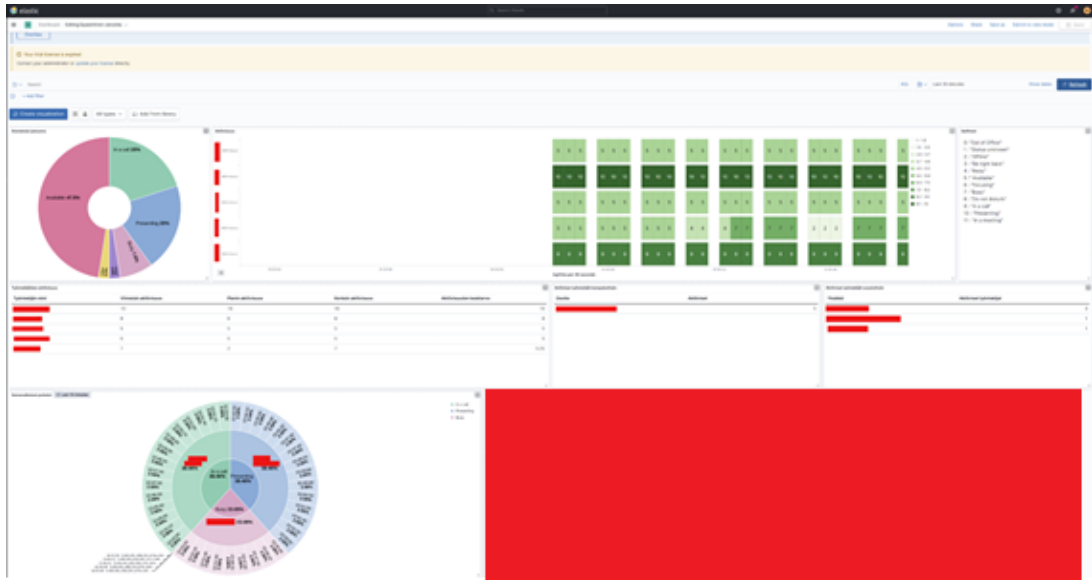
Esimerkkikoodi 7. Logstash-konfiguraatio. Logstash kuuntelee saapuvaa HTTP-liikennettä. Saapuvan pyynnön mukana saapuvasta JSON-muotoisesta datasta poistetaan kentät "host" ja "headers", jonka jälkeen data lähetetään Elasticsearch-palvelimelle indeksoitavaksi.

## 4.2 Kibana

Kibana on käyttöliittymä Elasticsearch-hakumootorille. Kibanaan on mahdollista luoda näkymiä ja paneeleita, joilla Elasticsearch-indeksiin tallennettua dataa on helppo visualisoida käyttäjälle. Kibanan käyttö aloitetaan valitsemalla haluttu indeksi Elasticsearchista ja luomalla paneeli, johon lisätään yksittäisiä näkymiä. Paneelista on mahdollista valita, miltä aikaväliltä kerättyä dataa tarkastellaan. Tämä mahdollistaa hyvin tarkan tunti- tai päiväkohtaisen käyttäjän aktiivisuuden seuraamisen. On myös mahdollista rajoittaa näytettäviä käyttäjiä yksikön, osoitteen, tittelin, nimen tai minkä tahansa muun kerätyn parametrin perusteella.

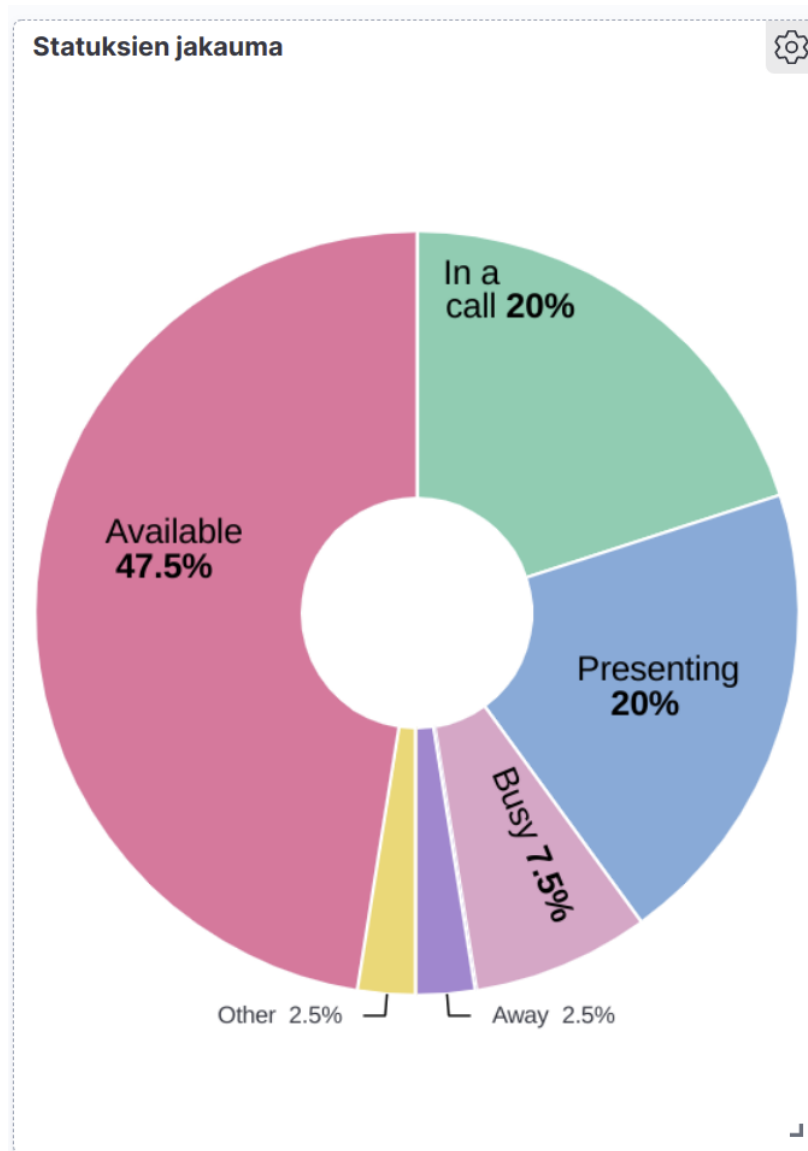
Kaikki Kibanaan luodut paneelit ja näiden sisältämät näkymät tallentuvat Elasticsearch-indeksiin, joten näkymien luomisen jälkeen tallennettu indeksi sisältää nämä näkymät. Projektiin toteutettiin yksi paneeli. Paneeli sisältää 6 näkymää, jolla esitellään erilaisia tapoja hyödyntää kerättyä tietoa (kuva 17).

Käyttäjän on mahdollista visualisoida kerättyä dataa myös muilla kuin insinööriyössä esitellyillä tavoilla luomalla omia näkymiä.



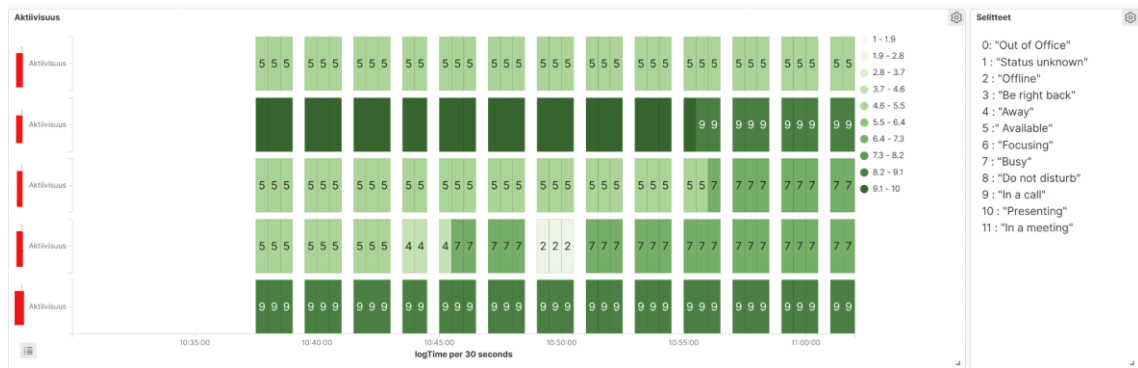
Kuva 17. Paneeli sisältää kaikki käyttäjien seuraamiseen luodut näkymät. Kuvasta on peitetty henkilötiedot punaisilla palkeilla.

Ensimmäinen paneeliin luoduista näkymistä sisältää yleiskatsauksen valittujen käyttäjien aktiivisuuden jakautumisesta. Näkymä kerää käyttäjien statukset ja näyttää näiden prosentuaalisen osuuden kaikista statuksista ympyrädiagrammissa. Näkymä ei itsessään mahdollista kovin tarkkaa statuksien seuranta, mutta sen avulla on helppo luoda yleiskatsaus valitun käyttäjäjoukon aktiivisuudesta (kuva 18).



Kuva 18. Yleiskatsaus käyttäjien aktiivisuudesta.

Toinen paneeliin luotu näkymä on jo huomattavasti tarkempi. Y-akselilla vasemmassa reunassa lukee seuratun käyttäjän nimi ja X-akselilla on käyttäjän numeerinen aktiivisuus tietyllä ajanhetkellä. Taulukon oikeassa reunassa on selitteet aktiivisuuden numeerisille arvoille. Näkymän avulla on helppo seurata yksittäisen käyttäjän aktiivisuutta työpäivän aikana ja jopa saada selville käyttäjän päivittäinen työaika ja tauot. X-akselilla olevan aktiivisuuspalkin väri vaihtelee myös käyttäjän aktiivisuuden mukaan (kuva 19).



Kuva 19. Näkymä mahdollistaa käyttäjän aktiivisuuden seuraamisen valitulla ajan hetkellä. Kuvasta on peitetty henkilötiedot punaisilla palkeilla.

Kolmas näkymä näyttää käyttäjän nimen ja käyttäjän valittuna ajanjaksona aktiivisena vieretyn ajan. Näkymä myös kertoo käyttäjän viimeisimmän aktiivisuuden, seurantajakson pienimmän aktiivisuuden, korkeimman aktiivisuuden ja aktiivisuuden keskiarvon. Näkymää on helppo käyttää etsimään epäaktiivisimmat tai aktiivisimmat työntekijät valitulta aikaväliltä (kuva 20).

Työntekijöiden aktiivisuus				
Työntekijän nimi	Viimeisin aktiivisuus	Pienin aktiivisuus	Korkein aktiivisuus	Aktiivisuuden keskiarvo
[Redacted]	10	10	10	10
[Redacted]	9	9	9	9
[Redacted]	5	5	5	5
[Redacted]	5	5	5	5
[Redacted]	7	2	7	5.385

Kuva 20. Käyttäjän aktiivisuuden numeeriset arvot taulukossa. Kuvasta on peitetty henkilötiedot punaisilla palkeilla.

Neljäs näkymä mahdollistaa yksikön aktiivisten työntekijöiden määrän seuraamisen halutulla aikavälillä. Näkymä listaa vasemmanpuoleisessa listauksessa yksikön ja näyttää sen sillä hetkellä aktiiviset työntekijät. Näkymän avulla voi seurata yksiköiden kokoa ja aktiivisuutta (kuva 21).

Aktiiviset työntekijät osastoittain	
Yksikkö	Aktiiviset työntekijät
Opintoasiainpalvelut	3
Tiedonhallinta- ja järjestelmäpal...	1
Tietohallintopalvelut	1

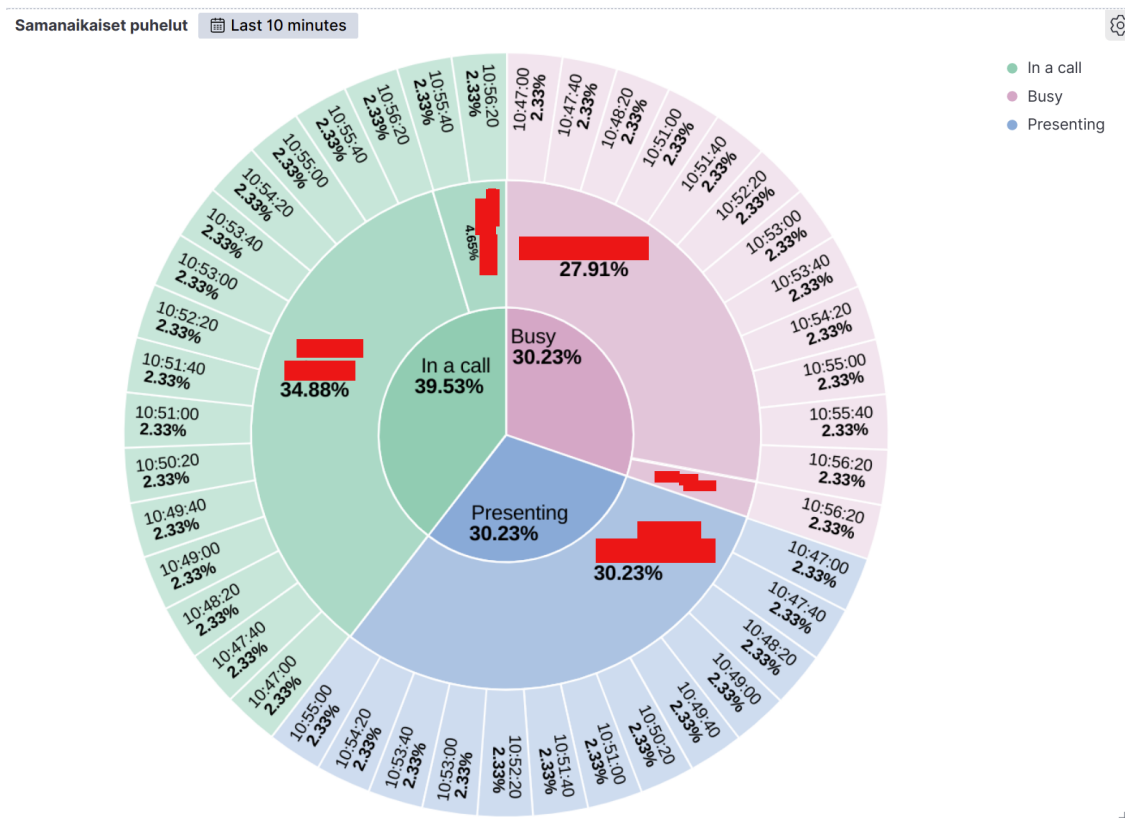
Kuva 21. Aktiivisten työntekijöiden seuraaminen yksiköittäin.

Viides näkymä mahdollistaa käyttäjien aktiivisuuden seuraamisen kampuksittain. Kampuksittain jaottelu kertoo yleissilmäyksen kampuksen aktiivisen henkilökunnan määrästä. Näkymällä voi myös hakea eri yksiköiden työntekijöiden jakautumista kampuksien välillä (kuva 22).

Aktiiviset työntekijät kampuksittain	
Osoite	Aktiiviset
Myllypurontie 1,00920 Helsinki	5

Kuva 22. Käyttäjien aktiivisuudet kampuksittain.

Kuudes näkymä paljastaa kaikkien käyttäjien "Aktiivinen"-statusta aktiivisemmat statukset valitulla ajan hetkellä. Statuksen vieressä lukee kaikkien samaan aikaan kyseisen statuksen omaavien henkilöiden nimet (kuva 23). Näkymän näyttämän tiedon avulla on mahdollista päätellä, ketkä henkilöt ovat samanaikaisesti puhelussa toistensa kanssa tai selvittää tiettyyn kokoukseen osallistuneet henkilöt organisaation sisältä.



Kuva 23. Käyttäjien samanaikaiset aktiivisuudet ympyrädiagrammissa. Kuvasta on peitetty henkilötiedot punaisilla palkeilla.

## 5 Sovelluksien kontittaminen

Käyttäjien Teams-aktiivisuustiedon kerääminen haluttiin tehdä alustariippumattomaksi ja sovelluskokonaisuuden pystyttämisen haluttiin olevan mahdollisimman yksinkertaista. Tämä toteutettiin Docker-tuoteperheellä. Docker tarjoaa alustariippumattoman eristetyn ajoympäristön sovelluksille. Docker Compose mahdollistaa useammasta Docker-kontista koostuvan sovelluskokonaisuuden ja näiden välisen verkon pystyttämisen yhdellä komennolla. Lisäksi sovellusten tarvitsemat parametrit on mahdollista määrittellä yhteen konfiguraatiotiedostoon.

Sovelluksen kontittamisen pohjana hyödynnettiin ELK-Docker-projektia, jonka tarkoituksena on tarjota käyttövalmis pohja ELK-pinon pystyttämiseen. (spujadas/elk-docker: Elasticsearch, Logstash, Kibana 2022.) ELK-Docker-

projektiin kuuluvaan Docker Compose kofiguraatitiedostoon lisättiin Teams-tietojen keräämisessä käytettävän NodeJS-sovelluksen konfiguraatio ja parametrit (esimerkkikoodi 8).

```
playwright:
  build: ./playwright
  stdin_open: true
  tty: true
  volumes:
    - type: bind
      source: ./playwright
      target: /home/project
  environment:
    - teamsUser=${username}
    - teamsPassword=${password}
    - logRate=${logRate}
  networks:
    - elk
  working_dir: /home/project
  depends_on:
    - kibana
```

Esimerkkikoodi 8. Docker-Compose-määrittelytiedostoon lisätty Teams-aktiivisuustietoja keräävän Playwright-sovelluksen konfiguraatio

## 6 Tulokset ja pohdinta

Tässä luvussa pohditaan projektista kerättyjä tuloksia, esiin nousseita huomioita ja miten aktiivisuustietoja olisi mahdollista hyödyntää haitallisen tahon toimesta. Lisäksi pohditaan sovelluksen mahdollista jatkokehitystä.

### 6.1 People Finderin henkilötiedot

People Finder-sivustolta kerättyä tietoa käytetään projektissa henkilölistauksen luomiseen. Sen sisältämä tieto on osa Metropolia Ammattikorkeakoulun avoimuuden kulttuuria. Organisaation sisäisten ja ulkopuolisten henkilöiden on helppo ottaa yhteyttä oikeaan henkilöön, kun henkilökunnan jäsenten yhteystiedot ovat helposti haettavissa.

People Finderin saavutettavuus ulkoisesta verkosta ilman autentikoitumista ja sen tarjoamat henkilötiedot mahdollistavat potentiaaliset kohdennetut

tietojenkalasteluyritykset. Organisaation tietosuojavaltuutetun tai henkilön esimiehen nimissä lähetetty viesti voi saada käyttäjän varomattomaksi verrattuna satunnaisesta lähteestä tulevaan viestiin. Mikäli People Finder halutaan pitää yleisesti saavutettavissa olevana, tulee organisaation käyttäjien tietoturvakoulutukseen panostaa ja erityisesti korostaa saapuvien yhteydenottojen aitouden tarkistamista.

## 6.2 Teams-aktiivisuuden kerääminen

Teams-aktiivisuuden kerääminen vaatii organisaation tunnuksia, joten organisaation ulkopuolisten tahojen on vaikeaa hyödyntää kyseistä tietoa. Tietojen keräämisen vaara kohdistuu erityisesti organisaation sisäiseen väärinkäyttöön. Esimies voi potentiaalisesti seurata alaistensa aktiivisuutta ilman näiden tietoa asiasta ja käyttää keräämäänsä tietoa työntekijöiden eriarvoistamiseen. Toisen työntekijän epäaktiivisuuden tai matalan aktiivisuuden raportointi esimiehelle tai työyhteisölle voi myös asettaa työntekijöitä eriarvoiseen asemaan. Mikäli käyttäjä käyttää työkoneitaan vapaa-ajalla, pystyvät organisaation jäsenet myös seuraamaan työntekijän vapaa-ajalla koneella viettämää aikaa.

Mikäli organisaation ulkopuolinen taho saisi organisaation tunnuksia haltuunsa, voisi käyttäjien aktiivisuustietoa ja asemaa organisaatiossa hyödyntää apuvälineenä sosiaalisessa hakkeroinnissa esimerkiksi kiristyksen keinona yksittäistä työntekijää kohtaan.

Teams-viestintäsovelluksen jakama aktiivisuustieto on oletusarvoisesti koko organisaation jäsenten saatavissa, joten organisaation tulee huomioida väärinkäytön mahdollinen riski ja pohtia aktiivisuustilan näkyvyyden hyötyä verrattuna sen mahdollisiin haittoihin.



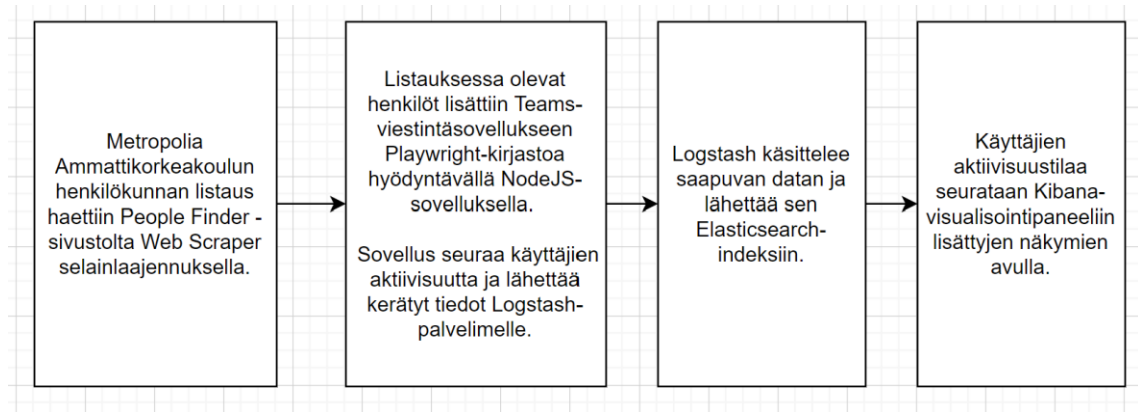
### 6.3 Kehityskohteet ja tavoitteiden saavuttaminen

Projekti ei tule saamaan jatkokehitystä, sillä ilman lupaa tapahtuva aktiivisuustietojen kerääminen ja säilyttäminen rikkoisi törkeästi käyttäjien tietosuojaa ja yksityisyyttä. Projektin tarkoitus oli todistaa Teams-viestintäsovelluksen aktiivisuustietojen potentiaalisen väärinkäytön mahdollisuus ja herättää sillä keskustelua. Tässä projekti saavutti tavoitteensa.

## 7 Yhteenveto

Työssä haettiin Google Chrome -selaimen Web Scraper -selainlaajennuksella People Finder -hakusivustolta Metropolia Ammattikorkeakoulun henkilökunnan jäsenten tiedot CSV-tiedostoon. Playwright-testauskirjastoa hyödyntävä NodeJS-sovellus lisäsi CSV-tiedostossa listatut käyttäjät Teams-viestintäsovelluksen suosikit-listaukseen selainkäyttöliittymän kautta. Tämän jälkeen sovellus seurasi käyttäjien Teams-viestintäsovelluksen aktiivisuustilaa ja lähetti tiedot JSON-muodossa Logstash-palvelimelle.

Logstash lähetti tiedot Elasticsearch-indeksiin. Kibanaan luotujen näkymien avulla Elasticsearch-indeksiin tallennettua Teams-viestintäsovelluksen tarjoamaa aktiivisuustietoa sai visualisoitua kuuden eri näkymän avulla. Koko prosessi on esitetty kuvassa 24. Kibanan näkymien avulla käyttäjän aktiivisuutta ja työaika saati seurattua hyvin tarkasti ilman, että käyttäjä on tietoinen aktiivisuutensa seuraamisesta. Työssä pohdittiin lopuksi Teams-viestintäsovelluksen tarjoaman aktiivisuustiedon väärinkäytön mahdollisuuksia.



Kuva 24. Prosessikaavio havainnollistaa käyttäjien aktiivisuustilan keräämisen vaiheita.

## Lähteet

Metropolia Ammattikorkeakoulu. 2021. Metropolia People Finder – Henkilöstö ja opiskelijat. Viitattu 5.6.2021. Saatavissa <https://peoplefinder.metropolia.fi>.

Blomqvist, K., Sivunen, A. and Vartiainen. 2020. ETÄTYÖ SUOMESSA KORONAVIRUSPANDEMIAN AIKANA. Viitattu 2.1.2021. Saatavissa <https://futuremote.fi/wp-content/uploads/2020/12/Etatyo-Suomessa-koronaviruspandemian-aikaan-2020.pdf>.

Web Scraper. 2021. Making web data extraction easy and accessible for everyone. Viitattu 5.6.2021. Saatavissa <https://webscraper.io>

Microsoft Corporation. 2021. *User presence in Teams - Microsoft Teams* | Microsoft Docs. Viitattu 4.6.2021. Saatavissa <https://docs.microsoft.com/en-us/microsoftteams/presence-admins>.

Chrome Web Store. 2021. ChroPath - Chrome Web Store. Viitattu 4.6.2021. Saatavissa <https://chrome.google.com/webstore/detail/chroPath>.

Playwright. 2021. Playwright: Fast and reliable end-to-end testing for modern web apps. Viitattu 3.6.2021. Saatavissa <https://playwright.dev/>

Github. 2022. spujadas/elk-docker: Elasticsearch, Logstash, Kibana. Viitattu 3.6.2021. Saatavissa <https://github.com/spujadas/elk-docker>.