



Satakunnan ammattikorkeakoulu  
Satakunta University of Applied Sciences

VILLE NIEMI

# **Interaktiiviset Karttasovellukset**

TIETOJENKÄSITTELYN KOULUTUSOHJELMA  
2020

Tekijä(t) Niemi, Ville	Julkaisun laji Opinnäytetyö, AMK	13 huhtikuu 2020
	Sivumäärä 31	Julkaisun kieli Suomi
Julkaisun nimi <b>Interaktiiviset karttasovellukset</b>		
Tutkinto-ohjelma Tietojenkäsittelyn koulutusohjelma		
<p>Tämän opinnäytetyön tavoitteena oli antaa lukijalle tarvittavat tiedot karttasovelluksen valintaan asiakkaalle tehtävään projektiin, jossa tarvitsee piirtää kohteita kartalle ja niistä saa painaessa lisätietoa kohteesta. Vertailtaviksi karttasovelluksiksi valittiin kolme tunnetuinta kartta API:a Googlelta, Leafletilta ja Mapboxilta.</p> <p>Opinnäytetyön teoriaosuus keskittyi työssä vertailtavien karttasovelluksien taustoihin, suosion lähteiden taustoihin sekä karttasovelluksien ominaisuuksiin. Teoriaosuuden luvuissa käytiin lyhyesti luvun kyseisten aiheiden historiaa ja esiteltiin saatuja tietoja näistä. Lisäksi korostettiin keskeisiä ja tärkeitä asioita opinnäytetyölle.</p> <p>Toiminnallisessa osuudessa oli tarkoitus rakentaa testisovellus, jossa otetaan käyttöön vertailtavat karttasovellukset. Luvussa esiteltiin testisovelluksen käyttöympäristöä ja karttasovelluksien käyttöönottoa. Lisäksi esiteltiin saatuja tuloksia.</p> <p>Vertailussa käytiin läpi karttasovelluksista saatuja tuloksia suosiosta, ominaisuuksista ja testisovelluksesta saaduista tehokkuuden luvuista. Tuloksia analysoitiin yleisesti ja tulevaisuuden näkökulmasta oman osaamisen perusteella.</p> <p>Opinnäytetyön lopputuloksena oli tavoite saada esille tarvittavat tiedot karttasovelluksista, joiden perusteella lukija osaa tehdä päätöksen näiden väliltä. Teoriaosuus antaa hyvin suuntaa siihen, miten muitakin sovelluksia kuin karttasovelluksia voi vertailla keskenään. Toiminnallisessa osuudessa tuli hyvin selville tietoa komponenttien käyttöönotosta Reactissa ja testauksessa käytettävistä työkaluista. Lopulta päädyttiin selvään tulokseen mikä karttasovellus sopii mihinkin tilanteeseen.</p>		
Asiasanat API, Google Maps, karttasovellus, Leaflet, Mapbox, ohjelmistorajapinta		

Author(s) Niemi, Ville	Type of Publication Bachelor's thesis	13 April 2020
	31	Language of publication: Finnish
Interactive Map Software's		
Degree programme Degree programme in information technology		
<p>In this thesis object is to give reader necessary information to choose map software for client project. In this project goal is to draw points to map and clicking them gives out information about those points. Chosen map software's for comparing are three most known map API's from Google, Leflet and Mapbox.</p> <p>Thesis theory part focuses on comparable map software's backgrounds, backgrounds of popularity sources and map software's features. In these chapters was short summary about comparable software's and found information was presented about them. Also, important and major parts were highlighted for this thesis.</p> <p>In production part testing application was built for map software's implementation. In this chapter testing environment was presented and implementation. Also produced testing results were shown.</p> <p>In comparing map software's results were reviewed for popularity, features and the results given by built test software. Results were analyzed generally and from perspective of future by writer's own knowledge.</p> <p>End results goal was to give reader enough information about chose map software's so reader can choose between them. Theory part gave good points how to compare other software's than purely made for maps. In product part we got good view about implementing software's in React application and tools to use in testing. In the end clear result was reached about what map software fits any situation.</p>		
Key words API, application programming interface, Google Maps, Leaflet, Mapbox, map software		

# SISÄLLYS

## SANASTO JA LYHENTEET

1 JOHDANTO.....	6
2 TAUSTA .....	7
2.1 Google Maps API.....	7
2.2 Leaflet.....	7
2.3 Mapbox GL JS .....	8
2.4 Tavoitteet .....	8
3 SUOSION DATAN LÄHTEET .....	9
3.1 GitHub .....	9
3.2 Npm Trends.....	10
3.3 Stack Overflow.....	11
4 API-RAJAPINTOJEN OMINAISUUDET.....	12
4.1 Google Maps API.....	12
4.2 Leaflet.....	13
4.3 Mapbox GL JS .....	15
5 TEHOKKUUS.....	17
5.1 Testiympäristö ja testaus.....	18
5.2 Google Maps API.....	19
5.2.1 Testitulokset .....	20
5.3 Leaflet.....	21
5.3.1 Testitulokset .....	22
5.4 Mapbox GL JS .....	23
5.4.1 Testitulokset .....	23
6 VERTAILU .....	25
6.1 GitHub .....	25
6.2 NPM Trends.....	25
6.3 Stack Overflow Trends .....	26
6.4 Ominaisuudet .....	26
6.5 Tehokkuus.....	27
7 YHTEENVETO.....	29
LÄHTEET .....	30

## LIITTEET

## SANASTO JA LYHENTEET

API	Application programming interface, suomeksi ohjelmointirajapinta.
GIS	Global Information System, suomeksi paikkatietojärjestelmä.
KML	Keyhole Markup Language, on XML-merkintä kieli koordinaateille kartastossa.
macOS	Macintosh operating systems, on Applen tietokoneiden käyttöjärjestelmä.
NPM	Node Package Manager, tarkoittaa paketinhallinta järjestelmää JavaScript ohjelmointikielelle.
iOS	iPhone OS, on Applen mobiililaitteiden käyttöjärjestelmä
PubNub	Ohjelmistotalo Californiassa, joka on erikoistunut API-kutsujen välittämiseen.
Qt	Lausutaan ”cute”, on ohjelmistojen kehitysympäristö.
React	Facebookin kehittämä kehitysympäristö sovelluksille.
React Native	On kehitysympäristö mobiili sovelluksille.
SDK	Software Development Kit, suomeksi ohjelmistokehityspaketti.

## 1 JOHDANTO

Interaktiivinen karttasovellus on kartta, joka sallii zoomauksen, spesifien kohteiden tunnistamisen, kyselyjen tekemisen kartassa käytettävästä datasta, raporttien luomisen ja muut keinot käyttämällä tai visualisoimalla valittua informaatiota kartalla. Interaktiivisessa kartassa käytetään GIS (Global Information System), joka mahdollistaa tarkan datapisteen näyttämisen kartalla. Toisin kuin staattisissa kartoissa interaktiiviset kartat sisältävät suuren määrän ominaisuuksia, joilla voi parantaa monimutkaisen datan näkyvyyttä. Näihin ominaisuuksiin kuuluvat esimerkiksi:

- Kyky näyttää tai piilottaa eri kerroksia informaatiota napin painalluksella.
- Tekstilaatikkojen lisäys datapisteisiin, joista saa lyhyen selityksen painamalla.
- Zoomaus ulos tai sisään antaen käyttäjälle tarkennuksen pisteestä tai laajan aluenäkymän.
- Data on mahdollista päivittää nopeasti vaikuttamatta käyttäjään.
- Kartan pisteet on mahdollista linkittää ulkoisiin tiedostoihin kuten kuvat, videot ja graafit.

Ensimmäisenä kaupallista verkko karttapalvelua alkoi tarjoamaan MapQuest yritys vuonna 1996. Alkuperäisinä palveluina olivat karttasovellus ”Interactive Atlas” ja navigaatio palvelu autoille ”TripsQuest”. (MapQuest 1997.)

Tämän opinnäytetyön tehtävänä on antaa tietoa lukijalle sopivista karttasovelluksista, jotta lukija voi tehdä parhaan päätöksen näiden karttasovelluksien väliltä suoritettavaan asiakkaan projektiin. Projektin sovelluksen kartalla on tarkoitus näyttää huollettavia asiakkaan kohteita ja muita kohteiden toiminnallisuuksia. Projektin sovellus tehdään Reactilla TypeScript-ohjelmointikielellä, joten tulen tutkimaan karttasovelluksia Reactin näkökulmasta.

## 2 TAUSTA

Application programming interface (API) käytetään monissakin web-pohjaisissa sovelluksissa. Toisin kuin käyttöliittymä ihmisille API toimii eräänlaisena ohjelmistolta toiselle ohjelmistolle palveluna.

API:n avulla ohjelmoijan ei tarvitse jatkuvasti keksiä pyörää uudelleen. Tämän ongelman voi ratkaista löytämällä sopivan API:n, jolla ohjelmoija välttää joutumasta rakentamaan esimerkiksi oman karttasovelluksen. Oma karttasovellus tarvitsee myös datan taustalle, jotta saisimme edes pienen interaktiivisen karttasovelluksen. Nykyaikana erilaisia API-rajapintoja on lukuisia ja niillä on hyvät ja huonot puolensa. Tästä syystä yritykselle sopivan API:n valinta asiakas projektille on strategisesti tärkeä päätös. (Hoffman 2018.)

### 2.1 Google Maps API

Google julkaisi Google Maps API:n kesäkuussa 2005, joka luotiin yhdistämällä kahden sivuston valmistamat kartastot. Tämä mahdollisti ohjelmistokehittäjille liittää Googlen karttasovelluksen verkkosivustoilleen. Google Maps API oli ilmainen ja käytettävissä ilman rajoituksia kesäkuuhun 2018 asti. Muutosten tullessa voimaan heinäkuun 16. päivä ilmoitettiin palvelun käyttämiseen vaadittavan API-avain linkitetynä Google Cloud-käyttäjätiliin. Jotta API-avaimen saa luotua täytyy käyttäjän luoda tili itsellensä ja antaa pankkikortin tiedot mahdollista laskutusta varten, vaikka käyttäisikin ilmaisversiota. Ilmaisversiota voi käyttää vain sivustolla, jossa on vähemmän kuin 25 000 kävijää päivittäin. Päivittäisen maksimimäärän ylittyessä tarvitsee käyttäjä yritys version Google Maps API:sta. Google Maps API on JavaScript ohjelmointikieli pohjainen kirjasto ja oli käytetyin API 2011. (Rijo 2020.)

### 2.2 Leaflet

Vladimir Agafonkin julkaisi Leafletin 2011, jota hän kehitti työskennellessään Cloud-Madella vuonna 2010. Hän loi Leafletin Web Maps API:n vanhasta koodista, joka oli CloudMaden tarjoama palvelu. Leaflet toimii tehokkaasti kaikilla suurilla tietokone- ja mobiilialustoilla. (Lovelace 2017.)

Nykyään Avoimen lähdekoodin JavaScript ohjelmana Leaflet toimii interaktiivisena karttana mobiiliystävällisille verkkosovelluksille.

### 2.3 Mapbox GL JS

Mapboxin JavaScript pohjainen API tuotiin markkinoille Googlen kartaston haastajaksi. Alun perin Mapbox luotiin Leafletin pohjalta, mutta nykyään heillä on tarjonnassa kokonaan Leafletistä irrotettu Mapbox GL JS.

Mapbox GL JS on osa järjestelmästä riippumatonta Mapbox GL -ekosysteemiä. Tähän ekosysteemiin myös kuuluvat natiivit SDK:t sovelluksille Adroidilla, iOS:llä, macOS:llä, Qt:lla ja React Nativella. Mapbox tarjoaa rakennuspalat mihin tahansa työstettäviin alueominaisuuksiin, joita on esimerkiksi kartat, haku ja navigaatio. (NPM 2020a.)

### 2.4 Tavoitteet

Vaikka tämän opinnäytetyön API:t ovat JavaScriptin pohjalta kirjoitettuja, niin silti kyseiset API:t on valmistettu eri lailla ja toimivat monilla erilaisilla tavoilla. Pyrin tuomaan opinnäytetyössäni esiin aikaisemmin mainittujen API:en positiiviset sekä negatiiviset puolet, joiden avulla lukija pystyy tekemään päätöksen näiden API:en väliltä. Jotta voin saavuttaa tämän, tulen vertailemaan näitä kirjastoja kolmessa eri kategoriassa: suosio, ominaisuudet ja tehokkuus.

Suosiota tulen mittaamaan keräämällä ja analysoimalla dataa isoimmilta ohjelmointiin liittyvistä pilvialustoista, jotka ovat GitHub, Npm trends ja Stack Overflow verkkosivustot. Ominaisuudet tulen keräämään API:en teknisestä dokumentaatiosta sekä tulen käymään läpi mahdollisia rajoituksia ja hinnastoa. API:en tehokkuutta tullaan katsomaan hyvin pienen liiketoimintalogiikan mukaisesti, jokaisen API:n kohdalla samassa sovelluspohjalla. Rakennettua sovellusta tarkastelen tämän jälkeen käyttämällä React developers toolsilla. Sovelluksen koodi tulee olemaan saatavilla GitHubissa.



## 3 SUOSION DATAN LÄHTEET

### 3.1 GitHub

GitHub pilvialustalla ohjelmistokehittäjät voivat säilyttää ohjelmistojensa koodia varmuuskopiona tai jaettavaksi muille ohjelmistokehittäjille. GitHub on tunnettu myös ohjelmistokehityksen projektinhallinnasta, jolla ohjelmointikoodia voi helposti kehittää ja jakaa projektin työryhmän kesken. Ajatellaan meillä olevan runko ohjelmointikoodista ja jokainen kehittäjä työstää tästä rungosta omaa oksaansa. Kun kehittäjä saa oksansa valmiiksi hän voi ilmoittaa sen runkoon liitettäväksi, ennen lopullista liitosta runkoon yleensä työryhmän kokonein kehittäjä tarkastaa oksan. GitHubilla on ollut 2019 toukokuusta lähtien yli 37 miljoonaa käyttäjää sekä yli 100 miljoonaa tietovarastoja. Näistä tietovarastoista ainakin 28 miljoonaa on julkisia, mikä on tehnyt GitHubista suurimman lähdekoodin ylläpitäjän. (Gousios, Vasilescu, Serebrenik & Zaidman 2008, 834.) Taulukossa 1 on esiteltyinä valittujen API:en statistiikkatiedot GitHubista.

	Tähdet	Haarukat	Ongelmat
Leaflet	26 000	4 400	405
Google	4 300	875	184
Mapbox	5 600	1 300	631

Taulukko 1. API:en statistiikkatiedot GitHubista 27.01.2020 (GitHub 2020a; GitHub 2020b; GitHub 2020c).

GitHubissa käyttäjätilin hallitsija voi merkitä tähdillä kiinnostavia tietovarastoja, joiden kehitystä hän haluaa pitää silmällä. Tähdillä merkityistä tietovarastoihin liittyvistä aiheista GitHub tarjoaa materiaalia käyttäjälle esimerkiksi sivuston uutispuolella, jonka aiheet on määritelty käyttäjän kiinnostuksen kohteiden mukaisesti.

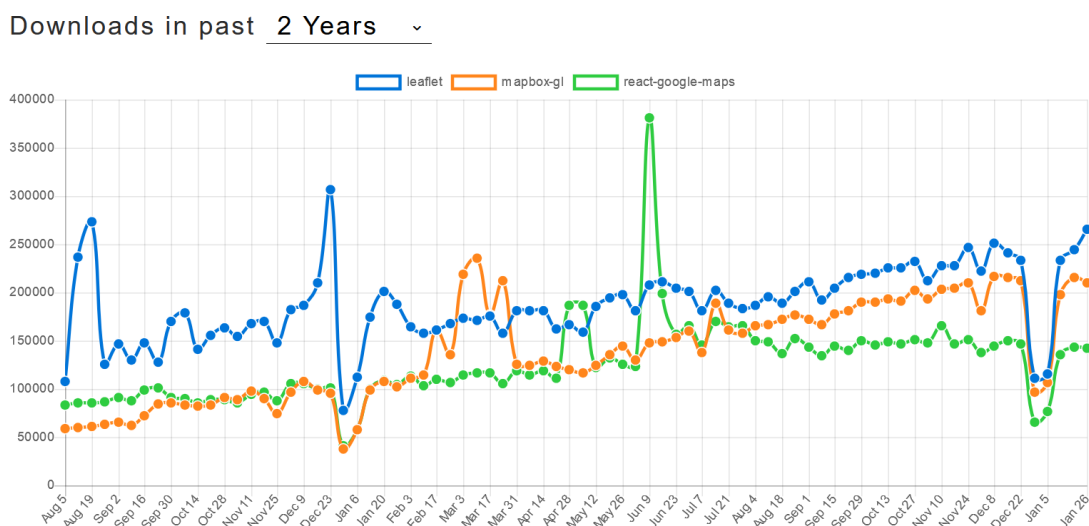
GitHubin käyttäjät voivat halutessaan tukea toisen käyttäjän projektia tai käyttää sitä pohjana omaan projektiin prosessilla, jota kutsutaan haarukoinniksi. ”Haarukan” luominen tekee persoonallisen kopion toisen käyttäjän projektista. Haarukka toimii eräänlaisena siltana alkuperäisen tietovaraston ja tästä tehdyn kopion välillä. Kopiossa tehdyistä muutoksista voi lähettää liitospyynnön alkuperäisen tietovaraston omistajalle,

tämä voi halutessaan lisätä kopiassa olevat muutokset omaan tietovarastoonsa. GitHubissa haarukointi on sosiaalisen ohjelmoinnin ydin. (GitHub 2020d)

Jokaisessa ohjelmistoprojektissa on backlog, jossa on listattuna käyttäjätarinat, ongelmat ja kehityskohteet. GitHubin tietovarastoissa nämä ovat listattuna ongelmat osissa. Ongelmat ovat hyvä tapa pitää kirjalla tehtäviä, parannuksia ja bugeja projektissa. GitHubissa ongelmat toimivat kuin sähköpostit, mutta niitä voi jakaa ja keskustella niistä työryhmän kesken. Ongelmia voi myös pitää hyvänä tapana mitata tietovaraston kiinnostavuutta muille GitHubin käyttäjille, koska muutkin käyttäjät voivat raportoida bugeja tai antaa laajennus- sekä parannusehdotuksia. (GitHub 2020e)

### 3.2 Npm Trends

Node Package Manager (NPM) on maailman suurin ohjelmisto rekisteri. NPM:ää käytetään jokaiselta mantereelta varsinkin avoimenlähdekoodin ohjelmoijien toimesta ohjelmistopakettien lainaukseen ja jakamiseen. Monilla yrityksillä on myös NPM käytössä hallitakseen yksityistä kehitystä. (NPM 2020b.) Kaaviossa 1. näytetään Npm Trends sivustolta dataa kahden vuoden sisältä NPM kautta ladatuista JavaScript-paketeista.

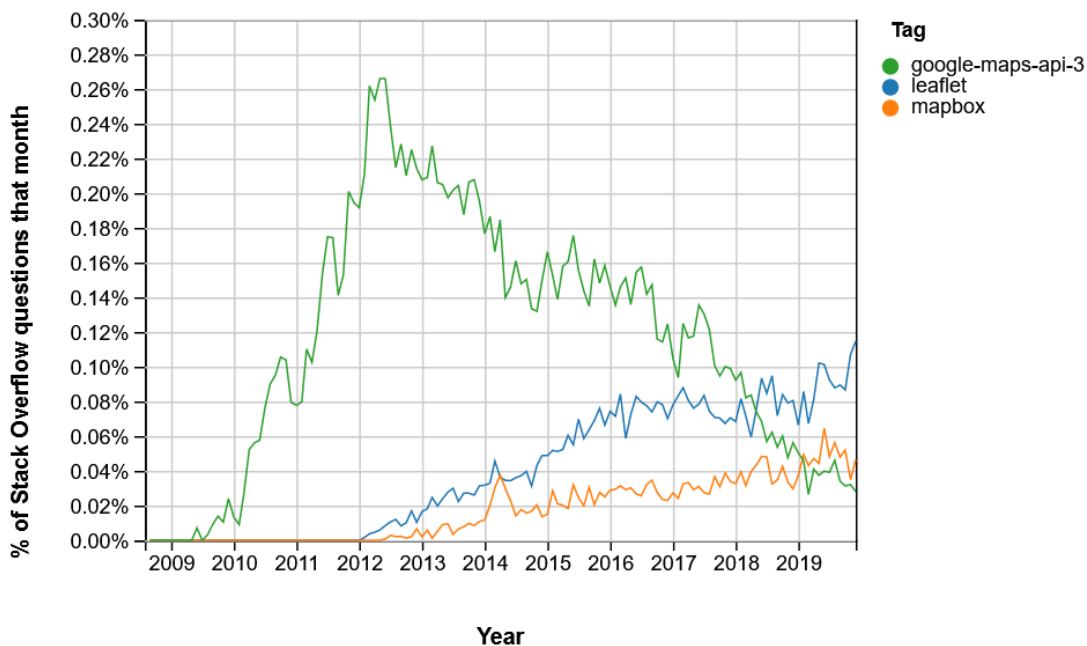


Kaavio 1. Leaflet, Google -ja Mapbox API

### 3.3 Stack Overflow

Vuonna 2008 perustettu Stack Overflow on suurin luotettu online-yhteisö jokaiselle ohjelmoinnista kiinnostuneelle, joka haluaa kehittyä ohjelmoijana, jakaa kokemustansa tai rakentaa uraansa. Joka kuukausi Stack Overflowissa käy enemmän kuin 50 miljoonaa uniikkia vierasta etsimässä ratkaisua ohjelmointiongelmiansa, auttamassa muiden ohjelmoijien ongelmissa, kehittämässä uusia taitoja tai etsimässä mahdollista työpaikkaa. (Stack Overflow 2019.)

Stack Overflow on loistava valinta suuren käyttäjämääränsä ja kokonsa puolesta, kun halutaan tutkia eri ohjelmointikielien, alustojen, pakettien ja kirjastojen kiinnostavuutta globaalilla tasolla ohjelmoijayhteisössä. Kaaviossa 2. on 10 vuoden ajalta kuvaus Google Maps, Leafletin -ja Mapbox API:en kysymysten määrästä, joka kuukaudelta.



Kaavio 2. Kyselyjen määrä Google Maps, Leaflet -ja Mapbox API:sta vuodesta 2009 vuoden 2019 loppupuolelle asti (Stack Overflow 2020).

## 4 API-RAJAPINTOJEN OMINAISUUDET

Tämä luku tarjoaa perustiedot jokaisesta ohjelmistokirjastosta. Informaatio on kerätty jokaisen ohjelmistokirjaston dokumentaatiosta. Tulen käymään läpi niiden avainominaisuuksia, käyttövaatimuksia ja mahdollisia rajoituksia.

### 4.1 Google Maps API

Googlen eri alustoille on saatavissa laaja dokumentaatio Googlen kehittäjä sivustolla [cloud.google.com](https://cloud.google.com). Sivustolta saa ohjeet, mitä vaaditaan ohjelmistokirjaston käytön aloittamiseen ja aloitusoppaat yksinkertaisiin toimintoihin sekä tarkat tekniset tiedot veteraaniohjelmoijille.

Googlen maantieteellistä kattavuutta pidetään parhaimpana heidän satelliittiensä, katukuvausautojen ja käyttäjien tuottamien korjauksien ansiosta. Google Mapsissa käytetään JSON-tyyppistä syntaksia, joka latautuu heti kartan avautuessa. Kartan näkyvyyttä, väriä ja karttaelementtien läpinäkyvyyttä voi muokata vapaasti. Googlen tunnettu katukuvaominaisuus tarjoaa interaktiivisen panoraaman eri pisteistä maailmalla. Katukuvauksesta on myös tarjolla liiketilojen omistajille palvelu, jossa voidaan integroida katukuvaus heidän tilojensa sisälle, jotta he voivat tarjota asiakkaille virtuaalikäynnin (esim. Kuva 1). Google Maps pystyy myös näyttämään maantieteellisiä ominaisuuksia KML-syntaksin avulla. Näihin kuuluvat kiinnostavat kohteet, kuvat, monikulmat ja vielä esimerkiksi rikosstatistiikasta viimeisimpään informaatioon maanjäristyksistä. Googlen kattava tuki eri kielille tarjoaa palvelun 1,25 biljoonalle käyttäjälle heidän omalla äidinkielellään. Tarvittaessa voi tukeutua Googlen isoon yhteisöön ja laajaan kehittäjäjoukkoon antamaan lisätukea. (Bulatovych 2018.)



Kuva 1. Kuvakaappaus Googlen katukuvauksen integraatiosta liikkeessä (Google 2020a).

Google Maps API vaatii käyttäjältä käyttäjätilin ja luottokorttitiedot siltä varalta, että ilmaisversion raja ylittyy. Hinnastosta ei ole ensikertalaisena helppoa saada selvyyttä. Google antaa käyttäjälle joka kuukausi \$200 edestä luottoa, tämän summan ylittämisen jälkeen Google aloittaa tilin laskuttamisen.

Google takaa siis käyttäjille joka kuukausi 14 000 upotetun kartan latausta, 100 000 staattisen kartan latausta, 28 000 dynaamisen kartan latausta, 28 000 staattisen katukuvan latausta tai 14 000 dynaamisen katukuvan latausta. Täytyy ottaa myös huomioon, että ilmaisversiossa samanaikaisten kyselyjen määrä on vain 10. (Google 2020b.)

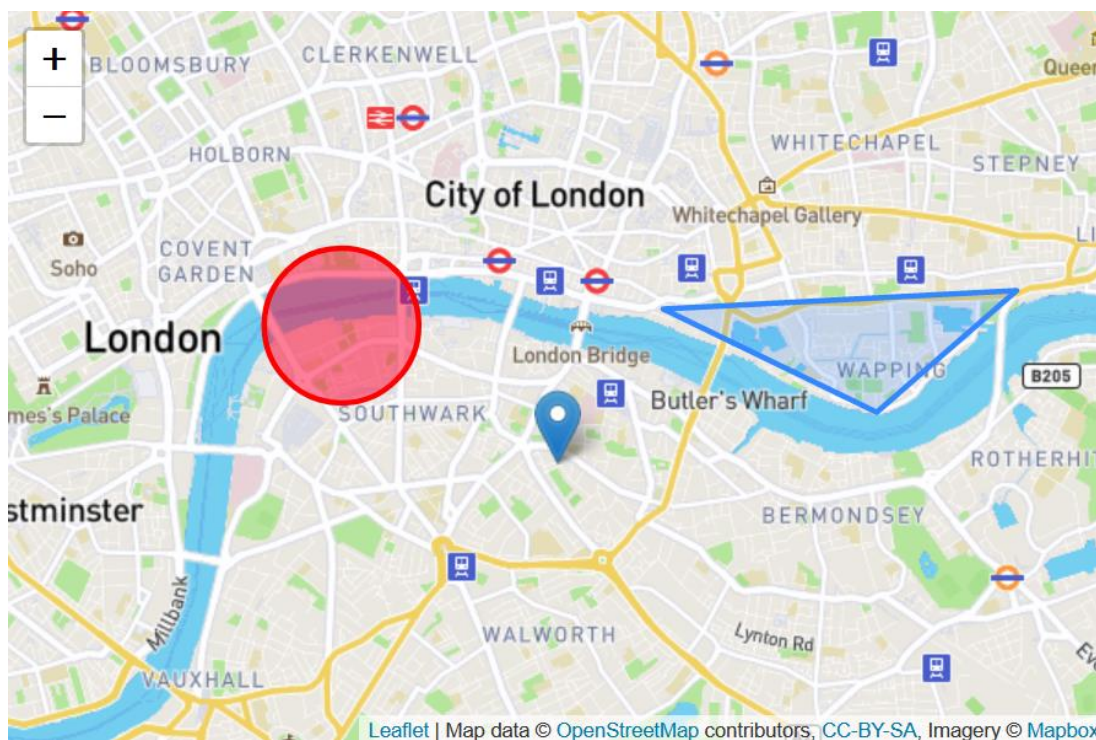
#### 4.2 Leaflet

Dokumentaatio Leafletin sivustolla [leafletjs.com](https://leafletjs.com) kattaa tiedot kaikista tarvittavista ominaisuuksista. Sieltä saa myös hyvät käyttöoppaat käyttöönotosta ja listan erilaisista plugineista.

Leaflet tarjoaa kehittäjille vertaansa vailla olevan määrän kontrollia ja houkuttelee alan tekijöitä jatkuvasti kehittyvällä kehittäjävaikutteisella API:lla sekä kasvavalla plugin kirjastolla. Leafletin kartan implementaatio on yksinkertaista kehitysnäkökulmasta, kun ei tarvitse arvuutella lisenssejä tai odottamattomia datamuutoksia. Leafletin kehitysnopeudelle pärjää vain sen kanssa yhteistyössä olevat avoimen lähdekoodin projektit ja kolmannen osapuolen projektit, joita rakennetaan Leafletin ympärille. (Creative Bloq Staff 2014.)

Leafletin kartoissa ei ole omaa datalähdettä, mutta oletukseltaan sen kanssa käytetään OpenStreetMappia. OpenStreetMapin tarjoamaa dataa voi tarvittaessa muokata omaaloitteisesti. Datalähteeksi voi valita muitakin, Leafletissä esimerkiksi Mapboxin. Muita datalähteitä käyttäessä kannattaa pitää mielessä käyttöehdot ja niiden rajoitukset ja toimia niiden rajojen mukaisesti. Kuvassa 2 on Leafletin kartta Lontoon keskustasta OpenStreetMapin datalla ja Mapboxin karttakuvalla.

Koska Leaflet on avoimen lähdekoodin ohjelmisto sen kanssa ei tarvitse luottokorttitietoja, toisin kuin Googlen kanssa. Täytyy silti pitää mielessä, että Leaflet tarjoaa vain esitystavan eikä dataa.

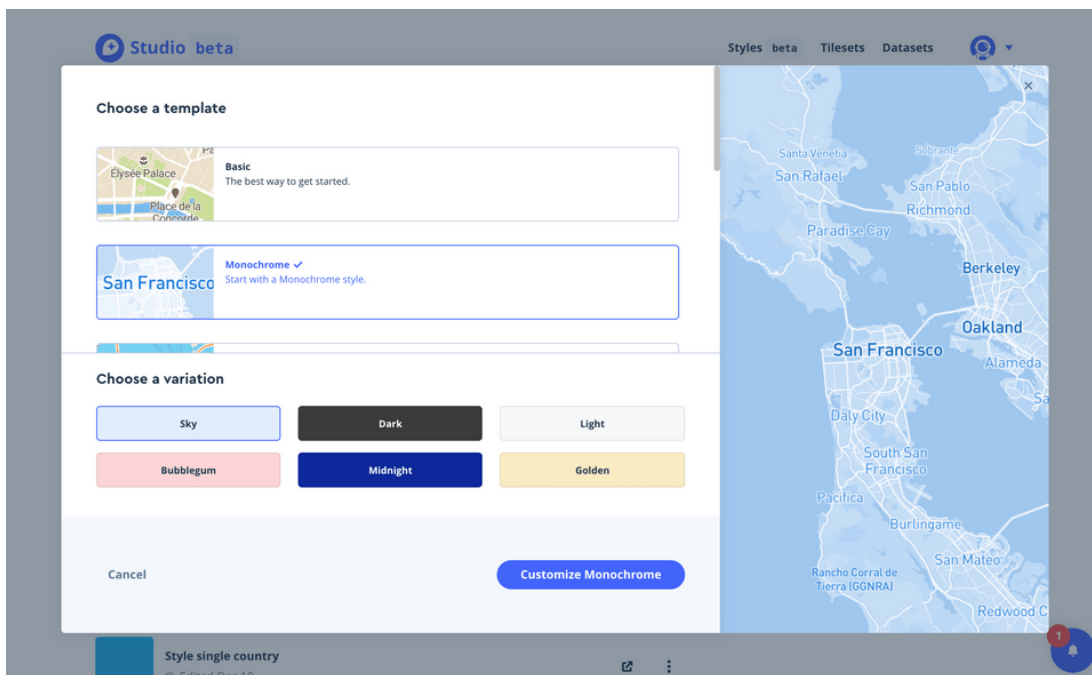


Kuva 2. Esimerkkejä Leafletin karttamerkeistä (Leaflet 2020).

### 4.3 Mapbox GL JS

Mapboxin dokumentaatio heidän sivustollansa docs.mapbox.com on jäsennelty hyvin ja siellä on saatavilla lukuisia esimerkkejä eri toiminnoista sekä tyylivinkkejäkin mihin tahansa mitä oletkin työstämässä.

Mapbox ei tarjoa valmista karttajärjestelmää vaan se on kuin laatikko legoja, joista kehittäjä voi luoda kartan haluamallaan palikoilla. Voit esimerkiksi luoda kartan yrityksen brändiin sopivalla tyyllillä asettamalla fontteja, väriskeemoja tai sopivia toiminnallisuuksia. Mapboxin karttojen SDK:t ovat avointa lähdekoodia ja se on saatavissa GitHubissa, jossa Mapboxin koodi on näkyvässä, analysoitavissa ja paranneltavissa. Mapboxilla on mahdollista upottaa interaktiivisia ja muokattavia vektorikarttoja natiivisiin sovelluksiin monilla eri alustoilla. Yhteistyö PubNubin kanssa on tarjonnut Mapboxille ajankohtaisen datan lähettämisen dynaamisiin karttoihin sen visualisointiin reaaliajassa. Tähän sisältyy myös toiminnallisuudet omaisuuden seurannasta, geolokaation ja lämpökuvakartat. Vektorikartoilla voi käyttää Mapboxin karttoja ilman verkkoyhteyttä ja saman voi myös toteuttaa luomalla sovelluksen mobiili SDK:lla. Mobiiliversio lataa myös resursseja välimuistiin automaattisesti käytön aikana. Kuvassa 3. on esimerkki Mapbox Studio editorista, johon pääsee Mapbox-käyttäjätilin kautta (Bulatovych 2018.)



Kuva 3. Kuva aloitusikkunasta Mapbox Studio editorissa (Mapbox 2020a).

Myös Mapboxin kanssa tarvitaan käyttäjätili, kuten Google Maps API:lla. Mapboxin ilmaisversiolla on taattu, joka kuukausi: 50 000 upotetun kartan latausta, 50 000 staattisen karttakuvan latausta, tarkkoja koordinaattilatauksia 100 000, 200 000 vektori- ja staattisentiilisetin latausta tai 100 000 navigaatio hakua. (Mapbox 2020b.)



## 5 TEHOKKUUS

Tehokkuudesta kerättiin dataa tekemällä pieni testisovellus, jossa käytin jokaista kartasto API:a. Testisovelluksessa käytin samaa liiketoimintalogiikkaa kuin asiakkaalle tehtävässä sovelluksessa. Testauksessa käytin React Developers Toolsia.

React Developer Tools on selainlaajennus, jonka on luonut Facebook-tiimi. Tällä työkalulla voi tutkia React-komponenttien hierarkiaa selaimen sisällä. Laajennus lisää uuden osan selaimen, jolla pääsee näkemään sovelluksen komponenttien vaikutukset muissa komponenteissa. Tämän avulla pystyy suunnittelemaan kunnollisen komponentti rakenteen. Tällä hetkellä saatavissa Chrome ja Firefox selaimilla. (Developerdrive staff 2018.)

Testisovelluksessa on kolme eri karttanäkymää jokaiselle API:lle ilman mitään ylimääristä, joka voisi vaikuttaa mitattaviin tuloksiin. Jokainen kartta käyttää vain vakio tyyliasetuksia ja tiedostojansa. Testasin karttoja luomalla niille kohteita eri koordinaateihin ja kohdetta avatessa saa näkyymiin lisätietoja kohteesta kartalla.

Kuvassa 4 on GenerateMarkers funktion koodi. GenerateMarkers on parametrinon funktio, jonka alussa alustetaan lista koordinaateista. Koordinaatit koostuvat leveys- ja pituusaste numeroparista. Funktio lähtee täyttämään koordinaatteja listaan While-silmukkaan kovakoodatun määrän mukaisesti. Kuvan tapauksessa silmukka pyörii 99 kertaa, joten funktio palauttaa 100 koordinaattia, koska alustuksessa on annettu ensimmäinen koordinaatti. Tulen testaamaan kumminkin myös 100 koordinaatin lisäksi 500 ja 1000 koordinaatilla. Koordinaattien listan täytössä leveys- ja pituusaste numerot luodaan satunnaisesti -90 ja 90 väliltä.

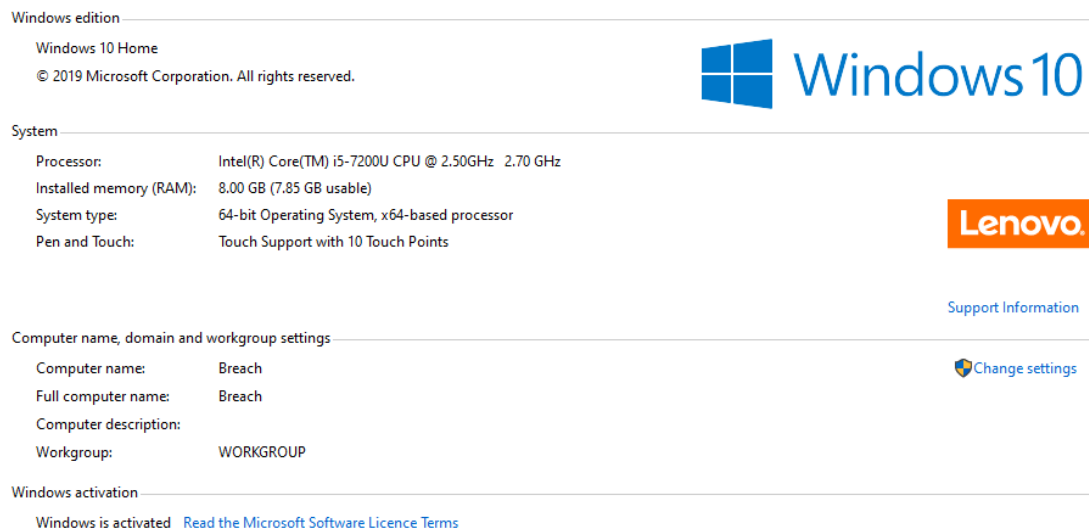
```
function GenerateMarkers() {  
  const locations: [[number, number]] = [[61.497939, 23.763335]];  
  
  function handleRandomLocations() {  
    const min = -90.0;  
    const max = 90.0;  
  
    const lat = min + Math.random() * (max - min);  
    const lng = min + Math.random() * (max - min);  
  
    return [lat, lng] as [number, number];  
  }  
  
  function fillLocations() {  
    var i = 0;  
  
    while (i < 99) {  
      locations.push(handleRandomLocations());  
      i++;  
    }  
    return locations;  
  }  
  
  return (  
    fillLocations()  
  );  
};  
  
export default GenerateMarkers;
```

Kuva 4. Testidatan koordinaattien luova funktio.

### 5.1 Testiympäristö ja testaus

Sovelluksen kartastojen nopeudet ja API-kutsujen määrä kerättiin ja testattiin Firefox selaimella. Testaus tehtiin lokaalisti Windows 10 käyttöjärjestelmällä Lenovo YOGA 510-15IKB kannettavalla tietokoneella. Tietokoneessa on 2.7GHz i5 prosessori ja 8 GB RAM:ia, joka on nähtävissä kuvassa 5.

View basic information about your computer



Windows edition

Windows 10 Home

© 2019 Microsoft Corporation. All rights reserved.

System

Processor:	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz
Installed memory (RAM):	8.00 GB (7.85 GB usable)
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	Touch Support with 10 Touch Points

Computer name, domain and workgroup settings

Computer name:	Breach
Full computer name:	Breach
Computer description:	
Workgroup:	WORKGROUP

Windows activation

Windows is activated [Read the Microsoft Software Licence Terms](#)

Kuva 5. Tekniset tiedot testauksessa käytettävästä kannettavasta tietokoneesta.

Sovellusta testattiin lokaalisti kannettavalla tietokoneella valokuituverkossa, jotta vältyttäisiin mahdollisilta virheiltiltä. Testejä oli kokonaisuudessaan kolme, joissa mitattiin nopeutta ja API-kutsujen määrää. Testejä oli kolme, jotta saatiin mitattua kartan kohteiden määrän vaikutusta API:en tehokkuuteen. Nopeudet mitattiin millisekuniteina.

## 5.2 Google Maps API

Kuvassa 6 on TypeScript tiedosto, joka määrittelee Google Maps API komponentin. Tiedosto alkaa välttämättömien komponenttien lisäyksellä Reactin ydinkomponenteista MyGoogleMaps tyylitiedostoon. Ilman tyylitiedostossa olevaa määritelyä korkeutta kartalle se tulostuisi vakiona nollan pikselin korkeudella, joka johtaisi siihen, että karttaa ei näkyisi sivustolla. Seuraavaksi defaultPropsissa on annettu kartan keskitys koordinaatti sekä zoom arvo. Zoom vaikuttaa siihen, kuinka lähelle keskitystä kartta tarkennetaan sen latautuessa. Seuraavaksi kehitetään kohteet kartalle tiedostoon lisätyn GenerateMarkers komponentin avulla. Tämän jälkeen Google Maps API komponentti viedään annetuilla tiedoilla. GoogleMapReactilla on tarvittavat keskitys ja zoom arvot. Sillä on myös tarvittava API-avain, joka vaaditaan Google Maps API:n kutsuihin. Kohteet käydään läpi Map-silmukalla, jolla viedään jokainen kohde MyMarker komponenttina, jolle annetaan kohteen koordinaatti. MyMarker komponentissa itsessään on vain määritely kohteelle ikoni kartalle ja ponnahtusikkuna toiminto kohdetta klikatessa.

```

import React, { Component } from "react";
import GoogleMapReact from "google-map-react";
import MyMarker from "./MyMarker";
import GenerateMarkers from "../GenerateMarkers";
import "./MyGoogleMap.scss";

class MyGoogleMap extends Component {
  defaultProps = {
    center: {
      lat: 61.497939,
      lng: 23.763335,
    },
    zoom: 5,
  };

  markers = GenerateMarkers();

  render() {
    return (
      <div className="map-styles">
        <GoogleMapReact
          bootstrapURLKeys={{ key: "AIzaSyCTGCxn0EVmxUvE69T6XtC
          defaultCenter={this.defaultProps.center}
          defaultZoom={this.defaultProps.zoom}
        >
          {this.markers.map((location, index) => (
            <MyMarker key={index} {...{ lat: location[0], lng:
          ))}
        </GoogleMapReact>
      </div>
    );
  }
}

export default MyGoogleMap;

```

Kuva 6. Google Maps API komponentin määrittelmä.

### 5.2.1 Testitulokset

Taulukossa 3 on Google Maps API:n nopeudet millisekunneissa. Nopeudet on eroteltu API:n koodin latausnopeuteen ja karttatiilien latausnopeuteen. Tuloksia on kolmelta eri testikerralta 100:sta kohteesta, 500 kohteesta ja 1000 kohteesta kartalla. API-kutsuja oli joka kerralla 31 kappaletta, mutta Googlen kehittäjä sivustolla

cloud.google.com laskutettavia API-kutsu kertoja oli silti karttanäkymän testilataus-kertojen mukaisesti.

Google Maps API	100 kohdetta	500 kohdetta	1000 kohdetta	Keskiarvo
Koodin lataus	440	535	558	511
Karttakuvien lataus	223	590	840	551

Taulukko 2. Google Maps API kartan testitulokset millisekunneissa.

### 5.3 Leaflet

Leaflet komponentin määrittelevä TypeScript tiedosto kuvassa 7 alkaa normaalisti komponenttien lisäyksellä. Reactin ydinkomponenttien lisäksi on osat React-leaflet komponentista, jolla kokoamme kartan. Lisättyä on myös Leafletin oma tyylitiedosto, jota ilman kartta ei voisi toimia. LeafletMap komponentilla on parametri `locations`, joka on lista kohteista, mikä annetaan sille LeafletView näkymän kautta. Seuraavaksi vakio `position` alustetaan kartan keskityskoordinaatilla. Funktiolla `handleMarker`, jonka parametrit ovat koordinaatit ja indeksinumero tällä funktiolla voi komponentti palauttaa kohteen ja sen Pop-upin kartalle. Lopulta LeafletMap viedään alustetuilla tiedoilla. Map sisältää keskitys- ja tarkennusarvot. Kartta kootaan `TileLayer`illa jolle annetaan `attribution`in arvo, jossa mainitaan kartassa käytetty datalähde. Datalähde annetaan `url-attribuutille`, tässä tapauksessa datalähde on `OpenStreetMap`ilta. Mapin sisässä viedään myös aikaisemmin mainitut kohteet `handleMarker` funktion ja `Map`-silmukan avulla.

```

import React, { FC } from "react";
import { Map, TileLayer, CircleMarker, Popup } from "react-leaflet";
import "leaflet/dist/leaflet.css";
import "../LeafletMap.scss";

interface IProps {
  locations: [[number, number]];
}

const LeafletMap: FC<IProps> = ({ locations }) => {
  const position: [number, number] = [61.497939, 23.763335];

  function handleMarker(location: [number, number], index: number) {
    return (
      <CircleMarker key={index} center={location} color="red" radius={5}>
        <Popup>Kohde: {index + 1}</Popup>
      </CircleMarker>
    );
  }

  return (
    <div className="leaflet-map">
      <Map center={position} zoom={5}>
        <TileLayer
          attribution='&copy; <a href="http://osm.org/copyright">OpenStreetMap</a>'
          url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
        />
        {locations.map((location, index) => handleMarker(location, index))}
      </Map>
    </div>
  );
};

export default LeafletMap;

```

Kuva7. Leaflet kartta komponentin määrittelmä

### 5.3.1 Testitulokset

Toisin kuin aikaisemmin tehdyssä testissä Leafletillä ei ole ladattavaa koodia. Leafletilla on ainoastaan ladattavana karttatiilet ja niiden lähde riippuu käyttäjän omasta valinnasta. Tässä tapauksessa on käytössä OpenStreetMapin kartta.

Leaflet	100 kohdetta.	500 kohdetta.	1000 kohdetta.	Keskiarvo
Karttakuvien lataus	221	279	332	277

Taulukko 3. Leaflet komponentin testitulokset millisekunneissa.

## 5.4 Mapbox GL JS

MapboxMap komponentin määrittely kuvassa 8. alkaa kuin aikaisempienkin komponenttien. Ainoana erona, että lisätään Mapboxin GL komponentti ja heidän tyylitiedostonsa samalla tavalla kuin Leafletin kanssa. Alussa alustetaan mapboxgl:ssä käytettävä API-avain, jonka saa Mapboxin sivustolta käyttäjätunnusten takaa. React componentDidMountissa alustetaan vakio map uudella mapboxgl kartalla, jolle annetaan joku tyyli lukuisista erilaisista saatavissa olevista Mapboxin sivustolta. Uudelle kartalle annetaan myös keskityskoordinaatti ja tarkennusarvo, kuten aikaisemmissakin kartoissa. Locations muuttujaan kehitetään GenerateMarkers funktiolla kohteet. Nämä kohteet käydään läpi silmukassa ja luodaan uusia merkkejä Pop-up kartalle kohteiden koordinaattien mukaisesti. Kartan liikkeessa setState funktiolla varmistetaan, että kohteet pysyvät oikeilla koordinaateillaan. Lopulta kartta vietään mapContainerin tyyleillä HTML-elementissä.

### 5.4.1 Testitulokset

Mapbox eroaa muista kartastoista sillä, että on vain yksi karttakuvan kutsu, joka kattaa koko kartan. Tämä näkyy isommassa määrässä koodin latauksia ja kartalla liikkeessä on karttakuvien latauksien sijasta Mapbox komponentin koodin latauksia. Myös Mapboxilla API kutsuja on useita kuin Google Maps API:lla, vaikka Mapboxin sivustolla käyttäjän API-statistiikkatiedoissa näkyy vain yksi kutsu per sivun lataus.

Mapbox GL JS	100 kohdetta	500 kohdetta	1000 kohdetta	Keskiarvo
Koodin lataus	236	312	358	302
Karttakuvien lataus	32	47	78	52

Taulukko 4. Mapbox GL JS komponentin määritelmä TypeScript tiedostossa.

```

import "../MapboxMap.scss";
import "mapbox-gl/dist/mapbox-gl.css";

mapboxgl.accessToken =
  "pk.eyJ1Ijoibml1bWl2aSIsImEiOiJjazhoMTd2NGQwMmVuM2VtOWliNXh5a2JlIn0.CL6pr

class MapboxMap extends React.Component {
  mapRef = React.createRef();

  componentDidMount() {
    const map = new mapboxgl.Map({
      container: this.context,
      style: "mapbox://styles/mapbox/streets-v11",
      center: [23.763335, 61.497939],
      zoom: 5,
    });

    var locations = GenerateMarkers();
    locations.map((location) =>
      new mapboxgl.Marker()
        .setPopup(
          new mapboxgl.Popup()
            .setHTML(`<div>${location}</div>`)
            .setMaxWidth("300px")
          )
        .setLngLat(location)
        .addTo(map)
    );

    map.on("move", () => {
      this.setState({
        lng: map.getCenter().lng.toFixed(4),
        lat: map.getCenter().lat.toFixed(4),
        zoom: map.getZoom().toFixed(2),
      });
    });
  }

  render() {
    return (
      <div>
        <div className="sidebarStyle">
          <div>
            Longitude: {23.763335} | Latitude: {61.497939} | Zoom: {5}
          </div>
        </div>
        <div ref={el => (this.context = el)} className="mapContainer" />
      </div>
    );
  }
}

export default MapboxMap;

```

Kuva 8. Mapbox karttakomponentin määrittelmä tiedosto.



## 6 VERTAILU

### 6.1 GitHub

GitHub tuo uniikin näkökulman komponentti markkinoille, koska se tarjoaa informaatiota, mistä komponenteista käyttäjät ovat kiinnostuneita eniten, millä komponentilla on eniten kehittäjiä ja millä komponentilla on eniten ”ongelmia”. Tämä data ei ole silti kaikista luotettavinta, koska vanha komponentti on voinut kerätä pidemmän aikaa seuraajia ja monimutkaisella komponentilla on isoin ongelmalista. Tämä on hyvä pitää mielessä katsoessa dataa GitHub taulukossa 1.

Leafletillä on selvästi enemmän tähtiä muihin verrattuna 20 000:n tähden erolla. Leaflet on myös ainoa, jolla kehitys on ollut puhtaasti GitHubissa pidemmän aikaa. Google ja Mapbox ovat suhteellisen tasoissa toistensa kanssa noin reilulla tuhannen tähden erolla. Leaflet jatkaa johtoa haarukoinnissa kolmen tuhannen erolla seuraavana olevaan Mapboxiin, mutta ei läheskään niin isolla erolla kuin tähtien kanssa. Mapbox vie voiton ongelmissa, joka voi joko tarkoittaa isoa suosiota tai isoa määrää oikeita ongelmia. Mapboxin ollessa uusin API näistä kolmesta voi selittää isoa ongelmalistaa. Silti on aika selvää Leafletin olevan suosituin karttasovellus GitHubissa, Mapbox seuraa uutena haastajana toisena ja Google on kolmantena.

### 6.2 NPM Trends

NPM Trendin data näyttää kuinka monta pakettia on ladattu jokaisesta komponentista kahden vuoden sisällä. Tämä data voi olla harhaanjohtavaa, koska jotkut komponentit ovat isompia ja tarvitsevat vähemmän kolmannen osapuolen tukea antavia paketteja. Kuviota 1. katsoessa voi huomata Leafletin kaksi vuotta sitten vahvasti alkaneen johdon latauksien määrässä, vaikka sillä on huomattavasti vähemmän ominaisuuksia kuin kilpailijoilla, eikä se tarjoa kartalle datalähdettä. Silti kahden vuoden aikana Mapbox on päässyt hyvin lähelle Leaflettiä, kun taas Google on pysynyt tasaisesti samalla tasolla.

### 6.3 Stack Overflow Trends

Stack Overflow on globaali ohjelmistokehittäjien yhteisön pilvialusta, jossa kehittäjät voivat jakaa koodia ja auttaa toisiaan. Katsoessa Stack Overflow dataa voi saada hyvän indikaation siitä mitä käytetään ja mistä puhutaan ohjelmistokehittäjien kesken maailmanlaajuisesti. Toki helpommat komponentit saavat vähemmän kysymyksiä kuin monimutkaiset, mikä voi johtaa harhaanjohtavaan kuvaan suosioista.

Kuviossa 2. näkyy prosentit kysymyksistä Leafletin, Googlen ja Mapboxin välillä 2009-2019. Voimme heti nähdä Googlen olleen kartastomarkkinoilla aikaisemmin kuin kilpailijansa. Korkeimmillaan Google oli vuonna 2012, jolloin myös Leaflet ja Mapbox ovat päätyneet Stack Overflowin kysymyksiin. Tästä on alkanut Leafletin ja Mapboxin nousu sillä aikaa, kun Googlen suosio lähti tippumaan. Leaflet on noussut jyrkemmin kuin Mapbox, mikä voi johtua sen kehittäjäpainotteisemmasta kehityksestä muihin verrattuna.

Google näyttää jo saavuttaneen potentiaalinsa Stack Overflowissa ja sillä aikaa Leaflet ja Mapbox jatkavat kasvua. Voi aika turvallisesti sanoa, että kehittäjät ovat siirtyneet Googlen leiristä Leaflettiin tai Mapboxiin. Leafletin ollessa hieman monimutkaisempi käyttää sanoisin sen ja Mapboxin olevan tasoissa toistensa kanssa.

### 6.4 Ominaisuudet

Käyttöön otossa kaikki kolme API:a olivat hyvin samankaltaisia, lukuun ottamatta Mapboxin ja Googlen käyttäjätilin luontia ja API-avaimen hankintaa. Mielenkiintoista kyllä koordinaattien käytössä sama tyyppimuoto kävi kaikille kolmelle. Leaflet vaikuttaa hieman työläämmältä kilpailijoihin verrattuna, mutta omien komponenttien käyttö kartassa on hyvin houkuttelevaa ohjelmistokehittäjän näkökulmasta Mapboxin ja Googlen kanssa tämä ei ole yhtä yksinkertaista. Leafletin kanssa on myös helpompi ottaa oma kartastodata käyttöön, tämä on myös Mapboxillakin mahdollista. Tosin halutessa tyylikkään omanlaisen kartan tämä on helppo toteuttaa Mapboxin lukuisista erilaisista tyyleistä. Mapboxin ja Googlen statistiikkatiedot voivat olla hyvinkin hyödyllisiä projektissa olevissa kohteisiin pääsyssä, jos halutaan käyttää samaa karttaa navigaatiossa.

Kaikkien kolmen dokumentaatio olivat kattavia. Googlen dokumentaatio oli varmasti laajin, mutta sieltä oli myös vaikea löytää tietoa ja hahmottaa sitä. Mapboxin dokumentaatiota oli helppo navigoida ja löysin heti tarvittavat esimerkit toimintoihin mitä halusin. Leafletin dokumentaatio oli hieman näiden kahden väliltä toki pienempi, koska siinä on vähemmän ominaisuuksia. Leafletillä on mahdollista lisätä paljon kolmannen osapuolen lisäosia, joilla voi laajentaa ominaisuuksien määrää.

Googlen ja Mapboxin ilmaisversioissa on hieman erilaisuuksia. Mapboxilla saa 36 tuhatta upotetun kartan latausta enemmän, mutta Googella saa ladata 50 tuhatta enemmän staattisia karttakuvia. Mapboxista ei löytynyt ilmaisversiossa rajoitusta samanaikaisesti API kyselyihin toisin kuin Googlen maksimissaan 10 samanaikaista latausta.

## 6.5 Tehokkuus

Kaikkia API-komponentteja ei ole rakennettu samanlaisesti ja ne voivat toimia eri lailla eri tyyppisissä ja kokoisissa sovelluksissa. Silti pieni sovellus, jonka tein tätä tutkimusta varten pitäisi antaa jotain vihjettä näiden API:en toiminnasta. Koodi-esimerkeistä näkee, että karttojen määrittely on hyvin samankaltaista ja rivien määrä on lähellä toisiansa. Toiminnassa oli hieman eroavaisuuksia, esimerkiksi Leafletia käyttäessä testisovelluksessa kartta ei lataa ollenkaan koodia, kun taas Mapbox ja Google lataavat paljonkin. Google ja Leaflet lataavat myös paljon karttakuvia, kun Mapbox lataa yhden karttakuvan. Kaikkien API lataukset toimivat kumminkin asynkronisesti

Jokaista karttakomponenttia kohden tehtiin kolme testiä, jotka koottiin taulukoihin 2., 3. ja 4. Keskiarvot näistä tuloksista on kerätty taulukkoon 6. Googlen API:n ollessa vanhin ja sen laajan ominaisuuslistan vuoksi voidaan sen olettaa olevan muita hitaampi. Silti Googella on paljon eroa Mapboxiin, joka on lisännyt ominaisuuksia jatkuvasti ja on silti pysynyt tehokkaana. Leafletia voisi pitää kevyimpänä API:na, mutta toimintojen lisääminen oman sovelluksen kautta voi tuoda kuormitusta asiakkaan laitteistolle. Niin kauan, kun ei haluta hirveästi erilaisia tai raskaita toimintoja niin Leaflet on kevyin API käyttää, mutta muussa tapauksessa kevyin on Mapbox GL JS.

Taulukko 6. Kootut keskiarvot testatuista nopeuksista millisekunneissa.

	Leaflet	Google Maps API	Mapbox GL JS
Karttakuvien lataus	277	551	52
Koodin lataus	*	511	302

\*Leafletilla ei ollut koodin latausta.

## 7 YHTEENVETO

Tämän opinnäytetyön tarkoitus oli vertailla kolmea suosituinta kartasto API:a suosiossa, ominaisuuksissa ja tehokkuudessa, jotta lukija voisi tehdä päätöksen näiden väliltä. Tämän tutkimuksen aikana informaatiota API:en suosioista kerättiin suurilta palveluntarjoajilta Github, NPM ja Stack Overflow. GitHubissa suosion voitti Leaflet suurella tähtien määrällä muihin kilpailijoihinsa verrattuna, jotka olivat tasoissa toistensa kanssa. NPM latauksissa Leaflet on pitänyt vahvaa ykkössijaa, mutta Mapbox on kirinyt sen kiinni ja jos kuvaajaan on uskomista, Mapbox luultavasti menee Leafletin ohitse. Stack Overflowissa Google oli saavuttanut huippunsa jo monta vuotta sitten, jonka jälkeen Mapbox ja Leaflet ovat tehneet tasaista nousua. Leaflet on hieman korkeammalla kuvaajassa kuin Mapbox, joka voi johtua sen monimutkaisuudesta.

Ominaisuuksissa tuli selville, että Leaflet on selvästi luotu ohjelmistokehittäjille, joita ei haittaa tehdä omat komponenttinsa karttaa varten. Mapboxilla taas voi tehdä visuaalisesti monipuolisempia karttoja ja sillä on laajasti ominaisuuksia tarjolla. Googlelakin on paljon ominaisuuksia, mutta käyttöönotto on hieman hankalampaa kilpailijoihin verrattuna ja ilmaisversion rajoitukset voivat tulla vastaan nopeasti.

Tehokkuutta testattiin tekemällä sovellus, jossa käytetään jokaista API:a. Tätä mitattiin latausnopeuksissa. Leaflet oli latauksissa nopein, joka selittyy vähäisillä ominaisuuksilla. Tämän jälkeen ei niinkään kaukana oli Mapbox, jonka latausnopeuksiin kohteiden määrä ei niinkään vaikuttanut. Google oli selvästi raskain käyttää, joka voi selittyä sen iällä ja laajalla valikoimalla ominaisuuksia.

Tulevaisuutta ajatellen Mapbox on selvästi nousussa sekä suosiossa, tehokkuudessa ja helppokäyttöisyydessä. Leaflet on kevyin käyttää näistä kartoista. Toki se vaatii aikaa oppimiseen, mutta sillä saa tehtyä täysin räätälöityjä ratkaisuja. Googlen aikakausi alkaa olemaan pikkuhiljaa ohi elleivät saa tehtyä kevyempää ja tehokkaampaa ratkaisua.

## LÄHTEET

- Creative Bloq Staff. 2014. Is Leaflet a better tool than Google Maps? Viitattu 19.02.2020. <https://www.creativebloq.com/web-design/leaflet-google-maps-121413738>
- Developerdrive Staff. 2018. 20 Best Tools for React Development. Viitattu 20.02.2020. <https://www.developerdrive.com/20-best-tools-for-react-development/>
- GitHub 2020a. Leaflet. Viitattu 27.01.2020. <https://github.com/leaflet/leaflet>
- GitHub 2020b. React Google Maps. Viitattu 27.01.2020. <https://github.com/tom-chentw/react-google-maps>
- GitHub 2020c. Mapbox GL JS. Viitattu 27.01.2020. <https://github.com/mapbox/mapbox-gl-js>
- GitHub 2020d. Forking Projects. Viitattu 28.01.2020. <https://guides.github.com/activities/forking/>
- GitHub 2020e. Mastering Issues. Viitattu 30.01.2020. <https://guides.github.com/features/issues/>
- Google 2020a. Google Maps Platform FAQ. Viitattu 22.01.2020. [https://developers.google.com/maps/faq?csw=1#tos\\_commercial](https://developers.google.com/maps/faq?csw=1#tos_commercial)
- Google 2020b. Street View. Viitattu 13.02.2020. <https://developers.google.com/streetview>
- Google 2020c. Pricing for maps, routes, and places. Viitattu 13.02.2020. <https://cloud.google.com/maps-platform/pricing/sheet/>
- Gousios, G., Vasilescu, B., Serebrenik, A. & Zaidman, A. 2008. Lean GHTorrent: GitHub Data on Demand. The Netherlands: Delft University of Technology & Eindhoven University of Technology, 834.
- Hoffman C. 2018. What is an API. Viitattu 21.01.2020. <https://www.howto-geek.com/343877/what-is-an-api/>
- Leaflet. 2020. Leaflet Quick Start Guide. Viitattu 19.02.2020. <https://leafletjs.com/examples/quick-start/>
- Lovelace, R. 2017. Testing web map APIs – Google vs OpenLayers vs Leaflet. Viitattu 21.01.2020. <https://web.archive.org/web/20171103225922/http://robinlovelace.net/software/2014/03/05/webmap-test.html>
- Mapbox 2020a. Create a custom style. Viitattu 10.02.2020. <https://docs.mapbox.com/help/tutorials/create-a-custom-style/>

Mapbox 2020b. Mapbox pricing. Viitattu 19.02.2020. <https://www.mapbox.com/pricing/>

MapQuest 1997. Welcome to MapQuest! Viitattu 10.04.2020. <https://web.archive.org/web/19971211230909/http://www.mapquest.com:80/>

NPM 2020a. Mapbox-gl, Viitattu 24.01.2020. <https://www.npmjs.com/package/mapbox-gl>

NPM 2020b. About npm. Viitattu 30.01.2020. <https://docs.npmjs.com/about-npm/>

Npm Trends. 2020. Leaflet vs Mapbox-gl vs React-google-maps. Viitattu 30.01.2020. <https://www.npmtrends.com/leaflet-vs-mapbox-gl-vs-react-google-maps>

Roos, D. 2007. How to Leverage an API for Conferencing. Viitattu 21.01.2020. <https://money.howstuffworks.com/business-communications/how-to-leverage-an-api-for-conferencing1.htm>

Stack Overflow 2019. About. Viitattu 31.01.2020. <https://stackoverflow.com/company>

Stack Overflow 2020. Stack Overflow Trends. Viitattu 31.01.2020. <https://insights.stackoverflow.com/trends?tags=leaflet%2Cmapbox%2Cgoogle-maps-api-3>