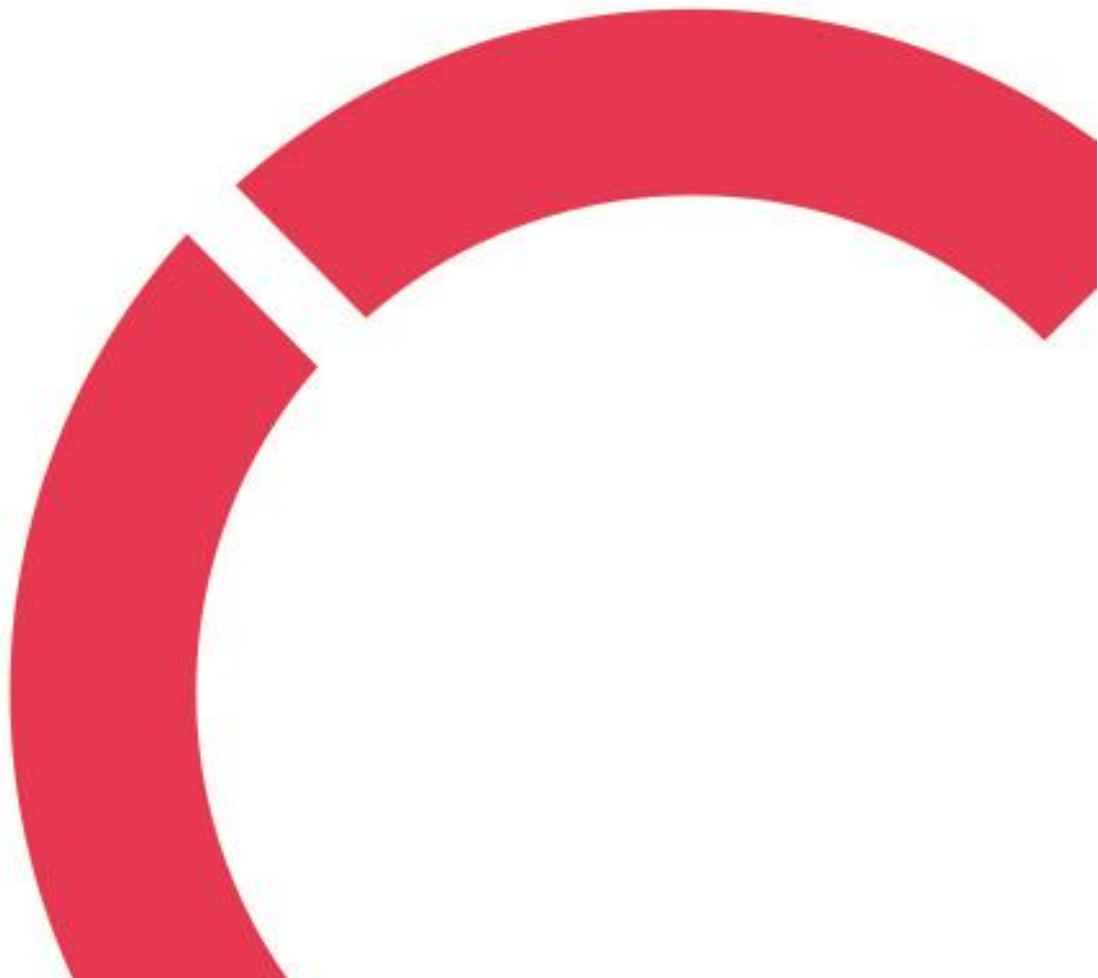


Salla Leppänen

**GOOGLLEN TARJOAMIEN RAJAPINTOJEN HYÖDYNTÄMINEN
LIIKETOIMINNASSA**

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tieto- ja viestintäteknikka
Huhtikuu 2022**



TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Centria-ammattikorkeakoulu	Aika Huhtikuu 2022	Tekijä/tekijät Salla Leppänen
Koulutus Tieto- ja viestintäteknikka		<input checked="" type="checkbox"/> AMK <input type="checkbox"/> YAMK
Työn nimi GOOGLLEN TARJOAMIEN RAJAPINTOJEN HYÖDYNTÄMINEN LIIKETOIMINNASSA		
Työn ohjaaja Jari Isohanni		Sivumäärä 32
Työelämäohjaaja Jussi Yli-Korpela		
<p>Opinnäytetyön tavoitteena oli tutkia Googlen tarjoamien rajapintojen hyödyntämistä liiketoiminnassa. Aihe valikoitui yrityksen toiveesta tutkia ja testata Googlen tarjoaman rajapinnan luotettavuutta käytötarkoitukseensa.</p> <p>Teoriaosuudessa aihetta lähestyttiin yksinkertaisesti sillä, mikä rajapinta on, kuinka se toimii ja miksi se on hyödyllinen yritykselle. Lisäksi käytiin lyhyesti läpi tärkeimpiä kriteereitä rajapinnan valinnassa. Sen jälkeen tutkittiin mikropalveluiden eroja verrattuna monoliittiseen järjestelmään ja nostettiin esille rajapintojen käytetyimpiä tekniikoita. Lopuksi käytiin läpi Googlen tarjoamia rajapintoja, saatavilla olevia työkaluja ja dokumentaatiota.</p> <p>Käytännön osuus koostui kahdesta käyttötapauksesta. Ensimmäisessä käyttötapauksessa tehtiin yritykselle sovellus, joka hyödyntää Googlen tarjoamaa rajapintaa. Toisessa käyttötapauksessa tehtiin esimerkkisovellus Googlen tarjoamasta Cloud Text-to-Speech-rajapinnasta ja käytiin läpi rajapinta-kohtaista dokumentaatiota, josta nostettiin esiin kehityksen kannalta tärkeimpiä asioita.</p> <p>Lopputuloksena todettiin, että Google tarjoaa yritykselle vakaita rajapintoja, joita on mahdollista hioa paremmaksi lisäpalveluiden avulla. Käyttöönottoon on saatavilla laajasti dokumentaatiota ja tietoa. Rajapintoja on mahdollista kehittää eri koodikielillä ja koodiesimerkkejä on tarpeeksi saatavilla. Google myös tarjoaa käyttäjilleen hyödyllisen ilmaisen kokeilujakson käyttöönoton testaukseen. Lisäksi rajapinnan hallintapaneelista voidaan seurata liikennemääriä, latenssia ja mahdollisia virheitä. Saatavilla on myös hyödyllinen mobiilisovellus, jonka avulla on mahdollista tehdä perustoimenpiteitä ja pysyä ajan tasalla ongelmatilanteissa. Myös tukipalveluita ja yhteisön tukea on tarvittaessa saatavilla riittävästi.</p>		
Asiasanat API, HTTP, mikropalvelut, rajapinta, REST		

ABSTRACT

Centria University of Applied Sciences	Date April 2022	Author Salla Leppänen
Degree programme Information and Communication Technologies		
Name of thesis UTILIZING GOOGLE INTERFACES FOR BUSINESS		
Centria supervisor Jari Isohanni	Pages 32	
Instructor representing commissioning institution or company Jussi Yli-Korpela		
<p>The aim of the thesis was to study the use of interfaces provided by Google in business. The topic was chosen based on the company's desire to study and test the interface provided by Google and to evaluate its reliability for its intended use.</p> <p>In the theoretical part, the topic was approached simply by what the interface is, how it works, and why it benefits the company. In addition, the main selection criteria for the user interface were briefly reviewed. This was followed by an examination of the differences between microservices and the monolithic system, as well as the most commonly used interface technologies. Finally, we went through the interfaces, tools, and documentation provided by Google.</p> <p>The practical part consisted of two use cases. In the first use case, an application was made for a company that takes advantage of the interface provided by Google. In the second case, the Cloud Text-to-Speech interface provided by Google was used as an example application, and the documentation and key issues for development were reviewed. In the second case,</p> <p>The results of this study show that Google provides the company with stable interfaces that can be improved with additional services. Comprehensive documentation and information is available for implementing the interface. Interfaces can be developed in different code languages and enough code examples are available. Google also offers for users a useful free trial to test the interface. In addition, the interface control panel can be used to monitor traffic volumes, latency and possible errors. A useful mobile app is also available from the control panel, allowing you to manage the most common functions and stay up to date on errors. Support services and community support are also available as needed.</p>		
Key words API, Application Programming Interface, HTTP, microservices, REST		

KÄSITTEIDEN MÄÄRITTELY

API

Application Programming Interface on sovellusohjelmointirajapinta, joka mahdollistaa moduulien kommunikoinnin keskenään hyödyntäen toistensa toimintoja ja tietoja.

gRPC

Google Remote Procedure Call on Googlen kehittämä avoimen lähdekoodin siirtotekniikka, joka käyttää HTTP 2.0:aa siirtoprotokollanaan.

HTTP

Hypertext Transfer Protocol on tunnetuin ja käytetyin verkon siirtoprotokolla, jota selaimet ja verkkopalvelimet käyttävät tiedonsiirtoon.

HTTPS

Hypertext Transfer Protocol Secure on HTTP-siirtoprotokollan ja TLS/SSL-protokollan yhdistelmä, jota käytetään suojattuun tiedonsiirtoon verkossa.

IoT

Internet of Things tarkoittaa esineiden liittämistä internettiin.

JSON

JavaScript Object Notation on kevyt tekstin syntaksi ja siirtomuoto.

OAuth

Open Authorization on käyttöoikeusprotokolla.

REST

Representational Stateless Transfer on arkkitehtuurimalli, joka voi kutsua verkossa olevia resursseja HTTP:n yli tilattomasti käyttäen HTTP-metodeja. REST noudattaa tiettyä sääntöjoukkoa ja silloin se on RESTful.

RPC

Remote Procedure Call on verkon siirtoprotokolla

URI

Uniform Resource Identifier on syntaksi, jolla viitataan abstraktiin tai fyysiseen resurssiin.

URL

Uniform Resource Locator on merkkijono, jolla kerrotaan resurssin sijainti.

XML

Extensible Markup Language on tekstin syntaksi tiedonvälityksessä ja tiedon tallennusmuoto.

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

1 JOHDANTO	1
2 API-ARKKITEHTUURI	2
2.1 API-tyypit.....	3
2.2 Kuinka API toimii?	5
2.3 API:en hyödyt liiketoiminnassa	5
2.4 API:n valintakriteerit	6
3 MIKROPAAVELUT	8
4 REST API	11
4.1 REST API:n arkkitehtoniset säännöt.....	11
4.2 HTTP-protokolla.....	13
4.3 HTTPS-protokolla.....	16
5 GOOGLLEN TARJOAMAT API:T	17
6 GOOGLLEN TARJOAMIEN API:EN HYÖDYNTÄMINEN LIIKETOIMINNASSA	20
7 TULOKSET JA POHDINTAA	28
LÄHTEET	29
LIITTEET	
KUVIOT	
KUVIO 1. API:n toiminta.....	5
KUVIO 2. Esimerkki monoliittisestä sovelluksesta	8
KUVIO 3. Esimerkki verkkokaupasta toteutettuna mikropalveluina	9
KUVIO 4. Esimerkki HTTP-pyynnön rakenteesta	13
KUVIO 5. Esimerkki HTTP-vastauksen rakenteesta	14
KUVAT	
KUVA 1. Google Cloud-projektin luominen.....	21
KUVA 2. Google API:n käyttöönotto.....	22
KUVA 3. Service Accountin luominen osa 1	23
KUVA 4. Service Accountin luominen osa 2	23
KUVA 5. Python-virtuaaliympäristön luominen	24
KUVA 6. API:n asiakaskirjaston asentaminen kehitysympäristöön	25
KUVA 7. Esimerkkisovellus Googlen Cloud Text-to-Speech rajapinnasta	26
KUVA 8. Rajapinnan testin tulokset.....	27
TAULUKOT	
TAULUKKO 1. Yleisimmät HTTP-pyynnön tilakoodit	15
TAULUKKO 2. HTTP-metodien tilakoodit	15

1 JOHDANTO

Digitalisaation myötä integraatioiden tarve liiketoiminnassa on lisääntynyt voimakkaasti viime vuosina. Application Programming Interface (API) eli sovellusohjelmointirajapintojen avulla järjestelmät saadaan toimimaan saumattomasti keskenään tuottoisina kokonaisuuksina hyödyntäen yrityksen keräämää dataa ja järjestelmiä. Tietoja voidaan kuljettaa automaattisesti eri järjestelmien ja sovelluksien välillä. Näin saadaan tehostettua yrityksen liiketoimintaa ja otettua myös kaikki hyöty irti jo olemassa olevasta tiedosta. API on usein jonkun yrityksen myytävä palvelu, jonka käyttöönotto on huomattavasti nopeampaa kuin koko ohjelman keksiminen alusta asti itse. Tämän opinnäytetyön tarkoituksena onkin kartoittaa erityisesti Googlen tarjoamia rajapintaratkaisuja ja työkaluja yritykselle.

Opinnäytetyön aihe valikoitui yrityksen toiveesta toteuttaa loppusovellus, joka hyödyntää Googlen tarjoamaa rajapintaa, ja testata, voisiko se olla tarpeeksi luotettava käyttötarkoitukseensa. Aihetta lähdetään lähestymään ensin yksinkertaisesti siitä, mikä API on ja miten se toimii. Tuodaan esille, mihin kaikkeen rajapintoja voidaan hyödyntää ja tutkitaan, mitä hyötyjä yritykselle rajapintojen käytöstä on. Käydään myös läpi tärkeimpiä kriteereitä, jotka kannattaa ottaa huomioon rajapinnan valintaprosessissa. Tutustutaan tarkemmin käytetyimpiin tekniikoihin, kuten mikropalveluihin ja REST API:in ja sen käyttämään HTTP-protokollaan. Lisäksi pohditaan toteutusvaihtoehtojen eroja, kuten perinteistä monoliittista järjestelmää verrattuna mikropalveluihin.

Opinnäytetyön käytännön osuudessa tutustutaan Googlen tarjoamaan API-valikoimaan, dokumentteihin ja kehitystyökaluihin. Arvioidaan Google Cloud Platform -alustaa, joka toimii hallintatyökaluna rajapinnoille ja palveluille. Pohditaan alustan hyödyllisyyttä rajapintojen raportoinnissa ja ylläpidossa. Lisäksi tehdään kaksi käyttötapausta Googlen tarjoamista rajapinnoista. Ensimmäisessä käyttötapauksessa kehitettiin yritykselle loppusovellus hyödyntäen Googlen tarjoamaa Enterprise-rajapintaa. Toisessa käyttötapauksessa analysoidaan Googlen Cloud Text-to-Speech-rajapinnan käyttöönottoa kehittäjän näkökulmasta ja tehdään siitä esimerkkisovellus. Lopputuloksissa pohditaan Google API:en käyttöönoton sujuvuutta, dokumentoinnin laatua, kehitystyökaluja sekä rajapintojen toimivuutta käyttötarkoituksiinsa.

2 API-ARKKITEHTUURI

Application Programming Interface (API) eli sovellusohjelmointirajapinta yksinkertaistaa ohjelmistokehitystä mahdollistaen tiedon ja toimintojen vaihtamisen helposti ja turvallisesti kahden tai useamman sovelluksen välillä. API:en vuorovaikutus voi tapahtua verkossa tai ilman sitä. Rajapinta sijaitsee usein kahden sovelluksen välissä kuljettaen tietoa toistensa välillä. Se on ikään kuin välikerros, joka määrittää säännöt sille, kuinka järjestelmät kommunikoivat keskenään. (Suciu 2021; Macoveiciuc 2020; IBM Cloud Education, 2020.) API voi olla itsessään tuotteistettu palvelu, asiakkaalle tarjottava palvelu tai rajapinta tarjottavaan palveluun. Se voi olla osana fyysistä tuotetta kaupan päällisenä, kuten IoT-laitteissa, tai vaikkapa rajapinta fyysiseen laitteeseen tai alustaan kytkeytymiseen. (Moilanen, Niinioja, Seppänen & Honkanen 2018, 58–59.)

Rajapinnat ovat valloittaneet maailman hyvin lyhyessä ajassa ihmisten huomaamatta, ja monet suosituimmista verkkosovelluksista eivät olisi edes mahdollista toteuttaa ilman rajapintoja. API voi olla esimerkiksi hyvin kevyt ohjelma, joka lukee ja ohjaa fyysistä laitetta ja laitteen sensoreita käsitellen suurta määrää dataa. API voi yhdistää esimerkiksi kaksi yrityksen sisäistä ohjelmistoa sisäisessä verkossa (Suciu 2021). Rajapintoja löytyy nykyään lähes joka kodista, toimistosta, ajoneuvoista, hisseistä ja jopa puutarhan robottiruohonleikkurista. API:n avulla voidaan määrittää esimerkiksi laitteen tai ajoneuvon tila, huollon tarve ja sen hetkinen sijainti. (IBM Cloud Education 2020; Moilanen ym. 2018, 53–54.)

API-kutsussa osapuolten on ymmärrettävä toisiaan ja käytettävä samoja sääntöjoukkoja ja tietotyyppejä toimiakseen. Kaikki API:t eivät ole teknisesti samanlaisia, vaan on olemassa erilaisia API-tyyppejä, arkkitehtuurimalleja, protokollia ja käyttöoikeustasoja, joilla ne toimivat. Nämä on hyvä ottaa huomioon API:a valitessa, koska on tärkeää valita oikeantyyppiset rajapinnat tarkoitukseensa. Esimerkiksi julkisesti tietoja jakava API ei sovi yrityksen sisäiseen käyttöön, vaan yrityksen on suositeltavaa valita sisäinen rajapinta rajoitetuilla käyttöoikeuksilla. Yleisimpiä API-arkkitehtuureja on kolmenlaisia: Representational State Transfer (REST), Simple Object Access Protocol (SOAP) ja Remote Procedural Call (RPC). (Juliver 2021a; Castellani 2020.) Näistä suosituinta REST API:a käsitellään myöhemmin tarkemmin.

API-taloudessa hyödynnetään tehokkaasti ja nopeasti muiden yritysten resursseja lisäarvon tuottamiseksi omille asiakkaille, ja se mahdollistaa yritykselle potentiaalisia liiketoimintamahdollisuuksia. API:t ovat uusi normi, ja niitä hyödyntämällä yrityksellä on mahdollisuus vaikuttaa asemaansa, vauhdittaa menoa uusille markkina-alueille ja mahdollistaa liiketoiminnan kasvu uudella tavalla. API-talouden avulla voidaan vastata nopeammin kehittyviin asiakastarpeisiin, kehittää yhteistyötä muiden toimijoiden kanssa ja parantaa asiakastyytyväisyyttä. Tämä mahdollistaa myös yritykselle enemmän taloudellista etua, säästöjä ja jatkokehitysmahdollisuuksia, johtuen API:en tuomasta teknologia- ja toimittajariippumattomuudesta. (Moilanen ym. 2018, 207; Moilanen 2019; Haglund 2018.)

Yritykselle rajapinta voi olla myytävä palvelu tai tyypillinen kumppanirajapinta, joka yhdistää yritysten tai alustojen välisiä palveluita vaihtamalla dataa ohjelmien tai sovellusten kesken. Rajapinnat voivat muun muassa mahdollistaa yrityksen pääsyn moninkertaiseen määrään asiakkaita verrattuna omaan alustaan ja tuottaa sitä kautta yritykselle lisää myyntiä. Esimerkkinä kumppanirajapinnasta on Keskon ja Alibaban välinen rajapinta, joka mahdollistaa suomalaisten tuotteiden näkyvyyden ja ostettavuuden 700 miljoonalle ihmiselle. Rajapinta voi olla myös esimerkiksi suosittelutoimintoja tarjoava rajapinta, joka mahdollistaa verkkokaupan käyttäjälle personoidumman asiakaskokemuksen. (Moilanen 2019; Haglund 2018.)

2.1 API-tyypit

API:t luokitellaan kahdesta neljään tyyppiin ja määritelmiä on olemassa useita. Kin Lanen mukaan API:t ovat joko sisäisiä (internal) tai yksityisiä (private), kun taas esimerkiksi RestCase-blogin kirjoittaja luokittelee API:t kolmeen tyyppiin; sisäinen (internal), kumppani (partner) ja julkinen (public). (Lane 2015; Levin 2017.) API-talous 101-kirjan kirjoittajat päätyivät luokittelemaan API:t Kin Lanen mukaisesti kahteen tyyppiin (Lane 2015). Axwayn Stephane Castellani taas luokittelee artikkelissaan API:t neljään tyyppiin; avoin eli julkinen, kumppani, sisäinen ja yhdistelmärajapinta (composite). (Castellani 2020; Moilanen ym. 2018, 73–74.) Käydään lyhyesti läpi kaikki neljä API-tyyppiä.

Avoin API eli julkinen API on kehittäjien tai muiden käyttäjien saatavilla tietyin rajoituksin. Rajapinnan tietoihin ja palveluihin voi päästä käsiksi joko täysin avoimesti tai se voi vaatia toimiakseen API-avaimen, rekisteröinnin tai muun todennuksen. Jotkut julkiset API:t voivat olla ilmaisia tiettyyn rajaan asti, mutta useimmiten maksu määräytyy API-kutsujen määrän mukaisesti (Juliver 2021a). Julkisia ra-

rajapintoja on saatavilla maailmanlaajuisesti internetin kautta eri palveluntarjoajilta, ja johtavat API-keskittymät löytyvät Piilaaksosta (Google), Seattlesta (Amazon) ja Washington D.C.:stä sekä Lontoosta. Julkisten API:en määrä kasvaa koko ajan voimakkaasti. (Moilanen ym. 2018, 37.) Esimerkiksi Programmableweb.com-sivusto tarjosi vuoden 2018 alussa noin 20 000 julkista API:a, kun tällä hetkellä niitä on saatavilla jo yli 24 000. (Programmableweb 2022; Moilanen ym. 2018, 94–96; Castellani 2020.) Maailman suurin API-katalogi RapidAPI Hub tarjoaa tällä hetkellä yli 30 000 rajapintaa (RapidAPI 2022a).

Partner API eli tutummin kumppanirajapinta on usein yritysten välinen rajapinta, johon vaaditaan erityiset käyttöoikeudet ja joka ei ole julkisesti saatavilla (Castellani 2020). Kumppanirajapinta mahdollistaa yritykselle paremman tavan hallita resurssejaan ja sitä, kuka ja miten niitä voidaan käyttää (Juliver 2021a). Yleensä kumppanirajapinta yhdistää usean yrityksen tuottamia palveluita, mikä hyödyttää sekä asiakasta että yritystä. Asiakas saa paremman asiakaskokemuksen ja yritys lisää palvelujensa myyntiä. (Moilanen ym. 2018, 13.) Kumppanirajapintojen hyödyntäminen yksinkertaisuudessaan voi olla hyvin suoraviivaista, koska API:n tarjoaja on tehnyt valmiiksi opastuksen käyttöönottoon ja kertoo, mihin API:a voidaan hyödyntää ja millä ehdoilla sitä voidaan käyttää. Yrityksen on kuitenkin hyvä pohtia kumppanirajapintojen valinnassa, kenelle haluaa avata keskeisiä prosessejaan ja tietojään. (Moilanen ym. 2018, 84.)

Sisäinen API tarkoittaa yrityksen omien palveluiden kehittämiseen käytettävää rajapintaa, jota ei ole tarkoitettu kolmansien osapuolien käyttöön. Se on tarkoitettu yrityksen sisäisen tiedonsiirron tehostamiseen eri järjestelmien välillä. Näitä voivat olla esimerkiksi taloushallinnon ja henkilöstöhallinnon tukena käytettävät API:t tai valmiissa tuotteessa tarjolla oleva rajapinta, joka on tarkoitettu asiakkaan tai kumppanin käyttöön. Sisäinen API sisältää usein tietoja, joita ei kannata antaa kumppanien tai etenkin kilpailijoiden tietoon. (Juliver 2021a; Moilanen ym. 2018, 73–74.)

Yhdistelmärajapinnat yhdistävät usean API:n toiminnallisuuksia tai tietoja yhdeksi kokonaisuudeksi API-kutsuja ketjuttamalla tai yhdistämällä. Niitä käytetään esimerkiksi mikropalveluissa yhdistämään tietoja useasta palvelusta yhden tehtävän suorittamiseksi ja yhtenäisen vastauksen saamiseksi. Yhdistelmärajapintojen avulla API-kutsujen kokonaismäärä vähenee ja palvelimen kuormitus pienenee ja prosessi nopeutuu. (Juliver 2021a; Castellani 2020; Tarang 2021.)

2.2 Kuinka API toimii?

API on ohjelma, jossa on määritetty säännöt sille, kuinka laitteet ja järjestelmät kommunikoivat keskenään. Se on esimerkiksi välikerroksena asiakassovelluksen ja verkkopalvelimen tai tietokannan välissä. Tämä pyyntöjen ja vastausten prosessi alkaa asiakassovelluksen kutsulla (request) tietojen hakemiseksi. Pyyntö käsitellään Uniform Resource Identifier (URI) -tunnisteen kautta. Kun API on vastaanottanut kelvollisen pyynnön, se ottaa yhteyttä ulkoiseen verkkopalvelimeen tai ohjelmaan. Palvelin lähettää API:lle vastauksen (response), joka sisältää pyydytetyt tiedot. Lopuksi API siirtää tiedot prosessin aloittaneelle asiakassovellukselle (KUVIO 1). (IBM Cloud Education 2020.)



KUVIO 1. API:n toiminta (mukaiillen Fourtress 2020)

2.3 API:en hyödyt liiketoiminnassa

Rajapinnat parantavat tiedonvälitystä sovellusten ja järjestelmien välillä, ja näin ollen eri lähteistä haettua tietoa on mahdollista hallinnoida keskitetysti ja päällekkäinen työ vähenee (Hakola 2018). Rajapinnat parantavat organisaatioiden reagoitokykyä ja nopeuttavat prosesseja (Moilanen ym. 2018, 121). Niiden avulla voidaan luoda parempi ja personoidumpi asiakaskokemus, kun esimerkiksi asiakaspalvelujärjestelmä ja tilausjärjestelmä toimivat saumattomasti yhteen (Haglund 2018). API-tarjoajien valmiit rajapinnat mahdollistavat organisaatioille nopeamman kehityksen, kun kaikkea ei tarvitse keksiä ja kokeilla alusta asti itse. Kehitystiimit voivat oppia ulkopuolisilta tekijöiltä ja heille jää enemmän aikaa keskittyä sovelluksen käyttökokemukseen. (Moilanen ym. 2018, 118–119.)

Rajapintojen käyttö yrityksen sisäisessä tiedonsiirrossa mahdollistaa paremman skaalautuvuuden tulevaisuudessa, ja se on paljon tehokkaampaa, turvallisempaa ja helpommin seurattavissa (Juliver 2021a).

Rajapinnat mahdollistavat myös yritysten IT-ratkaisuissa esimerkiksi tallennuskapasiteetin tai muun palvelun hankkimisen tarpeensa mukaan pilvialustoista muutamalla API-kutsulla, kun taas ennen jouduttiin hankkimaan tähän vaadittavaa fyysistä levytilaa, ohjelmistolisenssejä tai palvelinlaitteistoa. API:n avulla ainoastaan käytetystä kapasiteetista tai palvelusta maksetaan ja palvelusta voidaan helpposti luopua heti tarpeen päättyessä. (Moilanen ym. 2018, 54–55.)

2.4 API:n valintakriteerit

API:n valinnassa olisi hyvä käyttää apuna jotakin tarkistuslistaa, johon listataan valintakriteerit rajapintaa ostettaessa. Tarkistuslistan suositellaan sisältävän ainakin hinnoittelumallit ja palvelutaso, käyttöehdot, API-tarjoajan luotettavuus, datan käsittely ja säilytys sekä tietoturvan toteutus ja määräysten noudattaminen. Ennen ostopäätöstä olisi hyvä päästä testaamaan ja käymään läpi tarkistuslistalla olevat kriteerit. (Moilanen ym. 2018, 160.) Kehittäjälle taas tärkeimpiä kriteereitä ovat luotettavuus, riittävä dokumentaatio, koodiesimerkit, käyttötapaukset sekä helppouden tunne ja intuitiivisuus. Tärkeää on saada API käyttöön mahdollisimman nopeasti ja vaivattomasti. (Moilanen ym. 2018, 109, 164.) Määrittäviä valintakriteereitä ovat myös API-tarjoajan tuki ja mahdollisuus seurata rajapinnan statusta ja analytiikkaa, mikä helpottaa vian etsintää ongelmatilanteissa (Moilanen ym. 2018, 172).

Valintaprosessin helpottamiseksi rajapintojen vertailu ja käyttöönotto on tehty API-katalogeissa helpoksi. Esimerkiksi maailman suurin API-katalogi RapidAPI Hub näyttää rajapinnoista tärkeimmät tiedot jo hakuvaiheessa. Alusta näyttää tuhansille rajapinnoille lyhyen yleiskatsauksen, josta ostaja voi nopeasti nähdä rajapinnan vahvistetun luotettavuuden ja alkuperän sekä kolme mittaria, jotka kertovat rajapinnan suosion alustalla, API-kutsujen keskimääräisen latenssin ja palvelutason eli onnistuneiden kutsujen prosenttiosuuden viimeisen kuukauden aikana. Rajapintakuvauksesta pääsee lukemaan muuan muassa API:iin liittyvää keskustelua ja näkemään hinnoittelutiedot, ja rajapintaa pystytään nopeasti testaamaan interaktiivisen liittymän kautta ennen käyttöönottoa. (RapidAPI 2022b; RapidAPI 2021.)

Hinnoittelumallit vaihtelevat palveluntarjoajan ja rajapinnan mukaan. Moni API-tarjoaja antaa Googlen tavoin asiakkailleen ilmaisen kokeilujakson, joka tarjoaa hyvän mahdollisuuden tutustua rajapintaan ja opetella sen käyttöä ennen ostopäätöstä. Kokeilujakson jälkeen hinta on usein tiettyyn rajaan asti ilmainen ja ainoastaan rajan ylittävistä pyynnöistä tai tiedosta laskutetaan. Hinta taas määräytyy

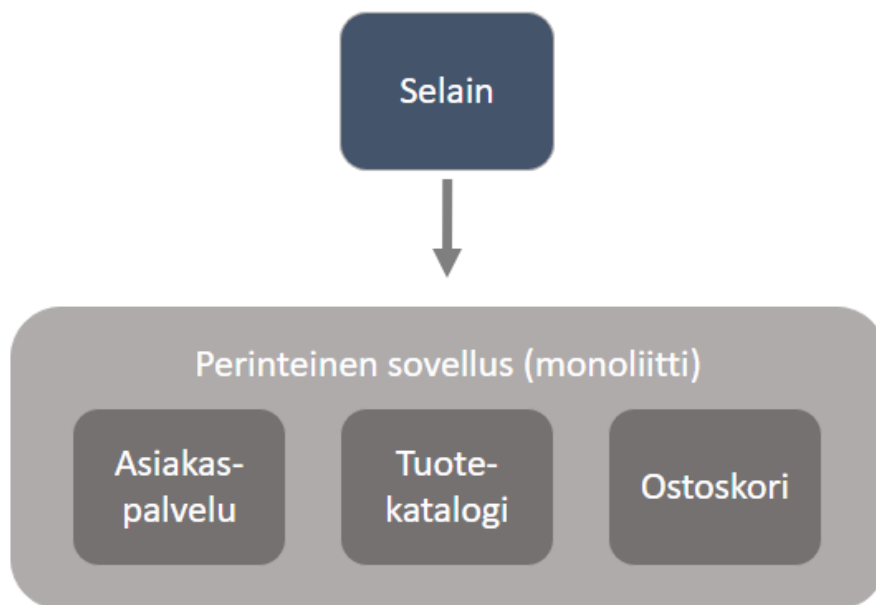
ostettavan rajapinnan, pyyntöjen määrän ja palvelutason mukaisesti. Esimerkiksi Googlen rajapinnoista on saatavilla eri tekniikoilla toteutettuja rajapintoja, joista kalliimpi tuottaa paremman tuloksen, ja lisäpalveluita ostamalla lopputuloksesta voidaan hioa parempi esimerkiksi koneoppimisen avulla. Lisäksi tukipalveluita rajapinnan käyttöön on saatavilla eri hintaluokissa. (Moilanen ym. 2018, 170; Google 2022m; Google 2022n.)

Rajapinnat ovat alttiita väärinkäytöksille verkossa, joten tärkeää on varmistaa ostettavan rajapinnan riittävä tietoturva arkaluonteisten tietojen suojaamista ja käyttäjän todentamista varten. Todennuksella varmistetaan käyttäjän henkilöllisyys ennen palvelimen resurssien käyttämistä. Onnistuneen todennuksen jälkeen käyttäjän oikeus resursseihin voidaan vielä tarkistaa valtuutuksella. Rajapinnan pitäisi käyttää suojattua HTTPS-yhteyttä ja rajapinnan pääsyhallintaan esimerkiksi API-avainta tai OAuth-todennusmenetelmää. API-avain sisältää todennuksen ja OAuth-todennuksen lisäksi myös valtuutuksen, jonka myötä OAuth-menetelmää pidetään turvallisempana vaihtoehtona. (Banach 2022; RapidAPI 2022c.)

Hyvissä ajoin ennen projektin alkua on hyvä ottaa huomioon myös API:n elinkaari, kuten jatkokehitysmahdollisuudet ja lopputulema sekä investoinnin arvo suhteessa sen tuottamaan hyötyyn. Yrityksen on tärkeää pohtia, tuottaako rajapinta lisäarvoa yritykselle. Arvo kannattaa laskea euromääräisesti, jotta saadaan selkeä kuva kehityshankkeen arvon tuotannosta. (Alfame 2015; Moilanen ym. 2018, 26–27, 144.)

3 MIKROPAVELUT

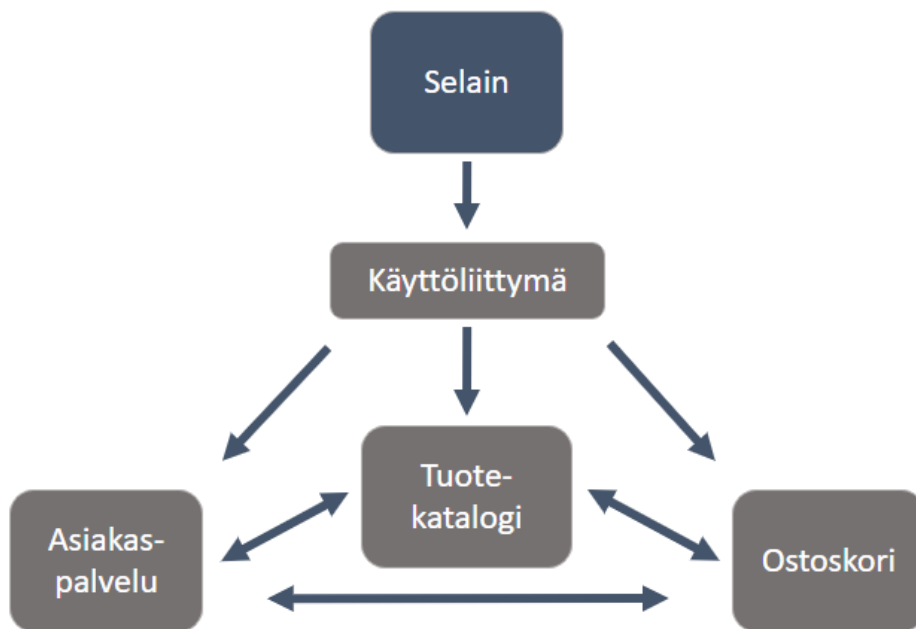
Mikropalvelut ovat yksi tämän hetken keskeisimmistä pilvipalveluiden komponenteista, ja ne ovat monella tapaa perinteisiä monoliittisiä sovelluksia parempia. Mikropalvelut arkkitehtuurimallina on alkanut yleistyä kuitenkin vasta vuonna 2014. (Moilanen ym. 2018, 198.) Perinteinen monoliittinen sovellus sisältää kaikki toiminnallisuudet yhdessä paketissa, ja kaikki toiminnallisuudet ovat riippuvaisia toisistaan. Monoliittista sovellusta päivitettäessä joudutaan päivittämään kerralla koko sovellus, mikä aiheuttaa usein käyttökatkon koko sovelluksessa. Monilla aloilla järjestelmien käyttöikä on vuosikymmeniä, ja useimmista yrityksistä löytyy edelleen vielä järjestelmiä, jotka ovat monoliittisiä. Monoliittinen sovellus on toisaalta helpommin hallittavissa, koska kaikki sovelluksen toiminnallisuudet kulkevat sovelluksen mukana yhtenäisenä kokonaisuutena. Kuviossa 2 on esimerkki monoliittisestä sovelluksesta, joka kuvaa Web-kauppaa, jossa kaikki toiminnallisuudet ovat yhden sovelluksen sisällä (KUVIO 2). (Tarvainen 2021; Angelova 2020; Wallenius 2022c.)



KUVIO 2. Esimerkki monoliittisestä sovelluksesta (mukaiillen Wallenius 2022c)

Mikropalvelu on kokonaisuus, jossa toteutetaan toiminnallisuudet yksittäisinä palveluina, ja toiminnallisuudet kommunikoivat keskenään rajapintojen välityksellä. Mikropalveluita voidaan kehittää tai muuttaa useimmiten yksi toiminnallisuus kerrallaan, jolloin se ei vaikuta kerralla koko järjestelmän

toimintaan. (Angelova 2020; Wallenius 2022c.) Mikropalveluissa yksi komponentti hoitaa yhden tehtävän, kuten kuvion 3 esimerkissä web-kaupan asiakaspalvelu on yksi palvelu ja ostoskori oma palvelunsa. Palvelut kutsuvat toisiaan yhteisten sääntöjen mukaisesti, ja voivat olla myös riippuvaisia toisistaan. Mikropalvelut vievät vähemmän prosessitehoa verrattuna perinteiseen monoliittiseen sovellukseen. Niiden päivittäminen on yksi merkittävimmistä syistä valita mikropalvelut monoliittisen sijaan. Mikropalvelu koostuvat pienemmistä komponenteista, jolloin riittää että päivitetään ainoastaan yksi komponentti ja siitä riippuvaiset komponentit kerrallaan. Se mahdollistaakin nopeammat päivityssykliä, ja näin ollen uusien ominaisuuksien nopeamman julkaisun, koska päivitettävä palvelu on huomattavasti pienempi kuin perinteinen monoliittinen kokonaisuus. (Tarvainen 2021; Wallenius 2022c.)



KUVIO 3. Esimerkki verkkokaupasta toteutettuna mikropalveluina (mukaillen Wallenius 2022c)

Sovelluksen skaalautuvuus eli toiminnan kasvattaminen tai laajentaminen resursseja lisäämällä, on yksi tärkeimmistä tekijöistä ohjelmistokehityksessä. Teknisesti sovellusten protokollat ja laitteisto pystyvät vastaamaan tarvittaessa suurempaan määrään pyyntöjä minuutissa. Tämä on hyvä ottaa huomioon jo kehityksen alkuvaiheessa. Perinteisessä monoliittisessä sovelluksessa skaalaaminen vaatii uuden palvelimen, laajentaen yhden palvelun sijaan koko sovelluskokonaisuutta, jolloin osa palvelimen

suorituskyvystä meneekin pelkän palvelimen pyörittämiseen. Mikropalveluissa voidaan skaalata ainoastaan se palvelu, joka tarvitsee lisää resursseja ja näin ollen tarvitaan vähemmän prosessitehoa saman palvelun tuottamiseen. (Kazmi 2021; Hoogenraad 2021; Wallenius 2022c.)

Yksi suosituimmista tavoista hyödyntää mikropalveluita on laittaa yksittäiset sovellukset kontteihin. Konttiteknoologioista tunnetuin ja käytetyin on Docker. Se mahdollistaa kevyen, mutta tehokkaan ajoympäristön konteissa oleville sovelluksille. Konttiteknoologia on hyvin samantapainen, kuin virtuaalipalvelin, mutta huomattavasti kevyempi. Virtuaalipalvelin sisältää kontin tavoin kaiken mitä ohjelman ajamiseen tarvitaan, mutta se sisältää myös paljon turhia tiedostoja käyttöjärjestelmiseen. Virtuaalipalvelin vie siitä syystä huomattavasti enemmän tilaa ja resursseja konttiteknoologiaan verrattuna. Kontti ei sisällä käyttöjärjestelmää, vaan ohjelman lisäksi ainoastaan toiminnan kannalta tarpeelliset komponentit. Tästä syystä sen suorituskyky on paljon parempi, ja siirtäminen vie vähemmän aikaa. Lisäksi kontti mahdollistaa kehittäjälle samanlaisen standardoidun kehitysympäristön, kuin tuotannossakin. Usein yksi kontti sisältää ohjelman yhden toiminnallisuuden, ja kokonaisen ohjelman kontitukseen tarvitaankin useampi kontti. Yhden toiminnallisuuden sijoittaminen omaan konttiin mahdollistaa nopeamman siirron kehityksestä testiin ja edelleen tuotantoon. Näin ollen kehityssykli nopeutuu huomattavasti. (Wallenius 2022a; Wallenius 2022b.)

4 REST API

REST on suosituin API-arkkitehtuurimalli, jonka pohjalta suurin osa mobiili- ja web-sovelluksista on rakennettu. Esimerkiksi Facebook, Youtube, Google ja Twitter käyttävät REST-rajapintoja sovelluksissaan. REST API:n on kehittänyt Roy Fielding vuonna 2000 skaalautuvien sovellusten väliseen kommunikointiin, ja siitä onkin tullut yleisin tapa sovellusten ja komponenttien yhdistämiseen myös mikropalveluarkkitehtuurissa (IBM Cloud Education 2021). REST rakentuu Hypertext Transfer Protocol (HTTP)-protokollan päälle, jonka varaan oikeastaan koko internet rakentuu. (Juliver 2021b; Full Stack Open 2022; Kivisaari 2016).

REST API:a suositaan muihin arkkitehtuureihin verrattuna, koska se on yksinkertainen ja joustava. Se käyttää vähemmän kaistanleveyttä, jolloin se sopii paremmin käyttöön web-sovelluksissa. (Bansal 2021.) REST antaa tietyt rajoitukset API-kehitykselle, mutta jättää silti kehittäjälle mahdollisuuden rakentaa rajapinta valintansa mukaan lähes millä tahansa menetelmillä (Juliver 2021b). REST API tukee useita tietomuotoja, ja sitä voidaan kehittää lähes millä tahansa koodikielellä (IBM Cloud Education 2021).

REST API:sta puhuttaessa keskeisimpiä termejä ovat asiakas (client), resurssi (resource) ja palvelin (server). Asiakas on API:a käyttävä henkilö tai ohjelma, joka pyytää hakemaan tietoa tai muutoksia sovellukseensa. Resurssilla tarkoitetaan yksilöivän tunnisteiden omaavaa mitä tahansa tietoa, jonka API tarjoaa asiakkaalle, kuten esimerkiksi valokuvaa verkkosivuilla. Resurssin yksilöllinen tunniste on URL-osoitteessa domainin jälkeinen kauttaviivalla erotettu osa. Palvelin vastaanottaa asiakkaan pyynnöt, ja sisältää asiakkaan pyytämät resurssit. (Juliver 2021b.)

4.1 REST API:n arkkitehtoniset säännöt

REST API:a kutsutaan myös nimellä RESTful API. Rajapinta on silloin RESTful, kun se noudattaa kuutta arkkitehtonista sääntöä, joista viisi se vaatii teknisesti toimiakseen ja kuudes on valinnainen. (Red Hat 2020; Juliver 2021b.)

Ensimmäisenä sääntönä asiakkaalla ja palvelimella on oltava yhtenäinen käyttöliittymä. Kommunikonin on tapahduttava samaa viestintäprotokollaa ja muotoilutapaa käyttäen yhteensopivuuden varmistamiseksi. Yleisin REST-rajapintojen viestintäprotokolla on HTTP. Resurssin yksilöllinen tunniste on URL-osoitteessa, ja vastaus pyyntöön saadaan JSON (JavaScript Object Notation) -muodossa. Ilman yhtenäistä viestintää kommunikointi voi johtaa esimerkiksi tiedon katoamiseen tai sekoittumiseen. (Juliver 2021b; Castellani 2020.)

Toiseksi asiakkaan ja palvelimen täytyy olla riippumattomia toisistaan. Asiakas tietää ainoastaan pyytämänsä resurssin tunnisteiden URI:n. Palvelimeen tai asiakassovellukseen tehdyillä muutoksilla ei saa olla vaikutusta toisiinsa. API-tapahtuma alkaa aina asiakkaan aloittamalla pyynnöllä, ja ainoastaan asiakas voi tehdä pyyntöjä palvelimelle ja palvelin vastata asiakkaalle, eikä toisinpäin. (IBM Cloud Education 2021; Juliver 2021b.)

Kolmas sääntö on välimuistin käytettävyys. Aina kun on mahdollista, resurssien olisi hyvä olla välimuistissa asiakas- tai palvelinpuolella, jotta niitä ei tarvitsisi hakea aina uudestaan palvelimelta. Lisäksi palvelimen vastauksessa tulee olla tieto siitä, voidaanko resurssi tallentaa välimuistiin, ja kuinka kauan se siellä säilyy. Välimuistin avulla tiedot saadaan nopeasti ladattua paikallisesti muistista asiakkaan käyttöön, ja se lyhentää latausaikaa ja säästää palvelinresursseja. (IBM Cloud Education 2021; Juliver 2021b.)

Neljänneksi REST API:n istuntojen tulee olla tilattomia (stateless). Jokaisen pyynnön ja vastauksen tulee sisältää siihen tarvittavat tiedot istunnon loppuunsaattamiseksi, eikä palvelimelle tallennu mitään asiakaspyyntöön liittyviä tietoja. Palvelin ei muista aiempien pyyntöjen tietoja, vaan tulkitsee asiakkaan pyynnön aina uudeksi pyynnöksi. Tilattomien siirtojen etuna on se, ettei ylimääräistä muistia tarvita vanhojen tietojen säilyttämiseksi tai hakemiseksi. Lisäksi sovellus on helpommin skaalautuva, koska lisämuistin tarpeesta ei tämän tiedon osalta tarvitse huolehtia. (Juliver 2021b; IBM Cloud Education 2021.)

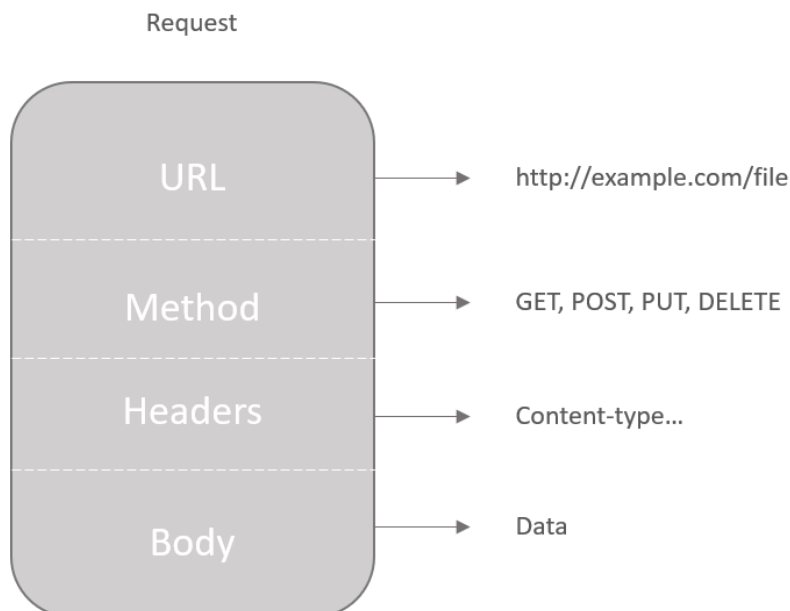
Viidenneksi asiakkaan ja palvelimen kommunikoinnin pitää tapahtua samassa muodossa myös kerroksista riippumatta. Ei pidä olettaa, että istunnon osapuolet ovat suorassa yhteydessä toisiinsa. Todellisuudessa asiakkaan ja palvelimen välissä on välipalvelimia, jotka on esimerkiksi tarkoitettu lisäämään tietoturvaa tai jakamaan verkkoliikennettä. (Juliver 2021b.)

Kuudenneksi REST API:n vastaukset voivat sisältää myös suoritettavaa koodia, ja koodin tulee toimia ainoastaan asiakkaan pyynnöstä. Koodi voi olla esimerkiksi Javaa. (Juliver 2021b; IBM Cloud Education 2021.)

4.2 HTTP-protokolla

HTTP on suosituin ja käytetyin protokolla verkon yli toimivalle viestinnälle. HTTP on suurimmalle osalle kehittäjistä tunnettu ja siitä syystä helpoin käyttää, skaalata ja päivittää. Se on joustava kehittää ja nopeampi kuin muut protokollat. API-pyyntöjen vastaukset voidaan tallentaa välimuistiin, mikä nopeuttaa toistuvien kyselyiden käsittelyä. (Suciu 2021.)

HTTP-pyyntö koostuu yleensä neljästä osasta: resurssipolusta, HTTP-metodista tai verbistä (method), otsikoista (headers) ja viestin rungosta (body). Neljä perusmetodia ovat GET, POST, PUT ja DELETE. Metodi kertoo, mitä resurssille tehdään. Resurssipolku on usein URL-muodossa, jossa domainin jälkeinen kauttaviivalla erotettu osa sisältää resurssin kohteen tiedot. Resurssi voi olla tiedosto, kuva tai vaikka web-sivu, ja tietomuotoja ovat JSON ja XML. (Suciu 2021; Juliver 2021b; Kivisaari 2016;). Kuviossa 4 on esimerkki HTTP-pyyntön rakenteesta.

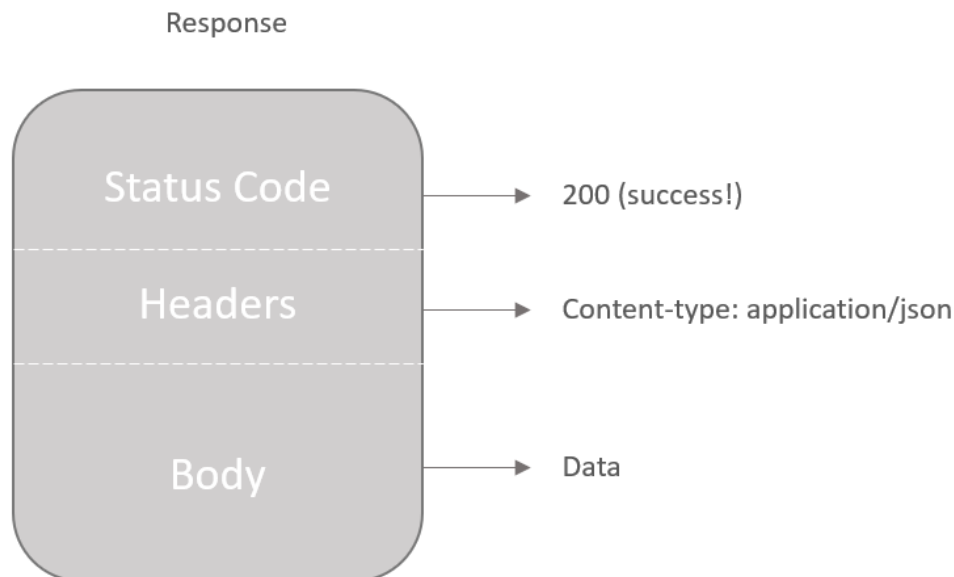


KUVIO 4. Esimerkki HTTP-pyyntön rakenteesta (mukaillen Suciu 2021)

HTTP-metodeista GET-pyyntömenetelmä hakee resurssia tai resursseja tunnuksen perusteella palvelimelta, ja jos pyydetty resurssi löytyy palvelimelta, se vastaa asiakkaalle pyydetyillä tiedoilla. POST luo uusia resursseja. PUT päivittää tietyn resurssin tunnuksen perusteella, ja korvaa olemassa olevan tiedon tai luo uutta mikäli tietoa ei vielä ole olemassa. DELETE taas poistaa resurssin palvelimelta tunnuksen perusteella. (Codecademy 2022; Kumar 2019.)

Otsikko sisältää parametreja, kuten Accept-kentän, joka ilmaisee palvelimelle minkälaisen vastauksen asiakas voi hyväksyä. Content-Type-otsikko taas ilmoittaa palvelimelle muun muassa pyynnön body-osiossa olevan tiedon mediatyypin ja merkkikoodausstandardin (charset). Esimerkiksi image/jpg on kuvatiedoston mediatyyppi ja text/xml on tekstitiedoston mediatyyppi. Otsikkovaihtoehtoja on paljon, ja ne määritetään pyynnön tavoitteiden ja tarpeiden mukaisesti. (Codecademy 2022; Suci 2021; Verma 2021.)

HTTP-vastauksen rakenne eroaa pyynnöstä sillä, että URL-osoite ja menetelmä on korvattu tilakoodilla, joka ilmoittaa asiakkaalle pyynnön onnistumisesta, virheestä tai muusta olennaisesta tiedosta pyyntöön liittyen. Lisäksi vastaus sisältää edelleen otsikot ja rungon (KUVIO 5). (Suci 2021.)



KUVIO 5. Esimerkki HTTP-vastauksen rakenteesta (mukaiillen Suci 2021)

Tilakoodi on kolminumeroinen luku ja jokaisella koodilla on oma merkityksensä. Jokaisella HTTP-metodilla on oma metodikohtainen tilakoodinsa, joka palautetaan aina pyynnön onnistuessaan. Tila-

koodit ryhmitellään viiteen luokkaan numeroiden perusteella: 100–199 informatiivinen, 200–299 vastaus onnistuneelle pyynnölle, 300–399 uudelleenohjaukset, 400–499 asiakasvirheet ja 500–599 palvelinvirheet. (Codecademy 2022; Suciu 2021.) Taulukossa 1 esitetty yleisimmät tilakoodit, jotka kehittäjän on hyvä tietää. Taulukossa 2 on esitetty HTTP-metodikohtaiset tilakoodit.

TAULUKKO 1. Yleisimmät HTTP-pyyntöjen tilakoodit (mukaillen Codecademy 2022)

Tilakoodi	Kuvaus
200 (OK)	Standardi vastaus onnistuneelle HTTP-pyynnölle.
201 (CREATED)	Standardi vastaus HTTP-pyynnölle, joka johti onnistuneeseen tiedon luomiseen.
204 (NO CONTENT)	Standardi vastaus onnistuneelle HTTP-pyynnölle, jossa vastauksen rungossa ei palauteta mitään.
400 (BAD REQUEST)	Pyyntö on virheellinen syntaksivirheen, liian suuren koon tai muun asiakasvirheen vuoksi.
403 (FORBIDDEN)	Asiakkaalla ei ole oikeuksia käyttää resurssia.
404 (NOT FOUND)	Resurssi ei ole saatavilla tällä hetkellä. Se voi olla joko poistettu tai sitä ei ole vielä olemassa.
500 (INTERNAL SERVER ERROR)	Yleinen virhekoodi odottamattomaan vikaan, jos tarkempia tietoja ei ole saatavilla.

TAULUKKO 2. HTTP-metodien tilakoodit onnistuneelle pyynnölle (mukaillen Codecademy 2022)

HTTP-metodi	Tilakoodi
GET	200 (OK)
POST	201 (CREATED)
PUT	200 (OK)
DELETE	204 (NO CONTENT)

Esimerkki yksinkertaistetusta GET-pyyntöstä ja vastauksesta voisi olla seuraavanlainen:

Request: *GET/example.html* Accept: *text/html*. Response: 200 (OK) Content-type: *text/html* (Codecademy 2022).

4.3 HTTPS-protokolla

Hypertext Transfer Protocol Secure (HTTPS) on HTTP-protokollan salattu versio ja ensisijainen verkon siirtoprotokolla tällä hetkellä. W3Techs.comin mukaan tällä hetkellä 78,6 % kaikista verkkosivuista käyttää HTTPS-protokollaa (W3Techs 2022). HTTPS muodostaa salatun yhteyden HTTP-pyyntöille ja vastauksille. Se käyttää salattuun yhteyteen pääasiallisesti Transfer Layer Security eli TLS-protokollaa, joka on Secure Sockets Layer eli SSL-protokollan seuraaja. Selaimessa salatun yhteyden tunnistaa URL-osoitteesta, joka alkaa https:-muodossa, ja selaimen osoitepalkin edessä on lukon kuvake. Salaamaton yhteys taas alkaa http:-muodossa, ja selain varoittaa käyttäjää salaamattomasta sivusta. HTTPS-protokolla salaa käyttäjän ja palvelimen välisen verkkoliikenteen ulkopuolisilta hyökkääjiltä. Esimerkiksi käyttäjän ostotapahtumassa hyökkääjä näkee käyttäjän syöttämät luottokorttitiedot satunnaisina merkkeinä salauksen ansiosta. Ainoastaan lähettäjä ja vastaanottaja näkevät tiedon salaamattomana. (Pecanek 2020; Jukarainen 2019.)

Salattua HTTPS-protokollaa varten verkkosivuille tarvitaan TLS/SSL-varmenne. Varmennetta suositellaan hankittavaksi turvallisuuden ja sitä kautta luotettavuuden takia. Varmenteet voi myöntää tarkastettu toimija Certificate Authority, jonka oikeus varmenteiden myöntämiseen tarkastetaan vuosittain. Varmennetta on saatavissa EV-, OV- ja DV-tason sertifikaateilla. EV-sertifikaatti (Extended Validation) on tunnistuksen korkein taso, ja se edellyttää organisaation ja sivustoa pyörittävän palvelimen tarkistamista. Toiseksi korkein taso on OV-sertifikaatti (Organisation Validation), joka edellyttää organisaation tunnistusta. DV-sertifikaatti (Domain Validation) on matalimman tason tunnistus, ja se tarkoittaa ainoastaan varmenteen hakijan oikeuden käyttää sivustoa pyörittävää palvelinta. DV-sertifikaattia käyttäviä varmenteita on saatavilla ilmaiseksi ja sitä suositellaankin ainoastaan verkkosivustoille, jotka eivät käsittele arkaluonteisia tietoja. EV- ja OV-sertifikaatteja suositellaan verkkosivustoille, jotka sisältävät kirjautumisen tai maksamisen. TLS/SSL-varmenteita tarjoavat eri toimijat ja hinnoittelu alkaa ilmaisesta noin muutamaan sataan euroon vuodessa. (Pecanek 2020; Jukarainen 2019.) Esimerkiksi Telia tarjoaa TLS/SSL-varmennetta yrityksille 99–390 eurolla vuodessa sivuston laajuuden mukaan (Telia).

5 GOOGLLEN TARJOAMAT API:T

Google tarjoaa tällä hetkellä API-kirjastossaan noin 370 julkista API:a. Tarjolla olevia rajapintoja pääsee selaamaan joko Google APIs Explorerin kautta tai Google Cloud Platform-konsolin kautta. Rajapintoja on saatavilla REST-, RPC- tai gRPC-tekniikoilla. API-kirjastossa on rajapintoja eri aihealueittain, kuten karttapalveluita, mobiilirajapintoja, kehitys- ja laskentatyökaluja ja muun muassa mainontaan, big dataan, koneoppimiseen, varastointiin, turvallisuuteen ja verkostoitumiseen liittyviä rajapintoja. Googlen API-kirjaston tarjoamista rajapinnoista puolet on Enterprise-rajapintoja eli nimenomaisesti yrityksille tarkoitettuja rajapintoja. Google on antanut Enterprise-rajapinnoille erityisiä lupauksia, kuten korkean tason vakauden, vanhenemiseen ja muutoksiin liittyvän erityisen huolellisuuden, sekä riittävät varoitukset ja tukipalvelut, jotta yritys voi luottaa pitkäaikaisestikin Googlen tarjoamaan palveluun. Tämän opinnäytetyön käyttötapauksista molemmat ovat Enterprise-rajapintoja. (Google 2022l; Google 2022k; Google 2022o.)

Googlen Cloud Console sisältää paljon hyödyllisiä kehitystyökaluja kehittäjälle. Näitä työkaluja ovat esimerkiksi Cloud Shell, Google Cloud CLI-työkalut, Google Cloud Pricing Calculator sekä Google Cloud Platformin sisältämät monipuoliset työkalut. Googlen kehittäjätyökalut vaativat kuitenkin rekisteröitymisen tai kirjautumisen Google-tunnuksilla ennen käytön aloittamista. Lisäksi Google Cloud -verkkosivuilla on laajasti dokumentteja ja ohjeita esimerkiksi työkalujen käyttöön, hinnoitteluun ja käytön aloittamiseen. Google Cloud Console on saatavilla myös älypuhelimien, mikä helpottaa seuranta esimerkiksi ongelmatilanteissa, ja mahdollistaa perustoimenpiteiden suorittamisen. (Google 2022d; Google 2022a; Google 2022b; Google 2022g)

Google Cloud Platform toimii API:en hallintapaneelina ja seurantatyökaluja. Sen kautta voidaan etsiä rajapintoja API-kirjastosta, päästään käsiksi rajapintakohtaiseen dokumentointiin, hinnoitteluun ja tukipalveluihin. API-dokumentti sisältää ainakin yleiskuvauksen rajapinnasta, ohjeen käyttöönottoa varten, hinnoittelutiedot sekä koodi- ja käyttöesimerkkejä eri koodikielille ja alustoille rajapinnan mukaan. Googlen alustan avulla pystytään ottamaan rajapinta käyttöön nopeasti, ja hallinnoimaan siihen liitettyjä projekteja ja todennusmenetelmiä sekä muun muassa tekemään muutoksia käyttöoikeuksiin, laskutukseen, ja asettamaan tarvittaessa kiintiöitä budjetille. Lisäksi alusta tarjoaa API-kutsuihin liittyvää lokia, analytiikkaa ja mittareita, joilla pystytään valvomaan esimerkiksi API-kutsujen liikennettä, latenssia ja virheprosentteja. Hallintapaneelin kojelautaa on mahdollista muokata itselleen sopivaksi,

mittareita voidaan suodattaa haluttujen arvojen mukaisesti, ja rajapinnoille voidaan määrittää hälytyksiä virhetilanteiden varalta. Näiden tietojen avulla kehittäjä pystyy löytämään myös helpommin ratkaisu mahdollisiin API-kutsuihin liittyviin ongelmiin. (Google 2022a.)

Googlen APIs Explorer on hakutyökalu, joka mahdollistaa rajapintojen nopean hakemisen ja testaamisen ilman koodaamista. Tämä ominaisuus on saatavilla osalle Googlen rajapinnoista, ja sen saa avattua joko dokumentista ”Try it!”-napista tai rajapinnan käyttöönoton yhteydestä ”Try this API”-napista. APIs Explorer on ikään kuin widget, jolla voidaan kokeilla, miten pyyntö ja vastaus toimivat kyseisellä rajapinnalla. Ohjelmalle syötetään pyydetyt tiedot ja painetaan Execute-nappia, jolloin nähdään API-kutsun tulos. (Google 2022c.)

API:n laskutusta, hallintaa ja todennettuja kutsuja varten rajapinnalle täytyy luoda uusi projekti ja määrittää todennusmenetelmä, jolla käyttäjä tunnistetaan. Google suosittelee luomaan erilliset projektit testausta, kehitystä ja tuotantoa varten. Projektin tunnus toimii tunnisteena API-kutsussa. Google tarjoaa uudelle käyttäjälle 300 dollarin ilmaisen kokeilujakson API:n testaamiseen ja käyttöönottoon (GoogleN). Hinnoittelu riippuu rajapinnasta ja sen käyttötarkoituksesta. Esimerkiksi Cloud Translation API:n hinnoittelu määräytyy sen kääntämien merkkien tai asiakirjojen käännettyjen sivujen mukaisesti. API:n perusmallissa ensimmäiset 500 000 merkkiä kuukaudessa on ilmaisia, ja sen jälkeen hinta on noin 18 euroa per miljoona merkkiä. Asiakirjojen hinta määräytyy käytettävän käännoismallin mukaisesti noin 0,007–0,22 euroa sivulta. Google Maps API:n hinnoittelu lasketaan käytetyn karttamallin ja ladattujen karttojen mukaisesti. Google tarjoaa myös kätevän työkalun joillekin rajapinnoille hintojen laskemiseen. Työkaluun valitaan käytettävä API ja syötetään pyydetyt tiedot, jolloin laskuri antaa hinta-arvion sen hetkisten hintojen mukaisesti. Laskutusta voidaan seurata Google Cloud Platformin kautta ja asettaa myös budjetit yllätyksien varalta. (Google 2022m; Google 2022p.)

Kaikki Googlen API:t käyttävät TLS-varmennetta HTTP-kutsuissa. Käyttäjän todennukseen käytetään API-avainta, Service Accountia tai OAuth-menetelmää. API-avain on merkkijono, joka on yleensä salatussa muodossa. Avain on helppo luoda ja nopea käyttää. Se tunnistaa API-kutsun lähettävän projektin, ja pystyy liittämään sen käyttötietoihin ja varmistamaan, että projektilla on käyttöoikeudet kyseiseen resurssiin. OAuth-menetelmä on muodoltaan hyvin samanlainen kuin API-avain, mutta turvallisempi vaihtoehto. Se voidaan liittää käyttäjätileihin, ja sen avulla voidaan määrittää eri oikeuksia käyttäjille. Tätä käytetään usein silloin, kun kehittäjälle halutaan antaa kaikki oikeudet API-toimintoihin ja käyttäjätietoihin. Service Account taas on tili, jota käytetään API:n tai virtuaalikoneen todennukseen

yksittäisen käyttäjän sijaan. Tilille voidaan määrittää IAM-rooleja, jotka määrittävät pääsyn resursseihin. (Kumar 2019; Google 2022g; Google 2022e.)

Google tarjoaa käyttäjilleen eri tukivaihtoehtoja riippuen käytetystä palvelusta ja tukitasosta. Tukitasoja on neljä: Basic, Standard, Enhanced ja Premium. Basic-taso on käyttäjälle ilmainen, ja se sisältää pääsyn dokumentointiin, yhteisötukeen ja laskutustukeen. Standard-tasolla asiakkaalla on 29 dollarin kuukausimaksulla oikeus englanninkieliseen tukeen, ja mahdollisuus tehdä tukipyyntö Googlen alustalla tai ottaa Googleen yhteyttä sähköpostitse. Tällä tasolla vastaus luvataan 4 tunnin sisällä arkipäivisin paikallisten aukioloaikojen puitteissa. Google suosittelee Standard-tasoa pienille ja keskisuurille yrityksille. Laajempia Enhanced- ja Premium-tasoja suositellaan keskisuurille ja suurille yrityksille. Nämä tasot sisältävät alempien tasojen lisäksi muun muassa nopeamman vastauksen tukitapauksiin, tukipalvelut ympäri vuorokauden, laajemman kielituen sekä kolmannen osapuolen teknisen tuen. (Google 2022f.)

6 GOOGLLEN TARJOAMIEN API:EN HYÖDYNTÄMINEN LIIKETOIMINNASSA

Ensimmäisessä käyttötapauksessa kehitettiin yritykselle loppusovellus hyödyntäen Googlen tarjoamaa Enterprise-rajapintaa. Yrityksen pyynnöstä dokumentointi ja loppusovellus salataan tässä opinnäytetyössä. Toinen käyttötapaus tehtiin Googlen Cloud Text-to-Speech API:sta. Käyttötapauksessa käydään läpi rajapintakohtaista dokumentaatiota, hinnoittelua ja koodiesimerkkejä. Nostetaan dokumentaatiosta esille API:n tärkeimpiä ominaisuuksia sekä käyttöönoton kannalta olennaisia ohjeistuksia. Tehdään yksinkertainen esimerkki rajapinnan käyttöönotosta ja testataan sen toimintaa.

Google Cloud Text-to-Speech API on erityisesti yrityskäyttöön suunniteltu Enterprise API. Se muuntaa parhaimmillaan tekstin luonnolliselta kuulostavaksi puheeksi käyttäen neuroverkkomalleja. API:a voidaan hyödyntää esimerkiksi lisäämällä yrityksen verkkosivuille asiakkaan vuorovaikutusta helpottava käyttöliittymä, joka muuntaa tekstin puheeksi. API:n tuottama puhe voidaan määritellä 40 eri kielelle ja 220 eri äänivaihtoehdolla, ja ääni voidaan muuntaa joko Standard- tai WaveNet-muotoon. Nämä äänimuodot käyttävät eri tekniikkaa ja WaveNet kuulostaa hyvin paljon luonnollisemmalta Standardiin verrattuna. Kaikki kielet ja äänivaihtoehdot on lueteltu dokumentissa. Rajapintaa pääsee heti kokeilemaan eri äänivaihtoehdoilla rajapinnan esittelysivulla yksinkertaisen käyttöliittymän avulla. Käyttöliittymään syötetään muunnettava teksti, valitaan halutut asetukset ja painetaan napista, jolloin saadaan tulokseksi esimerkki API-kutsun vastauksesta. (Google 2022h; Google 2022i; Google 2022j.)

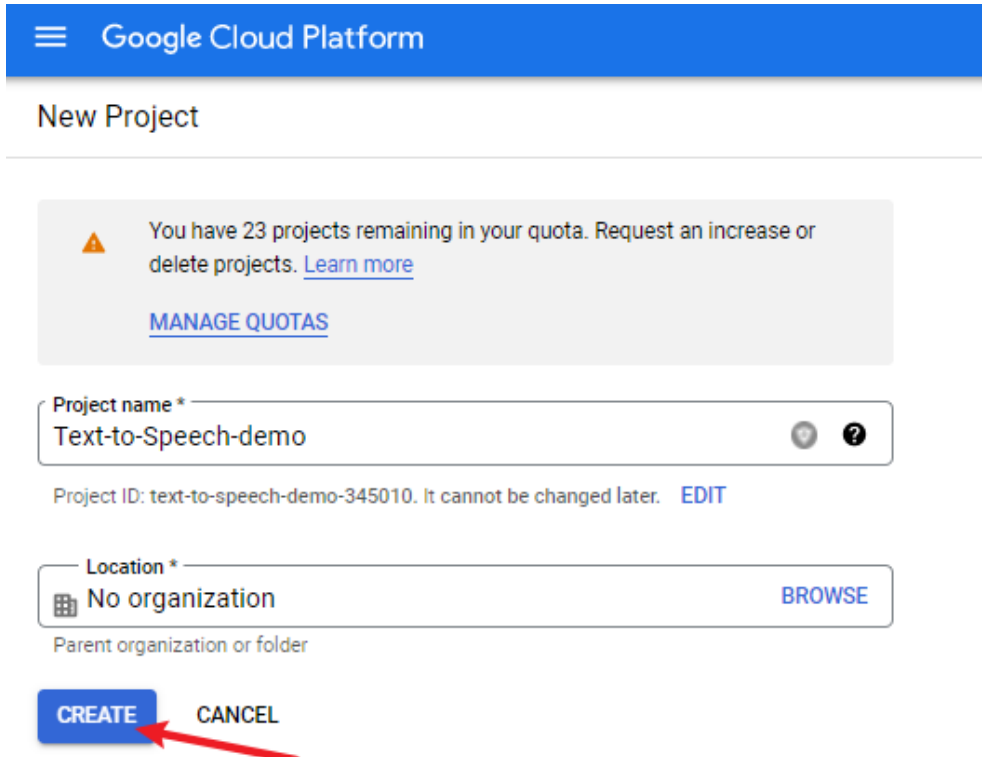
Kyseisen API:n dokumentaatio sisältää laajasti tietoa rajapintaan liittyen, kuten esimerkiksi tarkempaa tietoa hinnoittelusta, kiintiöistä ja rajoituksista sekä rajapintakohtaisista tukipalveluista ja tekniikoista. Ohjeistuksia ja koodiesimerkkejä on saatavilla yleisimmille koodikielille ja eri toiminnallisuuksille erikseen. Koodin voi avata suoraan GitHubissa tai Googlen omassa Cloud Shell Editorissa muokkaamista varten. Lisäksi dokumentti tarjoaa pikaoppaita nopeaan käyttöönottoon. (Google 2022i.)

Cloud Text-to-Speech API:n hinnoittelu lasketaan puheeksi muunnettujen merkkien mukaan kuukausitasolla. Standard-muodossa ensimmäiset 4 miljoonaa merkkiä kuukaudessa ovat ilmaisia ja rajan ylittävät merkit maksavat noin 3,5 euroa miljoonaa merkkiä kohden. WaveNet-äänimuodon ensimmäiset miljoona merkkiä ovat ilmaisia kuukaudessa, ja rajan ylittävistä merkeistä laskutetaan noin 14 euroa miljoonaa merkkiä kohden. Google antaa myös rajoituksia API-kutsuille. Maksimimäärä merkkejä

yhtä pyyntöä kohden on 5 000 ja pyyntöjä voi tehdä 1 000 minuutissa, ja merkkien määrä ei saa olla minuutin aikana yli 500 000. (Google 2022i.)

Cloud Text-to-Speech-rajapinnasta on saatavilla REST API tai RPC API. Asiakaskirjasto on saatavilla seuraaville koodikielille: C#, Go, Java, Node.js, PHP, Python ja Ruby. Esimerkkikoodeja on saatavilla kielille Go, Java, Node.js, PHP ja Python. Tässä esimerkissä tehdään REST API ja käytetään Python-koodikieltä ja koodieditorina Visual Studio Codea. Asiakaskirjaston suorittamista varten tarvitaan todennusmenetelmä, ja tässä esimerkissä käytetään Service Accountia ja tilin yhteyteen luotua API-avainta. (Google 2022i.)

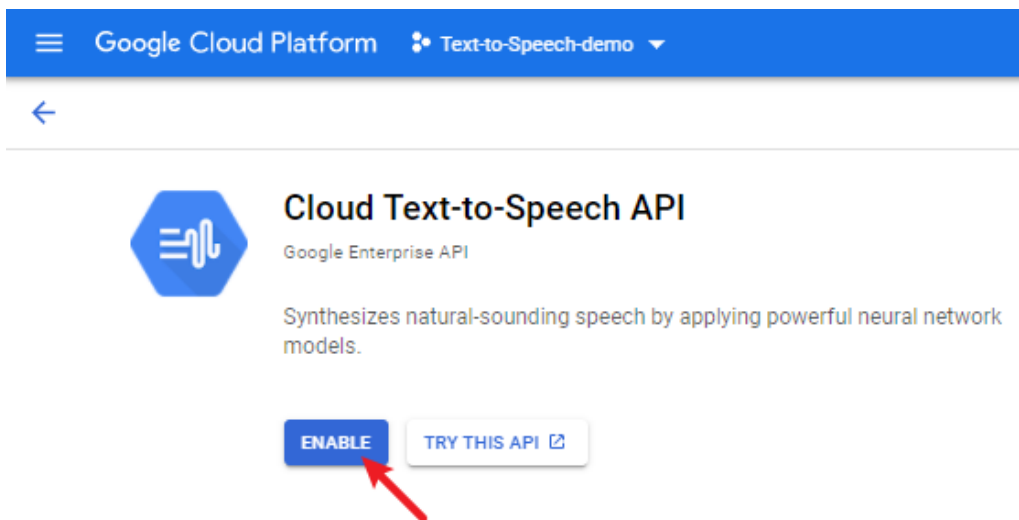
Aluksi kirjaututaan Google-tunnuksilla Google Cloud Platformiin ja avataan vasemmasta sivupalkista ”Billing”-osio ja käydään laittamassa maksukortin tiedot laskutusta varten. Tämän jälkeen luodaan uusi projekti valitsemalla konsolin sivuvalikosta ”IAM & Admin” ja klikataan kohdasta ”Create a Project”. Projektille annetaan valinnainen nimi, ja tässä esimerkissä käytetään nimeä ”Text-to-Speech-demo”. Sijainti jätetään muuttamatta, ja luodaan projekti klikkaamalla ”Create”-napista (KUVA 1). (Google 2022a.)



The screenshot shows the Google Cloud Platform interface for creating a new project. At the top, there is a blue header with the Google Cloud Platform logo and a hamburger menu icon. Below the header, the page title is "New Project". A warning message states: "You have 23 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)". Below the warning is a link labeled "MANAGE QUOTAS". The form contains several fields: "Project name *" with the value "Text-to-Speech-demo", "Project ID: text-to-speech-demo-345010. It cannot be changed later. [EDIT](#)", "Location *" with the value "No organization" and a "BROWSE" button, and "Parent organization or folder". At the bottom, there are two buttons: "CREATE" and "CANCEL". A red arrow points to the "CREATE" button.

KUVA 1. Google Cloud -projektin luominen (Google 2022a)

Tämän jälkeen otetaan juuri luotu projekti käyttöön yläpalkista projektivalikosta. Seuraavaksi etsitään haluttu rajapinta, ja otetaan se käyttöön. Avataan sivuvalikosta API-kirjasto kohdasta ”APIs & Services”, ja klikataan kohdasta ”Library”. Etsitään kirjastosta Cloud Text-to-Speech API, ja siirrytään rajapinnan käyttöönottosivulle. Sivulta nähdään tuotteen lyhyt yleiskuvaus, tärkeimmät tiedot hinnoittelusta, ja päästään käsiksi rajapintakohtaiseen ohjeistukseen ja dokumentaatioon. Tätä rajapintaa pääsee myös kokeilemaan APIs Explorerilla näkyvillä olevasta ”Try This API”-napista. Otetaan nyt API käyttöön klikkaamalla ”Enable”-napista (KUVA 2). (Google 2022a.)



KUVA 2. API:n käyttöönotto (Google 2022a)

Käyttöönoton jälkeen ohjaudutaan API:n hallintasivulle, jossa on nähtävillä rajapintaan liittyvät tärkeimmät mittarit, kiintiöt, todennusmenetelmät ja kustannusten seuranta. Sivulla myös muistutetaan käyttäjälle, että API:n käyttöön vaaditaan todennusmenetelmä. Valitaan seuraavaksi Credentials-välilehti, klikataan kohdasta ”Create credentials”, ja valitaan näkyville tulevasta valikosta ”Service account” (KUVA 3). Valitaan Service Accountille kuvaava nimi, ja jätetään tässä kohtaa valinnaiset kohdat täyttämättä, ja klikataan lopuksi kohdasta ”Done” (KUVA 4). (Google 2022a.)

Google Cloud Platform Text-to-Speech-demo Search Products, resources, docs (/)

API APIS & Services

Enabled APIs & services

Library

Credentials

OAuth consent screen

Domain verification

Page usage agreements

API/Service Details DISABLE API

Cloud Text-to-Speech API
Synthesizes natural-sounding speech by applying powerful neural network models.

By Google Enterprise API

Service name	Type	Status
texttospeech.googleapis.com	Public API	Enabled

METRICS QUOTAS **CREDENTIALS** COST

+ CREATE CREDENTIALS

Credentials compatible with this API

To view all credentials visit [Credentials in APIS & Services](#)

Remember to configure the OAuth consent screen [CONFIGURE CONSENT SCREEN](#)

- OAuth client ID
Requests user consent so your app can access the user's data
- Service account**
Enables server-to-server, app-level authentication using robot accounts
- Help me choose
Asks a few questions to help you decide which type of credential to use

KUVA 3. Service Accountin luominen osa 1 (Google 2022a)

Create service account

- Service account details**

Service account name
TextToSpeechDemo
Display name for this service account

Service account ID *
texttospeechdemo X ↺
Email address: texttospeechdemo@text-to-speech-demo-345010.iam.gserviceaccount.com

Service account description
Describe what this service account will do

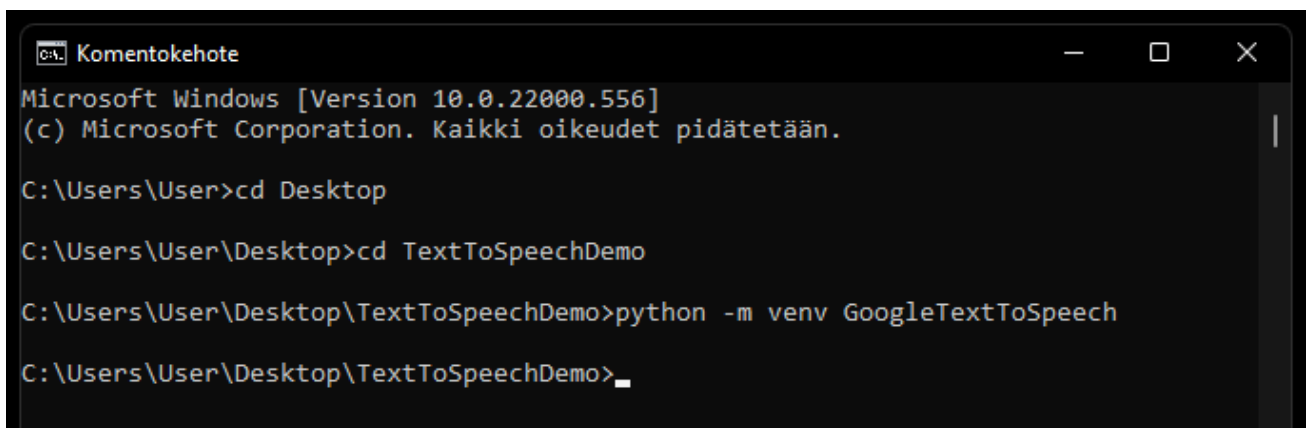
[CREATE AND CONTINUE](#)
- Grant this service account access to project (optional)**
- Grant users access to this service account (optional)**

[DONE](#) [CANCEL](#)

KUVA 4. Service Accountin luominen osa 2 (Google 2022a)

Service Accountin luomisen jälkeen sivusto palaa takaisin API:n hallintasivulle, ja nyt Credentials-välilehdellä on nähtävissä juuri luotu Service Account, jonka yhteyteen luodaan API-avain todennusta varten. Tilin kohdalta klikataan kynän kuvaa, jonka kautta päästään muokkaamaan kyseistä Service Accountia. Tämän jälkeen valitaan yläpalkista ”Keys”, ja klikataan kohdasta ”Add Key” ja edelleen ”Create new key”. Valitaan käytettäväksi avaintyypiksi JSON, ja painetaan ”Create”. Avain tallentuu laitteelle automaattisesti tai valittuun kansioon. (Google 2022a.)

Seuraavaksi luodaan projektille paikallinen kansio, johon kaikki sovellukseen liitetyt tiedostot tallennetaan. Tämän jälkeen avataan kansio komentokehoteella, ja luodaan kansioon Python-virtuaaliympäristö kuvan mukaisella komennolla, ja annetaan kansiolle nimeksi ”GoogleTextToSpeech” (KUVA 5). Siirretään API-avaimen JSON-tiedosto luotuun kansioon, ja avataan kansio Visual Studio Codella muokkaamista varten.



```
C:\Users\User>cd Desktop
C:\Users\User\Desktop>cd TextToSpeechDemo
C:\Users\User\Desktop\TextToSpeechDemo>python -m venv GoogleTextToSpeech
C:\Users\User\Desktop\TextToSpeechDemo>_
```

KUVA 5. Python-virtuaaliympäristön luominen

Luodaan seuraavaksi GoogleTextToSpeech-kansioon uusi tiedosto nimellä demo.py, ja otetaan koodieditorissa käyttöön Python-virtuaaliympäristö. Ladataan Google API:n asiakaskirjasto kehitysympäristöön terminaalien kautta ja Googlen dokumentin ohjaamalla komennolla (KUVA 6). (Google 2022i.)

The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal is running PowerShell commands to set up the development environment. The commands executed are:

```

ps://aka.ms/PSWindows

PS C:\Users\User\Desktop\TextToSpeechDemo\GoogleTextToSpeech> & c:/Users/User/Desktop/TextToSpeechDemo/GoogleTextToSpeech/Scripts/Activate.ps1
(GoogleTextToSpeech) PS C:\Users\User\Desktop\TextToSpeechDemo\GoogleTextToSpeech> & c:/Users/User/Desktop/TextToSpeechDemo/GoogleTextToSpeech/Scripts/python.exe c:/Users/User/Desktop/TextToSpeechDemo/GoogleTextToSpeech/demo.py
Moi
(GoogleTextToSpeech) PS C:\Users\User\Desktop\TextToSpeechDemo\GoogleTextToSpeech> pip install --upgrade google-cloud-texttospeech

```

The command `pip install --upgrade google-cloud-texttospeech` is highlighted with a red box in the terminal. The status bar at the bottom indicates the current file is `demo.py` in the `GoogleTextToSpeech` project, using Python 3.9.10 with a virtual environment.

KUVA 6. API:n asiakaskirjaston asentaminen kehitysympäristöön

Demo.py-tiedostoon luodaan yksinkertainen sovellus Googlen koodiesimerkin mukaisesti. Sovellus kutsuu Googlen rajapintaa annetuilla parametreilla, ja vastaus saadaan Googlen rajapinnasta äänitiedoston muodossa (KUVA 7). Sovellukseen on liitetty tiedostopolku json-tiedoston sisältämään API-avaimeen, jonka perusteella Google todentaa API-kutsun lähettäneen käyttäjän. Esimerkissä rajapintaa pyydetään palauttamaan annettu teksti naisen äänellä suomeksi ja mp3-tiedostomuodossa. Rajapinta vastaa odotetusti pyyntöön ilman virheilmoituksia, mikä on nähtävillä myös Google Cloud Platform-hallintapaneelistä, jossa virheilmoitusten määrä on nolla (KUVA 8). Kuvasta 7 nähdään myös sovelluksen tuloksena syntynyt demo.mp3-tiedosto. (Google 2022i; Google 2022a).

```

demo.py - GoogleTextToSpeech - Visual Studio Code
demo.py > ...
1 import os
2 from google.cloud import texttospeech
3
4 os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'serviceAccountKey.
  json'
5
6 client = texttospeech.TextToSpeechClient()
7
8 inputText = texttospeech.SynthesisInput(text='Hei, tämä on esimerkki
  Googlen rajapinnan käytöstä!')
9
10 voice = texttospeech.VoiceSelectionParams(
11     language_code="fi-FI", ssml_gender=texttospeech.SsmlVoiceGender.
  FEMALE
12 )
13
14 audio_config = texttospeech.AudioConfig(
15     audio_encoding = texttospeech.AudioEncoding.MP3
16 )
17
18 response = client.synthesize_speech(
19     input=inputText, voice=voice, audio_config=audio_config
20 )
21
22 with open("demo.mp3", "wb") as out:
23     out.write(response.audio_content)
24     print('Audio luotu onnistuneesti tiedostoon "demo.mp3"')
25

```

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

```

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\User\Desktop\TextToSpeechDemo\GoogleTextToSpeech> & c:/Users/User/Desktop/TextToSpeechDemo/GoogleTextToSpeech/Scripts/Activate.ps1
(GoogleTextToSpeech) PS C:\Users\User\Desktop\TextToSpeechDemo\GoogleTextToSpeech> & c:/Users/User/Desktop/TextToSpeechDemo/GoogleTextToSpeech/Scripts/python.exe c:/Users/User/Desktop/TextToSpeechDemo/GoogleTextToSpeech/demo.py
Audio luotu onnistuneesti tiedostoon "demo.mp3"
(GoogleTextToSpeech) PS C:\Users\User\Desktop\TextToSpeechDemo\GoogleTextToSpeech>

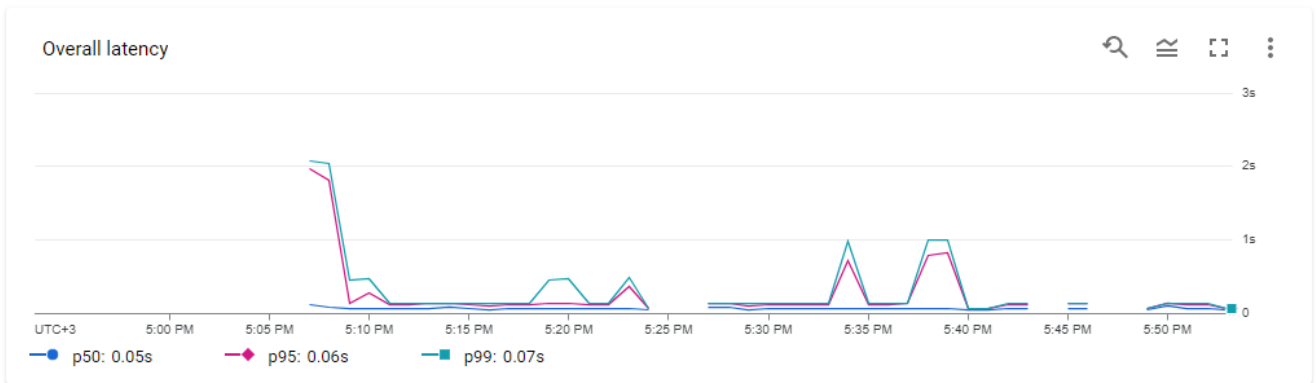
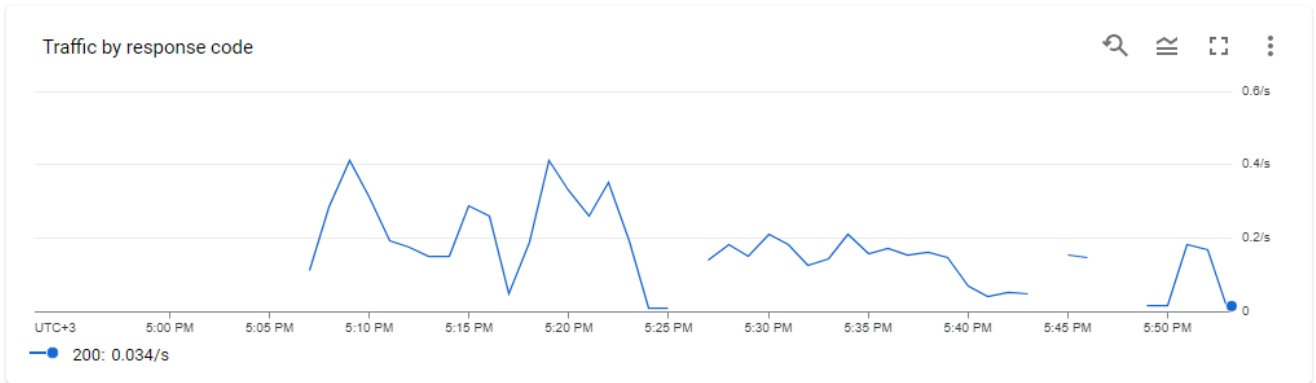
```

powershell
 Python
 Python
 Python

Ln 31, Col 1 Spaces: 4 UTF-8 CRLF Python 3.9.12 ('GoogleTextToSpeech': venv) Go Live Reload

KUVA 7. Esimerkkisovellus Cloud Text-to-Speech API:sta (Google 2022i)

Testauksen jälkeen rajapinnan hallintapaneelistä voidaan nähdä myös testin tuottamat liikennemäärät ja latenssit. Rajapinta tuotti viimeisen tunnin aikana 429 onnistunutta pyyntöä ilman virheitä. Pyyntöistä 95 %:lla latenssi oli alle 1,455 sekuntia ja keskimääräinen latenssi 0,092 sekuntia (KUVA 8). (Google 2022a.)



Methods

Method ↑	Requests	Errors	Avg latency	99th percentile latency ?
google.cloud.texttospeech.v1.TextToSpeech.SynthesizeSpeech	429	0	0.092 seconds	1.455 seconds

KUVA 8. Rajapinnan testin tulokset (Google 2022a)

7 TULOKSET JA POHDINTAA

Opinnäytetyön tavoitteena oli tutkia Googlen tarjoamia rajapintoja ja niiden hyödyntämistä yrityksen liiketoiminnassa. Google tarjoaa yrityskäyttöön sopivia rajapintoja, ja lupaa asiakkailleen vakaan ja turvallisen käyttökokemuksen. Rajapintojen hallintapaneeli on hyödyllinen ja selkeä, vaikkakin tuki-palvelut jäävät vähäisiksi ilmaisella perustasolla. Kuitenkin yritys voi hyvinkin edullisesti hyödyntää Googlen rajapintoja, ja saada niistä lisäarvoa liiketoimintaansa. Google tarjoaa alkuun käyttäjälle ilmaisen kokeilujakson, joka on hyödyllinen tapa aloittaa rajapinnan kehitystyö ja testaus.

Käyttötapauksien osalta rajapintojen tutkimisessa ja käyttöönotossa huomattiin, että Google tarjoaa valtavasti dokumentaatiota ja verkkosivustoja käyttöönottoon. Tämä on tietysti kehittäjän kannalta hyvä asia, mutta toisaalta myös paljon aikaa vievä, mikäli haluaa olla hyvin perillä asioista. Dokumentaatiossa oli myös jonkin verran toistuvuutta, ja samoja koodiesimerkkejä löytyi useasta paikasta. Omasta näkökulmasta dokumentointi voisi olla hieman suppeampaa ja selkeämpää. Koodiesimerkit olivat kuitenkin hyödyllisiä, ja niitä oli riittävästi. Lisäksi hallintapaneelin seuranta- ja analytiikkatyökalut antoivat hyödyllistä tietoa rajapinnan käyttäytymisestä erityisesti mahdollisten ongelmatilanteiden varalta.

Teknisesti ensimmäisen käyttötapauksen osalta rajapinnan tuottama tulos oli kohtalaisen hyvä, ja tulosta olisi mahdollista jatkokehittää vielä paremmaksi ostamalla rajapintaan liittyviä lisäpalveluita. Toinen käyttötapaus tuotti halutun tuloksen ilman virheitä, ja käyttöönotto oli helppoa. Pohdittavaksi toisen käyttötapauksen osalta jäi kuitenkin se, kuinka nopeasti rajapinta toimii suuremmalla määrällä tekstiä ja tehokäytössä.

Yleisesti ottaen Googlen rajapinnat ovat yrityskäyttöön vakaita ja luotettavia. Tietoturva on hyvällä tasolla ja API-kutsut toimivat onnistuneesti. Lisäksi rajapintojen käyttöönotto oli helppoa hyvän dokumentaation ja koodiesimerkkien ansiosta. Dokumenteista löytyi kaikki tarvittava tieto, vaikkakin dokumenttien selaamiseen saa kulutettua huomattavasti aikaa. Kehittäjälle jätetään myös joustavasti mahdollisuuksia käyttää itselleen sopivia tekniikoita. API:en hallintapaneeli oli helppokäyttöinen ja analytiikka hyödyllistä. Lisäksi rajapintojen käyttöönoton tueksi löytyi paljon yhteisön tukea, ja Googlen tukipalveluitakin on saatavilla kohtuulliseen hintaan tarvittaessa.

LÄHTEET

- Alfame. 2015. *12 vinkkiä ohjelmistokehityksen ostajalle*. Saatavissa: https://www.alfame.com/hs-fs/hub/395117/file-2654455542-pdf/files/12_vinkkia_ohjelmistokehityksen_ostajalle_Alfame.pdf. Viitattu 23.2.2022.
- Angelova, A. 2020. *10 Reasons Why Microservices Are the Future*. Wiredelta. Saatavissa: <https://wiredelta.com/10-reasons-why-microservices-are-the-future/>. Viitattu 23.2.2022.
- Banach, Z. 2022. *How to ensure REST API security*. Saatavissa: <https://www.invicti.com/blog/web-security/rest-api-web-service-security/>. Viitattu 3.4.2022.
- Bansal, S. 2021. *REST API (Introduction)*. Saatavissa: <https://www.geeksforgeeks.org/rest-api-introduction/>. Viitattu 14.3.2022.
- Castellani, S. 2020. *What are the different types of APIs?* Axway. Saatavissa: <https://blog.axway.com/amplify-products/api-management/different-types-apis>. Viitattu 6.3.2022.
- Codecademy. 2022. *What is REST?* Saatavissa: <https://www.codecademy.com/article/what-is-rest>. Viitattu 16.3.2022.
- Fourtress. 2020. *REST API TESTING*. Saatavissa: <https://fourtress.nl/en/blog/rest-api-testing/>. Viitattu 6.3.2022.
- Full Stack Open. 2022. *Node.js ja Express*. Saatavissa: https://fullstackopen.com/osa3/node_js_ja_express#rest. Viitattu 14.3.2022.
- Google. 2022a. *Google Cloud Platform*. Saatavissa: <https://console.cloud.google.com/>. Viitattu 20.3.2022.
- Google. 2022b. *Google Cloud Pricing Calculator*. Saatavissa: <https://cloud.google.com/products/calculator>. Viitattu 20.3.2022.
- Google. 2022c. *Using the APIs Explorer*. Saatavissa: <https://cloud.google.com/monitoring/api/apis-explorer>. Viitattu 20.3.2022.
- Google. 2022d. *Cloud Console-Your integrated Google Cloud Platform management console*. Saatavissa: <https://cloud.google.com/cloud-console>. Viitattu 20.3.2022.
- Google. 2022e. *Service accounts*. Saatavissa: <https://cloud.google.com/iam/docs/service-accounts>. Viitattu 22.3.2022.
- Google. 2022f. *Getting support with Cloud Customer Care*. Saatavissa: <https://cloud.google.com/support/docs/>. Viitattu 22.3.2022.
- Google. 2022g. *Google Cloud APIs*. Saatavissa: https://cloud.google.com/apis/docs/overview#about_cloud_apis. Viitattu 22.3.2022.

- Google. 2022h. *Text-to-Speech*. Saatavissa: <https://cloud.google.com/text-to-speech>. Viitattu 22.3.2022.
- Google. 2022i. *Text-to-Speech documentation*. Saatavissa: <https://cloud.google.com/text-to-speech/docs>. Viitattu 23.3.2022.
- Google. 2022j. *Standard and WaveNet voices*. Saatavissa: <https://cloud.google.com/text-to-speech/docs/wavenet>. Viitattu 23.3.2022.
- Google. 2022k. *Google APIs Explorer*. Saatavissa: <https://developers.google.com/apis-explorer>. Viitattu 15.3.2022.
- Google. 2022l. *Welcome to the API Library*. Saatavissa: <https://console.cloud.google.com/apis/library?project=hello-world-sean-200116>. Viitattu 15.3.2022.
- Google. 2022m. *Cloud Translation pricing*. Saatavissa: <https://cloud.google.com/translate/pricing>. Viitattu 2.4.2022.
- Google. 2022n. *Google Cloud pricing*. Saatavissa: <https://cloud.google.com/pricing>. Viitattu 3.4.2022.
- Google. 2022o. *Google Enterprise APIs*. Saatavissa: <https://cloud.google.com/apis/docs/resources/enterprise-apis>. Viitattu 3.4.2022.
- Google. 2022p. *Getting started*. Saatavissa: <https://cloud.google.com/apis/docs/getting-started>. Viitattu 3.4.2022.
- Haglund, J. 2018. *Onnistunut API-strategia kääntää dataa tuotoksi*. Saatavissa: <https://www.al-fame.com/blog/onnistunut-api-strategia-kaantaa-dataa-tuotoksi>. Viitattu 25.3.2022.
- Hakola, M. 2018. *Sisäiset API:t tukevat ketterämpää kehittämistä*. Saatavissa: <https://blog.digia.com/sisaiset-apit-tukevat-ketterampaa-kehittamista>. Viitattu 25.3.2022.
- Hoogenraad, W. 2021. *Skaalautuvuus ohjelmistovaatimuksena, merkityksenä ja määritelmänä*. Saatavissa: <https://www.itpedia.nl/fi/2021/07/20/schaalbaarheid-als-software-requirement-betekenis-en-definitie/>. Viitattu 9.3.2022.
- IBM Cloud Education. 2021. *REST APIs*. Saatavissa: <https://www.ibm.com/cloud/learn/rest-apis>. Viitattu 14.3.2022.
- IBM Cloud Education. 2020. *Application Programming Interface (API)*. Saatavissa: <https://www.ibm.com/cloud/learn/api>. Viitattu 6.3.2022.
- Jukarainen, M. 2019. *Https-salaus ja mitkä ovat http ja https erot*. Saatavissa: <https://mikaeljukarainen.com/https-salaus/>. Viitattu 28.3.2022.
- Juliver, J. 2021a. *4 Types of APIs All Marketers Should Know*. Saatavissa: <https://blog.hubspot.com/website/types-of-apis>. Viitattu 14.3.2022.
- Juliver, J. 2021b. *REST APIs: How They Work and What You Need to Know*. Saatavissa: <https://blog.hubspot.com/website/what-is-rest-api>. Viitattu 14.3.2022.

- Kazmi, R. 2021. *Scalable Applications: Curious Why Scalability Matters?* Saatavissa: <https://www.koombea.com/blog/why-scalability-matters-for-your-app/>. Viitattu 9.3.2022.
- Kivisaari, T. 2016. *API:t ovat modernin integraatiostrategian ydin*. Saatavissa: <https://blog.digia.com/rest-api>. Viitattu 14.3.2022.
- Kumar, N. 2019. *Introduction to APIs in Google*. Saatavissa: <https://nidhinkumar.medium.com/introduction-to-apis-in-google-4273ef99a5b0>. Viitattu 17.3.2022.
- Lane, K. 2015. *In The Future There Will Be No Public vs. Private APIs*. Saatavissa: <https://apievangelist.com/2015/02/03/in-the-future-there-will-be-no-public-vs-private-apis/>. Viitattu 14.4.2022.
- Levin, G. 2017. *Internal vs External APIs*. Saatavissa: <https://blog.restcase.com/internal-vs-external-apis/>. Viitattu 14.4.2022.
- Macoveiciuc, A. 2020. *Beginner's Guide to APIs, Protocols and Formats*. Saatavissa: <https://frontend-digest.com/beginners-guide-to-apis-protocols-and-data-formats-f80cf7f30425>. Viitattu 7.3.2022.
- Moilanen, J. 2019. *Vierasbloggaus: API-talous mullistaa liiketoiminnan mahdollisuudet*. Saatavissa: <https://www.alfame.com/blog/api-talous-mullistaa-liiketoimintamahdollisuudet>. Viitattu 1.4.2022.
- Moilanen, J., Niinioja, M., Seppänen, M., Honkanen, M. 2018. *API-talous 101*. Helsinki: Alma Talent Oy.
- Pecanek, M. 2020. *What is HTTPS? Everything You Need to Know*. Saatavissa: <https://ah-refs.com/blog/what-is-https/>. Viitattu 28.3.2022.
- Programmableweb. 2022. *Search the Largest API Directory on the Web*. Saatavissa: <https://www.programmableweb.com/category/all/apis>. Viitattu 6.3.2022.
- RapidAPI. 2022a. *The Next-Generation API Platform*. Saatavissa: <https://rapidapi.com/>. Viitattu 2.4.2022.
- RapidAPI. 2022b. *Welcome to the RapidAPI Hub*. Saatavissa: <https://rapidapi.com/hub>. Viitattu 2.4.2022.
- RapidAPI. 2021. *FAQs - RapidAPI Hub*. Saatavissa: <https://docs.rapidapi.com/docs/faqs>. Viitattu 3.4.2022.
- RapidAPI. 2022c. *API Authentication – What is API Authentication?* Saatavissa: <https://rapidapi.com/blog/api-glossary/api-authentication/>. Viitattu 3.4.2022.
- Red Hat. 2020. *What is a REST API?* Saatavissa: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>. Viitattu 14.3.2022.
- Suciu, D. 2021. *API Architecture: The HTTP Protocol and Its Importance*. Saatavissa: <https://medium.com/api-world/api-architecture-the-http-protocol-and-its-importance-aeba0fe46f91>. Viitattu 15.3.2022.

Tarang, A. 2021. *Application Programming Interfaces (API) and its Types*. Saatavissa: <https://www.geeksforgeeks.org/application-programming-interfaces-api-and-its-types/>. Viitattu 6.3.2022.

Tarvainen, J. 2021. *Mikropalvelut – mitä käyttöä arjessa?* Saatavissa: <https://www.sol-teq.com/fi/blogi/mikropalvelut-mita-kaytttoa-arjessa>. Viitattu 10.3.2022.

Telia. *Turvaa saattisi TLS/SSL-sertifikaatilla*. Saatavissa: <https://www.telia.fi/yrityksille/infrapalvelut/tietoturva/ssl-palvelinvarmenne>. Viitattu 1.4.2022.

Verma, P. 2021. *HTTP headers | Content-Type*. Saatavissa: <https://www.geeksforgeeks.org/http-headers-content-type/>. Viitattu 18.3.2022.

W3Techs. 2022. *Usage statistics of Default protocol https for websites*. Saatavissa: <https://w3techs.com/technologies/details/ce-httpsdefault>. Viitattu 1.4.2022.

Wallenius, N. 2022a. *Konttiteknologia – mitä kontit ovat ja mitä hyötyä niistä on?* Saatavissa: <https://niklaswallenius.fi/konttiteknologia-mita-hyotya/>. Viitattu 24.3.2022.

Wallenius, N. 2022b. *Mikä on Docker ja mitä hyötyä siitä on?* Saatavissa: <https://niklaswallenius.fi/mika-on-docker/>. Viitattu 24.3.2022.

Wallenius, N. 2022c. *Mitä mikropalvelut ovat?* Saatavissa: <https://niklaswallenius.fi/mikropalvelut/>. Viitattu 8.3.2022.