

Riku Pelkonen

# OHJELMOINTIRAJAPINNAT KILPAILUTEKIJÄNÄ



JYVÄSKYLÄN YLIOPISTO  
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA  
2021

# TIIVISTELMÄ

Pelkonen, Riku

API kilpailutekijänä

Jyväskylä: Jyväskylän yliopisto, 2021, 29 s.

Tietojärjestelmätiede, kandidaatintutkinto

Ohjaaja(t): Halttunen, Veikko

Tässä tutkimuksessa tarkastellaan kirjallisuuskatsauksen keinoin, millä eri tavoin ohjelmointirajapinnat voivat olla merkittävä kilpailutekijä yrityksille. Tutkimuksessa määritellään mikä ohjelmointirajapinta on, mitkä ovat eri ohjelmointirajapintaluokkien eroavaisuudet, sekä käsitellään ohjelmointirajapintojen historiaa ja kehitystä paremman yleiskuvan saamiseksi aiheesta. Tutkimustuloksien johtopäätökset osoittavat ohjelmointirajapintojen voivan olevan merkittävä kilpailutekijä kasvattamalla organisaation sisäistä tehokkuutta ja innovointia, tehostamalla markkinointia sekä mahdollistamalla datan kaupallistamisen. Suurimmat taloudelliset hyödyt ohjelmointirajapintojen avulla saavutetaan sisäisiä kustannuksia alentamalla, mutta positiivinen tuottoaste voidaan saavuttaa myös suorien ja epäsuorien ohjelmointirajapintojen käyttömaksujen ansiosta. On myös selvää, että ohjelmointirajapintojen hyötyjä ei saavuteta, ellei niitä käytä kukaan. Tutkielmassa ei oteta kantaa, kuinka ohjelmointirajapinta tulisi jalkauttaa tai toteuttaa näiden hyötyjen saavuttamiseksi. Tutkielmasta käy ilmi myös, että ohjelmointirajapintojen mahdollistamia hyötyjä tavoitellessaan organisaation tulisi laatia juuri sen liiketoimintaan soveltuva strategia ohjelmointirajapinnan käyttöönotosta tai -hyödyntämisestä. Tutkielmassa keskitytään ennen kaikkea liiketoiminnan sekä ohjelmointirajapinnan tarjoajan näkökulmaan.

Asiasanat: API, Application Programming Interface, ohjelmointirajapinta, kilpailutekijä, liiketoiminta

## **ABSTRACT**

Pelkonen, Riku

API as a competitive advantage

Jyväskylä: Jyväskylän yliopisto, 2021, 29 pp.

Information Systems, Bachelor's Thesis

Supervisor(s): Halttunen, Veikko

The goal of this research is to recognize different ways of how API can be significant competitive advantage for business. This research has been performed by literature review. For achieving better general view of the topic, I am defining different API-classes and have an overview of API history and development. Findings show that API can be significant competitive advantage by raising organization's internal performance and innovation, optimizing marketing, and allowing data monetization. Most significant economic advantages are achieved by lowering internal costs. Positive return on investment can also be achieved by direct and indirect API-access fee. It is clear though for achieving these advantages, one must utilize the API, by itself it does not produce said advantages. This research does not take a position how organizations should implement or practice API to gain these advantages. In this research it is also shown that aiming to benefit APIs advantages one should prepare strategy that fits to itself in order to implement API. This research is focused most of all from the standpoint of business and API-provider.

Keywords: API, Application Programming Interface, competitive advantage, business

## TAULUKOT

TAULUKKO 1 API-luokittelu.....	9
--------------------------------	---

# SISÄLLYS

TIIVISTELMÄ .....	2
ABSTRACT .....	3
TAULUKOT .....	4
SISÄLLYS.....	5
1 JOHDANTO.....	6
2 OHJELMOINTIRAJAPINTA.....	8
2.1 Määrittely.....	8
2.2 Kehitys.....	12
3 OHJELMOINTIRAJAPINTOJEN VAIKUTUKSET .....	17
3.1 Hyödyt.....	17
3.2 Hyötyjen taloudelliset vaikutukset .....	19
4 JOHTOPÄÄTÖKSET JA YHTEENVETO .....	24
LÄHTEET .....	26

# 1 JOHDANTO

Ohjelmointirajapinnat (engl. application programming interface, API) ovat nykypäivänä lähes kaikkialla siellä, missä on tietotekniikkaakin. Ne ovat meidän kaikkien käyttämien lukemattomien järjestelmien ja sovelluksien keskiössä taustavaikuttajina (Meng, Steinhardt & Schubert, 2018), mahdollistaen muun muassa eheän tiedonsiirron sekä monille organisaatioille elintärkeän pilvilaskennan. Ohjelmointirajapinnat mielletään usein vain teknologiaksi, joka mahdollistaa järjestelmien integraatiot. Pyrin tässä tutkielmassa laajentamaan tuota käsitystä ja selvittämään tutkimusongelman, miten eri tavoin ohjelmointirajapinnat voivat olla merkittävä kilpailutekijä erilaisille organisaatioille? Keskityn tutkielmassa erityisesti liiketoiminnan sekä ohjelmointirajapinta-tarjoajan näkökulmaan ja ohjelmointirajapintojen mahdollistamiin hyötyihin, tutkielmassa ei oteta kantaa teknisiin kysymyksiin ja ongelmiin, kuten kuinka ohjelmointirajapinta pitäisi rakentaa tai käyttöönottaa erilaisissa organisaatioissa. Tutkimusongelmaa pyritään ratkaisemaan selvittämällä ensin mahdollisimman kattavasti ohjelmointirajapintojen mahdollistamat hyödyt kirjallisuuskatsauksen keinoin useita eri tutkimuksia analysoiden sekä kooten niiden johtopäätöksiä. Selvitystyössä yksi ilmi tullut hyötykategoria, tehokkuutta kasvattavat hyödyt, osoittautuu tutkimuksen kannalta tärkeimmäksi hyödyksi taloudellisesta näkökulmasta katsottuna sisäisiä kustannuksia alentamalla. Tämän lisäksi tässä tutkielmassa osoitetaan ohjelmointirajapintojen voivan olevan merkittävä kilpailutekijä erilaisille organisaatioille muun muassa kasvaneen innovoinnin ja datan kaupallistamisen avulla.

Motivaatio tutkimukselle syntyi havainnosta, ettei ohjelmointirajapintojen hyötyjä liiketoiminnan näkökulmasta ole tutkittu paljoa. Erityisesti taloudellisilla tunnusluvuilla mitattuja tutkimuksia on toteutettu hyvin niukasti, mutta myös kokonaisvaltainen yhteenveto hyödyistä on ollut puutteellista. Tutkielma on toteutettu kirjallisuuskatsauksena, jonka tavoitteena on näiden tutkimuksien tunnistamien hyötyjen yhteen kokoaminen sekä saada vastaus tutkimusongelmaan. Aineisto kirjallisuuskatsaukseen on haettu ensisijaisesti Jykdokin, Google Scholarin ja Scopuksen tietokannoista, aineiston luotettavuutta on arvioitu JUFO-luokituksen ja viittausten lukumäärien avulla. Tutkimuksen hyväksytyt

aineistot ovat JUFO-luokitukseltaan perustasoa, johtavaa tasoa tai korkeinta tasoa. Tieteellisen aineiston lisäksi tutkielmaan on otettu lähteiksi myös organisaatioiden itsensä julkaisemia uutisia, kaksi tutkimusaiheen ytimeen keskittyvää yliopisto-tutkimusta, sekä yksi aiheelle merkittävä blogikirjoitus. Näiden JUFO-luokittelemattomien lähteiden käytössä on huomioitu erityisen tarkka lähdekriittisyys ja väitteiden paikkansapitävyyttä on pyritty vahvistamaan muista lähteistä. Aineistoa tutkimukseen on haettu termein "api", "application programming interface", "api economy", "api benefits", "rest api", "open api", "private api", "public api", "api performance", "monetizing api", "web api", "api revenue", "api advantage", "api productivity".

Tutkielman luvussa 2 selvennetään mikä ohjelmointirajapinta on, sekä rajataan mihin näkökulmiin tutkielmassa keskitytään. Luvussa määritellään oleelliset ohjelmointirajapintaluokat ja pyritään tekemään selkeät erot näiden luokkien välille, vaikka ne eivät sitä tutkijoiden keskuudessa ole. Seuraavaksi käsitellään ohjelmointirajapintojen historiaa ja kehitystä ensisijaisesti liiketoiminnan näkökulmasta. Luvussa ilmenee myös, minkä merkittävien teknologioiden kehityksen keskiössä ohjelmointirajapinnat ovat olleet, sekä mitkä ovat ohjelmointirajapintojen trendejä tänä päivänä. Kolmannessa luvussa käsitellään ohjelmointirajapintojen mahdollistamia hyötyjä, rajattuina tehokkuutta kasvattaviin tekijöihin, markkinoinnin hyötyihin, sekä datan kaupallistamiseen ja siten pyritään vastaamaan tutkimusongelmaan. Myöhemmin luvussa tarkastelen tarkemmin kahta tutkimusta, joiden avulla pyrin vastaamaan kysymykseen, missä määrin nämä hyödyt voivat organisaatiota taloudellisesti hyödyntää. Lopuksi teen yhteenvedon tutkielmasta ja sen tuloksista, sekä pohdin jatkotutkimustarvetta.

## 2 OHJELMOINTIRAJAPINTA

Määrittelen tässä luvussa ohjelmointirajapinnan sekä sen eri luokat. Määrittelyn jälkeen käyn läpi ohjelmointirajapinnan historiaa sekä kehitystä. En sukella syvälle teknisiin seikkoihin, vaan keskityn luvussa erityisesti liiketoiminnan, sosiaalisuuden ja merkittävien ohjelmointirajapintojen mahdollistamien teknologioiden näkökulmaan. Tämän takia nostan luvussa esiin havainnollistavia yritystarinoita, jotka ovat olleet merkittävässä roolissa ohjelmointirajapintojen historiassa. Viimeiseksi teen lyhyen katsauksen ohjelmointirajapintojen nykytilanteeseen.

### 2.1 Määrittely

Tutkielman tärkein käsite on ohjelmointirajapinta eli API (Application Programming Interface). API mahdollistaa eri sovelluksien tiedon, toiminnallisuuksien ja muiden resurssien jaon toisten sovellusten ja kolmansien osapuolien kanssa (Bahrami, Park, Li & Chen, 2018). API voi olla tietystä ohjelmointikielestä riippuvainen, tai se voi olla niin sanottu REST API, jolloin sitä voidaan käyttää verkon välityksellä standardisoidun HTTP-protokollan mukaisesti ohjelmointikielestä riippumatta (Wang, Keivanloo & Zou, 2014). API:a hyödyntävät esimerkiksi ohjelmointikirjastot, ohjelmointikehykset, ohjelmistokehityspaketit (SDK, Software development kit) sekä käyttöliittymäkirjastot. Voidaan todeta, että lähes jokainen koodirivi, joita useimmat ohjelmoijat kirjoittavat, käyttävät API-kutsuja, etenkin jos projektissa on käytössä sekä sisäisiä että avoimia API:a (Myers & Stylos, 2016.) Suosittujen palveluiden API:t, kuten Twitterin, Googlen, Facebookin, Netflixin ja AccuWeatherin API:t käsittelevät yli miljardi API-kutsua päivittäin jo vuonna 2012 (Evans & Rahul, 2016). On siis selvää, että tämän päivän liiketoiminnassa API on merkittävä tekijä ja niitä voidaan löytää kaikkialta sieltä mistä koodiakin.

API:t voidaan jakaa eri kategorioihin ja tutkimuksen kannalta yksi merkittävimmistä API-kategorioista on web-API. Web-API yhdistää itsenäisesti yllä-



pidettävät sovellukset ja web-API:n käyttö voi seurata erilaista protokollaa, kuten SOAP ja RESTful (Sohan, Anslow & Maurer, 2015). Yleisesti web-API:na voidaan kuitenkin pitää mitä tahansa API:a, joka kommunikoi verkon ylitse käyttäen HTTP-protokollaa (Tan, Fan, Ghoneim, Hossain & Dustdar, 2016). Tässä tutkimuksessa web-API:sta puhuttaessa tarkoitetaan juuri tätä yleistynyttä käsitettä, eli yleisesti verkon välityksellä kommunikoivaa API:a, ottamatta kantaa tarkemmin eri protokolleihin.

API-tarjoaja on organisaatio tai yksityishenkilö, joka tarjoaa ohjelmointirajapintoja muiden käyttöön. API-tarjoaja (API producer) on täten organisaatio, joka toteuttaa API:n toimeenpanon (de Souza & Redmiles, 2009). API-tarjoaja voi tarjota API:n sekä organisaation sisäiseen, että ulkoiseen käyttöön. API-käyttäjä (API consumer) taas on luonnollisesti organisaatio tai yksityishenkilö, joka käyttää API-tarjoajan tarjoamaa API:a eli pääosin tekee API:lle pyyntöjä, jotka API toteuttaa (de Souza & Redmiles, 2009). Käsittelen tutkielmassa myös termiä alustatalous, joka on verkotettua, digitaaliseen alustaan perustuvaa liiketoimintaa (Mckee, 2017). Lyhykäisyydessään alustataloudessa on kyse kokonaisuudesta, jossa eri liiketoimintojen osat ovat yhteydessä toisiinsa ja siten hyötyvät toisistaan. Yhtenä esimerkkinä voidaan pitää vaikkapa matkailuun liittyvää alustataloutta; lentoyhtiö toteuttaa lentämisen, taksiyrittäjä kuljetuksen hotelliin, joka tekee yhteistyötä paikallisen oppaan kanssa. Asiakas voisi esimerkiksi varata kaikki nämä palvelut yhdeltä nettisivulta. Näin kaikki osapuolet hyötyvät toisistaan vaikka ne ovat erillisiä yrityksiä, sillä yhtenäinen matka asiakkaalle alustatalouden ansiosta voi tuottaa parempaa asiakaskokemusta. API-talous on alustataloutta, joka on aikaansaatu ensisijaisesti API:en avulla.

API:t voidaan jakaa erilaisiin luokkiin niiden käyttötarkoitusten mukaan. Eri API:en luokkamäärittelyt eivät kuitenkaan ole riidattomia ja yksiselitteisiä, eikä niitä käytetä täysin yhtenäisin termein kirjallisuudessa. Erityisesti myös muuhun kuin organisaation sisäiseen käyttöön tarkoitettulla API:lla on monta termiä, näitä ovat muun muassa avoin API (open API), julkinen API (public API), julkistettu API (published API) ja ulkoinen API (external API). Selkeyden vuoksi jaan tässä tutkielmassa API:t neljään luokkaan, jotka ovat avoin API, julkinen API, sisäinen API ja kumppani-API. Alla olevassa taulukossa (taulukko 1) pyrin hahmottelemaan eri API-luokkien ominaisuuksia ja eroavaisuuksia.

TAULUKKO 1 API-luokittelu

API-luokka	Käyttäjät rajattu API-tarjoajan toimesta	API ja sen dokumentaatio vapaasti löydettävissä	API-käyttäjät
Avoin API	X	X	ulkoisia
Julkinen API		X	ulkoisia
Sisäinen API	X		sisäisiä
Kumppani-API	X		ulkoisia

Avoimelle API:lle on olemassa monta määrittelyä sekä tieteellisissä teksteissä, että internetin blogeissa ja kehittäjien suurissa suosioissa olevilla nettisivuilla. Erityisesti avointa- ja julkista API:a käytetään sekä synonyyminä, että erillisinä konsepteina. Käsittelen seuraavaksi tieteellistä kirjallisuutta näihin määrittelyihin liittyen ja pyrin tekemään niiden välille selkeän eron. Yksi määrittely on julkinen API (public API). Määrittelyssä API:n tekee julkiseksi se, että se on käytettävissä organisaation verkon ulkopuolelta ja API-tarjoaja tarjoaa avoimesti API-dokumentaation sekä toiminnalliset yksityiskohdat saadakseen kolmansien osapuolien kehittäjien kiinnostusta API:a kohtaan (Wulf & Blohm, 2020). Määrittelystä ei käy ilmi, onko API:n käyttö maksullista vai maksutonta. Toisen määrittelyn mukaan julkinen API (public API) on silloin, kun API-käyttäjät ovat sisäisiä, tai ainakin API-tarjoaja tietää keitä sen käyttäjät ovat (De Souza & Redmiles, 2009). Tutkimuksessaan he käyttävät public API termiä siten, että API on julkinen, jos API-tarjoaja ja -käyttäjä ovat samasta organisaatiosta. Julkisen API:n lisäksi heidän mukaansa API voi olla myös julkistettu API (published API). Heidän tutkimuksessaan julkistettu API eroaa julkisesta API:sta siten, että API on julkistettu API silloin kun sen käyttäjinä ovat muut kuin API-tarjoajan kehittäjät. Tutkijat (Benzell, Lagarda & Van Alstyne, 2017) luokittelevat API:t kahteen pääluokkaan, avoimiin (open) ja suljettuihin (closed). Mikäli API:in pääsy tapahtuu organisaation ulkopuolelta, on se silloin avoin API. Jotta API voi olla avoin, heidän mukaansa siihen täytyy sallia kolmansien osapuolien pääsy, jotka saavat samalla organisaation sisäiset resurssit vapaasti käyttöönsä. Pääsy voidaan antaa tietyille käyttäjille API-avaimien kautta, tai yleisesti kaikille ilman autentikointia. Heidän mukaansa idea avoimen API:n takana on rakentaa perustukset API-ekosysteemille. Zachariadis ja Ozcan (2016) käyttävät termiä ulkoinen API (external API). Ulkoista API:a voidaan käyttää heidän mukaansa organisaation liiketoiminnallisten resurssien julkaisuun ulkoisille kohderyhmille, näitä resursseja ovat esimerkiksi informaatio, palvelut ja tuotteet. Heidän ulkoisen API:n määrittelystä ei käy ilmi, ovatko resurssit kaikkien vapaassa käytössä, vai esimerkiksi maksumuurin takana. Ulkoisen API:n lisäksi he käyttävät tutkimuksessaan myös termiä julkinen tai avoin API (public or open API) joka on käytettävissä lähes kaikille, jotka hyväksyvät API-tarjoajan ehdot ja edellytykset sekä maksavat API:n käyttömaksun. Yu, Silveira ja Sundaram (2016) pitävät julkisen API:n (public API) määrittelyssä avaintekijöinä ketteryyttä, yksinkertaisuutta ja helppoa käyttöönottoa.

Kaikissa edellä mainituissa määrittelyissä yhteistä on, että avoimen API:n kautta voidaan käyttää organisaation määrittelemiä resursseja. Ristiriitaista kuitenkin on, ovatko näiden resurssien käyttö ilmaista vai maksullista ja se, onko API-käyttäjiä rajattu API-tarjoajan toimesta, vai saako sitä käyttää kuka tahansa. Väärinkäsitysten välttämiseksi määrittelen tässä tutkielmassa avoimen API:n siten, että API on avoin, jos API ja sen dokumentaatio ovat avoimesti löydettävissä, mutta sen käyttö on maksullista tai rajoitettua API-tarjoajan toimesta.

Kuten aiemmin tuli ilmi, avoimen ja julkisen API:n eroavaisuudet eivät ole selvät, eikä se, voidaanko niitä pitää toistensa synonyymeinä. Selkeyden vuoksi erittelen tutkielmassani julkisen API:n avoimesta API:sta. Julkisella API:lla tar-

koitetaan tutkielmassani julkisesti saatavalla olevaa API:a, joka tarjoaa organisaation sisäisiä resursseja ilmaiseksi kenelle tahansa. Vaikka kuka tahansa voi käyttää julkista API:a maksutta, käytön edellytyksenä voi silti olla käyttäjien autentikointi esimerkiksi API-avaimen avulla. Esimerkkinä julkisesta API:sta voidaan pitää Digi- ja väestötietoviraston tarjoamaa avoindata.fi -verkkosivustoa. Verkkosivujensa mukaan sieltä on löydettävissä kaikki Suomen avoin data, verkkosivusto tarjoaa noin 1800 erilaista tietoaineistoa, joita kuka tahansa voi vapaasti käyttää. Sivustolta löytyy esimerkiksi rajapinta Helsingin pysäköintipaikkojen käytöstä, jonka API:n avulla käyttäjä saa lukumäärän kyseisellä alueella sillä hetkellä käynnissä olevien pysäköintien määrästä (Avoindata, 2020). Tätä rajapintaa hyödyntäen kuka tahansa voisi tehdä vaikkapa verkkosivuston, jossa dataa visualisoidaan kartalla ja olisi näin hyödyllinen suurelle yleisölle, ehkäpä jopa liiketoiminnallisessa mielessä. Julkinen API on siis julkisesti saatavilla oleva API, jonka dokumentaatio ja käyttöohjeet ovat julkisesti kenen tahansa löydettävissä, ja jota kuka tahansa voi käyttää ilman maksua.

Zachariadis ja Ozcan (2016) käyttävät termiä yksityinen API (private API). Heidän mukaansa yksityinen API voi olla joko sisäinen API (internal API) organisaation sisäiseen käyttöön, tai ulkoinen API (external API), jotka ovat erittäin kustomoituja ja suunniteltu sekä toteutettu tietyille organisaation ulkoisille yhteistyökumppaneille. Yksityiset API:t ovat heidän mukaansa tarkasti rajattu vain tiettyjen tunnettujen kehittäjien käyttöön. Sisäinen API (internal API) on heidän mukaansa taas API, jota käytetään yleensä organisaation sisäisten suurien järjestelmien integroimiseen ja/tai niiden väliseen datan ja toiminnallisuuksien vaihdantaan. Tutkijoiden mukaan sisäinen API voi parantaa organisaation eri osastojen yhteistyötä ja datan vaihdantaa, yhdistää palveluita ja liiketoimintoja, edistää työntekijöiden tehokkuutta ja jopa auttaa parempaan asiakaskokemukseen. Parempi asiakaskokemus saadaan aikaan esimerkiksi siten, että tarvittaessa koko organisaatiolla asiakaspalvelusta myyntiosastoon on saatavilla asiakkaan tiedot, joita voidaan käyttää asiakaskohtaamisessa. Usein sisäisetkin API:t noudattavat web-APIen rakenteita ja käytänteitä, jotta niitä käyttävät kehittäjät voivat hyötyä enemmän yleisestä tiedosta tiettyjen järjestelmien käytäntöjen sijaan. Benzell ja kumppanit (2017) käyttävät termiä suljettu API (closed API). Heidän määritelmänsä mukaan suljettu API on saatavilla vain organisaation sisäisille toimijoille, tai läheisille yhteistyökumppaneille. Suljettuun API:in pääsy tapahtuu heidän mukaansa API-tarjoajan valtuuttamana. Yu, Silveira ja Sundaram (2016) käyttävät termiä yritys-API (enterprise API). He eivät määrittele yritys-API:a tarkemmin, mutta sen voidaan olettaa tarkoittavan vastaavaa, suljettua ja tarkalle joukolle suunnattua API:a. Yhteistä kaikille määritelmille on se, että API-käyttäjät ovat tarkkaan rajattuja. Ristiriitaista määritelmässä on se, ovatko API-käyttäjät ainoastaan organisaation sisäisiä, vai myös toisen organisaation kehittäjiä. Selkeyden vuoksi käytän tässä tutkielmassa sisäisen API:n määritelmää siten, että API on sisäinen, jos se on vain organisaation sisäisessä käytössä.

Yu ja kumppanit käyttävät tutkimuksessaan myös kumppani-API (partner API) termiä, heidän mukaansa kumppani-API on API, jonka avulla hoidetaan sisäiset ja ulkoiset integraatiot. Zachariadis ja Ozcan (2016) luokittelevat määrittelyyn sopivat API:t kumppani-API:ksi (partner API). Kumppani-API:n määrittelyyn kuuluvat heidän mukaansa API:t, jotka mahdollistavat kommunikoinnin ja/tai integraation kahden organisaation välillä. Heidän mukaansa kumppani-API saa usein alkunsa sisäisestä API:sta, joka julkistetaan toisen organisaation käyttöön yhteistyön ja innovoinnin mahdollistamiseksi. Tunnettuna liiketoiminnan esimerkkinä tällaisesta API:sta voidaan pitää Netflixiä, sen API:n avulla esimerkiksi TV-valmistajat, pelikonsolit ja erilaiset kannettavat laitteet voivat yhdistää Netflixin suoratoiston järjestelmäänsä. Tämä on auttanut Netflixiä laajentumaan monille alustoille suhteellisen edullisesti, mutta kuitenkin pitämään tarjontansa omassa hallinnassaan (Fenger, Henriksen, Hedman & Xiao, 2018). Erittelen tutkielmassani kumppani-API:n sisäisestä API:sta, koska mielestäni API ei ole enää sisäinen, jos se on toisen organisaation käytössä, vaikka API-käyttäjät olisivat tarkkaan rajattuja. Tutkielmassani kumppani-API on API, jonka avulla tehdään organisaatioiden väliset integraatiot, jotka mahdollistavat tietojen tai toiminnallisuuksien siirron organisaatioiden välillä.

## 2.2 Kehitys

Tässä luvussa käyn läpi API:n historiaa, sekä sen kehitystä. En sukella syvälle teknisiin sovelluskehyksiin, ohjelmistokehityspaketteihin tai käyttöliittymäkirjastoihin, vaan keskityn luvussa erityisesti liiketoiminnan, sosiaalisuuden ja merkittävien API:en mahdollistamien teknologioiden näkökulmaan. Tämän takia nostan luvussa esiin havainnollistavia yritystarinoita, jotka ovat olleet merkittävässä roolissa API:n historiassa. Viimeiseksi teen lyhyen katsauksen API:en nykytilanteeseen.

API:n merkittävän kaupallisen käytön voidaan katsoa alkaneen noin vuonna 2000. Ensimmäiset alan pioneerit määrittelivät API:t myynnin ja kaupankäynnin hallinnan tueksi tehostamaan päivittäisiä liiketoimintoja. Yleisinä suunnannäyttäjinä pidetään Salesforcea ja eBayta, ne julkaisivat API:nsa yleiseen käyttöön vuonna 2000 (Kopecký, Fremantle & Boakes, 2014). Salesforcea pidetään ensimmäisenä organisaationa, joka ymmärsi heidän asiakkaidensa tarvitsevan menetelmän, jolla he voivat jakaa dataa eri liiketoimien ja sidosryhmien kesken. API oli ratkaisu tähän ongelmaan ja se julkaistiin helmikuussa vuonna 2000 (Kopecký, Fremantle & Boakes, 2014.). Salesforce oli samalla myös ensimmäinen pilvipalvelua tarjoaja toimija, joka toteutti yritystason kokoisen verkkosovelluksen ja -API:n, sekä toimitti kenties ensimmäisen SaaS (Software-as-a-Service, ohjelmisto palveluna) -palvelun (Lane, 2016). Salesforce on siis ollut jo 2000-luvun alussa merkittävä tekijä sekä suunnannäyttäjä ja on sitä vielä tänäkin päivänä. SalesForcen julkaiseman ensimmäisen API:n voidaan nähdä kuuluvan kumppani-API:en luokkaan. Pian SalesForcen jälkeen eBay lanseerasi vuoden 2000 marraskuussa eBay API:n. API lanseerattiin, jotta Ebay pystyisi

standardisoimaan tavan, jolla monet sovellukset olivat yhteydessä heidän verkkosivustoonsa, joko laillisesti tai laittomasti. Ebayn julkaiseman API:n voidaan katsoa olevan julkinen API. Standardisoinnin lisäksi API:n tarkoitus oli myös helpottaa kumppaneiden ja kehittäjien mahdollisuuksia rakentaa omaa yritystä eBayn ekosysteemin ympärille (Kopecký, Fremantle & Boakes, 2014) ja siten kasvattaa myös eBayn markkina-asemaa. Ebayta pidetään johtavana edelläkävijänä web-API:en osalta ja se on nykyäänkin yksi menestyneimmistä kehittäjien ekosysteemeistä (Lane, 2016).

Parin vuoden jälkeen seuraava merkittävä tekijä nähtiin API:en suhteen, kun Amazon julkaisi heinäkuussa 2002 AWS:n (Amazon.com Web Services) (Amazon, 2002). AWS mahdollisti kehittäjien sisällyttää Amazon.com sisältöä ja toiminnallisuuksia omiin verkkosivustoihin, joka luonnollisesti hyödytti molempia osapuolia. AWS mahdollisti esimerkiksi kolmansien osapuolien etsiä ja esitellä Amazon.com tuotteita omilla verkkosivuillaan. Heti ensimmäisestä AWS:n julkaisupäivästä lähtien Amazonilla oli käytössä yhteistyökumppanuusmahdollisuus, joka toi kehittäjille osan tuotoista, jos asiakas meni heidän linkkejään pitkin Amazoniin (Lane, 2016.), AWS oli siis alkuaikoinaan kumppani-API. Nykypäivänä AWS on paljon laajempi kokonaisuus ja markkinoiden isoimpia tekijöitä erityisesti pilvilaskennan osalta. Amazonin perustaja Jeff Bezos toteutti vuonna 2002 ”API-mandaatin”, kenties kaikkien aikojen näkyvimmän tempauksen yrittää siirtää API liiketoiminnan keskiöön. Mandaattiin kuului muun muassa käsky datan ja toiminnallisuuksien siirtäminen tiimistä toiseen vain ja ainoastaan API:en kautta, tottelemattomat saavat potkut (Benzell, Lagarda & Van Alstyne 2017.). Mandaatin voidaan nähdä olleen varsin tehokas, sillä Tan kumppaneineen (2016) kertoo AWS:n liikevaihdon tulevan sata prosenttisesti API:en avulla.

Liiketoimintojen lisäksi API:en hyödyntäminen sisällöntuotannon, web-linkkien, kuvien ja muiden medioiden julkaisussa alkoi vuosina 2003–2006. API:n hyödyntäminen sosiaalisissa palveluissa on ollut merkittävä katalyytti niiden räjähdysmäiseen kasvuun. Jousha Schachter perusti vuonna 2003 verkkosivuston nimeltä del.icio.us. Sillä oli käytössään hyvin yksinkertainen tunnistejärjestelmä, jonka avulla käyttäjät kykenivät merkitsemään mihin aiheeseen mikäkin nettisivu kuuluu. Täten käyttäjät pystyivät hakemaan tunnisteiden avulla myös muiden käyttäjien tunnistamia nettisivuja, käyttäen del.icio.us API:en nettisivuilla. Esimerkiksi lentokoneita hakiessa riitti, kun haki osoitetta <http://del.icio.us/tag/airplane> ja sivu palautti kaikki airplane-tunnisteella merkityt verkkosivut. Merkittävää verkkosivulla oli sen saumaton käyttökokemus, se oli ensimmäinen konkreettinen esimerkki, kuinka verkko voi toimittaa HTML-, RSS- ja XML-sisältöä ihmiselle helposti ymmärrettävällä URL-rakenteen avulla (Lane, 2016.). Koska del.icio.us haut rakennettiin ihmiselle helposti ymmärrettävään URL muotoon, se auttoi kehittäjiä ja käyttäjiä omaksumaan API:n käytön helposti ja näyttämään suuntaa, kuinka API:t voidaan rakentaa käyttäjäystävälliseksi.

Selainten kehittyessä tukemaan JavaScriptiä ja samanaikaisesti Ajaxin yleistyessä lopputulos johti siihen, että selainten oli käytännöllistä ladata vain

osa sivun sisällöstä kerrallaan sen sijaan, että koko verkkosivusto päivitetäisiin (Murugesan, 2007). Tästä suuresta muutoksesta käytetään usein nimitystä Web 2.0 (Kopecký, Fremantle & Boakes, 2014). Web 2.0 voidaan katsoa syntyneen noin vuosina 2004–2005, vaikka vielä tänäkään päivänä ei ole täysin riidatonta, mitä Web 2.0 tarkalleen ottaen on, ehkäpä sen aikaansaamasta valtavasta muutoksesta johtuen (Murugesan, 2007). Web 2.0-sivustojen käyttämä JavaScript tarvitsee toimiakseen API yhteyden serveriin ja kun se on kunnossa, API:sta voidaan tehdä suhteellisen pienellä vaivalla julkinen muiden sivujen ja sovelluksien käyttöön (Kopecký, Fremantle & Boakes, 2014). Tämä on edesauttanut ja kiihdyttänyt kyseisten API:en käyttöä muilla verkkosivuilla ja -ohjelmissa, kasvattaen yleisesti API:en määrää valtavasti.

Neljä kuukautta Google Maps -sovelluksen julkaisun (Google, 2005a) jälkeen valtavan kysynnän vuoksi Google julkaisi julkisen Google Maps API:n kesäkuussa 2005 (Google, 2005b). Google Maps API:n julkaisun taustalla oli jälleen kerran muiden suosittujen palveluiden tavoin pyrkimys vastaamaan valvomattomiin tekijöihin, jotka väärinkäyttivät sovellusta. Google Maps API muotoutui lopulta merkittäväksi suunnannäyttäjäksi, sen voidaan katsoa aloittaneen API-mashupien trendin. Mashup on sovellus, joka yhdistää informaatiota ja/tai toiminnallisuuksia useista eri lähteistä, saaden aikaan uuden palvelun, mikä luonnollisesti luo lisäarvoa sen käyttäjille kahden erillisen lähteen sijaan. Mashupit ovat yleensä rakennettu API:en avulla (Murugesan, 2007.). Syyskuussa 2006 Twitter esitteli Twitter-API:n maailmalle (Twitter, 2006). Tämän julkisen API:n julkaisun syyt olivat hyvin pitkälti samat kuin eBay API:ssa ja Google Maps API:ssa; API julkaistiin vastauksena Twitteriin kohdistuneeseen tiedonharavointiin (scraping) ja siihen tehtyihin epävirallisiin sekä valvomattomiin (rogue) API:in. Twitter-API on yksi tärkeimmistä API-ekosysteemin edelläkävijöistä, näyttäen miten suureksi ekosysteemi voi kasvaa avaamalla ulkoisille kehittäjille pääsy API:in ja antaen julkisen API-ekosysteemin rakentaa itse itsensä.

Maaliskuussa 2006 julkistettu Amazon S3 (Amazon Simple Storage Service) oli varasto internetissä, joka jo alle vuoden käyttöönoton jälkeen sisälsi yli kaksi biljoonaa objektia ja käsitteli säännöllisesti reilu miljoona API-kutsua joka sekunti (Newcombe, Rath, Zhang, Munteanu, Brooker & Deardeuff, 2015). Amazon S3 oli aluksi pelkkä avoin API ilman graafista käyttöliittymää, jota voitiin käyttää datan tallentamiseen ja noutamiseen, milloin ja missä tahansa. Amazon Simple Storage Service pyrki tarjoamaan tallennustilaa alhaisin kustannuksin ja laajasti saavutettavalla käytöstä maksettavalla (pay-as-you-go) -veloitusmallilla (Palankar, Iamnitchi, Ripeanu & Garfinkel, 2008). Tämä antoi kaikille kehittäjille pääsyn samaan datan tallennusinfrastruktuuriin, jota Amazon itse käytti oman maailmanlaajuisen verkkosivustojensa ylläpitoon (Amazon, 2006a). Samalla myös API:en käyttötarkoitus laajeni pelkästä datan tai yksinkertaisen toiminnallisuuden toimittamisen lisäksi kokonaisen laskentainfrastruktuurin toimitamiseen. Saman vuoden elokuussa Amazonilta nähtiin toinen julkaisu, Amazon EC2 (Amazon Elastic Compute Cloud) (Amazon, 2006b). Amazon Elastic Compute Cloud on verkkopalvelu, joka tarjoaa skaalautuvaa pilvilaskentaa verkossa. Käytännössä tämä tarkoittaa eri organisaatioiden mahdollisuutta

vuokrata tietokoneita Amazon EC2 -palvelin keskuksista, suorittaakseen erilaisia ohjelmia omiin käyttötarkoituksiinsa (Wang & Ng, 2010). Amazon EC2 ei alkuaikoinaan tarjonnut graafista käyttöliittymää, vaan sitä käytettiin API:n avulla. Näin Amazon aloitti API:n avulla uudenlaisen tietojenkäsittelymallin, joka tunnetaan tänä päivänä pilvilaskentana. Tarvittaessa saatavilla oleva, vain käytöstä maksettava pilvilaskenta on ollut todella suosittua ja merkittävää nyky-yhteiskunnassa, erityisesti kaupallisten verkkosovellusten keskuudessa sen joustavan ja kustannustehokkaan mallin takia (Jackson, Ramakrishnan, Muriki, Canon, Cholia, Shalf, Wasserman & Wright, 2010).

2010 perustettu Stripe lanseerattiin julkiselle yleisölle syyskuussa 2011. Stripe oli rakennettu ensisijaisesti kehittäjille ja se onnistui erinomaisesti API-suunnittelussa, -dokumentoinnissa, -tuessa ja hinnoittelussa liittyen maksujen integrointiin, liiketoimintaan ja asiakasratkaisuihin (Rao, 2011). Stripeä voidaan pitää yhtenä erinomaisena esimerkkinä sen suhteen, kuinka erityisesti maksu-API:t, mutta myös API:t yleensä voidaan tehdä kehittäjäystävällisesti huomioiden suunnittelu ja dokumentointi.

API:n alkuaikoina 2000-luvun alussa rakennettiin ensin verkkosivu tai muu käyttöliittymä, sen jälkeen myöhemmin siihen sopiva API. Nyt työjärjestys on muuttunut, ensiksi rakennetaan usein kehittäjäystävällinen API, jonka jälkeen vasta verkkosivu tai muu käyttöliittymä sen päälle käyttäjäksi. Tämä API-ensin -malli arvostaa ja painottaa erinomaisen API:n kehittämistä. Erinomaisen API:n kehitys on epäilemättä työlästä, mutta se todennäköisesti palkitaan myöhemmin, etenkin koska API:n ensimmäiset asiakkaat ovat usein samasta organisaatiosta (Kopecký, Fremantle & Boakes, 2014.). Viime vuosien kasvavana trendinä ovat olleet hallitut web-API:t. Hallitut web-API:t vaativat yhteyden salliakseen API-avaimen, joka täytyy olla jokaisen pyynnön mukana. Avaimia käytetään usein käyttäjän autentikointiin, jotta API:n käyttö voidaan esimerkiksi kaupallistaa, mutta toisaalta myös väärinkäytösten ehkäisemiseksi, ennen kaikkea API:in pääsyn kontrolloimiseksi sekä rajoittamiseksi (Farrell, 2009). On tutkittu, että jo vuonna 2012 oli ainakin kymmenen API-tarjoajaa, jotka hallitsivat yli miljardi API kutsua päivässä (Kopecký, Fremantle & Boakes, 2014.), joten käyttäjänhallinta on välttämätöntä valtavan volyymin vuoksi. Tänä päivänä tämän kokoluokan API-tarjoajia on todennäköisesti huomattavasti enemmän ja API-hallinnalle siten entistä enemmän kysyntää.

Yhtenä selittäväenä tekijänä tämän päivän API:n valtavalle suosiolle voidaan pitää sen toimintaperiaatetta ja modulaarista rakennetta. Valmis API itsessään on sitä käyttävälle aina jossain määrin julkinen moduuli, mutta varsinaisen API:n toteutus tapahtuu eri moduulissa, joka taas on yksityinen. Näin yksityistä moduulia voidaan kehittää ja muuttaa, vaikuttamatta julkiseen moduuliin toimintaan. API-tarjoaja voi tarjota omia toiminnallisuuksiaan tuhansille kehittäjille, joiden ei tarvitse tietää kuinka API itsessään on ylipäättänsä toteutettu, vain sillä on merkitystä, mitä kutsuja API:lle voidaan tehdä ja mitä niiden vastauksena saadaan. Täten API:a voidaan rauhassa kehittää vaikuttamatta niitä käytäviin kehittäjiin tai organisaatioihin (De Souza & Redmiles, 2009), minkä seurauksena API:sta voi kehittyä yhä kiinnostavampi kohde, joka taas lisää uu-

sia käyttäjiä. API:n suuret käyttäjämäärät lisäävät kiinnostusta ja kehitystä edelleen, joten hyvin toteutettu, merkityksellinen ja käyttäjäystävällinen API löytää kyllä yleisönsä. API:en kasvava määrä ruokkii niiden omaa kasvua. Erilaisia API:a yhdistelemällä saadaan uusia API-mashupeja ja palveluita, joita yhdistelemällä syntyy taas uusia API-mashupeja ja palveluita (Kopecký, Fremantle & Boakes, 2014). Internetin kaikista kattavin julkinen API-hakemisto programmableweb.com (Wulf & Blohm, 2020) pitää kirjaa kirjoitushetkellä yli 24-tuhannesta API:sta. Määrä on kasvanut valtavasti, sillä 2005 perustettu sivusto sisälsi vielä 2008 ainoastaan noin tuhat API:a. Trendi vaikuttaa edelleen nousevalta ja API:t tulevat todennäköisesti olemaan tulevaisuudessa kasvavissa määrin yhä merkittävämmässä osassa kaikilla aloilla.



### 3 OHJELMOINTIRAJAPINTOJEN VAIKUTUKSET

Tässä luvussa käsittelen API:en mahdollistamia hyötyjä eri organisaatioille ja pyrin siten vastaamaan tutkimusongelmaan. Rajaan hyödyt kolmeen osioon, jotka ovat tehokkuutta kasvattavat hyödyt, markkinoinnin hyödyt, sekä datan kaupallistaminen. Lopuksi käsittelen, miten nämä hyödyt voivat vaikuttaa organisaatioihin taloudellisesti.

#### 3.1 Hyödyt

API tarjoaa monia tapoja kasvattaa organisaation sisäistä tehokkuutta. De Souza ja Redmilesin tutkimuksessa (2009) nousi esiin kolme merkittävintä hyötyä, jotka ovat heidän mukaansa yleisesti tunnistettuja ammatinharjoittajien ja tutkijoiden keskuudessa. Tutkijoiden mukaan nämä hyödyt ovat saavutettavissa API:n kolmiosaisen roolin ansiosta, jotka ovat API sopimuksena (APIs as Contracts), API rajoina (APIs as Boundaries), sekä API kommunikaatioväylänä (APIs as Communication mechanisms).

API-sopimuksella tutkijat tarkoittavat sopimusta API-tarjoajan ja API-käyttäjän välillä. Sopimus osapuolien välillä mahdollistaa organisaatiossa eri sidosryhmien rinnakkaisen ja itsenäisen työn. Sopimuksen lisäksi API voidaan nähdä rajoina kehittäjien välillä. Rajat auttavat tutkijoiden mukaan siinä, ettei kehittäjien tarvitse eivätkä ne edes pääse tietämään, mitä työtä kollega tekee. Näin kehittäjät voivat keskittyä omaan erikoisalaansa, eikä resursseja mene hukkaan työntekijöiden tehdessä päällekkäistä työtä. Rajat samalla myös minimoivat kehittäjien vaikutusta toisiinsa, toisen kehittäjän työ ei vaikuta merkittävästi toiseen API:n modulaarisen rakenteen ansiosta. Näitä rooleja tukevat myös Lindmanin, Horkoffin, Hammoudan ja Knaussin (2018) näkemykset. Heidän mukaansa API tarjoaa käytännöllisen käyttöliittymän palvelun tarjoamiseen (eli edesauttavat sopimuksien noudattamista), käyttöoikeuksien hallintaan ja kolmansien osapuolien kehitykseen, jotka asettavat selkeät rajat API:n käytön suhteen. API:t voidaan nähdä myös tutkijoiden näkemyksen mukaan

tehokkaana viestintäkanavana kehittäjille. API:t fokusoivat intuitiivisesti sitä, mistä tarkalleen ottaen tulisi keskustella. De Souzan ja Redmilesin tutkimuksessa esimiehet pystyivät myös rajaamaan kommunikaation API:n suhteen kahden osapuoleen, yksi API-käyttäjä kommunikoi yhden API-tarjoajan kanssa, sen sijaan että tiimien välillä useammat kehittäjät kommunikoisivat keskenään. Nämä kolme roolia täyttämällä API auttaa kehittäjiä työskentelemään rinnakkaisesti ja eristyksissä, keskittyen vain oleelliseen osaan kollegoiden töistä, kommunikoiden tehokkaasti.

Tehokkuuden lisäksi API:en avulla voidaan saavuttaa markkinoilla kilpailuetua neljällä avaintekijällä, jotka ovat tiedon kerääminen, kerätyn tiedon hyödyntäminen, verkottuminen ja ekosysteemi (Bala & Subramaniam, 2015). API mahdollistaa valtavan tiedonkeruun asiakkaan käyttäytymisestä, joka voidaan nähdä jokaisen päivittäisessä elämässä; monet verkkosivut ilmoittavat niissä käytettävistä evästeistä, jotka keräävät tietoa muun muassa siitä mitä vierailija verkkosivulla tekee. Kerättyä tietoa hyödyntämällä voidaan ennustaa kuluttajan käyttäytymistä, jota käytetäänkin runsaasti, kohdennettu mainonta on arkipäivää käytännössä kaikille verkon käyttäjille. Jo vuonna 2014 Facebook tienasi 3.85 miljardia dollaria mainonnan avulla ainoastaan vuoden viimeisellä neljänneksellä (Bataineh, Mizouni, El Barachi & Bentahar, 2016). Farahatin ja Baileyn (2012) tutkimuksen mukaan kohdennetun mainonnan avulla yksi klikkaus mainokselle maksaa 4.5 kertaa vähemmän verrattuna kohdentamattomaan mainontaan, joten valtavia tietomääriä hyödyntämällä voidaan saavuttaa merkittäviä hyötyjä. Verkottumisen hyödyt saavutetaan tutkijoiden mukaan panostamalla API:n käyttöliittymään sekä dokumentaatioon. Intuitiivinen, helppokäyttöinen käyttöliittymä ja selkeästi dokumentoitu API auttaa organisaatiota lisäämään ulkoisia API-käyttäjiä ja siten myös dataa, jota organisaatio voi jälleen hyödyntää. Balan ja Subramaniamin mukaan rakentamalla ekosysteemi API:n ja sen keräämän datan avulla organisaatio voi laajentaa liiketoimintaansa jopa täysin uusille aloille, joiden tuottamat hyödyt voivat muuttaa koko organisaation liiketoimintamallia. Nämä tutkijoiden tunnistamaa neljä avaintekijää siis ruokkivat toinen toisiaan, mutta toisaalta ovat myös toisistaan riippuvaisia. Ekosysteemin synty tuskin on mahdollista ilman kerätyn datan hyödyntämistä, eikä dataa taas saada kerättyä, jos API:lla ei ole yhtäkään sisäistä tai ulkoista käyttäjää.

Balan ja Subramaniamin tunnistamia kilpailutekijöitä, erityisesti tiedon keräämistä voidaan edelleen jalostaa sekä tuotteistaa dataa kaupallistaessa. Kehittyneet datan analysointityökalut ja pilvilaskenta ovat saaneet aikaan sen, että datan kaupallistaminen ei ole enää vaikeasti saavutettava asia ja sen avulla saavutetaan sekä suoria että välillisiä hyötyjä (Najjar & Kettinger, 2013). Wixom ja Ross (2017) esittävät tutkimuksessaan kolme erilaista tapaa kaupallistaa dataa; parantamalla sisäisiä liiketoiminnan prosesseja ja päätöksiä datan avulla, yhdistämällä data ydinpalveluun, tai myymällä data jo olemassa oleville ja uusille markkinoille. Teoriassa organisaatiot voivat pyrkiä useampaan tapaan kaupallistaa dataa, mutta Wixomin ja Rossin mukaan käytännössä on paras ainakin

aloittaa tunnistamalla omalle organisaatiolle yksi paras vaihtoehto ja pyrkiä jalkauttamaan se.

Välittömin tapa kaupallistaa dataa tutkijoiden mukaan on hyödyntää sitä päätöksenteossa. Organisaatiot saavat tästä erityisesti hyötyä, kun datan hyödyntäminen osoitetaan henkilöille, jotka ovat keskiössä päätöksenteossa, kuten asiakasrajapinnassa toimivat, tuotteen kehityksestä vastaavat tai tuotannon johtajat. Microsoft alkoi parantaa datan hyödyntämistään 2014. Vuodessa he rakensivat uuden asiakastietojärjestelmän, joka sisälsi keskitetyksi kaikki oleelliset tiedot heidän asiakkaistaan. Tutkimuksen mukaan tämä uusi järjestelmä säästi myyntihenkilöstöltä kymmenestä viiteentoista minuuttiin per myyntimahdollisuus. Microsoftin liikevaihdon ollessa 93.58 miljardia dollaria vuonna 2015 (Microsoft, 2015) näitä myyntimahdollisuuksia voidaan olettaa olleen varsin runsaasti, joten uusi järjestelmä ja siten datan hyödyntäminen päätöksenteossa todennäköisesti vähensi työtunteja merkittävästi.

Ydinpalvelun laajentaminen datan avulla voi tapahtua esimerkiksi yhdistämällä siihen raportointia, hälytyksiä tai muuta informaatiota luodakseen sille lisäarvoa. Wixomin ja Rossin mukaan yhdistetty data ja analytiikka tulisi kuitenkin olla samalla tasolla laadun suhteen ydinpalvelun kanssa, jolloin ne ovat ikään kuin lisäosia ydintuotteelle. Esimerkkinä he kertovat Capital One Financial Corp. -pankin tekemästä laajennuksesta luottokorttitapahtumiin. Pankki lisäsi tiliotteeseen jokaiseen tapahtumaan kauppiaan logon, jotta asiakas tiliotetaan tarkastaessa pystyi helpommin tarkastamaan tapahtumat väärinkäytösten varalta. Lopputuloksena asiakkaat olivat tyytyväisempiä palveluun ja siten käyttivät luottokorttejansa enemmän. Vastaavia pienehköjä, mutta merkittäviä laajennuksia nähdään tänä päivänä jatkuvasti eri aloilla, joten niiden arvonlisäykset kokonaisuudessaan ovat todennäköisesti valtavia.

Vaikka intuitiivisesti datan kaupallistaminen tarkoittaisi sen myymistä, tutkijat pitävät sitä kuitenkin kaikista hankalimpana tapana kaupallistaa dataa. Tämä johtuu heidän mukaansa siitä, että datan myyminen vaatii uniikin liiketoimintamallin, jota useimmat organisaatiot eivät pysty toteuttamaan. Kaikki kolme tapaa kaupallistaa dataa vaativat Wixomin ja Rossin tutkimuksen mukaan vahvaa johtajuutta, selvän strategian, investointeja ja sitoutumista. Isoimmiksi ongelmiksi datan kaupallistamiseksi he tunnistavat ensinnäkin datan saatavuuden, heidän tutkimuksessaan vain neljännes organisaatioista tarjosi työntekijöilleen ja asiakkailleen helpon pääsyn tarvittavaan dataan. Saatavuuden lisäksi toinen ongelma on myös johtajuuden puute, datan kaupallistamiseen tarvitaan vahvaa muutosjohtajaa, joka saa työntekijät näkemään tärkeän arvon uudessa muutoksessa.

### **3.2 Hyötyjen taloudelliset vaikutukset**

Tutkimuksia, joista selviäisi konkreettisia taloudellisia tunnuslukuja API:n tuottamista taloudellisista hyödyistä liiketoiminnalle, vaikuttaisi olevan varsin niukasti. Uskoisin tämän johtuvan API:n monista eri hyödyntämistavoista erilais-

sa organisaatioissa, mutta myös sen epäsuorista hyödyistä. Tarkastelen seuraavaksi tutkimuksia, jotka pyrkivät selvittämään, kuinka API-käyttöönotto vaikuttaa yrityksen suorituskykyyn, sekä kuinka organisaatioissa voidaan tuottaa arvoa API:n avulla taloudellisilla tunnusluvulla mitattuina.

Benzell, Lagarda ja Van Alstyne tekivät vuonna 2017 tutkimuksen, jonka kohteena olevat yritykset olivat suuria, keskimäärin markkina-arvoltaan 2.3 miljardin dollarin kokoisia vuonna 2013. Tutkimuksen tarkoituksena oli selvittää, miten API-käyttöönotto vaikuttaa yrityksen suorituskykyyn. Yritykset jakautuivat skaalaan, jossa pienimmän yrityksen markkina-arvo oli 10 miljoonaa dollaria ja suurimman yli 180 miljardia dollaria. Tutkimus toteutettiin analysoimalla yrityksiä taloudellista kehitystä sekä API-käyttöä kuvaavaa dataa vuosilta 2013–2016. Yrityksiä oli tutkimuksen alussa 58 ja määrä kasvoi 239 yritykseen vuoteen 2016 mennessä API-käyttöönottojen yleistyessä. Yrityksiä oli tutkimuksessa jälleenmyynnistä, IT:stä, valmistusteollisuudesta ja muista määrittelemättömistä teollisuuden aloista. Tutkimuksen yrityksillä oli usein API, joka taipui moniin eri sisäisiin ja ulkoisiin tarpeisiin, tai useampi erillinen API eri tarkoituksiin. Yhden API:n yhtä tarkoitusta varten omistavia yrityksiä oli vähäisesti.

Tutkimuksen tekijöille yllättävänä havaintona ilmeni se, että yritykset saivat mitattavia taloudellisia hyötyjä API-käyttöönotolla eniten kustannuksia alentamalla myynnin kasvattamisen sijaan, eli sisäisten API:en avulla. Tärkeänä ja ehkä hieman myös itsestään selvänä havaintona voidaan pitää ilmennyttä seikkaa siitä, että mitä enemmän yrityksessä tehdään API-kutsuja, joko sisäisiä tai ulkoisia, sitä paremmin se korreloi positiivisiin taloudellisiin muutoksiin. Pelkkä API-käyttöönotto ei siis vielä itsessään tuota taloudellisia hyötyjä, sitä pitää myös hyödyntää paljon.

Tutkijat analysoivat tutkimustuloksissaan vahvimman korrelaation nettotulojen kasvun ja B2B-API kutsujen välillä. Kumppani-API:n mahdollistamien integraatioiden lisäksi tämän voidaan olettaa liittyen myös muuhun organisaatioiden väliseen toimintaan, kuten maksullisen avoimen API:n hyödyntämiseen. Sisäisten API-kutsujen runsas määrä taas korreloi suuresti yrityksiä kustannuksien karsimisessa, joka oli tutkimuksen tuloksien osalta merkittävin markkina-arvon kasvattaja. Sisäiset API-kutsut auttavat epäilemättä tehokkuuden kasvattamisessa muun muassa aiemmin mainitun rinnakkaisen ja itsenäisen työn mahdollistuessa. Yrityksiä tulojen nähtiin kasvavan myös julkisten API:en avulla, vaikkakaan ei niin merkittävin osin kuin sisäisten-, avoimien- ja kumppani-kutsujen tapauksissa. Tutkimuksen mukaan eri API:en tai eri API-kutsujen liikennettä seuraamalla voidaan siis tehdä arvioita, miten eri taloudelliset tunnusluvut kehittyvät. Yllättävänä tekijänä tutkijat havaitsivat sen, että sisäiset API:t kasvattivat myyntiä vahvemmin kuin avoimet API:t kustannuksien alentamisen lisäksi. Asiakastiedon parempi saavutettavuus yrityksen sisäisesti ja aiemmin puhuttu yhtenäisempi asiakaskokemus sisäisen API:n ansiosta voisivat olla tekijöitä, jotka ovat siihen auttaneet. Tutkijat tekivät selvän havainnon myös API:n käytön intensiteetin ja yrityksen taloudellisten tulosten

vahvasta positiivisesta yhteydestä, mikäli API-kutsujen osalta liikennettä ei ole, ei taloudellisia hyötyjäkään saavuteta.

Benzell ja kumppanit arvioivat tutkimuksessaan API:n käyttöönoton kasvattavan yrityksen markkina-arvoa keskimäärin 12.7 prosenttia, kaikkein varovaisimmissakin arvioinneissa markkina-arvon kasvu on keskimäärin 2.23 prosenttia. Tutkimuksen päätulokseksi tutkijat siis arvioivat API:n käyttöönoton olevan merkittävästi yhteydessä organisaation nettotuloksen, markkina-arvon ja liikevoiton kasvuun. API:n käyttöönoton ei kuitenkaan nähty vaikuttavan merkittävästi nettotulokseen regressioanalyysissä, mikä hiukan heikentää tutkimuksen loppupäätelmiä. Tutkijat toteavat API:n käyttöönoton olevan erityisen kiinnostava investointi etenkin IT-alan firmoille, mutta muidenkin alojen organisaatiot voivat varmasti hyötyä siitä, erityisesti sisäisiä kustannuksia alentamalla sisäisen API:n avulla. Tutkimusaineiston ollessa kattava useamman vuoden ajalta ja sisältäen organisaatioita eri toimialoilta, tutkimusmenetelmät sekä niillä saadut johtopäätökset vaikuttavat varsin päteviltä.

Wulf ja Blohm tekivät vuonna 2020 tutkimuksen, jonka avulla he pyrkivät selvittämään, miten organisaatiot voivat tuottaa arvoa API:en avulla. Tutkimuksen aineistona heillä oli 152 organisaatiota, jotka tarjoavat muille organisaatioille tuottoa tavoittelevasti tavalla tai toisella API:a. Tutkimuksen tulosten yleistyksen mahdollistamiseksi organisaatioita kerättiin IT- ja viestintäalalta (41 %), koulutuslta (14 %), markkinoinnista (10 %), jälleenmyynnistä (7 %) ja viihdealalta (5 %). Useimmat organisaatioista olivat Yhdysvalloista (38 %), Saksasta (12 %), Iso-Britanniasta (9 %), Sveitsistä (7 %) ja Ranskasta (5 %). Tutkimuskysymykset pyrittiin asettelemaan mahdollisimman hyvin siten, etteivät ne vääristäisi vastauksia. Tutkimusasetelma vaikuttaa mielestäni tarkkaan harkitulta sekä luotettavalta. Wulf ja Blohm esittävät tutkimuksessaan API-tarjoajien tarjoavan yhtä tai useampaa kolmesta erilaisesta palvelusta, jotka sisältävät kukin omat erityispiirteensä liittyen API:n arkkitehtuurin ja hallinnon (governance) suunnitteluun. Tutkimuksessa ei huomioitu sisäistä API:a ja sen tuottamia hyötyjä lainkaan, vaan siinä keskityttiin ainoastaan organisaation ulkoisiin käyttäjiin.

Ensimmäinen näistä palveluista on ammatilliset palvelut (professional service). Ammatillisen palvelun API tarjoaa infrastruktuuria, alustaa, dataa tai sovellusta palveluna (SaaS, software-as-a-service). Ammatillisen palvelun API tarjotaan yleensä asennettavana ohjelmistona tai verkkosivuilla saatavilla olevana palveluna. Pilvipalveluihin pääsyn käyttäjä saa joko suoralla tai epäsuoralla maksulla. Tällaisia palveluita ovat esimerkiksi Amazon S3 API ja Google Maps API. Heidän määrittelemä ammatilliset palvelut voidaan rinnastaa tässä tutkielmassa aiemmin määriteltyyn kumppani-API:in.

Tutkijoiden määrittelemä toinen palvelu, välipalvelu (mediation service), tarjoaa pääsyn API-tarjoajan alustan resursseihin. Näitä resursseja hyödyntämällä kolmannen osapuolen kehittäjät suunnittelevat ja toteuttavat alustan ydinpalvelua täydentäviä lisäpalveluita alustan loppukäyttäjille. Esimerkkejä tällaisesta välipalvelu-API:sta ovat Facebook Graph API ja Twitter Direct Mes-

sage API. Tutkimuksen välipalvelun määrittelyllä tarkoitetaan samaa määrittelyä, mitä aiemmin tässä tutkielmassa määrittelyllä avoimella API:lla.

Kolmas tutkijoiden määrittelemä palvelu on avoimen datan palvelu API (open asset service). Avoimen datan palvelu API antaa helpon ja ilmaisen pääsyn kenelle tahansa organisaation sisäisiin IT-resursseihin. Pääsy voi olla organisaation määrittelemään dataan, infrastruktuuriin tai sovelluksiin maksutta, eli kyseessä on tämän tutkielman määrittelyksen mukainen julkinen API. Julkisia API:a ovat muun muassa Github API ja avoindata.fi-API.

Tutkijat käsittelevät kahta tunnistamaansa liiketoiminnan päätavoitetta, arvon sisäistämistä (value appropriation) ja arvon tuottamista (value generation). Arvon sisäistys aikaansaadaan saavuttamalla positiivinen API:n tuottoaste (engl. return on investment, ROI) aikaansaatuisten tulovirtojen avulla. API:n tuottoaste kertoo suhteen, kuinka paljon API:n kehitykseen ja toimintaan sijoitettu pääoma on tuottanut verrattuna sijoitettuun pääomaan. Arvoa taas tuotetaan tutkimuksen mukaan saavuttamalla korkean tason API-diffuusiota, joka voi tuottaa organisaation sisäistä innovointia. API-diffuusiolla API-tarjoaja ei pyri suoriin kaupallisiin hyötyihin, vaan sen ensisijainen tehtävä on parantaa sisäistä innovointia. Tutkijoilla on myös tutkimuksessaan kaksi tärkeää näkökulmaa, rinnakkaistuotanto (economies of scope in production) ja rinnakkaisinnovointi (economies of scope in innovation). Rinnakkaistuotannolla tarkoitetaan ilmiötä, jossa tuotteen samanaikainen yhteistuotanto (joint production) käy edullisemmaksi kuin puolivalmisteen ja lopputuotteen tuottaminen erillään (Panzar & Willig, 1981). Gawer (2014) määrittelee rinnakkaisinnovoinnin ilmiöksi, jossa kahden tuotteen rinnakkainen innovointi on edullisempaa kuin tuotteiden innovointi erillään.

Kerättyään ja analysoituaan tutkimuksen aineiston, tutkijat päättelivät tutkimuksen analyysien tuloksien olevan pääosin linjassaan heidän hypoteesien kanssa. Tutkijat onnistuivat osoittamaan heidän ensimmäisen hypoteesinsa oikeaksi, kumppani-API rinnakkaistuotannon näkökulmasta tuottaa positiivisen tuottoasteen. Tutkijat osoittavat kumppani-API:en standardisoitujen käyttöliittymien helpottavan integraatioissa, koska kehittäjien ei tarvitse nähdä niin paljon vaivaa tutustuakseen palvelun toiminnallisuuksiin ja syntaksiin. Tämän voidaan arvioida olevan yksi tekijä siinä, miksi hypoteesi onnistuttiin toteamaan todeksi. Toinen hypoteesi sai tutkimuksessa myös vahvistuksen, avoin API rinnakkaistuotannon näkökulmasta tuottaa myös positiivisen tuottoasteen.

Julkiset API:t tähtäävät suoran tulon kasvattamisen sijaan uusiin tapoihin kasvattaa arvoa osallistamalla kolmannen osapuolen kehittäjiä käyttämään organisaation julkista APIa. Täten API:t voivat laajaa mielenkiintoa herättämällä tuottaa uusia mashupeja ja sisäistä innovointia, mikä lopulta voi näkyä myös taloudellisena menestyksenä. Tutkimuksen kolmannessa hypoteesissa oletettiin julkisten API:en tuottavan API-diffuusiota rinnakkaisinnovoinnin avulla. Tutkimus tuki myös tätä kolmatta hypoteesia, tuloksista selvisi julkisen API:n tuottaman rinnakkaisinnovoinnin olevan positiivisesti yhteydessä API-diffuusioon. API-diffuusion lisäksi nämä julkiset API:t myös rohkaisevat API-tarjoajaa kanssakäymään ulkoisten kehittäjien kanssa, sillä se voi muiden hyötyjen lisäksi

auttaa organisaatiota houkuttelemaan uusia kehittäjiä palvelukseensa, koska he ovat jo saattaneet tutustua API:n käyttöön ja todenneet sen toimivaksi. Fengerin, Henriksenin, Hedmanin ja Xiaon (2018) tutkimuksen loppupäätelmät tukevat tätä, heidän tutkimuksessaan julkisen API:n eniten keskusteltu ja siten merkittävin tekijä on yhteistyö. Heidän tutkimuksensa osoitti, että julkisen API:n arvonluonti perustuu välillisiin tuottoihin, joita ei voida suoraan mitata, kuten brändäykseen, innovointiin ja yhteistyöhön. Wulf ja Blohm eivät kuitenkaan saaneet varmistusta neljännelle hypoteesilleen, avoin API rinnakkaisinnovoinnin näkökulmasta ei ole positiivisesti yhteydessä API-diffuusioon. Täten he toteavat avoimen API:n tuottavan rinnakkaisinnovoinnin olevan riittämätöntä saadakseen aikaan API-diffuusiota.

Tutkijat tulevat siihen loppupäätelmässään tulokseen, että API-tarjoajien tulisi tarkoin miettiä, minkälaista API-tyyppiä ne lähtevät tarjoamaan ja mikä toimii heidän arvon luomisen strategian mukaisesti. Mielestäni tutkijoiden tutkimustulokset ovat merkittäviä. He osoittivat, että kumppani- tai avoimella API:lla voidaan saavuttaa positiivinen tuottoaste suorien tai epäsuorien käyttömaksujen avulla. Myös heidän osoittamat julkisen API:n hyödyt ovat huomattavia, kasvanut sisäinen innovointi sekä mahdollisesti uusien kehittäjien houkuttelemisen organisaation palvelukseen ovat varmasti tekijöitä, joita jokainen organisaatio osaa arvostaa.

## 4 JOHTOPÄÄTÖKSET JA YHTEENVETO

Tässä luvussa kertaan tutkielman oleelliset vaiheet sekä ilmi tulleet seikat. Niiden avulla teen lopulliset johtopäätökset sekä pohdin niiden merkityksiä. Lopuksi huomioin tutkielman rajoitteet sekä ehdotan jatkotutkimustarvetta.

Sekalaisten, ristiriitaisten ja epä johdonmukaisten termien vuoksi pyrin aluksi selventämään ja tekemään eroja, mitä tarkoitetaan avoimilla-, julkisilla-, sisäisillä- ja kumppani-API:lla. Samoilla termeillä eri kirjallisuuksissa eri asioita tarkoittaessa aikaansaa tarpeen siitä, että termistön yhteneväisyyttä kaivattaisiin lisää. Huomioin tähän tutkimukseen vain tieteellisten tekstien määrittelyjä eri API-luokitteluista, blogikirjoitusten ja verkko-oppaiden määritelmien kirjo on vielä tätä suurempi. API:n historiaa ja kehitystä tarkastelemalla huomattiin, miten merkittävässä roolissa API:t ovat olleet kaikkien meidän laajasti käyttämien järjestelmien ja sovelluksien keskiössä. API:en kehityksen myötä syntyneet merkittävät teknologiat hyödyttävät suuresti monilla alueilla, eri alojen tutkimustyöt olisivat epäilemättä ainakin hidastuneet ilman pilvilaskennan valtavaa tehoa.

API:n mahdollistamat ja tunnistetut hyödyt eri organisaatioille ovat kiistattomia. Tässä tutkielmassa ilmi tulleet tehokkuutta kasvattavat tekijät, kuten parempi rinnakkainen ja itsenäinen työ, päällekkäisen työn tekemisen minimoiminen sekä kommunikaation tehostaminen ovat asioita, jotka toteutuessaan voivat olla merkittävä kilpailutekijöitä mille tahansa organisaatiolle. Tehokkuutta kasvattavien hyötyjen lisäksi API:t mahdollistavat myös datan keräämisen ja siten ekosysteemien rakentamisen. Tutkielmassa selvinneet mahdollisuudet hyödyntää API:n keräämää dataa päätöksenteossa, ydinpalvelua laajentaessa tai pelkästään dataa myydessä auttavat varmasti eri organisaatiota saavuttamaan kilpailuetua markkinoilla. Tutkielmassa ilmi tulleet taloudelliset hyödyt voidaan saavuttaa ensisijaisesti kustannuksia alentamalla sisäisiä API:a hyödyntäen. Tuloja organisaatiot voivat saavuttaa kumppani- ja avointen-API:en suorilla ja epäsuorilla käyttömaksuilla. Tutkielmassa on selvinnyt myös organisaatioiden mahdollisuus suurempaan myyntiin sisäisten API:en mahdollistaman paremman asiakaskokemuksen avulla. Epäsuoria taloudellisia hyötyjä voidaan saavuttaa myös julkisen API:n aikaansaamalla sisäisellä innovoinnilla



sekä ulkoisia kehittäjiä houkuttelemalla. Kaiken kaikkiaan API:en voidaan todeta olevan merkittävä kilpailutekijä oikein hyödynnettynä.

Tässä tutkielmassa otettiin kantaa API:n mahdollistamiin kilpailutekijöihin, mutta ei siihen, miten nämä kilpailutekijät aikaansaadaan API:n avulla. Vaikka API:en mahdollistamat hyödyt ovat selviä, API:en itsensä toteuttaminen ei ole itsestään selvää. Tässä tutkielmassa ilmi tullut seikka API-kutsujen ja API:n tuottaman hyötyjen korrelaatiosta on merkki tästä, API ei tuota hyötyjä, ellei sitä käytetä. Runsaasti kirjallisuudessa ja tieteellisessä keskustelussa esillä ollut API-suunnittelu on myös erittäin tärkeässä roolissa, huonosti suunniteltu ja toteutettu API ei varmasti tuota merkittäviä tuloksia. Lisäksi on selvää, että eri API:t tuottavat erilaisia hyötyjä eri organisaatioille, joten yksiselitteistä yleistä vastausta siihen, kuinka kaikkia hyötyjä saavuttava yksi API tulisi tehdä, tuskin saadaan aikaan. Organisaation pyrkiessä hyödyntämään API:a liiketoiminnassaan sen tulisikin tarkkaan suunnitella ja toteuttaa juuri sen liiketoimintaan sopiva API.

Tutkielman aiheen jatkotutkimus voisi liittyä teoreettisen perustan tai viitekehysten luomiselle, jota hyödyntämällä organisaatio saisi parhaat edellytykset jalkauttaa jokin neljästä API-luokasta omaan käyttöön. Myös kattavasti taloudellisia resursseja mittaavia tutkimuksia API-käyttöön otosta tarvittaisiin lisää niiden niukkuuden vuoksi, jotta tämänkin tutkimuksen johtopäätöksille saataisiin lisää tukea.

## LÄHTEET

Amazon. (16.6.2002). Amazon.com Launches Web Services; Developers Can Now Incorporate Amazon.com Content and Features into Their Own Web Sites; Extends "Welcome Mat" for Developers. Haettu osoitteesta <https://press.aboutamazon.com/news-releases/news-release-details/amazoncom-launches-web-services/>

Amazon. (13.3.2006a). Announcing Amazon S3 - Simple Storage Service. Haettu osoitteesta <https://aws.amazon.com/about-aws/whats-new/2006/03/13/announcing-amazon-s3---simple-storage-service/>

Amazon (24.8.2006b). Announcing Amazon Elastic Compute Cloud (Amazon EC2) – beta. Haettu osoitteesta <https://aws.amazon.com/about-aws/whats-new/2006/08/24/announcing-amazon-elastic-compute-cloud-amazon-ec2---beta/>

Avoindata. (15.6.2020). Rajapinta Helsingin pysäköintipaikkojen käytöstä. Haettu osoitteesta <https://www.avoindata.fi/data/fi/dataset/rajapinta-helsingin-pysakointipaikkojen-kaytosta>

Bahrami, M., Park, J., Liu, L., & Chen, W. P. (2018, April). API learning: applying machine learning to manage the rise of API economy. In Companion Proceedings of the The Web Conference 2018 (pp. 151-154).

Bala I. & Subramaniam M., (13.4.2015). Are You Using APIs to Gain Competitive Advantage? Harvard Business Review. Haettu osoitteesta <https://hbr.org/2015/04/are-you-using-apis-to-gain-competitive-advantage>

Bataineh, A. S., Mizouni, R., El Barachi, M., & Bentahar, J. (2016). Monetizing personal data: a two-sided market approach. *Procedia Computer Science*, 83, 472-479.

Benzell, S., Lagarda, G., & Van Alstyne, M., The Impact of APIs in Firm Performance (May 21, 2017). Boston University Questrom School of Business Research Paper No. 2843326

De Souza, C. R., & Redmiles, D. F. (2009). On the roles of APIs in the coordination of collaborative software development. *Computer Supported Cooperative Work (CSCW)*, 18(5), 445-475.

Evans, P. & Basole, R. (2016). Economic and Business Dimensions: Revealing the API Ecosystem and Enterprise Strategy via Visual Analytics.

Association for Computing Machinery. *Communications of the ACM*, 59(2), 26.

Farahat, A., & Bailey, M. C. (2012, April). How effective is targeted advertising?. In *Proceedings of the 21st international conference on World Wide Web* (pp. 111-120).

Farrell, S. (2009). API Keys to the Kingdom. *IEEE Internet Computing*, 13(5), 91-93.

Fenger, F., Henriksen, S. W., Hedman, J., & Xiao, X. (2018). Open API: An Investigation of its Business Value.

Gawer, A. (2014). Bridging differing perspectives on technological platforms: Toward an integrative framework. *Research policy*, 43(7), 1239-1249.

Google. (8.2.2005a). Mapping your way. Haettu osoitteesta <https://googleblog.blogspot.com/2005/02/mapping-your-way.html>

Google. (29.6.2005b). The world is your JavaScript-enabled oyster. Haettu osoitteesta [https://googleblog.blogspot.com/2005/06/world-is-your-javascript-enabled\\_29.html](https://googleblog.blogspot.com/2005/06/world-is-your-javascript-enabled_29.html)

Jackson, K. R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., Wasserman, H. J. & Wright, N. J. (2010, November). Performance analysis of high performance computing applications on the amazon web services cloud. In *2010 IEEE second international conference on cloud computing technology and science* (pp. 159-168). IEEE.

Kopecký, J., Fremantle, P. & Boakes, R. (2014). A history and future of Web APIs. *it - Information Technology*, 56(3), 90-97.

Lane, K. (2016) API Evangelist: Some Milestones From The Last 15 Years Of Web API History [blogikirjoitus]. Haettu osoitteesta <https://apievangelist.com/2016/03/27/some-milestones-from-the-last-15-years-of-web-api-history/>

Lindman, J., Horkoff, J., Hammouda, I., & Knauss, E. (2018). Emerging perspectives of application programming interface strategy: A framework to respond to business concerns. *IEEE Software*, 37(2), 52-59.

Mckee, D. (2017). The platform economy: Natural, neutral, consensual and efficient? *Transnational legal theory*, 8(4), 455-495.

Meng, M., Steinhardt, S., & Schubert, A. (2018). Application programming interface documentation: what do software developers want?. *Journal of Technical Writing and Communication*, 48(3), 295-330.

- Microsoft. (31.7.2015). Annual Report 2015. Haettu osoitteesta  
<https://www.microsoft.com/investor/reports/ar15/index.html>
- Murugesan, S. (2007). Understanding Web 2.0. *IT professional*, 9(4), 34-41.
- Myers, Brad, ja Jeffrey Stylos. "Improving API Usability." *Communications of the ACM* 59, no. 6 (2016): 62-69.
- Najjar, M. S., & Kettinger, W. J. (2013). Data Monetization: Lessons from a Retailer's Journey. *MIS Quarterly Executive*, 12(4).
- Newcombe, C., Rath, T., Zhang, F., Munteanu, B., Brooker, M., & Deardeuff, M. (2015). How Amazon web services uses formal methods. *Communications of the ACM*, 58(4), 66-73.
- Palankar, M. R., Iamnitchi, A., Ripeanu, M., & Garfinkel, S. (2008, June). Amazon S3 for science grids: a viable solution?. In *Proceedings of the 2008 international workshop on Data-aware distributed computing* (pp. 55-64).
- Panzar, J. C., & Willig, R. D. (1981). Economies of scope. *The American Economic Review*, 71(2), 268-272.
- Rao, L. (30.9.2011). Sequoia-Backed Stripe Wants To Disrupt The Online Payments Space. Haettu osoitteesta  
<https://techcrunch.com/2011/09/30/sequoia-backed-stripe-launches-to-disrupt-the-online-payments-industry-with-a-developer-friendly-platform/>
- Sohan, S. M., Anslow, C., & Maurer, F. (2015, June). A case study of web API evolution. In *2015 IEEE World Congress on Services* (pp. 245-252). IEEE.
- Tan, W., Fan, Y., Ghoneim, A., Hossain, M. A., & Dustdar, S. (2016). From the service-oriented architecture to the web API economy. *IEEE Internet Computing*, 20(4), 64-68.
- Twitter. (20.9.2006). Introducing the Twitter API. Haettu osoitteesta  
[https://blog.twitter.com/en\\_us/a/2006/introducing-the-twitter-api.html](https://blog.twitter.com/en_us/a/2006/introducing-the-twitter-api.html)
- Wang, G., & Ng, T. E. (2010, March). The impact of virtualization on network performance of amazon ec2 data center. In *2010 Proceedings IEEE INFOCOM* (pp. 1-9). IEEE.
- Wang, S., Keivanloo, I., & Zou, Y. (2014, November). How do developers react to restful api evolution?. In *International Conference on Service-Oriented Computing* (pp. 245-259). Springer, Berlin, Heidelberg.
- Wixom, B. H., & Ross, J. W. (2017). How to monetize your data. *MIT Sloan Management Review*, 58(3).

- Wulf, J., & Blohm, I. (2020). Fostering value creation with digital platforms: A unified theory of the application programming interface design. *Journal of Management Information Systems*, 37(1), 251-281.
- Yu, Y., Silveira, H., & Sundaram, M. (2016, October). A microservice based reference architecture model in the context of enterprise architecture. In *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)* (pp. 1856-1860). IEEE.
- Zachariadis, M. & Ozcan, P. (2016). The API Economy and Digital Transformation in Financial Services: The Case of Open Banking. *SSRN Electronic Journal*.