



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Tuomas Mäkelä

# B2C-asiakashankintaputken automatisointi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Ohjelmistotuotanto

Insinöörityö

12.4.2021

Tekijä Otsikko	Tuomas Mäkelä B2C-asiakashankintaputken automatisointi
Sivumäärä Aika	32 sivua 12.4.2021
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintäteknikka
Ammatillinen pääaine	Ohjelmistotuotanto
Ohjaajat	Kehitysjohtaja Henri Haho Yliopettaja Auvo Häkkinen
<p>Insinööriyössä oli tavoitteena automatisoida kotisiivousyritys Sahera Koti Puhtaaksi Oy:n uusiasiakashankintaan käytettävä asiakashankintaputki. Kyseessä on kokonaisuus, joka kattaa myynnin vaiheet aina yhteystietojen hankinnasta asiakashankintatiimin yhteydenottoon. Toteutetun automaation avulla oli tarkoitus parantaa asiakasyrityksen asiakashankintatiimin tehokkuutta tuomalla postinumeron perusteella validoidut yhteystiedot asiakashankintatiimin soittojärjestelmään automatisoidusti. Ratkaisussa pyrittiin ottamaan huomioon myös yksityisyyden suojaan ja toimialueiden muuttumiseen liittyviä kehityskohtia, joita projektin määrittelyn yhteydessä havaittiin.</p> <p>Insinööriyön teoriaosassa käsiteltiin sovellusten välistä tiedonsiirtoa perehtymällä HTTP-pyyntöjen ja -vastausten rakenteeseen. Lisäksi teoriaosassa vertailtiin kahta yleisintä tekniikkaa, joilla sovellusten välisen tiedonsiirron käynnistäviä muutoksia voidaan tarkkailla seurattavassa sovelluksessa. Projektin toteuttamista käsittelevissä osioissa käytiin läpi asiakashankintaputken rakentaminen aina ongelmien määrittelystä uuden putken käyttöönottoon.</p> <p>Opinnäytetyön lopputuloksena käyttöön otettiin helposti ylläpidettävä automaatio, joka yhdistää kaikki yrityksen asiakashankintaan käyttämät sovellukset. Opinnäytetyö pyrkii kuvaamaan, kuinka maantieteellisesti eri alueilla toimiva yritys voi kustannustehokkaasti automatisoida potentiaalisten asiakkaiden validoinnin. Validoinnin avulla yrityksen myyntiorganisaation aika saadaan hyödynnettyä mahdollisimman tehokkaasti parantaen samalla alueellisen myynnin ennakoitavuutta.</p> <p>Tässä opinnäytetyössä kuvattavat prosessit ovat sovellettavissa lähes kaikille eri aloille – vain mielikuvitus on rajana.</p>	
Avainsanat	Integromat, myyntiautomaatio, asiakashankinta

Author Title	Tuomas Mäkelä Automatization of B2C Customer Acquisition Funnel
Number of Pages Date	32 pages 12 April 2021
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Instructors	Henri Haho, Head of Development Auvo Häkkinen, Principal Lecturer
<p>The goal of this project was to automate the customer acquisition funnel used by a home cleaning company Sahera Koti Puhtaaksi Oy. The purpose of the automation was to improve the efficiency of the company's customer acquisition team by importing validated contacts into the team's call system automatically. The validation process of the leads was based on the leads' postal codes. The solution also sought to take into account development tasks related to the protection of privacy and changes in the company's working areas.</p> <p>The theoretical part of the work focused on data transfer between applications, covering the structures of HTTP requests and responses. In addition, the theoretical part focused on a comparison of the two most common techniques that can be used to monitor the changes that trigger the transfer of data between applications. The sections dealing with the implementation of the project covered the construction of the customer acquisition funnel from the definition of problems to the commissioning of the new funnel.</p> <p>As a result of the thesis, an easy-to-maintain entity was introduced, connecting all the applications used by the company for customer acquisition. The thesis describes how a company operating in different geographical areas can automate the validation of potential customers. With the help of validation, the time of the company's sales organization can be utilized as efficiently as possible, while improving the predictability of regional sales.</p> <p>The processes described in this thesis apply to almost all different fields - only the imagination is the limit.</p>	
Keywords	Integromat, sales automation, customer acquisition

## Sisällys

### Lyhenteet

1	Johdanto	1
2	Lähtötilanne	2
2.1	Toimialueiden hallinnan haasteet	3
2.2	Soittojärjestelmän ominaisuuksien hyödyntämättömyys	4
2.3	GDPR-vaatimuksien täyttäminen	5
3	Ohjelmistojen välinen tiedonsiirto	5
3.1	HTTP-protokolla	5
3.1.1	HTTP-pyyntöjen rakenne	6
3.1.2	HTTP-vastausten rakenne	7
3.2	Ajastettu muutosten tarkkailu	8
3.3	Tapahtumaperusteinen muutosten tarkkailu	9
4	Integromat	11
4.1	Moduulit	11
4.2	Skenaariot	15
4.3	Herätteet	16
4.4	Reitittimet	16
5	Uudistettu asiakashankintaputki	19
5.1	Asiakashankintaputken uusi rakenne	20
5.2	Postinumeroiden vertailu	21
5.3	Liidin tallentaminen CRM:ään	23
5.4	Liidin lisääminen soittojärjestelmään	24
6	Käyttöönotto	25
7	Yhteenveto	28
	Lähteet	30

## Lyhenteet

API	Application Programming Interface eli ohjelmointirajapinta, joka mahdollistaa ohjelmistojen välisen vuorovaikutuksen standardoitujen pyyntöjen avulla.
CRM	Customer Relationship Management eli asiakkuudenhallinta.
CSV	Pilkulla erotetut arvot (Comma-Separated Values) on tiedostomuoto, jota käytetään yleisesti yksinkertaisen tiedon siirtämisessä ohjelmasta toiseen.
GDPR	General Data Protection Regulation on henkilötietojen käsittelyä sääntelevä laki.
HTTP	Hypertext Transfer Protocol on protokolla, jonka avulla selaimet ja palvelimet siirtävät tietoa.
JSON	JavaScript Object Notation on dataformaatti, jota käytetään laajalti tiedon välittämiseen ohjelmistojen välillä.
SAAS	Software as a Service on pilvipohjainen ohjelmisto, jonka käytöstä maksetaan usein kuukausi- tai vuosimaksua.
URL	Uniform Resource Locator on internetissä olevien resurssien nimeämiseen ja määrittelemiseen käytettävä standardi.

## 1 Johdanto

Projektin tavoitteena oli automatisoida asiakasyrityksen uusasiakashankintaan käyttämä asiakashankintaputki. Asiakashankintaputkella tarkoitetaan kokonaisuutta, joka kattaa myynnin eri vaiheet yhteystietojen hankkimisesta asiakashankintatiimin yhteydenottoon. Työssä toteutettavan asiakashankintaputken automatisoinnin avulla oli tarkoitus tehostaa ja helpottaa asiakashankintatiimin päivittäistä työntekoa. Myös yrityksen toimialueiden ja asiakastietojen hallintaa pyrittiin samalla kehittämään, ja näin vähentämään toimialueiden supistamisen tai laajenemisen aiheuttamaa työkuormaa myynti- ja markkinointitiimeissä.

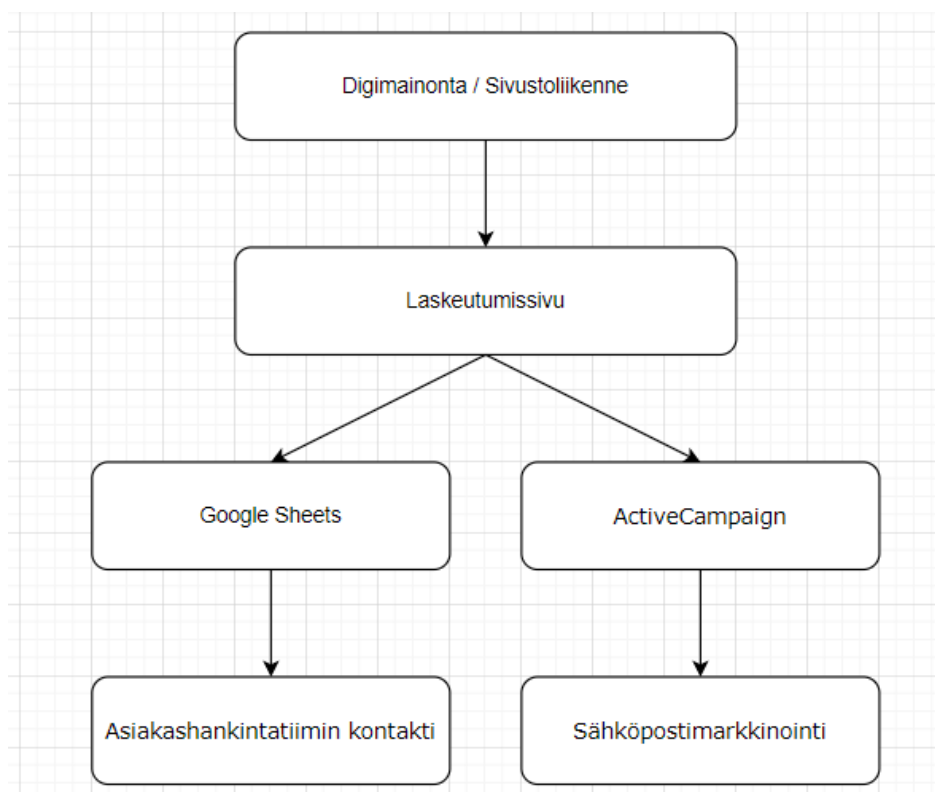
Työn asiakasyrityksenä oli Tampereella vuonna 2011 perustettu kotisiivousyritys Sahera Koti Puhtaaksi Oy (myöhemmin Koti Puhtaaksi). Yritys on yksi Suomen suurimmista kotisiivousalan yrityksistä. Vuonna 2020 yrityksen liikevaihto oli lähes 5 miljoonaa euroa. Etenkin viimeisen muutaman vuoden aikana Koti Puhtaaksi on kasvanut voimakkaasti. Eriyisen nopeaa kasvu oli vuonna 2019, jolloin yritys ylsi jopa yli 120 % liikevaihdon kasvuun. (1.) Nopean kasvun myötä muun muassa asiakashankinnan prosessit eivät enää vastanneet kasvaneen volyymin asettamia vaatimuksia. Uusasiakashankinnan prosessien automatisointi nousikin korkeimman prioriteetin tehtäväksi työsuhteeni alkupuolella. Tässä insinööriyössä kuvatussa automatisaation ja työkalujen integroimisen avulla säästetään vuosittain jopa satoja tunteja työaika. Muita automaation kautta tavoiteltuja hyötyjä ovat muun muassa jatkomarkkinointimahdollisuuksien parantuminen ja potentiaalisten asiakkaiden tietojen organisoidumpi hallinnointi.

Tämän insinööriyön toisessa luvussa alustetaan syitä projektin toteuttamiseen. Kolmannessa luvussa perehdytään ohjelmistojen väliseen tiedonsiirtoon ja tutustutaan tarkemmin HTTP-protokollaan (Hypertext Transfer Protocol). Neljännessä luvussa esitellään projektin toteuttamisessa käytettävää visuaalista automaatiotyökalua, ja keskitytään erityisesti toiminnallisuuksiin, joita toteutetussa automaatiossa on käytetty. Viidennessä luvussa syvennytään toteutetun automaation rakenteeseen. Kuudennessä luvussa käydään läpi automaation käyttöönottoa ja pieniä korjauksia, joita automaation toimintaan tehtiin käyttöönoton jälkeen. Seitsemännessä ja viimeisessä luvussa kerrataan projektin kulkua ja otetaan katsaus tulevaan.

Automaatiossa käytetyt ratkaisut on pyritty kuvaamaan työssä riittävällä tarkkuudella, jotta tämän työn lukija voi halutessaan toistaa samankaltaisia, automaattisesti potentiaaliset asiakkaat validoivia automaatioita omissa liiketoiminnassaan.

## 2 Lähtötilanne

Koti Puhtaaksi Oy:n uusasiakashankinta toimii kokonaisuutena hyvin suoraviivaisesti. Aluksi potentiaalisen asiakkaan (myöhemmin liidi) yhteystiedot hankitaan erilaisin digimarkkinoinnin keinoin. Yhteystietojen hankkimisen jälkeen asiakashankintatiimi ottaa liidiin yhteyttä puhelimitse ja kartoittaa hänen siivoustarpeensa. Liidi lisätään myös yrityksen sähköpostilistalle, minkä kautta yrityksen palveluita tai yhteistyötarjouksia voidaan markkinoida hänelle myös jälkikäteen. Kuvassa 1 on kuvattu yksinkertaistettuna asiakashankintaputki siinä muodossa, mistä projektissa lähdettiin liikkeelle.



Kuva 1. Asiakashankinnan rakenne ennen projektin aloittamista.

Kuten kuvasta käy ilmi, laskeutumissivun lomake välitti liidin yhteystiedot suoraan Google Sheets -taulukkoon. Asiakashankintatiimi käytti kyseistä taulukkoa päivittäisessä työssään soittolistana ja asiakasmuistiinpanojen kirjauspaikkana. Liidien yhteystiedot välitettiin suoraan lomakkeelta myös CRM- (Customer Relationship Management) eli asiakkuudenhallintajärjestelmä ActiveCampaigniin, jonka kautta yritys markkinoi omia ja yhteistyökumppaneiden palveluita ja tuotteita sähköpostitse.

Alkuperäiseen asiakashankintaputkeen liittyi useita haasteita. Päivittäistä työntekoa ja sen johtamista vaikeuttivat erityisesti manuaalista työtä vaativat tehtävät, jotka virheettömyyden lisäksi veivät myös aikaa. Tunnistetut haasteet voidaan tiivistää kolmeen pääkohtaan: toimialueiden hallinnan haasteisiin, soittajärjestelmän ominaisuuksien hyödyntämättömyyteen ja henkilötietojen säilytyksen puutteisiin. Nämä haasteet on kuvattu seuraavissa luvuissa.

## 2.1 Toimialueiden hallinnan haasteet

Koti Puhtaaksi toimii Suomen laajuisesti kuudella toimialueella, jotka muodostuvat kyseisen alueen aluetoimiston ympärille. Toimialueet on rajattu suhteellisen tarkasti postinumeroiden avulla. Asiakashankintaputkeen päätyvistä liideistä kuitenkin noin 20 % asuu yrityksen toimialueiden ulkopuolella, minkä takia heille soittamiseen ei ole hyödyllistä käyttää asiakashankintatiimin aikaa.

Yritys oli alun perin ratkaissut liidien asuinpaikan tarkistamisen siten, että kaikki noin 350 toimialueiden postinumeroa oli lisätty liidien yhteystietoja sisältävän Google-taulukon ehdollisen muotoilun sääntöihin. Soittajat näkivät liidin postinumeron sisältävän solun taustaväristä, asuiko liidi jollain yrityksen toimialueista. Värikoodien avulla voitiin välttää suurin osa puheluista toimialueiden ulkopuolella asuville liideille.

Postinumeroiden hallinnointi ehdollisen muotoilun säännöissä oli kuitenkin monilta osin vaivalloista. Suurin syy tähän oli, että ehdollisen muotoilun säännöt eivät tukeneet minikäänlaisia hakutoimintoja. Esimerkiksi postinumeroa poistaessa yrityksen työntekijän tuli selata manuaalisesti läpi koko noin 350 postinumeron listaus taulukon sivupalkissa.



Tästä johtuen postinumeroiden muokkaaminen ja tarkistaminen oli hidasta. Lisäksi postinumeroita saatettiin myös vahingossa poistaa tai niiden järjestystä muuttaa. Usein tahattomasti tehdyt poistot ja järjestyksen muutokset ehdollisen muotoilun säännöissä sotkivat sääntöjoukon toiminnan ajoittain jopa pidemmäksikin aikaa, mikä vaikeutti asiakashankintatiimin soittotyötä.

## 2.2 Soittojärjestelmän ominaisuuksien hyödyntämättömyys

Postinumeroiden muokkaamisen kankeutta ja tahatonta muokkaamista isompi ongelma oli, ettei asiakashankintatiimi pystynyt hyödyntämään kaikkia soittojärjestelmän tuomia etuja asiakashankintaputken kautta hankittuihin liideihin. Isoin ongelma oli, että asiakashankintatiimi joutui lisäämään liidit soittojärjestelmään käsin. Yksitellen tehtyyn liidien lisäämiseen kului todella paljon tehokasta soittoaikaa. Valmiiden soittolistojen avulla asiakashankkijat olisivat voineet vain avata puhelinyhteyden, jolloin järjestelmä olisi puhelun päättymisen jälkeen soittanut seuraavalle liidille automaattisesti.

Kymmeniä toimialueella asuvia liidejä saattoi jäädä myös kontaktoimatta. Tämä johtui siitä, että liideille ei aina pystytty kysyntätilanteesta johtuen soittamaan siinä järjestyksessä, kun ne oli onnistuttu hankkimaan. Asiakashankkijat saattoivat myös soittaa samalle henkilölle useita kertoja, mikä ei ollut tarkoituksenmukaista. Syy ylimääräisiin puheluyrityksiin oli, että asiakashankkijat eivät tieneet toistensa puheluista, ja sama henkilö oli saattanut jättää yhteystietonsa useampaan otteeseen.

Ensimmäisenä ratkaisuyrityksenä manuaalisesti validoituja liidejä alettiin siirtämään CSV-tiedostoina (Comma-Separated Values) soittojärjestelmään. Näin soittajille saatiin valmiit listat soitettavaksi ja taulukoiden kanssa työskentelyä saatiin vähennettyä voimakkaasti. Pian kuitenkin ymmärrettiin, ettei ratkaisu ollut pidemmän päälle kestävä. Järjestelmään lisättävien soittolistojen muodostaminen manuaalisesti tuhansien liidien massasta oli paitsi työlästä ja hidasta, myös virheeltistä. Lopulliseksi tehtäväkseni jäi siis ratkaista, miten voisimme validoida liidit postinumeron perusteella ja lisätä heidät soittojärjestelmään yhteydenottoa varten niin, ettei kyseiseen tehtävään tarvinnut käyttää kenenkään työaikaa.

### 2.3 GDPR-vaatimusten täyttäminen

Omalta osaltaan painetta yhteystietojen ja muistiinpanojen keskittämiseen Google-taulukon sijaan CRM:ään ja soittojärjestelmään lisäsi GDPR:n (General Data Protection Regulation) asettamat vaatimukset henkilötietojen käsittelylle. GDPR on keväällä 2018 EU-maissa voimaan astunut henkilötietojen käsittelyä sääntelevä laki, joka takaa henkilölle oikeuden muun muassa saada tai korjata tietoja, joita yrityksellä on hänestä hallussaan. Henkilöllä on myös oikeus pyytää kaikkien häntä koskevien tietojen poistoa. (2.) Tietojen kerääminen massiiviseksi paisuneesta Google-taulukosta henkilötietojen käsittelyyn tarkoitettuihin järjestelmiin oli siis tarpeen, jotta esimerkiksi lain edellyttämät, tietojen hallintaan liittyvät pyynnöt voitaisiin käsitellä jatkossa tehokkaammin.

## 3 Ohjelmistojen välinen tiedonsiirto

Haasteiden tunnistamisesta asti oli selvää, että sopivan ratkaisun löytämiseksi liidien yhteystietoja tuli pystyä siirtämään eri ohjelmistojen välillä. Tästä syystä ratkaisussa tuli hyödyntää useamman eri ohjelmiston rajapintoja. Rajapinnat ovat joukko sääntöjä ja käytänteitä, joita noudattamalla sovellus voi käyttää toisen ohjelmiston palveluita ja resursseja. Rajapinnat eli API:t (Application Programming Interface) toimivat liittyminä eri ohjelmistojen välillä ja mahdollistaisivat näin projektin toteuttamiseksi vaadittavat tiedonsiirron väylät. (3.)

Rajapintojen toiminta perustuu päätepisteisiin (engl. endpoint). Päätepisteet ovat rajapintadokumentaatioissa kerrottuja URL-osoitteita (Uniform Resource Locator), joihin kehittäjät voivat ohjata HTTP-pyyntöjä käyttääkseen tarvitsemiansa resursseja. (4.)

### 3.1 HTTP-protokolla

HTTP on protokolla eli yhteiskäytäntö, joka on yksi internetin yleisimmistä ja laajalti hyväksytyistä sovellusprotokollista. HTTP on asiakkaiden ja palvelimien yhteinen kieli, joka mahdollistaa modernin verkon. (5.) Protokollan toiminta perustuu HTTP-asiakkaiden

HTTP-palvelimille lähettämiin pyyntöihin ja pyynnön vastaanottaneiden palvelimien vastauksiin. Palvelimilta saatavat vastaukset sisältävät joko asiakkaan pyytämät tiedot tai tilakoodin ja -viestin, jotka kertovat asiakkaalle, miksi pyydettyjä tietoja ei voitu toimittaa. (6.)

Eri tarkoituksiin soveltuvia HTTP-metodeja on useita erilaisia. Kaksi yleisintä metodia ovat GET- ja POST-metodit, jotka ovat myös toteutetun projektin kannalta kaksi keskeisintä metodia. GET-metodin avulla asiakas pyytää palvelimelta haluamaansa dataa, kuten esimerkiksi verkkosivuja tai tiedostoja. POST-metodin avulla asiakas voi lähettää uutta dataa palvelimelle tai päivittää olemassa olevaa dataa palvelimella. (7.)

HTTP-pyyntö ja -vastaukset ovat rakenteeltaan samankaltaisia, useille riveille hajautuvia tekstitiedostoja (8). Seuraavassa on perehdytty, miten HTTP-protokollan mukaiset pyynnöt ja vastaukset rakentuvat ja millaista tietoa ne voivat esimerkiksi sisältää.

### 3.1.1 HTTP-pyyntöjen rakenne

HTTP-pyyntöjen ensimmäinen rivi on aina yhden rivin mittainen aloitusrivi (engl. start line). Aloitusrivi rakentuu kolmesta elementistä, joista ensimmäinen elementti kertoo pyynnön metodin avulla palvelimelle mitä tehdään. Aloitusrivin toinen elementti kertoo pyynnön kohteen (engl. request target), joka on yleensä URL-osoite. Kolmas ja viimeinen elementti määrittää käytettävän HTTP-version. Lisäksi se alustaa pyynnön muuta osaa ja kertoo palvelimelle, mitä protokollaversiota vastauksessa odotetaan. (8.)

Pyynnön toisesta rivistä alkavat otsakkeet (engl. headers). Niiden avulla määritellään pyyntö ja mahdollisen runko-osan sisältö (engl. body). Otsakkeet voidaan jaotella kolmeen eri kategoriaan: koko pyyntöä koskevat yleiset otsakkeet (engl. general headers), pyyntöä määrittävät pyyntöotsakkeet (engl. request headers) ja mahdollisen rungon määrittelevät esityksen metatietojen otsakkeet (engl. representation metadata headers). (8.)

HTTP-pyyntöön kolmas elementti on aina tyhjä rivi, joka kertoo pyynnön metatietojen päättymisestä. Yleisesti ottaen HTTP-pyyntöjen neljäs elementti, runko-osa, löytyy lähinnä tietoa lähettävistä pyynnöistä. Runko-osat voidaan jaotella kahteen kategoriaan riippuen siitä, sisältävätkö ne yhden (engl. single-resource bodies) vai useamman resurssin (engl. multipart-resource bodies). (8.)

Kuvitteellisen CRM-järjestelmän rajapintaan lähetettävä HTTP-POST-pyyntö voisi näyttää yksinkertaistettuna seuraavalta.

Metodi	Kohde	Versio	
↓	↓	↓	
			POST http://exampleapp.com/api/leads HTTP/1.1
			Content-Type: application/json ← Otsakkeet
			Accept: application/json ← Tyhjä rivi
			{
			"firstname": "John",
			"lastname": "Doe" ← Runko
			}

Kuva 2. Esimerkki HTTP-pyyntöön rakenteesta.

Aloitusrivillä määritellään pyynnön metodi, kohde ja HTTP-versio. Toisella ja kolmannella rivillä ovat pyynnön otsakkeet. Content-Type-otsakkeen avulla kerrotaan palvelimelle, että pyynnön runko-osassa on JSON-objekti (JavaScript Object Notation). Accept-otsakkeen avulla asiakas informoi palvelimelle odottavansa vastausta JSON-muodossa. (9.) Pyynnön neljäs rivi on tyhjä ja sen roolina on erottaa otsakkeet ja runko. Viides rivi aloittaa pyynnön runko-osan, jossa rajapintaan välitetään liidin etu- ja sukunimen sisältävä JSON-objekti.

### 3.1.2 HTTP-vastausten rakenne

HTTP-vastausten ensimmäisestä rivistä käytetään nimeä tilarivi (engl. status line). Sen ensimmäinen elementti kertoo vastauksessa käytetyn protokollan version. Toinen ja kol-

mas elementti kuvaavat vastauksen tuloksen kolminumeroisen tilakoodin ja lyhyen tilatekstin avulla. (10.) Aiemman esimerkin mukainen pyyntö rajapintaan voisi saada aikaan kuvan 3 mukaisen vastauksen.

Versio	Tilakoodi	Tilateksti	
↓	↓	↓	
HTTP/1.1 201 Created			
Content-Type: application/json			← Otsakkeet
Date: Mon, 29 Mar 2021 09:12:31 GMT			← Tyhjä rivi
{			
	"uniqueId": "7328937729",		← Runko
	"success": true		
}			

Kuva 3. Esimerkki HTTP-vastauksen rakenteesta.

Kuvan esimerkin vastaus kertoo vastauksessa käytetyn protokollan olevan 1.1-versio ja vastaanotetun pyynnön saaneen aikaan 201 luotu -tilan. Kuvitteellinen liidin lisääminen sovellukseen on siis kuvan vastauksen mukaan onnistunut. Vastauksen toisella rivillä alkavissa otsakkeissa määritellään vastauksen sisältävän JSON-objektin runko-osassa, minkä lisäksi kolmannella rivillä on tapahtuman aikaleima. Neljäs rivi on tyhjä, se kertoo asiakkaalle vastauksen metatietojen osuuden päättymisestä. Viidennellä rivillä aloitetaan JSON-objekti, joka sisältää rajapinnan kautta lisätyn liidin uniikin tunnisteen ja vahvistaa boolean-arvon avulla lisäämisen onnistuneen.

### 3.2 Ajastettu muutosten tarkkailu

Rajapinnan avulla toteutetulla integraatiolla tulee olla keino tarkkailla muutoksia, joita valitussa päätepisteessä tapahtuu. Tällä hetkellä kaksi yleisintä työkalua muutosten tarkkailuun ovat ajastettuihin API-kutsuihin perustuva *pollaus* (engl. *polling*) ja ennalta määritettyjen tapahtumien myötä aktivoituvat *webhookit*. Näistä kahdesta pollaaminen on perinteinen tapa toteuttaa päätepisteen muutosten tarkkailu. Sen toiminta perustuu kehitettyyn algoritmiin, joka lähettää kyselyn valittuun päätepisteeseen tietyin väliajoin ja saa palvelimelta vastauksen, onko muutoksia tapahtunut. (11.)

Pollauksella on kuitenkin kaksi selkeää heikkoutta, koska se toimii aina ajastetun rutiinin mukaisesti. Siirrettävä data voi olla vanhentunutta, kun se saapuu asiakkaalle, mikä voi aiheuttaa ongelmia. Lisäksi jatkuvien kyselyiden takia molempien rajapinnan kautta kommunikoivien ohjelmistojen resursseja kulutetaan mahdollisesti jatkuvasti turhaa.

Havainnollistavana esimerkkinä pollausrutiinin heikkoudesta voidaan ajatella tilannetta, jossa urheiluverkkokaupan saldokysely tehdään kello 14:00. Tähän aikaan myyjällä on suksia 2 paria jäljellä. Kello 14:17 aktiivinen hiihtoharrastaja tilaa molemmat suksiparit, mikä laskee oikean saldon 0:aan. Seuraava saldon tarkastus tapahtuu ajastetun kyselyrutiinin mukaisesti 15:00. Vasta 43 minuutin päästä ostotapahtumasta toteutettava seuraava kysely tarkoittaa sitä, että varaston saldo verkkokaupan nettisivuilla näyttäisi väärää lukemaa kello 14:17 – 15:00 välisen ajan. Tänä aikana verkkokaupan saldo voisi siis periaatteessa painua negatiiviseksi, mikäli samaa suksiparia ostettaisiin lisää kyseisessä aikaikkunassa. Tilannetta voisi korjata tihentämällä pollausten väliä tunnin sijasta esimerkiksi 3 minuuttiin. Tällöinkin riski väärälle saldolukemalle sivuilla olisi olemassa, mutta todennäköisyys olisi paljon pienempi.

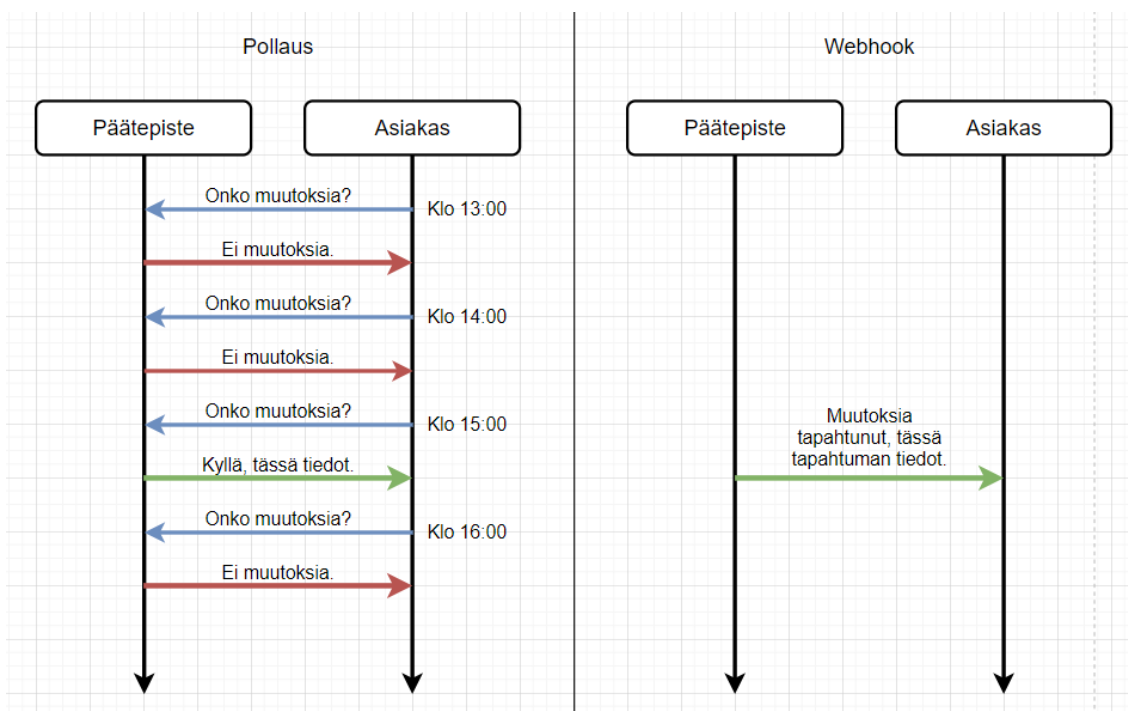
Integraatiotyökalu Zapierin vuonna 2013 tekemä tutkimus arvioi pollaamisen tehokkuutta yleisellä tasolla. Tutkimuksen mukaan 98,5 % pollauksista menee hukkaan, koska asiakkaan saamat vastaukset eivät sisällä muuttunutta tietoa. (12.) Turhat kyselyt siis vievät todella paljon resursseja tiedonsiirron molemmilta osapuolilta ja mahdollisesti aiheuttavat näin myös lisäkustannuksia käyttäjälle. Näiden tulosten valossa pollausrutiinin tihentäminen ei siis olisi välttämättäärkevin ratkaisu verkkosivuilla näytettävän varastosaldon ajantasaisuuden parantamiseksi.

### 3.3 Tapahtumaperusteinen muutosten tarkkailu

Viimeisen vuosikymmenen ajan kehittäjillä on ollut tarjolla vaihtoehto, jonka avulla voidaan välttää turhat pollaukset ja epäajantasaiset tiedonsiirrot. Webhookit ovat asiakkaan tarjoamia päätepisteitä, joihin asiakas vastaanottaa ilmoituksia haluamistaan tapahtumista reaaliajassa. (13.) Webhookeja käytettäessä ei ole tarvetta kehittää ajastettua pollausrutiinia muutosten tarkkailuun, koska muutokset seurattavassa ohjelmistossa saavat aikaan tapahtumasta ilmoittamisen asiakkaalle (14).

Webhookeja voidaan käyttää teknisesti kahdella eri tavalla. Yksinkertaisimmillaan asiakas vain vastaanottaa ilmoituksen tapahtumasta webhookiin. Tällöin asiakas eli ilmoituksen vastaanottanut osapuoli lähettää erikseen GET-pyyntöön tapahtumatietojen saamiseksi. Toinen vaihtoehto on, että webhookin osoitteeseen ohjattu POST-pyyntö sisältää tapahtuman tiedot HTTP-pyyntöön rungossa ja asiakas saa kaikki haluamansa tiedot jo ilmoituksen vastaanottamisen yhteydessä. (14.)

Tietojen vastaanottamisen jälkeen asiakas käsittelee saadun datan ennalta määritellyn logiikan avulla. Logiikka voi sisältää esimerkiksi toimintoja, joiden avulla vastaanotetut tiedot tallennetaan tietokantaan tai lähetetään sähköposti automatisoidusti. (15.) Kuvassa 4 on havainnollistettu tekniikoiden keskeisin ero, eli miten ajastettu pollausrutiini toimii verrattuna webhookien mahdollistamaan tapahtumaperusteiseen (engl. event-based) muutosten tarkkailuun.



Kuva 4. Ajastetun pollauksen ja webhookien toiminnan ero.

Kuten kuvasta käy ilmi, webhookeja käytettäessä resursseja ei tuhlaa lähettämällä kyselyjä ja vastauksia turhaan. Webhookien tehokkuus perustuu siihen, että ilmoituksia, ja

asetuksista riippuen myös tapahtumatiedot, lähetetään asiakkaalle *ainoastaan* muutosten tapahtuessa. Jo aiemmin mainitun, Zapierin tekemän tutkimuksen mukaan turha pollaus aiheuttaa keskimäärin 66-kertaisen määrän työtä palvelimelle webhookeihin verrattuna. (11.) Webhookkien mahdollistamaa tapahtumaperusteista muutosten tarkkailua hyödyntämällä voidaan säästää huomattava määrä resursseja, ja sitä kautta myös rahaa.

## 4 Integromat

Toimivan automaatiokokonaisuuden rakentamiseksi, oli tarpeellista yhdistää yrityksen verkkosivut sekä CRM- ja soittojärjestelmä samaan prosessien ketjuun. Tähän tarkoitukseen työkaluksi valittiin Integromat, joka on prosessien automatisointiin käytettävä web-pohjainen SaaS-sovellus (Software as a Service). Sen avulla on mahdollista yhdistää rajattomasti ohjelmistoja ja palveluita, minkä lisäksi sillä voidaan tarvittaessa myös muokata ja suodattaa dataa ennen tiedonsiirtoa palvelusta toiseen. Automaatioiden rakentaminen onnistuu visuaalisen käyttöliittymän kautta, mikä tekee automaatiokokonaisuuden hahmottamisesta ja suunnittelusta helpompaa.

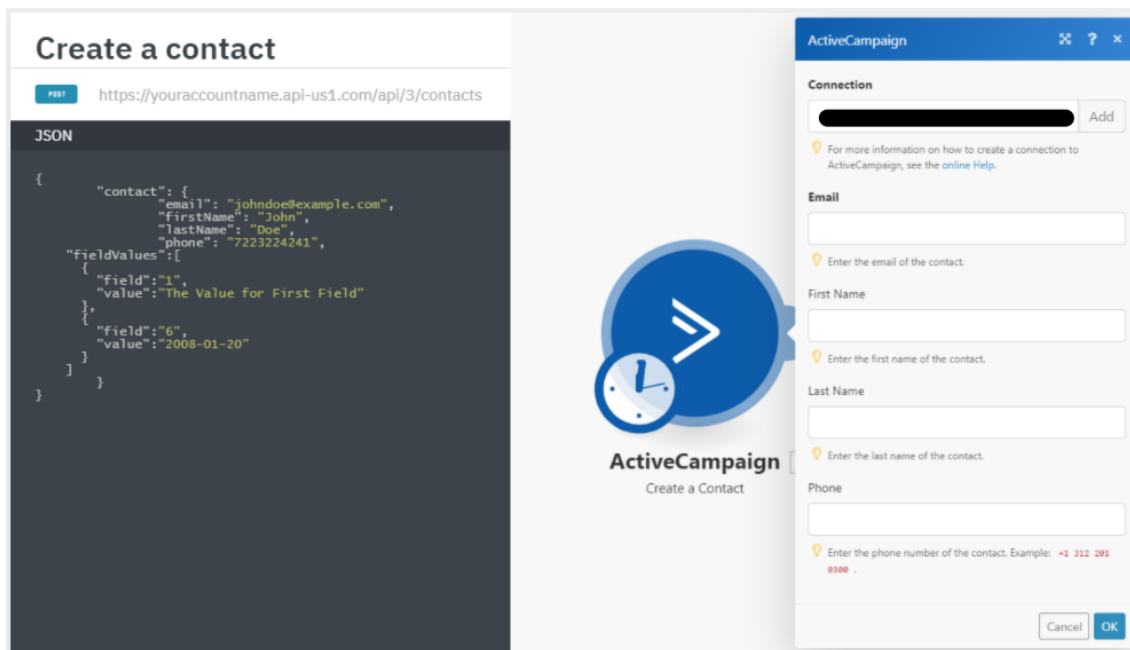
Tšekissä vuonna 2016 perustetulle palvelulle on kertynyt puolen vuosikymmenen aikana yli 224 000 käyttäjää. Sen suurimpia asiakkaita ovat muun muassa Facebook, Zalando, Adidas, Yhdysvaltojen armeija ja YK. (16.)

### 4.1 Moduulit

Integromatin visuaalinen käyttöliittymä perustuu sovellusmoduuleihin. Moduulit ovat Integromatin alustalleen kehittämiä elementtejä, joiden avulla valitun sovelluksen rajapinnan toimintoja voi hyödyntää rakennettavassa automaatiossa. Vuoden 2020 lopussa Integromatista löytyi 629 sovelluksen moduuleja ja uusia moduuleja on vuoden 2020 kolmannelta neljännekseltä lähtien lisätty valikoimaan päivittäin (16).

Kuvassa 5 on rinnastettuna kuvakaappaus CRM-järjestelmä ActiveCampaignin rajapintakuvauksesta ja Integromatista löytyvä ActiveCampaign Create a Contact -moduulista.





Kuva 5. Vasemmalla kuvakaappaus ActiveCampaignin rajapintakuvauksesta ja oikealla kuvakaappaus Integromatin ActiveCampaign Create a Contact -moduulista (17).

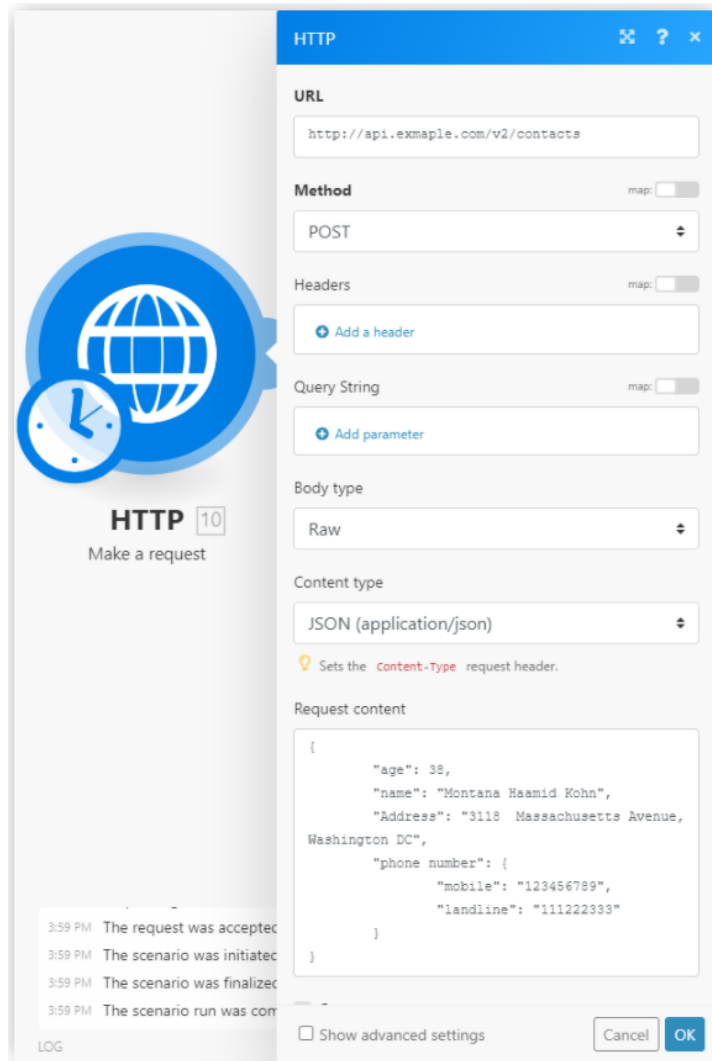
Kuten kuvasta voi nähdä ja moduulin nimestä päätellä, Integromatista löytyvä Create a Contact -moduuli on rakennettu ActiveCampaignin rajapinnan Create a contact -toiminnon mukaisesti. Molemmat yllä kuvatuista ratkaisuista mahdollistavat siis uuden kontaktin luomisen ActiveCampaigniin.

Eri järjestelmien moduulien käyttö vaatii aina yhteyden luonnin haluttuun järjestelmään. Yhteyden luominen onnistuu valitusta sovellusmoduulista löytyvän Add-painikkeen kautta. Painikkeen kautta avautuvat ohjeet kunkin moduulin ja käyttäjän ohjelmiston yhdistämiseen tarvittaviin toimiin. Kuvassa 6 on esimerkki, mitä ActiveCampaignin ja Integromatin yhdistäminen vaatii.

Kuva 6. Kuvakaappaus modaalista, joka aukeaa moduulin Add-painikkeesta.

Esimerkiksi ActiveCampaign-moduuleissa käytettävän yhteyden luontia varten yhteydelle tarvitaan nimi. Käyttäjä voi määrittää vapaasti yhteyden nimen, joka toimii jatkossa Integromatissa tunnisteena luodulle yhteydelle. Lisäksi yhteyden muodostamista varten tarvitaan ActiveCampaignista löytyvä rajapinta-avain ja ActiveCampaignin käyttäjänimi. Yhteys kuhunkin ohjelmistoon tarvitsee luoda vain kerran, koska Integromat muistaa luodut yhteydet muissa saman ohjelmiston moduuleissa.

Ymmärrettävästi Integromat ei kuitenkaan tue kaikkia markkinoilta löytyviä sovelluksia ja niiden rajapintoja. Muun muassa tästä syystä Integromatista löytyy myös vapaasti määriteltävien HTTP-pyyntöjen lähettämiseen käytettävä moduuli. Kyseisen moduulin avulla Integromat on käytännössä mahdollista yhdistää mihin tahansa rajapintaan. Kuvassa 7 on esimerkki Integromatin sivuilta, jossa määritellään JSON-objektin sisältävä HTTP-POST-pyyntö.

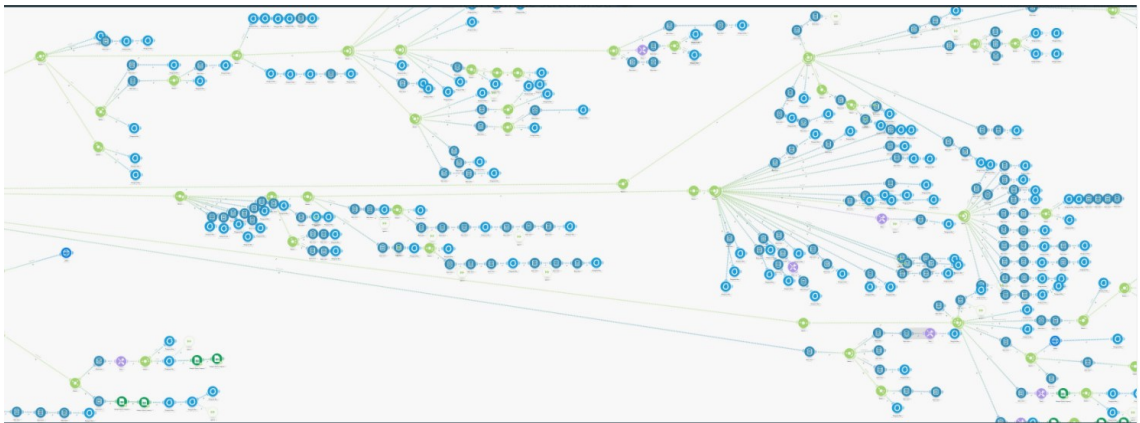


Kuva 7. Esimerkki HTTP Make a Request -moduulin kautta määriteltävästä POST-pyyntöstä (18).

HTTP Make a Request -moduuliin määritellään ensin URL, johon pyyntö kohdistetaan. Tämän jälkeen määritellään HTTP-pyyntön metodi, joka kuvan esimerkissä on tietojen lähettämiseen käytettävä POST. Lopuilla kentillä määritellään muun muassa pyynnön otsakkeiden sisältö ja mahdollinen runko-osassa siirrettävä hyötykuorma (engl. payload).

## 4.2 Skenaariot

Skenaario (engl. scenario) on Integromatin nimitys yhdelle automaatiokokonaisuudelle. Skenaariot koostuvat käyttäjän valitsemista moduuleista, jotka määrittävät, kuinka automaatiossa käsiteltäviä tietoja muokataan ja siirretään eri sovellusten ja palveluiden välillä. (19.) Yksi skenaario voi olla yksikertaisimmillaan esimerkiksi sähköpostin liitetiedoston tallentaminen Dropboxiin automatisoidusti. Skenaario voi kuitenkin koostua jopa tuhansista erilaisista yhteen kytketyistä moduuleista. Integromat ei ole rajoittanut skenaarioiden laajuutta, minkä takia sillä on mahdollista rakentaa erittäin monimutkaisiakin automaatioita. Kuvan 8 kuvakaappauksessa on Integromatin Twitter-julkaisun mukaan suurin skenaario, joka heidän alustaltaan löytyy.



Kuva 8. Yli 800 moduulista koostuva skenaario, joka ylläpitää Telegram-bottia (20).

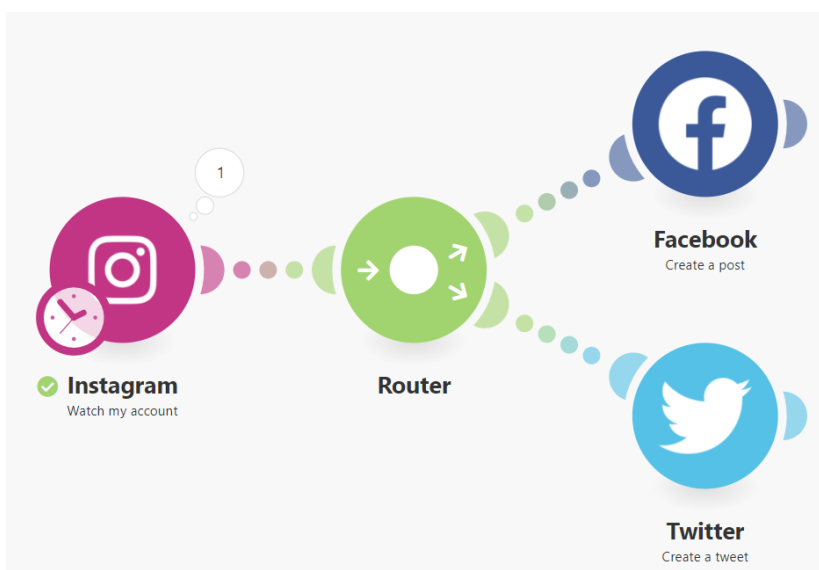
Tässä insinööriyössä toteutettu asiakashankintaputki on yksi skenaario, joka vastaa uuden liidin validoinnista sekä yhteystietojen tallennuksesta yrityksen CRM- ja soittojärjestelmiin.

### 4.3 Herätteet

Jokainen skenaario tarvitsee herätteen (engl. trigger), joka aloittaa kyseisen skenaarion sisältämien moduulien suorittamisen. Integromatista löytyy kahdenlaisia herätteitä: ajastettuja eli pollaus-tekniikalla toimivia ja tapahtumaperusteisesti aktivoituvia. Tapahtumaperusteisesti aktivoituvat herätteet voivat hyödyntää muun muassa webhookeja. (21.)

### 4.4 Reitittimet

Reitittimet (engl. router) ovat Integromatin työkalu skenaarioiden jakamiseen useampaan polkuun. Ne mahdollistavat myös saman datan prosessoinnin kussakin automaatiopolussa eri tavalla. (22.) Kuvassa 9 on Integromatin sivuilta löytyvä esimerkki, jossa Integromat on asetettu tarkkailemaan käyttäjän Instagram-tiliä. Kun tilille tehdään uusi julkaisu, Integromat erittelee julkaisun kuva- ja tekstisisällöt ja luo niistä automaattisesti myös Facebook-postauksen ja twiitin.

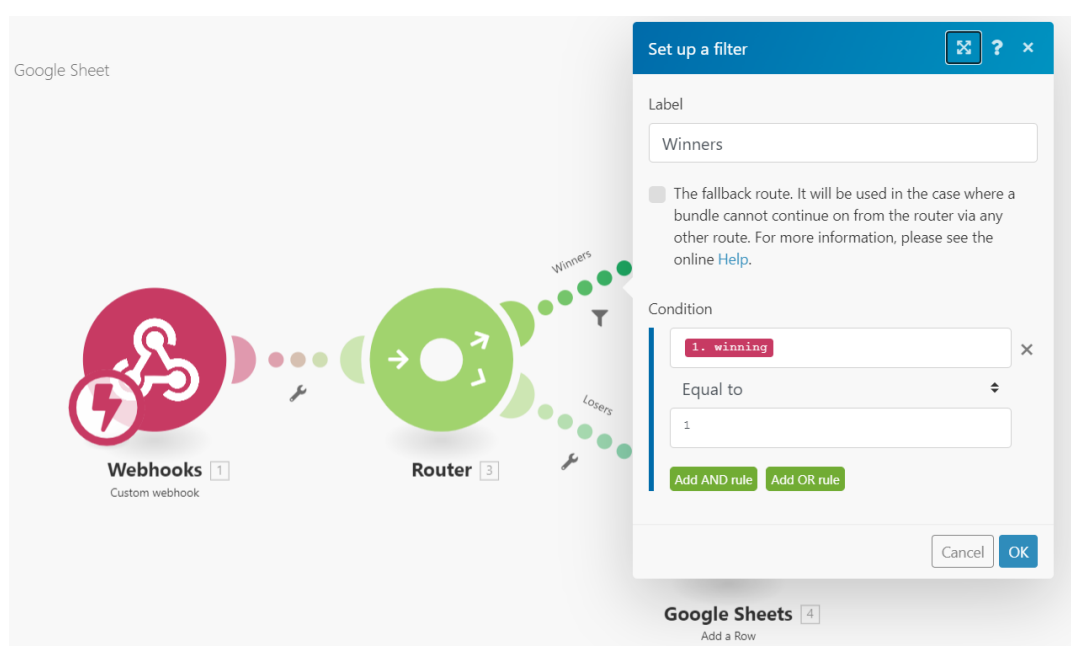


Kuva 9. Esimerkki reitittimen toiminnasta, joka vastaa julkaisun tekemisestä automatisoidusti Facebookiin ja Twitteriin (23).

Kuvan 9 mukainen automaatio säästäisi esimerkiksi yrityksen somevastaavan aikaa huomattavasti, kun hän saisi tehtyä julkaisut kaikkiin kolmeen eri kanavaan tekemällä postauksen manuaalisesti vain Instagramiin.

### Suodattavat reitittimet

Reitittimien avulla skenaarioihin voi tuoda suodatusmahdollisuuksia ja sitä kautta vaihtoehtoisia suorituspolkuja (22). Kuvassa 10 havainnollistetaan reitittimen toimintaa suodattimena.

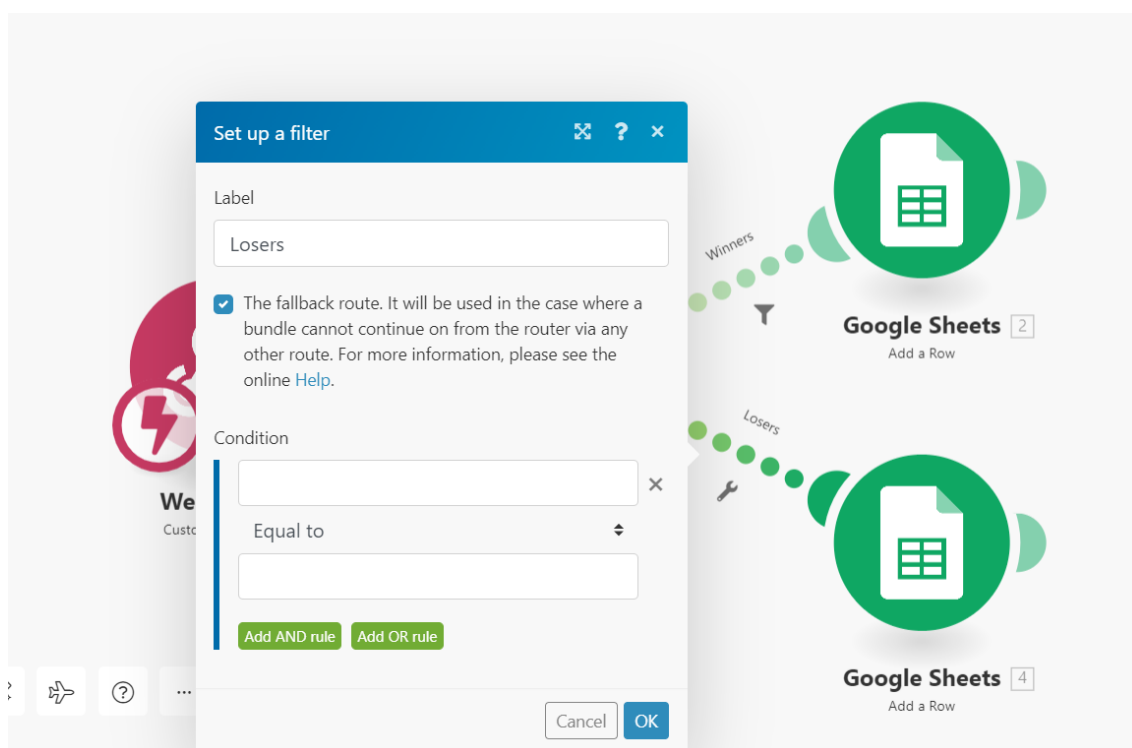


Kuva 10. Kuvakaappaus skenaariosta, jossa osallistujan tiedot viedään arpajaisten tuloksen mukaan eri Google-taulukkoon.

Kyseinen automaatio vastaanottaa arpajaisiin osallistuneen henkilön tiedot ensin Integromatissa määriteltyyn webhookiin. Tämän jälkeen tietojen vastaanottamisesta käynnistyvä skenaario aloittaa tietojen prosessoinnin. Mikäli osallistujan tiedot sisältävä JSON-objekti sisältää winning-kohdassa arvon 1, osallistujan tiedot pääsevät eteneämään automaatiopolun päässä olevaan Google Sheets Add a Row -moduuliin. Kyseinen moduuli lisää osallistujan tiedot uutena rivinä voittajien tietoja sisältävään Google-taulukkoon.

## Varareitit

Varareitit (engl. fallback route) ovat if/else -tyyppisiä suorituspolkuja, joihin edetään vain, mikäli skenaario ei voi edetä mitään toista reittiä pitkin automaation loppuun (24). Aiemmin esitelty arpajaisautomaatio on esimerkiksi rakennettu niin, että mikäli osallistujan datan sisältämän JSON-objektin winning-kohdan arvo on jotain muuta kuin 1 (käytännössä siis 0), hänen tietonsa eivät pääse etenemään skenaariossa Winners-suodattimen läpi. Tällöin osallistujan tiedot ohjataan varareittiä pitkin toiseen automaatiopolkuun, jonka päätteeksi hänen tietonsa lisätään häviäjien taulukkoon (kuva 11).



Kuva 11. Varareitin avulla toteutettu vaihtoehtoinen päätös automaatiolle.

Kuvissa 10 ja 11 esitetyn vaihtoehtoisia reittejä sisältävän skenaarion avulla saadaan kätevästi eriteltyä arpajaisissa voittaneet henkilöt muista arpajaisiin osallistuneista henkilöistä. Edellä kuvattu skenaario helpottaisi arpajaisien järjestäjää, kun kaikkien osallistuneiden joukosta ei tarvitsisi alkaa erittelemään henkilöitä, joille pitää toimittaa palkinto.

## 5 Uudistettu asiakashankintaputki

Ongelmakohtien tunnistamisen jälkeen aloitettiin asiakashankintaputken uudelleensuunnittelu. Suunnittelussa keskityttiin ensisijaisesti automatisoimaan toimialueella asuvien liidien validointi ja lisääminen soittojärjestelmän soittolistoihin. Kuten luvussa 2 kerrottiin, myös asiakasmuistiinpanojen siirtäminen Google-taulukoista asianmukaisesti järjestelmiin ja toimialueiden ylläpidon helpottaminen nivoutuivat samaan kokonaisuuteen, joten nekin huomioitiin ratkaisua luonnostellessa.

Alusta asti oli selvää, että vanha asiakashankintaputki sinänsä toimi oikein. Sitä vain täytyi täydentää. Lisättävän osan tuli pystyä käsittelemään sekä liidien validointi että tiedonsiirto ActiveCampaignin ja Herobase Outboundin rajapintoihin. Varteenotettavimmiksi ratkaisuvaihtoehtoiksi muodostuivat oman rajapinnan rakentaminen sekä visuaaliset automaatiotyökalut Zapier ja Integromat.

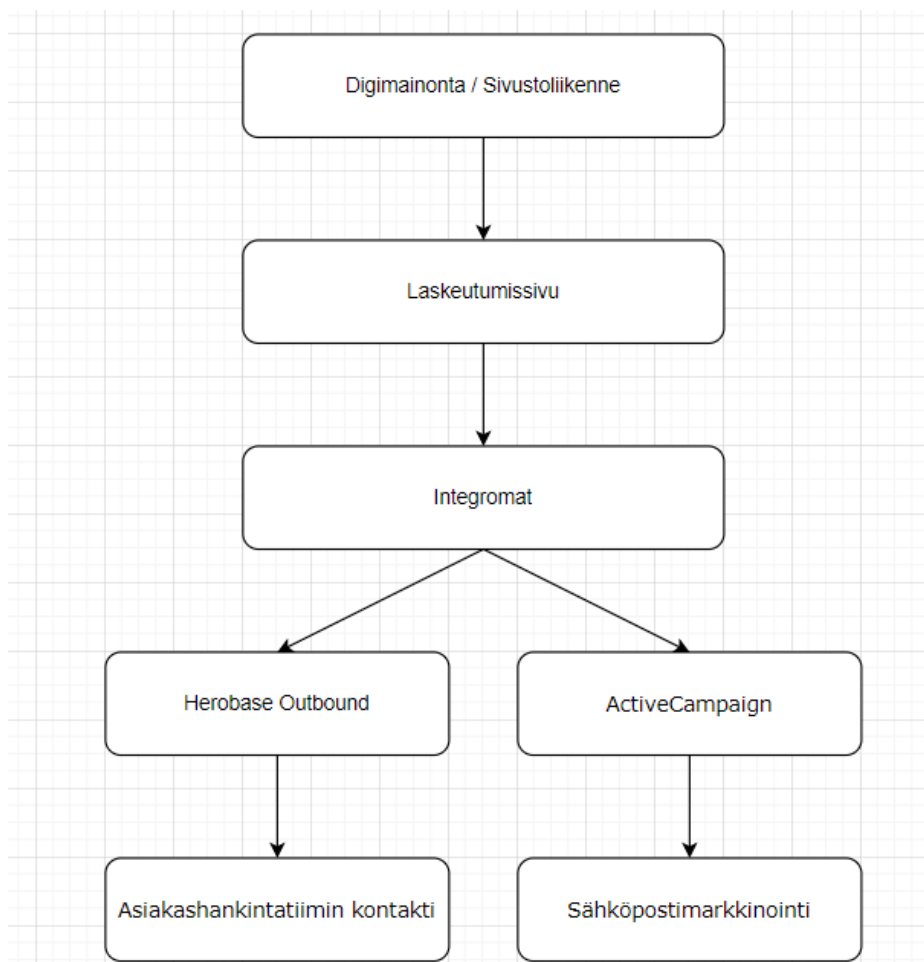
Oman rajapinnan tekeminen rajattiin kuitenkin melko nopeasti pois vaihtoehtojen joukosta, koska sen kehittäminen olisi vienyt paljon aikaa. Aikataulupaineen lisäksi visuaalista automaatiotyökalua puolsi myös se, että oman rajapinnan avulla toteutetun automaation toiminnan muuttamiseen olisi aina vaadittu teknisesti harjaantunutta henkilöä. Visuaalinen automaatiotyökalu sen sijaan mahdollistaisi muutokset joustavasti pienen perehdytyksen jälkeen kaikkien tarvittavien henkilöiden toimesta.

Automaatiotyökaluista Integromat ja Zapier tarjosivat mahdollisuuden toteuttaa postinumeroiden tarkistamiseen tarvittava datan vertailu ja sovellusten yhdistäminen samalla alustalla niin, että automaation rakentaminen ja ylläpito olisi vaivattomampaa. Ratkaisu päätettiin toteuttaa Integromatia käyttäen, koska sen kautta oli saatavilla kaikki tarvittava 1/10 hinnalla Zapieriin verrattuna. Vuositasolla pelkän automaatiotyökalun valinnalla yritys säästää useita tuhansia euroja.



## 5.1 Asiakashankintaputken uusi rakenne

Asiakashankintaputken rakenne ei siis itsessään muuttunut kovinkaan paljoa projektin aikana. Ainoa isompi muutos oli Integromatin ja Herobase Outboundin eli soittojärjestelmän lisääminen osaksi kokonaisuutta. Kuvassa 12 on esitetty asiakashankintaputken uusi rakenne.

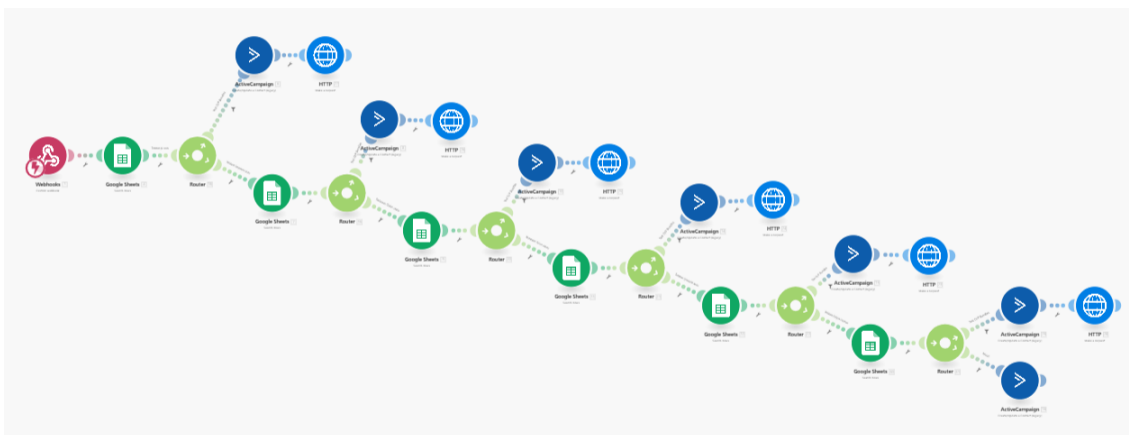


Kuva 12. Yksinkertaistettu kaavio uudesta asiakashankinnan rakenteesta.

Uusi osuus alkaa, kun Integromatissa määritelty webhook vastaanottaa laskeutumissivulla olevan lomakkeen lähettämän JSON-objektin. Tietojen vastaanottamisen jälkeen Integromat aloittaa liidin validointiin ja tallentamiseen käytettävän skenaarion suorittamisen.

Webhook valikoitui automaation herätteeksi kahdesta syystä. Webhookin avulla on mahdollista toteuttaa skenaarion suorittaminen reaaliajassa. Reaaliaikainen toiminta on tärkeää, koska liidi odottaa saavansa yhteystietojen antamisen jälkeen sähköpostin yritykseltä. Toisekseen webhookin avulla toteutettu heräte mahdollistaa matalammat ylläpitokustannukset, koska skenaario ei jatkuvasti pollaa yrityksen sivuston tietokantaa tai muuta sijaintia, jonne liidien tietoja kerättäisiin. Webhookilla toteutetun herätteen avulla säästytään noin 40 000 pollaukselta kuukaudessa, minkä ansiosta yritys säästää joka kuukausi noin 60 euroa. Vuositasolla pelkän herätevalinnan avulla saavutettava säästö on yli 700 euroa.

Kuvassa 13 on kuvattuna projektin aikana kehitetty skenaario eli automaatio kokonaisuudessaan.



Kuva 13. Kuvakaappaus kehitetystä skenaariosta, joka syntyi projektin lopputuloksena.

Kaiken kaikkiaan liidin validoinnista ja validoinnin tuloksesta riippuen yhteystietojen tallentamisesta haluttuihin järjestelmiin vastaava skenaario koostuu 20 eri sovellusmoduulista. Lisäksi siihen kuuluu 6 reititintä, joiden avulla automaation toimintaa on optimoitu.

## 5.2 Postinumeroiden vertailu

Koko skenaarion peruspilarina toimii liidien asuinalueen automaattinen validointi. Validointi toteutetaan vertaamalla liidin yhteydenottolomakkeelle ilmoittamaa postinumeroa

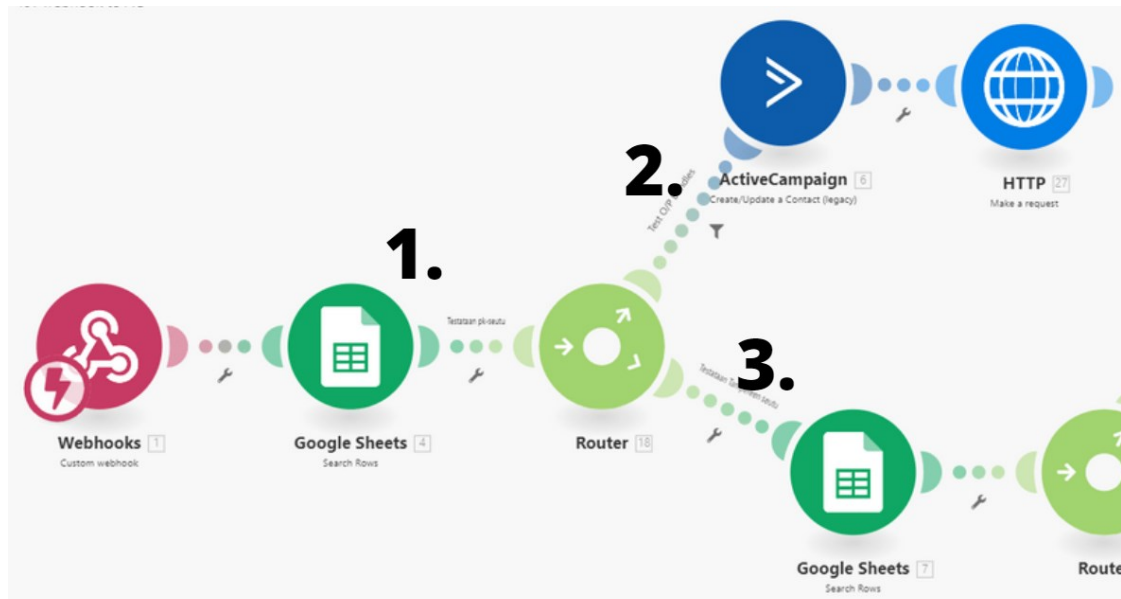
yrittäjien toimialueiden postinumeroihin. Tähän käyttötarkoitukseen postinumerot kerättiin Google Sheetsin ehdollisen muotoilun säännöistä omaan Google-työkirjaan, jossa jokaisen toimialueen postinumerot ovat lajiteltuina omiin työkirjajaksokkeisiin.

Postinumeroiden kerääminen siihen tarkoitettuun työkirjaan mahdollistaa vertailun lisäksi myös huomattavasti helpomman tavan ylläpitää postinumeroita kaikille niistä päätävillä henkilöille. Uuden postinumeron ottaminen mukaan onnistuu lisäämällä vain uusi rivi työkirjaan. Erityisesti uuden toimialueen avauksen yhteydessä yrityksellä tulee lähes väistämättä tarvetta poistaa syystä tai toisesta huonoksi todettuja postinumeroita toimialueesta. Nykyisin postinumeroiden poistaminen onnistuu poistamalla haluttu rivi työkirjasta. Skenaario osaa ottaa muutokset huomioon saman tien ja jatkaa toimintaa siitä hetkestä eteenpäin päivitetyn postinumerolistauksen mukaisesti.

Integromatilla on myös helppo suorittaa vertailu webhookin sisältämän asiakasdatan ja postinumerot sisältävän työkirjan välillä. Integromatista löytyy vertailuun sopiva Google Sheets Search Rows -moduuli. Kyseisen moduulin avulla voidaan etsiä vastaavuuksia Google-työkirjan sisältämien arvojen ja Integromatissa käsiteltävän datan arvojen välillä.

Postinumeron vertaaminen on rakennettu siten, että skenaario vertaa yhden alueen eli yhden työkirjajaksokkeen kerrallaan. Vertailussa hyödynnetään varareittejä, jotta kuluetaan minimimäärä operaatioita eli moduulien suorittamisia skenaarion sisällä. Integromatin hinnoittelu perustuu suoritettujen operaatioiden määrään, joten kulujen hallinnan kannalta on fiksuin käyttää minimimäärä operaatioita. Operaatiomäärien minimoimiseksi postinumeroiden vertailu suoritetaan kaupunkien kokojärjestyksessä: mitä isompi kaupunki, sen suurempi todennäköisyys vastaavuuden löytymiselle, ja siten vähemmälle operaatiomäärälle.

Kuvassa 14 on lähikuva skenaarion alusta, jossa suoritetaan ensimmäinen vertailu vastaanotetun JSON-objektin ja Google Sheetsissä sijaitsevan postinumerotaulukon välillä.



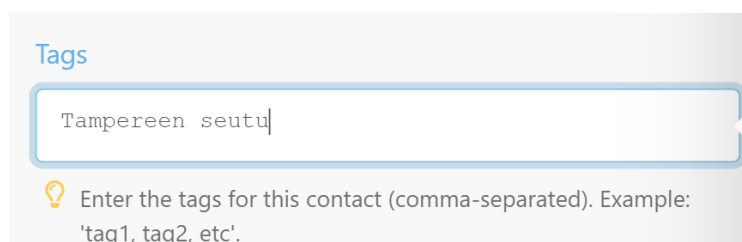
Kuva 14. Ensimmäinen postinumeroverailu, jossa etsitään vastaavuutta pääkaupunkiseudun osalta.

Kohdassa 1 skenaario vertaa liidin postinumeroa pääkaupunkiseudun toimialueen postinumeroihin käyttäen edellä mainittua Google Sheets Search Rows -moduulia. Kohtaan 2 on asetettu suodatin, joka päästää liidin tiedot ActiveCampaignin Create a Contact -moduuliin ja HTTP Make a Request -moduuliin vain, jos kohdassa 1 suoritettu vertailu on tuottanut enemmän kuin 0 tulosta. Mikäli vertailutulosten määrä on 0, edetään ensimmäiseen varareittiin (kohta 3), jossa aloitetaan sama vertailuprosessi Pirkanmaan toimialueen postinumeroiden osalta.

### 5.3 Liidin tallentaminen CRM:ään

Yritys otti kesällä 2020 käyttöön ActiveCampaign-nimisen CRM-järjestelmän eli asiakkuudenhallintajärjestelmän. Vuonna 2003 perustettu ActiveCampaign on vuosien varrella kehittynyt sähköpostimarkkinointiohjelmistosta monipuoliseksi CRM-järjestelmäksi, minkä takia kyseinen SaaS-ohjelmisto täyttää yrityksen tarpeet sekä sähköpostimarkkinoinnille että liidikannan ylläpidolle.

Jokainen automaation läpi kulkeva liidi viedään automaation loppuksi ActiveCampaigniin myöhempää käyttöä varten. Liidin asuessa toimialueella hänelle lisätään toimialueesta kertova tunniste (engl. tag). Tunnisteen lisääminen onnistuu suoraan ActiveCampaign Create a Contact -moduulin kautta, joka vastaa uuden liidin luomisesta CRM:ään (kuva 15).



Kuva 15. Lisätään Tampereen seutu -tunniste liidille, jonka postinumero on löytynyt Google-taulukosta Tampereen alueen postinumerojen joukosta.

Tunnisteen lisäämisen ansiosta jokainen toimialueella asuva liidi voidaan segmentoida tehokkaasti. Segmentoinnin ansiosta hankituille liideille voidaan kohdistaa sähköpostilla alueellista markkinointia tulevaisuudessa. Markkinoinnin alueellinen kohdistaminen parantaa jokaisen toimialueen myynnin ennustettavuutta, koska muuttuneisiin kysyntätilanteisiin voidaan puuttua nopeasti muun muassa sähköpostikampanjan avulla. Lisäksi tunnisteiden avulla on helppo tarkistaa ActiveCampaignista liidien määrä kullakin toimialueella. Näin yritys pysyy ajan tasalla siitä, kuinka paljon markkinointisatsauksia kullekin toimialueelle kannattaa tehdä.

#### 5.4 Liidin lisääminen soittojärjestelmään

Onnistuneen postinumerovertailun jälkeen automaatiopolun loppuun asti päätyvät liidit lisätään automatisoidusti myös asiakashankintatiimin käyttämään soittojärjestelmään. Jokaiselle toimialueelle on luotu soittojärjestelmään oma soittolista. Skenaariossa aiemmin suoritetun postinumerovertailun ansiosta liidit voidaan ohjata suoraan oikean toimialueen soittolistalle. Toimialuekohtaisten soittolistojen ansiosta asiakashankintatiimi pystyy tarpeen mukaan kartoittamaan halutulla toimialueella asuvien liidien palvelutarvetta tehostetusti ja paikkaamaan näin esimerkiksi peruuntuneita siivouskäyntejä.

Käytössä oleva Herobase Outbound ei ole ainakaan nykyisellään Integromatin tukema ohjelmisto. Tästä syystä yhteys Herobasen rajapinnan ja Integromatin välillä on toteutettu insinööriyön neljännessä luvussa esitellyn HTTP Make a Request -moduulin avulla.

Liidien lisääminen Herobaseen toteutetaan FIFO-periaatteella (first in first out). Käytännössä tämä tarkoittaa sitä, että liidien palvelutarve kartoitetaan kunkin kaupungin soittolistalla saapumisjärjestyksessä. Näin saadaan varmistettua, että kaikki liidit kontaktoidaan organisoidusti.

## 6 Käyttöönotto

Automaatio otettiin käyttöön ensimmäistä kertaa 16.12.2020, jolloin uusi asiakashankintakampanja julkaistiin. Käyttöönoton yhteydessä automaatio ei kuitenkaan sisältänyt vielä rajapintayhteyttä soittojärjestelmään, vaan ensimmäiset kolme kuukautta liidit lisättiin soittojärjestelmään CSV-tiedostoina asiakashankintatiimin tarpeiden mukaan. Soittojärjestelmän yhdistäminen viivästyi muusta työstä, koska yritys vaihtoi alkuvuodesta 2021 soittojärjestelmää.

Projektissa toteutettu automaatio saatiin otettua lopullisena kokonaisuutena käyttöön vuoden 2021 maaliskuun puolen välin tienoilla, kun yhteys soittojärjestelmään saatiin avattua. Tämän insinööriyön kirjoitushetken mennessä automaatio on onnistuneesti validoinut ja segmentoinut jo lähes 17 000 liidin yhteystiedot haluttuihin järjestelmiin. Myös kaikki aiemmin hankitut liidit on saatu siirrettyä käsin Google Sheets -taulukoista samoihin järjestelmiin. Näin ollen yrityksen vuosien varrella hankkima liidikanta löytyy nyt ja jatkossa kokonaisuudessaan asianmukaisista työkaluista useiden eri taulukkotiedostojen ja -rivien sijaan.

Nykyisin GDPR:n edellyttämät toimet koskien liidin henkilötietojen muuttamista tai poistamista voidaan toteuttaa tehokkaasti muutamissa minuuteissa. Vastaanotetut henkilötietojen muutos- ja poistopyynnöt voidaan käsitellä ripeästi, koska kunkin liidin tiedot kertyvät keskitetysti ainoastaan CRM:ään ja soittojärjestelmään. Nopeutuneen pyyntöjen käsittelyn lisäksi uusi toimintamalli pienentää riskiä, että pyynnön tekijälle luovutetaan väärää tietoa. Kun henkilötiedot ja muistiinpanot kertyvät organisoidusti niiden hallintaa

varten suunniteltuihin järjestelmiin, tietojen tahaton muokkaaminen ei ole enää samalla tavalla mahdollista kuin aiemmassa toimintamallissa, jossa kaikki tiedot säilytettiin taulukoissa. Automaatio on siis jo ensimmäisten kuukausien aikana edistänyt yrityksen asiakashankinnan toimintaa ja liidikannan hallintaa jokaisella ennen projektin aloittamista määritellyllä osa-alueella.

Automaatiossa on käyttöönoton jälkeen ilmennyt oikeastaan vain kaksi ongelmaa, jotka on onneksi pystytty Integromatin avulla ratkaisemaan suoraan. Seuraavassa otetaan katsaus havaittuihin ongelmiin ja niihin löytyneisiin ratkaisuihin.

#### Virheelliset syötteet

Skenaariossa ilmeni noin muutama viikko käyttöönoton jälkeen ongelma, joka ajoittain kaatoi koko automaation toiminnan. Virhelokien tutkimisen myötä ilmeni, että automaatio saattoi kaatua, mikäli jonkin moduulin suorittaminen kesken skenaarion toiminnan epäonnistui. Esimerkiksi tilanne, jossa liidi syötti sähköpostiosoitteensa perään pisteen tai jonkin muun sinne kuulumattoman merkin, saattoi aiheuttaa virhetilanteen ActiveCampaignin moduulissa. Virhetilanne taas kaatoi koko automaation.

Automaation uudelleenkäynnistäminen onnistui verrattain vaivattomasti. Ensin tuli korjata vain virheellinen syöte virhelokin kautta ja laittaa automaatio takaisin päälle. Integromat jatkoi sitten toimintaa siitä, mihin se oli jäänyt ennen skenaarion kaatumista. Ongelma oli, että skenaarion kaaduttua, yksikään liidi ei kuitenkaan liikkunut automaatiossa eteenpäin. Tästä syystä esimerkiksi ActiveCampaigniin luotu automaattinen vastaussähköposti ei välittynyt liidille reaaliaikaisesti, kuten oli tarkoitus.

Skenaarion vahtiminen ja välitön korjaaminen inhimillisten virheiden seurauksena ei kuitenkaan ollut mielekäästä työn ja vapaa-ajan tasapainon kannalta. Pienen etsinnän jälkeen Integromatista löytyi Incomplete Executions -valinta automaatiolle. Valinnan aktiivimalla skenaario tallentaa syystä tai toisesta epäonnistuneet suoritukset omaan jonoonsa, ja automaatio jatkaa muutoin toimintaa normaalisti. Integromatin pääkäyttäjä saa sähköpostilla tiedon epäonnistuneesta suorituksesta ja voi käydä automaation historiasta korjaamassa käsin virheellisen suoritteen, jotta epäonnistunutkin suorituskerta saadaan vietyä automaation loppuun asti. (25.)

## Liian nopea yhteydenotto

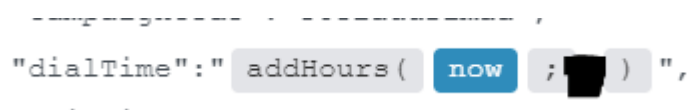
Toinen hienosäätö jouduttiin tekemään Herobase Outbound -rajapintayhteyden luonnin jälkeen. Esimerkkikoodissa on JSON-objektin rakenne, joka soittojärjestelmän rajapintaan lähetetään.

```
{
  "campaignCode": "string",
  "campaignName": "string",
  "leadPoolName": "string",
  "leadSegmentUniqueId": "string",
  "orgCode": "string",
  "status": "string",
  "dialTime": "Date",
  "priority": "int",
  "data": {
    "string": "string"
  },
  "customData": {
    "string": "string"
  },
  "dialFromTime": "TimeSpan",
  "dialToTime": "TimeSpan"
}
```

Esimerkkikoodi 1. Herobase Outboundin rajapintadokumentaatiosta löytyvä JSON-malli uuden liidin tuomiseen (26).

Ongelma oli, että mikäli esimerkkikoodista löytyvä dialTime-kohta jätettiin määrittelemättä, liidi tuli soittolistalla heti soittoon. Asiakashankinnan luonteen takia ei ollut tarkoituksenmukaista soittaa samalla sekunnilla liidille, joka oli juuri jättänyt yhteystietonsa. Tarvittiin siis ratkaisu, jolla liidille saatiin asetettua ennalta määrätty viive ennen asiakashankintatiimin yhteydenottoa, ja näin rakennettua aiemmin mainittu FIFO-järjestys.

Integromatista löytyi onneksi suoraan tarvittavat funktiot tämän ongelman korjaamiseen skenaarion sisällä. Kuvassa 16 näkyy ratkaisu, jolla muokattu ajastus saadaan toteutettua.



```
"dialTime": " addHours ( now ; [REDACTED] ) ",
```

Kuva 16. Kuvakaappaus JSON-objektiin lisätyistä valmisfunktioista.



AddHours-funktioon lisätään now-muuttujan avulla tapahtuman aikaleima ISO 8601 -muodossa. Now-arvon avulla addHours-funktio saa käyttöönsä tapahtuma-ajan UTC 0 -muodossa ja lisää siihen kokonaislukuna käyttäjän määrittelemän tuntimäärän. (27.)

## 7 Yhteenveto

Insinööriyönä toteutetun projektin tavoitteena oli parantaa asiakashankintatiimin päivittäisen työn tehokkuutta. Projektin päätavoitteeksi nousi valmiiksi validoiduista liideistä koostuvien soittolistojen muodostaminen asiakashankintatiimin soittojärjestelmään. Kyseisen tehtävän automatisoinnin yhteydessä pyrittiin vähentämään myös muita manuaalisia, aikaa vieviä ja virhealttiita tehtäviä yrityksen myyntiorganisaatiossa.

Projektin aikana kehitetty ratkaisu perustuu visuaalisen automaatiotyökalun, Integromatin, tarjoamiin toimintoihin. Integromatin avulla saatiin automatisoitua koko prosessi, joka kattaa kaiken potentiaalisten asiakkaiden validoinnista rajapinnan kautta tapahtuvaan soittolistalle lisäämiseen. Kehitetyn automaation avulla saatiin ratkaistua myös muut määrittelyn yhteydessä tunnistetut haasteet.

Projektin lopputuloksena tuotettua ratkaisua voidaan pitää tarkoituksenmukaisena. Automaation avulla on kirjoitushetken mennessä validoitu ja segmentoitu lähes 17 000 potentiaalista asiakasta. Liidien segmentoinnit niin soittojärjestelmässä kuin asiakkuudenhallintajärjestelmässä ovat auttaneet yritystä jatkamaan kasvua vallitsevasta epidemiatilanteesta huolimatta. Projektia edeltäneiden prosessien avulla nykyinen reagointinopeus virustilanteen aiheuttamiin muutoksiin, kuten kasvaneisiin siivousten peruutusmääriin, ei olisi ollut mahdollista.

Toteutettu automaatio mahdollistaa pienet muutokset jouhevasti, minkä takia siihen ei ainakaan tällä hetkellä kohdistu tulevia kehityssuunnitelmia. Sen sijaan projektin myötä hankittua tietotaitoa on tarkoitus lähteä hyödyntämään muihin myynti- ja asiakaspalvelutiimeissä havaittuihin ongelmakohtiin. Esimerkiksi tieto lahjakorttien ostosta yrityksen asiakaspalveluun välittyy nykyisellään 5 minuutin välein pollaamalla. Muuttamalla tiedon-

siirron heräte tapahtumaperusteiseksi saataisiin tieto välittymään reaaliajassa asiakaspalveluun, mikä helpottaisi asiakaspalvelun päivittäistä työtä. Lisäksi vältettäisiin yli 280 turhaa pollausta vuorokaudessa, mikä toisi vuositasolla noin 100 euron säästöt.

Koen, että projektin aikana hankittu osaaminen tulee varmasti kantamaan pitkälle myös omassa työelämässä. Perustan näkemykseni siihen, että erityisesti pk-yrityksissä turhan monet asiat tehdään yhä manuaalisesti. Tämä avaa mahdollisuuksia projektin myötä karttuneen tietotaidon soveltamiseen hyvinkin erilaisten solmukohtien ratkaisemiseen.

## Lähteet

- 1 Fonecta Finder. Verkkoaineisto. <[https://www.finder.fi/Siivouspalvelu/Koti+Puh-  
taaksi+Oy/Helsinki/yhteystiedot/2578751](https://www.finder.fi/Siivouspalvelu/Koti+Puh-<br/>taaksi+Oy/Helsinki/yhteystiedot/2578751)> Luettu 12.3.2021.
- 2 Tietosuojavaltuutetun toimisto. Usein kysyttyä EU:n tietosuoja-asetuksesta. Verk-  
koaineisto. <<https://tietosuoja.fi/gdpr>> Luettu 20.3.2021.
- 3 Arun, K; Nayagam, M Gomathy. 2014. Building Applications with Social Network-  
ing API's. Verkkoaineisto. Eswar Publications. <[https://www-proquest-  
com.ezproxy.metropo-  
lia.fi/docview/1564299028/fulltextPDF/7C0E746DADA54ADBPQ/](https://www-proquest-<br/>com.ezproxy.metropo-<br/>lia.fi/docview/1564299028/fulltextPDF/7C0E746DADA54ADBPQ/)> Luettu  
6.3.2021.
- 4 RapidAPI. Endpoint – What is an API Endpoint? Verkkoaineisto. <[https://ra-  
pidapi.com/blog/api-glossary/endpoint/](https://ra-<br/>pidapi.com/blog/api-glossary/endpoint/)> Luettu 20.3.2021.
- 5 Grigorik, Ilya. 2017. HTTP protocols. Verkkoaineisto. O'Reilly Media, Inc.  
<[https://learning.oreilly.com/library/view/http-proto-  
cols/9781492030478/ch01.html](https://learning.oreilly.com/library/view/http-proto-<br/>cols/9781492030478/ch01.html)> Luettu 4.4.2021.
- 6 Danielyan, Edgar. 2001. Understanding Internet protocols: HTTP and HTTPS.  
Verkkoaineisto. Element K Journals. <[https://search-proquest-com.ezproxy.met-  
ropolia.fi/docview/191126922/abstract/F4956E1D4B4A4F6BPQ/](https://search-proquest-com.ezproxy.met-<br/>ropolia.fi/docview/191126922/abstract/F4956E1D4B4A4F6BPQ/)> Luettu  
21.3.2021.
- 7 W3Schools. HTTP Request Methods. Verkkoaineisto.  
<[https://www.w3schools.com/tags/ref\\_httpmethods.asp](https://www.w3schools.com/tags/ref_httpmethods.asp)> Luettu 21.3.2021.
- 8 Mozilla. HTTP Messages. Verkkoaineisto. <[https://developer.mozilla.org/en-  
US/docs/Web/HTTP/Messages](https://developer.mozilla.org/en-<br/>US/docs/Web/HTTP/Messages)> Luettu 24.3.2021.
- 9 Mozilla. HTTP headers. Verkkoaineisto. <[https://developer.mozilla.org/en-  
US/docs/Web/HTTP/Headers](https://developer.mozilla.org/en-<br/>US/docs/Web/HTTP/Headers)> Luettu 24.3.2021.
- 10 Pollard, Barry. 2019. HTTP/2 in Action. Verkkoaineisto. Manning Publications.  
<[https://learning.oreilly.com/library/view/http2-in-action/9781617295164/kin-  
dle\\_split\\_011.html](https://learning.oreilly.com/library/view/http2-in-action/9781617295164/kin-<br/>dle_split_011.html)> Luettu 4.4.2021.
- 11 Ravishankar, Vivek. 2017. WEBHOOKS V.S. POLLING | YOU'RE BETTER  
THAN THIS. Verkkoaineisto. <[https://blog.cloud-elements.com/webhooks-vs-pol-  
ling-youre-better-than-this](https://blog.cloud-elements.com/webhooks-vs-pol-<br/>ling-youre-better-than-this)> Luettu 26.2.2021.

- 12 Sandoval, Kristopher. 2017. Stop Polling and Consider Using REST Hooks. Verkkoaineisto. <<https://nordicapis.com/stop-polling-and-consider-using-rest-hooks/>> Luettu 11.4.2021.
- 13 Geene, Mark. 2016. Modernize your API with webhooks. Verkkoaineisto. Infoworld Media Group. <<https://search-proquest-com.ezproxy.metropolia.fi/trade-journals/modernize-your-api-with-webhooks/docview/1822380764/se-2>> Luettu 27.2.2021.
- 14 Ram, Prashant. 2018. What is a Webhook? Verkkoaineisto. <<https://codeburst.io/what-are-webhooks-b04ec2bf9ca2>> Luettu 6.3.2021.
- 15 Requestbin. 2019. Webhooks — The Definitive Guide. Verkkoaineisto <<https://requestbin.com/blog/working-with-webhooks/>> Luettu 7.3.2021.
- 16 Integromat. 2021. The Year in Review - 2020 at Integromat. Verkkoaineisto. <<https://www.integromat.com/en/blog/integromat-2020-in-review>> Luettu 13.2.2021.
- 17 ActiveCampaign. Create a Contact. Verkkoaineisto. <<https://developers.activecampaign.com/reference#create-a-contact-new>> Luettu 22.3.2021.
- 18 Integromat. HTTP. Verkkoaineisto. <<https://support.integromat.com/hc/en-us/articles/360006248533-HTTP>> Luettu 22.3.2021.
- 19 Integromat. Creating a Scenario. Verkkoaineisto. <<https://support.integromat.com/hc/en-us/articles/360001984693-Creating-a-Scenario>> Luettu 13.2.2021.
- 20 Twitter. Verkkoaineisto. <<https://twitter.com/integromat/status/1223271456328601600/photo/1>> Luettu 12.3.2021.
- 21 Integromat. Triggers. Verkkoaineisto. <<https://www.integromat.com/en/feature/triggers>> Luettu 13.2.2021.
- 22 Integromat. Router. Verkkoaineisto. <<https://support.integromat.com/hc/en-us/articles/360001988754-Router>> Luettu 12.3.2021.
- 23 Integromat. Router. Verkkoaineisto. <<https://www.integromat.com/en/feature/router>> Luettu 12.3.2021.
- 24 Integromat. The Fallback Route Explained. Verkkoaineisto. <<https://www.integromat.com/en/academy/tutorials/the-fallback-route-explained/>> Luettu 12.3.2021.

- 25 Integromat. Incomplete executions. Verkkoaineisto. <<https://support.integromat.com/hc/en-us/articles/360001988854-Incomplete-executions>> Luettu 22.3.2021.
- 26 Herobase Outbound. Hero Outbound API. Verkkoaineisto. <[https://wshero02.herobase.com/api-docs/#!/leads/postleads\\_post\\_3](https://wshero02.herobase.com/api-docs/#!/leads/postleads_post_3)> Luettu 4.4.2021.
- 27 Integromat. Date & Time Functions. Verkkoaineisto. <<https://support.integromat.com/hc/en-us/articles/360005771793-Date-Time-Functions>> Luettu 22.3.2021.