Järjestelmäintegraatio ostoreskontran prosessissa – case yritys X

Anssi Junttila

Haaga-Helia ammattikorkeakoulu
Tradenomi (YAMK)
Liiketoiminnan teknologiat
Master-opinnäytetyö
2025

Tiivistelmä

Tekijä
Anssi Junttila

Tutkinto
Tradenomi (YAMK)

Opinnäytetyön nimi
Järjestelmäintegraatio ostoreskontran prosessissa – case yritys X

Sivu- ja liitesivumäärä
45 + 3

Tämän opinnäytetyön tapaustutkimuksella oli tarkoitus selvittää miten yritys X:llä on järjestelmäintegraatio toteutettu ostoreskontran prosessissa. Tämän lisäksi opinnäytetyössä perehdytään yleisellä tasolla järjestelmäintegraation toteutustapoihin ja integraatioarkkitehtuuriin.

Järjestelmäintegraatiolla saadaan muutoin keskenään yhteensopimattomat järjestelmät keskustelemaan ja välittämään tietoa keskenään. Järjestelmäintegraatio ei ole yksittäinen tuote, vaan keinovalikoima toimintamalleja ja tekniikoita, joilla informaatiota voidaan siirtää järjestelmästä toiseen. Järjestelmäintegraatiolla automatisoidaan manuaalisia tehtäviä informaation siirtämisessä järjestelmästä toiseen, tällä varmistetaan myös informaation eheys, sekä parannetaan tuottavuutta, koska säästynyt aika voidaan käyttää muihin tehtäviin.

Kirjallisuuskatsauksessa perehdytään erilaisiin integraatioarkkitehtuureihin ja tekniikoihin. Tarkoituksena tutkimuksessa ei ole tehdä integraatioita tai valmiita sovelluksia, vaan opinnäytetyössä pohditaan yritys X:lle sopivia tapoja toteuttaa järjestelmäintegraatio. Teoriaosuuteen on siten valikoitu yleisimpiä järjestelmäintegraation tapoja, joita tarkastellaan ylätasolla niiden etujen ja haittojen kautta, sekä kuinka ne olisivat toteutettavissa.

Tiedonhankintamenetelmänä tapaustutkimuksessa käytin puolistrukturoitua teemahaastattelua. Haastateltavina olivat yritys X:n asiantuntijat erilaisilla taustoilla, joilla oli vahva asiantuntemus ostoreskontran prosessista yrityksessä tai olivat muutoin järjestelmäintegraation asiantuntijoita yrityksessä. Haastattelut toteutettiin kahden viikon aikana Teams-etäyhteydellä, koska haastateltavat olivat eripuolilta Suomea ja tutkimustilanne olisi kaikille haastateltaville samanlainen. Haastatteluiden aineistoa analysoin dokumenttianalyysin keinoin, litteroitua aineistoa lukemalla yhä uudestaan pyrkien löytämään yhdistäviä teemoja haastateltujen vastauksista.

Tutkimuksella onnistuin selvittämään miten yritys X:n ostoreskontran integraatiot on toteutettu nykytilanteessa. Teemahaastattelulla sain hyvin tietoa ostoreskontran integraatioiden nykytilanteesta, sekä siitä miten järjestelmäintegraatio tulisi tulevaisuudessa toteuttaa. Haastattelujen tuloksia peilaten hankittuun teoriapohjaan pystyin vastaamaan asetettuihin tutkimuskysymyksiin. Opinnäytetyö onnistui annetuissa tavoitteissa, selvittämään yrityksen ostoreskontran nykytilanteen. Tutkimuksessa löysin myös kehityskohteita, kuinka integraatioita tulisi jatkossa toteuttaa, jatkokehityskohteet riippuvat paljon siitä minkälaisia valintoja yritys X:ssä tullaan tekemään järjestelmäuudistusten yhteydessä.

Asiasanat
Järjestelmäintegraatio, ohjelmistorajapinta, integraatioalusta, ostoreskontra

Sisällys

1 Johdanto

Yritys X on suuri valtakunnallinen taloushallinnon palvelukeskus. Yritys X tuottaa asiakkailleen kokonaisvaltaisen kattavasti taloushallinnonpalvelut, kuten kirjanpidon, reskontrat ja palkanlaskennan palvelut, lisäksi yritys myös vuokraa talouden ammattilaisia, kuten talouspäällikkö- ja taloussihteeripalveluja asiakkailleen.

Työskentelen yritys X:ssä ostoreskontrapalveluissa järjestelmän pääkäyttäjänä ja prosessinomistajana. Vaikka integraatiot koskevat yrityksessä kaikkia järjestelmiä ja prosesseja on tutkimus rajattu koskemaan ainoastaan ostoreskontran prosessia, jotta työn laajuus ei paisu liian suureksi.

Yrityksen nykyisissä taloushallintojärjestelmän ja ostolaskujenkäsittelyjärjestelmän versioissa ei ole ohjelmointirajapintaa, näiden järjestelmien välillä tiedonsiirto on toteutettu SFTP-tiedonsiirroilla. Yritys X:ssä on aloitettu uuden talous ERP-järjestelmän vaatimusten määrittely ja markkinakartoitus, ohjelmointirajapinta tulee olemaan vaatimuksena uuden ERP-järjestelmän kilpailutuksessa. Uuden ERP-järjestelmän käyttöönoton myötä on päätetty myös uuden ostolaskujenkäsittelyjärjestelmän version käyttöönotosta, jossa tulee olemaan käytössä ohjelmointirajapinta.

Yritys X:ssä tiedonsiirrot ostoreskontraan liittyvien järjestelmien välillä suoritetaan nykyään pääsääntöisesti SFTP-tiedonsiirroilla. SFTP-tiedonsiirrot vaativat ajastukset tiedostojen uloskirjoitukseen, sekä sisään lukuun, että tiedonsiirtoon järjestelmien välillä, tämä aiheuttaa huomattavasti viiveitä ja toisinaan tehtävät epäonnistuvat järjestelmissä. Epäonnistuneet tai viivästyneet tehtävät aiheuttavat viivettä prosessissa, mikäli esimerkiksi muuttunut laskentatunniste tai toimittajatieto ei ole siirtynyt järjestelmästä toiseen.

Opinnäytetyön tarkoituksena on selvittää ostoreskontran prosessin näkökulmasta, tutkimuskysymykset:

K1 mitä hyötyjä API-integraatioiden käytöstä olisi saavutettavissa?

K2 tulisiko uuteen ERP-järjestelmään liittyvät integraatiot sisällyttää kilpailutukseen vai tulisiko hankkia erillinen integraatioalusta?

K3 onko yrityksellä riittävästi omaa integraatio osaamista?

Kehittämistyön tarkoituksena on lisäksi lisätä omaa osaamistani järjestelmäintegraatiosta sekä tuottaa yritys X:lle lisää tietoa uuden ERP-järjestelmän hankinnan päätöksenteon tueksi vaatimusmäärittelyihin ja kilpailutukseen. Opinnäytetyössä keskitytään ostoreskontran prosessiin liittyviin integraatioihin, yritys X:n muut palvelut ja prosessit on rajattu opinnäytetyön ulkopuolelle.

1.1 Keskeiset käsitteet

API	Ohjelmointirajapinta, jonka avulla eri järjestelmät voivat jakaa tietoa (Moilanen *et al.*, 2018).
SFTP	Suojattu tiedonsiirtoprotokolla yleensä asiakas – palvelin välillä (Mikä on SFTP? - Kattava opas SFTP:hen ja sen etuihin, s.a.).
SaaS	Tuote, joka toimitetaan palveluna siten, että siihen pääsee internetin kautta helposti (IT Terminology - A Glossary of Tech Terms for Beginners | IT Career Center | CompTIA, s.a.).
Pilvipalvelu	Virtuaalinen infrastruktuuri tai palvelu, joka toteutetaan verkkoyhteyksien kautta (IT Terminology - A Glossary of Tech Terms for Beginners | IT Career Center | CompTIA, s.a.).
Legacy -järjestelmä	Perinnejärjestelmä. Järjestelmä, jonka teknologia on vanhentunutta, mutta joka on yhä käytössä (Babati, s.a.).
ESB	Viestinvälistysalusta, joka jakaa tietoa siihen liitettyjen komponenttien välillä (Churchville ja Nolle, 2021).
iPaaS	Integraatioalusta palveluna, synkroninen integraatio sovellusten ja prosessien välillä (Kuntz, 2021).
ERP	Yrityksen toiminnan ohjausjärjestelmä, suunniteltu hallitsemaan ja yhdistämän yrityksen ydinliiketoiminnan prosessit yhdeksi järjestelmäksi (Gillis, 2025).
SOA	Palvelukeskeinen arkkitehtuuri malli (Chen, 2024).

Abstract Syntax Notation One, XML ja JSON	Tietojen kuvaamiseen ja siirtämiseen käyttäviä formaatteja. Yhdistävä tekijä on se, että ne määrittelevät rakenteen, jonka avulla tietoa voidaan esittää standardoidussa muodossa eri järjestelmien välillä (Bigelow, 2023).
HTTP, FTP, Open Data Protocol ja Advanced Messaging Queuing Protocol	Ovat verkkoprotokollia, joita käytetään tiedon siirtämiseen eri tavoin. Perustuvat asiakas-palvelin-malliin sekä tukevat standardoitua tiedonsiirtoa eri järjestelmien välillä. (Bigelow, 2023).
IaaS	Integraatio palveluna, jossa palvelutuottaja rakentaa integraatiot järjestelmien ja sovellusten välille (Cole, 2022).

Opinnäytetyössä käytetään Mendeley-lähdeviittaustyökalua, tyylinä Cite Them Right 12th edition – Harvard.

2 Järjestelmäintegraation eri tavat

Järjestelmäintegraatiolla tarkoitetaan valikoimaa toimintatapoja ja teknologioita joilla toisistaan eroavat ja muutoin yhteensopimattomat tietotekniset järjestelmät on mahdollista saada toimimaan automatisoidusti keskenään. Järjestelmäintegraatio ei ole yksittäinen sovellus, vaan kuin liimaa, jolla yksittäisistä sovelluksista rakennetaan yritykselle tarkoituksenmukainen kokonaisuus. Integraatioiden yksinkertaistaminen ei ole itsetarkoitus ja riippuu aina järjestelmistä, mutta mitä vähemmän liittymiä integraatioratkaisussa tarvitaan sitä helpommaksi ja edullisemmaksi ylläpitäminen tai uuden järjestelmän integrointi tulee olemaan (Tähtinen 13-20, 2005).

Järjestelmäintegraatiolla tavoitellaan laajasti hyötyjä manuaalisen työn vähenemisestä, koska tietoja ei tarvitse syöttää manuaalisesti järjestelmästä toiseen. Automatisoinnin myötä myös inhimillisen virheen mahdollisuus pienenee, toiminta tehostuu koska pirstaloitunut tieto saadaan kerättyä yhteen paikkaan, data saadaan myös nopeammin esille liittyvissä järjestelmissä automaation myötä. Integraation suunnittelussa tulee ottaa huomioon tarpeet mitä halutaan integroida, miksi ja kartoittaa hyvin alkutilanne missä mennään tällä hetkellä, sekä kehitettävät kohteet, jotta järjestelmäintegraatio ei mene kiireessä pieleen (Alfame, 2018).

Järjestelmäintegraatio ei ole pelkästään tiedonsiirtämistä eri järjestelmien ja sovellusten välillä, vaan se on enemmänkin orkesterin johtamista, jolla pyritään oikea-aikaiseen kestävään viestin välitykseen. Hyvin suunnitellulla integraatio arkkitehtuurilla parannetaan käyttäjäkokemusta sekä sääntöjen noudattamista ja parannetaan yrityksen joustavuutta. Järjestelmäintegraatio on edennyt siiloutuneista työpöytä ratkaisuista kohti älykkäitä järjestelmiä, jotka kykenevät reaaliaikaiseen tiedonsiirtoon, joka skaalautuu ketterästi (Muthukrishnan, 2025).

Legacy-järjestelmä on vanhentunutta teknologiaa, järjestelmä tai ohjelma, joka on vanhentunut eikä vastaa nykyisiä tietojärjestelmien vaatimuksia ja odotuksia. Legacy-järjestelmä on voi olla hyvin voimakkaasti sidoksissa yrityksen liiketoimintaan ja sen vaihtaminen voi olla hyvin työlästä ja aiheuttaa merkittäviä riskejä. Legacy-järjestelmä voi olla vuosikymmeniä vanha ja niissä voi olla suuria haasteita integroinnissa muihin järjestelmiin sekä tieturvaongelmia. Legacy-järjestelmä voi aiheuttaa organisaation integraatioissa suuria haasteita, jos järjestelmässä ei ole ohjelmointirajapinta mahdollista on useiden point-to-point -integraatioiden rakentaminen ja ylläpitäminen työlästä. Järjestelmäintegraatioita on yleensä tarpeen tehdä eri järjestelmien ja sidosryhmien kesken esimerkiksi osto- ja myyntilaskut, kirjanpito, palkanlaskenta, maksuliikenne, arkisto, viranomaiset ja asiakkaiden järjestelmät (Babati, s.a.).

Järjestelmäintegraatiolla IT-kontekstissa on tavoitteena saattaa yhteen eri järjestelmiä niin, että kunkin sisältämä data tulee osaksi kattavampaa järjestelmää. Tavoitellaan mieluiten mahdollisimman

nopeaa ja helppoa tiedon siirtämistä järjestelmästä toiseen. Ohjelmistojen ja laitteiden yhdistäminen vaatii organisaatiolta usein räätälöityä arkkitehtuuria tai sovellusmäärityksiä. Järjestelmäintegraatiossa on valittavana useita eri tapoja tarpeen, käyttötyypin ja järjestelmävaatimusten mukaan. Esimerkkejä integraatiomenetelmistä ovat:

- Pystysuuntaisessa integraatiossa yrityksen erillään toimivat alijärjestelmät liitetään toimimaan keskenään luomalla siiloja järjestelmien toiminnallisuuksien perusteella. Pystysuuntaisessa integraatiossa jokainen kerros tai elementti toimii ylöspäin.
- Vaakasuuntaisessa integraatiossa määritellään alijärjestelmä, joka kommunikoi muiden alijärjestelmien kanssa. Suorien integraatioiden määrä vähenee, mikä nopeuttaa integraatioprosessia ja tuo joustavuutta sekä kustannussäästöjä
- Point-to-point -integraatio tunnetaan myös spagetti-integraationa, koska jokaisesta järjestelmästä tulee tehdä erillinen liittymä toiseen integroitavaan järjestelmään.
- Yhteinen tietomuoto auttaa integroinnissa ja edistää automaatiota. Tarkoituksena on välttää tilannetta, jossa jokainen prosessi vaatii muuntimen, jolla muutetaan tietoja muiden sovellusten muotoihin. Toteutuksessa käytetään yrityssovellusten integraatiota, joka mahdollistaa yhden järjestelmän tietomuodon muunnoksen hyväksymisen toisessa järjestelmässä. Esimerkiksi postinumeron muuntaminen kaupungin nimeksi

  (Yasar ja Ehrens, s.a.).

Järjestelmäintegraatiolla tavoitellaan automatisoinnin lisäystä ja prosessien virtaviivaistamista. Automaattinen tiedonkulku järjestelmästä toiseen helpottaa ja nopeuttaa prosessien toimintaa. Saavutettavuus tiedon synkronointi paranee, koska ei tarvitse esimerkiksi odottaa, että tieto tallennetaan manuaalisesti järjestelmästä toiseen. Datan laatu paranee, koska automaattiset integraatiot poistavat manuaalisen syöttämisen virheet ja synkronointi tuo saman aineiston liitettyihin järjestelmiin. Lisääntynyt tuottavuus tuo parempaa kannattavuutta, toisteisen manuaali työn poisto tuo työaikaa tuottavampiin tehtäviin ja lisää tuottavuutta. Useimmat integroidut järjestelmät toimivat nykyisin huomattavissa määrin pilvessä, joka tuo skaalaetuja sen sijaan, että tarvittaisiin konesalitilaa yksittäisten järjestelmien integrointiin. Kustannustehokkuutta voidaan saada myös integraatio alustasta, jolloin resursseja ei ole tarpeen käyttää yksittäisten osajärjestelmien ylläpitoon. Integroidun järjestelmät myös usein vähentävät tarpeita erillisille tietovarastoille (Yasar ja Ehrens, s.a.).

2.1 Järjestelmäintegraation haasteet

Yrityksen prosesseille kriittiset legacy-järjestelmät, joita ei voida helposti korvata uudella, aiheuttaa järjestelmäintegraatiossa haasteita. Ongelmia voi aiheuttaa yhtenäisen tai yhdistävän tietorakenteen puuttuminen, tukijärjestelmien raskas kehys sekä pelkkä järjestelmän ikä. Tiedon integraatiohaasteet haittaavat prosessien yleistä tehokkuutta, koska huono tiedonvaihto estää nopean

kommunikaation yksiköiden välillä. IT-järjestelmät ovat nopeasti sekä jatkuvasti kehittyviä ja aiheuttavat muuttuvia vaatimuksia integraatioissa, joihin tulisi kyetä vastaamaan ripeästi. Integraatioprojektit ovat aikaa vieviä ja yleensä monimutkaisia ja vaikeita toteuttaa, ketterillä työtavoilla lyhyen tähtäimen ad hoc -muutokset kehittyvät hitaasti kohti täydellisempää integraatiota. Järjestelmäintegraatio vaatii laaja-alaista osaamista teknologioista, trendeistä sekä ohjelmointiosaamista ja järjestelmätuntemusta. Osaajien puute voi aiheuttaa haasteita, pelkästään uusimmat järjestelmät ja huippu integraatioratkaisut eivät riitä vaan tarvitaan myös osaavat integraatioasiantuntijat. Järjestelmäintegraatiossa on usein mukana useita osapuolia, järjestelmätoimittajia, integraatio toimittaja sekä tilaaja, mikäli vastuut eivät ole selvillä voi ongelma tilanteiden selvittämien olla hankalaa ja tulla kalliiksi. Oikean integraatiotyökalun valinta on tärkeää, markkinoilla on tarjolla runsaasti vaihtoehtoja pilvityökaluista hybridivaihtoehtoihin. Oikean tarpeisiin ja vaatimuksiin vastaavan vaihtoehdon löytäminen ja valinta voi olla haastavaa (Yasar ja Ehrens, s.a.).

Integraatioprojektin kokonaisuuden hahmottaminen ja hyvä suunnittelu mahdollistavat myös onnistuneen toteutuksen. Mikäli integraatioiden riippuvuudet sekä resurssit eivät ole riittävän hyvin selvillä, voidaan ajautua tilanteeseen missä aikataulut venyvät ja kumppaneita ei saada tekemään sovittuja toimenpiteitä ajoissa. Kompleksisuus ja työmäärä tulisi selvittää ennen projektin aloitusta, jotta kehityshankkeen toteutus onnistuu ja pysytään budjetissa. Kehittämisen laajuus tulisi selvittää ja rajata, harvoin on mahdollista uudistaa kaikkea kerralla eikä välttämättä ole järkevää yrittää muuttaa liian isoa kokonaisuutta yhdessä projektissa (Olli, 2022).

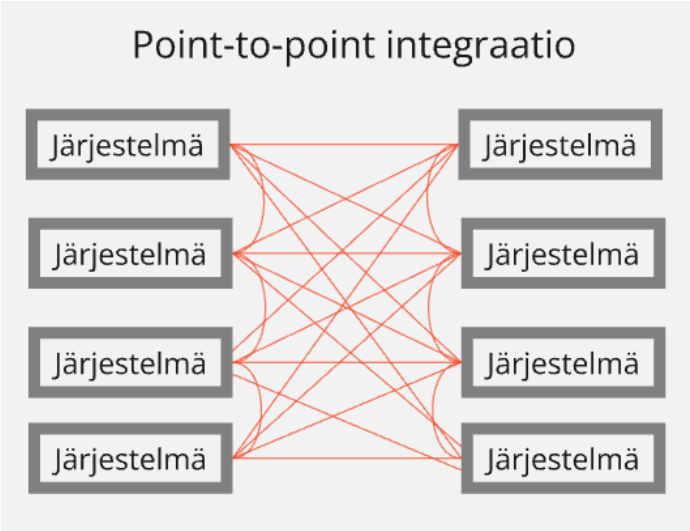
3 Integraatioarkkitehtuurit

Nykyisellä digitaalisella aikakaudella integraatioarkkitehtuurilla on suurempi merkitys kuin pelkkä tekninen strategia, järjestelmäintegraatio on välttämättömyys, jotta pystyy hyödyntämään dataansa tehokkaasti. Yrityksen integraatioarkkitehtuuri on strateginen lähestymistapa toteutettavaan järjestelmäintegraatioon, jolla toteutetaan saumaton tiedonkulku eri järjestelmien välillä. Integraatioarkkitehtuuri koostuu komponenteista, joita ovat erilaiset tavat toteuttaa järjestelmäintegraatiota, jokaisella vaihtoehdolla on vahvuutensa ja heikkoutensa, valinta riippuu aina yrityksen tarpeista sekä mahdollisista järjestelmien asettamista rajoituksista. Toteuttaminen ei ole yleensä täysin suoraviivaista vaan haasteita aiheuttavat usein legacy-järjestelmät, tietoturva, sekä useiden järjestelmien monimutkaiset integraatiot (SnapLogic, s.a.).

3.1 Point-to-point

Point-to-point -integraatioita rakennetaan yleensä tarpeen vaatiessa, jonka takia arkkitehtuuri rakentuu yleensä pitkän ajan kuluessa ja eri tekniikoilla. Tyypillisesti dokumentointi on yhtä vaihtelevaa kuin integraatioita tehneiden ohjelmoijien joukko, laajasti käytettynä point-to-point -integraatiot ovat tämän takia hankalasti hahmotettava ja ylläpidettävä integraatio ratkaisu (Tähtinen 23-30, 2005).

Kuvassa yksi havainnollistettu liittymien määrää point-to-point -integraatioratkaisussa. Point-to-point -integraatiossa data siirretään suoraan järjestelmästä toiseen, jolloin jokaisen järjestelmän välille tarvitaan omat liittymät, vaikka siirrettävä aineisto liittymien välillä olisi samaa. Yksinkertaisissa integraatioissa point-to-point, joissa samaa dataa ei tarvitse siirtää useiden järjestelmien välillä on silti toimiva ratkaisu (Tähtinen 59, 2005).



Kuva 1. Point-to-point -integraatio ratkaisu (Tähtinen 30, 2005) mukaillen.

On tilanteita, jolloin Point-to-point voi osoittautua hyvin järkeväksi tavaksi tehdä integraatioita:

- Kun integraation laajuus on matala ja sen ominaisuudet ovat yksinkertaisia, eikä integraation teko vaadi paljoa resursseja.
- Kun rakennettavana on vain yksi integraatio ja osaaminen toteutukseen on olemassa, eikä tekemiseen vaadita usein kallista ulkopuolista konsulttia.
- Kaikilla point-to-point -integraatioilla ei ole yhtäläinen vaikutus vaan liittymät vaativat priorisointia ja tulisi keskittyä integraatioihin, joista saadaan eniten hyötyä.

Point-to-point -integraatio toimii parhaiten, kun ylläpidettävä laajuus on pieni, ylläpidettävänä on vain vähän liittymiä ja näiden merkittävä vaikutus yrityksen liiketoimintaan (Griffith, 2024).

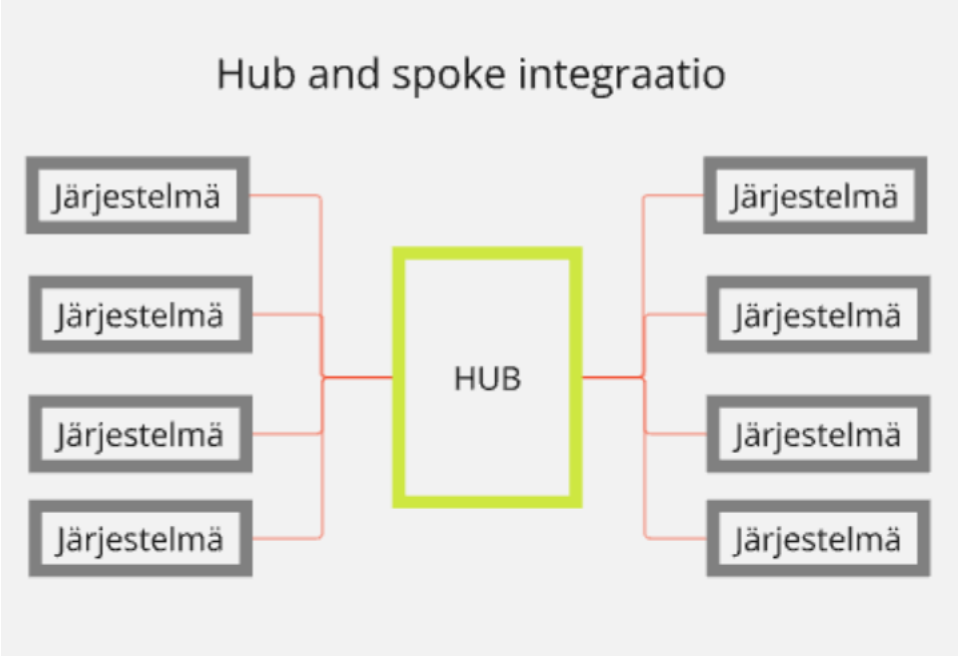
Point-to-point -integraatioiden käyttöä tulisi harkita tilanteissa kun:

- Integraation toteuttaminen vie aikaa, mikä voi viivästyttää muiden prosessien toimivuutta yrityksessä.
- Point-to-point -integraatio ei ole skaalautuva, mitä enemmän käytössä on integraatioita sitä enemmän, se vaatii työtä käyttöönotossa ja ylläpidossa.
- Tulevaisuuden muuttuvissa tarpeissa suuri määrä point-to-point -integraatioita aiheuttaa valtavan määrän ylläpitotyötä.
- Integraatio-osaamisen keskittyminen vain muutamien harteille yrityksessä luo suuren riskin, mikäli työntekijä vaihtaa tehtävää tai työpaikkaa.
- Ylläpito ja jäljitettävyys on vaikeaa ongelmatilanteissa koska keskitetty seurantatapa puuttuu. Läpinäkyvyys puuttuu ja vianetsintä voi aiheuttaa vaikeuksia.

Pisteestä pisteeseen integraatiot ovat hitaita, epäkäytännöllisiä suurissa järjestelmissä, eivät skaalaudu, eivät ole tulevaisuudenkestäviä, aiheuttavat riippuvuutta harvoista työntekijöistä ja ovat vaikeasti hallittavia. Integraatiovetoinen automaatioalusta on tehokkaampi ratkaisu (Griffith, 2024).

3.2 Hub and spoke

Hub and spoke -integraatiossa hub toimii viestinvälittäjänä kuten perinteinen puhelinkeskus eri järjestelmien ja sovellusten välillä. Järjestelmät eivät keskustele suoraan keskenään vaan kaikki yhteydet toteutetaan viestinvälittäjänä toimivan hubin kautta. Järjestelmien välissä oleva hub reitittää viestin ja tarvittaessa muunnoksen vastaanottajan tarpeen tai viestin määrittelyn perusteella. (Santos, 2016). Kuvassa 2. esitetty yksinkertaistettu malli Hub and spoke -integraatiosta, liittymien määrä verrattuna Point-to-point -integraatioon on huomattavasti pienempi.



Kuva 2. Hub and spoke -integraatio ratkaisu (Tähtinen 30, 2005) mukaillen.

Hub and spoke tarjoaa joustavan integraatioratkaisun, joka on helppo ymmärtää ja sen kanssa on yksinkertaista työskennellä. Hub and spoke -integraatiosta voidaan tehdä loputtomasti erilaisia variaatioita, integraatiotapa edistää uudelleen käyttöä ja monistamista. Kun integraatio järjestelmästä A tehdään keskittimen kautta järjestelmään B voidaan samaa liittymää monistaa myös järjestelmään C (Russom, 2008).

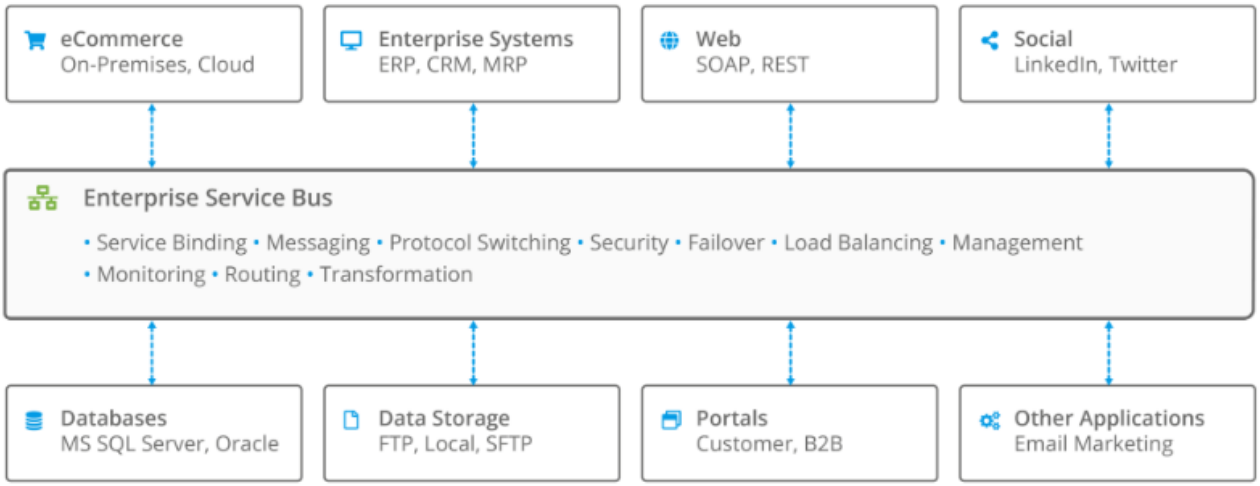
Koska Hub and spoke -integraatiossa kaikki liikenne kulkee viestinvälittäjän kautta, tulee jokaiseen integraatioon kaksi hyppyä point-to-point -integraation yhden sijaan. Viestinvälittäjä hub voi aiheuttaa viivettä viestien kulkemiseen verrattuna suoraan integraatioon ja viestinvälittäjästä voi tulla pullonkaula, koska kaikki liikenne kulkee sen kautta. Riskin voi aiheuttaa myös tilanne, jossa viestinvälittäjä vikaantuu tai on huollossa, jolloin kaikki liikenne sen kautta voi estyä (Hophe, 2003).

Hub and spoke -integraatiossa liika keskittäminen voi aiheuttaa ongelmia, kaikkia mahdollista informaatiota ei voida siirtää yhden keskittimen kautta. Liittymien määrän ja prosessien monimutkaisuuden kasvaessa hallittavuus heikkenee ja hubeja on hankalampaa hallita sekä ylläpitää ja laajentaa. Tästä syystä perinteiset Hub and spoke -integraatiot eivät skaalaudu hyvin, mikä johtaa siihen, että tarvitaan useampia hubeja ja tämän tyyppisen integraation ylläpitäminen ja kehittäminen tulee kalliiksi (O'Brien, 2008).

3.3 Palveluväylä - ESB

ESB Enterprise Servise Bus -integraatiosta käytetään nimeä palveluväylä ja se on yksi järjestelmäintegraation teknologia. Palveluväylän ensisijainen tehtävä on toimia sanomaväylänä järjestelmien välillä ja tunnetaan myös horisontaalisena integraationa. Palveluväylä ohjaa viestejä järjestelmien ja sovellusten välillä sen mukaan mikä ohjelmointi kieli kohteessa on käytössä. ESB on työkalu, jolle on käyttöä sekä hajautetussa laskennassa, että eri järjestelmien integroinnissa. Palveluväylä toimii sarjana kytkimiä, joilla voidaan ohjata viesti sovellusten välillä oikeaan paikkaan viestinsisällön ja sääntöjen perusteella (Churchville ja Nolle, 2021).

SOA Servive-Oriented Architechture suomennetaan yleensä palvelukeskeisenä arkkitehtuuri mallina. SOA -arkkitehtuurilla tavoitellaan nopeampia kehityssyklejä, helpompaa ylläpitoa, joustavuutta ja skaalaetuja. Palveluita ei ole koodattu sovelluksiin vaan julkaistu rekisteriin, josta sovellus hakee tarvittaessa uusimman version sitä tarvitessaan, tällä helpotetaan muuttuviin tarpeisiin sopeutumista ja liittyvien toimintojen integrointia (Chen, 2024). SOA on yleiskäyttöinen arkkitehtuurimalli, jossa ESB-integraatio on voi toimia teknologiana, karkeasti ESB voidaan määritellä kokonaisuudeksi teknologioita, joiden avulla eri järjestelmät voivat viestiä keskenään laitteisto ja ohjelmisto riippumattomasti (Tähtinen 143-146, 2005).



Kuva 3. ESB palveluväylä (Codeless Platforms, 2023)

Kuva 3 havainnollistaa palveluväylän toimintaa, kuvassa on useita eri järjestelmiä, sovelluksia ja palveluita integroitu toisiinsa palveluväylän avulla. ESB toimii keskitettynä viestikeskuksena sovellusten työnkulussa ja muodostaa yhteydet eri järjestelmien välillä sovittujen sääntöjen mukaa. Yleensä ESB sisältää viestimoottorin, tiedon reititys- ja muunnosominaisuuden sekä verkko- ja analytiikkaominaisuudet. Koska Legacy-järjestelmien käyttävät usein vanhentuneita protokollia ja omia dataformaatteja, on ne muutettava toimimaan SOA-verkkoprotokollien kanssa. Tästä syystä ESB on

tärkeä osa palvelukeskeistä arkkitehtuuria, sillä se mahdollistaa myös legacy-sovellusten yhdistämisen standardiprotokollilla, kuten SOAP tai JSON, joita käytetään tiedon siirtoon ja muokkaukseen (Codeless Platforms, 2023).

ESB-palveluväylällä voidaan ohjata työnkulkua ja sen avulla uusien komponenttien ja lisääminen ja käytössä olevien muuttaminen on helpompaa. Palveluväylällä voidaan valvoa tietoturvaa sekä vaatimustenmukaisuutta, se tarjoaa myös mahdollisuuden poikkeuksien ja suorituskyvyn seurantaan. ESB tarjoaa kuorman hallintaa, jossa useita eri komponentteja voidaan käynnistää suorituskyvyn parantamiseksi, sekä vikasietoisuutta, jonka avulla varmistetaan toiminnan häiriöttömyys häiriötilanteissa (Churchville ja Nolle, 2021).

Mikäli lainsäädäntö, tietoturva tai muut tekijät vaativat tietojen käsittelyn säilyvän yrityksen palvelimilla ESB-palveluväylän käyttö on hyvä ratkaisu. Palveluväylällä on yleensä tavoitteena ratkaista yrityksen kaikki integraatio tarpeet ja on kehitetty erityisesti tarpeeseen ratkaista pont-to-point -integraatioiden heikkoudet. ESB-integraatiolla on tärkeä rooli yritysten liittymien yhdistämisessä, mutta se ei kuitenkaan sovellu hyvin verkko tai pilvipohjaisiin sovelluksiin. Palveluväylän vahvuudet ovat asynkronisessa tiedonsiirrossa, joka perustuu tiedostojen kuten XML-pohjaisten sanomien siirtämiseen (Kuntz, 2021).

3.4 Integraatioalusta - iPaaS

iPaaS (integration platform as a service) on integraatioalusta pilvipohjaisena palveluna. Markkinoilla on runsaasti erilaisia tarjoajia, jotka tarjoavat integraatioalustoja erilaisilla kokoonpanoilla ja kyvykkyyksillä. Yleisesti integraatioalusta on joukko automatisoituja työkaluja, joilla yhdistetään eri ympäristöissä olevia järjestelmiä toisiinsa. Integraatioalusta sisältää yleensä valmiita liittimiä, sääntöjä sekä muunnoksi ja kytkimiä, joilla voidaan ohjata yrityksen integraatiovirtoja eri sovellusten ja järjestelmien välillä. Integraatio alusta on mahdollista hankkia joko valmiina palveluna tai niin, että yrityksellä on oma IT-henkilöstö, joka hoitaa ylläpidon, tyypillisesti integraatioalusta hankitaan kokonaisena palveluna, joka vapauttaa resurssia yrityksessä muihin tarpeisiin (Bigelow, 2023).

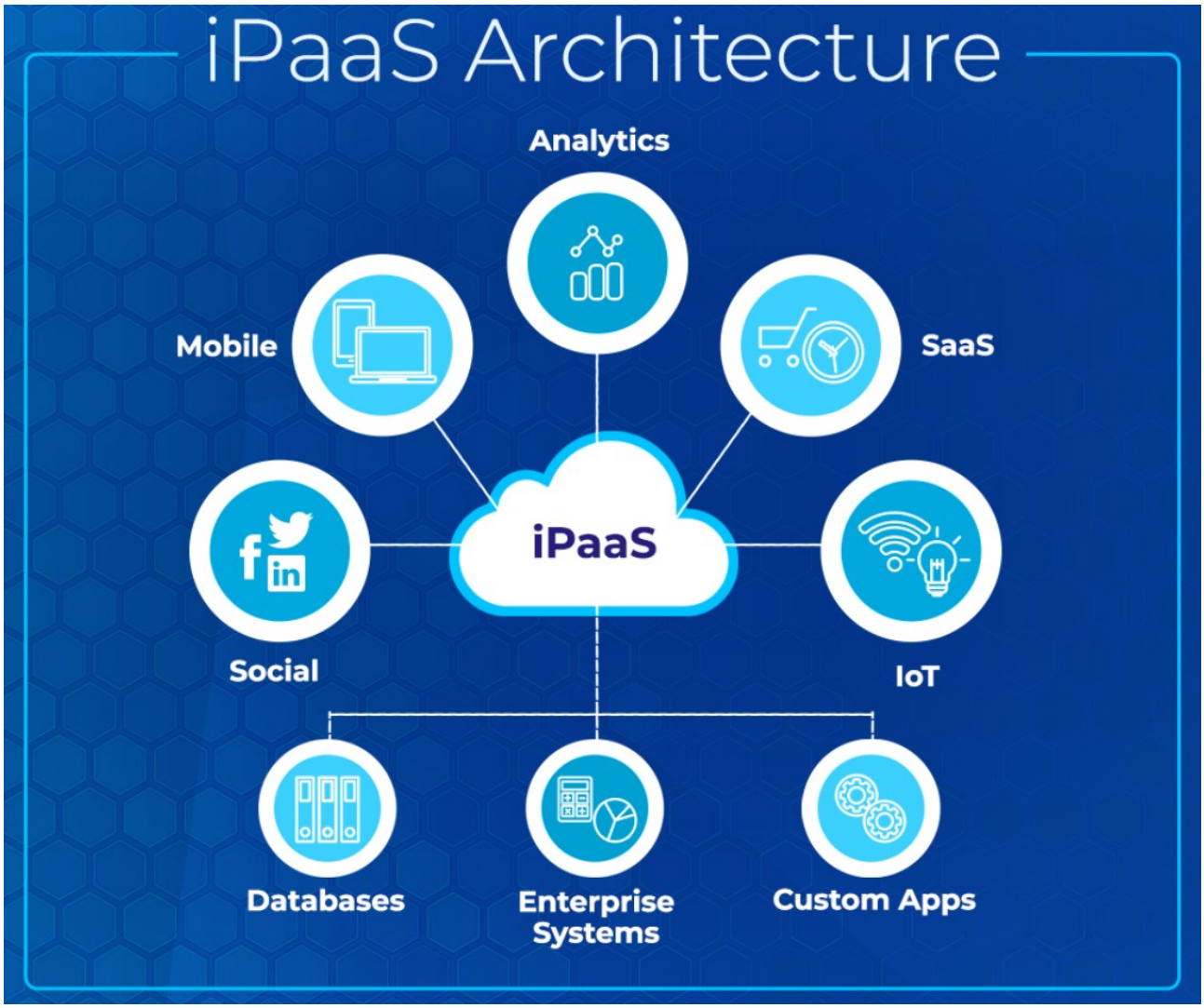
Integraatioalusta tarjoaa matalakoodi ympäristön, jossa myös ei tekniset käyttäjät voivat hallinnoida integraatioita vedä ja pudota periaatteella. Integraatio projekti alkaa tunnistamalla liitettävät järjestelmät tai tietopisteet, jotka voivat olla pilvipalveluita, asiakkaan palvelimelta toimivia legacy-järjestelmiä tai kolmannen osapuolen sovelluksia. Kun projektissa on järjestelmät yhdistetty integraatioalustaan, määritellään mahdolliset muunnokset sekä miten dataa liikutetaan järjestelmien välillä. Integraatioalusta varmistaa turvallisen tiedonsiirron, hoitaa API-hallinnan ja tarjoaa yleensä helppokäyttöisen käyttöliittymän integraatioiden reaaliaikaiseen seurantaan. Integraatioalusta on

mahdollista skaalata tarpeen mukaan, lisääntyvien integraatioiden tai kasvavan data volyymin mukaan (SRM Technologies, 2022).

Useimmat iPaaS-integraatioalustat ja työkalut tukevat keskenään samoja menetelmiä eri järjestelmien ja sovellusten yhdistämiseen, vaikka eivät sisällä kaikkia samoja integraatioita. iPaaS-integraatioalusta sisältää yleensä valmiita integraatiotyökaluja ja liittimet keskeisille tietostandardeille kuten kaupan, liikenteen, hallinnon tai terveydenhuollon sovelluksille. Valmius tukea tietoverkkoprotokollia kuten HTTP, FTP, Open Data Protocol ja Advanced Messaging Queuing Protocol sekä joustavuus mukautettujen liittimien rakentamiseen ja pääsymekanismien muokkaamiseen. Yleisesti integraatioalustat pystyvät käsittelemään, muokkaamaan ja puhdistamaan tietoa useissa muodoissa kuten Abstract Syntax Notation One, XML ja JSON. Suorituskyky säilyy ennustettavana suurissa datamäärissä sekä useissa saman aikaisissa suorituksissa, tukee reaaliaikaista käsittelyä tietojen muuntamisessa sekä eräajojen integroinnissa. Käyttäjähallinta, salaus ja kertakirjautuminen kuuluvat yleisiin ominaisuuksiin, kuten myös elinkaaren hallinta vikojen, viiveen, suorituskyvyn sekä resurssien käytön suhteen (Bigelow, 2023).

iPaaS-teknologiat tarjoavat monenlaisia hyötyjä, kuten nopeus, joustavuus, tietoturva, vianhaku ja kustannusten hallinta. Integraatioalusta tuo nopeutta ja parempaa datan saatavuutta, parhaimmillaan iPaaS kytkee yrityksen pilvipohjaiset palvelut ja sovellukset yhteen sopivaksi työkalusarjaksi. Integraatioalusta kykenee hallitsemaan lisääntyneen datamäärän ja tarjoaa mahdollisuuden reaaliaikaiseen integraatioon missä on tavoitteena minimoida yhteensopimattomien järjestelmien ongelmat. Jousta ja yksinkertainen virtuaalinen alusta yhtenäisen integraation pilvipohjaisten sekä Legacy-järjestelmien välille. iPaaS -järjestelmän sisäiset valvonta- ja uhkien havaitsemistyökalut sekä toimittajan vastuilla voidaan parantaa tietoturvaa, vastuuta ei kuitenkaan ole mahdollista siirtää tilaajan huolimattomuudesta (Bigelow, 2023).

Kuvassa 4 esitetään kuinka iPaaS toimii keskitason ratkaisuna, jonka tarkoituksena on helpottaa eri järjestelmien välistä integraatiota ilman tarvetta rakentaa erillisiä, monimutkaisia liitäntöjä erikseen jokaiseen integroitavaan järjestelmään. iPaaS mahdollistaa integraatiot pilvipohjaisten SaaS ohjelmien, paikallisten konesali sovellusten, mobiili- ja IoT-laitteiden sekä yritysten väliset integraatiot (Kuntz, 2021).



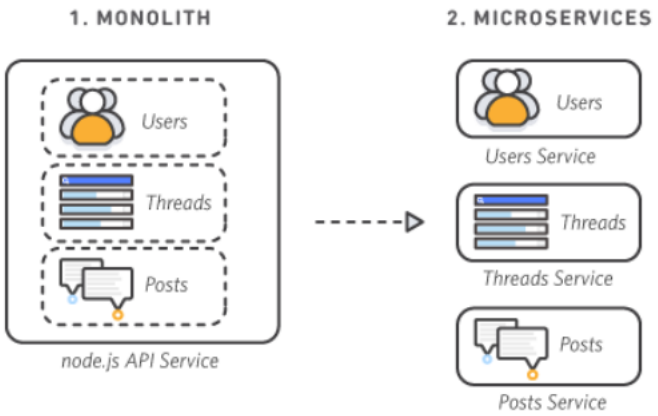
Kuva 4. iPaaS-integraatioalusta (SRM Technologies, 2022).

Integraatioalustan heikkoudet ja riskit. iPaaS-integraatioalustalla on yleisesti valmiita konnektoreita ja työkulkuja, joilla integraatioita on helppoa alkaa tekemään, ongelmia voi tulla, mikäli integraatioalustan valmiit komponentit eivät ole yritykselle sopivia. Integraatioalusta palvelut ovat tavallisesti matalakoodi tai koodivapaita sovelluksia, joissa käyttöliittymässä yhteyksiä sovelluksesta toiseen voidaan tehdä vedä ja pudota menetelmällä, tämä sopii hyvin yksinkertaisiin liittymiin, mutta monimutkaisemmissa liittymissä voidaan tarvita paljonkin ohjelmointia mikä voi tuoda yllättäviä kuluja, mikäli yritykseltä ei löydy osaajaa omasta takaa. Integraatioalusta on skaalautuva tiettyyn rajaan saakka, mikäli integroitavia järjestelmiä on paljon ja järjestelmissä on erilaiset sovelluskohtaiset liittymät, tietomuodot ja integrointimenetelmät on näistä haastavaa saada kokoon kokonaisvaltaista ratkaisua. Integraation hinnoittelurakenne voi vaihdella kuukausimaksullisesta, integraatio tai viestiperusteiseen tämä voi tehdä eri palveluiden vertailun vaikeaksi. Käyttöönoton kulut voivat nousta korkeiksi riippuen kuinka monimutkaisia integraatiot ovat ja kuinka paljon voidaan hyödyntää alusta

valmiita komponentteja. Alustan vaihtaminen voi olla vaikeaa, integraatioiden siirtäminen ei yleensä ole mahdollista tai taloudellisesti järkevää. (Raatikainen, 2025)

3.5 Mikropalvelu- vs monoliittinen arkkitehtuuri

Erityisesti mikropalvelu- ja monoliittinen arkkitehtuuri eroavat toisistaan rakenteeltaan, skaalautuvuudeltaan ja kehitysmalliltaan. Monoliittisessa ratkaisussa on keskitetty rakenne missä sovelluksen kaikki osat kuten käyttöliittymä, logiikka ja tietokannat ovat yhdessä kokonaisuudessa. Komponentit ovat tiukasti keskenään riippuvaisia, jossa yhden muokkaaminen vaikuttaa koko järjestelmään. Skaalautuvuudessa on ongelmia, koska vain osaa järjestelmästä ei voida skaalata vaan koko järjestelmä tulee päivittää ja skaalata kerralla. Monoliittisen palvelun käyttöönotto on yleensä yksinkertaisempaa ja helpompaa pienemällä projektilla. Mikropalveluarkkitehtuurissa on hajautettu rakenne, jossa sovellus on jaettu pienempiin itsenäisiin osiin, jotka suorittavat omaa tehtäväänsä ja kommunikoivat APIn välityksellä. Itsenäinen rakenne mahdollistaa sujuvamman päivityksen ja lisää vikasietoisuutta. Mikropalvelurakenne voi olla käyttöönotoltaan monimutkaisempi, komponenttien välinen viestintä, verkonhallinta ja tietoturva voivat aiheuttaa ongelmia (GeeksforGeeks, 2025). Kuva 5 havainnollistaa kuinka mikropalvelu- monoliittinen rakenne eroavat toisistaan



Kuva 5. Monoliittinen ja mikropalveluarkkitehtuuri (Mitä ovat mikropalvelut? | AWS, s.a.)

Mikropalveluissa jokainen komponentti toimii itsenäisesti ja voidaan kehittää, skaalata ja ottaa käyttöön ilman, että se vaikuttaa muihin palveluihin. Tämä tuo joustavuutta ylläpidossa ja kehittämisessä, koska voidaan muokata vain yhtä palasta kokonaisuudesta ilman että koko palvelurakennetta tuli kehittää, näin kehitys syklit saadaan lyhyemmiksi. Jokainen komponentti mikropalveluissa voidaan kehittää itsenäisesti, näin myös tiimit voidaan mitoittaa paremmin tarpeisiin oikean kokoisina. Jokainen palvelu on mahdollista kehittää tarpeeseen parhaiten sopivalla ohjelmalla tai työkalulla, joten kompromisseja ei ole välttämätön tehdä. Mahdollistaa koodin uudelleen käytön, ohjelmiston

jakaminen pieniin moduuleihin, jossa tietylle palvelulle kirjoitettua toimintoa voidaan käyttää myös rakennuspalana toiselle palvelulle. Mikropalvelurakenne tuo vikasietoisuutta, mikäli jokin palvelu epäonnistuu tai on pois käytöstä, se voi näkyä hitautena tai niin että jokin ominaisuus ei ole käytössä toisin kuin monoliittisessa rakenteessa vastaava voi aiheuttaa koko sovelluksen epäonnistumisen (Mitä ovat mikropalvelut? | AWS, s.a.).

Myös mikropalveluarkkitehtuurissa on heikkouksia, koska arkkitehtuuri on monimutkaisempi monoliittiseen verrattuna voi kehitystahti kokonaisuutena olla hitaampaa. Mikropalvelu voi vaatia uuden organisaatiotason, koska kehitystiimit voivat tehdä työtään omissa siiloissaan eivätkä välttämättä keskustele keskenään. Vian haku muodostuu mikropalveluissa haastavaksi johtuen sen monitahoisesta rakenteesta, yhteisen ohjelmointikielen puutteesta, kirjautumiskäytännöistä sekä monitoroinnista. Hajautettu organisaatiorakenne voi johtaa myös tilanteeseen, jossa ei ole selvää kenellä on omistajuus kokonaisuudesta, joka voi johtaa johtajuuden puutteeseen (Harris Chandler, s.a.).

Monoliittisen arkkitehtuurin etuja ovat käyttöönoton helppous, johtuen yksinkertaisemmasta kokonaisuudesta. Ohjelmointi on helpompaa kehityksessä ja muutos tarpeissa, kun käytössä on vain yksi ohjelmointikieli. Tämä tuo myös suorituskykyä, kun käytössä on vain yksi API verrattuna mikropalveluihin, joissa on käytössä useita eri API-liittymiä. Myös vianhaku helpottuu, kun koodi on yhdessä kokonaisuudessa, ongelman löytäminen käy nopeammin. Vastaavasti monoliittisen rakenteen heikkouksia varsinkin suuremmissa sovellus kokonaisuuksissa on kehitystyön monimutkaisuus ja hitaus. Toisin kuin mikropalveluissa yksittäiset komponentit eivät skaalaudu ja yhden moduulin virhetilanne voi kaataa koko sovelluksen. Valittu ohjelmointi kieli voi aiheuttaa teknologisia esteitä ja muutosten tekeminen voi olla mahdotonta tai hyvin kallista, monoliittinen sovellus on rajoitettu käyttämään valittua teknologiaa, tämä tarkoittaa, että pienetkin muutokset joissain komponenteissa vaatii muutoksia koko sovellukseen (Harris Chandler, s.a.).

3.6 Ohjelmointirajapinta – API

API on englanninkielinen lyhenne sanoista Application Programmable Interface, suomeksi tämä tarkoittaa ohjelmointirajapintaa. API voidaan yksinkertaisimmillaan käsittää rajapintana ohjelmassa tai järjestelmässä, joka sallii sovellusten keskustella keskenään ja mahdollistaa muun muassa pääsyn lukemaan tietoa tai kirjoittamaan tietoa järjestelmän tietokantaan. API voidaan nähdä myös tuotteena joka esimerkiksi kyselyllä järjestelmässä A palauttaa tietoa tietokannasta B. APIt mahdollistavat sujuvan tiedonsiirron järjestelmistä ja sovelluksista toisiin (Moilanen et al.,51–57 2018).

APIt vaikuttavat läpi internetin ja mahdollistavat erilaisten mobiilisovellusten toimimisen kuljettamalla tietoa sujuvasti paikasta toiseen. IoT *Internet of Things* eli esineiden internet rakentuu myös APIen varaan, joka tuo pienillä kustannuksilla yksinkertaisen tavan liikuttaa dataa fyysisen ympäristömme

laitteisiin. Taustalla toimivat APIt mahdollistavat esimerkiksi matkapuhelinten sovellusten saumattoman toimivuuden reaaliaikaisella tiedonsiirrolla. APIt yhdistävät järjestelmät suoraan toisiinsa mahdollistaen niiden keskustella keskenään, tämä lisää käyttäjäystävällisyyttä sekä tuo kustannustehokkuutta. Ohjelmistorajapinta tuo monia mahdollisuuksia automaatioon, API voidaan ohjelmoida tekemään toistuvia aikaa vieviä tehtäviä, jotta ihmisisellä jää enemmän aikaa käytettäväksi haastavampiin tehtäviin. API voi lisätä tietoturvaa tuomalla lisäkerroksen luvatonta järjestelmään tunkeutumista vastaan vaatimalla tunnistuksen kaikille kyselyille (Postman, s.a.).

API-integraatiot tuovat merkittäviä hyötyjä tiedonsiirtoon sovellusten välillä. Ohjelmistorajapinta mahdollistaa reaaliaikaisen tiedonsiirron, tuo lisää tehokkuutta automatisoiduilla tehtävillä vähentäen manuaalista työtä. Liittymien tekeminen onnistuu ilman kallista räätälöintiä käyttämällä olemassa olevia järjestelmiä, mahdollistaa pääsyn kolmannen osapuolen järjestelmiin esim. asiakkuudenhallinta. API tuo skaalaetuja yritysten voidessa laajentaa toimintoja ja lisätä ominaisuuksia ilman järjestelmän täydellistä uudistamista. API-integraatiot vaativat vankkaa teknistä osaamista ja käyttöönotot voivat olla monimutkaisia johtuen yhteensopivuusongelmista tai puutteellisesta dokumentaatiosta. Ohjelmistorajapintojen heikko todennus ja salaus voivat altistaa tietoturvariskeille ja tätä pidetään yleisesti API-integraatioiden suurimpana haasteena. APIt vaativat säännöllisiä päivityksiä, joiden muutoksista voi tulla toiminnallisia ongelmia, suorituskyky on riippuvaista ulkoisista palvelimista ja tekniset ongelmat voivat hidastaa sovelluksen toimintaa. APIen haasteisiin voidaan ennakolta varautua valitsemalla hyvin dokumentoitu ja turvallinen API, ottamalla käyttöön vahvat todennusmekanismit, jotka estävät luvatonta käyttöä ja suojaavat arkaluonteisia tietoja. API-yhteyksien säännöllinen seuranta ja päivitys, kaikki liittymät muuttuvat ajan myötä, mutta säännöllisellä valvonnalla voidaan havaita tietoturva-aukkoja ja estä suorituskyky ongelmia (Vamenture, 2025).

API-yhdyskäytävä (API gateway) on hallintatyökalu, joka on asiakkaan ja taustasovellusten välissä oleva joukko määritelmiä ja protokollia sovellusohjelmistojen integrointiin sekä rakentamiseen. Yhdyskäytävä toimii välityspalvelimena, joka hyväksyy APIen kutsut kokoaa niiden täyttämiseen tarvittavat palvelut ja palauttaa asianmukaisen tuloksen. API -yhdyskäytävä toimii ohjelmistona, joka sieppaa käyttäjän API kutsut ja välittää ne oikeaan taustapalveluun. API-yhdyskäytävä on tapa irrottaa asiakasliittymä backend-toteutuksesta jolloin, kun asiakas tekee API kutsun yhdyskäytävä jakaa sen useiksi pyynnöiksi ja välittää oikeisiin paikkoihin ja tuottaa oikean vastauksen pitäen kirjaa kaikesta (Red Hat, 2019a).

API management, eli API-hallinnalla viitataan prosesseihin, joilla on tarkoitus jakaa, ohjata ja analysoida ohjelmistorajapintoja millä yhdistetään yrityksen järjestelmiä ja ohjelmistoja pilvipalvelun välityksellä. API-hallinnalla tarjotaan sovellusliittymiä luoville tai käyttäville organisaatioille mahdollisuus varmistaa, että APIa käyttävien sovelluskehittäjien tarpeet täyttyvät. HTTP -pohjaiset ja REST APIt

ovat keskeisessä roolissa, kun yhdistetään mikropalveluja keskenään. API-hallinnalla varmistetaan tietoturvan ja sääntöjen noudattaminen organisaatiossa. API-hallinta työkalut mahdollistavat ohjelmistorajapintojen käytössä tietoturvan, valvonnan ja integroinnin. API-hallinta sisältää myös API-yhdyskäytävän mahdollistaa laajennettavuuden muihin järjestelmiin, kehittäjäportaalin sekä elinkaaren-, pääsynhallinnan ja analytiikan työkalut (Red Hat, 2019b).

3.7 Integraatio palveluna – IaaS

IaaS (Integration as a service) tarkoittaa integraatiota palveluna, jossa palvelutuottaja rakentaa integraatiot järjestelmien ja sovellusten kesken. Integraatio palveluna toimii pilvipalveluna ikään kuin SaaS -palvelun laajennuksena ja integraatiot ostetaan usein kuukausittaisena tai käytön mukaan veloitettavana palveluna. Kuten SaaS myös IaaS-palvelun tuottajat tarjoavat käyttöliittymän, infrastruktuurin eli palvelinten ylläpidon. Integraatio palveluna voi keventää yrityksen tarvetta palkata integraatioiden rakentamiseen, ylläpitoon ja hallintaan tarvittavia osaajia. IaaS-toimittajalta voidaan myös odottaa skaalautuvuutta, ketteryyttä toteutuksissa sekä laajempaa mahdollisuutta automatisoida prosesseja (Cole, 2022).

IaaS-palvelun avulla voidaan ulkoistaa joko integraatioiden rakentaminen, suunnittelusta ylläpitoon kokonaan tai osittain sopimuskumppanille. IaaS-palveluntuottajat lupaavat yleisesti nopeaa käyttöönottoa ja se onkin yleensä heidän ydinlupauksiaan, tähän tietysti vaikuttaa millaista alustaa integraatiotoimittaja käyttää ja kuinka haastavia tehtävät liittymät ovat. Toinen merkittävä asia on, ettei tilaaja tarvitse ohjelmointi- tai teknistä osaamista, koska palveluntuottaja hoitaa kaikki integraatiot, edellytyksenä tietysti on, että määrittelyt on tehty huolellisesti, jotta tilaaja saa integraatio toiminnot haluaminaan. Integraatiot ovat IaaS-palveluntuottajalle ydin liiketoimintaa, joten palvelut skaalautuvat hyvin, koska heidän tulee varmistua integraatioiden toimivuudesta myös tulevaisuudessa ja varautua myös kasvavaan liikennemäärään. Kustannukset vaihtelevat käyttöönoton kestosta, resursoinnista, määrittelystä ja integraatioiden toiminnallisuuksista, ohjelmointikielestä ja käytettävästä alustasta sekä ylläpidon kuluista. IaaS veloitus on yleisesti kuukausiveloitus kertakustannusten sijaan (Raatikainen, 2025).

IaaS ja iPaaS eroavat toisistaan merkittävästi. Valittaessa integraatio alustan, toimittaa tuottaja ainoastaan alustan, jonka avulla tilaaja itse tekee ja ylläpitää kaikki tarvitsemansa integraatiot. Mikäli integraatio tarpeet ovat pääsääntöisesti sisäisiä ja yrityksellä on tarvittava tekninen osaaminen integraatioiden tekemiseen iPaaS on varteenotettava vaihtoehto. Integraatiossa palveluna toimittaja huolehtii integraatioista kokonaisena palvelupakettina, jolloin tilaajan ei tarvitse itse ylläpitää liittymiä, palvelin infrastruktuuria tai teknistä osaamista integraatioista (Cole, 2022).

IaaS on pilvipohjainen toimitusmalli, jolla voidaan yhdistää konesali ja pilvipohjaiset palvelut toisiinsa, palveluntarjoaja tuottaa järjestelmäintegraation palveluna, jolloin asiakas voi keskittyä kehittämään prosessiaan. IaaS veloitus riippuu tyypillisesti integraatioiden ja liittymien määristä, joten suuremmalla liikennemäärällä, myös veloitus on suurempi, tilaajan skaalaaminen ympäristössä ylös tai alaspäin vaikuttaa yleensä myös suoraan palvelun veloitukseen. Tyypillisesti IaaS-malli integraatioiden toteutuksessa on käytössä pienissä ja keskisuurissa yrityksissä, jolloin resursseja jää yrityksessä käyttöön tärkeämpiin kohteisiin. Suurimmat hyödyt saadaan laadukkaasta arkkitehtuuri rakenteesta, eikä tilaajan tarvitse investoida konesali infraan tai huolehtia laitteisto päivityksistä. IaaS käyttöönotto kustannukset jäävät matalammiksi, koska ei tarvetta infralle, osaamiselle tai ylläpidolle vaan veloitus tulee vain siitä mitä käytetään (McDonald, 2023).

IaaS-integraatio on avaimet käteen ratkaisu, jossa tarjotaan valmis ratkaisu pisteestä pisteeseen ja tavoitellaan ketterää tapaa järjestää yrityksen integraatiot. Tilaajan tulee arvioida yrityksen IT-infrastruktuuri ennen käyttöönottoa, mitä järjestelmiä ja sovelluksia on mahdollista ja järkevää integroida IaaS-palvelulla ja mitkä esimerkiksi tietoturvasyistä olisi järkevämpää järjestää toisella tavalla. Toisin kuin perinteisissä integraatioratkaisuissa IaaS ratkaisulla vältytään merkittävistä alkuinvestoinneista koska veloitus tulee yleensä kulutuksen mukaisesti, mikä mahdollistaa myös skaalautumisen ylös ja alas kulloisenkin tarpeen mukaan, jolloin veloitus tulee sen hetkisen käytön mukaisesti. IaaS tarjoaa ratkaisun reaaliaikaiseen tiedonsiirtoon, joka mahdollistaa myös erilaiset tietomuunnokset palvelujen ja sovellusten välillä tarpeen mukaan (Rose-Collins, 2024).

3.8 Ohjelmistorobotiikka - RPA

Ohjelmistorobotiikalle RPA (robotic process automation) voidaan siirtää ihmiseltä suuri volyymiset toisteiset tehtävät, ohjelmistorobotti matkii ihmisen työskentelyä ohjelman käyttöliittymässä. Yksinkertaisimmillään robotti voi olla tehty nauhoittamalla ihmisen työskentely käyttöliittymässä ja tarvittaessa robottia voidaan muokata ohjelmistorobotiikan työkalulla. Kehittyneemmät robotit käyttävät myös konenäköä tunnistaakseen näytöltä ikoneita, tekstiä ja muokatakseen toimintaa sen mukaan. Ohjelmistorobotti tekee sen mitä sen on ohjeilmoitu tekemään, joten sen toimivuus on paljolti kiinni siitä, kuinka hyvin se on määritelty. Mikäli ohjelman käyttöliittymään tulee muutoksia, se vaatii muutoksia myös robotille, koska muutokset voivat pysäyttää robotin työskentelyn. Ohjelmistorobottia voidaan suorittaa valvotusti tai ilman valvontaa, valvotut robotit käynnistetään työntekijän pyynnöstä ja valvomattomat käynnistyvät ajastettuna. Ohjelmistorobotiikkaa käytetään monilla toimialoilla tehostamassa toimintoja, kuten asiakaspalvelussa sähköisten allekirjoitusten varmistaminen ja skannattujen asiakirjojen liittäminen. Robotiikka tehostaa toimintoja ja vähentää manuaalista työtä sekä lisää tarkkuutta (Gillis, 2024).

Ohjelmistorobotiikalla voidaan hoitaa myös integraatioita, koska RPA matkii ihmisen työskentelyä se voi toimia liittymänä kahden järjestelmän välillä siirtäen dataa järjestelmästä toiseen. Ohjelmistorobotiikan merkittävänä etuna verrattuna ohjelmistorajapintaan on se, ettei RPA vaadi ohjelmointiosaamista vaan sitä voidaan tehdä koodivapaasti ja kehitystyö on usein API rajapintaan verrattuna nopeaa. Kun integroitavana on monimutkainen järjestelmä ovat ohjelmistorobotin rajoitetut kyvyt usein riittämättömiä ja ohjelmistorajapinnan käyttö tehokkaampaa. Myös pilvipohjaisten ohjelmistojen vaatimukset tiedonhallinnassa suosii APIn käyttöä. Integraatiotavan käyttö riippuu ohjelmistoista ja vaatimuksista sekä ohjelmistorobotiikalla, että ohjelmistorobotiikalla on hyvät ja huonot puolensa, mikäli ei ole tarvetta täydelle integraatiolle voi RPA olla täysin riittävä (Bridgwater, 2019).

Ohjelmistorobotiikalla (RPA) voidaan integroida monipuolisesti sovelluksia ja järjestelmiä, myös legacy-työpöytä sovellusten, joissa ei API integraatio mahdollisuutta olisi tehtävissä RPA:lla. RPA työkaluihin on kehitetty myös koodivapaita API-rajapintoja, joilla mahdollistetaan sujuva tiedonsiirto eri sovellusten, järjestelmien tai tietokantojen välillä. Ohjelmistorobotti voidaan asettaa käynnistymään myös tapahtuman perusteella, esimerkiksi saapuvan sähköpostin tai valmiiksi merkityn tehtävän perusteella, jolloin ne voivat reagoida ja siirtää tietoa reaaliajassa. Ohjelmistorobotiikalla on myös mahdollista rakentaa erilaisia tiedon validointi- ja muunnossääntöjä ennen tiedon siirtoa seuraavaan järjestelmään, tällä tietoa voidaan rikastaa sekä varmentaa yhteensopivuus vastaanottavan järjestelmän kanssa. Legacy-järjestelmistä puuttuu tyypillisesti ohjelmistorajapinta, jolla integraatio voitaisiin sujuvasti toteuttaa. RPA-työkalulla voidaan tehdä botteja, jotka navigoivat sujuvasti legacy-järjestelmässä toistamalla ihmiskäyttäjän toimia käyttöliittymässä, tällöin saadaan automatisoitua runsaasti manuaalista työtä. Ohjelmistorobotiikalla voidaan pidentää legacy-järjestelmien, joista puuttuu API-tuki käyttöikää ja lisätä organisaation tuottavuutta kustannustehokkaasti (Das, 2023).

Usein RPA on edullisempi ja erityisesti nopeampi vaihtoehto järjestelmäintegraation tarpeeseen, joka on mahdollista toteuttaa yrityksen olemassa oleviin IT-järjestelmiin. RPA botilla on mahdollista saada vaikuttavuutta nopeasti, sillä yksinkertaisen prosessin opetus robotille on mahdollista toteuttaa muutamassa päivässä ja muutoksia botille on mahdollista tehdä, mikäli järjestelmä konfiguraatioon tai käyttöliittymään tulee muutoksia. Ohjelmistorobotiikan työkalut ovat usein koodivapaita, joten niiden tuottaminen ei yleensä vaadi ohjelmointiosaamista, vaan näitä voi tehdä niin sanotut kansalaiskehittäjät (Luukka, 2017).

Ohjelmistorobotiikka tiedonsiirrossa järjestelmäintegraation sijaan on hyödyllistä, mikäli organisaatiossa ei ole paljoa teknistä osaamista, onnistuu tiedonsiirron automatisointi sääntöpohjaisilla prosesseilla RPA työkaluilla. Tiedonsiirto on pääsääntöisesti automaatioon perustuvaa, eikä yrityksellä ole halua uudistaa kaikkia olemassa olevia prosesseja. Tiedonsiirtoratkaisulla on kiire, RPA käyttöönotto ja automaation teko on varsin nopeaa ja ratkaisu saadaan kiireellisestikin tuotantoon,

kuitenkaan RPA ratkaisua ei tulisi käyttää laastariratkaisuna, jolla nopeasti korjataan toimimatonta prosessia. Ohjelmistorobotiikalla voidaan automatisoida prosesseja, mutta se ei kuitenkaan ole paras vaihtoehto, jos halutaan integroida useita järjestelmiä, tai yhdistää useita palveluita toisiinsa, sekä tilanteessa, jossa yritys on luopumassa legacy-järjestelmistä (Smartbridge, 2024).

Ohjelmistorobotiikan kehittäjän ei tarvitse tuntea miten robotisoitava kohde on ohjelmoitu vaan riittää, että tunnetaan käyttöliittymä, jossa robotti työskentelee. Ohjelmistorobotin kehittämistä voisi kuvailla esimerkillä, jossa uudelle työntekijälle opastetaan kuinka järjestelmässä suoritetaan tehtäviä, koska robotti jäljittelee ihmisen työskentelyä käyttöliittymässä. Ohjelmistorobotit jaetaan yleensä kolmeen luokkaan, valvotut, valvomattomat ja hybridi robotit. Valvomattomassa automaatiossa robotti pystyy tekemään itsenäisiä päätöksiä ilman ihmisen avustusta, valvotussa automaatiossa robotti vaatii ihmisen apua automaatiossa, ja hybridi tapauksessa työnkulut sisältävät sekä valvomatonta, että valvottua automaatiota (Parikh, 2022).

3.9 Tekoäly järjestelmäintegraatiossa

Tekoälyä ei itsessään tee järjestelmäintegraatiota, mutta se voi toimia useassa kohdassa integraatiota työkaluna. Tekoäly voi havaita poikkeamia ja lisätä tehokkuutta API-yhdyskäytävien ja integraatioalustojen kautta. Järjestelmäintegraatio kehittyy edelleen kohti automaattisempaa ja älykkäämpää tiedonsiirtoa. Tekoälyavusteiset API-integraatiot iPaaS-alustoilla ovat tulossa entistä yleisemmiksi, tuoden tehokkuutta ja vähentäen manuaalista puuttumista tiedonsiirrossa. Tekoäly voi toimia osana ohjelmistorobotiikka ja päätellä virhetilanteissa, miten prosessin tulisi jatkua. Tekoäly järjestelmäintegraatiossa, päättelee milloin tiedonsiirtoa tarvitaan, se voi tehdä tarvittaessa tietomuunnoksia eri järjestelmien välillä ja auttaa virhetilanteiden ratkomisessa (Muthukrishnan, 2025).

iPaaS-integraatioalustaa valittaessa tulisi varmistaa, että se tukee tekoäly sovelluksia ja yrityksellä on pääsy kriittisiin tietoihin, joita tekoälysovellusten hyödyntäminen edellyttää. Pilviteknologialle rakennettu nykyaikainen iPaaS ratkaisu mahdollistaa joustavan ja skaalautuvan alustan, jolle soveltuvia tekoälyratkaisuja on mahdollista rakentaa, tällainen alusta tarjoaa laajan valikoiman liitimiä, joita tarvitaan tekoälyintegraatioiden tekemiseen. Tekoälyn lisäys integraatiossa laajentaa työkulkujen tekijäjoukkoa, erilaisella taustalla ja osaamisen omaavat tekijät voivat työskennellä jatkossa yhdessä. Tekoälyn lisäys moninkertaistaa kehityssyklien nopeuden jatkossa, sen sijaan, että yritettäisiin uudistaa kokonaisia järjestelmiä tai luoda uusia prosesseja tyhjästä, tulisi keskittyä lisäämään tekoälyä olemassa oleviin työnkulkuihin. Yrityksen tulisi pystyä tunnistamaan prosessit, joissa tekoälyn lisäyksellä olisi saavutettavissa nopeita ja helppoja hyötyjä ja jatkaa käyttöönottoa näistä saatujen oppien avulla (Waldron, 2024).

Erilaiset tekoälyratkaisut kuten koneoppiminen (ML, machine learnin) ja luonnollisenkielen prosessointi (NLP, natural language processing) voivat nopeuttaa ja poistaa virheitä nykyaikaisista heterogeenisista järjestelmistä, jotka usein koostuvat monista eri osista millä kaikilla on oma arkkitehtuurinsa. Tekoäly voi analysoida ennustaa tietovirtoja, joista voidaan tunnistaa virheitä aiheuttavia uhkia ja ennustaa huoltotarpeita. Tekoäly voi mahdollisesti myös parantaa tietoturvaa käyttämällä havainnointia poikkeukselliseen toimintaan ja käyttämällä mukautuvia oppimismalleja. Hajautetut infrastruktuuri arkkitehtuuri tuo vikasietoisuutta ja mukautuu paremmin ympäristön muutoksiin, tekoäly integrointi hajautettuihin malleihin lisää vikasietoisuutta sekö mahdollistaa uusia kehityspolkujaseuraaviin teknologioihin (Seth, 2025).

4 Tutkimusmenetelmät

Laadulliseen tutkimukseen, kuten muihinkin tutkimuksiin kuuluu tavanomaisesti aiheen valinta, tutkimus-, tavoitteiden asettelu, kysymysten muotoilu sekä rajaus, teoreettinen viitekehys kirjallisuuden avulla, lähestymistavan valinta, tutkimusmenetelmät, aineiston hankinta, aineiston analysointi ja tulosten raportointi. Laadullisen tutkimuksen asetelma on joustava ja eri vaiheissa voi olla päällekkäisyyttä, tutkimuksen edetessä tutkija voi melko joustavasti palata aiempiin valintoihin ja tehdä niihin muutoksia. Laadullisessa tutkimuksessa aineiston hankinnassa käytetään tavanomaisesti haastatteluja, dokumenttiaineistoja, havainnointia tai mahdollisesti näiden yhdistelmiä (Siltaoja ja Sorsa, 17-25, 2020).

4.1 Lähestymistapa

Tapaustutkimus on kehittämistyölle hyvin sopiva lähestymistapa, kun tutkimuskohteena on esimerkiksi tuote, palvelu tai prosessi ja tavoitteena on tuottaa kehittämisehdotuksia tai ideoita. Tapaustutkimuksessa pyritään tuottamaan syvällistä ja yksityiskohtaista tietoa tutkittavasta kohteesta. Kohteita tapaustutkimuksessa on usein vain yksi, mutta tapaustutkimus voi kohdistua myös useampaan tapaukseen tärkeää kuitenkin on, että kohde ymmärretään kokonaisuutena. Kirjallisuuteen perehdyttäessä on tärkeää löytää taustat ja menetelmät, joiden pohjalta on käsitelty vastaavia ongelmia kuin omassa työssä. Tapaustutkimusta voi tehdä niin laadullisin kuin määrällisin ja näiden menetelmien yhdistelmällä (Ojasalo, Moilanen ja Ritalahti, 52-54 2014).

Tämän opinnäytetyön lähestymistapana on tapaustutkimus, joka sopii tässä tilanteessa, kun kehittämiskohteena on prosessi ja kehittämistyön tavoitteena on tuottaa kehittämisehdotus nykyiseen prosessiin. Tässä tapauksessa ei ole mahdollista päästä testaamaan prosessia todellisessa ympäristössä, vaan tarkoituksena on verrata teknologioita ja tehdä päätelmiä niiden sopivuudesta yritys X:lle.

4.2 Aineiston hankintamenetelmät

Laadullisen tutkimuksen (kvalitatiivisen) avulla pyritään usein kuvaamaan tosielämän tapahtumia. Laadullisessa tutkimuksessa puhutaan harkinnanvaraisesta näytteestä otoksen sijaan, tutkimuskohteena on sellainen huolella valittu tapahtuma, josta halutaan lisätietoa. Oleellista on tutkimuskohteen tarkka kuvaus sekä tulkintojen tarkka perustelu, jotta lukija voi tehdä johtopäätöksiä tutkimuksen luotettavuudesta. Tiedonhankintamenetelminä käytettäviä haastatteluja voidaan pitää eri tavoin kuten strukturoituina lomakehaastatteluina tai vapaamuotoisempina puolistrukturoituina kuten teema-, yksilö- tai ryhmähaastattelu, joissa kysymykset ovat valmiiksi laadittu mutta näiden lisäksi voidaan tehdä myös tarkentavia kysymyksiä (Ojasalo, Moilanen ja Ritalahti, 104-108, 2014).

Haastattelu on aineistonkeruumenetelmänä hyvin joustava, laadullisessa tutkimuksessa tutkijan tehtävänä on tulkita haastateltavan omakohtaisia havaintoja ja tulkintoja tutkittavasta aiheesta. Haastattelu on vuorovaikutteista ja tärkeää tunnistaa, että osapuolet vaikuttavat aina toisiinsa, jotta uskottavien päätelmien teko aineistosta olisi mahdollista tulisi haastattelut aina tallentaa. Haastattelussa on metodologinen etu siinä, että haastateltaviksi voidaan valita sellaisia henkilöitä, joilla tiedetään etukäteen olevan tietoa tai kokemusta tutkittavasta aiheesta (Siltaoja ja Sorsa, 231-239, 2020).

Aineiston hankintamenetelminä käytetään puolistrukturoituja teemahaastatteluja, haastattelen organisaatiossa järjestelmäasiantuntijoita sekä palvelu- ja tietohallintopäällikköä, palvelujohtajaa ja integraatioasiantuntijaa, jotka tuntevat ostoreskontran prosessia sekä käytössä olevia järjestelmiä. Haastattelemalla saan tietoa prosessin nykytilasta ja toivottavasti myös näkemystä siitä, miten prosessia voisi kehittää jatkossa. Haastattelut toteutetaan etäyhteydellä Microsoft Teams-sovelluksella ja haastattelut tullaan myös tallentamaan.

4.3 Aineiston analyysimenetelmät

Aineiston hankinnan jälkeen laadullisen aineiston analysointi jatkuu yleensä kahdesta suunnasta, lukemalla aineistoa yhä uudelleen pyrkien löytämään yhteisiä teemoja, luokkia tai kategorioita, joilla aineistoa voisi ryhmitellä. Toisaalta taasen tutkija pyrkii hankkimaan lisää tietoa lukemalla aihetta käsitteleviä teorioita ja tutkimuksia, tiedot vaikuttavat myös siihen millaisia kategorioita tutkija pyrkii löytämään aineistosta. Analyysin tavoitteena on ymmärtää, kuvailla ja tulkita tutkimuksen ilmiöitä, niin ettei kirjoiteta puhtaista faktoista vaan pyritään avaamaan lukijalle aiheeseen liittyvä konteksti, jossa aihe esiintyy (Siltaoja ja Sorsa, 333-334, 2020).

Teemahaastattelu kannattaa nauhoittaa, litterointi käy helpommin haastattelun jälkeen ja itse haastatteluun pystyy keskittymään paremmin. Aineiston analyysitapa vaikuttaa siihen, kuinka aineisto tulisi kirjoittaa puhtaaksi, mikäli haastattelijaa kiinnostaa vain esiin tulleet asiat voidaan litterointi tehdä yleis- tai kirjakielellä. Aineistoa analysoitaessa luetaan haastatteluista saatu litteroitu aineisto useasti läpi, jonka jälkeen pyritään luokittelemaan aineisto ja haetaan yhtymäkohtia käytettyyn teoriaan. Litteroitu haastatteluaineisto puretaan teema-alueittain, tarkoituksena tarkastella aineistossa esiintyviä asioita ja ilmiöitä, jotka toistuvat haastateltavien kesken. Haastatteluaineiston yhteyksien tarkasteluun voidaan käyttää useita keinoja, kuten tyypittelyä, jossa asioita ryhmitellään yhteisten piirteiden mukaan, ääriryhmittely, jossa vastaajat jaotellaan vastakohtiin tai etsiä saaduista vastauksista poikkeamia. Dokumenttianalyysilla pyritään tekemään päätelmiä kirjalliseen muotoon saatetusta verbaalisesta aineistosta, analyysitavan vahvuus on sen herkkyydellä asiayhteydelle. Dokumenttianalyysia voidaan tehdä aineistolähtöisesti, jossa saatu haastatteluaineisto ohjaa analyysin tekoa tai teorialähtöisesti, jolloin analyysirunko rakennetaan kerätyn teoriapohjan perusteella (Ojasalo, Moilanen ja Ritalahti, 110-111, 136-140, 2014).

Aineistoa tulen analysoimaan teorialähtöisen dokumenttianalyysin keinoin, tapa sopii haastatteluiden analysointiin. Haastattelut puretaan kategorioihin, joita tulkitsen ja pyrin löytämään haastatteluaineistosta yhteisiä teemoja. Haastattelukysymykset liite 1 on laadittu avoimiksi ja voin tarvittaessa kysyä haastateltavilta tarkentavia kysymyksiä, mikäli vastauksista nousee esille johtolankoja, joilla pääsen syvemmälle aiheeseen.

5 Ostoreskontran integraation kehitysnäkymät

Tapaustutkimuksessa pyritään löytämään vastaukset tutkimuskysymyksiin: K1. mitä hyötyjä API-integraatioiden käytöstä olisi saavutettavissa? K2. tulisiko uuteen ERP-järjestelmään liittyvät integraatiot sisällyttää kilpailutukseen vai tulisiko hankkia erillinen integraatioalusta? Sekä K3. onko yrityksellä riittävästi omaa integraatio osaamista? Haastatteluilla pyrin löytämään vastauksia yritys X:n nykytilanteeseen, miten integraatioita on nykyisin toteutettu ja palveleeko integraatiot toimintoja ostoreskontran prosessissa. Selvitin minkälaisia integraatioita yritys X:llä on, kuinka ne on toteutettu, sekä mitä hyötyjä olisi saavutettavissa nykyaikaisella integraatio tekniikalla ohjelmistorajapinnalla, toisin sanoen minkälainen näkemys ja ymmärrys haastatelluilla on järjestelmäintegraation mahdollisuuksista. Yrityksen tarve hankkia uusi ERP-järjestelmä vaikuttaa merkittävästi integraatioiden toteutukseen, mitä mahdollisia vaihtoehtoja on valittavissa järjestelmäuudistuksen myötä ja mitä hyötyjä haastateltavat löytävät eri vaihtoehdoista. Onko yritys X:llä omaa integraatio osaamista, onko sitä käytetty ja voisiko sitä hyödyntää enemmän? Mitä hyötyjä olisi saavutettavissa, jos integraatiot tehtäisiin itse tai ostettaisiin palveluna muualta.

5.1 Haastatteluiden kulku

Tutkimushaastattelut on toteutettu Teams-kokouksina etäyhteydellä viikoilla 10 ja 11, etäyhteydellä sen vuoksi, että haastateltavat asuvat eri puolilla Suomea. Haastateltavaksi olin kutsunut yritys X:n asiantuntijoita, jotka työskentelevät tai tuntevat ostoreskontran prosessin, sekä yrityksen integraation asiantuntijoita. Sain haastateltua kaikki kahdeksan toivomaani asiantuntijaa, joiden vastaukset haastattelukysymyksiin (liite 1) toivat esiin varsin yhtenevän kuvan yritys X:n integraatioiden nykytilanteesta, sekä näkymistä miten järjestelmäintegraatiota tulisi kehittää. Haastattelujen vastaukset luonnollisesti vaihtelivat riippuen haastatellun kokemuksesta ja käytännön osaamisesta liittyen aiheeseen. Opinnäytetyö on rajattu koskemaan ostoreskontran prosessia, jotta työn laajuus ei paisuisi liian suureksi. Haastatellut olisivat mieluusti laajentaneet näkemyksen koskemaan kaikkia yritys X:n prosesseja ja integraatioita kokonaisuudessaan, koska usein integraatiot koskevat useita eri prosesseja ja olisivat järkevä suunnitella yhtenä kokonaisuutena. Puolistrukturoitu haastattelu avoimilla kysymyksillä toimi hyvin ja mahdollisti lisäkysymysten tekemisen tarvittaessa, mikäli jokin vastaus vaati lisätarkennuksia tai laajempaa avaamista.

Haastatteluilla sain kerättyä kattavan katsauksen yritys X:n integraatioiden nykytilanteeseen sekä asiantuntijoiden näkemystä mahdollisuuksiin järjestelmäuudistusten myötä. Opinnäytetyön teoriakatsaus integraatioiden tarkoitukseen, eri toteutustapoihin ja arkkitehtuuriin lisäsivät ymmärrystäni siitä, kuinka järjestelmäintegraatiota voidaan toteuttaa ja mikä tekniikka voisi olla toimivaa yritys X:n tapauksessa. Teoriatausta auttaa minua analysoimaan haastattelututkimuksen tuloksia ja näiden avulla pystyn vastaamaan tutkimuskysymyksiin. Haastattelutilanne vaati usein hieman

haastattelukysymysten taustan avaamista ja tarkentamista, koska haastattelukysymykset olivat varsin ylätasolla ja joitain termejä oli tarpeen avata haastateltaville.

Tutkimusta tehdessäni minulle aukesi järjestelmäintegraation moninaiset mahdollisuudet, erilaisia tekniikoita toteuttaa integraatio on paljon ja sopivan vaihtoehdon valitseminen vaihtelee monesta tekijästä. Valintaan vaikuttavat yrityksen omat resurssit, käytössä olevat järjestelmät, jotka voivat sulkea jonkin vaihtoehdon kokonaan pois, tarvitseeko siirrettävää dataa muokata tai rikastaa vai onko samalle tiedolle käyttöä useammassa eri järjestelmässä. Integraatiot on mahdollista ostaa kokonaan palveluna tai ne voidaan tehdä itse, mikäli yrityksellä on riittävästi osaamista. Integraatioiden tarkoitus ja mahdollisuudet on hyvä kirkastaa mielessä ja suunnitella yritykselle integraatioiden kokonaisarkkitehtuuri, jotta ei joudu tekemään moninkertaista työtä.

5.2 Integraation toteutustavat

Kuten Tähtinen, myös yritys X:n haastatellut asiantuntijat jakavat saman käsityksen integraatioiden tarpeellisuudesta ja siitä miksi järjestelmäintegraatiota tehdään. Integraatioita tarvitaan koska yritys X:ssä ei ole käytössä yhtä keskitettyä järjestelmää vaan useita eri tarkoitusta palvelevia järjestelmiä, joiden välillä informaatio tulee saada kulkemaan sujuvasti ja häiriöttä. Integraatioiden tarkoituksena on lisätä automaatiota, jotta tietoa ei jouduttaisi tallentamaan manuaalisesti järjestelmästä toiseen, integraatio mahdollistaa myös ajastetut automaattiset työnkulut, jolla saadaan myös osaltaan parannettua tehokkuutta. Haastatteluilla selvitin, ketkä integraatiot toteuttavat nykyisin ja yritys X:n valitseman strategian mukaisesti kaikki integraatiot ja liittymät tuottaa valittu it-kumppani, joka vastaa myös ylläpidosta. Kuten Babati kuvaili ja haastatellut vahvistivat, eivät yritys X:n nykyiset legacy-järjestelmät mahdollista nykyaikaisen API-rajapinnan käyttöä. Integraatiot on toteutettu niin sanotulla flat-file sftp -tiedonsiirrolla, jossa lähettävästä järjestelmästä kirjoitetaan tieto ulos tiedostoon, jonka integraatiossa toteutettu liittymä siirtää seuraavalle järjestelmälle sisään luettavaksi. Käytössä olevat integraatiot eivät sisällä älykkyyttä, jolla siirrettävään tietoon voisi tehdä muunnoksia, tämän vuoksi järjestelmien välillä tulee käyttää sovittua tietue muotoa. Vaihtoehtoisesti lähettävä tai vastaanottava järjestelmä joutuu tekemään muunnoksen tietueeseen.

Ostoreskontran käytössä olevat integraatiot ovat tyypillisesti nykyisin toteutettu point-to-point, tai hub to spoke -mallilla, mutta myös ohjelmistorobotiikalla on tehty muutama integraatio. Ohjelmistorobotiikan merkittävä heikkous on se, että se käyttää samaa käyttöliittymä kuin ihminen, mikäli käyttöliittymässä tulee pienikin muutos se vaikuttaa robotin työnkulkuun ja voi estää sen toiminnan. Ohjelmistorobotit on yritys X:ssä yleisesti tehty suorittamaan tiettyä prosessin osaa itsenäisesti, automatisoidut tehtävät eivät vaadi robotilta päättelyä vaan ovat esimerkiksi raportin suorittamista ja lähettämistä eteenpäin. Mikäli prosessissa tulisi poikkeus robotti pysähtyy ja lähettää palveluasiantuntijoille ilmoituksen epäonnistuneesta tehtävästä. Ohjelmistorobotit ovat ajastettuja, käynnistetään

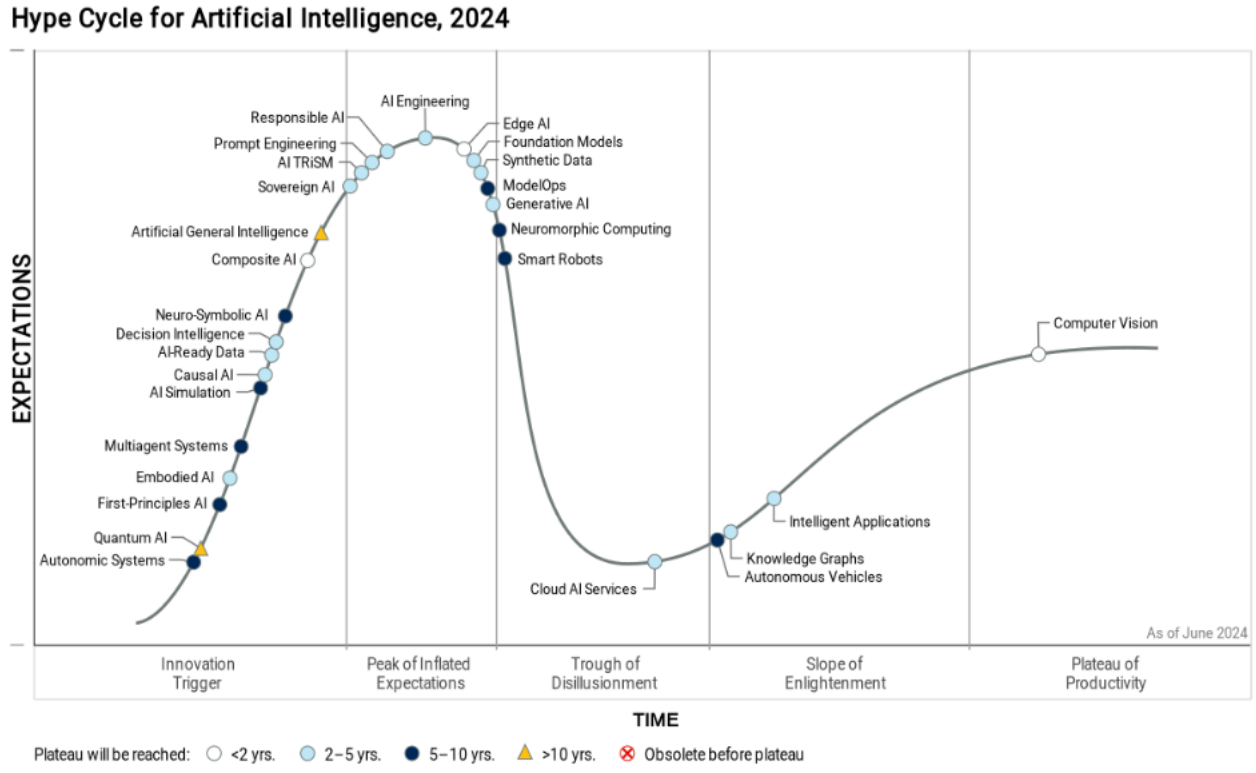
pyynnöstä taikka käynnistyvät herätteestä. Toistaiseksi käytettyihin robotteihin ei ole yhdistetty tekoälytoimintoja, eivätkä botit tee itsenäistä päättelyä, tekoäly voisi tuoda robotteihin suuren tuottavuusmahdollisuuden, mutta tietoturvan erityisesti henkilötietojen kanssa tulee olla erittäin huolellinen.

Valitun strategian mukaan it-kumppani tuottaa integraatiot, eikä yritys X:llä ole käytössä teknologiaa, jolla toteuttaa integraatioita itse. Koska valtaosa liittymistä on toteutettu point-to-point -integraationa, tulee näistä myös suurempi kulu, koska veloitus perustuu liittymien määrään. Mikäli liittymässä tulee vikaa voi point-to-point olla helpompi ja nopeampi korjata, mutta koska haastateltavien mukaan ostoreskontrassa satoja eri liittymiä, on point-to-point -liittymät ylläpidollisesti työläämpiä. Hub to spoke -integraatio voi vaikuttaa häiriötilanteessa useisiin eri liittymiin ja järjestelmiin ja haitata yritys X:n useita eri prosesseja. Koska it-kumppani tuottaa integraatiot on ostoreskontran järjestelmäasiantuntijoilla vain puutteellinen näkyvyys liittymien toimintaan, joka hankaloittaa prosessissa ongelmien ratkaisemista. Ostoreskontran järjestelmäasiantuntijoilla on näkyvyys ainoastaan siihen mitä integraatioita ja liittymiä on toteutettu, ja mistä mihin ne siirtävät informaatiota, mutta näkyvyys siihen, mitä liittymässä siirtyy ja onko liittymä toiminnassa puuttuu. It -kumppanin rakentamissa liittymistä pitäisi lähteä virheilmoitus, mikäli siirto epäonnistuu liittymässä, tämä ei kuitenkaan aina toimi vaan johtuen joko siitä ettei virheilmoitus toimi tai sitä ei ole huomattu määritellä liittymää tilatessa. Nykyisten integraatiototeutusten heikkous on myös siinä, etteivät ne skaalaudu hyvin tai ollenkaan lisääntyvän liikenteen ja liittymien mukaan.

Liittymien teko eri integraatioissa vaihtelee helposta hyvinkin vaikeaan, riippuen järjestelmästä ja miten liittymä toteutetaan. Nykytilanteessa kaikki liittymätyöt ostetaan kumppanilta, eikä sopimukset mahdollista yritys X:lle toimittajan järjestelmien käyttöä niin, että liittymiä voitaisiin tehdä itse ostoreskontran asiantuntijoiden toimesta. Haastateltavien yhtenevä näkemys on, että helppoja liittymiä tulisi olla mahdollista tehdä itse, jolloin saataisiin kustannushyötyjä, mutta omana työnä myös toteutusaikataulu voisi olla nopeampi, koska ei tarvitse käyttää ulkopuolista toimijaa. Ylläpidon ja vastuun kannalta voisi tulla haasteita, mikäli liittymiä rakentaisi sekä it-kumppani, että ostoreskontra itse. Jotta ostoreskontra voisi nykytilanteessa tehdä liittymiä myös itse, vaatisi se sopimusneuvotteluja it-kumppanin kanssa sekä selvää vastuunjakoa, miten ylläpito tapahtuu ja miten vastuu jakautuu ongelmatilanteissa, tämä voisi osoittautua hyvin hankalaksi.

Vaikka haastatteluissa ei tekoäly noussut esille asiantuntijoiden vastauksissa, kannattaisi tekoälyn tuomat mahdollisuudet ottaa myös huomioon integraatioarkkitehtuuria uusittaessa. Tekoälysovelluksista useimmille nykyään tutut ovat suurilla kielimalleilla toimivat keskustelevat chatbotit kuten ChatGBT. Tekoälysovelluksissa on ollut viimevuodet nopeaa kehitystä ja uusia sovelluksia ja kehityskohteita on tullut valtavan paljon, kuvan 6 Gartnerin Hype Cycle (Gartner, 2024) esittää useita tekoälyn

sovelluksia ja niiden kehitysvaiheita. Kuvasta 6 huomaa, että useimmat tekoälysovellukset ovat edelleen yli suurten odotusten vaiheessa, josta kehittyminen valtavirran tuotteeksi kestää vielä vuosia riippuen tuotteesta. Vaikka yritys X ei suoraan tekoälyä integraatioalustaansa tai integraatiosovelluksiin ole vielä hankkimassa, kannattaa teknologian kehitystä seurata ja ottaa se huomioon tulevaa alustaa valitessa. Kilpailutuksessa voisi esimerkiksi esittää vaatimuksen, jossa lisäpisteitä saa, jos pystyy esittämään realistisen kehityskartan tuotteeseen tuotaville tekoälyominaisuuksille.



Kuva 6. Gartner Hype Cycle for Artificial Intelligence 2024 (Gartner, 2024).

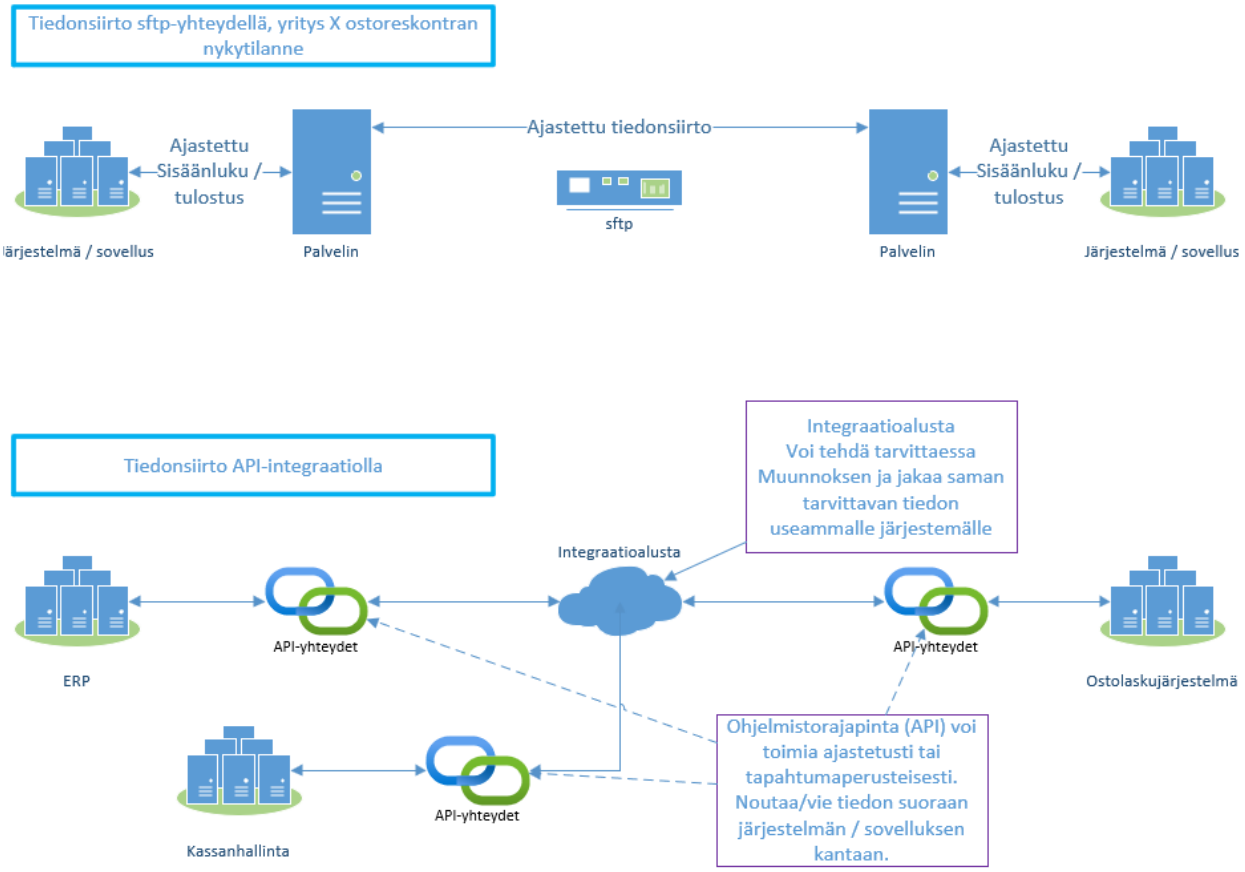
Tekoälyn tuomat mahdollisuudet tulisi ottaa huomioon riippumatta siitä, miten integraatiot yritys X:llä tultaisiin toteuttamaan esimerkiksi omalla integraatioalustalla tai integraatiot palveluna mallilla. Tekoälyn tuomat mahdollisuudet ovat suuret, joten sen integroimisen mahdollisuus hankittaviin järjestelmiin tulisi varmistaa. Viime vuosina tekoäly innostus on ollut korkealla chatbottien myötä ja yritys X:llä on ollut tekoälypilotteja ostolaskujen tiliöinnissä. Nykyisellään yritys X:n oma robotiikka tiimi on tehnyt runsaasti ohjelmistorobotteja, mutta tekoäly työkaluja näissä ei toistaiseksi ole käytetty. Tekoälyn integroiminen myös ohjelmistorobotteihin voisi tuoda lisää vikasietoisuutta ja skaalautuvuutta botteihin. Toki tekoälyn käytössä tulee aina ottaa huomioon myös tietoturva ja henkilötiedot, joiden kanssa yritys X on tiiviisti tekemisissä.

5.2.1 API-integraatio

Ohjelmistorajapinnalla on Vamenturen mukaan mahdollista tehdä reaaliaikaisia integraatioita ja sama huomio nousi myös haastatelluilla henkilöillä tärkeimpänä API:n ominaisuutena. API-integraatiot myös skaalautuvat hyvin mikä tuo hyötyjä yritys X:n tapauksessa, koska liittymiä on ostoreskontran prosessissakin paljon ja siirrettävää aineistoa paljon. Yritys X:ssä ei API -integraatioita ole ostoreskontrassa tehty vielä ollenkaan, tämä johtuu siitä, ettei nykyiset järjestelmät ole sitä mahdollistaneet. API-integraatioista saatava hyötyyn vaikuttaa käytössä olevien järjestelmien kokonaisarkkitehtuuri, minkälaista kyvykkyyttä järjestelmissä on ja mitä niillä on mahdollista toteuttaa.

Vahvuutena API-integraatioissa haastatellut näkevät nopeuden, joka mahdollistaa lähes reaaliaikaisen tiedonsiirron, mikä vähentäisi integraatiosta johtuvaa odotusaikaa sekä tyhjäkäyntiä prosessissa. Ohjelmistorajapinnalla integraatio voi noutaa ja viedä tietoa suoraan järjestelmän kannasta kantaan, mikä tuo laajempia hyödyntämismahdollisuuksia liittymiin ja ostoreskontran prosesseja olisi mahdollista muokata joustavammiksi, koska tietyt tehtävät eivät olisi enää kellonaikaan sidottuja. Nykyisin sftp-tiedonsiirroissa, useilla käyttäjillä on pääsy palvelimen kansioisin mihin liittymä siirtää tiedostot sisään luettavaksi tai noudettavaksi. Tämä voi aiheuttaa tilanteen, jossa käyttäjä voi käydä vahingossa tai tarkoituksella muokkaamassa tai poistamassa tiedoston, ohjelmistorajapinnalla toteutettavassa liittymässä tätä ongelmaa ei olisi, koska liittymä vie tiedon suoraan järjestelmän kannasta toiseen. Sftp:llä toteutetussa integraatiossa on tullut vastaan ongelmia, joissa siirrettävän tiedoston koko on kasvanut liian suureksi ja tiedosto on jouduttu jakamaan useammaksi pienemmäksi, mikä on aiheuttanut lisätyötä liittymässä, vastaavaa skaalautumisongelmaa ei olisi API-integraatiossa vaan se pystyisi siirtämään myös suuria määriä tietoa.

Kuva 7 havainnollistaa eroa sftp-tiedonsiirron ja API-integraation välillä. Kuvassa API-integraatiossa on mukana myös integraatioalusta, joka voi toimia välittäjänä sekä tehdä tarvittaessa muunnoksen siirrettävään aineistoon, ilman alustaa tämä olisi point-to-point -integraatio ohjelmistorajapinnalla. API-integraatio mahdollistaa joustavamman tapahtumapohjaisen tai tarvittaessa ajastetun tiedonsiirron, integraatioalustaan yhdistettynä voidaan yhtä tietoa jakaa usealle sitä tarvitsevalle järjestelmälle tai sovellukselle.



Kuva 7. Sftp ja API tiedonsiirron erot

Haasteena kaikki haastateltavat nostivat API-integraatioiden vaatiman osaamisen, ohjelmointirajapinnan vaativaa ohjelmointiosaamista yritys X:llä ei välttämättä ole riittävästi, jotta integraatioita olisi mahdollista tehdä itse. Vaikka integraatiot tilattaisiin it-kumppanilta, olisi kuitenkin hyvä olla parempi ymmärrys siitä, miten API-integraatio toimii, miten sitä voidaan hyödyntää, jotta liittymästä saataisiin paras hyöty ostoreskontraan tuotantoon. Jotta mahdollisia ongelmatilanteita pystyisi selvittämään sujuvasti pitäisi API-integraatiossa olla lokitus, jotta saataisiin selville mistä ongelma johtuu, nykyisillä sftp-integraatioilla voidaan tarkistaa saapunut tai lähtenyt aineisto ja etsiä virheellinen tiedosto sieltä.

5.2.2 Integraatioalusta

Kirjallisuuskatsauksen mukaan integraatioalustan tarjoajia on paljon ja alustan vakio työkalut ja integraatiot vaihtelevat merkittävästi. Pääsääntöisesti integraatioalustat ovat tilaajalle koodivapaita työkaluja niin, ettei liittymien teko alustalla vaadi ohjelmointiosaamista alustan valmiilla työkaluilla. Integraatioalustat ovat yleensä arkkitehtuuriltaan mikropalveluita ja mahdollistavat sekä API että sftp-pohjaisia integraatioita, ESB palveluväylät puolestaan ovat monoliittisia arkkitehtuuriltaan ja

alustan kyky on yleensä parhaimmillaan tiedostopohjaisessa siirrossa. Mikäli valmiit työkalut ja integraatiot eivät integraatioalustalla olisi riittäviä ostoreskontran tarpeisiin vaatisi uusien komponenttien työstäminen kuitenkin vankkaa ohjelmointiosaamista.

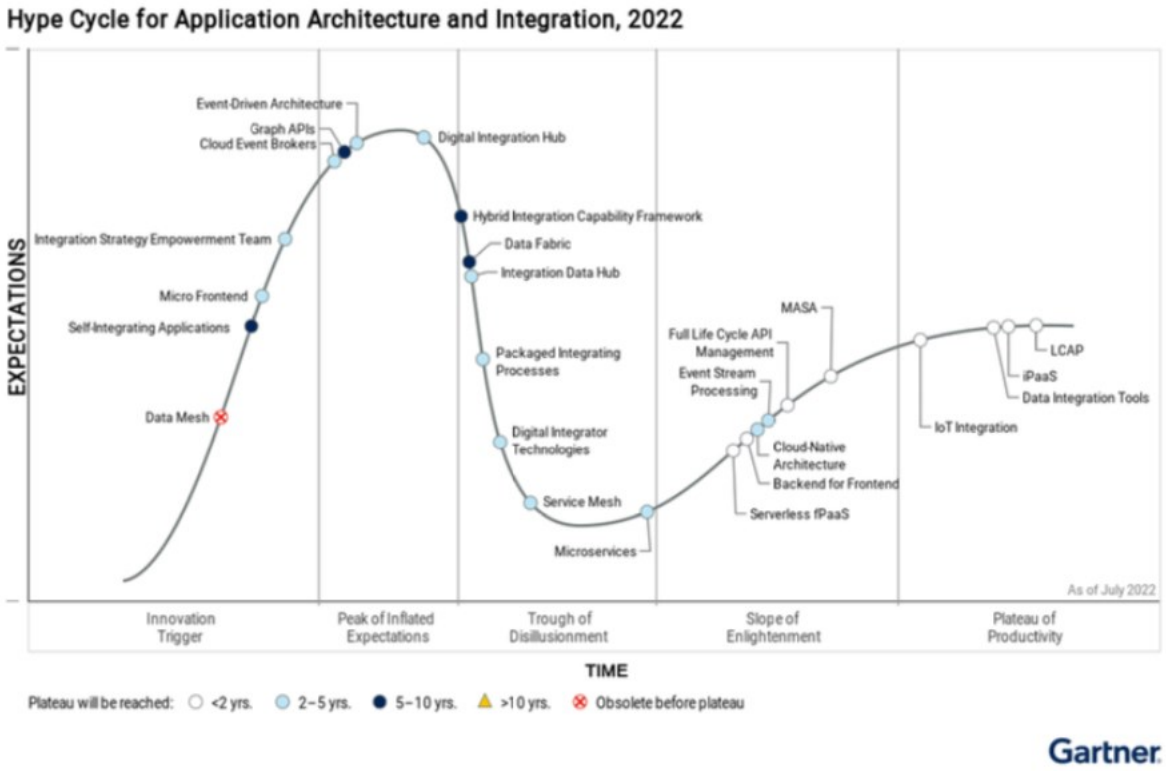
Haastatellut näkivät laajasti hyötyjä integraatioalustan käytössä, niin että asiantuntijoilla olisi vähintään monitorointi näkymä alustalle, mutta paras hyöty koettiin mahdollisuudella tehdä liittymiä alustalla itse. Alustan kyvykkyys käsitellä sekä tiedostopohjaisia siirtoja, että API-liittymiä koetaan hyvänä ominaisuutena, koska API-integraatiot ovat kuitenkin tulevien järjestelmien integraatiotapa, mutta joissain yhteyksissä tarvitaan edelleen sftp-siirtoja. Omana järjestelmänä käytettävä integraatioalusta parantaisi häiriönhallintaa, paremmalla näkyvyydellä liittymien toimintaan sekä mahdollisuudella muokata ja käynnistää liittymä uudelleen.

Haasteena alustan käytössä nähdään yritys X:ssä laajan integraatio-osaamisen puute, jota hyötyjen saavuttaminen oman henkilökunnan tekemisellä vaatisi. Huomiona nousi myös häiriötilanne, joka voisi estää koko alustan toimimisen, tällainen virhetilanne voisi aiheuttaa hyvin laajaa haittaa ja hitautta ostoreskontran palvelussa. Integraatioalustan mikropalvelu arkkitehtuurin rakenne on kuitenkin enemmän hajautettu, joten yhdessä integraatiossa oleva häiriö ei kuitenkaan todennäköisesti estäisi koko alustan toimintaa vaan vaikutukset jäisivät rajatuiksi.

Haastateltujen asiantuntijoiden mukaan integraatioalusta yksinkertaistaisi ostoreskontran integraatioiden tekemistä nykyiseen verrattuna. Nykyisellään alustan käyttöönottoa ei nähdä mielekkäänä, mutta järjestelmä uudistusten yhteydessä, kun integraatiot on tehtävä uusiksi, koetaan integraatioalustan käyttöönotto järkeväksi. Integraatioalustan käyttöönoton yhteydessä tuli myös yrityksen integraatioarkkitehtuuria suunnitella uusiksi, hyvänä vaihtoehtona haastatellut nostivat esille jonkinlaisen hybridimallin, niin että osan integraatioista voisi tehdä it-kumppani ja osan yritys itse. Mahdollisesti yritys X:llä olisi oma integraatioalusta millä olisi mahdollista tehdä yksinkertaiset liittymät ja haastavammat työt tilattaisiin edelleen it-kumppanilta.

Kirjallisuuskatsauksessa integraatioalustan hyviä puolia olivat erityisesti uuden liittymän toteuttamisen nopeus ja helppous olemassa olevalle integraatioon. Integraatioalustan valmiilla työkaluilla ja jo toteutetuilla integraatioilla yritys X:n järjestelmäasiantuntijat pystyisivät tekemään joustavasti ja nopeasti uuden liittymän. Erityisesti uuden asiakkaan käyttöönoton tilanteessa, jossa järjestelmäintegraatio olisi alustalla valmis ja tarve olisi uudelle tiedonsiirtoliittymälle jo tehdyllä integraatiolla yritys X pystyisi tekemään tarvittavan liittymätyön koodivapaalla integraatioalustalla itse. Mikäli olisi tarve uudelle järjestelmä integraatiolle, kannattaisi työ tilata valitulta It-kumppanilta vaativimmissa ja ohjelmointi vaativissa tapauksissa.

Gartnerin Hype Cycle kuvassa 8 esittää teknologioiden kypsyysastetta. Taulukossa esitetään kuinka uudella teknologialla on ylisuuret odotukset ja alun hypen jälkeen koittaa pettymysten laakso, jossa teknologioita on jo otettu enemmän käyttöön, mutta se ei edelleenkään ole laajasti vakuuttanut. iPaaS-integraatio on jo edennyt tuottavuuden tasangolle (plateau of productivity), jossa teknologiaa on otettu laajasti käyttöön ja se on osoittanut toimivuutensa.



Kuva 8. Gartner Hype Cycle (Gartner, 2022).

Kuten kuvassa 8 esitetään iPaaS teknologian kypsyyttä, on integraatioalusta palveluna yritys X:lle hyvä koeteltu teknologia ratkaisu integraatioratkaisuksi. Garterin Hype Cycle kuvassa 8 on vuodelta 2022 joten kuvassa ennustettu iPaaSin tuottavuuden laakso tulisi olla jo saavutettu ja kirjallisuuskatsauksessa tulikin jo esille teknologian toimivuus ja laaja käyttöönotto.

5.3 Uuden ERP -talousjärjestelmän integraatiot

Yritys X:n talouden pääjärjestelmä on käyttöikänsä päässä ja toimittaja on päättämässä sen ylläpidon lähivuosina, jonka vuoksi yritys X on pakotettu uuden ERP-järjestelmän hankintaan. Uuden ERP -hankinnan yhteydessä on tarkoitus myös rakentaa uusi ostolaskujenkäsittelyjärjestelmän kanta, jossa olisi käytössä ohjelmistorajapinta. Uusien järjestelmien myötä ostoreskontran integraatioiden arkkitehtuuri tulee suunnitella ja rakentaa uudelleen. Haastattelukysymyksillä pyrin selvittämään

miten integraatiot tulisi ottaa huomioon osana järjestelmähankintaa, pitäisikö uusi integraatio alusta olla vaatimuksena ERP-hankinnassa vai tulisiko integraatioalusta kilpailuttaa erillisenä hankintana.

Jos integraatioalusta hankitaan osana ERP -järjestelmän kilpailutusta, saataisiin tällä vähennettyä yritys X:n järjestelmätoimittajia mikä puolestaan selkeyttäisi toimittajayhteistyötä. Uudella ERP -toimittajalla olisi myös laaja ymmärrys järjestelmänsä integraatioiden rakentamisesta, joten odotuksena olisi, se että integraatiot tulisi tehtyä laadukkaasti ja nopealla aikataululla, toki integraatiotyössä on aina vähintään kaksi puolta, joiden tulee keskustella keskenään lähettävä ja vastaanottava puoli. Vaikka yksi toimittaja selkiyttäisi toimintaa, se voi tuoda lisäriskiä, mikäli yritys X olisi liian monessa asiassa riippuvainen yhdestä toimittajasta. Riskejä liittyy hintaan, joka tyypillisesti nousee ajan myötä, mutta tässä haastateltavat näkevät hinnoitteluriskin suurempana, mikäli myös integraatiot on kytketty ERP-toimittajaan. Mikäli kilpailutus tehtäisiin niin, että toimittajan integraatioalusta olisi optiona yritys X:llä olisi mahdollisuus tehdä teknologinen arvio siitä kannattaako alusta ottaa ERP -toimittajalta vai pitäisikö hankkia toiselta toimittajalta.

Ostoreskontralla on paljon integraatioita, jotka eivät liity pelkästään talousjärjestelmän nykyisen tai tulevan ja ostolaskujenkäsittelyjärjestelmän väliseen tiedonsiirtoon, vaan integraatioita on myös ostolaskujärjestelmästä kolmansiin järjestelmiin, sekä lähteviä, että tulevia liittymiä. Mikäli integraatioalusta hankittaisiin uudelta ERP-toimittajalta, pitäisi myös näiden muiden integraatioiden teko olla mahdollista tällä alustalla. Haastateltavien mukaan ERP-toimittajan integraatioalustan kautta tulisi pystyä tekemään integraatiot myös ERP liittymättömien järjestelmien välille, kuitenkin tässä nähtiin huolta siinä, miten se sopimuksellisesti mahdollistetaan. Integraatioalusta sitoisi yritys X:n liiaksi ERP-toimittajaan, mikäli tämä toimittaisi myös integraatioalustan. Jos ERP-toimittaja vaihdetaan myöhemmin, menisi myös kaikki tehdyt integraatiot uusiksi, tai mahdollisesti alustan ostaminen omaksi tuli hyvin kalliiksi. Puhtaasti ERP -järjestelmään sidottua integraatioalustaa ei nähdä järkevänä.

Integraatioalustan kilpailuttaminen omana hankintana nähtiin parempana vaihtoehtona, kuitenkin kaikki haastateltavat laajensivat näkökulmaa koskemaan koko yritys X:n kaikkia toimintoja ei pelkästään ostoreskontraa. Kun integraatioalusta olisi omana työkaluna nähdään myös integraatioiden teko joustavampana, vaikka luultavasti uusien integraatioiden teko olisikin alkuun hitaampaa uuden alusta järjestelmän ominaisuuksien opettelun vuoksi. Mikäli alusta olisi oma hankinta nähdään edelleen hyvänä vaihtoehtona se, että integraatioita tekisi osaltaan myös ulkopuolinen it-kumppani nykyinen tai mahdollisesti jokin muu, sekä yrityksen asiantuntijat itse.

Yritys X:n nykyinen integraatiototeutus vastaa varsin hyvin kirjallisuuskatsauksen integraatio palveluna (IaaS) vaihtoehtoa, jossa kaikki integraatiot on ostettu palveluntarjoajalta. Integraatio palveluna vapauttaa yrityksen resurssit muuhun käyttöön ja palveluntarjoaja järjestää käytettävän teknologian,

jolloin tilaaja maksaa vain tuloksista. Veloitus voi olla esimerkiksi kiinteä kuukausiveloitus tietyltä volyymilta tai niin, että uusilta integraatioilta veloitetaan rakentamisen kulu sekä kuukausittainen ylläpito kulu käytössä olevilta liittymiltä. Nykytilanne on yrityksen strategian mukainen, mutta hybridimallilla, jossa integraatioita tehtäisiin niiden sisällön mukaan joko itse tai tilattuna It-kumppanilta palvelisi paremmin yritys X:ää. integraatiopalveluna palvelee hyvin pieniä ja keskisuuria yrityksiä, joilla ei välttämättä ole omia resursseja tai osaamista omien integraatioiden kehittämiseen. Suuremmilla yrityksillä yleisemmin voi olla hyödyllistä hankkia oma infrastruktuuri integraatioiden tekemiseen, jolloin voidaan saada paremmin itselle räätälöidyt lopputulokset ja yritys voi itse huolehtia esimerkiksi tietoturvasta, joka yleensä on kriittinen asia.

5.4 Yritys X:n integraatio-osaaminen

Osaamisen tarve integraatiotyössä vaihtelee, kirjallisuuskatsauksen mukaan API-liittymän teko vaatii syvällisempää ohjelmointiosaamista, kun taasen sftp-siirrolla tehdyt point-to-point -liittymät eivät vaadi erityisen vahvaa osaamista. Integraatioalustan valmiit sovellukset ja palveluväylän valmiit liittymät ovat niin sanotusti matalakoodi tai koodivapaita sovelluksia, joilla pystyy integraatioita luomaan ja ylläpitämään ilman ohjelmointi osaamista. Mikäli integraatioalustalle tai palveluväylälle tulisi tehdä kokonaan uusi integraatio tai jokin tiedonsiirto vaatisi rikastamista tai uudenlaista muuntamista vaatisi tämä kuitenkin ohjelmointiosaamista. Ohjelmistorobotiikalla tehdyt integraatiot toteutetaan yleensä matalakoodi tai koodivapaalla työkalulla, jotka eivät vaadi ohjelmointiosaamista, mutta kuitenkin järjestelmän osaamista ja tuntemusta integroitavista järjestelmistä. Yleisesti ohjelmistorobotiikkaa käytetään integroitaessa legacy-järjestelmiä, joista puuttuu ohjelmistorajapinta. Yritys X:ssä on käytetty runsaasti ohjelmistorobotiikkaa poistamaan yksinkertaisia toisteisia tehtäviä, mutta integraatiossa sitä on käytetty vain muutamassa tapauksessa. Yritys X:llä on oma robotiikka tiimi, jolla on vankkaa osaamista ohjelmistorobottien teosta. Omalla robotiikka osaamisella on mahdollista tehdä joustavasti automaatioita, ylläpito ja mahdolliset muutostarpeet ovat myös nopeita tehdä omana työnä. Omana työnä tehty ohjelmistorobotti voidaan saada hyvin nopeasti tuotantoon ja viestintä on yleensä tehokkaampaa kuin ostetulta konsultilta tai järjestelmätoimittajalta.

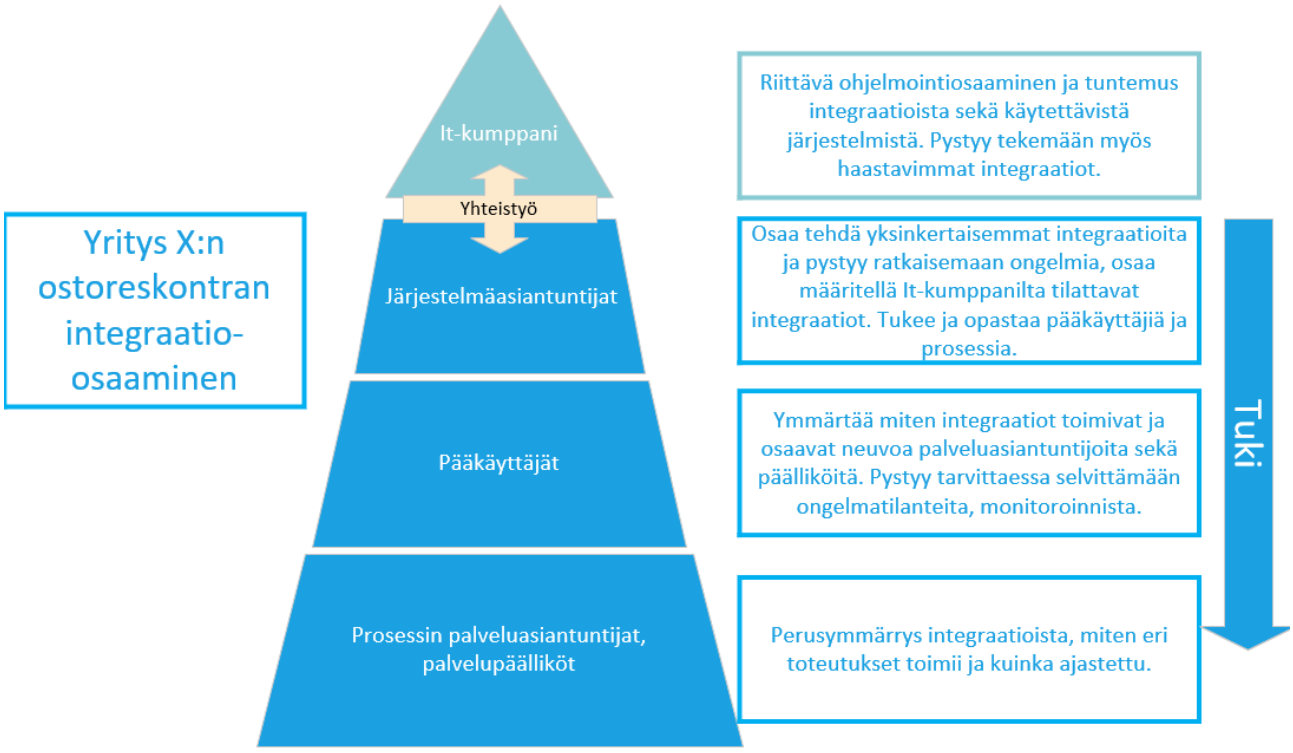
Yritys X:n strategian mukaan integraatio on nykyisin ostettu kumppanilta, mutta haastatteluissa nousi esille tarve omalle integraatio-osaamiselle. Integraatio osaamista tarvittaisiin lisää, jotta uusia liittymiä tilatessa määrittely työ onnistuisi laadukkaammin ja järjestelmäasiantuntijat ymmärtäisivät syvällisemmin sen mihin tarkoitukseen liittymä tilataan ja kuinka se käytännössä toimii. Myös vikatilanteiden selvitys ja häiriöhallinta onnistuisi paremmin, vianhaku kävisi nopeammin, kun asiantuntijat ymmärtäisivät liittymän toiminnan paremmin. Haastatellut totesivat myös, että ostoreskontran palveluasiantuntijoille tulisi opettaa perusteet integraatioiden toiminnasta, vähintään ylätasolla miten integraatiot on toteutettu ja miten data siirtyy liittymässä. Parempi ymmärrys integraatioista helpottaisi

ongelmanratkaisua ja voisi tuoda esiin kehitysideoita tekijätasolta, missä pullonkaulat yleensä tuotannossa tulevat vastaan.

Omana työnä tehdyille liittymille tunnistettiin tarve, jaettu näkemys oli se, että vasteaika paranisi ongelmatilanteissa ja uuden liittymän teko tapahtuisi nopeammin. Vasteaikaa pidettiin merkittävänä hyötynä omana työnä tehdyissä integraatioissa, ongelman ratkaisu on yleisesti kriittistä, jotta palvelutuotannolle ei aiheudu haittaa. Omana työnä esimerkiksi käyttöönotoissa projektityö voisi nopeutua, josta saataisiin hyötyjä projektin kustannuksille sekä yritys X:ssä, että myös asiakkaille alentuneina kustannuksia. Haastatellut asiantuntijat lisäsivät myös, että omana työnä tehdyt liittymät tulisivat edullisemmiksi kuin nykyisin ostetut integraatiot. Kuitenkaan mielekkäänä ei ollut tehdä yritys X:stä it- tai ohjelmointiyritystä, joten helpommat integraatiot olisivat mielekästä tehdä itse ja haastavat syvempää ohjelmointiosaamista vaativat tilattaisiin edelleen kumppanilta. Vaikka yritys X:llä on oma it-tiimi, joka tekee myös käytössä olevat ohjelmistorobotit, tulisi liittymä osaamista, olla myös ostoreskontran järjestelmäasiantuntijoilla integraatioiden tekemiseen. Järjestelmäasiantuntija toimivat yleensä projekteissa ja tukevat prosessia ongelmatilanteissa, sekä ovat määrittelemässä liittymiä integraatioissa. Kun järjestelmäasiantuntijoilla olisi parempi osaaminen integraatioista onnistuisi myös liittymä määrittelyt paremmin ja olisi vähemmän tarvetta myöhemmin korjata tehtyjä liittymiä ja ongelmatilanteen selvitys onnistuisi paremmin.

Osaamista integraatioista pitäisi olla kaikenlaista, mutta tärkeimpänä ymmärrystä miten integraatiot ja eri teknologiat toimivat. Kun yrityksen asiantuntijoilla olisi parempi käsitys eri teknologian toiminnasta onnistuisi myös tilattavan integraation ja liittymän määrittely paremmin. Paremmalla määrittelyllä saataisiin säästöjä nopeammalla toteutus ajalla sekä sillä ettei liittymään tarvitsisi tehdä muutoksia puutteellisen tai virheellisen määrittelyn vuoksi. Kun ostoreskontran palvelutuotannon palveluasiantuntijat ymmärtäisivät miten ja millä syklillä integraatio toimii, he osaisivat vastata asiakkaan kysymyksiin ja ratkoa helppoja ongelmia. Osaaminen integraatioista helpottaisi ymmärrystä siitä miten eri järjestelmät ovat sidoksissa toisiinsa ja miten prosessi etenee tapahtumasta toiseen.

Kuvassa 9 on havainnollistettu minkälaista integraatio-osaamista ostoreskontran prosessissa tuli tutkimuksen mukaan olla. It-kumppani tekisi tarvittaessa uudet ja haastavammat ohjelmointia edellyttävät integraatiot. Järjestelmäasiantuntijoilla tuli olla riittävä osaaminen helppojen liittymien tekoon, sekä pääkäyttäjien ja prosessin tekijöiden tukemiseen, järjestelmäasiantuntija toimisivat myös yhteistyössä It-kumppanin puoleen. Pääkäyttäjät ymmärtävät integraatioiden toiminnan ja miten liittymät on rakennettu, sekä osaavat selvittää ongelmatilanteita ja pystyvät tukemaan prosessia ongelmatilanteissa, mikäli jokin liittymä ei ole toiminut. Prosessin palveluasiantuntijat ymmärtävät ylätasolla, miten integraatiot ovat ajastettu ja toimivat ylätasolla.



Kuva 9. Ostoreskontran integraatio-osaaminen

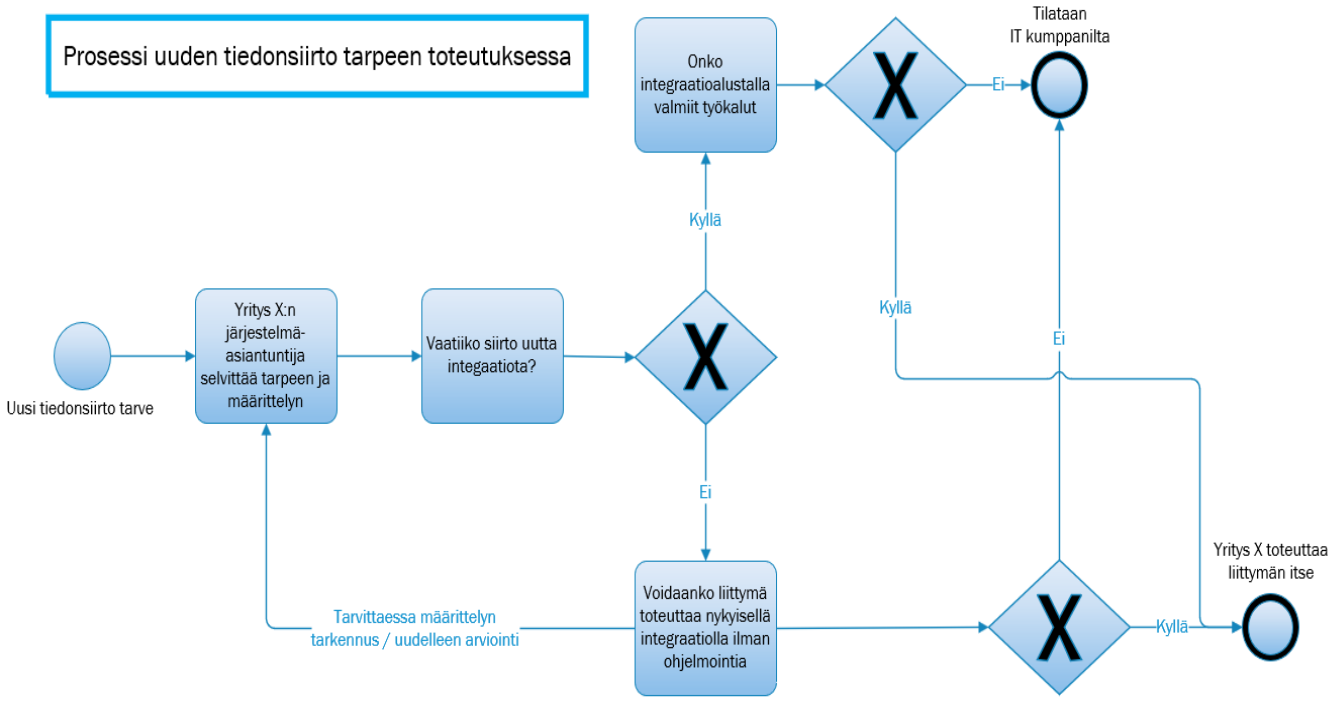
6 Tutkimuskysymyksiin vastaaminen

Opinnäytetyön tarkoituksena oli selvittää järjestelmäintegraation eri mahdollisuuksia yritys X:ssä ja löytää vastaukset tutkimuskysymyksiin. Tutkimuskysymyksillä pyrin löytämään vastaukset siihen mitä hyötyjä nykyaikaisella integraatioarkkitehtuurilla olisi mahdollista saavuttaa ja kuinka mahdollisen integraatioalustan hankinta olisi järkevä toteuttaa. Onko yritys X tehnyt itse integraatioita ja olisiko tarvittavaa osaamista niiden tuottamiseen.

Tutkimuskysymys K1. API-integraatio mahdollistaisi ostoreskontran prosessiin reaaliaikaisen, skaalautuvan tiedonsiirron. Ohjelmistorajapinnalla integraatio vie tai noutaa tiedon suoraan järjestelmän kantaan, erillisiä ajastettuja sisään luku tai tulostus tehtäviä ei tarvitsisi suorittaa. Toki API-liittymäkin voidaan ajastaa toimimaan tiettynä ajankohtana, jolloin sen ei tarvitse kokoaikaa olla tekemässä kyselyjä järjestelmässä. API-integraation myötä ostoreskontran prosessia voitaisiin miettiä paremmin toimivaksi, kun prosessi ei olisi niin riippuvainen liittymän ajastuksista.

Tutkimuskysymys K2. Integraatioalusta nähtiin järkevänä vaihtoehtona toteuttaa ostoreskontran integraatiot. ERP-järjestelmä kannattaisi kilpailuttaa, joko ilman integraatioalustaa tai niin, että alusta olisi tarjottavana optiona, jolloin olisi mahdollista tehdä teknologinen analyysi siitä kannattaisiko alusta ottaa ERP-järjestelmä toimittajalta vai ostaa muualta. Mikäli integraatioalusta tulisi osana ERP-järjestelmää tulisi siitä olla mahdollista tehdä integraatioita myös järjestelmiin, jotka eivät liity suoraan uuteen ERP-järjestelmään. Paras vaihtoehto olisi kilpailuttaa ja hankkia integraatioalusta omana hankintana, jolloin yritys X:llä olisi parempi hallinta sen käytössä ja mahdollisesti myös integraatioiden teko itse olisi mahdollista.

Tutkimuskysymys K3. Yritys X:llä on omaa osaamista integraatioiden tekemiseen, mutta integraatiotyö on ostettu valitun strategian mukaan it-kumppanilta. Vaikka yritys X:llä on osaajia, joilla riittäisi kompetenssi integraatioiden tekemiseen, ei niitä kuitenkaan ole tehty, joten aloitus vaatisi todennäköisesti vähintään perehtymistä asiaan. Integraatioita on haastavuudeltaan monen tasoisia ja integraatioalustan valmiilla työkaluilla liittymien teko onnistuisi myös ostoreskontran järjestelmäasiantuntijoilta alustan perehdytyksen jälkeen. Integraatio-osaamista ja tietoisuutta olisi kuitenkin hyödyllistä saada myös enemmän ostoreskontran prosessin tekijöiden tasolle. Kaikkiaan kaiken tasoista integraatio-osaamista tarvittaisiin yritys X:ssä lisää. Kun yritys X:llä olisi omaa integraatio-osaamista ja toteutettavat liittymät olisi jaettu tehtäväksi omana työnä ja It-kumppanilta ostettuna, voisi uuden tiedonsiirto tarpeen prosessi kulkea kuvan 10 mukaisesti.



Kuva 10. Prosessi uuden tiedonsiirto tarpeen toteutuksessa

6.1 Arviointi ja tutkimuksen luotettavuus

Tutkimuksen teoriapohjaan integraatioiden tarkoitukseen sekä eri tekniikoihin olen tutustunut kattavasti, lähteinä on kirjallisuutta sekä artikkeleita laajasti eri toimijoilta. Vaikka lähteistä ei löydy vertaisarvioituja tutkimuksia on aineistoja tutkittu laajasti eri lähteistä, joista on saatu kattava kuva saatavilla olevista yleisistä vaihtoehtoisista tekniikoista ja integraation toteutustavoista. Artikkeleja, blogeja, sekä eri IT-yritysten kirjoituksiin perehtymällä sain rakennettua yhtenäisen käsityksen parhaista käytännöistä. Pidän käyttämiäni lähteitä kattavana ja luotettavana teoriapohjana opinnäytetyölle.

Tutkimushaastattelut onnistuin pitämään suunnitellusti ja sain haastateltua kahdeksan asiantuntijaa, joilla oli hyvä käsitys ostoreskontran prosessista ja sen tarvitsemista integraatioista. Kokonaisuudessa pidän haastatteluja onnistuneina ja niistä sain hyvän kuvan, kuinka yritys X:ssä integraatioita on toteutettu. minkälaisia kehitysideoita asiantuntijoilta nousi haastatteluissa. Haastattelut tuovat tutkimukseen luotettavan kuvan integraatioista ostoreskontran prosessissa.

Havaintoja ei voida yleistää koskemaan alaa laajemmin, koska integraatioiden tarpeet vaihtelevat eri yrityksissä käytettävien järjestelmien kyvykkyydestä, sekä näkemyksestä kuinka integraatiot tulisi toteuttaa. Löydökset voidaan kuitenkin yleistää koskemaan yritys X:n muita prosesseja, mikäli muutoksia tehtäisiin ostoreskontran prosessissa, vaikuttaisi se välttämättä myös muihin prosesseihin ja toimintoihin.

6.2 Menetelmät ja tutkimusetiikka

Puolistrukturoitu teemahaastattelu toimi hyvin tiedonhankintamenetelmänä ja koin saavani riittävästi aineistoa. Litteroitua haastatteluaineistosta etsin toistuvia teemoja, joita purin korteille tutkittavaksi. Tekoälyä en käyttänyt haastatteluaineistoa analysoidessa, mutta uskoisin että se olisi voinut nopeuttaa aineiston työstämistä. Haastattelujen saamaani aineistoa olen peilannut kirjallisuuskatsaukseen ja tehnyt päätelmiä näiden perusteella, haastattelussa esiin nousseiden asioiden myötä olen myös täydentänyt teoriataustaa, jotta saisin paremman käsityksen erilaisista järjestelmäintegraatioiden mahdollisuuksista ja vaihtoehdoista. Yksilöhaastattelujen sijaan hyvä vaihtoehto tiedonhankintaan olisi voinut olla myös ryhmähaastattelu tai työpaja, jossa olisi voinut keskustelujen kautta tulla enemmän aineistoa ja uudenlaisia ideoita.

Opinnäytetyötä tehdessä on noudatettu hyviä tieteellisiä käytäntöjä, lähdeaineistoihin on viitattu asianmukaisella tavalla. Opinnäytetyön aihe sovittiin yhteistyössä Yritys X:n kanssa ja tutkimuksen aloituksesta on sovittu. Tiedonhankintamenetelmänä käytettyyn haastatteluun pyydetty lupa Yritys X:ltä ja haastatelluille asiantuntijoille on ennen haastattelua selvitetty mihin tarkoitukseen tuloksia käytetään. Yritys X on saanut tutustua ja lausua opinnäytetyöhön ennen julkaisua, jotta julkaistuksi ei tule liikesalaisuuksia.

6.3 Jatkokehitys

Uuden ERP-järjestelmän myötä kannattaisi miettiä myös koko integraatioarkkitehtuuri uudelleen ostoreskontran osalta. Liittymien suunnittelu integraatiossa, mitkä ovat prosessin kannalta kriittisiä, joiden halutaan toimivan reaaliaikaisesti, esimerkiksi toimittajatiedot ja missä kohtaa riittää ajastus kerran, esimerkiksi tilit ja laskentatunnisteet tai muutaman kerran päivässä esimerkiksi tiliöintisäännöt. Myös yritys X:n ostoreskontran prosessia mahdollista tarkastella uudelleen uuden integraatioarkkitehtuurin myötä, jotta saataisiin turhaa odotusaikaa pois prosessista ja näin toimintaa tehostettua, tämä vaatisi edelleen tarkemman kuvauksen prosessista ja pullonkaulojen etsintää.

Integraatio osaamista yritys X:ssä tulisi lisätä jotta, liittymiä tilatessa määrittelytyö onnistuisi paremmin. Ostoreskontran prosessin työntekijöille olisi tarpeen saada perusymmärrystä siitä, miten liittymät on toteutettu ja mitä ne mahdollistavat. Ymmärrys integraatioista voisi helpottaa päivittäistä työtä, koska kehitysideat nousevat usein prosessin tekijöiltä. Integraatiot on nykyisen strategian mukaisesti ostettu it-kumppanilta, liittymien rakentaminen kannattaisi tehdä niin sanotusti hybridimallilla, jolloin liittymiä voisi tehdä sekä itse, että tarvittaessa tilata it-kumppanilta.

Ohjelmistorobotiikka tarjoaa mahdollisuuksia etenkin yritys X:n legacy-järjestelmien tiedonsiirrossa, joista puuttuu ohjelmistorajapinnat. Yrityksellä on omaa robotiikka osaamista, jota kannattaisi käyttää hyödyksi enemmän myös tiedonsiirrossa helposti ohjelmistorobotiikalla toteutettavissa

kohteissa. Ohjelmistorobottien tekeminen ei vaadi ohjelmointiosaamista, joten bottien teon oppiminen voisi olla mahdollista myös pääkäyttäjillä tai järjestelmäasiantuntijoilla, jotka eivät kuitenkaan ole koodaavia osaajia. Näin myös automatisoinnin mahdollisuuksia voisi nousta enemmän esiin ja olisi mahdollista saada myös muuta kuin integraatioon liittyvää kehitystä.

7 Yhteenveto

Tämän opinnäytetyön tarkoituksena oli selvittää miten järjestelmäintegraatio on toimii yritys X:n ostoreskontrassa. Kokemukseni järjestelmäintegraatiosta ennen opinnäytetyön tekoa oli varsin pintapuolista, joka toi omat haasteensa tutkimuksen tekemiseen, mutta tutkimuksen teko oli opettavaista. Opinnäytetyön aiheet kuitenkin sivusivat osaamistani ja rooliani työpaikassani yritys X:ssä, mikä helpotti näkökulman löytämisessä. Tutkimus oli mielenkiintoinen, mutta osaamiseni integraatioteknologiasta asetti omat haasteensa. Haasteita tuli myös henkilökohtaisista ajankäytön mahdollisuuksista, koska tein opinnäytetyön työn ohella.

Minulla ei ollut juurikaan kokemusta järjestelmäintegraatiosta ennen opinnäytetyön tekoa, mutta sain hahmoteltua työhön mielekkään näkökulman ja lähestymistavan, jolla sain mielestäni kuvattua, kuinka yritys X:n ostoreskontrassa tehdään integraatioita ja miten niitä kannattaisi jatkossa tehdä. Tutkimustyö lisäsi osaamistani järjestelmäintegraatioissa, kun kirjoitin kirjallisuuskatsausta ja tutustuin lähdemateriaaleihin. Pääsääntöisesti aineisto oli englanninkielistä, sillä suomenkielisiä lähteitä on heikosti saatavilla.

Opinnäytetyössä toteutettiin haastattelututkimus, jolla oli tarkoitus saada kuva siitä, miten yrityksen integraatiot nykyisin toteutetaan ja miten ne olisivat tulevaisuudessa järkevä toteuttaa. Haastattelut onnistuivat mielestäni hyvin ja sain niiden avulla tarvittavat tulokset, nykytiedolla luultavasti kiinnittäisin enemmän huomiota haastattelutilaisuuteen ja aiheen alustukseen. En ole aiemmin tehnyt haastattelututkimusta ja sen suunnittelu ja toteutus oli mielenkiintoinen haaste, sain tästä hyödyllistä oppia, jota tulen käyttämään myös nykyisessä työssäni jatkossa.

Lähteet

Alfame (2018) *Järjestelmäintegraatio, mitä se on selkokielellä?* Available at: https://www.alfame.com/ajankohtaista/jarjestelmaintegraatio-mita-se-on-selkokielella (Accessed: 12 January 2025).

Amazon AWS (s.a.) *Mitä ovat mikropalvelut?* | *AWS*. Available at: https://aws.amazon.com/microservices/ (Accessed: 15 February 2025).

Babati, B. (2019) *Legacy System Integration*. Available at: https://www.youredi.com/blog/legacy-system-integration (Accessed: 13 October 2023).

Bigelow, S.J. (2023) *What is iPaaS? Guide to Integration Platform as a Service*. Available at: https://www.techtarget.com/searchcloudcomputing/definition/iPaaS-integration-platform-as-a-service (Accessed: 9 February 2025).

Bridgwater, A. (2019) *RPA series: Digital Workforce - does RPA integration beat APIs?* Available at: https://www.computerweekly.com/blog/CW-Developer-Network/RPA-series-Digital-Workforce-does-RPA-integration-beat-APIs (Accessed: 16 February 2025).

Chen, M. (2024) *What Is SOA (Service-Oriented Architecture)?* Available at: https://www.oracle.com/service-oriented-architecture-soa/ (Accessed: 1 February 2025).

Churchville, F. and Nolle, T. (2021) *What is Enterprise Service Bus (ESB)?* | *Definition from TechTarget*. Available at: https://www.techtarget.com/searchapparchitecture/definition/Enterprise-Service-Bus-ESB (Accessed: 1 February 2025).

Codeless Platforms (2023) *What Is An ESB (Enterprise Service Bus)? - Definition And Examples*. Available at: https://www.codelessplatforms.com/docs/what-is-an-esb/ (Accessed: 1 February 2025).

Cole, Z. (2022) *Integration as a Service (IaaS) - Definition and Benefits - Perspectium*. Available at: https://www.perspectium.com/blog/integration-as-a-service/ (Accessed: 16 February 2025).

CompTIA (s.a.) *IT Terminology - A Glossary of Tech Terms for Beginners* | *IT Career Center* | *CompTIA*. Available at: https://www.comptia.org/content/guide/information-technology-terminology#section3 (Accessed: 8 February 2025).

education-wiki (s.a.) *Mikä on SFTP? - Kattava opas SFTP: hen ja sen etuihin*. Available at: https://education-wiki.com/4452322-what-is-sftp (Accessed: 1 September 2023).

Gartner (2022) *Hype Cycle for Application Architecture and Integration, 2022* | *Jitterbit*. Available at: https://info.jitterbit.com/gartner-hype-cycle-report.html (Accessed: 17 April 2025).

Gartner (2024) *Hype Cycle for Artificial Intelligence, 2024* | *annotated by SHIH YEN*. Available at: https://readwise.io/reader/shared/01j346cwkhtw8xqbazpgy6mx99/ (Accessed: 18 April 2025).

GeeksforGeeks (2025) *Monolithic vs. Microservices Architecture - GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/monolithic-vs-microservices-architecture/ (Accessed: 15 February 2025).

Gillis, A.S. (2024) *What is robotic process automation (RPA)?* | *Definition from TechTarget*. Available at: https://www.techtarget.com/searchcio/definition/RPA (Accessed: 16 February 2025).

Gillis, A.S. (2025) *Top 10 essential skills for ERP professionals in 2025* | *TechTarget*. Available at: https://www.techtarget.com/searcherp/feature/Essential-skills-for-ERP-professionals (Accessed: 8 February 2025).

Griffith, B. (2024) *Point-to-Point Integration: When to Use It and When Not To*. Available at: https://www.workato.com/the-connector/point-to-point-integration-drawbacks/ (Accessed: 16 January 2025).

Harris, C (s.a.) *Microservices vs. monolithic architecture* | *Atlassian*. Available at: https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith (Accessed: 15 February 2025).

Hophe, G (2003) *Hub and Spoke [or] Zen and the Art of Message Broker Maintenance - Enterprise Integration Patterns*. Available at: https://www.enterpriseintegrationpatterns.com/ramblings/03_hubandspoke.html (Accessed: 21 January 2025).

Kuntz, M (2021) *iPaaS or ESB?* Available at: https://blog.seeburger.com/ipaas-vs-esb-which-integration-approach-is-best-for-your-organisation/ (Accessed: 2 February 2025).

Luukka, E (2017) *RPA vs integration: The differences between the solutions*. Available at: https://digitalworkforce.com/rpa-news/robotic-process-automation-vs-integration-2/ (Accessed: 29 April 2025).

McDonald, J (2023) *What is Integration as a Service (IaaS)? A definition from WhatIs.com*. Available at: https://www.techtarget.com/searchcloudcomputing/definition/integration-as-a-service (Accessed: 29 April 2025).

Moilanen, J. Niinioja, M. Seppänen, M. Honkanen, M. (2018) *API talous 101*. Alma Talent Oy.

Muthukrishnan, H. (2025) *Enterprise Integration: The Past, Present And AI-Driven Future*. Available at: https://www.forbes.com/councils/forbestechcouncil/2025/05/02/enterprise-integration-the-past-present-and-ai-driven-future/ (Accessed: 2 May 2025).

O'Brien, R. (2008) *Integration Architecture Explained - HubPages*. Available at: https://discover.hubpages.com/business/Integration-Architecture-Explained (Accessed: 13 October 2023).

Ojasalo, K, Moilanen, T ja Ritalahti, J (2014) *Kehittämistyön menetelmät : uudenlaista osaamista liiketoimintaan*. 3. uudistettu painos. [Helsinki] : Sanoma Pro.

Olli, S (2022) *Kuinka varmistat järjestelmäintegraation onnistumisen - Tieturi*. Available at: https://www.tieturi.fi/blogi/kuinka-varmistat-jarjestelmaintegraation-onnistumisen/ (Accessed: 8 February 2025).

Parikh, A (2022) *RPA Managed Services - How Does RPA Work & Where It Can be Used?* Available at: https://www.silvertouch.ca/blog/introduction-to-rpa/ (Accessed: 1 May 2025).

Postman (s.a.) *What is an API? A Beginner's Guide to APIs | Postman*. Available at: https://www.postman.com/what-is-an-api/ (Accessed: 12 April 2025).

Raatikainen, P (2025) *iPaaS vs. Integration as a Service*. Available at: https://www.oneio.cloud/blog/ipaas-vs-integration-as-a-service?utm_source=chatgpt.com (Accessed: 14 February 2025).

Red Hat (2019a) *What does an API gateway do?* Available at: https://www.redhat.com/en/topics/api/what-does-an-api-gateway-do (Accessed: 16 February 2025).

Red Hat (2019b) *What is API management?* Available at: https://www.redhat.com/en/topics/api/what-is-api-management (Accessed: 16 February 2025).

Rose-Collins, F. (2024) *Integraatio palveluna -palvelun (IaaS) määritelmä ja edut*. Available at: https://www.ranktracker.com/fi/blog/definition-and-advantages-of-integration-as-a-service-iaas/ (Accessed: 1 May 2025).

Russom, P. (2008) *Data Integration Architecture: What It Does, Where It's Going, and Why You Should Care | TDWI*. Available at: https://tdwi.org/Articles/2008/05/27/Data-Integration-Architecture-What-It-Does-Where-Its-Going-and-Why-You-Should-Care.aspx?Page=1 (Accessed: 25 January 2025).

Santos, R (2016) *Hub-and-Spoke Architecture • Polarising*. Available at: https://polarising.com/hub-spoke-architecture/ (Accessed: 12 January 2025).

Seth, K (2025) *Breaking Barriers: Innovations in AI and System Integration*. Available at: https://www.analyticsinsight.net/artificial-intelligence/breaking-barriers-innovations-in-ai-and-system-integration (Accessed: 2 May 2025).

Siltaoja, M ja Sorsa, V (2020) *Laadullisen tutkimuksen näkökulmat ja menetelmät*, *Laadullisen tutkimuksen näkökulmat ja menetelmät*. Edited by P. Juuti and A. Puusa. [Helsinki] : Gaudeamus.

Smartbridge (2024) *RPA vs. Systems Integration and API Integration - Smartbridge*. Available at: https://smartbridge.com/rpa-vs-systems-integration/ (Accessed: 29 April 2025).

SnapLogic (s.a.) *Integration Architecture – Explanation & Overview*. Available at: https://www.snaplogic.com/glossary/integration-architecture (Accessed: 29 March 2025).

Das, S. (2023) *system integration using RPA*. Available at: https://intelgic.com/insights/system-integration-with-robotic-process-automation-rpa-tool-bridging-the-gap-between-legacy-systems-and-modern-software/ (Accessed: 29 April 2025).

SRM Technologies (2022) *What is iPaaS (Integration Platform as a Service)?* Available at: https://www.srmtech.com/knowledge-base/blogs/what-is-ipaas-integration-platform-as-a-service/ (Accessed: 9 February 2025).

Tähtinen, S. (2005) *Järjestelmäintegraatio : tarve, vaihtoehdot, toteutus*. Helsinki : Talentum (Tekniikka & talous).

Vamenture (2025) *Advantages and Disadvantages of API Integration for Business*. Available at: https://www.vamenture.com/blog/advantages-and-disadvantages-of-api-integration-for-business (Accessed: 16 February 2025).

Waldron, R. (2024) *iPaaS in an AI-driven world | InfoWorld*. Available at: https://www.infoworld.com/article/3481482/ipaas-in-an-ai-driven-world.html (Accessed: 2 May 2025).

Yasar, K. ja Ehrens, T. (2022) *What is System Integration? Definition, Methods, Challenges | TechTarget*. Available at: https://www.techtarget.com/searchcustomerexperience/definition/integration (Accessed: 2 February 2025).

Liitteet

Liite 1. Haastattelukysymykset

Ostoreskontran integraatiot

Anssi Junttila, opinnäytetyö

1.3.2025

Haaga-Helia

Nykytila

- Miksi integraatioita tehdään?
- Tehdäänkö integraatioita itse vai ostetaanko ne?
- Paljonko ostoreskontrassa on integraatioita?
- Miten ne on toteutettu?
  - Esim. api, sftp, rpa?
- Osaatko kuvailla miten Osren integraatiot on nykyisin tehty, esim. keskitetty hubin kautta / hajautettu point-to-point?

Teknologiat

- Mitä hyötyjä olisi keskitetyssä integraatiossa vrt. point-to-point integraatiot?
  - Mitä riskejä?
- Mitä hyötyjä näet api rajapinnan käytössä vs. sftp (flatfile)- tiedonsiirrossa?
  - Onko jotain haittoja?
- Mitä mieltä olet integraatioalustasta jolla voitaisiin hoitaa kaikki Osren integraatiot?
  - Mitä riskejä?

Uuden erp:n hankinta.

- Pitäisikö integraatiot
  - Ostaa osana erp-hankintaa?
  - Kilpailuttaa erillisenä hankintana?
- Pitäisikö erp-toimittajan alustan kautta pystyä tekemään integraatioita myös järjestelmiin jotka eivät liity uuteen erp-järjestelmään?
- Paljonko integraatiot maksaa? Kuinka suuri osuus erp-hankinnasta on integraatiota?

Osaaminen

- Onko omaa integraatio osaamista
  - Pitäisikö olla ja minkälaista?
- Ostettu osaaminen
  - Mitä riskejä