

# **WEB-SOVELLUS MIKROPALVELUARKKITEHTUURILLA**

CASE: Helvar Supplier Portal -sovellus



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

Hämeenlinna, syksy 2019

Harriet Harsto

Tietojenkäsittelyn koulutusohjelma  
Hämeenlinna

---

<b>Tekijä</b>	Harriet Harsto	<b>Vuosi</b> 2019
<b>Työn nimi</b>	Web-sovellus mikropalveluarkkitehtuurilla	
<b>Työn ohjaaja</b>	Tommi Lahti	

---

## TIIVISTELMÄ

Opinnäytetyön tavoitteena oli raportoida ohjelmistoprojekti, joka toteutti web-sovelluksen toimeksiantajan alihankintatuotteiden ja raaka-aineiden tilaustarpeen osoittamiseksi. Sovelluksen tuli rakentua niin, että se oli helpposti tai suhteellisen pienellä työllä päivitettävissä tukemaan paitsi sovelluksen jatkokehitystä, myös komponenttien mahdollisia muutoksia. Opinnäytetyön tarkoituksena oli kerätä product backlogia eli kehitysjonolistaa, jota käytetään sovelluksen edelleen kehittämisessä ensimmäisen version julkaisun jälkeen. Opinnäytetyö raportoi sovelluskehitysprojektin vaiheita projektipäiväkirjan muodossa.

Projektin lopputuotteena julkaistiin mikropalveluarkkitehtuurilla rakennettu hybridipilvijärjestelmä, jossa on-premise-ohjelmiston tiedot kutsutaan API-rajapinnan avulla web-käyttöliittymään. Opinnäytetyössä tutkitaan pilviratkaisuja, Azuren työkaluja, ohjelmistokehitystä ja SAP-integraatiota.

Projektin lopputulos, Supplier Portal -web-sovellus, vastasi työn toimeksiantajan, Helvar Oy Ab:n vaatimuksia. Kehitysjonolistalla on tärkeitä kehityskohteita, joita voidaan priorisoida sovelluksen käyttäjien palautteen ja toimeksiantajan tarpeiden perusteella.

**Avainsanat** mikropalveluarkkitehtuuri, product backlog, pilvipalvelut, Azure, API

**Sivut** 34 sivua, joista liitteitä 4 sivua

Degree Programme in Business Information Technology  
Hämeenlinna

---

<b>Author</b>	Harriet Harsto	<b>Year</b> 2019
<b>Subject</b>	Web application based on microservices architecture	
<b>Supervisors</b>	Tommi Lahti	

---

#### ABSTRACT

This thesis introduces a web application development project. The web application is used by the client's supply team to order product resources from supply vendors. The requirement was to build the application on a server free, platform-based cloud service that is future proof and easily developed and updated for further user needs.

As a main goal, this thesis project introduces a web application that is designed on microservice architecture. Being a hybrid cloud solution, this Microsoft Azure cloud-based application requests data from on-premise software via an API interface into the user interface.

Another goal was to collect a product backlog of development requests that appeared during the project and can be developed into features after the first version is published. Being a project management report, the thesis also includes a project diary. The thesis also introduces technologies behind the web application, including cloud services, Azure tools, software development and API interfaces.

The web application called Supplier Portal meets the requirements of the client, Helvar Oy Ab. As a result, the product backlog includes many important development requirements, that can be prioritized and groomed based on user feedback and business needs.

**Keywords** microservice architecture, cloud services, Azure, API, product backlog

**Pages** 34 pages including appendices 4 pages

## SISÄLLYS

1	JOHDANTO .....	4
2	TOIMEKSIANTAJA .....	5
3	OHJELMISTOKEHITYS.....	6
3.1	Product backlog .....	6
3.2	Supplier Portal -projektimenetelmä .....	8
4	SAP-TOIMINNANOHJAUSJÄRJESTELMÄ .....	9
4.1	Materiaalinhallinta.....	9
5	PILVIPALVELUT .....	12
5.1	PaaS eli Platform as a Service .....	14
5.2	Hybridipilvi .....	14
5.3	Mikropalveluarkkitehtuuri .....	14
5.4	Microsoft Azure Blob Storage .....	15
5.5	Microsoft Azure Logic Apps .....	16
5.6	Microsoft Azure AD.....	18
6	API-OHJELMOINTIRAJAPINNAT .....	19
6.1	Supplier Portal -rajapinta .....	19
7	TULOKSET .....	21
7.1	Projektin taustat .....	21
7.2	Vanha ratkaisu .....	21
7.3	Uusi Supplier Portal -sovellus.....	22
7.4	Supplier Portal Product Backlog.....	24
7.5	Projektinhallinta.....	25
8	YHTEENVETO .....	27
8.1	Tulevaisuudennäkymiä .....	27
	LÄHTEET .....	29

### Liitteet

Liite 1      Supplier Portal -projektipäiväkirja

## 1 JOHDANTO

Opinnäytetyöni johdattaa lukijan ohjelmistoprojektiin, jossa kehitetään toimeksiantajalle uusi verkkoselaimessa toimiva käyttöliittymä. Opinnäytetyössäni tarkastelen projektia ikään kuin kahdesta roolista; toimeksiantajan eli tilaajan edustajana projektijohtajan ominaisuudessa, sekä tietojenkäsittelyn tradenomiopiskelijana, joka haluaa tutkia tarkemmin lopputulokseen vaikuttavia teknologisia valintoja. Opinnäytetyössäni esittelen tiivistetysti projektin lopputuloksen kannalta tärkeimpiä käsitteitä sekä erilaisiin teknologiaratkaisuihin johtaneita päätöksiä. Pidän projektipäiväkirjaa, jonka perusteella lukija saa kuvan projektin vaiheista, ja pääsee seuraamaan kehittymistäni projektin kokonaiskuvan hallinnassa. Lukija saa myös käsityksen siitä, miten lopputuloksena valmistuva käyttöliittymä on rakentunut ja millainen lopputuloksesta tulee.

Projektin tavoitteena oli toteuttaa Supplier Portal -nimellä kulkeva web-sovellus, jonka avulla toimeksiantaja voi tehdä tilauksen toimittajiltaan helposti, vaivattomasti ja turvallisesti. Sovelluksen tuli olla rakentunut niin, että se on helposti ja/tai suhteellisen pienellä työllä päivitettävissä tukemaan paitsi sovelluksen jatkokehitystä, myös komponenttien mahdollisia muutoksia.

Opinnäytetyöraporttini esittelee ensin projektin taustat. Tämän jälkeen kappaleissa 3-6 käydään läpi projektissa käytettäviä teknologioita ja perustellaan, miksi ne valittiin projektin toteutukseen ja miten toteutus suunniteltiin. Kappaleessa 7 kuvataan projektin lopputulos ja lopuksi koostetaan projektin yhteenveto ja pohditaan seuraavia kehitysmahdollisuuksia. Projektipäiväkirja löytyy liitteestä 1.

Opinnäytetyön tutkimuskysymykset ovat:

- *Miten käyttöliittymät toteutetaan?*
- *Miten integraatiotyönkulut toteutetaan?*
- *Miten yhteydet SAP ERP -järjestelmään toteutetaan?*

## 2 TOIMEKSIANTAJA

Opinnäytetyön toimeksiantaja on Helvar Oy Ab. Helvar on vuonna 1921 perustettu suomalainen perheyritys, joka valmistaa valaisinkomponentteja sekä valonohjausjärjestelmiä ja -palveluita. Yrityksen pääkonttori Espoossa, elektroniikkatehdas Karkkilassa ja osaamiskeskukset Englannissa sekä Tukholmassa. Yli 90 % Helvarin valmistamista tuotteista viedään Suomen ulkopuolelle.

Toimeksianto tehtiin Helvarin valaisinkomponentteja valmistavalle Lighting Essentials -liiketoimintayksikölle. Yksikön Karkkilassa toimiva tehdas valmistaa LED-liitäntälaitteita, LED-moduuleja, elektronisii liitäntälaitteita ja magneettisia kuristimia. Tehdas tuottaa vuodessa satoja tuhansia elektroniikkatuotteita. Raaka-aineita hankitaan useista eri kanavista. Alihankintakumppaneita on Suomessa, Euroopassa ja Kiinassa. Hankintaosastolla työskentelee 5 henkilöä.

Helvarin IT-toiminnoista vastaa IT Services & Operations Lead -tittelillä toimiva henkilö, joka toimi tämän opinnäytetyön ohjaajana toimeksiantajan puolelta. Työssään hänen apunaan toimii lähitukea antava ICT Support -henkilö. Tietoliikenneinfrastruktuuri pääpiirteittäin on ulkoistettu ulkoiselle toimijalle, joka huolehtii mm. palvelimista, päätelaitteista, käyttötuesta, tietoliikenneyhteyksistä sekä erilaisista yrityksen toiminnan tueksi käytettävistä sovelluksista ja ohjelmistoista, joita käytetään laajasti mm. tuotannossa, hankinnassa, myynnissä ja taloushallinnossa.

### 3 OHJELMISTOKEHITYS

Ohjelmistokehitys (myös sovelluskehitys, ohjelmistotuotanto) tarkoittaa tietokoneohjelman valmistamista ja sen dokumentointia asiakastarpeen ja liiketoimintatarpeen perusteella.

Ohjelmistokehityksessä on valitusta suunnittelumallista riippumatta usein samankaltaisia vaiheita, jotka voidaan jakaa pienempiin osiin. Ensimmäisenä asiakkaan vaatimukset ohjelmiston toiminnallisuuksista, jonka jälkeen ohjelmisto suunnitellaan teknisesti vaatimusten mukaiseksi. Implementointivaiheessa suunnitelma toteutetaan, eli ohjelmoidaan ja dokumentoidaan, jonka jälkeen se testataan ja käyttöön otetaan verifiointi- ja validointivaiheessa. Ylläpitovaihe voi sisältää päivityksiä ja versionhallintaa tai vain varmistaa tuotteen sujuvan toiminnan elinkaarensa loppuun. (Salminen, 2018)

Ohjelmistosuunnitteluun on vuosien kuluessa kehitetty erilaisia prosessimalleja, joiden avulla voidaan paremmin hallita ohjelmistoprojektin kulkua ja varmistua määrittelyn mukainen lopputuote. Prosesseilla on paljon yhteneväisyyksiä, mutta suurin erotus on siinä, työskennelläänkö mallissa iteratiivisesti vai eteneekö se lineaarisesti eteenpäin vaihe kerrallaan. (Salminen, 2018)

Ensimmäisenä varsinaisena ohjelmistoprosessimallina pidetään 70-luvulla kehitettyä vesiputousmallia, jossa työvaiheet seuraavat tiettyä järjestystä. Vesiputousmalli tai sen variantit ovat edelleen käytössä suurimmassa osassa ohjelmistoyrityksiä, ja sen teoria, vahvuudet ja heikkoudet tunnetaan hyvin. (Salminen, 2018)

Ketterät menetelmät eli agile development -menetelmät perustuvat jatkuvaan kehittämiseen, iterointiin ja prototyyppeihin. Tunnetuimpia malleja on Japanista lähtöisin oleva Scrum, joka ei ole varsinaisesti menetelmä, vaan enemmän viitekehys tai toimintatapa, jonka sisällä voi olla erilaisia menetelmiä, työkaluja ja prosesseja. Scrumissa ohjelmistokehitys rakentuu erimittaisten syklien ympärille, joista tärkeimmät ovat sprintti ja päivä. Sprintillä tarkoitetaan vaihtelevan mittaista kehitysjaksoa, jonka jälkeen tuote on ainakin periaatteessa julkaisuvalmis. (Salminen, 2018)

#### 3.1 Product backlog

Product backlog, eli tuotteen kehitysjono, on ketterän ohjelmistokehityksen ohjaamiseen tarkoitettu tapa määrittellä, arvioida ja priorisoida tuotteeseen ehdotettuja toiminnallisuuksia liiketoiminnan tarpeiden mukaan. Product backlog muodostetaan user storyista, muutamaan lauseeseen tiivistetyistä käyttötarinoista, joilla kuvataan paitsi erilaiset käyttäjätarpeet, myös asiakkaalle arvokkaat toiminnat. User storyt ovat

selkokielisiä ja helposti ymmärrettäviä ja yleisesti käytettyjä ketterissä kehitysmenettelyissä. Tarinasta selviää aktori, eli kuka tai mikä tekee, mitä tehdään ja mahdollisesti mitä lisäarvoa tekeminen tuottaa käyttäjälle. (Salminen, 2018)

User storyn voi määritellä esimerkiksi käyttämällä seuraavaa mallia, jonka avulla määritellään niin aktori, toiminnallisuus kuin lisäarvokin (Salminen, 2018):

As a <TYPE OF USER>,  
I want to <DO SOMETHING>  
so that I can <DESIRED BENEFIT>

Esimerkki Supplier Portalin product backlogin user storysta edellä mainitun mallin mukaisesti:

”Hankintaspesialistina haluan tallentaa ennustenäköymän pdf-dokumenttina.”

Product backlog voi olla alustava tai priorisoitu. Alustava product backlog koostuu listasta user storyja, joiden ei ole tarkoitus vielä rakentaa täydellistä listaa vaatimuksista, vaan niitä voidaan tarkentaa vielä matkan varrella. Kun alustava lista user storyistä on kerätty, ne järjestetään tärkeysjärjestykseen eli priorisoidaan ja työmäärä arvioidaan karkeasti. (Salminen, 2018)

User storyjen toteuttamiskelpoisuutta voidaan arvioida *INVEST*-periaatteella. *Independent* tarkoittaa, että user storyjen tulisi olla mahdollisimman itsenäisiä ja toisistaan riippumattomia ja *negotiable* kertoo user storyn muokattavuudesta. *Valuable to user or customer* eli asiakkaalle tai käyttäjälle tuotettu arvo tulee toteutua ja user storyn toteuttamisesta koituva työmäärä tulee olla arvioitavissa eli *estimable*. *Small* tarkoittaa, että user story on tarpeeksi pieni; mikäli viikon työpanos ylittyy, on hyvä pohtia toiminnallisuuden jakamista vielä pienempiin osiin. Viimeinen tarkistuskohta on *testable*, eli user story tulee olla toteutettu hyväksyttävästi. *INVEST*-kohtiin vastataan yksinkertaisesti kyllä tai ei. Mikäli jokin kohdistaa saa vastauksen ei, pitää user storyn toteuttamiskelpoisuutta miettiä uudelleen. Jos kaikkiin kohtiin saadaan vastaus kyllä, tulee seuraavaksi priorisoida user storyt tärkeysjärjestykseen. (Cohn, 2009)

Hyvä product backlog on yksinkertainen työkalu, mutta huonosti toteutetuna ne saattavat olla liian pitkiä, yksityiskohtaisia tai vaikeita käyttää. Pichlerin (2010) määritelmä hyvälle backlogille on ”DEEP”, joka on lyhenne sanoista *Detailed appropriately, Estimated, Emergent* ja *Prioritised*. *Detailed appropriately* tarkoittaa, että prioriteettijärjestyksessä korkeimmalla olevat user storyt ovat suhteellisen pieniä ja tarkemmin arvioituja, kun taas alemman prioriteetin user storyt voivat olla vielä isompia ja karkeampia. *Estimated* viittaa samaan asiaan kuin *INVEST*-periaatteessa, eli



user storyn työmäärän arvioitavuuteen. Emergent viittaa product backlogin jatkuvasti muuttuvaan luonteeseen: listalla jo olevia user storyja muokataan, uudelleenarvioidaan, hiotaan tai poistetaan ja uusia user storyja lisätään käyttäjien ja asiakkaiden palautteen mukaisesti. Tämän takia product backlogia huolletaan (backlog grooming) poistamalla vanhat tai tarpeettomat user storyt, lisätään uusia, pilkkotaan isompia user storyja tarpeen mukaan pienemmiksi sekä tarkastellaan ja arvioidaan sekä uusia että vanhoja user storyja projektin edetessä ja ymmärryksen kasvaessa. (Pichler, 2010)

### 3.2 Supplier Portal -projektimenetelmä

Supplier Portal -projektin lähtökohtana oli valmis sovellus, joka haluttiin rakentaa toiminnallisesti samanlaiseksi mutta uudella teknologialla ja arkkitehtuurilla. Projekti haluttiin pitää kevyenä ja ketteränä, ja se noudatti Minimum Viable Product -menetelmän periaatetta. Minimum Viable Product eli MVP tarkoittaa ohjelman ensimmäistä, käyttökelpoista versiota. Ensimmäiseen versioon identifioidaan toiminnallisuuden kannalta vähimmäisvaatimukset. (Olsen, 2015)

Supplier Portal -sovelluksen osalta vähimmäisvaatimus oli toistaa nykyinen toiminnallisuus uudessa ympäristössä uudelleen arkkitehtuuriin perustuen. Projekti päätettiin toteuttaa nopealla aikataululla niin, että ensimmäisessä vaiheessa tuotetaan loppukäyttäjälle toiminnallisesti samanlainen ympäristö aiemman version kanssa, ja sovelluksen varsinaista toimintaa kehitetään vasta seuraavassa vaiheessa. Viikon välein projektipalaverissa sovittiin sprintin omaisesti seuraavat vaiheet, tehtävät ja vastuuhenkilöt, ja käytiin läpi edellisen palaverin sovitut asiat ja projektin eteneminen.

Projektin edetessä esiin tulleet jatkokehitysehdotukset kerättiin alustavaan product backlogiin. Kun MVP-julkaisu eli ensimmäisen vaiheen kehitystyö on saatu valmiiksi, voi toimeksiantaja arvioida backlogin kehityskohteita INVEST-mallin mukaisesti ja priorisoida listalta tärkeimmät kohteet priorisoiduksi product backlogiksi seuraavaan kehityssprinttiin. Supplier Portal -projektin aikana kerätty kehitysjonolistaus on esitelty kappaleessa 7.4.

## 4 SAP-TOIMINNANOHJAUSJÄRJESTELMÄ

SAP SE on saksalainen yritys, jonka SAP-nimellä tunnetut tuotteet ovat maailman eniten käytettyjä ERP-järjestelmiä. SAP tulee sanoista Systems Applications and Products in data processing. ERP, enterprise resource planning, eli suomeksi toiminnanohjausjärjestelmä, on yrityksen tietojärjestelmä, joka tuottaa ja integroi tietoa yrityksen eri liiketoimintaprosesseista, esimerkiksi tuotannosta, jakelusta, alihankinnasta, varastonhallinnasta, kirjanpidosta ja laskutuksesta. (SAP, 2019a)

SAPin toimintaperusteisiin kuuluu välitön ja tehokas tiedon siirtäminen. Käytännössä tämä tarkoittaa, että integraatiot SAPin erilaisten moduulien kesken ovat reaaliaikaisia, ja muutos toisessa moduulissa näkyy välittömästi siihen linkitettyssä toisessa moduulissa. (Kogent Learning Solutions, 2010, s. 2)

Helvarin käyttämä SAP-versio on SAP ERP NetViewer, ja se on ns. On-Premise-palvelu, eli yrityksen omalle palvelimelle asennettu ohjelmisto. Se on ollut käytössä vuodesta 2008, ja SAP SE on ilmoittanut lakkauttavansa kyseisen SAP ERP -järjestelmän ylläpidon, kehittämisen ja tukipalvelut vuoteen 2025 mennessä (SAP, 2019b). Helvarin lähitulevaisuuteen on siis jo suunniteltu toiminnanohjausjärjestelmän uudelleenkilpailutus ja samalla on hyvin todennäköistä, että valittava palvelu tulee olemaan Software-as-a-Service, eli pilvessä sijaitseva SaaS-ohjelmisto, jota käytetään internetin välityksellä.

Helvar käyttää SAP ERP -ympäristöä monipuolisesti, joista tähän opinnäytetyöhön rajataan mukaan vain projektiin liittyvä materiaalinhallintamoduuli MM.

### 4.1 Materiaalinhallinta

MM- eli Material Management -moduulissa hallitaan yrityksen materiaalin hankintaa. Moduulissa käsitellään pääsääntöisesti materiaaleja (raaka-aineita) ja alihankkijoiden ja/tai toimittajien master-tietoja. Moduulissa hallitaan myös materiaalivarastojen prosesseja, kuten inventointia, sisäänostoja ja uusintatilauksia. MM-moduuli varmistaa, että vaadittavaa tuotetta tai raaka-ainetta on saatavilla riittävästi tietyllä hetkellä. (Kogent Learning Solutions, 2010, s. 21)

Helvarilla MM-moduuli tuottaa tietoa paitsi varastolle ja tuotannonsuunnittelulle, myös myynnille. MM-moduulin perusteella myyntiorganisaatio saa tietoa tuotteiden saatavuudesta tietyinä ajankohtana. Yritys käyttää MM-moduulia raaka-aineiden hankinnan lisäksi toimittajilta tilattavien valmiiden tuotteiden hankintaan, johon Supplier Portal -sovellus on integroitu.

Supplier Portal -sovelluksen perustana on toimittajakohtainen ennustetaulu (kuva 1), jota hallitaan MF50-toiminnalla. Ennustetaulu kertoo kullekin tuotteelle tai raaka-aineelle eli nimikkeelle viikkokohtaisen ennustetun tarpeen, ikään kuin suunnitelman tuotteen halutusta saatavuudesta määriteltynä ajankohtina toimittajalta. Kun hankinnan kohteena oleva nimike perustetaan SAPIin, sille voidaan määritellä linkitys Supplier Portalin. Tämän jälkeen MF50-ennustetaulussa on erikseen rivit tarvemäärille, joiden tulee näkyä Supplier Portalissa. Kun tarve on tallennettu SAPIin, tapahtuu niin sanottu ”Call Off” eli kotiinkutsu – tällä tarkoitetaan sitä vahvistettua määrää, jota hankittavaa tuotetta/materiaalia tarvitaan toimitettavaksi kullakin viikolla. Supplier Portalin ennustenäkymä on rakennettu selkeyttämään hankittavien tuotteiden saapumisennusteiden kokonaishallintaa, sillä vastaavaa raporttia ei suoraan SAPista saa aikaiseksi.

**Planning Table for Repetitive Manufacturing: Change Mode**

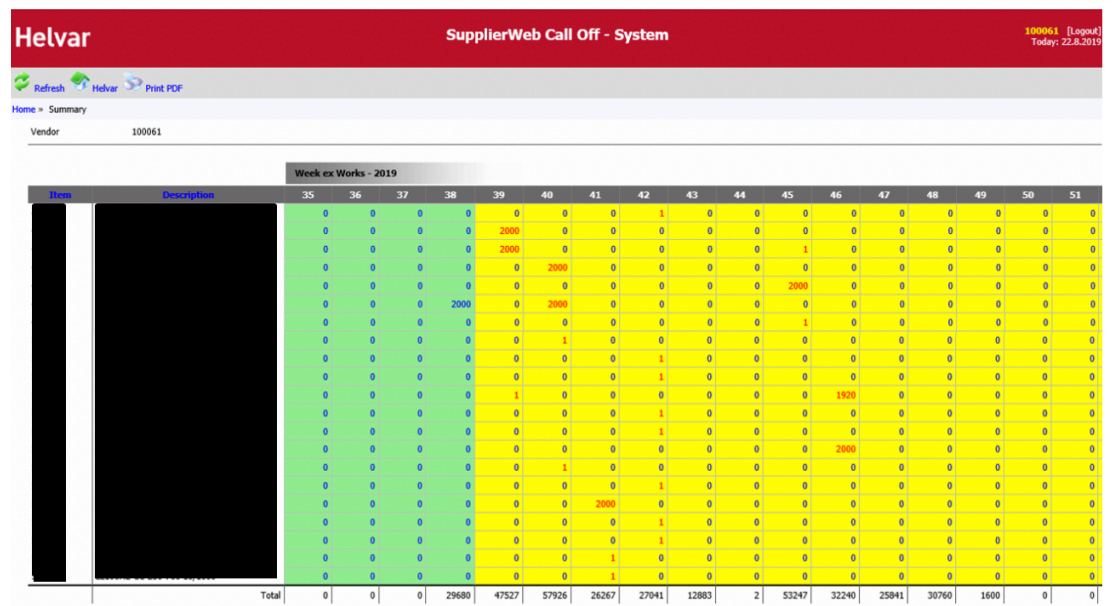
Material	Un	043 / 2019	044 / 2019	045 / 2019	046 / 2019	047 / 2019	048 / 2019	049 / 2019	050 / 2019	051 / 2019	052 / 2019	053 / 2019
001	%				4,057	12,748	34,233	7,69	8,338	6,344	0,001	5,002
Required - capacity	H				5,68	17,85	47,93	10,77	11,67	8,88	0,002	5,6
Available - capacity	H	140	140	140	140	140	140	140	140	140	140	112
WH/001	%				1,333	4,349	15,468	3,055	2,851	2,987		
Required - capacity	H				4	13,05	46,4	9,17	8,55	8,96		
Available - capacity	H	300	300	300					300	300	300	240

Material data	Un	043 / 2019	044 / 2019	045 / 2019	046 / 2019	047 / 2019	048 / 2019	049 / 2019	050 / 2019	051 / 2019	052 / 2019	053 / 2019
Total reqmts	PC	900	900	900	900	900	900	900	900	0,001	0,001	
Available quantity	PC	-6738	-7638	-6538	4562	7662	8762	9862	8962	8962	8962	896
1000	PC											
2000 WH	PC						2000	2000				
Other receipts	PC			2000	12000	4000						
Available quantity (fixe)	PC	-6738	-7638	-6538	4562	7662	8762	9862	8962	8962	8962	896
Requirements (fixed)	PC	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,00
StLoc: not pind 0009	***											
Total reqmts	PC											
Available quantity	PC				2000	2000	4000	4000	4000	4000	4000	400
1000	PC				2000		2000					
2000 WH	PC											
Not yet assigned	PC											
Available quantity (fixe)	PC				15662	15662	9862	9862	9862	9862	9862	986
Requirements (fixed)	PC				0,001	0,001	0,001	0,001	0,001	0,001	0,001	0,00
StLoc: not pind Q009	***											
Total reqmts	PC											

Kuva 1. MF50-ohjelman nimikekohtainen ennustenäkymä (nimike- ja toimittajatiedot peitetty)

Call Offin jälkeen hankintaspecialisti ilmoittaa toimittajalle, että tarvemäärä on vahvistettu. Vahvistusta/ilmoitusta ei pysty tekemään automaattisesti, vaan se vaatii manuaalisen sähköpostin kirjoittamisen. Saatuaan vahvistusviestin, toimittaja kirjautuu toimittajatunnuksellaan Supplier Portaliin, ja tallentaa itsellensä vahvistetut tarpeet avautuvasta näkymästä (kuva 2). Ennustetaulua ei saa SAPista tulostettua excel-muotoon, joten Supplier Portalista tulostettava selkeä taulukko on sekä toimittajalle, että hankintaspecialistille helposti seurattava muoto. Vihreät sarakkeet ovat vahvistettuja tarpeita. Ennustenäkymä on viikkokohtainen, ja viikkonumero on 56 vuorokautta eli 8 viikkoa aikaisempi, kuin tuotteiden todellinen tarvepäivämäärä vastaanottajalla. Näin pystytään ennakoimaan rahdin vaatima aika.



	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
Total	0	0	0	29680	47527	57926	26267	27041	12883	2	53247	32240	25941	30760	1600	0	0

Kuva 2. Supplier Portalissa näkyvät tarpeet nimikekohtaisten ennustetaulujen perusteella (nimiketiedot peitetty)

Kun toimittaja lähettää hankintaspecialistille vahvistuksen tuotteiden saapumisesta, poistetaan kyseiset määrät MF50 ennustetaulusta ja lisätään ME38-tauluun "Scheduling Agreement Delivery Schedule" kahdelle eri sivulle. Uudet nimikkeet on lisättävä tauluun nimikkeen luomisen jälkeen. Tämä työ tapahtuu täysin manuaalisesti. Jos toimittaja pystyisi vahvistamaan toimitukset suoraan Supplier Portaliin, ja se kirjoittaisi vahvistukset SAPIin ME38-tauluun ja poistaisi MF50-taulusta automaattisesti, nopeutaisi se merkittävästi hankintaspecialistin työtä.

## 5 PILVIPALVELUT

Pilvipalvelut, englanniksi cloud computing, tarjoavat sovelluksia, dataa, tietojenkäsittelyresursseja ja IT-hallintaa verkkopalveluna Internetissä. Tietojenkäsittelyn maailmassa se on verrattain uusi, radikaali bisnesmalli ja on muuttanut tapaamme toimittaa ja käyttää tietotekniikkaa. Pilvipalveluita voi käyttää missä ja milloin niitä tarvitseekin, helposti skaalautuvasti ja maksaen vain tarvittavista resursseista. (Murugesan & Bojanova, 2016, s. 3)

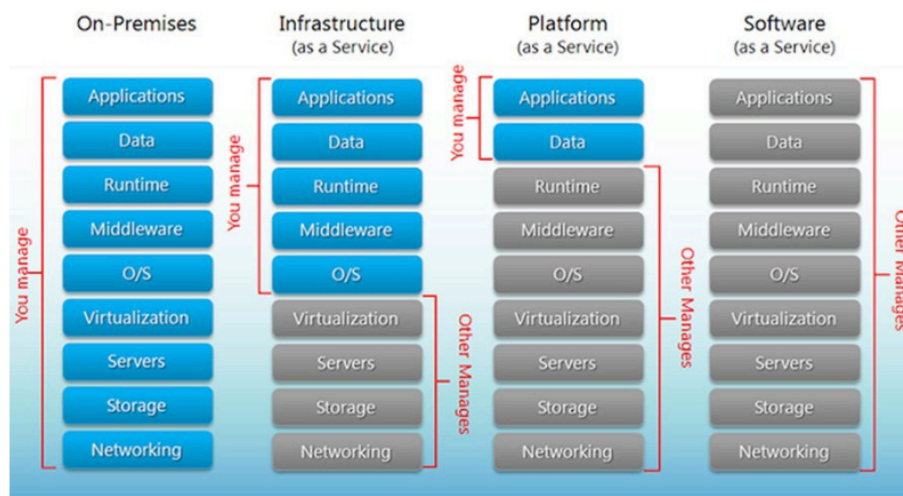
Pilvipalveluilla on omat erityispiirteensä, joilla se eroaa muista tietotekniikan muodoista. Tilauspohjainen itsepalvelu, laajat verkkoyhteydet, resurssien yhdistäminen, nopea joustavuus ja skaalautuvuus niin tallennustilan, nopeuden, kuin käyttäjienkin määrässä, sekä resurssien saattaminen yhtäaikaaisesti monien käyttäjien ulottuville ovat ominaisuuksia, joiden avulla pilvipalvelut pystyvät omaksumaan uusia liiketoimintamalleja. (Murugesan & Bojanova, 2016, s. 6)

Pilvipalveluilla on monia etuja. Verrattuna perinteisiin konesali- ja tietotekniikkaratkaisuihin, pilvipalvelujen hankinta- ja ylläpitokustannukset ovat matalammat, resursseja voi hetkessä kasvattaa tarpeen mukaan ja saattaa tuhansien käytettäväksi ympäri maailman. Tietoturva pilvipalveluissa on usein parempi kuin yksityisten konesalien. Erityisen tärkeää on jaettu pääsy tiedostoihin, dataan ja erilaisiin sovelluksiin ja mahdollisuudet, joita pilvipalvelut tarjoavat ilman investointeja omaan IT-infrastruktuuriin. (Murugesan & Bojanova, 2016, s. 10)

Pilvipalvelut ovat yleistyneet nopeasti, ja erilaisten pilvipalvelujen määrä kasvaa vauhdilla. Palveluntarjoajia on koko ajan enemmän. Suurimmat toimijat kuten Amazon, Microsoft, Google, Salesforce ja IBM saavat rinnalleen pienempiä toimijoita, jotka kaikki tarjoavat erilaisia työkaluja pilvipalveluihin. (Murugesan & Bojanova, 2016, s. 17)

Pilvipalveluita luokitellaan erilaisiin palvelumalleihin teknisten ominaisuuksien mukaan. SPI-pilviluokitus (Software Platform Infrastructure) jakaa palvelumallit kolmeen liiketoimintaprosesseja tukevaan palvelukerrokseen: pilveä voidaan käyttää sovelluspalveluna (SaaS eli Software as a Service), sovellusalustapalveluna (PaaS eli Platform as a Service) tai infrastruktuuripalveluna (IaaS Infrastructure as a Service) (Merelä, 2012, s. 5). Pääluokkien lisäksi palveluita voidaan jakaa muillakin tavoilla; yleinen lyhenne pilvipalvelulle on XaaS eli X-as-a-Service, jossa X voidaan korvata eri palvelumalleilla (Merelä, 2012, s. 8). Kuvassa 3 on kuvattu pilvipalveluiden pääluokkien eri palvelukerroksia, josta näkee selkeästi vastualueiden erot eri palvelukerroksissa.

## Separation of Responsibilities



Kuva 3. Pilvipalveluiden pääluokat (Bort, 2013)

Myös pilvityyppejä on erilaisia, ja ne jaotellaan omistajuussuhteen mukaan. Yksityinen pilvi on oma tai vuokrattu palvelu, jossa resurssit ovat ainoastaan yrityksen omassa käytössä. Yhteisöllisessä pilvessä infrastruktuuria voidaan jakaa useiden osapuolten kanssa yhteisiin intresseihin, mutta resurssit on jaettu ainoastaan määritellyn yhteisön käyttöön. Julkisen pilven palvelut ovat avoimesti saatavilla yleisölle internetissä. Ne voivat olla maksuttomia tai maksullisia palveluja, ja se tarjoaa lähes rajattomat resurssit ja käyttäjät. Hybridipilvi on yhdistelmä kahdesta tai useammasta edellä mainitusta pilvityypistä. (Merelä, 2012, s. 13)

Supplier Portal on PaaS-mallin mukainen sovelluspalvelu ja sen alustaksi valittiin Microsoftin Azure. PaaS-mallista kerrotaan tarkemmin seuraavassa luvussa. Kuten aiemmin on kerrottu, on vastaavia pilvipalveluita tarjolla mm. Amazonilla ja Googlella. Toimeksiantajayrityksen IT-infrastruktuuriin kehitetään integraatioalustaa, jonka avulla yrityksen monien eri rajapintojen yhdistäminen olisi jatkossa joustavampaa, helpompaa ja vähemmän räätälöidyistä integraatioyhteyksistä koostuvaa. Tätä integraatioalustaa on rakennettu Azuressa, joten myös Supplier Portal päätettiin perustaa Azuren pilveen. Näin se on myöhemmin helposti liitettävissä integraatioalustaan, ja se voi tulevaisuudessa esimerkiksi käyttää samoja työnkulun ja integroinnin automaatioon käytettäviä yhdistäjiä, joista lisää kappaleessa 5.5. Supplier Portal on hybridipilviratkaisu, josta kerrotaan tarkemmin kappaleessa 5.2.

## 5.1 PaaS eli Platform as a Service

PaaS- eli pilvialustapalvelut ovat palveluita, joiden avulla asiakkaan on mahdollista luoda omia ohjelmistoja PaaS-palveluntarjoajan virtuaalisesta palvelinympäristöstä asiakkaalle lohkottujen palvelujen avulla. Palveluntarjoaja määrittelee hyväksytyt ohjelmointikielet ja ohjelmistot, mutta itse sovellus-/ohjelmistokehitys jää asiakkaalle. Loppukäyttäjä käyttää palvelua käyttöliittymän kautta, jonne sovelluskehittäjä on kehittänyt palveluntarjoajan työkaluilla ja sovellusrajapinnan avulla erilaisia palveluita. Asiakas käytännössä maksaa PaaS-palvelun saatavuudesta. Asiakas hallinnoi sovelluksiaan ja palveluitaan ja palveluntarjoaja tyypillisesti huolehtii kaikesta muusta, kuten palvelimista, tallennustilasta ja yhteyksistä. PaaS-palvelut tukevat web-sovelluskehityksen koko elinkaarta: siellä voi kehittää, testata, käyttöönottaa, hallinnoida ja päivittää sovelluksia. PaaS-palvelun avulla vältetään kalliit ja monimutkaiset koko infrastruktuurin käyttöönottoon ja ylläpitoon liittyvät kustannukset ja toimenpiteet, eikä erillisiä ohjelmistolisenssejä tarvita. (Mäkinen, 2015, s.10; Microsoft, 2019h)

## 5.2 Hybridipilvi

Hybridipilvi on ratkaisu, jossa yhdistetään kahden tai useamman eri teknologian parhaita puolia. Hybridi-ympäristö koostuu erilaisista komponenteista, ja tarkoitus on ottaa käyttöön vain niitä komponentteja, jotka parhaiten sopivat kehitettävälle ohjelmistolle, toimintatavalle tai organisaatiolle. Hybridipilvessä itse ohjelmisto on asennettu omalle palvelimelle ”On-Premise”, kun ohjelman lisäosat tai muu ohjelman ympärille rakennettu palvelu toimii pilvessä (Visma Software, n.d.).

Tämän opinnäytetyön Supplier Portal perustuu hybridikäyttötapaukseen: ytimessä on On-Premise-palveluna SAP, joka pyörii yrityksen palvelimilla. Kaikki muut toiminnalliset komponentit toimivat pilvessä, ja tietoa siirretään on-premise- ja pilvijärjestelmien välillä. Jokainen käytetyistä komponenteista on itsenäinen kokonaisuutensa, ja niitä voidaan kehittää, versioida ja muokata itsenäisesti. Tämä on erittäin tärkeä ominaisuus Supplier Portalin kannalta, koska se haluttiin rakentaa helposti muunneltavaksi ja toisista komponenteista riippumattomaksi.

## 5.3 Mikropalveluarkkitehtuuri

Mikropalveluarkkitehtuuri, microservices architecture, on jatkuvasti kasvava ohjelmistokehitysmuoto, jossa sovellus on rakennettu erilaisista toisistaan riippumattomista palveluista, jotka kehitetään, testataan, käytetään ja versioidaan itsenäisesti. Tällainen erilaisista mikropalveluista koottava sovellus antaa yrityksille pilvipalveluiden nopeuden ja skaalautuvuuden, olipa heidän valitsema alusta tai työkalu mikä tahansa. (Microsoft, 2019a)

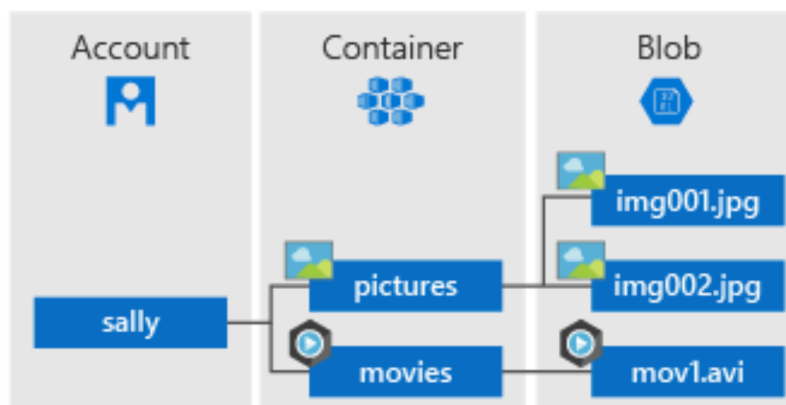
Supplier Portal haluttiin rakentaa palvelinvapaaksi, alustapalveluista koostuvaksi sovellukseksi ilman jäykkää infrastruktuuria, jotta se on niin sanottu ”future proof” eli valmis myös tulevaisuudessa tapahtuville päivityksille ja muutoksille ilman raskasta ohjelmointityötä. Mikropalveluarkkitehtuuri soveltuu Supplier Portalin kehittämiseen siis erinomaisesti. Supplier Portalin arkkitehtuuriratkaisun on suunnitellut IT-partnerin sovellusarkkitehti toimeksiantajan projektitiimin vaatimuksien ja Supplier Portal -web-sovelluksen määrittelyjen perusteella.

#### 5.4 Microsoft Azure Blob Storage

Azure Blob Storage on Microsoftin tarjoama tallennusratkaisu Azure-pilvipalvelussa. Blob Storage on erittäin skaalattava, kestävä ja käytettävissä oleva objektisäilö. Blob Storage on optimoitu tallentamaan valtaviakin määriä jäsentelemätöntä tietoa, eli niin kutsuttuja blob-objekteja (Binary Large Object), joita voi luoda, päivittää, noutaa ja poistaa. (Microsoft, 2019b)

Blob Storageen voi ottaa HTTP/HTTPS-yhteyden mistä ja milloin vain. Blob Storageen tiedot on saatavilla monin eri tavoin: REST-rajapinnan avulla, Azureen sisäänrakennetulla PowerShell-komentorivityökalulla, Azure CLI -komentorivillä tai Azuren Storage client -kirjastojen avulla, joita on taas saatavilla useille eri ohjelmointikielille. (Microsoft, 2019d)

Blob Storageen resurssit jakautuvat kolmeen osaan (kuva 4): Storage-tiliin, tilissä olevaan säilöön (Container) ja blob-objekteihin, jotka tallennetaan säilöön. Storage-tilin nimi toimii säilön nimiavaruutena, ja jokaisen säilön tallennetun blob-objektin osoite sisältää yksilöllisen Storage-tilin nimen. Storageen säilöön (Container) voi järjestellä objektit kansiorakennetta muistuttavaan tapaan. Säilöjen määrää Storage-tilissä ei ole rajoitettu, ja kussakin säilössä voi olla rajoittamaton määrä objekteja. Siksi Azuren Blob Storageen skaalautuvuus on erinomainen. (Microsoft, 2019d)



Kuva 4. Blob Storageen resurssien suhteet toisiinsa. (Microsoft, 2019d)



Blob Storageen voi säilöä kolmen tyyppisiä blob-objekteja: kiinteitä, liitettäviä ja sivu-objekteja. Kiinteitä objekteja, kuten tekstiä tai binääritiedostoja (videot, kuvat), käsitellään yksitellen kokonaisuuksina. Liitettävät objektit ovat kokonaisuuksia kuten kiinteätkin objektit, mutta niihin voidaan lisätä tietoa tai toimintaa; esimerkiksi virtuaalikoneiden lokidataa. Sivu-objektit tallentavat jopa 8 teran kokoisia satunnaisia käyttöoikeustiedostoja, esim. virtuaalisia kovalevyjä, joita Azuresa pyörittävät virtuaalikoneet käyttävät. (Microsoft, 2019d)

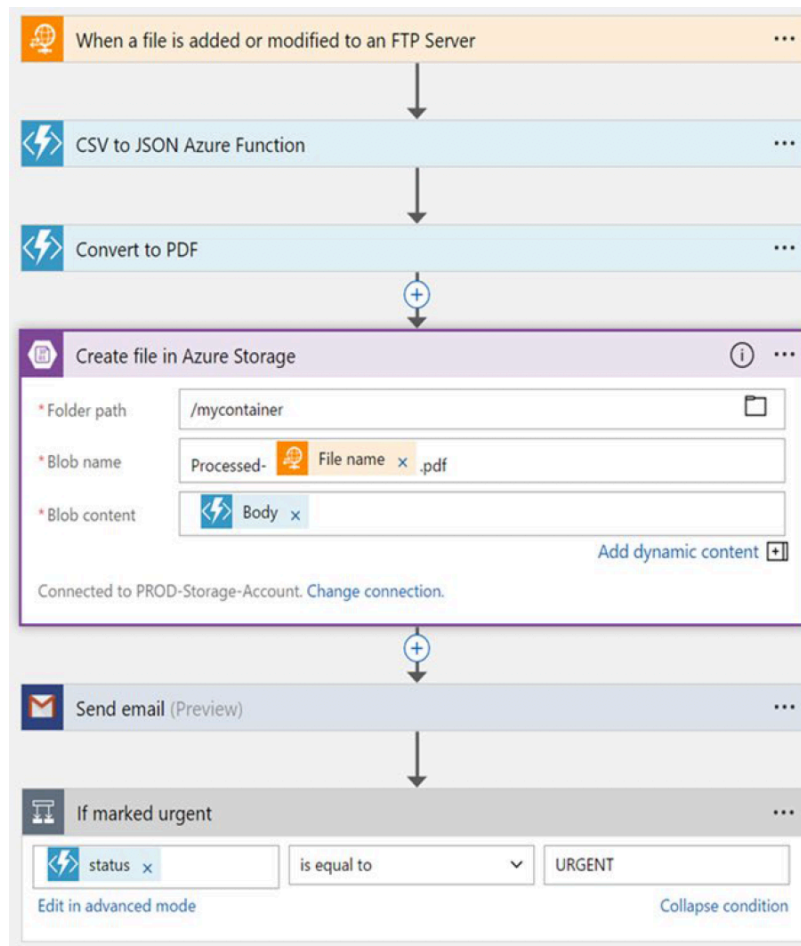
Supplier Portalin kehityksen ensimmäisessä toteutuksessa Blob Storageen säilötään käytännössä vain yksi JavaScriptin React-kirjastolla koodattu tiedosto, joka päivityksen yhteydessä aina korvataan uudella versiolla. Jatkossa Storageen on mahdollista tallentaa esimerkiksi pdf-tiedostoja arkistointia varten tai lokidataa.

## 5.5 Microsoft Azure Logic Apps

Azuren Logic Apps on monipuolinen työkalu, jota voi käyttää esimerkiksi erilaisten Software as a Service (SaaS) -palvelujen, on-premise-palvelujen ja pilvipalveluiden integrointiin ja työnkulun automatisointiin. Logic Apps tarjoaa valmiiksi satoja yhdistäjiä eli connectoreita moniin ohjelmistoihin ja palveluihin, kuten esimerkiksi Salesforce, Office 365, Twitter, Dropbox ja Google. (Microsoft, 2019c)

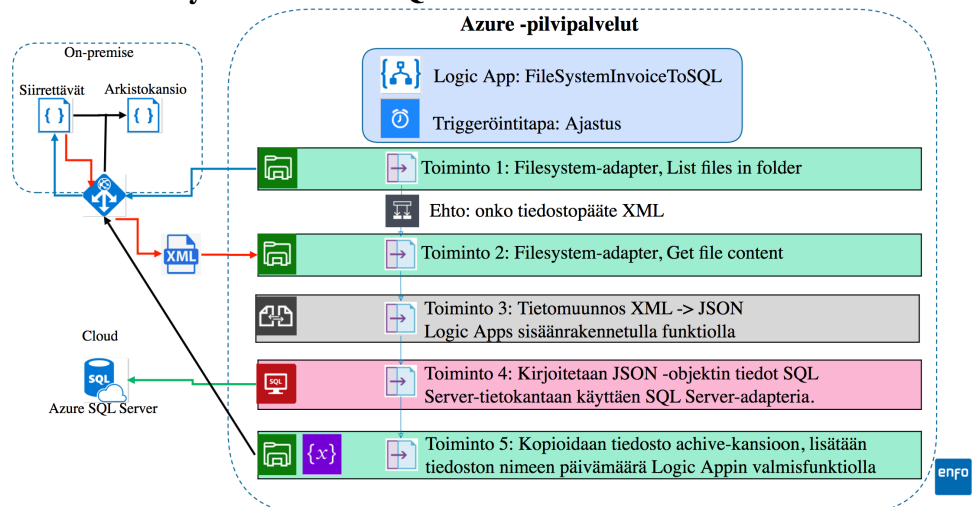
Logic Appsin avulla erilaisia työnkuluja ei tarvitse ohjelmoida erikseen jollakin ohjelmointikielellä, vaan Logic Appsien avulla rakennetaan erilaisista vaiheista, ketjuista ja ehdoista koostuvia työnkuluja, joita toteutuksessa tarvitaan. Näitä työnkulun osia voivat olla esimerkiksi API-kutsut, pyynnöt lukea dataa tietokannasta tai sähköpostin lähetys. Logic Appsin avulla ohjelmistokehittäjät voivat siis rakentaa työnkuluja jopa täysin ilman koodin kirjoittamista. Logic Appsien avulla työnkulkujen rakentaminen voi olla merkittävästi nopeampaa ja helpompaa kuin saman toiminnallisuuden ohjelmoiminen esimerkiksi C#- tai Java-ohjelmointikielillä. (Microsoft, 2018)

Logic App -työnkuluja rakennetaan Azuren palvelussa visuaalisella Logic App Designer -työkalulla erilaisia vaiheita yhdistelemällä. Vaikka työnkulut rakennetaankin ilman ohjelmointia, ne kirjoittautuvat todellisuudessa JavaScript Object Notation -muotoon eli JSON-muotoiseen tekstitiedostoon. Kuvassa 5 on esimerkki Logic App Designerin näkymästä ja erilaisista vaiheista ja triggereistä työnkulussa. (Microsoft, 2018) Kuvassa 6 Supplier Portalin kehittäjän esimerkki hybridikäyttötapauksesta, jossa on-premise-järjestelmästä siirretään ja muunnetaan tietoa Logic Appin avulla.



Kuva 5. Esimerkki Logic App Designer -työkalun avulla tehtävästä työkalusta (Microsoft, 2018)

### Demo 1: FileSystemInvoiceToSQL



Kuva 6. Esimerkki Logic App -työkalusta hybridikäyttötapauksessa. (Enfo, 2019)

Logic Appsin rooli Supplier Portalin toiminnassa on varsin suuri; käytännössä se hoitaa kaiken älykkään toiminnan taustalla, kun loppukäyttäjälle näkyvä käyttöliittymä avataan. Logic App -komponenteilla tarkistetaan autentikointitiedot, pyydetään viestejä SAPista, muunnetaan ne sopivaan muotoon ja tuodaan datat käyttöliittymään. Logic App -työnkuluilla voidaan helposti rakentaa myös lisätoiminnallisuuksia, joita kehitysjonolistaan on kerätty; esimerkiksi tallentaa dokumentteja Blob Storageen, kirjoittaa lokitietoja tai lähettää sähköpostivahvistuksia. Toimeksiantajan tietoturva-protokolla ei salli Logic Appien tarkempaa avaamista tässä raportissa.

## 5.6 Microsoft Azure AD

Azure Active Directory eli Azure AD on Microsoftin pilvipohjainen kirjautumismalusta. Alun perin se on kehitetty Office 365:n kirjautumisia varten, jotta käyttäjät pystyivät käyttämään Microsoftin palveluja, kuten sähköpostia, SharePointia ja Teamsia samalla käyttäjätunnuksella ns. Single Sign-On (SSO) -periaatteella. Sittemmin palvelusta on tehty globaali kirjautumismalusta, jonka toiminnan voi yhdistää todella moniin palveluihin. Valmiita integraatioita erilaisille SSO-ratkaisuille Azure AD:ssa on valmiina tuhansia. (Microsoft, 2019e; Ropponen, 2019)

Tietoturvan kannalta Azure AD lisää suojaa valtavasti. AD kerää статистиikkaa käyttäjien kirjautumisista mm. siitä, millä, milloin ja mistä käyttäjät kirjautuvat palveluihin. AD:n avulla voidaan estää pääsy epäilyttävistä sijainneista, rakentaa erilaisia kirjautumissääntöjä kuten kaksivaiheinen tunnistautuminen ja tarjota käyttäjille turvallinen ja automaattinen tapa suojata käyttäjätietonsa ja -tunnuksensa eri maiden lakien ja asetusten vaatimalla tavalla. (Microsoft, 2019e)

Vanhan Supplier Portalin käyttäjähallinta oli rakennettu sovellukseen omaksi tietokannakseen. Käyttäjähallinnan ylläpito oli hieman epäselvää, sillä selkeitä dokumentteja ei ollut. Järjestelmään kirjaututtiin toimittajatunnuksella, joka tuli SAPista. Sekä hankintaspesialisti, että toimittajan käyttäjä kirjautuivat sovellukseen samalla tunnuksella. Mikäli salasana tai käyttäjät vaihtuivat, tuli uudet kirjautumistiedot päivittää kahdelle käyttäjälle.

Uuden sovelluksen käyttäjähallinta toteutettiin Azure AD:lla. Tällöin käyttäjähallinta on samalla turvallista ja helppoa, eikä erillistä ylläpitoa tarvita. Azure AD:n etuna on myös ilmainen peruskäyttö. Monitahoinen tunnistus vaatii maksullisen version (P1), joka Helvarilla on käytössä; helvarilaisten käyttäjien tunnistukseen on mahdollista käyttää Helvarin SSO-logiikkaa. Azure AD antaa myös turvaa tulevaisuuden kehitykselle; AD:n tuomat mahdollisuudet on vaivaton ottaa käyttöön jatkokehitysvaiheissa.

## 6 API-OHJELMOINTIRAJAPINNAT

Ohjelmointirajapinta (API, application programming interface) määrittelee, miten suljettu ohjelmisto tarjoaa tietoa tai palveluja muiden ohjelmistojen käytettäväksi. Rajapintojen avulla ohjelmoijat ja ohjelmistokehittäjät saavat käyttöönsä dataa ja palveluja ulkoisista lähteistä. (Michel 2010).

Rajapintojen käyttö on kasvanut räjähdysmäisesti. Kaikki tietokoneohjelmat pystyvät siirtämään tietoa rajapintojen läpi. Kuten API-manifesti (API:Suomi community, 2016) sanoo, ”Toimivat rajapinnat ovat edellytys digitaalisten ekosysteemien luomiselle ja skaalaamiselle”.

Rajapintoja on kolmea päätyyppiä: julkinen rajapinta (open API, public API), yhteistyökumppanirajapinta (partner API) ja yksityinen rajapinta (internal API). Julkisessa eli avoimessa rajapinnassa tieto on kaikkien saatavilla ilman rajoituksia. Yhteistyökumppaneille julkaistu rajapinta avataan tietyin rajoituksin ja tietojen tai palvelujen saaminen edellyttää usein jonkinlaista sovittua työkulkua tai tunnistautumista. Yksityiset rajapinnat ovat vähiten tunnettuja, sillä niitä käytetään yleensä vai organisaatioiden sisällä paremman tuottavuuden ja palvelujen uudelleenkäytettävyyden vuoksi. (Castellani, 2017)

### 6.1 Supplier Portal -rajapinta

Supplier Portal tarvitsee tiedot ennustetauluun yksityisellä palvelimella sijaitsevasta on-premise SAP-toiminnanohjausjärjestelmästä. SAP tarjoaa erilaisia mahdollisuuksia tuoda informaatiota rajapintoihin, joista niitä voi käydä kutsumassa monenlaisilla tavoilla. Tässä projektissa yhteys tehdään SAP Web Services -toiminnon avulla, jossa SAPin funktioita voidaan tuoda kutsuttavaksi rajapintaan.

Web Service on World Wide Web Consortiumin (W3C) määritelmän mukaisesti vapaasti suomennettuna web-sovelluspalvelu, jonka avulla sovellukset voivat kommunikoida XML-viestien ja internet-protokollien avulla verkon yli. (Endrei, 2004, s. 32) Web Service -sovelluspalveluissa on aina palveluntarjoaja ja palvelun käyttäjä.

SAPissa Web Service -toiminnot tehdään SOA Manager -työkalun avulla, ja niitä voidaan kutsua SOAP-protokollan avulla. Supplier Portal tekee Microsoft Logic Apps -työkulkujen avulla kyselyjä rajapintaan poimiakseen tarvittavat tiedot Supplier Portalin käyttöliittymään näkyville. Ensimmäisen vaiheen Supplier Portal tarvitsee tietoja kolmesta SAP-funktiosta, joihin on tehty Web Service: ZSW\_SAG\_GETDETAIL, ZSW\_GET\_PART\_NUMBERS ja ZSW\_GET\_PLA\_ORDERS.

SOAP eli Simple Object Access Protocol on sovellusohjelmien välinen viestipohjainen tietoliikenneprotokolla, joka on World Wide Web Consortiumin (W3C) ylläpitämä standardi. Sen avulla voi lähettää ja vastaanottaa sanomia. SOAP käyttää viestimiseen XML-muotoisia sanomia, jotka lähetetään yleisimmin HTTP-protokollan avulla verkon yli. SOAP on käyttöjärjestelmä-, kommunikointikanava- ja ohjelmointikieliriippumaton, joten sen avulla on mahdollista kutsua monin eri tavoin erilaisista lähteistä ja käyttää muitakin kommunikaatiokanavia kuin HTTP-protokollaa. Ainoa edellytys on, että palvelun käyttäjä pystyy vastaanottamaan ja käsittelemään XML-muotoisia viestejä. (Endrei, 2004, s. 117)

XML-muotoinen SOAP-viesti on jakautunut kolmeen osaan. Kirjekuoren eli SOAP Envelopen sisälle on paketoitu viestin tiedot: vapaaehtoinen otsikko SOAP Header sekä itse viestisisältö eli SOAP body. Viestit muunnetaan joko RPC- eli Remote Procedure Call -muotoiseksi tai sanomaperusteiseksi. (Endrei, 2004, s. 118) Kuvassa 7 on esimerkki SAPin lähettämästä viestistä Supplier Portalin tekemään SOAP-kutsuun koskien ZSW\_GET\_PART\_NUMBERS-funktiota; otsikkokenttä on tyhjä, ja body-osiossa näkyy kutsuttava funktio ja sitä vastaava tieto.



Kuva 7. Esimerkki Supplier Webin SOAP-viestistä SOAPUI-ohjelmassa

## 7 TULOKSET

Koska tämä opinnäytetyö on projektityyppinen toteutus, esitellään tässä kappaleessa projektin taustat ja kuvaillaan vanhaa ratkaisua, jonka perusteella uusi ratkaisu toteutettiin. Kappaleessa 7.3. esitellään uuden ratkaisun arkkitehtuuri ja kuvaillaan toimiva demoversio, sillä lopullinen valmis julkaisu ei ehtinyt valmiiksi tämän opinnäytetyön raportin viimeistelyyn mennessä. Projektin aikana ylläpidetty product backlog eli projektin kehitysjonolista esitellään kappaleessa 7.4. Koko projektin vaiheita pääsee lukemaan liitteessä 1 olevasta projektipäiväkirjasta, johon on koottu projektin vaiheet tapaamisten ja yhteydenpitojen muodossa.

### 7.1 Projektin taustat

Supplier Portal -projektin tarve syntyi, kun toimeksiantaja vaihtoi IT-palvelutarjoajansa keväällä 2019. Yritys on ulkoistanut kaikki IT-tarpeensa yhdelle toimijalle kahta yrityksen omilla palkkalistoilla olevaa IT-asiantuntijaa lukuun ottamatta. Yhteistyökumppani huolehtii yrityksen lähes kaikesta IT-infrastruktuurista aina päätelaitteita, palvelimia, tukitoimia ja sovelluskehitystä myöten.

Olemassa oleva Supplier Portal on aiemman IT-toimittajan rakentama sovellus, ja vaihdon jälkeen Supplier Portalista olisi koitunut jatkuvia kustannuksia vanhalle toimittajalle. Jotta lisenssi- ja palvelinkuluista sekä vanhasta IT-toimittajasta päästäisiin kokonaan eroon, tehtiin päätös Supplier Portalin tuottamisesta uuden IT-toimittajan avulla. Heti alussa tehtiin päätös toteuttaa projekti nopealla aikataululla ”Minimum Viable Product” -periaatteella, eli ensimmäisessä vaiheessa tuotetaan toiminnallisesti samanlainen ympäristö, ja sen toimintaa kehitetään vasta seuraavassa vaiheessa. Teknisiä ratkaisuja kuitenkin haluttiin modernisoida niin, että sovelluksen komponentteja, kuten toiminnanohjausjärjestelmää tai datasäiliötä, on mahdollista muuttaa tai päivittää suhteellisen helposti.

Lisäksi kehitystyön aikana kerättiin alustavaa product backlogia, kehitysjonolistaa, eli jatkossa kehitettäviä toiminnallisuuksia tai ominaisuuksia, jotka voidaan toteuttaa ensimmäisen version julkaisun jälkeen. Kehitysjonolistalle lisättiin kohteita koko projektin ajan, ensimmäisestä tapaamisesta lähtien.

### 7.2 Vanha ratkaisu

Aiempi Supplier Portal on websovellus, johon kirjaututtiin toimittajan (eli alihankkijan) käyttäjätunnuksella ja salasanalla. Kirjautumisikkunan jälkeen avautui yhteenvedonäkymä, jossa näkyi toimittajalle määritelty määrä ennustettavia viikkoja, tuotteet ja niiden hankittavat määrät tuleville viikoille. Lukittu viikko näkyi punaisella, eikä sen määriä ei voinut enää

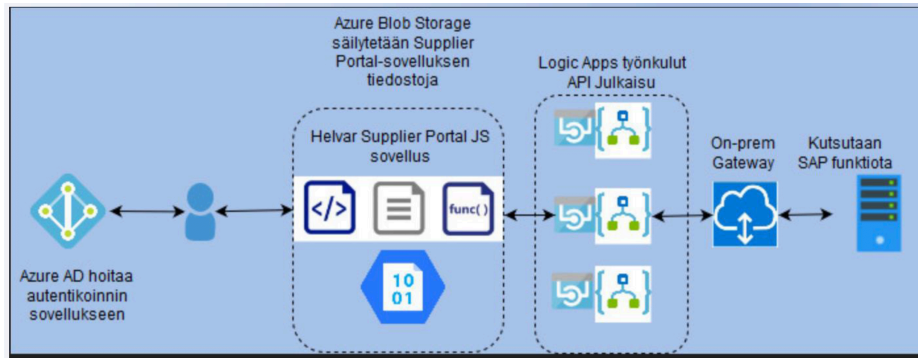
muuttaa. Vahvistetut viikot näkyivät vihreinä, ja loput ennusteet keltaisena. Ennusteet tulivat SAPista rajapinnan avulla reaaliaikaisesti; taulu päivittyi, kun yhteenvetonäkymä avautui, tai kun selain päivitettiin. Yhteenvetonäkymässä oli ennustetaulun lisäksi päivytyspainike, linkki Helvarin kotisivuille sekä painike, josta pystyi tallentamaan ennustenäkymän pdf-dokumenttina.

Käytännössä tilausprosessissa on kaksi aktiivista toimijaa: Helvarin hankintatiimin käyttäjä, joka syöttää SAPIin tarvittavat ennusteet, ja toimittaja, joka kirjautuu Supplier Portaliin ja tarkastaa sieltä Helvarin tarpeet. Toimittaja vahvistaa toimitusmäärät, jonka jälkeen Helvarin hankintaspesialisti siirtää vahvistetut määrät SAPissa MF50-ennustetaulusta ”Scheduling Agreement Delivery Schedule” -tauluun, kahdelle eri sivulle. Siirretyt määrät poistuvat Supplier Portalin ennustenäkymästä, sillä niiden saapuminen on jo vahvistettu.

Vanha Supplier Portal oli ulkoasultaan selkeä, visuaalinen ennuste toimittajalle, mutta prosessi vaati paljon manuaalista työtä. Se toimi vain yhteen suuntaan: SAPista web-sovellukseen. Sen alkuperäinen toimintatapa on ollut kaksisuuntainen, mutta jossain vaiheessa takaisinkirjoitus SAPIin on katkenut. Kaikki muutokset SAPissa täytyi tehdä käsin, koska Supplier Portal ei voinut kirjoittaa tietoja takaisin SAPIin. Supplier Portaliin kirjaututtiin toimittajatunnuksella (vendor), eli sekä helvarilainen käyttäjä, että toimittaja kirjautuivat samalla tunnuksella. Kirjautumistiedoista tallennettiin vain viimeisin sisäänkirjautuminen, josta ei siis voinut päätellä, oliko tietoja tarkastellut helvarilainen vai toimittaja. Supplier Portal ei myöskään tallentanut minkäänlaista dataa tai lokia. Toisin sanoen, jos ennustetta SAPissa muutettiin, se muuttui myös Supplier Portalissa, eikä vanhaa ennustetta ollut enää mahdollista nähdä mistään. Tämän vuoksi jokainen vahvistettu ennuste tallennettiin käsin kopioimalla exceliin Supplier Portalin ennustetaulusta, ja Helvarin hankintaspesialisti lähetti sähköpostilla kuittauksen toimittajalle ennustenäkymän vahvistuksesta. Koska minkäänlaista lokia tai versiodataa ei ollut saatavilla, oli käsin tallentaminen ainoa keino tarvittaessa esittää vahvistetut määrät, jos niitä myöhemmin olisi ali-hankkijan mielestä muutettu.

### 7.3 Uusi Supplier Portal -sovellus

Projektin lopputuotteena syntynyt Supplier Portal -sovellus noudattaa modernia hybridipilviarkkitehtuuria, jonka toiminnan osia on kuvailtu aiemmissa kappaleissa. Kuvassa 8 on yhteenveto Supplier Portalin arkkitehtuurista, josta näkee arkkitehtuurin osa-alueet: autentikoinnin Azure AD:n avulla, Azure Blob Storageen säilötyt tiedostot sisältäen itse sovelluksen, sekä integraatiotoiminnot; dataa kutsutaan Helvarin palvelimella sijaitsevasta SAP-toiminnanohjausjärjestelmästä Logic Apps-työnkulkujen ja rajapintatoimintojen avulla. On-premise Gateway välittää tietoja enkryptattuja eli salattuja kanavia pitkin Azure Service Busissa, Azuren viestinvälitysjärjestelmässä, joka takaa tietoturvallisen ja varman tiedonsiirron.



Kuva 8. Helvar Supplier Portal -sovelluksen arkkitehtuuri

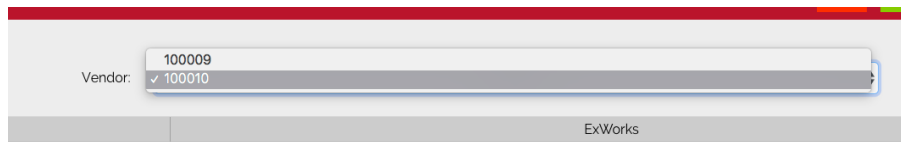
Supplier Portal -sovelluksen lopullinen versio ei ehtinyt valmiiksi ennen tämän opinnäytetyön valmistumista. Sen demoversio ja sovitut seuraavat vaiheet tulevat tuottamaan alkuperäisen Minimum Viable Product -periaatteen mukaisen sovelluksen sovitun aikataulun puitteissa. Kuvassa 9 on kuvakaappaus demoversion ennustenäköymästä, jossa on näkyvillä eri enusteviikkojen värit, tuotenimikkeitä, toimittajan valinta pudotusvalikosta ja päivitys- sekä tulostuspainikkeet.

Helvar Supplier Portal		Refresh Print Excel Locked Confirmed Trade off Zone										
Vendor: 100010		ExWorks										
Item	Name	week44	week45	week46	week47	week48	week49	week50	week51	week52	week01	
11131	EPCOS.B32922-A2224-M3	0	1800	5280	2000	3000	4000	5000	2000	3000	2000	
73600..	IC-MCU-8-BIT-CMOS-SOIC18	3000	0	1000	2000	3000	4000	5000	2000	3000	2000	
73600..	IC-MCU-8-BIT-CMOS-SOIC15	3000	0	1000	2000	3000	4000	5000	2000	3000	2000	
73600..	IC-MCU-8-BIT-CMOS-SOIC16	5000	0	7000	2000	3000	4000	5000	2000	3000	2000	
73600..	IC-MCU-8-BIT-CMOS-SOIC17	3000	0	1000	2000	3000	4000	5000	2000	3000	2000	
73600..	IC-MCU-8-BIT-CMOS-SOIC16	5000	0	7000	2000	3000	4000	5000	2000	3000	2000	
73600..	IC-MCU-8-BIT-CMOS-SOIC17	3000	0	1000	2000	3000	4000	5000	2000	3000	2000	

Kuva 9. Supplier Portal -demoversion käyttöliittymä



Uuden Supplier Portalin ennustenäkömää sisältää samat tiedot, kuin aieminkin, uusien toimintojen lisäksi. Sisään kirjautuneelle Helvarin käyttäjälle on erittäin helppo vaihtaa toimittajanäkymää uudesta ”Vendor”-pudotusvalikosta (kuva 10) aiempaan versioon verrattuna, jossa käyttäjä kirjautui sisään kunkin toimittajan kirjautumistiedoilla, yksi toimittaja kerrallaan.



Kuva 10. Toimittajan vaihtaminen pudotusvalikosta

Visuaalisesti ulkoasu vastaa paremmin toimeksiantajan nykybrändiä ja käyttöliittymä on selkeä. Ennustetaulun värit ovat totutut punainen, vihreä ja keltainen ja värien selitteet näkyvät myös pdf- ja excel-tulostuksissa. Excel-tulostus toimii ennustenäkömästä, kuten kuvasta 11 voi nähdä. Excel-tulostus onkin erinomainen parannus ja nopeuttaa hankintaspesialistin työtä merkittävästi samalla vähentäen inhimillisen virheen aiheuttamia riskejä, mitkä manuaalisella taulukon kopioimisella, liittämällä, muotoilulla ja tallentamisella saattavat olla mahdollisia.

Item	name	week44	week45	week46	week47	week48	week49	week50	week51
11131	EPCOS,B32922-A2224-M3	0	1800	5280	2000	3000	4000	5000	20
11131	EPCOS,B32922-A2224-M3	0	1800	5280	2000	3000	4000	5000	20
11131	EPCOS,B32922-A2224-M3	0	1800	5280	2000	3000	4000	5000	20
7360004	IC-MCU-8-BIT-CMOS-SOIC18	3000	0	1000	2000	3000	4000	5000	20
7360004	IC-MCU-8-BIT-CMOS-SOIC18	3000	0	1000	2000	3000	4000	5000	20
7360004	IC-MCU-8-BIT-CMOS-SOIC18	3000	0	1000	2000	3000	4000	5000	20
7360005	IC-MCU-8-BIT-CMOS-SOIC15	3000	0	1000	2000	3000	4000	5000	20
7360005	IC-MCU-8-BIT-CMOS-SOIC15	3000	0	1000	2000	3000	4000	5000	20
7360005	IC-MCU-8-BIT-CMOS-SOIC15	3000	0	1000	2000	3000	4000	5000	20
7360006	IC-MCU-8-BIT-CMOS-SOIC16	5000	0	7000	2000	3000	4000	5000	20
7360006	IC-MCU-8-BIT-CMOS-SOIC16	5000	0	7000	2000	3000	4000	5000	20
7360006	IC-MCU-8-BIT-CMOS-SOIC16	5000	0	7000	2000	3000	4000	5000	20
7360007	IC-MCU-8-BIT-CMOS-SOIC17	3000	0	1000	2000	3000	4000	5000	20
7360007	IC-MCU-8-BIT-CMOS-SOIC17	3000	0	1000	2000	3000	4000	5000	20

Kuva 11. Demoversion ennustetaulun tulostus exceliin

## 7.4 Supplier Portal Product Backlog

Projektin alussa ja sen aikana kerättiin alustavaa product backlogia. Backlogin user storyt eivät välttämättä vielä täytä INVEST-periaatteita, joten niiden työstäminen ja priorisointi on tehtävä ennen seuraavaa kehitysvaihetta ja toteuttamiskelpoisuuden arviointia. Backlogista näkee selkeästi kehitystarpeet ja sen perusteella voi teettää työmäärä- ja kustannusarvioita.

Backlogissa rooli ”ostaja” tarkoittaa Helvarin hankintatiimin jäsentä ja ”toimittaja” hankittavan tuotteen toimittajaa.

Supplier Portal alustava product backlog:

- Ostaja haluaa vahvistaa ennustetarpeen toimittajalle suoraan Supplier Portalista. Vahvistus tulee lähettää sähköpostilla toimittajan profiilissa olevaan osoitteeseen ja liittää mukaan pdf / excel vahvistuksesta.
- Toimittaja haluaa vahvistaa toimituksen suoraan Supplier Portalista. Toimitusvahvistuksen tulee muuttaa vahvistettujen saapumisten siirrot SAPin MF50-ennustetauluista ME38 Scheduling Agreement Delivery Schedule -saatavuustauluun.
- Toimitusvahvistuksesta tulee olla erillinen, tuotekohtainen näkymä Supplier Portalissa.
- Toimittaja haluaa jättää kommentin toimitusvahvistuksen yhteydessä. Kommentti tulee näkyä vahvistusnäkyssä ja SAPissa.
- Ostaja haluaa jättää kommentin toimittajan vahvistukseen. Kommentti tulee näkyä vahvistusnäkyssä ja SAPissa.
- Kaikki Supplier Portalin kirjautumistapahtumat ja muutokset tulee tallentua lokitiedostoon. Muutoksien kohdalla tulee näkyä, kuka muutoksen on tehnyt.
- Supplier Portalista voi tallentaa ennustenäkyvän pdf:n Supplier Portalin pilveen myöhempää käyttöä varten
- Helvar-käyttäjät voivat tarkastella lokitiedostoja ja tallennettuja tiedostoja pilvessä

## 7.5 Projektinhallinta

Yksi tämän opinnäytetyön lähestymiskulmia oli projektinhallinta, eli toimiminen toimeksiantajan projektipäällikkönä. Projektinhallinnan näkökulmasta projekti eteni loogisesti eteenpäin ja koen onnistuneeni roolissani. Pystyin pitämään kiinni sovitusta aikatauluista, käymään ammattimaisia keskusteluja ja hankkimaan tietoja itsenäisesti projektin eri osapuolilta, sekä tekemään päätöksiä projektin etenemisestä. Vaikeinta minulle oli pysyä Minimum Viable Product -periaatteessa, sillä uusi arkkitehtuuri houkutteli lisäämään uusia, toivottuja ominaisuuksia jo projektin ensimmäisessä vaiheessa. Pystyin kuitenkin toimeksiantajana pitäytymään alkupe- räisessä sopimuksessa ja keräämään nämä halutut toimenpiteet kehitysjo- nolistalle.

Projektin alku oli hieman hidas kesälomien vuoksi; ensimmäinen alkupalaveri oli pidetty ennen kesälomia ja varsinainen projektin kick-off pidettiin syyskuussa. Tänä aikana toimittaja oli kuitenkin suunnitellut arkkitehtuuria

ja tehnyt pieniä testejä, joten kick-off-tapaamisessa pystyttiin käymään läpi jo detaljeja ja päättää ensimmäisen testiversion julkaisupäivä sekä itse sovelluksen lopullinen julkaisupäivä. Toimittajan vahva integraatio-osaaminen ja organisaatiomme tuntemus oli selkeästi projektia vauhdittava voima.

Pohdin, olisiko hankintatiimin jäsenestä projektitiimissä ollut hyötyä. Toisaalta hankintatiimin työkuorma on suuri, ja avoin organisaatiokulttuurimme kannustaa kävelemään suoraan kyseisten henkilöiden luokse kysymään avoimia kysymyksiä aina, kun on tarve. Projekti opetti minulle, että jatkossa aion vielä paremmin selvittää kaikki tarvittavat, käytettävät ja haluttavat toiminnot, jotta mitään toiminnallisuuksia ei jää huomioimatta tai unohdu. Tähän auttaa huolellinen, projektin kokoon sopiva vaatimusmäärittely ja kaikkien osapuolten sitoutuminen siihen. Tässä projektissa sitä ei tehty, koska kehitysprojektin pääpiirteet olivat selkeät ja haluttiin siirtyä reippaassa aikataulussa eteenpäin; kuten seuraavassa Yhteenveto-kappaleessa todetaan, tälle projektille vaatimusmäärittelyksi riitti Minimum Viable Product -periaate.

## 8 YHTEENVETO

Opinnäytetyön tutkimuskysymyksiksi asetettiin:

- *Miten käyttöliittymät toteutetaan?*
- *Miten integraatiotyönkulut toteutetaan?*
- *Miten yhteydet SAP ERP -järjestelmään toteutetaan?*

Opinnäytetyöraportti on vastannut tutkimuskysymyksiin. Tutkimuskysymyksen asettelu oli tärkeää, sillä projektin kokonaisuus on sen verran laaja, että ilman kysymysten asettamaa aiherajausta opinnäytetyön raportti olisi voinut sisältää kaikkea käyttöliittymäsuunnittelusta React-kielillä ohjelmointiin. Lopputuloksena on raportti, joka esittelee paitsi lopputuotteen, myös projektin kulun ja tutkimuskysymysten vastaukset.

Demoversion perusteella Supplier Portal -sovellus tulee julkaisuhetkellä olemaan täysin alkuperäisen Minimum Viable Product -määritelmän ja mikropalveluarkkitehtuurin mukainen, eikä sen aikana tarvinnut tehdä kompromissejä, vaan kaikki tekniset ratkaisut, työnkulut ja rajapintaintegraatiot toimivat kuten ensimmäisessä arkkitehtuurisuunnitelmassa oli niiden ajateltu rakentuvan. Projektin lopputuotteen hyöty tulee näkymään vasta käytössä, mutta jo nyt siinä on kolme merkittävää käyttäjälle näkyvää parannusta aiempaan: selkeämpi käyttäjähallinta, sisäänkirjautuneen Helvar-käyttäjän mahdollisuus tarkastella samassa istunnossa kaikkien toimittajien ennustetauluja sekä excel-tulostus. Nämä kaikki tehostavat työprosesseja hankintaosastolla ja näin nopeuttavat työntekoa ja vapauttavat henkilöresursseja muihin toimintoihin.

Supplier Portal -projekti on ollut todella mielenkiintoinen ja opettavainen. Olen paitsi oppinut paljon hankintaprosesseista yrityksessämme, myös SAPin toiminnasta ja erityisesti integraatioista ja pilvipalveluista, jotka kiinnostavat minua henkilökohtaisesti paljon. Varsinkin Azuren maailmaan oli helppo uppoutua ja pohtia kaikki mahdollisuuksia, joita erilaiset pilviarkkitehtuuriratkaisut tarjoavat. Myös product backlogin ylläpitäminen oli hyödyllistä, ja mielenkiinnolla jään odottamaan, mitä kehityslistan kohteita priorisoidaan listalla korkeimmalle ja miten lista kehittyy ajan kuluessa.

### 8.1 Tulevaisuudennäkymiä

Seuraava kehitysvaihe on priorisoida ja groomata product backlog. Tärkeimpiä kehityslistan kohteita mielestäni on luoda takaisinkirjoitusprosessi, josta jää myös lokimerkintä. Lisäksi halutaan kehittää Supplier Portalin kautta käytävää viestintää mm. vahvistusikkunassa tehtyjen muutosten automaattisista ilmoituksista ja ennustemäärien vahvistuksen kuitauksesta. Näiden muutosten avulla Supplier Portal voidaan ottaa käyttöön laajemmin erilaisten toimittajien kanssa. Uudet toiminnot myös

tehostavat hankintatiimin toimintaa entisestään poistaen manuaalista, käsin tehtävää työtä ja vapauttaen aikaresurseja muuhun käyttöön. Ajankäytöllisesti voidaan puhua jopa usean kymmenen minuutin säästöstä yhden toimittajan kohdalla viikoittain. Vuositasolla säästetty aika voi siis olla todella merkittävä.

Entä voisiko Supplier Portal toimina referenssiprojektina muita vastaavia portaaleja varten? Kuinka helposti saisimme rakennettua esimerkiksi asiakasportaalin, jossa Supplier Portalin tapaan asiakas voisi seurata hänelle vahvistettuja toimituksia tai tehdä niihin muutoksia, jopa tilauksia? Supplier Portalin taakse rakennettu mikropalveluarkkitehtuuri sopii monenlaisien sovellusten kehittämiseen, eikä em. asiakasportaalin rakentaminen olisi tämän kokemuksen perusteella kovinkaan hankala tai vaikea prosessi.

Supplier Portalin teknologia on tehty kestämään tulevaisuuden muutokset. Minkä tahansa ERP-järjestelmän Helvar lähivuosina valitseekin, on sen integroiminen Supplier Portaliin vain kevyt muutosprosessi Logic Appseissa ja lähdekoodissa. Myös product backlogissa olevien kehityskohdeiden toteuttaminen ei nykyarkkitehtuurissa tule vaatimaan suurta tai kallista kehitysprojektia, vaan monet niistä voidaan ottaa hyvinkin kevyellä työllä käyttöön. Supplier Portalin tulevaisuuteen ja kehitykseen eniten vaikuttava asia on, tarjoaako mahdollinen uusi ERP vastaavan raportoinnin jo valmiina moduulina. Olisi helppo ajatella, ettei Helvar olisi toimialallaan ainoa samanlaisia toimittajahaasteita ratkova teollisuusyritys, ja että joku jossain muualla on jo kehittänyt täysin valmiin moduulin moderneihin, pilvipohjaisiin toiminnanohjausjärjestelmiin. Toistaiseksi kuitenkin Supplier Portal tulee valmistuttuaan toimimaan päivittäistyökaluna yrityksen hankintatiimissä.

## LÄHTEET

API:Suomi community (2016). API-manifesti. Haettu 18.10.2019 osoitteesta <http://apimanifesti.fi/>

Bort, J. (2013). 10 Most Important in Cloud Computing. Haettu 10.10.2019 osoitteesta <https://www.businessinsider.com.au/10-most-important-in-cloud-computing-2013-4>

Castellani, S. (2017). What are the different type of APIs? Haettu 18.10.2019 osoitteesta <https://apifriends.com/api-creation/different-types-apis/>

Cohn, M. (2009). User Stories Applied For Agile Software Development (13. painos). Haettu 18.10.2019 osoitteesta <http://athena.ecs.csus.edu/~buckley/CSc191/User-Stories-Applied-Mike-Cohn.pdf>

Endrei, M. (2004). Patterns: Service-Oriented Architecture and Web Services. IBM Redbooks. Haettu 17.9.2019 osoitteesta <https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/reader.action?docID=3306398>

Enfo Oyj (2019). Azure Integration Demo. Esitysmateriaali integraatioesimerkistä.

Kogent Learning Solutions (2010). SAP MM Black Book. New Delhi: Dreamtech Press.

Mäkinen, J. (2015). PaaS-palveluiden teknologia ja liiketoimintamallit. Jyväskylän Yliopisto. Haettu 10.10.2019 osoitteesta <https://pdfs.semanticscholar.org/4fac/0faa6f80e4ca178ea15f282cc631d4ec7e4a.pdf>

Merelä, M. (2012). Pilvipalveluarkkitehtuuri. Helsingin yliopisto, Helsinki. Haettu 10.10.2019 osoitteesta <https://docplayer.fi/1057617-Pilvipalveluarkkitehtuuri.html>

Microsoft (2018). Azure Integration Services. Haettu 15.10.2019 osoitteesta <https://azure.microsoft.com/en-us/resources/azure-integration-services/>

Microsoft (2019a). .NET Microservices: Architecture for Containerized .NET Applications. Haettu 3.9.2019 osoitteesta <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/index>

Microsoft (2019b). Azure Blob -säilö. Haettu 13.8.2019 osoitteesta [https://flow.microsoft.com/fi-fi/connectors/shared\\_azureblob/azure-blob-säilö/](https://flow.microsoft.com/fi-fi/connectors/shared_azureblob/azure-blob-säilö/)

Microsoft (2019c). Azure Logic Apps Documentation. Haettu 18.8.2019 osoitteesta <https://docs.microsoft.com/en-us/azure/logic-apps/>

Microsoft (2019d). Introduction to Azure Blob storage. Haettu 13.8.2019 osoitteesta <https://docs.microsoft.com/fi-fi/azure/storage/blobs/storage-blobs-introduction>

Microsoft (2019e). What is Azure Active Directory? Haettu 18.8.2019 osoitteesta <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-what-is>

Microsoft (2019f). What is Paas? Haettu 10.10.2019 osoitteesta from <https://azure.microsoft.com/en-us/overview/what-is-paas/>

Murugesan, S., & Bojanova, I. (2016). Encyclopedia of Cloud Computing. Chichester, Iso-Britannia: John Wiley & Sons, Inc. Haettu 13.9.2019 osoitteesta <https://ebookcentral-proquest-com.ezproxy.hamk.fi/lib/hamk-ebooks/detail.action?docID=4526670>

Olsen, D. (2015). Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback. Haettu 13.10.2019 osoitteesta <http://ebookcentral.proquest.com/lib/hamk-ebooks/detail.action?docID=1895170>

Pichler, R. (2010). Make Your Product Backlog DEEP. Haettu 21.10.2019 osoitteesta <https://www.romanpichler.com/blog/make-the-product-backlog-deep/>

Ropponen, S. (2019). Azure AD on globaali kirjautumislusta - ja paljon muuta. Haettu 12.9.2019 osoitteesta <https://www.tietokeskus.fi/azure-ad-on-globaali-kirjautumislusta-ja-paljon-muuta/>

Salminen, L. (2018). Sovelluksen suunnittelumenetelmät -kurssin verkkoinen, Moodle. Hämeen ammattikorkeakoulu. Haettu 20.10.2019 osoitteesta <https://moodle.hamk.fi>

SAP (2019a). ERP ja digitaalinen ydin. Haettu 21.8.2019 osoitteesta <https://www.sap.com/finland/products/erp.html>

SAP (2019b). SAP Support Strategy. Haettu 21.8.2019 osoitteesta <https://support.sap.com/en/offerings-programs/strategy.html>

Visma Software. (n.d.). Pilvi vai on-premise ERP? Opas oikean ratkaisun löytämiseen. Haettu 21.8.2019 osoitteesta <http://suunta.visma.fi/pilvi-vai-on-premise-erp>

## Supplier Portal -projektipäiväkirja

### 16.8.2019 Supplier Portal pikakatsaus, Teams-puhelu

Toteuttajan projektipäällikkö esitteli Supplier Portalin uuden suunnitellun arkkitehtuurin, teknisen toteutuksen pääpiirteet, projektin lähtökohdat sekä ehdotukset projektin etenemiseksi. Sovittiin kutsua koolle kesälomien jälkeen "kick-off" -tapaaminen, johon mukaan tulevat myös toteuttajan kehittäjät ja arkkitehti.

### 22.8.2019 Vanhan Supplier Portalin ja SAPin toimintojen esittely

Toimeksiantajan hankintaspesialistin kanssa käytiin läpi vanhaa Supplier Portalia (josta hankintatiimi käyttää nimeä Call-Off) ja tutkittiin, mitä vaiheita koko prosessiin kuuluu alkaen ennusteen luomisesta ja päättyen saapumispäivän vahvistamiseen. Hankintaspesialisti kertoi ensin alihankinnan käytännöistä yleensä ja SAPin transaktioista, joita ennusteiden ja vahvistusten käsittely vaatii. Sen jälkeen tutustuttiin olemassa olevaan Supplier Portaliin ja katsottiin, miten muutokset SAPissa näkyvät Supplier Portalissa, miten alihankkija eli toimittaja sitä käyttää, ja mitä kaikkea tapahtuu, ennen kuin ennuste on vahvistettu ja hankittavien tuotteiden saapumispäivä varmistunut.

### 12.9.2019 Projektitiimin kick-off

Projektitiimi kokoontui Karkkilaan Supplier Portal -projektin viralliseen aloituspalaveriin. Aluksi esiteltiin projektitiimin jäsenet; toteuttajalta läsnä olivat projektijohtaja, integraatioarkkitehti, järjestelmäintegraattori sekä poissaolevana ohjelmistokehittäjä.

Ensin käytiin läpi tämänhetkinen tilanne: Kuvattiin projektin lähtötilanne ja ensimmäisen vaiheen tavoite: luoda Supplier Portalista modernimpi versio edellisestä ja saada se toimimaan uudessa ympäristössä mahdollisimman pian.

Uuden Supplier Portalin pääpiirteet kerrattiin: Ei lisenssi- tai palvelinmaksuja, hyvin pienet kuukausikulut Azuren palveluista. Ei palvelimia tai kiinteää infraa, vaan Supplier Portal luodaan Azuren alustapalveluihin pääpainon ollessa Azuren Logic Appseissa ja Azure Blob Storagessa. Uusi, moderni ja kehittyvä Supplier Portal tulee vähentämään manuaalista työtä ja sen käyttöastetta on mahdollista kasvattaa tämän hetkisestä muutamasta aktiivisesta alihankkijasta useisiin kymmeniin.

Käyttäjälle näkyvä käyttöliittymä on Reactilla tehtävä web-sovellus, joka näkyy Blob Storageen säilötystä koodista loppukäyttäjälle hänen kirjautuessaan sisälle Supplier Portaliin; kutsurajapinta hakee Logic Appilla SAPista



reaaliaikaisesti palautejoukon, joka täyttää käyttöliittymän ja tekee toiminnot. Koodin päivittäminen on helppoa; käytännössä uusi koodi tallennetaan Storageen vanhan tilalle, ja päivitetään selain. Kaikki valittavat komponentit tukevat Azure AD -tunnistautumista.

Sovittiin ensimmäisen vaiheen tavoitteet: Luodaan nykyisen sovelluksen toiminnot modernisti uuteen Supplier Portaliin: yhteenvetotaulu sekä print as pdf arkistointia ja varmistussähköpostia varten. Päätettiin ottaa yksi uusi toiminnallisuus mukaan: Export to excel. Tällöin saadaan yksi manuaalinen toiminto pois, kun alihankkijan tai hankintaspesialistin ei tarvitse kopioida Supplier Portalin ennustetaulun näkymää käsin Exceliin. Muut uudet toiminnallisuudet, ideat ja jatkokehitystoiveet kirjataan ylös product backlogiin. Keskusteltiin myös sovelluksen ulkoasun uudistamisesta vastaamaan paremmin toimeksiantajan nykyistä brändi-identiteettiä.

Keskusteltiin käyttäjähallinnasta ja tutkittiin senhetkistä käyttäjätietolomaketta. Lomakkeessa oli useampi kenttä, mutta niistä vain muutama oli käytössä: Vendor user number (SAPin toimittajanumero) ja salasana, Description-kenttään kirjoitettu alihankkijayrityksen nimi, sekä ennustetaulussa näkyvien viikkojen määrä sekä transport time -kentässä näkyvien päivien määrä. Pohdittiin käyttämättä jääneiden kenttien tarkoitusta ja tarpeellisuutta uudessa versiossa. Pohdittiin, hoidetaanko tunnistautuminen jatkossa nykyiseen malliin toimittajanumerolla vai otetaanko käyttöön henkilöön liittyvä tunnistautuminen. Azure AD:n työkalut on valittu käyttäjätunnusten hallintaan ja keskusteltiin sen mahdollisuuksista.

Sovittiin, että toimeksiantajan projektipäällikkö selvittää pääkäyttäjien ja testaajien käyttäjähallintaa, sekä selvittää ennustetaulun näkymiä varten hankintatiimiltä seuraavat asiat:

- Näytettävien viikkojen oletusmäärä
- Näytettävien viikkojen maksimimäärä
- Tarkista tuleeko toimitusaikaviikkojen määrä SAPista
- Nykyisten käyttäjien listaus ja keitä näistä viedään uuteen
- Hankintatiimin pääkäyttäjä
- Hankintatiimin testaaja

Aikataulusta sovittiin, että ensimmäinen testausversio on näytillä viimeistään 7.10., jolloin projektitiimi kokoontuu seuraavan kerran. Jos näytettävää on aikaisemmin, se julkaistaan testattavaksi heti kun mahdollista. Projektin valmistuspäivämääräksi sovittiin 4.11.2019.

#### 7.10.2019 Ensimmäinen demoversio

Toimittaja julkaisi ensimmäisen toimivan demoversion. Demoversio käytti demodataa, eikä siinä ollut käyttäjätunnistusta. Versiosta pystyi kuitenkin tutkia käyttöliittymää ja taustalla sovelluksen arkkitehtuuri toimi suunnitellusti. Käyttöliittymän ulkoasua korjataan seuraavaan versioon.

#### 10.10.2019 Sähköposti / SAP-kutsut

Toimittaja esitti kysymyksiä SAP-kutsujen käytännön toteutuksesta. Tiedossa on, mitä SAP-funktioita Supplier Portalin pitää pystyä SAPista kutsumaan, mutta kuinka nämä funktiot saatetaan rajapintaan. Toimittaja pyysi selvittämään vanhan Supplier Portalin toimittajalta, onko funktioista tehty SAP Web Servicejä, joita voisi kutsua SOAPilla uudesta Supplier Portalista. Vanha sovellus käytti kutsumiseen erillistä SAP connectoria, jonka tekeminen uuteen sovellukseen olisi mahdollista, mutta se ei olisi uuden arkkitehtuurimäärittelyn mukainen, vaan se vaatisi kustomoiden ratkaisun rakentamisen SAPin ja Supplier Portalin välille.

#### 16.10.2019 Viikkopalaveri

Käynnistettiin viikoittain tapahtuvat tilannekatsauspuhelut, joissa käydään läpi edelliset sovitut asiat ja seuraavat vaiheet. Tutkittiin demoversion käyttöliittymää ja sovittiin korjaustoimenpiteistä. Sovittiin myös seuraavat toimenpiteet: aitojen toimittajatietojen vieminen Azure AD -palveluun ja pdf-tulostuspainikkeen lisääminen.

#### 21.10.2019 Web Service -tilannepäivitys

Selvityspyynnön tuloksena tuli ilmi, ettei Web Servicejä ollut luotu aiemmin, vaan vanha Supplier Portal käytti Remote-Enabled Modulea. Sovittiin, että laitetaan vanhalle toimittajalle pyyntö työmäärä- ja kustannusarviosta Web Servicen luomisesta, joka sisältäisi ensimmäisessä vaiheessa tarvittavat funktiot.

#### 23.10.2019 Viikkopalaveri

Käytiin läpi hankintaosaston antamat palautteet demoversion käyttöliittymästä. Samalla tutkittiin uutta julkaisua demoversiosta, johon oli lisätty tulostuspainike myös pdf-tulostusta varten, lisätty ennusteviikkoja sekä korjattu aiemmassa versiossa nähdyt käyttöliittymän visuaaliset ongelmat. Seuraavaa vaihetta ei päästä juurikaan jatkamaan, ennen kuin yhteys SA-Piin saadaan.

### 6.11. Viikkopalaveri

Vanha IT-toimittaja on onnistunut luomaan Soamanagerin SAP-funktioille. Web Service luvattiin valmiiksi 11.11., jonka jälkeen voidaan yhdistää Logic App hakemaan todellinen data Supplier Portaliin viimeisiä testejä varten. Demoversioon oli lisätty enemmän viikkoja ja dataa, jotta skaalautuvuutta voi testata. Excel-tulostus onnistuu, mutta selaimesta riippuen yli 10 viikon ennustenäkyvät eivät vielä tulostu exceliin miellyttävästi.

### 11.11. Web Servicet valmiina

Vanha IT-toimittaja vahvisti, että SAPin Web Servicet ovat valmiit testattavaksi. Pienien tarkistusten jälkeen yhteys saatiin toimimaan, ja SAP-funktioita voi kutsua SOAPUI-ohjelmalla Helvarin palvelimelta.

### 13.11. Viikkopalaveri

Viikkopalaverissa käytiin läpi edellisen viikon toimenpiteet ja päivitettiin tiedot SAP-yhteyden osalta. Sovittiin, että 20.11. demoversiossa toimii AD-autentikointi sekä ennustetaulun datat tulevat SAPista. PDF-tulostuksessa tarkennetaan vielä tulostuksen asettumista, kun näkymässä on useita viikkoja.