Opinnäytetyö (AMK)

Tieto- ja viestintätekniikan koulutus

2019

Rasmus Niinimäki

JÄRJESTELMÄINTEGRAATION TOTEUTUKSEN MIGRAATIO SEKÄ KEHITYS MULELLA

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ AMK | TIIVISTELMÄ
TURUN AMMATTIKORKEAKOULU
Tieto- ja viestintätekniikan koulutus
Syksy 2019 | 17 sivua

Rasmus Niinimäki

JÄRJESTELMÄINTEGRAATION TOTEUTUKSEN MIGRAATIO SEKÄ KEHITYS MULELLA

Tämän opinnäytetyön tavoitteena oli opiskella järjestelmäintegraatiota sekä Mule ja IIB teknologioita sekä järjestelmäintegraation kehityksen vaiheita. Järjestelmäintegraatio mahdollistaa organisaation sovelluksien tiedonkuljetuksen yhden toteutuksen alle. Datapisteiden määrän kasvaessa sekä eri tyyppisien sovelluksien käyttötarpeen ilmentyessä järjestelmäintegraatio nousee esille validina vaihtoehtona toteuttaa näiden välinen viestiliikenne.

Työssä selvitettiin käytetyt teknologiat sekä käytetty arkkitehtuurimalli. Näiden pohjalta käytiin läpi integraatiototeutuksen suunnittelu, kehitys sekä lopputulos. Työssä perehdyttiin myös toteutettuun integraatiosovellukseen ja sen migraatioprosessiin. Lopuksi esiteltiin järjestelmäintegraatio toteutuksen kehityksessä esiin tulleet keskeisimmät huomiot. Opinnäytetyön ohella toteutettiin aiheen mukainen asiakasprojekti.

Asiasanat:

järjestelmäintegraatio, enterprice service bus, Mule, IBM Integration Bus

BACHELOR´S THESIS| ABSTRACT
TURKU UNIVERSITY OF APPLIED SCIENCES
Degree programme in Information and Communication Technology
Fall 2019 | 17 pages

Rasmus Niinimäki

ENTERPRICE APPLICATION INTEGRATION IMPLEMENTATION'S MIGRATION AND DEVELOPMENT USING MULE

This thesis is a study in enterprise application integration with Mule and IIB technologies. The goal was to develop an integration implementation for the customer. Enterprise application integration makes transforming the organizations data through one system possible. As the amount of different datapoints and datatypes grows, implementing an enterprise application integration becomes a potential solution to make the needed data communications possible.

The thesis explains the technologies and software architecture model that were used for the project. The planning, execution, and outcome of the project are represented based on this knowledge. In this thesis we also look at the original integration which was migrated to the new environment. The most crucial observations that came up during the project are represented at the end. Customer project was executed besides thesis.

Key words:

enterprice application integraion, enterprise service bus, Mule, IBM Integration Bus

Sisältö

Käytetyt lyhenteet sekä sanasto

ESB	Enterprise Service Bus -arkkitehtuurimalli järjestelmäintegraatio toteutuksien kehittämiseen. Suom. palveluväylä
IIB	IBM Integration bus on väliohjelmisto toteuttaen ESB:tä
JDK	Java Development Kit on ohjelmistopaketti Java-tuotteisiin
SOA	Service Oriented Architecture. Tietojärjestelmien toiminnan suunnittelutapa. Suom. Palvelusuuntautunut arkkitehtuuri.
Virta	Graafinen mallinnus integraatiototeutuksesta

1. Johdanto

Työssä käsitellään asiakkaan olemassa olevan järjestelmäintegraatio palvelun siirtämistä Mule-ympäristöön. Esitellään käytetyt teknologiat sekä toimintatavat. Varsinainen työ asiakkaalle on erillisenä liitteenä luottamuksellisista syistä.

Työn tarkoituksena on kartoittaa sekä dokumentoida, kuinka integraatiojärjestelmä siirretään ympäristöstä toiseen. Työn tarkoituksena on myös antaa tarvittavat perustiedot teknologioista sekä yksityiskohtaisempia havainnointeja käyttäen materiaalia kehitysprojektista. Tavoitteena on muodostaa ymmärrettävä sekä kattava kokonaisuus aiheesta tukeutuen teoriaan sekä esimerkkeihin.

Työtä kirjoittaessa tutustuttiin laajasti järjestelmäintegraatiota sekä käytettyjä teknologioita käsitteleviin materiaaleihin. Vaikka yleisesti aiheesta löytyy kattava määrä materiaalia, ongelmaksi koitui tuoreimpien versioiden kirjallisen materiaalin saatavuus. Kirjoja eri teknologoista kyllä löytyi, muuta ne olivat useita vuosia vanhoja, ja tässä ajassa moni asia on ehtinyt muuttumaan. Tuoreemmista versiopäivityksistä löytynyt materiaali oli joko ohjelmistojulkaisijan oma dokumentaatio tai kehittäjien aiheesta kirjoittamat artikkelit. Virallisiin dokumentaatioihin pystyi kirjoittaessa suhtautumaan huomattavasti luotettavimmin kuin kehittäjien kirjoittamiin artikkeleihin. Näihin artikkeleihin piti suhtautua aina varauksella.

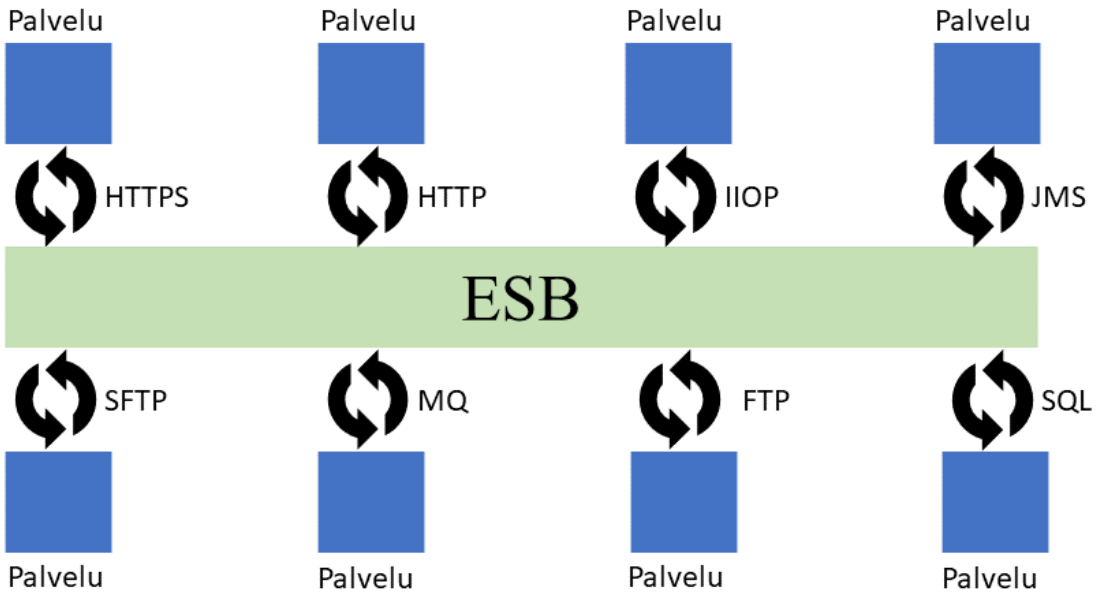
Työssä on järjestelmäintegraatiokattotermin alla keskitytty Enterprice Service Bus

-malliin. Järjestelmäintegraatiota on mahdollista toteuttaa monella muullakin tavalla. Vaikka tämä arkkitehtuurimalli ei ole tuorein vakiintunut tapa tehdä integraatiototeutuksia, ESB on silti vakiinnuttanut paikkansa toimivana alustana.

2. Teoriataustan kuvaus

2.1 Enterprice Service Bus -arkkitehtuurimalli

Enterprice Service Bus (ESB) on arkkitehtuurimalli, joka syntyi tarpeen kasvaessa siirtää suurempia määriä dataa nopeammin sekä edullisemmin. ESB mahdollistaa eri palveluiden ja sovelluksien välisen viestinnän (KUVA 1). ESB toimii alustana, johon eri lähteistä tulevat viestit saapuvat, ja toimittaa ne eteenpäin halutussa muodossa. Yhdistettävien teknologioiden sekä ESB:n välillä tieto liikkuu väyliä pitkin, joita kutsutaan busseiksi. ESB ei rajaa busseja lokaaliin tai internetympäristöön. Sen sijaan ESB pyrkii yhdistämään eri alustoilta tulleen tietoliikenteen yhden järjestelmän alle. Täten kehittäessä integraatiototeutuksia käyttäen ESB-alustaa mahdolliset muutokset sekä lisäykset ovat vaivatonta lisätä mukaan prosessiin. (Laliwala & Samad & Desai & Vyas 2013, 8.)



KUVA 1. ESB -arkkitehtuurimallin havainnollistaminen.

2.3 ESB osana palvelusuuntautunutta arkkitehtuuria

Palvelusuuntautunut arkkitehtuuri eli Service-oriented architecture (SOA) on tietojärjestelmien toteutukseen suunniteltu arkkitehtuurimalli. SOA-mallilla ohjelmistot ovat suunniteltu toimimaan joustavina sekä itsenäisesti, missä toiminnallisuudet ovat määritelty palveluiksi. (IBM Cloud Education, 2019)

ESB on oleellinen osa SOA-arkkitehtuurimallia. ESB mallintaa SOA-arkkitehtuurin palveluiden kommunikointiväylää. SOA-palvelut käyttävät siirtämiseen standardoituja tietoliikenneprotokollia kuten SOAP:ia HTTP:llä. ESB hoitaa viestin manipuloinnin sekä reitityksen halutulle palvelulle. ESB:n vahvuudet tulevat esille erityisesti työskennellessä vanhempien teknologioiden kanssa. SOA:n toteutus on mahdollista ilman ESB:tä, mutta eri teknologioiden kommunikointi sekä palveluväylien ylläpito on todettu koituvan ongelmaksi ilman tätä. (IBM Cloud Education, 2019)

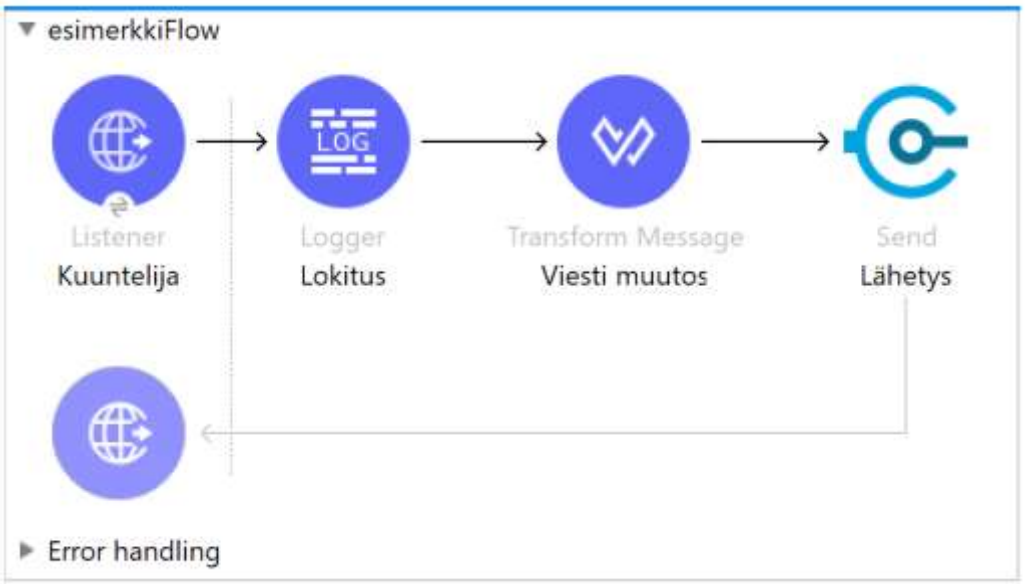
2.2 Mule -integraatiotyökalu

Mule on MuleSoftin kehittämä kevyt tapahtumapohjainen ESB sekä integraatiotyökalu. Mule tukeutuu Oraclen Javaan ja kehittäessä käytetty JDK on määriteltäessä kehitysalustassa. (Dossot & D'Emic & Romero, 6–8.)

Mulen mahdollistaa yksittäisen integraatiosovelluksen tasolta skaalaus koko yrityksen kokoisien integraatiototeutuksien alustaksi, missä pystytään esittämään yhtenäisesti eri tyyppisiä viestejä sekä liitettäviä laajennuksia. Laajennukset tuovat tuen integraatioalustan ja eri lähdejärjestelmien välille. Nämä laajennukset sisältävät eri toiminnallisuuksia kuten viestien kulkuyhteydet, integraatiototeutuksien tietoturvan sekä hallinnan. Kehittäjät pystyvät myös tuottamaan itse Muleen mukautettuja päivityspaketteja sekä parannuksia. (Dossot & D'Emic & Romero, 6–8.)

Mulen suoritus- sekä kehitysympäristönä toimii Eclipseen perustuva Anypoint Studio. Anypoint Studio mahdollistaa integraatiosovelluksen suorittamisen paikallisesti sekä visuaaliset muokkauseditorit integraationsovelluksen eri osa-alueille. Mule-sovellukset

koostuvat virroista, jotka on koottu pienemmistä komponenteista (KUVA 2). Nämä komponentit toteuttavat aina yhden toiminnon, ja niitä on mahdollista linkittää toisiin komponentteihin, mikä tekee integraatiototeutuksen tulkitsemisesta selkeää. Komponenttien avulla on mahdollista vastaanottaa dataa, muokata sitä sekä lähettää eteenpäin. Niiden avulla onnistuu myös lokitus, virheenhallinta, toisen virran kutsuminen sekä virran laukaiseminen. Synkroniset virrat pakottavat kutsujan odottamaan, kunnes se on suoriutunut loppuun. Asynkroninen virta toimii suorittaen toiminnot samanaikaisesti. (Mulesoft.com, 2019) (Bafna, 2017)



KUVA 2. Esimerkki virta Mule-ympäristössä

Mule runtime engine on integraatiosovelluksen suorittamisen ydin. Se pyörittää Mule-sovelluksia ja sitä käyttäen on mahdollista yhdistää eri laitteita, palveluita sekä järjestelmiä Mule-sovellukseen. Mule 4 on Mulen ajoympäristön tuorein julkaisu.

Datasense-toiminnolla suoritetaan tiedon muokkaus määriteltyä mallia vasten. Datasensen avulla Anypoint Studio muodostaa valmiin näkymän, jonka perusteella kehittäjä pystyy päättelemään, mitä dataa on käytössä. Datasense voi käyttää DataWeave-kieltä datan manipulointiin sekä muokkaamiseen.(Mulesoft.com, 2019)

Integraatiosovelluksella on aina jokin laukaisin. Se voi olla ajastin tai jokin tapahtumapohjainen komponentti. Tämän pohjalta Mule muodostaa Mule event

objektin, minkä se välittää seuraavalle komponentille. Yleisimpiä laukaisimia Mulessa ovat HTTP sekä JMS-kuuntelijat, ajastimet, rivioperaattorit tietokannassa sekä FTP-kansion kuuntelijat. (Tutorialspoint)

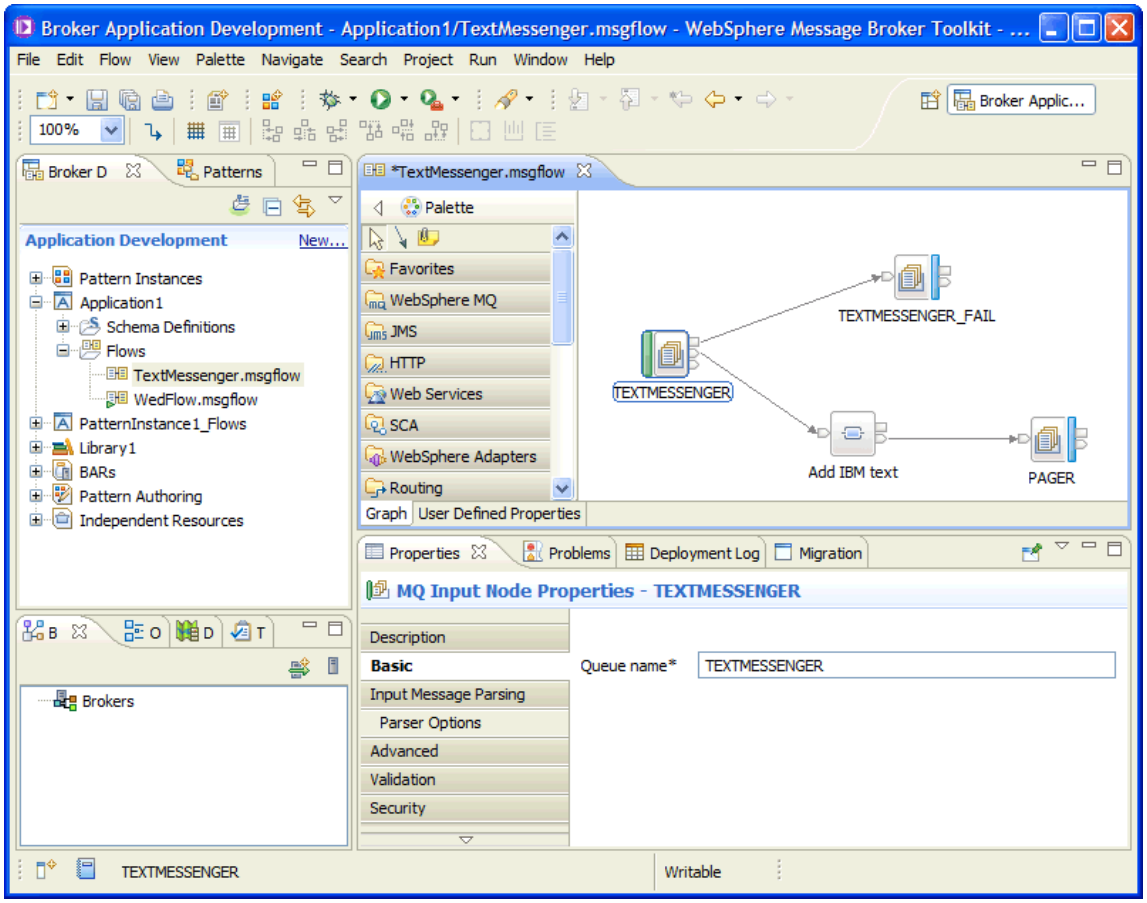
Munit on Anypoint Studioon integroitu teknologia, jonka avulla kehittäjä voi suorittaa automatisoituja testauksia Mule-virroille sekä rajapinnoille. Eri komponenttien toimivuuden testaamisen lisäksi MUnit mahdollistaa komponenttien toiminnan jäljittelyn. Tämä on hyödyllinen ominaisuus, kun testatessa ei pystytä tuottamaan jotain osaa integraatiototeutuksesta esim. laukaisua. MUnit mahdollistaa myös viestien kutsumisen varmentamisen sekä sisällön varmentamisen. (Bafna, 2017)

2.3 IBM Integration Bus

Integration Integration Bus (IIB) tunnettiin aiemmin nimellä WebSphere Message Broker. IIB on ESB, jolla on ESB:lle tyypilliset ominaisuudet sekä toiminnot. Se on yksi IBM keskeisistä integraatiotyökaluista hyödyntäen Microsoftin .NET sekä Oraclen Java-teknologioita. IIB tukeutuu IBM Websphere MQ-protokollaan, mutta kykenee hyödyntämään myös Java Message Serviceä, Web Servicejä, yhteyttä ERP-palveluihin, TCP/IP:tä sekä HTTP/S:ää. (Heritage ym. 2014, 37–38.)

IIB tarjoaa malleja, joita hyödyntäen kehittäjä käyttää suositeltuja tekniikoita sekä toimintamalleja kehittäessä alustalla integraatiototeutuksia. IIB:n tarjoamat resurssit ovat konfiguraatiota vaille valmiita. IIB:lle on rakennettu mahdollisuudet toimintaan monien muiden IBM WebSphere -tuotteiden kanssa. Lisäominaisuuksina IIB tarjoaa virheidenhallinnan, lokituksen, aikakatkaisun ilmoitukset sekä työmäärän hallinnoinnin. (Heritage ym. 2014, 38–41.)

IIB:n kehitys- sekä testausalustana toimii IBM Integration Toolkit (KUVA 3). Toolkit pohjautuu Eclipse alustaan ja sillä on erilaisia näkymiä, jotka ovat kokoelmia erilaisista integraatiosovelluksen graafisista editoreista sekä ikkunoista. Merkittävimmät ovat “Integration development perspective” integraationsovelluksen kehitykseen sekä “Debug perspective” ohjelmistovirheiden hallintaan. (IBM Corporation, 2019)



KUVA 3. Esimerkki virta IBM Integration Toolkit -ympäristössä. (IBM Corporation, 2019)

IIB:ssä kulkevat virrat suorittavat komponentteja monisäikeisesti. Komponenttien välinen yhteys määrittää, mitkä komponentit suoritetaan, missäkin tilanteessa. Yleensä virroilla on esimerkiksi virheenhallinnalle oma polku ja onnistuneelle suoritukselle oma. Virran on pakko sisältää syöttö komponentti, joka tuottaa virran viestille sisällön. (IBM Corporation, 2019)

Websphere MQ -viestijonot ovat keskeinen osa IIB-kehitystä. Niiden kautta ohjataan monesti saapuva sekä lähtevä data, virheenhallinta sekä arkistointi. Näitä viestijonoja sovelletaan myös muihin tarpeisiin IIB-virroissa.

3.0 Suunnittelu ja toteutus

Kuten missä tahansa kehitysprojektissa, on tärkeää ensin suunnitella, mitä lähdetään tekemään. Integraatiototeutuksen määrittelee yleensä asiakas tai kehittäjä parhaaksi näkemänsä tavan mukaan. Integraatiotyöskentelyssä on tapana myös määritellä, joko kirjallisena tai muulla tavalla integraatiototeutuksen tiedot, mikä helpottaa seuraavan kehittäjän työskentelyä sen kanssa.

3.1 Suunnittelu

Suunnittelussa on tärkeää perehtyä lähde- sekä kohdejärjestelmiin. Kehittäjän on hyvä ymmärtää myös, mitä tietoliikennettä kulkee integraatiototeutusta pitkin. Kun kehittäjällä on nämä järjestelmät tiedossa, hän voi paremmin hahmottaa kokonaisuuden sekä ymmärtää asiakkaan ympäristön että liiketoiminnan tarkoituksen. Selkeä kommunikointi asiakkaan ja kehittäjän välillä on tärkeää, koska tällöin voidaan paremmin välttyä integraatiototeutusta koskevilta väärinymmärryksiltä. Jos kehittäjä kaipaa selkeyttä tiedon muutoksiin sekä reitityksiin, hän voi muodostaa siitä taulukon tai kaavion, jotka kuvastavat näitä toimia. Kehittäjä voi tukeutua tähän aineistoon, kun hän muokkaa tietoa lähdejärjestelmästä kohdejärjestelmän muotoon.

Todennäköisesti kehittäjän organisaatiolla on myös vakiinnutettu toimintamalli integraatiokehityksen pohjaksi. Näillä varmistetaan, että kehittäjän tuottama jälki on helpommin tulkittavaa, koska todennäköisesti joku toinen kehittäjä tulee tarkastelemaan samaa työtä joko integraatiototeutusta päivittäessä tai seuraavaan ympäristöön siirryttäessä.

Jos kyseessä on vanhan integraatiosovelluksen migraatio, on tärkeää tutustua ensin vanhaan integraatiototeutukseen. Perehtymällä vanhaan integraatiototeutukseen kehittäjä näkee, miten se on toteutettu ja pystyy hyödyntämään sitä omassa työssään. On kuitenkin tärkeää, että ei käännä vanhaa integraatiototeutusta suoraan uudeksi, koska

integraatioteknologiat ovat yleensä kehittyneet ja todennäköisesti integraatiototeutukset on mahdollista toteuttaa tehokkaammaksi sekä yksinkertaisemmaksi.

3.2 Toteutus

Toteutuksessa kehittäjän tulisi keskittyä integraatiosovelluksen toiminnan varmuuteen. Saatujen tietojen varassa integraatiosovelluksen pitäisi toimia täydellisesti ottaen huomioon kaikki poikkeustapaukset, mitä esimerkiksi tietoliikenteessä voi tapahtua. Validoinnit sekä viestien muokkauksien pitää siis taipua kaikkiin sovittuihin järjestelmiin sekä teknologioihin.

Vaikka oletuksena on, että integraatiototeutus toimii kaikissa sallituissa olosuhteissa, on välttämätöntä sisällyttää virheidenhallinta.

Testaus on tärkeä osa kehitystä. Integraatiosovellusta testatessa on tärkeää ottaa kaikki erikoistapaukset sekä kuormamäärät huomioon. Tietyt ongelmatilanteet ilmenevät vasta, kun integraatiosovelluksen kuorma ylittää tietyn pisteen. On kuitenkin hyvä huomioida, millainen määrä dataa on oikeasti mahdollista tulla lähdejärjestelmästä, ettei keskitytä ongelmiin, jotka eivät ilmene käytetyllä kuorma määrällä.

3.3 Mule-toteutus

Mulella toteuttaessa integraatioyhteyttä on tärkeää ensin selvittää, millä ajoympäristöllä integraatiosovellusta tullaan suorittamaan. Tämän jälkeen päätetään, millaisella toteutusmallilla virtoja aletaan kehittämään. Tämä määrätään monesti organisaation tai kehittäjätiimin puolesta, mutta jos näin ei ole on olemassa Mulesoftin suosittelema ”reliability pattern” -malli, joka perustuu siihen, että viestiä ei menetetä, missään tapauksessa.

Domain-projektin hyödyntäminen on myös suositeltavaa uudelleen käytettävyyden lisäämiseksi sekä salattujen arvojen hallinnan helpottamiseksi. Samaa domain-projektia

pystyy käyttämään usean eri projektin kanssa tuoden valmiit konfiguraatiot domain-projektista. (Deepak Haldar, 2019)

DataSense Explorer -ominaisuuden avulla kehittäjä pystyy hahmottamaan, mitä dataa komponentille on menossa sisään ja mitä sieltä on tulossa ulos suorituksen jälkeen. Tämä mahdollistaa tehokkaat tiedonmuutokset DataWeave-kielen avulla. Kirjoittaessa komponenttien tietokenttiin Anypoint Studio osaa myös ehdottaa datan polkurakennetta sekä datatyyppejä.

Yhteenveto

Työssä käsiteltiin järjestelmäintegraatiototeutuksen siirtäminen ympäristöstä toiseen. Aluksi perehdyttiin alkuperäiseen integraatiototeutukseen ja sen pohjalta suunniteltiin kuinka sama integraatiosovellus tullaan toteuttamaan uudessa ympäristössä. Suunnittelun jälkeen siirryttiin kehitykseen. Työn esimerkkitoteutuksessa sovellettiin kaikkia näitä osa-alueita ja kehitettiin toimivia integraatioyhteyksiä yhden suoritusympäristön alle.

Päällimmäisenä työstä ilmeni, että mitä parempi suunnitelma sitä vaivattomampi toteutus. Selvällä suunnitelmalla pystytään vähentämään toteutusvaiheessa epäselvyyttä sekä kehittäjälle tulee varmuus soveltaa parhaaksi näkemänsä tavan mukaan. Teknologioiden aikaisempi kokemus nopeuttaa prosessia. Jos kyseessä on integraatiototeutuksen migraatio, edellisen integraationtoteutuksen dokumentaation laatu vaikuttaa huomattavasti uuden kehityksen nopeuteen sekä sujuvuuteen.

Lähteet

Chappell, Dave. 2004. Enterprise Service Bus. USA: Carolina, Sebastopol: O'Reilly. S.352. O'Reilly Series. ISBN:0596006756

David Dossot, John D'Emic, and Victor Romero. 2014. Mule in Action, Second Edition. USA, NY, Shelter Island: Manning Publications Co. S.404. ISBN: 1617290823

Deepak Haldar. 2019. Mule Domain Project & Deploying Domain Project w/Mule Standalone. Luettu 1.12.2019. https://apisero.com/tech-tutorials/f/mule-domain-project-deploying-domain-project-wmule-standalone

Dr. Zakir Laliwala, Abdul Samad, Azaz Desai, Uchit Vyas. 2013. Mule ESB Cookbook. UK, Birmingham. Packt Publishing Ltd. S.411. ISBN: 9781782164401

IBM Cloud Education. SOA (Service-Oriented Architecture). Luettu 1.12.2019. https://www.ibm.com/cloud/learn/soa

IBM Corporation. IBM Integration Bus Version 10.0 Documentation. Luettu 5.11.2019. https://www.ibm.com/support/knowledgecenter/en/SSMKHH_10.0.0/com.ibm.etools.msgbroker.helphome.doc/help_home_msgbroker.htm

Ian Heritage, Claus T. Jensen, Tamjit Jensen, Maria Luisa Lopez de Silanes Ruiz, Sambasivarao Nanduri, Juan Carlos Pineda, Abhinav Priyadarshi, Katherine Sanders, David Shute, Jaime Martin Talavera, Mark Taylor, John M. Zoltek, Jr. 2014. Integration Throughout and Beyond the Enterprise. USA: IBM Redbooks publication. S.88 ISBN: 0738439630

Jitendra Bafna. MUnit Testing With Mulesoft: Part I. 2017. Luettu 12.11.2019 https://dzone.com/articles/munit-testing-with-mulesoft-part-i

Jitendra Bafna. Understanding Anypoint Flows and Sub-Flows With MuleSoft. 2017. Luettu 10.11.2019. https://dzone.com/articles/understanding-flows-sub-flows-with-anypoint-muleso

Mulesoft, MuleSoft Documentation. Mulesoft.com. 2019. Luettu 14.11.2019 docs.mulesoft.com

Tutorialspoint. MuleSoft – Endpoints. Luettu 14.11.2019 https://www.tutorialspoint.com/mulesoft/mulesoft_endpoints

V-Soft Consulting. Mulesoft's Mule 4 Vs Mule 3 Release, Why Upgrade? Luettu 13.11.2019. https://blog.vsoftconsulting.com/blog/mule-4-vs-mule-3-release-which-is-more-beneficial