



TAMPEREEN TEKNILLINEN YLIOPISTO

**MIKKO ROIHUVUO**  
**DOKUMENTINHALLINTAJÄRJESTELMÄN JA**  
**TUOTANNOHJAUSJÄRJESTELMÄN**  
**INTEGRAATIO**

Diplomityö

Tarkastaja: professori Kari Systä  
Tarkastaja ja aihe hyväksytty Tieto-  
ja sähkötekniikan  
tiedekuntaneuvoston kokouksessa  
11. tammikuuta 2013

# TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Tietotekniikan koulutusohjelma

**ROIHUVUO, MIKKO:** Dokumentinhallintajärjestelmän ja tuotannonohjausjärjestelmän integraatio

Diplomityö, 45 sivua, 3 liitesivua

Toukokuu 2013

Pääaine: Ohjelmistotuotanto

Tarkastaja: professori Kari Systä

Avainsanat: M-Files, ohjelmistokehitys, SAP R/3, tiedonhallintajärjestelmät, tietojärjestelmäintegraatio

Tämä työ on kuvaus ohjelmistoprojektista, jossa kehitettiin yrityksen ostoprosesseja helpottava ja automatisoiva järjestelmä. Järjestelmä toteutettiin yhdistämällä M-Filesin ja SAP:in ominaisuuksia. Ohjelmistokehitys tehtiin M-Files Oy:n työntekijänä todelliselle asiakkaalle syksyllä 2012. Projektin tavoitteena oli tuottaa nopeasti asiakkaan toivoma järjestelmä ostotilausten ja ostolaskujen käsittelyyn. Toissijaisena tavoitteena oli kartuttaa SAP:iin ja sen tietojärjestelmäintegraatioihin liittyvää osaamista. Työssä esitellään järjestelmän osat, vaatimukset, toteutus, toteutuksen erityiset ongelmakohdat sekä arviointi.

M-Files on suomalainen dokumentinhallintajärjestelmä, joka auttaa yrityksiä säilyttämään ja organisoimaan dokumenttejaan ja pitämään ne helposti kaikkien työntekijöiden saatavilla. SAP puolestaan on tuotannonohjausjärjestelmä, joka kattaa lähes kaikki yritysten prosessit yhdessä tietojärjestelmässä.

Järjestelmien integraatio toteutettiin siten, että M-Files valittiin pääasialliseksi tallennuspaikaksi skannatuille laskuille ja ostotilaustulosteille. M-Filesin työnkulkujen avulla laajennettiin SAP:in tarjoamaa ostotilaustoiminnallisuutta lisäämällä siihen tarkastus- ja hyväksymisvaiheet. Lisäksi työnkulkujen avulla toteutettiin laskun luominen SAP:iin. Sen sijaan ostotilausten luominen ja tilattujen tavaroiden vastaanotetuksi merkitseminen tapahtuvat edelleen SAP:in käyttöliittymän kautta.

Käyttäjien työskentelyn helpottamiseksi järjestelmä muistuttaa ostoprosessin eri vaiheissa asianosaisia henkilöitä sähköpostitse lähetettävillä ilmoituksilla. Sähköpostin sisältämän linkin avulla käyttäjät pääsevät nopeasti tarkastelemaan huomiota vaativaa kohdetta järjestelmässä.

Projektissa onnistuttiin toteuttamaan asiakkaan toivoma järjestelmä. Toteutetun järjestelmän arkkitehtuurillisia päätöksiä on arvioitu DCAR-menetelmää yhdelle arvioijalle soveltaen. Arvioinnin perusteella järjestelmään liittyvät arkkitehtuurilliset päätökset olivat pääsääntöisesti hyviä.

## ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Information Technology

**ROIHUVUO, MIKKO:** Integration of a document management system and an enterprise resource planning system

Master of Science Thesis, 45 pages, 3 Appendix pages

May 2013

Major: Software engineering

Examiner: Professor Kari Systä

Keywords: M-Files, software development, SAP R/3, information management systems, software system integration

This thesis is a description of a software project that combined capabilities of both M-Files and SAP to facilitate and automate the company's procurement processes. The project was implemented for a real customer of M-Files Oy in the autumn of 2012. Objective of the project was to quickly produce the system desired by the customer for handling purchase orders and purchase invoices. The secondary objective was to develop skills related to SAP and SAP integrations. This thesis introduces the parts of the system, requirements, implementation, specific problem areas of the implementation, and the evaluation.

M-Files is a Finnish document management system that helps companies to keep documents safe, organised and easily available to all employees. SAP is an enterprise resource planning system, which implements almost all important business processes in one system.

In the implementation M-Files was chosen as the main repository for scanned purchase invoices and printable purchase orders. M-Files workflows extended SAP purchase order functionality with inspection and approval phases. Creating purchase invoices in SAP was also implemented with scripts in M-Files workflows. The original SAP UI was still used in creating purchase orders and keeping track of received goods.

To save the users' time the system reminds them with email notifications in different stages of the process. By following the link included in the email users can quickly review the items requiring attention.

The project managed to produce the system desired by the customer. Architectural decisions of the implemented system were evaluated with DCAR method adjusted to a single evaluator. The architectural decisions were evaluated mainly as good ones.

## **ALKUSANAT**

Ensimmäisenä haluan kiittää vaimoani Johanna Roihuvuota, jota ilman tästäkään ei olisi tullut mitään. Toisena haluan kiittää työni ohjannutta Kari Systää, jonka kannustava asenne oli tärkeää työn edistymiselle. Kiitos myös työnantajalleni M-Filesille diplomityön ja päivätyön mahdollistamisesta, erityisesti inhimilliselle esimiehelleni Veikko Juusolalle.

Tampereella, 6. huhtikuuta 2013

Mikko Roihuvuo

# SISÄLLYS

1	Johdanto.....	1
2	M-Files.....	3
2.1	Tehtävä ja tarkoitus.....	3
2.2	Suunnitteluperiaatteet ja arkkitehtuuri.....	4
2.3	Metatietorakenne.....	4
2.4	Työnkulut ja skriptaaminen.....	5
2.5	Ulkoiset tiedostolähteet ja OCR-moduuli.....	5
2.6	Ulkoiset kohdetyypit.....	6
2.7	M-Files SAP Connector.....	6
3	SAP ERP.....	8
3.1	Tehtävä ja tarkoitus.....	8
3.2	ArchiveLink.....	9
3.3	Transaktiot.....	9
3.3.1	ArchiveLinkin määrittämiseen liittyvät transaktiot.....	9
3.3.2	Ostotilausprosessiin liittyvät transaktiot.....	10
3.4	Rajapintafunktiot.....	11
3.4.1	Yleinen toimintaperiaate.....	11
3.4.2	RFC_READ_TABLE.....	12
3.4.3	BAPI_PO_RELEASE.....	12
4	Valmiit integraatiomodulit.....	13
4.1	KGS:n integraatiomodulit.....	13
4.1.1	Yleistä.....	13
4.1.2	KGS Content Server.....	13
4.1.3	KGS Index Download.....	14
4.1.4	KGS Activator.....	14
4.2	ExternalObjectTypeRefresher.....	15
5	Integraation toteutus.....	16
5.1	Vaatimukset järjestelmälle.....	16
5.2	Suunnitteluratkaisut.....	16
5.3	Kehitysympäristö.....	17
5.4	Järjestelmän kuvaus.....	17
5.4.1	Järjestelmäarkkitehtuuri.....	17
5.4.2	Ostoprosessi.....	20
5.5	Toteutetut työnkulut M-Filesissä.....	25
5.5.1	Työnkulkuihin perustuvasta ratkaisusta.....	25
5.5.2	Ostotilausten hyväksyntä.....	26
5.5.3	Ostotilaustulosteiden lähettäminen.....	27
5.5.4	Ostolaskujen hyväksyntä.....	29

5.6	SAP Wrapper.....	32
5.6.1	Tehtävä ja tarkoitus.....	32
5.6.2	Hyväksyjän löytäminen.....	32
5.6.3	Ostotilauksen vapauttaminen.....	33
5.7	Toteutuksessa ilmenneet ennakoimattomat ongelmat.....	33
5.7.1	Uusien kohteiden linkittyminen SAP-dokumentteihin.....	33
5.7.2	SAP taulujen moniosaiset pääavaimet.....	34
5.7.3	Ostotilaustulosteiden siirtyminen M-Filesiin ennen hyväksyntää.....	34
5.7.4	Ostotilauksen tietojen syöttäminen laskulle.....	34
5.7.5	Ostotilausten uudelleenhyväksyminen.....	35
5.7.6	Varastoon tilaaminen.....	36
5.7.7	Laskun viitenumeron syöttäminen laskulle.....	36
5.7.8	Laskun tiedoissa olevien virheiden paljastaminen ennen lähettämistä.....	36
5.7.9	Hylättyjen laskujen poistaminen.....	37
5.8	Toteuttamatta jätetyt ominaisuudet.....	37
5.8.1	Valuuttakurssin esittäminen M-Filesissa.....	37
6	Toteutetun ratkaisun arviointi.....	38
6.1	Projektin onnistumisen arviointi.....	38
6.2	Arkkitehtuurin arviointi DCAR-menetelmää soveltaen.....	38
6.2.1	Arviointimenetelmän kuvaus.....	38
6.2.2	Arviointimenetelmän soveltaminen.....	40
6.2.3	Arvioinnin tulokset.....	41
6.3	Jatkokehitysajatuksia.....	42
7	Yhteenveto.....	45
	Lähteet.....	46

## TERMIT JA NIIDEN MÄÄRITELMÄT

<b>COM</b>	Component Object Model. Microsoftin kehittämä standardi prosessien väliseen kommunikaatioon.
<b>DCAR</b>	Decision-centric architecture review. Päätöskeskeinen arkkitehtuurien arviointimenetelmä.
<b>KGS</b>	Saksalainen SAP:in dokumentti-integraatioihin erikoistunut ohjelmistoalan yritys.
<b>M-Files Oy</b>	Suomalainen dokumentinhallintaan erikoistunut ohjelmistoalan yritys.
<b>M-Files</b>	M-Files Oy:n kehittämä dokumentinhallintajärjestelmä.
<b>OCR</b>	Optical Character Recognition. Optinen tekstin tunnistaminen kuvasta.
<b>RFC</b>	Remote Function Call. Tekniikka, joka mahdollistaa SAP:in funktioiden kutsumisen toisista järjestelmistä.
<b>SAP AG</b>	Saksalainen tuotannonohjausjärjestelmiin erikoistunut ohjelmistoyritys.
<b>SAP</b>	SAP ERP -tuotannonohjausjärjestelmä.

# 1 JOHDANTO

M-Files on uusi ja nopeasti leviävä suomalainen dokumentinhallintajärjestelmä, joka tarjoaa helpon ja yksinkertaisen ratkaisun dokumenttien ja niihin liittyvän tiedon hallintaan. Yksinkertaisen dokumentinhallintaratkaisun lisäksi M-Files sisältää monipuolisia työkaluja ja ominaisuuksia, joiden avulla on mahdollista toteuttaa myös monimutkaisempia tiedonhallintaratkaisuja yritysten tarpeisiin.

SAP on jo 1970-luvulla kehitetty tunnettu tuotannonohjausjärjestelmä, jonka tavoitteena on kuvata kaikki keskeiset yrityksen prosessit yhdessä tietojärjestelmässä. Tässä tavoitteessa on onnistuttu hyvin. Vaikka SAP:in käyttöönottoprojektit ovat tunnetusti kalliita, SAP on silti levinnyt niin laajalle, että se on muodostunut eräänlaiseksi tuotannonohjausjärjestelmien standardiksi. Markkinoilla ei edelleenkään liene tarjolla toista järjestelmää, joka kuvaisi yritysten prosesseja yhtä monipuolisesti.

Laajasta toiminnallisuudestaan huolimatta SAP ei kuitenkaan vastaa kaikkiin yritysten tarpeisiin. Uuden toiminnallisuuden ostaminen SAP:iin voi olla kallista ja aikaavievää. Joskus päädytään ratkaisuihin, joissa tarvittavaa toiminnallisuutta toteutetaan SAP:in ulkopuolella erillisissä ohjelmistoissa.

Tämä diplomityö on raportti ohjelmistoprojektista, jossa toteutettiin yrityksen ostoprosessi osittain SAP:in ja osittain M-Filesin tarjoaman toiminnallisuuden avulla. Projekti on toteutettu M-Files Oy:n työntekijänä todelliselle asiakkaalle, jonka organisaatiossa oltiin samanaikaisesti toteuttamassa SAP:in käyttöönottoa ja laajentamassa M-Filesin käyttöä.

Projektin tavoitteena oli laajentaa SAP:in ostoprosessia M-Filesin avulla ja tehdä prosessia yksinkertaisemmaksi käyttäjän näkökulmasta. Ostoprosessin haluttiin sisältävän mahdollisimman vähän käsin tehtävää tietojen syöttämistä ja skannattujen tiedostojen siirtelyä. Järjestelmän haluttiin lisäksi lähettävän prosessin eri vaiheista aktiivisesti sähköpostimuistutuksia käyttäjille, ettei heidän tarvitsisi jatkuvasti käydä tarkistamassa järjestelmän tilaa.

M-Filesin avulla pystyttiin toteuttamaan skannattujen laskujen helppo tuonti järjestelmään. Myös ostotilaukset saatiin automaattisesti yhdistettyä skannattuihin laskuihin optisesti tunnistetun ostotilausnumeron perusteella. Lisäksi ostotilausten hyväksyntä sekä ostolaskujen tarkastaminen ja hyväksyntä pystyttiin automatisoimaan M-Filesin työkulkujen avulla.

Projektissa hyödynnettiin valmiita KGS:ltä hankittuja kolmannen osapuolen komponentteja. KGS on saksalainen SAP:in dokumentti-integraatioihin erikoistunut yritys, joka pyrkii ohjelmistotuotteillaan helpottamaan SAP:in liittämistä muihin järjestelmiin. KGS:ltä hankittujen komponenttien lisäksi projektin onnistumista edesauttoi myös se SAP-asiantuntemus, jota ostettujen ohjelmistojen ohjelmistotuen kautta oli mahdollista hyödyntää.



Noin vuotta ennen projektia M-Files Oy:ssä oli toteutettu M-Filesin ja SAP:in välillä tietoa siirtävä demonstraatiojärjestelmä suurta potentiaalista asiakasta varten. Demonstraatiojärjestelmää varten M-Filesiin oli hankittu ja liitetty KGS:n komponentit. Tämän järjestelmän kehittämistä ei ollut kuitenkaan jatkettu eikä sitä ollut koskaan otettu käyttöön yhdelläkään asiakkaalla. Lisäksi M-Files Oy:ssä oli toteutettu myös SAP Connector -komponentti tavarantoimittajatietojen tuomiseksi SAP:ista M-Filesiin. Nämä järjestelmät toimivat pohjana projektin ohjelmistokehitystyölle.

Tässä työssä kuvataan projektin tuloksena syntynyttä järjestelmää sekä arvioidaan syntynyttä järjestelmää ja sen arkkitehtuuria. Projektin vaiheet ja prosessit on rajattu työn ulkopuolelle. Niitä on kuitenkin kuvattu taustaksi niiltä osin, kun sen on koettu auttavan järjestelmän ymmärtämistä. Toteutetun järjestelmän arkkitehtuurin arviointiin on käytetty DCAR-menetelmän periaatteita soveltuvilta osin.

Työn luvussa kaksi esitellään toteutetun järjestelmän kannalta keskeisiä M-Filesin toimintaperiaatteita ja ominaisuuksia. Kolmannessa luvussa käydään läpi keskeisiä osia SAP-järjestelmästä. Projektissa käytetyt valmiit ohjelmistokomponentit esitellään neljännessä luvussa. Viidennessä luvussa kuvataan projektin toteutus ja lopputuote. Toteutetun järjestelmän arviointi ja jatkokehitysajatukset käydään läpi kuudennessa luvussa. Lopuksi esitetään yhteenveto työstä seitsemännessä luvussa.

## 2 M-FILES

### 2.1 Tehtävä ja tarkoitus

Jotta voitaisiin ymmärtää kahden järjestelmän integraatiota ja siihen liittyviä haasteita, on ensin tärkeää ymmärtää, millaisia integroitavat järjestelmät ovat, ja miten ne poikkeavat toisistaan. Koska projekti toteutettiin M-Files Oy:n työntekijänä, M-Filesiin liittyvää tietoa ja asiantuntemusta oli helposti saatavilla. SAP sen sijaan oli ennestään täysin tuntematon järjestelmä, ja siihen liittyvä tieto ja asiantuntemus oli yrityksessä vähäistä.

Yksittäisille tietokoneen käyttäjille omien sähköisten dokumenttien löytäminen ei ole yleensä erityisen vaikeaa. Dokumenttien määrän kasvaessa kansiorakennetta on mahdollista päivittää joustavasti luomalla uusia kansioita ja luokittelemalla tiedostoja eri kansioihin. Kun samoja tiedostoja sekä samaa kansiorakennetta käyttää useampi ihminen, tällainen dokumentinhallintaratkaisu muuttuu nopeasti riittämättömäksi. Kansiorakenteen monimutkaistuessa ongelmaksi tulee helposti se, että käyttäjät eivät tiedä, mihin kansioon dokumentti pitäisi tallentaa.

M-Files on suomalainen dokumentinhallintajärjestelmä, joka pyrkii tarjoamaan yksinkertaisen ratkaisun edelläkuvattuun ongelmaan. Lisäksi se tarjoaa myös jatkuvasti kasvavan määrän erilaisia ratkaisuja erilaisten yritysten prosessien tehostamiseksi. M-Files integroituu Windows-käyttöjärjestelmiin, jolloin se näkyy käyttäjälle ja sovelluksille virtuaaliasemana. Kirjautuminen M-Filesiin voidaan tehdä automaattisesti Windows-käyttäjätunnusten avulla, jolloin erillistä kirjautumista ei tarvita. Dokumentit voidaan tallentaa M-Filesiin mm. drag-and-drop-toiminnallisuuden avulla tai valitsemalla M-Files dokumentin tallennuspoluksi. Lisäksi M-Files integroituu myös yleisiin toimisto-ohjelmiin, kuten Microsoft Wordiin ja Outlookiin.

M-Filesissa ei käytetä dokumenttien organisoimiseen kansioita vaan näkymiä, joiden sisältö päivittyy automaattisesti, kun uusia dokumentteja tuodaan dokumenttivarastoon. Näkymät ovat tiettyjen metatietojen perusteella tallennettuja hakuja. Sama dokumentti voi tämän vuoksi löytyä useista näkymistä. Kun käyttäjä tallentaa M-Filesiin dokumentin, hänelle näytetään dokumentin metatietokortti. Siihen käyttäjä kuvailee tallennettavaa dokumenttia erilaisilla metatiedoilla. Metatiedon syöttäminen on tärkeää, koska sen avulla dokumentti löydetään myöhemmin helposti M-Filesista. Näkymät voidaan määritellä käyttäjäkohtaisesti.

Dokumentteja on myös mahdollista etsiä käyttämällä hakuominaisuuksia. Haut voidaan kohdistaa dokumentin nimen ja metatietojen lisäksi myös dokumenttien sisältöön.

## 2.2 Suunnitteluperiaatteet ja arkkitehtuuri

M-Files koostuu järjestelmänä M-Files Serveristä ja M-Files Clientistä. M-Files Server on palvelinohjelmisto, joka voi sijaita asiakkaan valitsemasta ratkaisusta riippuen joko pilvipalvelimella tai asiakkaan omalla Windows-palvelimella. M-Files Server käyttää tietojen tallentamiseen ratkaisusta riippuen FireBird-tietokantaa tai Microsoft SQL Serveriä. Dokumentit ja metatiedot on mahdollista tallentaa eri palvelimille käytön nopeuttamiseksi suurilla datamäärillä. [1]

Yksi M-Files Server voi sisältää useita dokumenttivarastoja (vault). Dokumenttivarastot ovat toisistaan erillisiä, ja niiden sisältämät dokumentit eivät näy toisiin saman palvelimen dokumenttivarastoihin. Palvelinten ja niiden sisältämien dokumenttivarastojen asetuksia ja metatietorakenteita muokataan M-Files Server Administrator -työkalulla. [1]

M-Files Client on asiakasohjelmisto, joka tulee asentaa jokaisen M-Files-käyttäjän tietokoneelle. M-Files Clientin asetuksiin määritetään yhteydet M-Files Serverin sisältämiin dokumenttivarastoihin, jonka jälkeen dokumenttivarastojen tiedostot näkyvät M-Files Clientin luoman virtuaaliaseman sisältönä käyttäjän tietokoneella. [1]

M-Files tarjoaa ulkoiseen toiminnallisuuden laajentamiseen ohjelmallisen rajapinnan eli API:n, jonka avulla voi mm. käyttää, muokata ja luoda dokumenttivarastojen dokumentteja [2]. Sisäisen käyttöliittymän räätälöintiin taas tarjotaan UI Extensibility Framework. Se on kokoelma ominaisuuksia, rajapintoja, ajoympäristöjä ja kirjastoja, jotka mahdollistavat M-Files Clientin räätälöinnin add-in-ohjelmien avulla [3].

## 2.3 Metatietorakenne

Metatietorakenne määrittää, millaisia dokumentteja ja metatietoja dokumenttivarastoon voi tallentaa. M-Filesiin on saatavissa paljon valmiita metatietorakenteita, joista voidaan usein pienellä työllä räätälöidä yrityksen tarpeisiin sopiva metatietorakenne.

Käytäntönä on, että kaikki M-Filesiin tallennettavat tiedostot ovat dokumentti-kohdetyyppejä. Metatietoja voidaan määritellä dokumenteille, mutta myös kohteille, jotka eivät sisällä tiedostoja. Tällaisia muita kohdetyyppejä voivat olla esimerkiksi asiakas, projekti tai työntekijä. Kohdetyyppejä voidaan määritellä synkronoitavaksi ulkoisesta lähteestä (ks. 2.6). [1]

Kohdetyyppejä, erityisesti dokumentteja, on mahdollista jakaa erilaisiin luokkiin. Luokkaan määritellään, mitä ominaisuuksia eli metatietoja näytetään oletuksena uuden kohteen metatietokortilla, ja mitä näistä käyttäjän on pakko täyttää. Kohteille on kuitenkin mahdollista lisätä myös luokkaan kuulumattomia ominaisuuksia.

Ominaisuusmäärittely (property definition) on nimetty M-Filesin metatietotyyppi, jolle määritellään, millaista tietoa ominaisuus sisältää: esim. tekstiä, lukuja tai linkkejä toisiin kohteisiin. Ominaisuusmäärittelyn käyttö voi olla rajattu vain yhteen

kohdetyyppiin tai sitä voi käyttää kaikissa kohteissa. Jokaiselle kohteelle tulee määrittää vähintään pakollinen nimiominaisuus.

Kohteiden sisältöön ja metatietoihin tehdyt muutokset tallentuvat M-Filesin muutoshistoriaan. Minkä tahansa dokumentin aikaisemman version voi avata M-Filesista, ja dokumentin voi pysyvästi palauttaa vanhaan versioon. Palauttamisessa ei menetetä uudempia versioita, vaan ne tallentuvat myös muutoshistoriaan.

## 2.4 Työnkulut ja skriptaaminen

Työnkulkujen (workflow) avulla voidaan M-Filesissä kuvata dokumentteihin liittyviä toistuvia prosesseja. Työnkulut koostuvat tiloista (state). Tarvittavat työnkulut ja tilat luodaan dokumenttivarastoon M-Files Server Administrator -työkalulla, kun dokumenttivarastoa otetaan käyttöön. Samalla määritetään, mitkä tilasiirtymät ovat sallittuja. Käyttäjä voi siirtää kohteen seuraavaan tilaan vain, jos hänellä on oikeudet tilasiirtymän tekemiseen. Kohdetyypille voidaan määrittää työnkulun tila, johon uudet kohteet asetetaan niiden luomisen yhteydessä. [1]

Tilasiirtymän yhteyteen on mahdollista määrittää tapahtumia, kuten metatiedon asettaminen tai sähköpostin lähettäminen käyttäjälle. Tilasiirtymiin voidaan kirjoittaa myös skriptejä VBScript-kielellä (Microsoft Visual Basic Scripting Edition), joka mahdollistaa skriptien ajamisen Windows-ympäristöissä ilman erillistä skriptikomponenttia [4]. Skriptien avulla on mahdollista toteuttaa myös monimutkaisempaa toiminnallisuutta tilasiirtymien yhteyteen. Skriptaamalla voidaan toteuttaa myös monimutkaisia esiehtoja tiloihin siirtymiselle ja tiloista poistumiselle sekä monimutkaista ominaisuusarvojen validointia. [1]

## 2.5 Ulkoiset tiedostolähteet ja OCR-moduuli

M-Filesin dokumenttivarastoihin on mahdollista määritellä ulkoisia tiedostolähteitä. Tällöin palvelin tarkkailee esimerkiksi verkkolevyllä sijaitsevaa hakemistoa, jonka sisälle syntyvistä tiedostoista luodaan automaattisesti kohteita M-Filesiin. Ulkoisen tiedostolähteen avulla voidaan esimerkiksi helposti määrittää tietyllä skannerilla skannatut dokumentit siirtymään automaattisesti M-Filesiin. Ulkoiselle tiedostolähteelle on mahdollista määrittää myös, mitä metatietoja tiedostosta luotavalle kohteelle asetetaan.

M-Filesin mukana toimitettavan OCR-moduulin avulla on mahdollista tunnistaa teksti kuvatiedostojen sisällöstä. M-Filesiin määritetään dokumentin sivu ja alue, jolta teksti tunnistetaan sekä ominaisuus, johon tunnistettu teksti tallennetaan. OCR-moduulin avulla voidaan myös PDF-muunnoksen yhteydessä tehdä automaattinen tekstintunnistus, jonka seurauksena M-Filesissa tehdyt haut kohdistuvat myös PDF-dokumenttien sisältöön. [1]

## 2.6 Ulkoiset kohdetyypit

M-Files mahdollistaa tietojen synkronoinnin ulkoisen tietokannan ja M-Filesin kohteiden välillä. Tämä tapahtuu määrittämällä kohdetyyppi ulkoiseksi. Määritettäessä kohdetyyppiä ulkoiseksi siihen määritellään ODBC-yhteysmerkkijono (ODBC connection string) ja SQL-kysely, jonka perusteella kohteita luodaan, poistetaan ja päivitetään M-Filesissä.

Erillisten pluginien avulla tämä toiminnallisuus on mahdollista yhdistää relaatiotietokantojen lisäksi myös muunlaisiin tietovarastoihin. Tällöin käytetty plugin ja yhteysasetukset määritellään yhteysmerkkijonossa normaalien ODBC-yhteysasetusten sijaan. SQL-kyselyn sijaan syötetään pluginikohtainen kyselymerkkijono, jonka perusteella plugin tuottaa SQL-kyselyä vastaavan tulostaulun. Sen sisällön perusteella ylläpidetään kohteiden tilaa samalla tavalla kuin normaalin tietokantayhteyden tapauksessa.

## 2.7 M-Files SAP Connector

M-Files SAP Connector on erillinen plugin-moduuli, joka toteuttaa M-Filesin ulkoisten kohdetyyppien synkronointiin tarkoitetun rajapinnan. SAP Connector mahdollistaa tietojen tuomisen M-Filesiin SAP:in sisäisestä tietokantataulusta. SAP Connectoria käytettäessä ulkoisen kohdetyypin yhteysmerkkijonoon määritetään yhteyteen tarvittavat SAP-palvelimen tiedot. Tällöin ulkoiseen kohdetyyppiin syötetään SQL-kyselyn sijaan SAP Connectorin vaatima kyselymerkkijono. Tämän kyselymerkkijonon perusteella SAP Connector päättelee taulun ja rivit, joista muodostetaan kohteita M-Filesiin, sekä sarakkeet, joista muodostetaan metatietoja kohteille.

SAP Connectorin avulla ei ole mahdollista päivittää SAP:in tietoja. Tämä johtuu siitä, että vaikka SAP:in sisäisten tietokantataulujen tiedon lukeminen on turvallista, taulujen sisällön suora muokkaaminen on kiellettyä. SAP:in tilan muuttaminen on sallittua ainoastaan SAP:in erikoistuneiden transaktioiden avulla. Jos taulujen sisältöön tehtäisiin muutoksia suoraan, riskinä olisi SAP:in sisäisen tilan rikkoutuminen. Koska SAP Connectorilla ei ole mahdollista päivittää SAP:in tietoja, myöskään sen avulla SAP:ista tuotuja tietoja ei ole mahdollista muuttaa M-Filesissä.

SAP Connector käyttää yhteyden muodostamiseen SAP .NET Connectoria. Se on .NET-ympäristöön toteutettu kirjasto, jonka avulla on mahdollista kutsua SAP:in rajapintafunktioita .NET-ohjelmasta. SAP .NET Connector konvertoi .NET-ohjelmasta tehdyt kutsut SAP:in vaatimaan RFC-muotoon ja lähettää ne RFC-protokollan kautta SAP:iin. [5]

Kirjastoa käytetään määrittelemällä yhteysasetukset RfcConfigParameters-olioon. Tämän avulla voidaan luoda RfcDestination-olio ja suorittaa yhteyskokeilu (Ping). RfcDestination-olion sisältämältä RfcRepositorylta pyydetään viite haluttuun SAP:in rajapinnan funktioon. Funktiolle halutut parametrit luodaan kirjaston avulla ja asetetaan

funktioon. Tämän jälkeen funktio ajetaan (invoke). Ajon jälkeen funktion paluuarvot voidaan pyytää funktiolta. [6]

SAP .NET Connectorilla voidaan kutsua myös sellaisia funktioita, jotka tekevät muutoksia SAP:iin. Projektissa tätä ominaisuutta käytettiin toteutettaessa ostotilausten vapauttamista SAP Wrapperiin (ks. sivu 33). Mikäli funktio tekee muutoksia SAP:iin, täytyy ajon jälkeen vielä sitoutua tehtyihin muutoksiin (commit). Tämä tapahtuu ajamalla BAPI\_TRANSACTION\_COMMIT-funktio. Funktion WAIT-parametrilla voidaan valita halutaanko sitoutuminen suorittaa synkronisesti, jolloin odotetaan kunnes sitoutuminen onnistuu tai epäonnistuu, vai asynkronisesti, jolloin ohjelman suoritus saa jatkua jo ennen kuin muutoksiin on sitouduttu. [6]

## 3 SAP ERP

### 3.1 Tehtävä ja tarkoitus

SAP AG (alunperin SAP Systemanalyse und Programmenentwicklung) on vuonna 1972 perustettu saksalainen ohjelmistoalan yritys, jonka päätuote on SAP ERP tuotannonohjausjärjestelmä. [7]

SAP on ensimmäinen yritys mallintaa kaikki keskeiset yrityksen prosessit yhdessä tietojärjestelmässä. Kun SAP:in kehittäminen aloitettiin 70-luvulla, tietokoneohjelmia ajettiin pääasiassa eräajoina. Yhtenä SAP:in keskeisenä tavoitteena ja haasteena oli tehdä ohjelmistoja, jotka reagoivat käyttäjien toimintaan välittömästi. [7]

Tarve SAP:in ja M-Filesin väliseen integraatioon on kasvanut jatkuvasti M-Filesin asiakaskunnan kasvamisen mukana. SAP on hyvin laajasti käytössä, ja SAP-osaaminen on kysyttyä. Projektin alussa tunnistettiin kuitenkin, että SAP-integraatioprojekteihin voi liittyä liiketoimintamahdollisuuksien lisäksi myös riskejä. Aiempien kokemusten perusteella SAP-integraatioprojektit voivat ylittää niille varatut budjetit ja pahimmillaan epäonnistua kokonaan. Juuri mikään muu ohjelmistoyritys ei ole saanut enempää negatiivista julkisuutta käyttöönottoon liittyvien ongelmien vuoksi kuin SAP [8]. Erityisesti tästä syystä tämän projektin määrittely pyrittiin pitämään tarkasti rajattuna tiedostaen, että projekti on vain pieni askel matkalla kohti kokonaisintegraatiota.

Kun SAP:in yhteydessä puhutaan transaktioista, tarkoitetaan eri asiaa kuin puhuttaessa transaktioista tietokantojen yhteydessä. Kaikki toiminta SAP:in käyttöliittymässä tapahtuu erilaisten transaktioiden välityksellä. SAP:issa transaktiolla tarkoitetaan tiettyä SAP:in toiminnallisuutta, johon sisältyy oma käyttöliittymä.

SAP:in käyttäjä käynnistää transaktion kirjoittamalla 4-5 merkkiä pitkän transaktiokoodin. Transaktio on mahdollista etsiä ja käynnistää myös hierarkkisista valikoista. Transaktioon liittyvä ohjelma käännetään ensimmäisellä käynnistyskerralla. Transaktion käyttöliittymän kautta käyttäjä saa tietoa SAP-järjestelmän tilasta ja voi muokata sitä. Yleisesti käytetyt transaktiot voidaan lisätä myös pikavalikkoon, ja näin vältytään jatkossa transaktion etsimiseltä ja transaktiokoodin kirjoittamiselta.

Erilaisia transaktioita on SAP:issa valtava määrä. Esimerkiksi samaan tietosisältöön voi olla useita erilaisia transaktioita. Sama toiminnallisuus voi olla joissain tilanteissa mahdollista saavuttaa useiden transaktioiden kautta. Lisäksi vanhoista transaktioista on olemassa uusia versioita, joissa esimerkiksi käyttöliittymää on parannettu.

Toiminnallisuuden jakaminen transaktioihin on mahdollistanut SAP:in kehittämisen muuttamatta vanhaa toiminnallisuutta. Toimintojen eristäminen on tehnyt mahdolliseksi hallita valtavia määriä erilaista toiminnallisuutta.

Mikäli transaktiot pysyvät yksinkertaisena, tämä SAP:in transaktioihin perustuva arkkitehtuuri on elegantti ja toimiva. Monet SAP:in transaktioista sisältävät kuitenkin

erittäin monimutkaisia tiedon syöttämiseen tarkoitettuja lomakkeita. Jopa lomakkeiden yksittäisiin kenttiin voi liittyä monimutkaisia alirutiineja.

Ilman koulutusta SAP on melko hankala käyttää. Monenlaisiin tilanteisiin sopivien monimutkaisten transaktioiden avulla voi olla vaikea tehdä yksinkertaisia asioita. Virhetilanteissa käyttäjälle näytettävät virheilmoitukset ovat usein luonteeltaan teknisiä. Niinpä ongelmatilanteissa joudutaan kääntymään kalliiden asiantuntijoiden puoleen.

SAP-integraatioihin liittyvän ohjelmistonkehityksen suurimpana ongelmana on toisaalta alkuun pääsemistä helpottavan itseopiskelumateriaalin puute sekä toisaalta tuotantojärjestelmää vastaavan ja harjoitteluun sopivan hiekkalaatikkojärjestelmän hankala saatavuus. SAP:iin liittyvästä osaamisesta on jatkuvasti pulaa, ja juuri tästä johtuen siihen liittyvä koulutus, konsultointi ja ohjelmistonkehitys on kallista.

SAP:in käyttöönottoprojekteja tekevät SAP AG:n sijaan pääasiassa paikalliset partneriyhtykset. Tämä lisää yhden välikäden SAP:in kehittäjien ja käyttäjien välille ja vaikeuttaa SAP:iin liittyvän yksityiskohtaisen teknisen tietämyksen saatavuutta. Käyttäjämäärien kasvaessa suuriksi tällainen kehitys on toisaalta väistämätöntä.

## 3.2 ArchiveLink

ArchiveLink on SAP:in osajärjestelmä, joka mahdollistaa SAP:in luomien dokumenttien tallentamisen ja säilyttämisen erillisessä dokumenttivarastossa, esimerkiksi verkkolevyllä. Ilman ArchiveLink-yhteyttä SAP ei tallenna tulostettavia dokumentteja ollenkaan, vaan se generoi ne aina tarvittaessa. ArchiveLink tarjoaa useita erilaisia tiedonsiirtomekanismeja dokumenttivaraston kanssa kommunikointiin. ArchiveLink-yhteys voidaan määritellä käyttämään mm. HTTP-protokollaa, joka mahdollistaa kommunikaation tätä tarkoitusta varten konfiguroidun web-palvelimen kanssa.

## 3.3 Transaktiot

### 3.3.1 ArchiveLinkin määrittelyyn liittyvät transaktiot

ArchiveLinkin käyttöönotto SAP:issa tapahtuu neljän eri transaktion avulla: OAC0, OAC2, OAC3 ja NACU.

Transaktiolla OAC0 (CMS Customizing content repositories) voidaan määritellä ArchiveLink-tietovarasto. Jokaisella tietovarastolla on kaksi merkkiä pitkä tunniste. OAC0-transaktion käyttöliittymässä tietovarastolle valitaan, mitä tiedonsiirtotapaa ja protokollaa käytetään. Kun käytetään HTTP-protokollaa kirjataan mm. palvelimen verkko-osoite ja siirtohakemisto, johon siirrettävät tiedostot tallennetaan väliaikaisesti. [9]



Transaktiolla OAC3 (SAP ArchiveLink: Links) hallinnoidaan SAP:in dokumenttityyppien ja tietovarastojen välisiä yhteyksiä. Ostotilaukset ovat SAP:issa MEOORDER- ja ostolaskut MIINVOICE-dokumenttityyppejä. OAC3:n avulla nämä dokumenttityypit voidaan linkittää luotuun ArchiveLink-tietovarastoon. Luotujen linkkien perusteella SAP päättelee, mitkä dokumentit tallennetaan mihinkin tietovarastoon.

Transaktiolla OAC2 (SAP ArchiveLink: Global doc. Types) voidaan hallinnoida dokumenttityyppejä ja sitä, minkä formaatin ja tiedostopäätteen mukaisia dokumentteja eri dokumenttityypeistä tallennetaan. Näin voidaan määrittää esimerkiksi, että ostotilaukset tallennetaan PDF-formaatissa.

Transaktiolla NACU (WFMC: Customizing Output Types) voidaan hallinnoida uusien dokumenttien luomiseen liittyviä oletustulostusasetuksia. Mikäli näitä oletusasetuksia ei ole määritetty, käyttäjä joutuu määrittämään tulostusasetukset erikseen jokaiselle luomalleen dokumentille. Tavalliset SAP käyttäjät eivät useinkaan osaa määrittää tulostusasetuksia, minkä vuoksi oletusasetusten määrittäminen on käytännössä yhtä tärkeää kuin teknisesti pakollisten asetusten määrittäminen.

### 3.3.2 Ostotilausprosessiin liittyvät transaktiot

Toteutetussa järjestelmässä käytettyjä ostotilausprosessiin liittyviä transaktioita on viisi: ME21N, MIGO, MIRO, MIR7 ja MIR4.

Järjestelmässä toteutettu tilauksellinen ostoprosessi alkaa ostotilauksen luomisella SAP:issa. Ostotilaukset luodaan transaktiolla ME21N. Transaktion käyttöliittymä on monimutkainen lomake erilaisten ostotilausten tietojen syöttämiseen. Aluksi ostotilaukselle valitaan toimittaja (vendor). Lomakkeen kenttiin voi kirjoittaa suoraan toimittajan tunnistenumeron tai toimittajan nimen. Mikäli käyttäjä ei tiedä toimittajan tunnistenumeroa tai tarkkaa nimeä, hän voi avata hakuikkunan kentän oikeassa reunassa olevasta painikkeesta.

Toimittajan jälkeen ostotilaukselle valitaan osto-organisaatio ja ostoryhmä. Sen jälkeen määritetään tilattavat tuotteet. Mikäli tilattava tuotenimike on syötetty SAP:iin, voidaan luoda tilausrivejä luoda nopeasti tuotenimikkeellä, mutta myös nimikkeettömät tilausrivit ovat mahdollisia.

Kun tuotteet tilataan varastoon, halutaan varmistua, että tuotteet on vastaanotettu varastoon ennen kuin lasku maksetaan toimittajalle. Tällaisiin ostotilauksiin merkitään, että kuittaus tuotteiden vastaanottamisesta vaaditaan.

Edellämainittujen lisäksi ME21N sisältää myös paljon muuta syötettävää tietoa, esimerkiksi maksuehtotietoja, verotietoja, laskutusosoitteita, yhteystietoja. Monet näistä tiedoista määritellään oletuksiksi lomakkeelle tai laskettavaksi automaattisesti muiden valittujen arvojen perusteella. Tämän vuoksi käyttäjän ei tarvitse tavallisesti täyttää kuin osa lomakkeen tiedoista.

Tavaran vastaanottaminen tapahtuu transaktion MIGO avulla. Varastotyöntekijä syöttää saapuneen tavaran mukana toimitetun ostotilausnumeron. Järjestelmä ehdottaa automaattisesti koko tilattua määrää vastaanotetuksi. Työntekijä hyväksyy oletuksen tai kirjaa todellisen vastaanotetun määrän lomakkeelle ja merkitsee tuotteet vastaanotetuksi.

Ostolaskut luodaan transaktiolla MIRO. Tärkeimpänä tietona lomakkeelle syötetään ostotilauksen numero. SAP hakee ostotilausnumeron perusteella lomakkeelle tiedot tilatuista tuotteista, jotka ovat vastaanotettu varastoon. Tämän lisäksi käyttäjä syöttää joitakin muita tietoja kuten laskun numeron, laskun päivämäärän ja laskun loppusumman. Pohjoismaihin konfiguroidussa SAP:issa voidaan määrittää myös laskun viitenumero. Transaktion ME21N tavoin myös transaktioon MIRO voidaan määrittää paljon lisätietoja, joista käyttäjän ei tavallisesti tarvitse välittää.

Kun käyttäjä on syöttänyt laskun tiedot, hän voi simuloida laskun tallentamista järjestelmään, jolloin voi selvittää myös laskun liittyviä ongelmia. Kun käyttäjä tallentaa laskun järjestelmään.

Mikäli ostotilaukset halutaan esikirjata, käytetään transaktiota MIR7. Esikirjaaminen tarkoittaa sitä, että laskun tiedot kirjataan SAP:iin myöhempää laskun luontia varten. Esikirjaamista kutsutaan SAP:issa käytetyssä terminologiassa pysäköinniksi (park) ja luomista lähettämiseksi (post). Esikirjatut laskut lähetetään transaktiolla MIR4. Laskun lähettämistä on mahdollista simuloida kuten transaktiossa MIRO.

## **3.4 Rajapintafunktiot**

### **3.4.1 Yleinen toimintaperiaate**

SAP:in transaktioiden toiminnallisuus on käytettävissä myös ulkoisen rajapinnan (API) kautta. Tätä tarkoitusta varten on olemassa valtava määrä rajapintafunktioita. Rajapintafunktioita kutsutaan RFC:n (Remote Function Call) avulla. SAP:in rajapintafunktiot on toteutettu SAP:ia varten kehitetyllä ABAP-kielellä. Uusia rajapintafunktioita on mahdollista toteuttaa myös itse. [10]

SAP mahdollistaa rajapintafunktioiden kutsumisen synkronisesti (sRFC), transaktionaalisesti (tRFC) ja jonotetusti (qRFC). Synkronisissa kutsuissa kutsuva järjestelmä jää odottamaan vastausta SAP:ista. Toteutetussa järjestelmässä käytettiin vain synkronisia RFC-kutsuja niiden yksinkertaisuuden ja helppokäyttöisyyden vuoksi. [11]

Transaktio SE37 (Function builder) on kehittäjille tarkoitettu monipuolinen työkalu SAP:in rajapintafunktioiden luomiseen, tarkasteluun ja testaamiseen. Transaktion SE37:n avulla voidaan tarkastella rajapintafunktioiden dokumentaation, parametrien ja paluuarvojen lisäksi myös itse toteutusta kooditasolla. Funktioita on mahdollista myös testata käsin syötetyillä parametreilla. Riittäväillä oikeuksilla funktioihin voi myös tehdä

muutoksia. Debug-moodissa myös funktioiden ajaminen rivi kerrallaan on mahdollista. Debug-ympäristö sisältää monipuoliset työkalut muuttujien ajonaikaisen tilan tarkasteluun.

Monipuolinen työkalu auttaa funktioiden tutkimisessa, mutta sopivan funktion löytäminen on yllättävän hankalaa. Nimen perusteella lupaavan kuuloiset funktiot eivät välttämättä toimikaan testattaessa odotetusti, ja lähes saman toiminnallisuuden toteuttavia funktioita voi olla useita. Osa funktioista ovat vanhentuneita, mutta niitä ei ole siitä huolimatta poistettu järjestelmästä. Seuraavaksi esitellään järjestelmässä käytetyt SAP:in rajapintafunktiot.

### **3.4.2 RFC\_READ\_TABLE**

RFC\_READ\_TABLE on todennäköisesti SAP:in yleisimmin käytetty rajapintafunktio. Sen avulla voidaan lukea SAP:in sisäisten tietokantataulujen sisältöä. Funktion suurin puute on se, että sillä ei voi tehdä liitoksia taulujen välille. Tämä ratkaisu on todennäköisesti tehty tehokkuussyistä: liitosoperaatiot ovat tunnetusti raskaita. Liitokset voidaan toki tehdä tarvittaessa asiakaspäässä. Funktio mahdollistaa kuitenkin taulujen rivien valitsemisen sisältöön liittyvillä ehdoilla. RFC\_READ\_TABLE -funktiota käytetään sekä SAP Connectorissa että SAP Wrapperissa.

### **3.4.3 BAPI\_PO\_RELEASE**

SAP:iin on mahdollista määrittää sääntöjä, joiden perusteella SAP päättelee, millaisia ostotilauksia eri käyttäjät saavat luoda. Kun käyttäjä ylittää valtuutensa esimerkiksi luomalla liian suuren tilauksen, SAP siirtää ostotilauksen estotilaan. Ennen kuin ostotilaus voidaan lähettää toimittajalle, tilaus tulee vapauttaa. BAPI\_PO\_RELEASE on funktio, jolla on mahdollista vapauttaa ostotilaus. Vapauttamiseen tarvitaan seuraavan tilan määrittävä kaksi merkkiä pitkä vapautuskoodi (release code). BAPI\_PO\_RELEASE -funktiota käytetään SAP Wrapperissa.

## 4 VALMIIT INTEGRAATIOMODUULIT

### 4.1 KGS:n integraatiomodulit

#### 4.1.1 Yleistä

Järjestelmässä käytettiin KGS:n dokumentti-integraatiomodulleja. Vaikka moduulien toiminnallisuus olisi ollut mahdollista toteuttaa itse, moduulien tekijöillä oli paljon arvokasta osaamista, jonka hankkiminen muuten olisi ollut hyvin kallista ja aikaavievää. Ohjelmistolisenssejä ostettaessa ostetaan samalla arvokasta osaamista. SAP-osaaminen ja -koulutukset ovat hinnoiteltu erityisen korkealle. Ottamalla yhteistyökumppaneiksi osaavia tahoja voitiin integraation toteutuksessa tukeutua näihin tahoihin ja saada vastauksia hankaliin kysymyksiin ilman kallista konsultointia tai koulutusta.

#### 4.1.2 KGS Content Server

KGS Content Server on KGS Softwaren julkaisema integraatiomoduli, joka on tarkoitettu helpottamaan dokumentinhallintajärjestelmien integrointia SAP:iin. SAP voidaan konfiguroida käyttämään dokumenttien tallennuspaikkana ArchiveLink-varastoa. KGS Content Server toimii ArchiveLink-varastona ja tallentaa SAP:in tallentamat tulostettavat dokumentit edelleen dokumentinhallintajärjestelmään.

KGS Content Server asennetaan Apache web-palvelimelle ja konfiguroidaan selaimella käytettävän konfigurointisivuston avulla. SAP ArchiveLink ottaa web-palvelimeen yhteyden http-protokollan avulla ladatessaan ja tallentaessaan tulostettavia dokumentteja ArchiveLink-varastoon. [12]

KGS Content Server vaatii toimiakseen dokumentinhallintajärjestelmäkohtaisen SAPALINK-moduulin, joka toimii adapterina KGS Content Serverin ja dokumentinhallintajärjestelmän välillä. [13] SAPALINK-moduuliin ohjelmoidaan tapa, jolla tulostettavat dokumentit ja siihen liittyvät tiedot tallennetaan KGS Content Serveriin liitettävään dokumentinhallintajärjestelmään sekä se, miten tallennetut dokumentit löytyvät dokumentinhallintajärjestelmästä. M-Filesin ohjelmointirajapintaa käyttävä SAPALINK-moduuli oli toteutettu M-Files Oy:ssä jo aikaisemman asiakasprojektin yhteydessä.

SAPALINK-moduuli tallentaa dokumentin tunnistetiedot M-Filesiin metatiedoiksi. Näitä dokumentin tunnistetietoja varten M-Filesiin luodaan viisi ominaisuusmäärittelyä. Näitä tietoja tarvitaan, että löydetään SAP:in pyytämä dokumentti M-Filesistä. Nämä tekniset tunnistamiseen tarvittavat ominaisuudet on tarkoitettu vain ohjelmallisesti luettavaksi, ja sen vuoksi ne on järkevää piilottaa käyttäjiltä.

### 4.1.3 KGS Index Download

KGS Index Download on KGS Content Serverin osajärjestelmä, joka mahdollistaa SAP:issa olevan tiedon lataamisen SAP:ista KGS Content Serverin kautta tallennetuille dokumenteille. KGS Index Downloadin yhteysasetukset määritellään KGS Content Serverin konfiguraatioikkunassa. KGS Index Download asennetaan KGS Content Serverin mukana, mutta se vaatii erillisenä myytävän lisenssiavaimen. [12]

KGS Index Download -järjestelmän mukana toimitetaan SAP funktiomoduli, joka tulee ladata SAP:iin. Tämän ZKGS\_IDX funktiomodulin Z\_KGSKEYEXPORT-funktiota muokkaamalla voidaan määrittää, mitä metatietoja tuodaan millekin dokumenttityypille. Funktio on toteutettu ABAP-kielellä, ja siihen voi tehdä muutoksia transaktion SE37 avulla. Järjestelmää varten funktiota muokattiin niin, että SAP:in sisäisestä taulusta haetaan luodun ostotilauksen tunnistenumero ja asetetaan tämä paluuarvoon, josta se luetaan M-Filesiin luotavan dokumentin metatiedoiksi. Tällä tavoin osataan yhdistää ostotilaukset oikeaan ostotilaukseen.

M-Files mahdollistaa mm. ominaisuusmäärittelyjen merkitsemisen aliaksella eli tunnistemerkkijonolla. Tämän vuoksi metatietojen lisääminen ei vaadi muutoksia käännettävään koodiin. Riittää, että lisätään aliaksia M-Filesin ominaisuusmäärittelyille ja muokataan SAP:issa olevaa Z\_KGSKEYEXPORT-funktiota transaktiolla SE37.

SAP vaatii funktioiden muuttamiseen käyttäjältä erillisen kehittäjäavaimen (Developer key). Tämä avain pyydetään SAP Basis adminilta eli SAP:in järjestelmänvalvojalta. Funktioon tehdyistä muutoksista syntyy SAP:iin siirtopyyntö (transport request). Nämä siirtopyynnöt täytyy vielä hyväksyä transaktiolla SE10 ennen kuin päästään näkemään muutosten lopullinen vaikutus. Muutoksia tekevät siirtopyynnöt tallentuvat SAP:iin, ja niiden avulla on myös mahdollista tehdä samat muutokset toisiin SAP-järjestelmiin. Tämä vähentää käsin tehtävän työn määrää, ja sen vuoksi inhimillisten virheiden riski pienenee.

### 4.1.4 KGS Activator

KGS Activator on KGS Softwaren julkaisema integraatiomoduli, joka on tarkoitettu helpottamaan dokumentinhallintajärjestelmien integrointia SAP:iin. Projektin kannalta tärkein KGS Activatorin ominaisuus oli se, että sen avulla on mahdollista luoda laskuja SAP:iin. [14] Laskujen luominen on toki mahdollista myös ilman KGS Activatoria kutsumalla suoraan SAP:in rajapintafunktioita. KGS Activatorin avulla tämä on kuitenkin mahdollista ilman erityistä ymmärrystä SAP:ista.

M-Files Oy:n aikaisemman projektin yhteydessä KGS Activatoria varten oli toteutettu yksinkertainen SAP Activator -kääremoduuli. Moduuli päätettiin jättää integraatiossa paikalleen.

## 4.2 ExternalObjectTypeRefresher

M-Files päivittää ulkoiset kohdetyypit kerran vuorokaudessa neljältä aamuyöllä. Ohjelmointirajapinnan kautta päivittyminen on kuitenkin mahdollista käynnistää funktiokutsulla. ExternalObjectTypeRefresher on yksinkertainen .NET ympäristöön toteutettu ohjelma, joka käynnistää konfiguraatitiedostoon määritettyjen ulkoisten kohdetyypien päivittymisen M-Filesin ohjelmistorajapinnan kautta. Päivitettävät ulkoiset kohdetyypit määritetään asetustiedostoon.

Windows-käyttöjärjestelmän ajastettujen tehtävien avulla ohjelman avulla ulkoiset kohdetyypit voidaan asettaa päivittymään viiden minuutin välein. Suurilla päivitettävillä tietomäärillä tämä saattaa tosin aiheuttaa kuormitusta verkolle ja palvelimille.

## 5 INTEGRAATION TOTEUTUS

### 5.1 Vaatimukset järjestelmälle

Tarve M-Filesin ja SAP:in integraatiolle lähti siitä, että asiakas halusi käsitellä ja arkistoida SAP:in tuottamia ostotilauksia ja toimittajilta saatuja ostolaskuja M-Filesissä. Asiakkaan työntekijöillä oli positiivisia kokemuksia M-Filesin käytöstä tavallisten työpöytäohjelmistojen tuottamien dokumenttien hallinnassa. Samaan aikaan integraation toteutuksen kanssa asiakasyrityksessä oli käynnissä SAP:in käyttöönotto. Yhtenä syynä M-Files:in ja SAP:in integraatioon oli todennäköisesti myös halu pehmentää SAP:in käyttöönottoa ennestään tutulla ja helppokäyttöisenä pidetyllä järjestelmällä.

Asiakkaan kanssa pidetyissä palaverissa tuli myös selväksi, että järjestelmän yksi tehtävä oli vaikuttaa yrityksessä vallitseviin ostotilausprosesseihin. Ongelmana oli, että yrityksessä tehdyistä ostoista ei usein tehty ostotilausta, ja tilaukset näkyivät johdolle siksi vasta laskutusvaiheessa. Mahdollisuus kirjata järjestelmään tilauksettomia laskuja haluttiin kuitenkin mukaan järjestelmään, mutta tämän haluttiin olevan hankalampaa kuin tilauksellisten laskujen kirjaaminen. Asiakas myös toivoi, että tilauksettomista laskuista voitaisiin päästä lähes kokonaan eroon tulevaisuudessa, ja sen vuoksi niihin liittyvään toiminnallisuuden toteuttamiseen ei haluttu käyttää paljon resursseja. Ostoprosessin eri vaiheissa järjestelmän haluttiin muistuttavan prosessiin osallistuvia henkilöitä sähköpostilla.

Ohjelmistokehityksen ketteryuden vuoksi lopulliset vaatimukset järjestelmälle voidaan lukea järjestelmän kuvauksesta. Projektin alussa tunnistetut vaatimukset tiedettiin riittämättömiksi, mutta niiden avulla oli kuitenkin mahdollista aloittaa järjestelmän kehittäminen.

### 5.2 Suunnitteluratkaisut

Yhtenä projektin tavoitteena oli käyttää mahdollisimman paljon valmista toiminnallisuutta ja välttää mahdollisuuksien mukaan muutosten tekemistä olemassaoleviin järjestelmiin. Syynä tämänsuuntaisten tavoitteiden muodostumiseen on toisaalta ajansäästö ja toisaalta pelko aikaisemmin toteutetun toiminnallisuuden rikkoontumisesta. Erityisesti SAP:in sisäisen toiminnallisuuden muokkaaminen ilman tähän soveltuvaa koulutusta koettiin vaaralliseksi. Tiedossa oli myös, että SAP:iin tehtävät muutokset jouduttaisiin hyväksyttämään SAP:in käyttöönotosta ja ylläpidosta vastaavalla yrityksellä. Näiden syiden vuoksi haluttiin välttää erityisesti SAP:iin tehtäviä muutoksia.

Integraatiota lähdettiin kehittämään ketterällä ohjelmistonkehitysprosessilla. Ketterä ohjelmistonkehitys soveltuu erityisesti sellaisiin asiakasprojekteihin, joissa kaikkia asiakkaan tarpeita ei ole mahdollista määrittellä ennen toteutusvaiheen aloittamista.

Projektin alussa määrittely ja sopimusneuvottelut tapahtuivat pääasiassa konsulttien ja asiakkaan välillä. Jos projektin haastavuus olisi osattu ennakoida, olisi ohjelmistonkehittäjän ja asiakkaan välinen suora viestintä kannattanut aloittaa heti projektin alussa. Tällä olisi voitu välttää integraation määrittelyyn liittyviä epäselvyyksiä. Toisaalta ohjelmistonkehittäjän mukaan tuominen liian aikaisessa vaiheessa voi tarpeettomasti monimutkaistaa alkuvaiheen neuvotteluja ja määrittelyjä pienissä ja riskittömissä projekteissa.

Vähäinen kokemus vastaavista projekteista vaikeutti projektin ongelmakohtien tunnistamista etukäteen. Projektin aikana määrittelyihin tuli useita kertoja muutoksia ja lisäyksiä. Nämä pystyttiin ottamaan ketterästi mukaan projektin laajuuteen.

### **5.3 Kehitysympäristö**

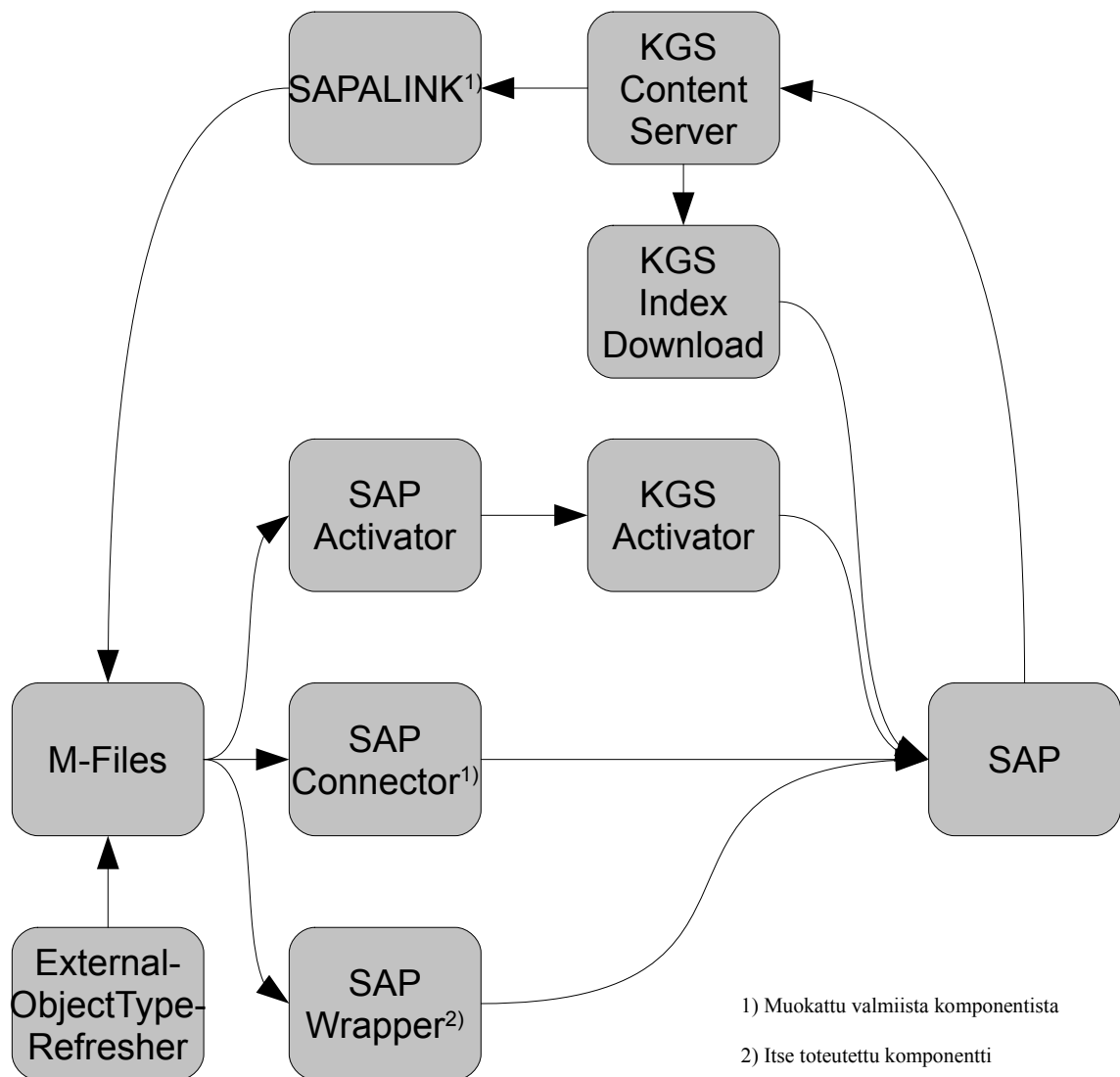
Aluksi integraatiota kehitettiin sisäistä tuotekehitystä ja demonstraatioita varten asennetun SAP:in kanssa. Noin puolessavälissä projektia aukesi mahdollisuus siirtyä kehittämään integraatiota etäyhteyden välityksellä lopullista tuotantojärjestelmää konfiguraatioiltaan vastaavaan SAP hiekkalaatikkoon (sandbox). Projektin päätteeksi integraatio siirrettiin vielä testijärjestelmään ennen tuotantojärjestelmään siirtämistä. Tässä kuvatut vaiheet tulivat projektille sen kanssa samaan aikaan käynnissä olevasta SAP-käyttöönottoprojektista.

### **5.4 Järjestelmän kuvaus**

#### **5.4.1 Järjestelmäarkkitehtuuri**

Aikaisemmin M-Files Oy:ssä toteutettu demonstraatiojärjestelmä oli luonnollinen lähtökohta projektin järjestelmälle. Demonstraatiojärjestelmässä oli toteutettu tilauksettomien ostolaskujen luonti SAP:iin M-Filesissa olevan metatiedon perusteella. Järjestelmä loi uuden laskun SAP:iin KGS Activatorin avulla ja linkitti skannatun laskun luotuun laskuun KGS Content Serverin avulla. Projektin alussa ajatus oli, että olemassaoleva toiminnallisuus pienin lisäyksin voisi kohtuullisen nopeasti tuottaa asiakkaan vaatiman järjestelmän. Tästä johtuen retrospektiivisesti ajatellen ilmiselvä näkökulma toteuttaa koko järjestelmä yhtenä integraatiomoduulina ei tullut projektin alussa edes mieleen. Toisaalta projektin alussa myöskään tekninen osaaminen ei ollut riittävällä tasolla tällaisen kokonaisjärjestelmän toteuttamiseen. Tällainen järjestelmän kehittäminen kohti yhtä integraatiomoduulia on kuitenkin selvästi jatkokehitystavoite projektille.





**Kuva 1:** Korkean tason arkkitehtuurikaavio

Korkean tason arkkitehtuurikaaviosta (Kuva 1) nähdään, että yksi järjestelmän heikkous on sen koostuminen monesta komponentista. Toisaalta yhtenä projektin tavoitteena oli säästää aikaa käyttämällä mahdollisimman paljon valmiita toiminnallisuutta. Kaikki moduulit SAP Wrapperia lukuunottamatta ovat projektissa hyödynnettyjä valmiita moduuleja. SAP Wrapper on toteutettu kokonaan projektia varten. Projektin sisällä tehtiin myös muutoksia SAP Connectoriin ja SAPALINK-moduuliin.

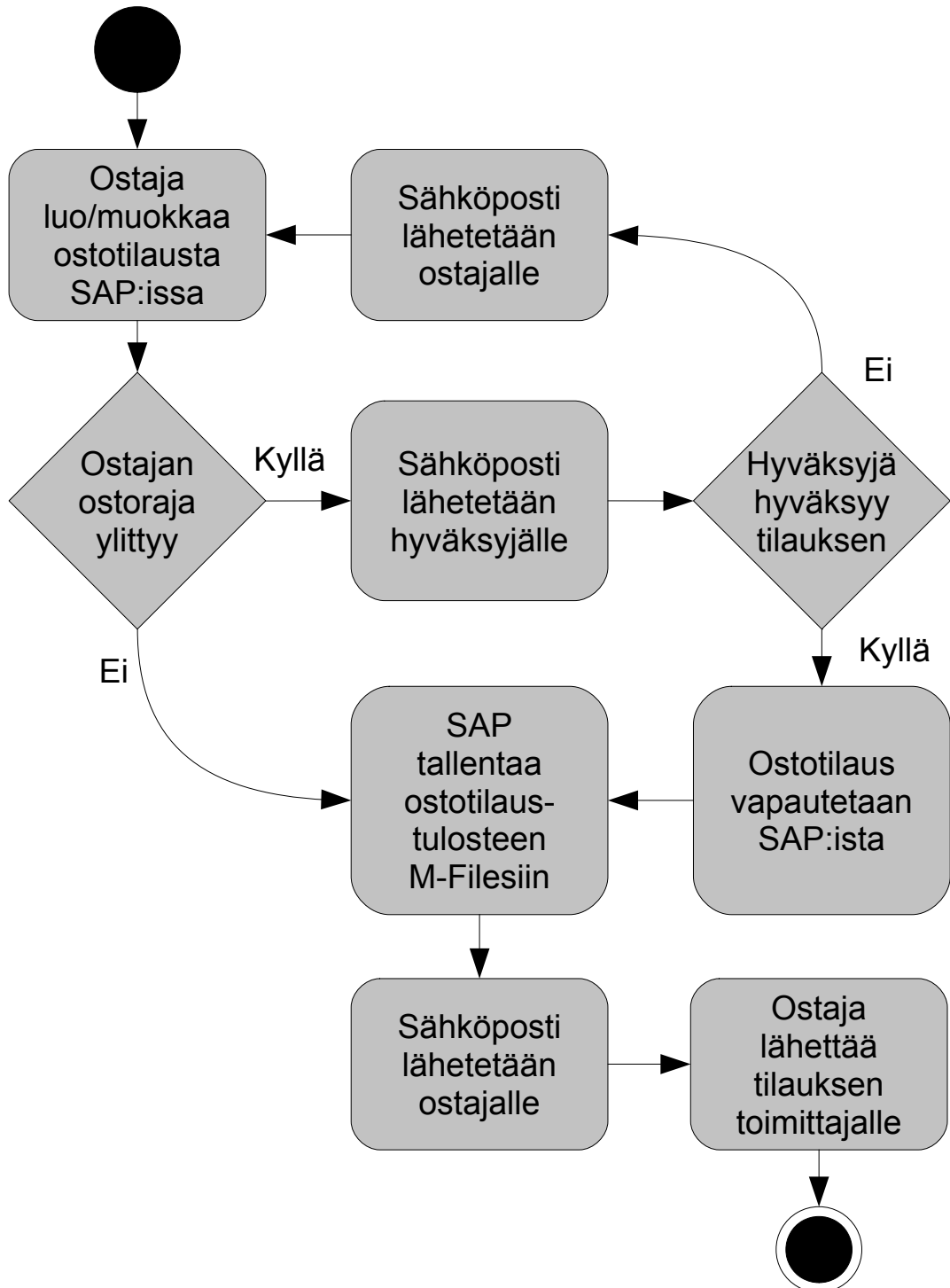
Järjestelmän osilla on erilaisia tehtäviä. SAP ArchiveLink tallentaa PDF-muotoiset ostotilaustulosteet http-protokollan avulla KGS Content Serveriin. KGS Content Server pyytää dokumenteille metatietoja KGS Index Download -osajärjestelmästä, joka kutsuu ZKGS\_IDX-funktiomodulin Z\_KGSKEYEXPORT-funktiota SAP:in rajapinnassa.

KGS Content Server tallentaa dokumentit metatietoineen edelleen SAPALINK-muuntimen kautta M-Filesiin. [12]

M-Filesin tilasiirtymiin määritetyt skriptit kutsuvat SAP Activatoriin käärittyä KGS Activator -moduulia, joka kutsuu SAP:in ZKGS\_DMSIMP-funktiomodulin funktioita SAP:in rajapinnassa. [14] Samoista skripteistä kutsutaan myös SAP Wrapper -moduulia, joka kutsuu SAP:in rajapinnan RFC\_READ\_TABLE- ja BAPI\_PO\_RELEASE-funktioita.

ExternalObjectTypeRefresher kutsuu M-Filesia ajastetusti ja pyytää päivittämään ulkoiset kohdetyypit. Tämä aiheuttaa sen, että M-Files kutsuu SAP Connectoria. SAP Connector kutsuu SAP:in rajapinnan RFC\_READ\_TABLE-funktiota hakien M-Filesin ulkoisiin kohdetyyppeihin määriteltyjen SAP:in taulujen sisältöjä. Mikäli SAP:in taulun tila ei vastaa M-Filesin ulkoisen kohtetyypin tilaa, SAP Connector päivittää ulkoisen kohtetyypin kohteet. Tämä ulkoisten kohtetyyppien päivitys tehdään ajastetusti viiden minuutin välein käyttäen apuna Windows-käyttöjärjestelmän ajastettuja tehtäviä (Scheduled tasks).

## 5.4.2 Ostoprosessi

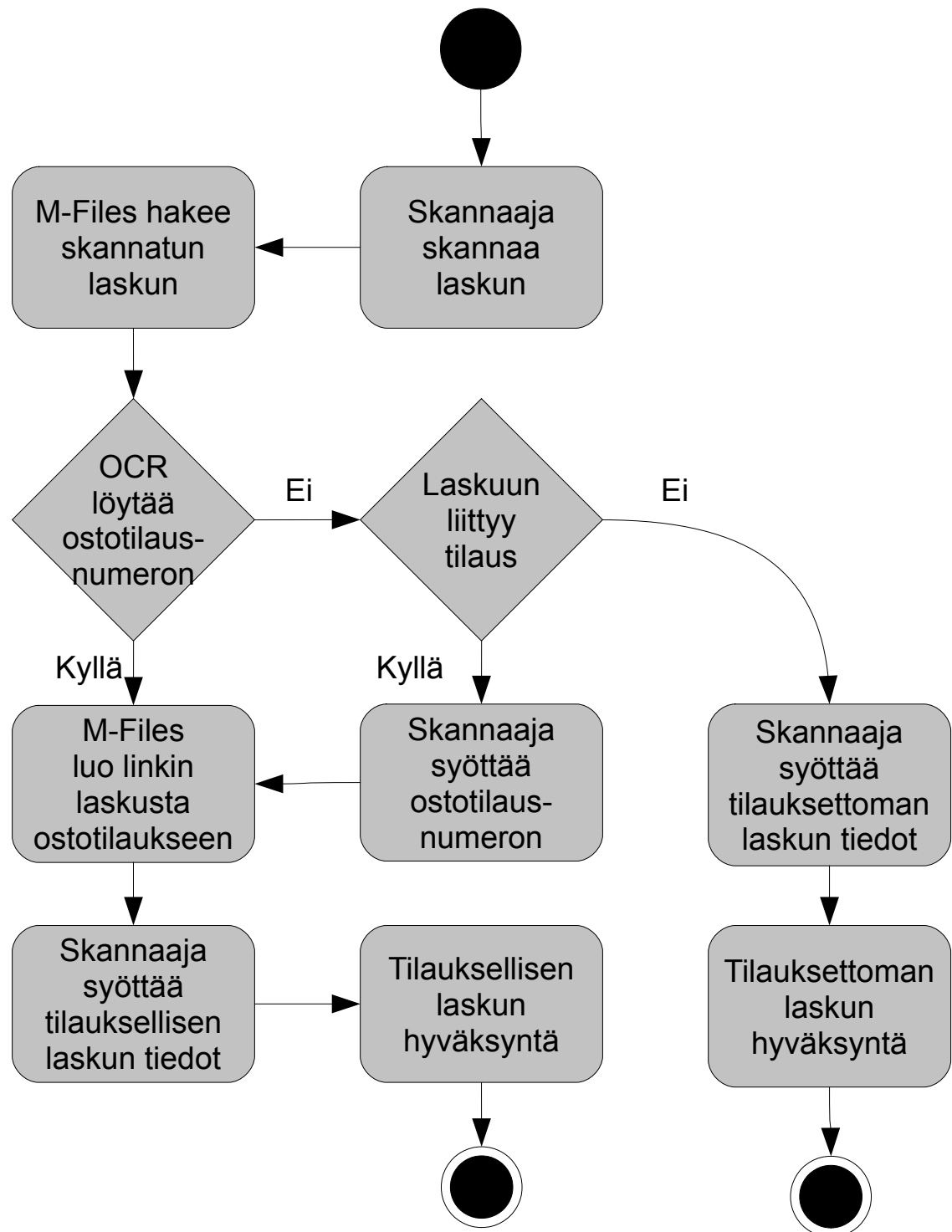


**Kuva 2:** Ostotilauksen hyväksyntäprosessi

Ostotilausprosessi alkaa siitä, kun ostaja luo ostotilauksen SAP:iin (Kuva 2). Ostotilauksen tiedot syötetään SAP:in transaktiolla ME21N (kuva). Mikäli ostotilauksen

loppusumma ylittää SAP:iin ostoryhmälle määritetyt rajat, ostotilaus vaatii hyväksynnän M-Filesissä. Tällöin ostotilauksen hyväksyjä saa sähköpostin, jossa olevasta linkistä hän pääsee tarkastelemaan tilauksen tietoja M-Filesissä. Siirtämällä tilauksen M-Filesissa hyväksyty-tilaan hän vapauttaa ostotilauksen SAP:issa.

Kun ostotilaus on vapautettu, SAP luo ostotilaustulosteen, joka tallennetaan M-Filesiin. Ostaja saa sähköpostissa linkin, jonka kautta hän pääsee tarkastelemaan syntynyttä ostotilaustulostetta, joka odottaa lähettämistä toimittajalle. Kun ostaja lähettää ostotilaustulosteen toimittajalle, hänen on mahdollista merkata ostotilaustuloste lähetetyksi M-Filesissä. Toimittaja saa ostotilauksen ja lähettää tilatut tuotteet ja laskun takaisin tilanneelle yritykselle.

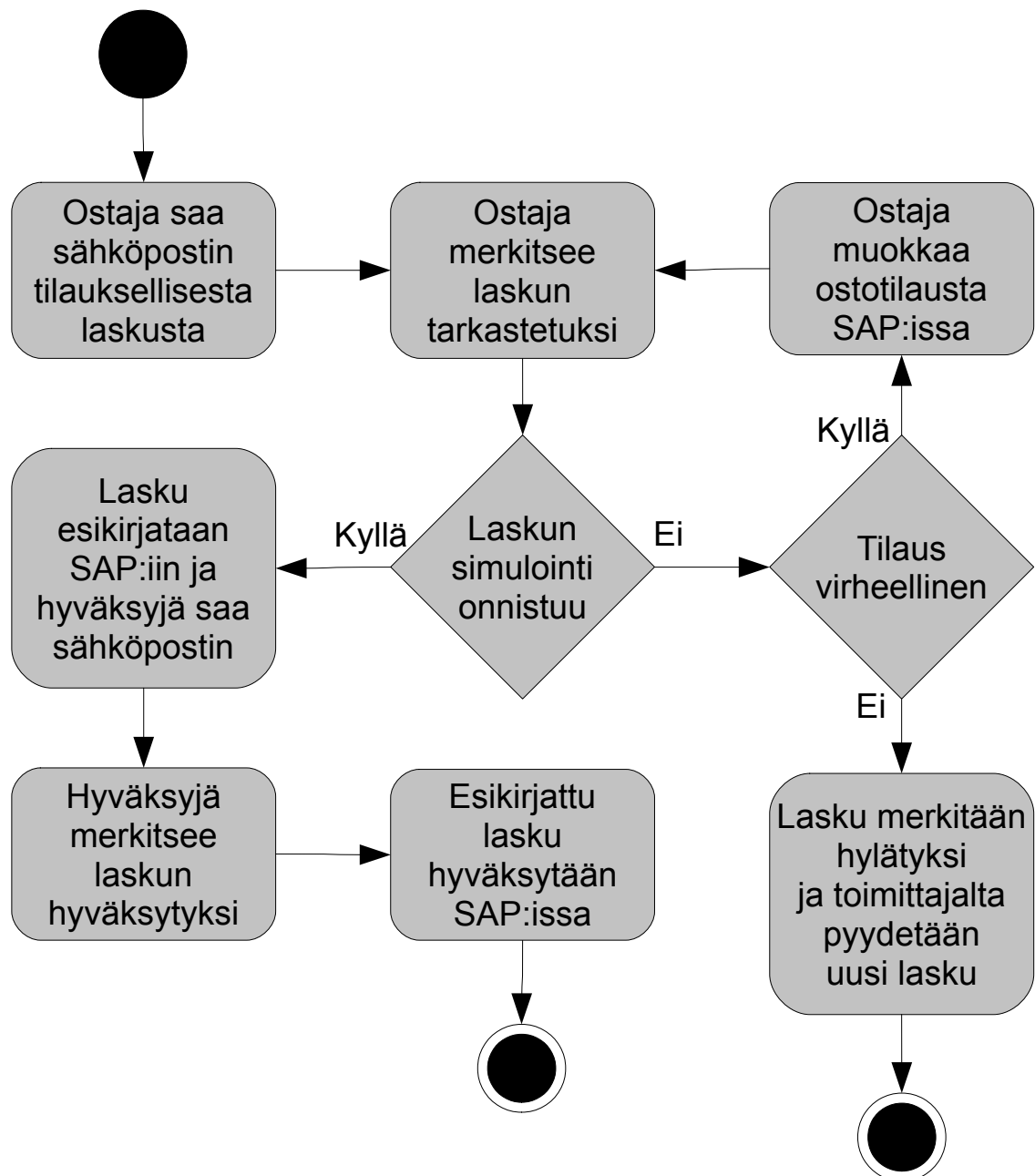


**Kuva 3:** Laskun skannaaminen ja syöttäminen järjestelmään (jatkuu kuviin 4 ja 5)

Ostolaskun käsittely alkaa, kun saapunut lasku luetaan skannerin avulla M-Filesiin ja ostotilauksen tunnistenumero etsitään automaattisesti OCR-moduulin avulla (Kuva 3). Skannaaja syöttää käsin laskun numeron, laskun päivämäärän, laskun loppusumman ja

laskun viitenumeron M-Filesiin syntyneelle laskukohteelle ja merkkää laskun vastaanotetuksi. Laskun kirjauspäivämääräksi asetetaan oletuksena nykyinen päivämäärä, mutta mikäli halutaan tästä poikkeava kirjauspäivämäärä, on se mahdollista syöttää myös käsin.

Jos OCR-moduuli ei löydä järjestelmään tallennettuja ostotilauksia vastaavaa tunnistenumeroa laskulta, skannaajan tulee selvittää, onko kyseessä tilauksellinen vai tilaukseton lasku. Jos lasku on tilauksellinen, skannaaja valitsee laskulle oikean ostotilauksen. Tilauksettomiin laskuihin ei liity järjestelmään tallennettua ostotilausta.

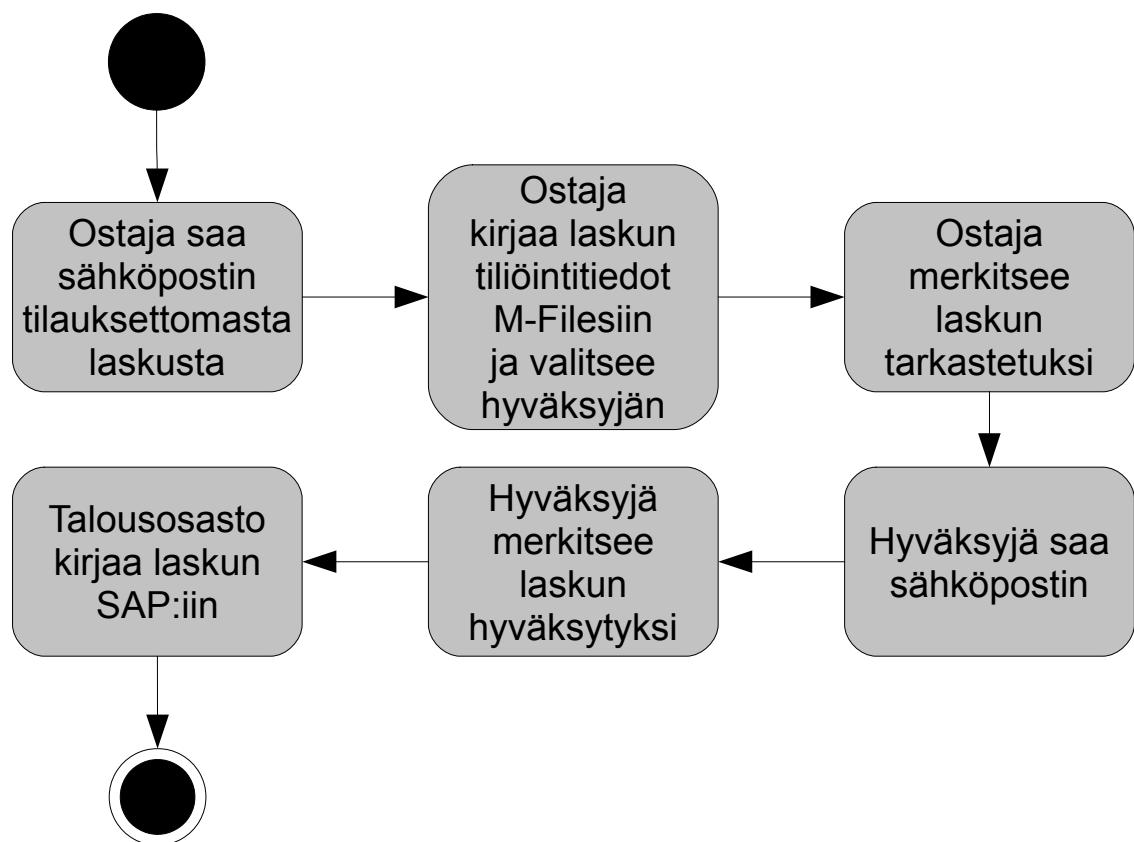


**Kuva 4:** Tilauksellisen laskun hyväksyntä (jatkuu kuvasta 3)

Kun skannattu lasku on tilauksellinen, ostaja saa sähköpostissa linkin laskuun, joka odottaa tarkastusta (Kuva 4). Tarkastettuaan, että laskun tiedot ovat oikein, hän merkitsee laskun tarkastetuksi. Tällöin laskun lähettämistä simuloidaan ja lasku esikirjataan SAP:iin. Mikäli laskun tiedoissa on vikaa, ostaja saa tiedon siitä M-Filesin virhediologissa. Hän voi korjata ostotilausta SAP:issa vastaamaan saatua laskua tai hylätä laskun ja pyytää toimittajaa lähettämään ostotilausta vastaavan laskun.

Mikäli tuotteet on tilattu varastoon, tavaran vastaanottamisen yhteydessä varastovastaava kirjaa tavaran vastaanotetuksi SAP:issa. Mikäli tuotteet on tilattu varastoon, mutta tuotteita ei ole vielä vastaanotettu, ostaja saa tiedon siitä M-Filesin virhediologissa tarkastuksen yhteydessä. Hän voi selvittää toimittajalta ja varastosta tavaroiden toimituksen tilaa tai jättää laskun odottamaan kunnes tavarat on vastaanotettu. Ostaja ei voi siis merkitä laskua tarkastetuksi ennen kuin tilatut tavarat on merkattu vastaanotetuksi SAP:issa.

Kun lasku on onnistuneesti tarkastettu, laskun hyväksyjä saa sähköpostissa linkin laskuun, joka odottaa hyväksyntää. Päätettyään hyväksyä laskun, hän siirtää kohteen hyväksyty-tilaan. Tällöin esikirjattu lasku lähetetään SAP:issa. Mikäli lasku hylätään missä tahansa vaiheessa, esikirjaus poistetaan SAP:ista.



**Kuva 5:** Tilauksettoman laskun hyväksyntä (jatkuu kuvasta 3)

Vaikka asiakasyrityksessä oli pyrkimys luopua tilauksettomista laskuista, oli tiedossa, että tämä vaatisi kohtuullisen pitkän siirtymävaiheen. Tämän vuoksi ostoprosessi oli toteutettava myös tilauksettomille laskuille. Koska ostoprosessia haluttiin kehittää ostotilausten suuntaan, asiakas halusi, että tilauksettomien laskujen käsittely jätetään mahdollisimman paljon käyttäjien vastuulle. Niinpä haluttiin, että tilauksettomia ostotilauksia ei automaattisesti syötetä SAP:iin, vaan tämä jätetään tarkoituksella käyttäjien tehtäväksi. Tarkastus- ja hyväksymisprosessi haluttiin toteuttaa kuitenkin myös tilauksettomille laskuille.

Järjestelmän näkökulmasta tilaukseton ostoprosessi alkaa siitä, että saapunut lasku skannataan M-Filesiin (Kuva 3). Ostotilauksen tunnistenumeroa etsitään OCR-moduulin avulla, mutta sitä ei löydetä, koska laskuun ei liity ostotilausta. Skannaaja syöttää M-Filesiin laskun tiedot, valitsee laskulle ostajan ja merkitsee, että kyseessä on tilaukseton lasku.

Ostaja saa sähköpostissa linkin laskuun, joka odottaa tarkastusta (Kuva 5). Ostaja syöttää laskun tiliointitiedot ja hyväksyjän M-Filesiin ja merkitsee laskun tarkastetuksi. Hyväksyjä saa sähköpostissa linkin laskuun, joka odottaa hyväksyntää. Päätettyään hyväksyä laskun hyväksyjä merkitsee laskun hyväksytyksi. Lopuksi talousosasto kirjaa laskun SAP:iin.

## **5.5 Toteutetut työkulut M-Filesissä**

### **5.5.1 Työnkuluihin perustuvasta ratkaisusta**

Päätös toteuttaa ratkaisu pääasiassa M-Filesin työnkulkujen avulla vaikutti projektin alussa ilmiselvältä. Ostotilausten ja -laskujen käsittely etenee vaiheittain tilasta toiseen suoraviivaisesti. Tämän lisäksi tilasiirtymiin voidaan määritellä skripteillä toiminnallisuutta, esimerkiksi kommunikointia SAP:in kanssa. Tilasiirtymän mahdollistaminen tarjoaa helpon tavan lisätä M-Filesin käyttöliittymään toiminnallisuutta.

Ratkaisussa on myös huonoja puolia. Koska integraatiossa käytetään työnkulkuja ja kohde voi kuulua vain yhteen työnkulkuun kerrallaan, SAP-integraatioon liittyville kohteille ei voi määritellä muita työnkulkuja. Käytettyyn työnkulkuun voidaan tosin tarvittaessa lisätä uusia tiloja.

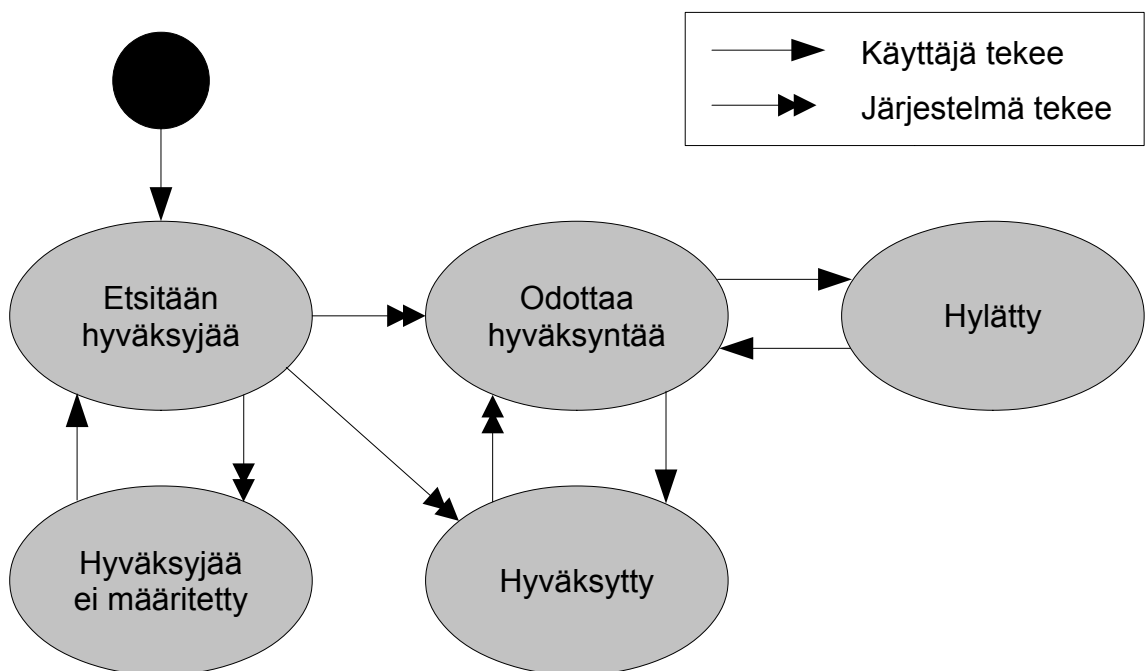
Toiminnallisuuden toteuttaminen VBScriptillä oli vaivalloista ja aikaavievää verrattuna saman toteuttamiseen esimerkiksi .NET-ympäristössä, koska hyvää kehitysympäristöä ei ollut saatavilla. Monet projektin alussa lyhyiltä ja yksinkertaisilta vaikuttaneet skriptit paisuivat projektin aikana toiminnallisten vaatimusten lisääntyessä. Toisaalta kutsumalla skripteistä suoraan integraatiomodulleja välttyttiin yhdeltä välimoduulikerrokselta, joka olisi saattanut monimutkaistaa arkkitehtuuria entisestään.



Tilasiirtymiin liittyvää toiminnallisuutta on toteutettu onnistuneesti toisissa järjestelmissä myös tarkkailemalla tasaisin väliajoin kohteiden tiloja M-Filesin ulkopuolelta. Tällöin skriptaamista ei tarvita, mutta toiminnot tapahtuvat pienellä viiveellä. Viive voi käyttäjän turhautumisen lisäksi joskus aiheuttaa vaikeasti ennakoitavia ja vakavia synkronointiongelmia, minkä vuoksi tilasiirtymäskriptien käyttäminen vaikuttaa tässä tapauksessa paremmalta vaihtoehdolta.

Työnkulkuihin perustuva ratkaisu vaikuttaa siis myös jälkikäteen melko hyvältä ratkaisulta. Yhden välimoduulikerroksen lisäämisestä ei tosin välttämättä olisi haittaa integraatiolle. Välimoduulikerroksen avulla toiminnallisuutta voitaisiin vähentää skripteistä, joista tarvitsisi vain kutsua välimoduulikerrosta. .NET-ympäristöön kirjoitetun koodin ylläpitäminen olisi huomattavasti helpompaa kuin skriptien ylläpitäminen. Tämä voisi olla myös askel kohti kokonaisintegraatiota.

### 5.5.2 Ostotilausten hyväksyntä



Kuva 6: Ostotilausten hyväksyntä - työnkulun tilat ja tilasiirtymät

Mikäli SAP:iin luodun ostotilauksen loppusumma alittaa ostavalle henkilölle määritetyn rajan, ostotilausten hyväksyntää ei tarvita. Tällöin SAP tallentaa ostotilaustulosten ArchiveLink-rajapinnan ja KGS Content Serverin kautta M-Filesiin. Syntynyt ostotilaustuloste viittaa ostotilausnumerolla ostotilaukseen. Tämän seurauksena M-Files tuo SAP:ista SAP Connectorin avulla myös puuttuvan ostotilauksen M-Filesiin. Ostotilaus siirtyy tässä tapauksessa suoraan alkutilasta Hyväksytty-tilaan (Kuva 6).

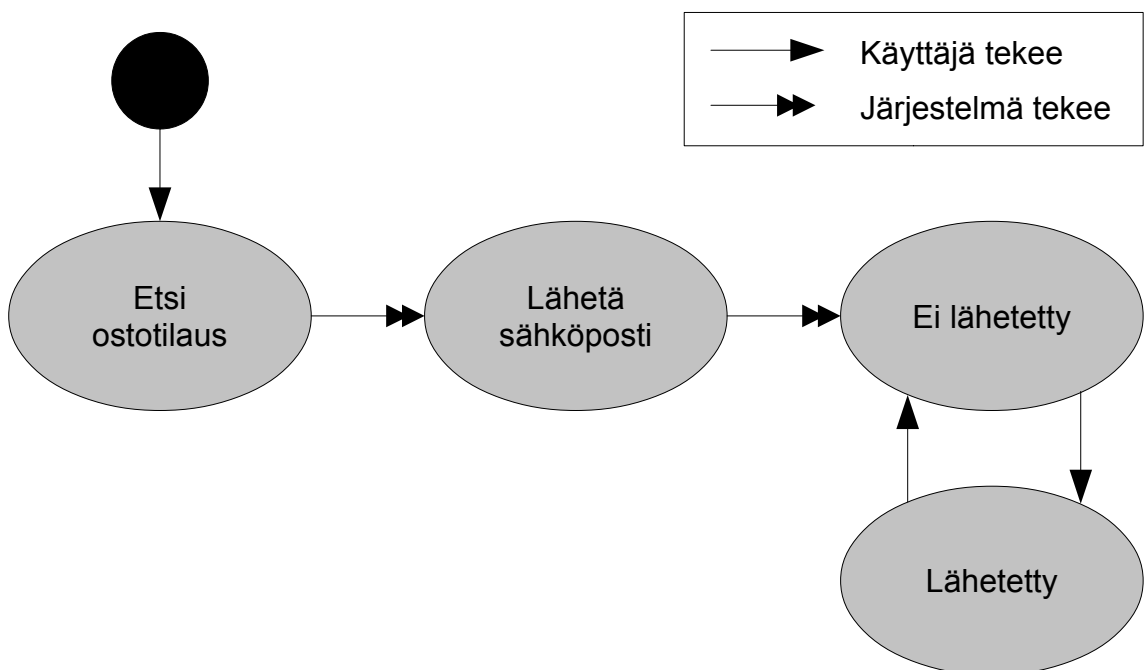
Ostotilausten hyväksyntää tarvitaan, kun ostotilauksen loppusumma ylittää ostavalle henkilölle määritetyn rajan. Tällainen ostotilaus siirtyy SAP:issa estotilaan, ja SAP ei tuota ostotilaustulostetta. Tilanne havaitaan SAP:in ulkopuolella seuraavan kerran, kun M-Filesin ulkoiset kohdetyypit päivittyvät. Tällöin estotilassa olevaa ostotilausta vastaava ostotilaus syntyy M-Filesiin ja sen Release indicator -ominaisuudesta voidaan tunnistaa sen olevan estotilassa.

Mikäli ostotilauksen ensimmäiselle riville on määritelty kustannuspaikka, lähetetään sähköposti-ilmoitus hyväksymistä odottavasta ostotilauksesta kustannuspaikan esimiehelle. Jos kustannuspaikkaa ei ole määritetty, ilmoitus lähetetään ostoryhmän esimiehelle.

Ostotilauksen hyväksyjä hyväksyy ostotilauksen siirtämällä ostotilauksen seuraavaan tilaan. Hän voi myös hylätä ostotilauksen, jolloin laskun hylkäämisestä lähetetään ilmoitus ostajalle. Kun hyväksyjä hyväksyy ostotilauksen, tilasiirtymään toteutettu skripti kutsuu SAP Wrapper -moduulia COM-rajapinnan kautta ja vapauttaa sen avulla ostotilauksen SAP:issa. Ostotilaus-kohteen Release indicator -ominaisuus päivittyy seuraavan kerran, kun SAP Connector päivittää ulkoiset kohdetyypit SAP:ista.

Ostotilauksen muokkaaminen on mahdollista vielä hyväksynnän jälkeenkin. Mikäli muokkaukset ylittävät ostajan ostorajan, voi aikaisemmin hyväksytty lasku vaatia uudelleenhyväksynnän. Tätä ongelmaa ja sen ratkaisua on kuvattu tarkemmin luvussa 5.7.5 sivulla 35.

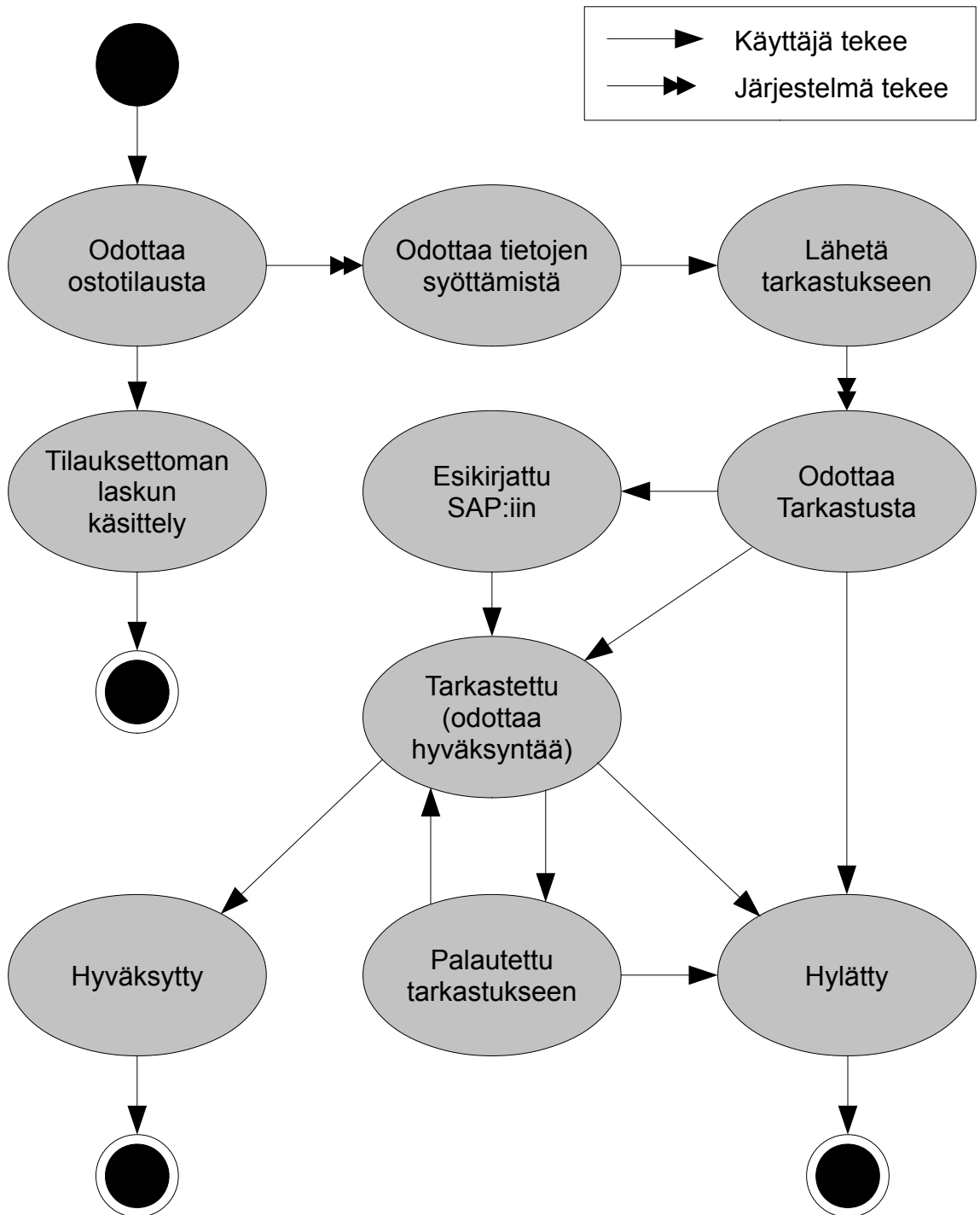
### 5.5.3 Ostotilaustulosteiden lähettäminen



Kuva 7: Ostotilaustulosteiden lähettäminen - työnkulun tilat ja tilasiirtymät

Kun ostotilaus hyväksytään SAP:issa, SAP tallentaa ostotilaustuloksen ArchiveLinkin kautta M-Filesiin. Etsi ostotilaus -tilassa ostotilaustulosteesta muodostetaan linkki sitä vastaavaan ostotilaukseen (Kuva 7). Kun ostotilaus löydetään, lähetetään siihen liittyvälle ostajalle sähköposti saapuneesta ostotilaustulosteesta. Kun ostaja lähettää M-Filesissa olevan ostotilaustulosteen sähköpostilla toimittajalle, hän voi siirtää sen Lähetetty-tilaan. Tällä tavalla hänen on helppo nähdä, mitkä ostotilaustulosteet ovat vielä lähettämättä.

## 5.5.4 Ostolaskujen hyväksyntä



**Kuva 8:** Ostolaskujen hyväksyntä - työnkulun tilat ja tilasiirtymät

Ostolaskujen hyväksyntä käynnistyy, kun M-Filesiin määritetty ulkoinen tiedostolähde noutaa skannatun laskun verkkolevyllä määritetystä kansioista. Ulkoinen tiedostolähde

muuntaa laskun PDF-muotoon ja luo laskua vastaavan kohteen M-Filesiin. Lasku asetetaan Odottaa ostotilausta -tilaan (Kuva 8).

Samalla ulkoinen tiedostolähde tunnistaa OCR-moduulin avulla laskun ensimmäisellä sivulla olevat numerot ja tallentaa ne laskun ominaisuuteen. Tähän ominaisuuteen määritelty validaatioskripti etsii sisällöstä kymmenen numeron pituisia ostotilausnumerokandidaatteja. Mikäli löytyy tasan yksi olemassaolevaa ostotilausta vastaava ostotilausnumero, lasku linkitetään ostotilaukseen.

Mikäli OCR-moduulilla ei löydetä sopivaa ostotilausnumeroa, skannaaja joko linkittää laskun ostotilaukseen käsin tai toteaa, että kyseessä on tilaukseton lasku. Jälkimmäisessä tapauksessa hän siirtää laskun Tilauksettoman laskun käsittely -tilaan.

Kun laskulle on automaattisesti tai skannaajan kirjaamana löytynyt tasan yksi ostotilausnumero, se siirtyy automaattisen tilasiirtymän avulla Odottaa tietojen syöttämistä -tilaan. Skannaaja kirjaa M-Filesin käyttöliittymän kautta lasku-kohteelle metatiedoiksi vielä laskun numeron, päivämäärän, loppusumman ja viitenumeron sekä siirtää laskun Lähetetty tarkastukseen -tilaan.

Lähetetty tarkastukseen -tilaan määritellyssä skriptissä laskulle haetaan ostaja ostoryhmästä ja hyväksyjä kustannuspaikasta tai ostoryhmästä samalla tavalla kuin ostotilaukselle. Odottaa tarkastusta -tilaan siirrytään automaattisesti tämän jälkeen. Tähän tilaan saavuttaessa lähetetään sähköpostia ostajalle. Nämä kaksi tilaa olisi voinut toteuttaa myös yhtenä tilana, mutta kahdella tilalla haluttiin varmistaa, että ostaja ja hyväksyjä löytyvät ennen kuin ostajalle lähetetään sähköpostia.

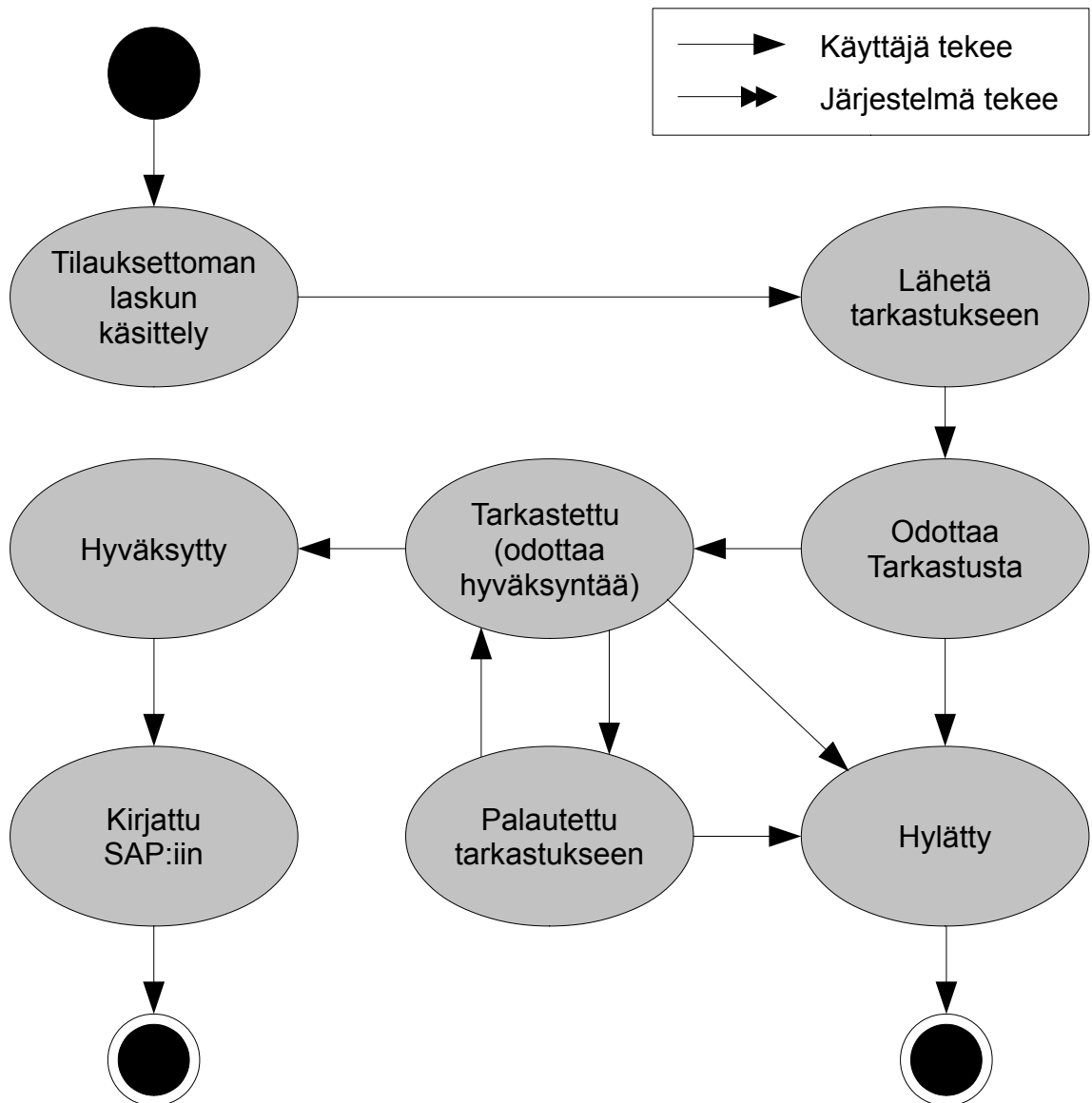
Ostaja tarkastaa laskun. Jos laskun tiedot ovat kunnossa ja laskutettavat tavarat ovat saapuneet, hän lähettää laskun hyväksyttäväksi siirtämällä kohteen Tarkastettu-tilaan. Tilaan määritetty skripti hakee ostotilauksen tiedot käyttämällä KGS Activatorin PurchaseOrderGetDetailS-funktiota, tekee tarvittavat muunnokset saamilleen ostotilauksen tiedoille ja esikirjaa uuden laskun SAP:iin DocumentPostExtS-funktion avulla. Ennen esikirjausta simuloidaan laskun lähettämistä samoilla tiedoilla. Mikäli simulointi epäonnistuu, estetään tilasiirtymä ja näytetään käyttäjälle virhedialogissa SAP:in simuloinnissa tuottama virheilmoitus.

Jos ostaja haluaa muokata laskua SAP:issa ennen sähköpostin lähettämistä hyväksyjälle, hän voi siirtää laskun Esikirjattu SAP:iin -tilaan. Simulointi ja esikirjaus toteutetaan tässä tilassa samalla tavalla kuin Tarkastettu-tilassa, mutta hyväksyjälle ei lähetetä vielä sähköpostia. Kun ostaja on tehnyt haluamansa muokkaukset SAP:issa, hän siirtää laskun Tarkastettu-tilaan. Tämä aiheuttaa sähköpostin lähettämisen hyväksyjälle. Koska lasku on jo esikirjattu, tilasiirtymä ei aiheuta uutta simulointia ja esikirjausta.

Mikäli simulointi ja esikirjaus onnistuu, lähetetään ilmoitus laskun hyväksyjälle. Hyväksyjä voi hyväksyä laskun siirtämällä sen Hyväksytty-tilaan, jolloin esikirjattu lasku tallentuu maksettavaksi SAP:issa. Tämä tehdään KGS Activatorin DocumentoPostExtS-funktion avulla käyttäen MIR4-transaktiota.

Hyväksyjä voi myös lähettää laskun uudelleentarkastettavaksi siirtämällä sen Palautettu tarkastukseen -tilaan. Tilasiirtymää tehdessä hyväksyjä voi myös kirjoittaa viestin ostajalle uudelleentarkastuksen syistä.

Jos laskua ei ole vielä hyväksytty, ostaja tai hyväksyjä voivat hylätä laskun. Jos lasku on ehditty esikirjata SAP:iin, esikirjaus poistetaan hylkäämisen yhteydessä kutsumalla KGS Activatorin SAPObjectDelete-funktiota.



Kuva 9: Tilauksettomien ostolaskujen hyväksyntä - työnkulun tilat ja tilasiirtymät

Jos laskuun ei liity ostotilausta, työnkulku muistuttaa ostotilauksellisen laskun työnkulkua, mutta laskuja ei kirjata SAP:iin missään vaiheessa (Kuva 9). Koska tietoa ostajasta ei saada ostotilauksesta, skannaaja joutuu kirjaamaan laskun metatietoihin, kenelle lasku lähetetään tarkastettavaksi. Saatuaan laskun tarkastettavaksi ostaja syöttää

laskun metatietoihin laskun tiliointitiedot ja laskun hyväksyjän. Kun hyväksyjä on hyväksynyt laskun, laskun tiedot syötetään käsin SAP:iin.

## 5.6 SAP Wrapper

### 5.6.1 Tehtävä ja tarkoitus

SAP Wrapper luotiin täydentämään valmiiden moduulien ja erityisesti KGS Activatorin puutteita. Tavoitteena oli luoda yksinkertainen tapa päästä kutsumaan mahdollisimman suoraan SAP:in rajapinnassa olevia funktioita.

KGS Activatorin käytön yhtenä ongelmana oli, että suuri osa sen rajapinnassa olevista funktioista palauttivat tietoa out-parametreissa. VBScriptin rajoitteiden vuoksi M-Filesin skripteistä ei voitu käyttää out-parametreja sisältäviä funktioita. Tärkeimmistä funktioista oli onneksi toteutettu skriptaamista varten out-parametrittomia versioita, joissa out-parametrit oli sisällytetty paluuarvorakenteeseen.

Muukin KGS Activatorin toiminnallisuus olisi ollut mahdollista saada käyttöön lisäämällä toiminnallisuutta SAP Activator-kääremoduuliin tai tilaamalla KGS:ltä halutuista funktioista out-parametrittomia versioita. Riskinä kuitenkin oli, että toiminnallisia puutteita paljastuisi enemmän ja niiden toteuttaminen olisi kallista ja aikaavievää. SAP Wrapperin rajapinta oli mahdollista suunnitella alusta asti niin, että sen kutsuminen M-Filesin skripteistä olisi mahdollisimman yksinkertaista. Oman moduulin toteuttaminen oli myös oppimisen kannalta tärkeää, kun tarkoitus on kuitenkin tulevaisuudessa toteuttaa vastaavia integraatioita ilman kolmannen osapuolen komponentteja.

SAP Wrapper on COM-komponentti, joka on toteutettu C#-kielellä .NET-ympäristöön. Kuten SAP Connector, se käyttää SAP .NET Connectoria kommunikaatioon SAP:in kanssa. Yhteysasetukset on määritelty XML-tiedostossa, joka ladataan rekisteröidyn DLL-tiedoston hakemistopolusta. Asetustiedostoon voidaan määritellä useita SAP-yhteyksiä.

### 5.6.2 Hyväksyjän löytäminen

Hyväksyjän löytäminen päätettiin toteuttaa lukemalla SAP:in sisäisiä tauluja suoraan, koska huomattiin, että tieto hyväksyjästä tarvitaan jo ennen kuin SAP Connector on päivittänyt tämän tiedon SAP:ista M-Files:iin.

SAP Wrapperiin päätettiin toteuttaa toiminnallisuus, jonka avulla on mahdollista lukea mitä tahansa SAP:in tietoa. Generisen toiminnallisuuden sijaan olisi voitu toteuttaa myös ainoastaan tämän ongelman ratkaisu, mikä olisi voinut säästää päivän työaikaa. Erikoistunut ratkaisu olisi myös siirtänyt toiminnallisuutta pois M-Filesin skripteistä. Niiden avulla monimutkaisen toiminnallisuuden toteuttaminen on huomattavasti hankalampaa kuin kehittyneillä korkean tason ohjelmointikielillä kuten

C#. Tällä olisi mahdollisesti säästetty toinen päivä työtä. Voidaan olettaa, että geneerisemmästä ratkaisusta on hyötyä tulevaisuudessa.

Taulujen luku tapahtuu SAP:in RFC\_READ\_TABLE-funktion avulla. Tietojen lukemista varten tehtiin sisäinen cursoriratkaisu, jonka avulla tiedot voi lukea rivi kerrallaan tulostaulusta. Yksinkertainen tietojen järjestäminen toteutettiin lähinnä pienimmän arvon löytämistä varten.

Jälkikäteen arvioituna on selvää, että geneerisyyteen pyrkivä ratkaisu oli tässä tapauksessa selvästi huonompi kuin erikoistunut ratkaisu. Kehitystyön aikana tällainenkin ilmiselvä asia voi olla vaikea huomata. Päätökseen vaikuttaa myös se, että kerran geneeriseksi suunnitellun ratkaisun pudottaminen erikoistuneeksi ratkaisuksi tuntuu helposti ohjelmistonkehittäjän näkökulmasta luovuttamiselta. Geneerisestä ratkaisusta voidaan tehdä erikoistunut ratkaisu myös lisäämällä kerroksia, mutta tämä taas monimutkaistaa arkkitehtuuria. Geneerisiä ratkaisuja voi siis olla järkevää välttää myös muista syistä kuin siitä ilmiselvästä syystä, että niiden toteuttaminen vaatii enemmän resursseja.

### 5.6.3 Ostotilauksen vapauttaminen

Järjestelmän tuli pystyä vapauttamaan ostotilaukset SAP:ista, kun niitä vastaavat ostotilaukset hyväksytään M-Filesissä. Tämä oli mahdollista SAP:in rajapinnan funktiolla BAPI\_PO\_RELEASE. Vapautusta varten tarvitaan myös vapautuskoodi, joka vaihtelee eri SAP-järjestelmien välillä. SAP:in tilaa muokkaavat transaktiot tulee varmistaa erillisellä BAPI\_TRANSACTION\_COMMIT-funktiokutsulla, joka tuo tehdyt muutokset näkyviin muille käyttäjille.

## 5.7 Toteutuksessa ilmenneet ennakoimattomat ongelmat

### 5.7.1 Uusien kohteiden linkittyminen SAP-dokumentteihin

Linkin muodostaminen ostotilausten ja ostotilaustulosteiden välille osoittautui ongelmalliseksi. Ongelman aiheutti se, että ostotilaustulosteet siirtyvät M-Filesiin välittömästi ostotilauksen luonnin yhteydessä, jos käyttäjän oikeudet riittivät ostotilauksen vapauttamiseen, mutta itse ostotilaukset siirtyvät vasta seuraavan viiden minuutin aikana, kun SAP Connector synkronoi tietoja SAP:ista.

Ongelma korjaantui koodimuutoksella SAPALINK-moduuliin. Koodimuutoksella moduuli saatiin toimimaan niin, että ostotilaustulosteen syntymisen yhteydessä SAP Connector saatiin synkronoimaan ostotilaukset SAP:ista, mikäli sopivaa ostotilausta ei löydy M-Filesistä.



### 5.7.2 SAP taulujen moniosaiset pääavaimet

Projektin alkuvaiheessa huomattiin, että osassa projektin kannalta tärkeitä SAP:in tauluja on moniosainen pääavain. SAP Connector kuitenkin vaati, että taulusta löytyy rivit yksilöivä avainominaisuus. Tämän vuoksi SAP Connectoriin toteutettiin mahdollisuus katentaa SAP-taulun sarakkeiden arvoja. Näin yksilöivä ominaisuus voitiin muodostaa yhdistämällä usean ominaisuuden arvoja myös tauluille, joiden yksilöivä pääavain koostui useammasta ominaisuudesta. Myös moniosaisen vierasavainten muodostaminen onnistuu vastaavasti. Samalla optimoitiin SAP-taulujen lukemisen sisäistä toteutusta niin, että vaikka sama ominaisuus toistuisi SAP Connectorille annetussa kyselyssä useita kertoja, SAP:ilta ominaisuus pyydetään vain kerran.

### 5.7.3 Ostotilaustulosteiden siirtyminen M-Filesiin ennen hyväksyntää

Ostotilauksen hyväksyjän olisi tarvinnut nähdä ostotilaustuloste voidakseen hyväksyä ostotilauksen. SAP ei kuitenkaan anna tulostaa tai tallentaa ostotilaustulostetta ennen hyväksyntää, ettei hyväksymätöntä ostotilausta lähetettäisi toimittajalle. Tämän vuoksi ostotilaustulostetta ei voitu tarkastella M-Filesissä ennen ostotilauksen hyväksymistä. Ongelmaa vaikeutti se, että SAP:in käyttöliittymästä on mahdollista tarkastella ostotilaustulostetta ennen hyväksyntää. Näin ollen asiakas oletti, että tämä olisi mahdollista myös M-Filesissa ennen hyväksyntää.

Hyväksyntää varten suunniteltiin M-Filesiin erityinen ostotilaus-näkymä, jossa hyväksymistä varten vaadittavat tiedot olisivat helposti tarkasteltavissa, ja jonka perusteella hyväksyjä voisi hyväksyä tilauksen. Tällaiseen näkymään olisi voitu päätyä muutenkin, koska tähän näkymään voitiin tuoda myös sellaisia tietoja, joita ei nähty SAP:in käyttöliittymän ostotilaustulosteissa.

### 5.7.4 Ostotilauksen tietojen syöttäminen laskulle

Yllättävän ja hankalan ongelman aiheutti se, että laskun luominen SAP:in käyttöliittymästä oli huomattavasti helpompaa kuin ulkoisen rajapinnan kautta. Käyttöliittymä haki automaattisesti suuren osan laskun vaatimista tiedoista suoraan ostotilauksesta, mutta ulkoisen rajapinnan funktio vaati näiden tietojen syöttämistä manuaalisesti. Asiakas luonnollisesti oletti, että laskun luominen onnistuisi M-Filesista yhtä helposti kuin SAP:in käyttöliittymän kautta.

Tähän ratkaisuksi tutkittiin jonkin aikaa mahdollisuutta käyttää käyttöliittymän alirutiineja ulkoisen rajapinnan kautta: SAP tarjoaa monipuolisia mahdollisuuksia käyttöliittymän rutiinien koodin ajonaikaiseen tarkasteluun. Yrityksestä päätettiin kuitenkin luopua, koska tätä käyttöliittymän alirutiinien koodia todettiin olevan liian paljon. Tuntemattoman järjestelmän koodin ajonaikainen tarkastelu on myös erittäin työlästä ja hankalaa.

Ainoaksi vaihtoehdoksi jäi yrittää toteuttaa mahdollisimman pitkälle SAP:in alirutiinin toiminnallisuutta muistuttava toteutus SAP:in ulkopuolelle. Ostotilausten tietojen siirto ostolaskulle oli kuitenkin riittävän yksinkertainen ongelma, että riittävästi SAP:in toiminnallisuutta vastaava toteutus saatiin toteutettua kohtuullisessa ajassa. KGS Activatorin funktio PurchaseOrderGetDetails auttoi ostotilauksen tietojen saamisessa. Tietojen syöttäminen laskulle toteutettiin VBScriptillä M-Filesin tapahtumankäsittelijässä.

### 5.7.5 Ostotilausten uudelleenhyväksyminen

Selvisi, että asiakkaan SAP-järjestelmässä hyväksytyjä ostotilauksia on mahdollista muokata vielä hyväksymisen jälkeen. Joissain tapauksissa tällaiset muokkaukset vaativat ostotilauksen uudelleen vapauttamisen SAP:issa. Tämän tilanteen varma tunnistaminen M-Filesista osoittautui ongelmalliseksi.

Kun ostotilaus luodaan SAP:ssa, ostotilaustuloste tallennetaan automaattisesti ArchiveLink-rajapinnan ja KGS Content Server -moduulin kautta M-Filesiin. Mikäli luotu ostotilaus ei kuitenkaan läpäise sille SAP:ssa määriteltyjä ehtoja, ostotilaus menee estotilaan (blocked). Tällöin ostotilaustulostetta ei synny, eikä SAP muutenkaan kerro ulospäin millään tavoin ostotilauksen syntymisestä. Tätä varten otettiin käyttöön ExternalObjectTypeRefresher-moduuli, joka käynnistää ulkoisten kohdetyyppien päivittämisen SAP:ista M-Filesiin. Käyttämällä Windows-käyttöjärjestelmän ajastettuja tehtäviä (scheduled tasks), kohdetyyppien päivittyminen saatiin käynnistymään minuutin välein. Tällä tavoin saatiin estotilassa olevat ostotilaukset pienellä viiveellä M-Filesiin vapautusta varten.

Ostotilausten tila voidaan M-Files:ssä päätellä vapautusindikaattori (release indicator) -kentästä. Kun ostotilaus hyväksytään M-Filesissä se vapautetaan SAP:issa. Indikaattori päivittyy kuitenkin vasta pienellä viiveellä. Muuten tämä ei olisi ongelma, mutta joissain tapauksissa ostotilausten muuttaminen hyväksymisen jälkeen voi aiheuttaa ostotilauksen siirtymisen estotilaan uudelleen. Tällöin ostotilaus pitää hyväksyä uudelleen M-Filesissä, että ostotilaus voidaan vapauttaa. Mikäli ostotilaus siirtyy estotilaan nopeasti hyväksynnän jälkeen, indikaattori ei ehdi päivittyä M-Filesiin vapauttamisen jälkeen. Tällöin estotilaan siirtyminen jää huomaamatta M-Filesissa.

Tähän ongelmaan ei löytynyt täydellistä ratkaisua. Siinä päädyttiinkin osittaiseen ratkaisuun, jossa uudelleenhyväksyminen käynnistetään tunnin päästä hyväksymisestä, mikäli indikaattori ei ole silloinkaan vielä päivittynyt. Esimerkiksi yhteyden katkeaminen saattaa tässä ratkaisussa aiheuttaa uudelleenhyväksymisprosessin käynnistymisen turhaan. Tätä tilannetta voi kuitenkin pitää hyvin harvinaisena, ja huonoimmassakin tapauksessa seurauksena on vain, että hyväksyjä joutuu hyväksymään ostotilauksen uudelleen.

### 5.7.6 Varastoon tilaaminen

Selvisi, että luotaessa laskua varastotilauksille SAP:in käyttöliittymän alirutiinit hakevat laskun tietoja ostotilauksen lisäksi myös materiaalidokumentista, joka syntyy, kun tavara merkitään vastaanotetuksi varastoon. Asiakas luonnollisesti oletti, että sama toiminnallisuus toteutuisi myös, kun laskua luodaan M-Filesin kautta. Koska SAP ei hae laskun ohjelmallisessa luonnissa mitään laskun tietoja automaattisesti, jouduttiin luomaan toiminnallisuus myös näiden tietojen kokoamiseen. KGS Activator -moduulin PurchaseOrderGetDetailS-funktiolla nämä tiedot saatiin M-Filesin skriptiin, jossa ne saatiin kohtuullisella työmäärällä syötettyä ostolaskulle.

### 5.7.7 Laskun viitenumeron syöttäminen laskulle

Hieman yllättäen laskun viitenumeron syöttäminen laskulle ei onnistunutkaan KGS Activatorin DocumentPostExtS-funktiolla. Selvisi, että pohjoismaissa yleisesti käytettyjä laskujen viitenumeroita ei käytetä juuri missään muualla maailmassa. Suomalaisen SAP-palveluntarjoajan järjestelmässä tämä näkyi laskun luontilomakkeessa ylimääräisenä viitenumerokenttänä, joka puuttui kokonaan ensimmäisen kehitysympäristön SAP-versiosta.

Koska tämän viitenumerotiedon siirtäminen automaattisesti M-Filesistä SAP:iin oli asiakkaalle tärkeää ja viitenumeron syöttäminen ei onnistunut KGS Activatorilla, ainoa vaihtoehto olisi ollut luopua KGS Activatorista laskun luonnissa ja toteuttaa sama itse omana moduulina. Tämä olisi aiheuttanut paljon lisätyötä projektiin, ja sen vuoksi ominaisuus päädyttiin ostamaan KGS:ltä.

### 5.7.8 Laskun tiedoissa olevien virheiden paljastaminen ennen lähettämistä

Määrittelyvaiheessa oletettiin, että laskun esikirjaaminen paljastaisi laskun tiedoissa olevat ongelmat, jotka myöhemmin tulisivat estämään lähettämisen. SAP ei kuitenkaan tee juurikaan laskun tietojen tarkistuksia esikirjaamisen yhteydessä. Käyttäjän näkökulmasta tämä aiheutti sen, että väärin syötettyjen laskujen virheilmoitukset näkyivät hyväksyjälle. Asiakas halusi, että nämä virheet näkyvät jo hyväksymiskierron aikaisemmassa vaiheessa ostolaskun tarkastajalle, joka voisi sen jälkeen tehdä tarvittavat korjaukset tilaukseen tai laskuun.

SAP tarjoaa käyttäjille mahdollisuuden simuloida laskun lähettämistä. Simulointi on myös mahdollista ulkoisen rajapinnan kautta, mutta harmillisesti esikirjattujen laskujen lähettämisen simulointi ei ole mahdollista, vaan kaikki laskun tiedot pitää syöttää simulointifunktion parametreihin. Nämä parametrit ovat samoja kuin laskun luomiseen tarkoitetun funktion parametrit. Laskun luomiseen oli kuitenkin käytetty projektissa KGS Activatoria, johon laskun tiedot syötettiin eri tavalla. Laskun tiedot olisi siis pitänyt konvertoida KGS Activatorin vaatimasta muodosta SAP:in simulointifunktion

vaatimaan muotoon. Konvertoinnin toteuttamisen jälkeenkin olisi ollut epävarmuutta siitä, vastaako simuloitu lasku varmasti esikirjattua laskua.

Tästä johtuen laskun simulointi päädyttiin ostamaan KGS:ltä. KGS Activatorin uusi versio mahdollisti laskun simuloinnin myös esikirjaamiseen käytetyllä DocumentPostExtS-funktiolla simulointiparametrin avulla. Simuloinnin onnistuessa voitiin lasku esikirjata kutsumalla funktiota uudelleen samoilla tiedoilla, mutta ilman simulointiparametria.

### **5.7.9 Hylättyjen laskujen poistaminen**

Asiakas halusi, että M-Filesissä hylätyt laskut poistettaisiin SAP:ista varmuuden vuoksi. Mikäli hylättyjä laskuja ei olisi poistettu SAP:ista, riskinä olisi ollut, että joku työntekijä olisi käynyt SAP:issa lähettämässä M-Filesissä hylätyn laskun. SAP:in rajapinnasta löytyi useita tähän periaatteessa soveltuvia funktioita, mutta näitä ei saatu yrityksistä huolimatta toimimaan. KGS Activatorin dokumentaation mukaan SapObjectDelete-funktion avulla piti olla mahdollista poistaa ostolaskuja [14]. Funktio ei kuitenkaan toiminut odotetusti. Aktiivisen KGS:n kanssa tapahtuneen kommunikaation ansiosta tämä ominaisuus saatiin korjattua korjauspäivitykseen, jonka oli alunperin tarkoitus korjata vain laskun viitenumeron ongelma ja laskun simuloinnin ongelma.

## **5.8 Toteuttamatta jätetyt ominaisuudet**

### **5.8.1 Valuuttakurssin esittäminen M-Filesissa**

Huomattiin, että SAP:in rajapinnan RFC\_READ\_TABLE-funktio katkaisee yli yhdeksän merkkiä pitkät valuuttakurssit vasemmalta. Ratkaisuun vaadittava aika arvioitiin tässä projektissa ratkaisun hyötyä suuremmaksi. Tämän vuoksi valuuttakurssi päätettiin jättää esittämättä M-Filesissa integraation ensimmäisessä versiossa. On todennäköistä, että valuuttakurssien esittämisen ongelmaan löytyy ratkaisu tutkimalla RFC\_READ\_TABLE-funktion parametreja ja sitä, missä muodoissa SAP:in taulujen tietoa on mahdollista lukea.

## 6 TOTEUTETUN RATKAISUN ARVIOINTI

### 6.1 Projektin onnistumisen arviointi

Asiakas oli lopputulokseen monilta osin tyytyväinen. Erityisesti joidenkin virheilmoitusten sisällöt ylittivät asiakkaan odotukset. Projekti kuitenkin ylitti sille sisäisesti annetun alkuperäisen työmääräarvion. Jonkin verran ylittyi myös asiakkaan toivoma projektin kesto.

Viivästymiset ja jopa projektin keskeyttämiseen johtavat ongelmat ovat tyypillisiä tietojärjestelmäintegraatioprojekteille. Projektia voidaan pitää onnistuneena, koska se pystyttiin viemään loppuun kohtuullisella aikataulun ylityksellä sidosryhmien pysyessä kohtuullisen tyytyväisinä koko projektin ajan.

SAP:iin ja SAP-integraatioihin liittyvä osaaminen lisääntyi merkittävästi projektin aikana. Hankitusta tietämyksestä tulee olemaan merkittävää hyötyä tulevissa SAP-integraatioprojekteissa. Yksi tällainen projekti oli jo alustavasti suunnitella projektin loppuvaiheessa.

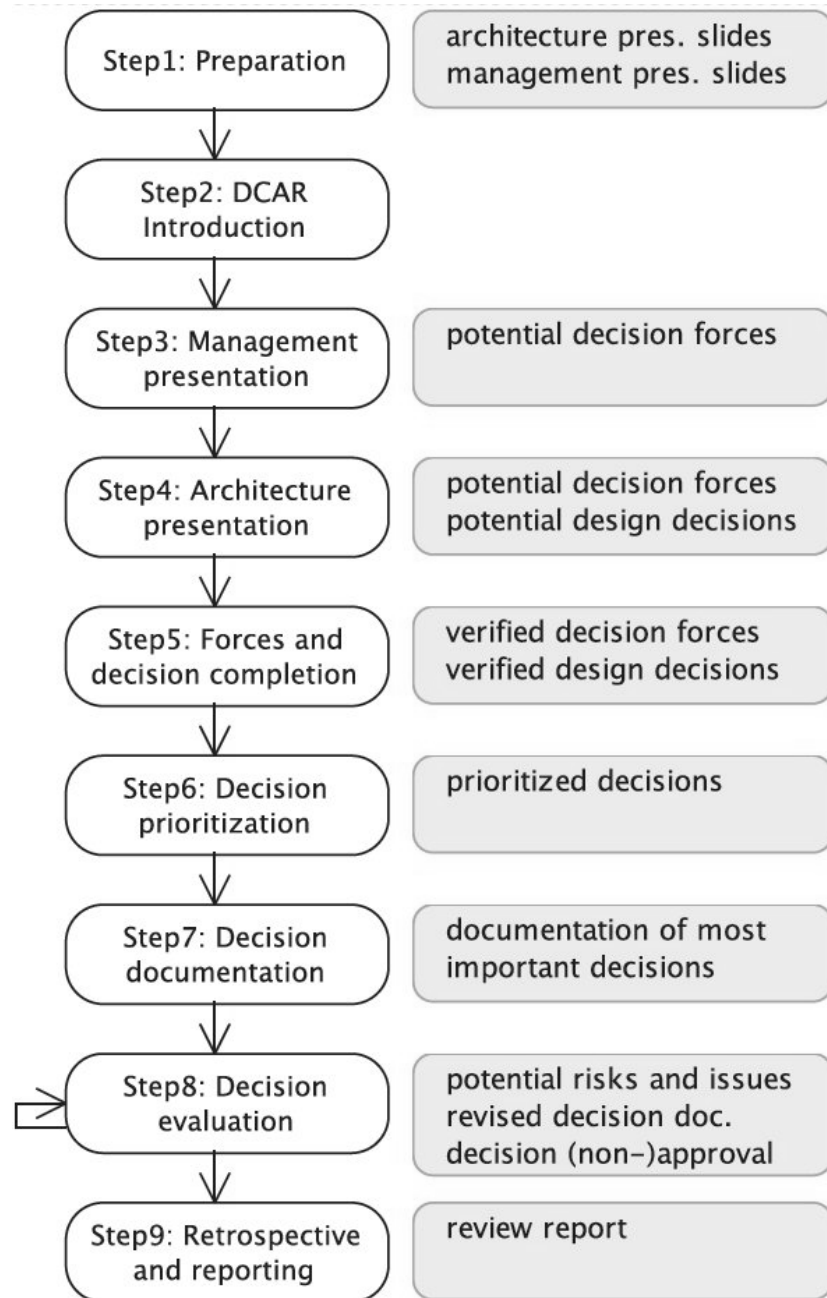
### 6.2 Arkkitehtuurin arviointi DCAR-menetelmää soveltaen

#### 6.2.1 Arviointimenetelmän kuvaus

DCAR eli päätöskeskeinen arkkitehtuurikatselmus (Decision-centric architecture review) on ohjelmistoarkkitehtuurien arviointiin tarkoitettu menetelmä, jonka lähtökohtana on arkkitehtuuristen päätösten analysointi. Menetelmä pyrkii tunnistamaan järjestelmän tärkeimmät arkkitehtuuriset päätökset ja päätöksiin vaikuttavat voimat (forces). Tärkeimpien päätösten löytämiseen menetelmässä käytetään pisteyttämistä, jota tekemässä tulisi ideaalisesti olla useita osallistujia. [15]

DCAR:ia on kehitetty ratkaisemaan ketterän ohjelmistokehityksen ja ohjelmistojen arkkitehtuurisen suunnittelun välistä ristiriitaa. Menetelmässä tunnistetaan, että raskaita ja aikaa vieviä arviointimenetelmiä ei haluta ottaa käyttöön ketterässä ohjelmistokehityksessä. Tämän vuoksi DCAR ei pyri muodostamaan kokonaiskuvaava arkkitehtuurista, vaan pyrkii nopeasti tunnistamaan ja ratkaisemaan optimaalisesti järjestelmän arkkitehtuuriset päätökset. [15]

Koska DCAR:issa ei ole välttämätöntä muodostaa kokonaiskuvaava arkkitehtuurista, sitä on myös helppo käyttää inkrementaalisesti, mikä osaltaan parantaa soveltuvuutta ketterään ohjelmistokehitykseen. Inkrementaalisessa käytössä DCAR:illa voidaan arvioida osaa koko järjestelmän arkkitehtuurillisista päätöksistä kerrallaan. [15] Tämä on hyödyllistä, koska erityisesti projektin alkuvaiheessa kaikkia arkkitehtuurisia päätöksiä ja niihin vaikuttavia voimia on hankala tunnistaa.



**Kuva 10:** DCAR-menetelmän vaiheet [17]

DCAR on määritelty periaatteina ja raameina arkkitehtuurikatselmuksille. Ratkaisu konkretisoi teoriaa ja tuo sitä lähemmäs potentiaalisia käyttäjiä. Yleisenä arviointimenetelmänä DCAR ei anna eikä voikaan antaa ohjeita siitä, millaiset arkkitehtuuriset ratkaisut ovat hyviä tai huonoja, vaan se pyrkii hyödyntämään olemassaolevan arkkitehtuurisen osaamisen paremmin ja tekemään päätöksentekoa näkyvämmäksi.

Arkkitehtuurikatselmukseen tulee osallistua vähintään arvioitavan järjestelmän arkkitehti ja katselmoijia, joilla on hyvä olla kokemusta arkkitehtuurisuunnittelusta.

Lisäksi suositellaan, että katselmukseen osallistuisivat asiantuntijat, joilla on arvioitavan järjestelmän kannalta keskeistä osaamista. Arkkitehtuurikatselmukseen voidaan ottaa myös muita projektin sidosryhmien edustajia. [16]

DCAR-arkkitehtuurikatselmus koostuu yhdeksästä vaiheesta (Kuva 10). Ensimmäinen vaihe on valmistautumista katselmukseen, eikä tapahdu itse katselmuksen aikana. Arkkitehtuurikatselmuksen alussa varmistetaan, että DCAR-menetelmä ja sen tavoitteet ovat tuttuja kaikille osallistujille. Tätä seuraavat järjestelmän esittelyt kahdesta eri näkökulmasta. Ensimmäisenä on johdon esitys, jonka tarkoituksena on esitellä järjestelmää liiketoiminnan näkökulmasta. Tässä osassa voidaan esitellä mm. tuote ja toimiala, liiketoimintaympäristö, asiakkaiden tarpeet, vaatimukset ja odotukset, rahoitustilanne, tunnetut tekniset rajoitteet sekä liiketoiminta- ja laatuavoitteet. [17]

Johdon esitystä seuraa järjestelmän arkkitehtuurinen esittely. Järjestelmän arkkitehti esittelee järjestelmän sekä siihen liittyvän korkean tason arkkitehtuurin, ratkaisuun johtaneet päätökset perusteluineen, käytetyt suunnittelumallit, käytetyt valmiskomponentit, odotettavissa olevat muutokset arkkitehtuuriin sekä arkkitehtuuriset ongelmat ja riskit. [17]

Esittelyjä seuraa katselmuksen yhteistoiminnallinen vaihe. Ensin tarkistetaan ja tarkennetaan esitysten aikana esille tulleita päätöksiä ja niihin vaikuttavia voimia. Päätöksistä ja niiden välisistä riippuvuuksista voidaan tarvittaessa tehdä päätössuhdekaavio (decision relationship diagram). Kun päätökset ja niihin vaikuttaneet voimat on listattu, päätökset priorisoidaan DCAR:in määrittämällä pisteytysmenettelyllä. Pisteiden laskennan jälkeen keskustellaan siitä, millaiset syyt eri osallistujilla johtivat heidän valintoihinsa. Vain korkeimmat pisteet saaneet päätökset analysoidaan. [17]

Tärkeimmiksi priorisoidut päätökset dokumentoidaan DCAR:in määrittelemille lomakkeille. Lomakkeella kuvataan tehty ratkaisu, ratkaistu ongelma, päätökselle vaihtoehtoiset ratkaisut sekä päätöstä puoltavat ja sitä vastustavat voimat. Arviointivaiheessa jokaista dokumentoitua päätöstä käsitellään vielä yhdessä 10-15 minuuttia. Osallistujat täydentävät päätöksen dokumentaatiota ja keskustelevat siitä puoltaako päätöstä riittävästi voimia suhteessa päätöstä vastustaviin voimiin. Viimeisessä vaiheessa arkkitehtuurikatselmuksessa tehdyistä löydöksistä tehdään raportti, johon kirjataan tehdyt päätökset, hylätyt vaihtoehdot sekä perustelut valitun ratkaisun puolesta ja sitä vastaan. [17]

### 6.2.2 Arviointimenetelmän soveltaminen

Tämän työn puitteissa ei ollut mahdollista järjestää useamman osallistujan arkkitehtuurikatselmusta. Tämän vuoksi arviointi on tehty yksin, mutta pyrkien kuitenkin toteuttamaan DCAR-menetelmää vaiheittain samalla tavalla, kuin sitä olisi toteutettu, jos arkkitehtuurikatselmus olisi järjestetty. On selvää, ettei tällä tavalla toteutetulla arvionnilla juuri tavoiteta kasvanutta arkkitehtuurista tietämystä, mutta

toisaalta tullaan kuitenkin dokumentoineeksi tärkeimpiä päätöksiä ja niihin vaikuttaneita voimia. Arkkitehtuurikatselmus tulee myös menetelmänä tutuksi, jolloin sen hyödyntäminen tulevaisuudessa on helpompaa.

Vaikka arkkitehtuurin arviointi on toteutettu vasta projektin jälkeen, päätöksiä on pyritty arvioimaan antamatta toteutetun ratkaisun vaikuttaa arvioinnin tuloksiin. Projektin aikana hankittu tietämys vaikuttaa joka tapauksessa arvioinnin tuloksiin. Tämän vuoksi arvionti ei anna samoja tuloksia, kuin olisi saatu, jos arviointi olisi toteutettu ennen projektia. Käytännössä jälkikäteen tehty arvionti pyrkii vastaamaan siihen, miten projekti kannattaisi toteuttaa, jos se jouduttaisiin toteuttamaan kokonaan uudelleen.

### **6.2.3 Arvioinnin tulokset**

Liiketoimintanäkökulmasta projektin keskeinen tuote on M-Files ja toimialana on ostoprosessit, laskutus ja laskujen arkistointi. Projekti on tuntiperustaisesti laskutettu asiakasprojekti, jonka aikataulu on sidottu SAP-järjestelmän käyttöönoton aikatauluun. Tämän vuoksi projekti ei saa kestää yli kahta kuukautta.

Aikaisemmin M-Files Oy:ssä on toteutettu skannattujen tilauksettomien ostolaskujen tallentaminen SAP:iin käyttäen KGS:ltä ostettuja ohjelmistokomponentteja sekä SAP Connector, joka mahdollista tietojen tuomisen M-Filesiin SAP:in sisäisistä tietokantatauluista.

Keskeisenä liiketoimintatavoitteena on toteuttaa asiakkaan näkökulmasta onnistunut projekti ja kerätä tietämystä ja kokemusta SAP:ista ja siihen liittyvistä integraatioista. Laatuvaatimuksista tärkein on se, että ostotilaukset ja laskut eivät katoa järjestelmästä. Toiseksi tärkeintä on, että järjestelmää on helppo käyttää.

Järjestelmän arkkitehtuuri on kuvattu luvussa 5.4.1. Arkkitehtuurista tunnistettiin taulukossa 1 listatut päätökset.



**Taulukko 1:** DCAR-arvioinnissa tunnistetut päätökset

Päätös	Pisteet
1. KGS Content Serveriä käytetään dokumenttien siirtoon M-Filesin ja SAP:in välillä.	0
2. KGS Activatoria käytetään laskujen tallentamiseen M-Filesista SAP:iin.	30
3. SAP Connectoria käytetään tiedon tuomiseen SAP:ista M-Filesiin.	20
4. SAP Wrapper toteutaan ostotilausten vapauttamiseksi SAP:issa.	0
5. Järjestelmän kohteiden eri vaiheita kuvataan M-Filesin työkuluilla ja tilasiirtymillä.	0
6. Tilasiirtymistä seuraava toiminnallisuus toteutetaan M-Filesin tilasiirtymien skripteissä.	50

Pisteytyksessä tärkeimmäksi päätökseksi osoittautui 6. päätös. Toiseksi tärkein päätös on 2. päätös. Kolmanneksi tärkein päätös on 3. päätös. Näistä päätöksistä toteutetut päätöskuvaukset ovat liitteessä 1. Muut päätökset jäivät pienemmälle prioriteetille, ja ne on sen vuoksi jätetty pois analyysistä. Päätös 6 riippuu päätöksestä 5, mutta muuten päätösten välillä ei vallitse riippuvuussuhteita, eikä suhdediagrammista ole tästä johtuen hyötyä.

Päätösten 2 ja 3 todettiin olevan hyviä. Päätöksen 6 kohdalla vaihtoehtoista ratkaisua tulee harkita. Kehitykseen käytetyn ajan säästäminen koettiin eniten päätöksiin vaikuttavaksi voimaksi.

### 6.3 Jatkokehitysajatuksia

Automaattinen ostotilausnumeron tunnistaminen skannatulta laskulta osoittautui alkuperäistä arviota helpommaksi. Vastaavasti SAP:iin integroituminen osoittautui vielä huomattavasti alkuperäistä arviota hankalammaksi. Toiminnallisuuden toteuttaminen SAP:in sisäisten mekanismien avulla olisi voinut tarjota yksinkertaisempia ja helpompia ratkaisuja projektin ongelmiin. Ilman soveltuvaa SAP-koulutusta on tosin vaikea arvioida, kuinka paljon uusia ongelmia tämä ratkaisu saattaisi aiheuttaa.

Yhtenä ongelmana integraatiossa on, että monissa teknisissä ongelmatilanteissa ollaan hyvin riippuvaisia kolmannen osapuolen moduulien valmistajan teknisestä tuesta. Riippuvuus kolmannen osapuolen moduuleista on siis myös riskitekijä integraatiolle, jos tuen jatkumiseen ei voida luottaa. Tämän lisäksi riippuvuuden poistamiseen on myös taloudellisia syitä. Riippumattomuus kolmannen osapuolen moduuleista mahdollistaisi myös integraation itsenäisen kehittämisen. Kolmannen osapuolen moduulien tarjoaman toiminnallisuuden toteuttamisen työmäärää ja toteutukseen vaadittavaa osaamista on

vaikea arvioida. Lisätyötä ja riskejä aiheuttaa myös toteutettujen projektien päivittäminen käyttämään korvaavia komponentteja.

Koska projekti toteutettiin yhdelle asiakkaalle, integraatiota ei ole tuotteistettu. Tästä johtuen integraation asentaminen on varsin työlästä: jokainen moduuli joudutaan asentamaan ja konfiguroimaan erikseen. Mikäli asennuksia jouduttaisiin tekemään usein, kannattaisi integraation asentaminen tehdä helpommaksi ja nopeammaksi. Tuotteistamisen lopullisena tavoitteena voidaan nähdä koko integraation asentuminen yhdestä asennustiedostosta ja kaikkien konfiguraatioiden keskitetty hallinta. Integraation monimutkaisuudesta ja räätälöityjen ratkaisujen suuresta määrästä johtuen on kuitenkin epätodennäköistä, että tuotteistamista tullaan koskaan viemään tälle tasolle.

Jokaista SAP:in tauluista SAP Connectorin avulla tuotavaa saraketta varten M-Filesiin joudutaan luomaan tietoa vastaava ominaisuusmäärittely. Mikäli näitä tuotavia tietoja on erityisen paljon, käsin tehtävä ominaisuusmäärittelyjen luominen voi olla työlästä. SAP Connectoria voitaisiin kehittää siten, että valittaessa tuotavia tietoja näitä vastaavat ominaisuusmäärittelyt luotaisiin automaattisesti.

SAP:in tietojen päivittäminen ei onnistu SAP Connectorilla, koska SAP:in taulujen suora päivittäminen on kiellettyä. Päivittäminen rajapintafunktioiden avulla taas vaatisi SAP Connectorilta tarkkaa ymmärrystä siitä, minkä rajapintafunktion avulla mitäkin arvoa voi päivittää. Tämän vuoksi SAP:ista tuodut tiedot ovat M-Filesissa vain luku-tyyppiä. Jotta muutoksia voitaisiin viedä M-Filesista SAP:iin, SAP Connectoriin tulisi ohjelmoida erikseen kullekin sarakkeelle, miten sen tietoja voidaan päivittää SAP:iin. Päivitystavan määrittelemisessä koko SAP:in tietosisällölle on valtavasti työtä. Järkevä lähestymistapa onkin lisätä tuettuja sarakkeita pikkuhiljaa asiakkaiden tarpeiden mukaan.

SAP Connectorin toiminnallisuutta voitaisiin laajentaa myös niin, että käyttäjä voisi luoda uutta tietoa SAP:iin luomalla uusia kohteita M-Filesiin. On kuitenkin todennäköistä, että SAP:in ja M-Filesin tiedonkäsittelyllisten paradigmojen erot aiheuttaisivat ongelmia tiedon lisäämisessä. Esimerkiksi ostotilauksen luominen SAP:issa synnyttää ostotilaus-taulun rivin lisäksi rivejä mm. ostotilausrivi-tauluun, tiliöinti-tauluun ja toimitusaika-tauluun. M-Filesissa jokainen näistä riveistä joudutaan kuvaamaan omana kohteenaan. On selvää, että useiden toisiinsa liittyvien kohteiden tietojen syöttämisestä on hankala toteuttaa hyvää käyttäjäkokemusta käyttäjälle. Käytännössä tämä vaatisi erityistä käyttöliittymän räätälöintiä M-Filesissa olevan UI Extensibility Frameworkin avulla tai merkittäviä lisäyksiä M-Filesin toiminnallisuuteen.

Joissain tilanteissa SAP:in tauluista SAP Connectorin avulla tuodun tiedon esitystapaa pitää muuttaa käyttäjälle sopivaksi. Tällöin joudutaan luomaan piilotettu ominaisuusmäärittely, jonka perusteella käyttäjälle näytettävän ominaisuusmäärittelyn arvo lasketaan käyttämällä skriptaamista. SAP Connectoria voitaisiin kehittää siten, että ainakin yksinkertaiset muunnokset voitaisiin tehdä SAP Connectorissa. Tämä voisi

vaatia tosin, että SAP Connectoriin sisällytettäisiin enemmän ymmärrystä siitä, mitä tietoja kulloinkin ollaan lukemassa. On myös mahdollista, että erikoistuneemmat SAP:in rajapintafunktiot tarjoavat tietoja paremmassa muodossa kuin RFC\_READ\_TABLE. Tietojen lukeminen erikoistuneiden rajapintafunktioiden avulla voisi joissain tilanteissa poistaa kokonaan tarpeen muunnosten tekemiseltä.

## 7 YHTEENVETO

Tässä työssä käsiteltiin M-Files Oy:n työntekijänä tehtyä ohjelmistoprojektia, jossa toteuttiin osa SAP:in ostoprosessista M-Filesissa.

Järjestelmän toteutuksen pohjana käytettiin aiemmin M-Files Oy:ssä toteutettua demonstraatiojärjestelmää. Ostotilausten ja ostolaskujen tarkastus- ja hyväksyntäkierto toteutettiin M-Filesin työnkulkujen avulla. Tilasiirtymien skripteissä toteutettiin mm. ostotilausten vapauttaminen, ostolaskun esikirjaaminen, esikirjatun ostolaskun hyväksyminen ja esikirjatun ostolaskun poistaminen.

Yksi keskeinen suunnittelupäätös oli, että työssä hyödynnetään mahdollisuuksien mukaan valmiita komponentteja. M-Filesin ja SAP:in väliseen kommunikaatioon käytetään järjestelmässä neljää komponenttia: KGS Content Serveriä, KGS Activatoria, SAP Connectoria sekä järjestelmää varten toteutettua SAP Wrapperia. KGS Content Serveriä varten täytyy konfiguroida ArchiveLink-yhteys SAP:iin. KGS:n komponentit vaativat lisäksi mukana toimitettavien funktiomodulien siirtämistä SAP:iin. SAP Connector ja SAP Wrapper käyttävät kommunikaatioon SAP .NET Connectoria.

Järjestelmä toteutti sille asetetut tavoitteet. Se mahdollistaa ostoprosessin sujuvan toiminnan. M-Filesin työnkulut määrittävät, missä vaiheessa tarkastus- ja hyväksyntäkiertoa ostotilaukset ja -laskut ovat. Järjestelmä muistuttaa eri vaiheissa automaattisesti ostajaa ja hyväksyjää sähköpostitse. Skannatut laskut siirtyvät automaattisesti M-Filesiin, ja niistä etsitään ostotilausnumero OCR-moduulin avulla. M-Files toimii myös arkistona käsitellyille laskuille.

Projektin tuloksena syntynyttä järjestelmää arvioitiin DCAR-menetelmää yhdelle arvioijalle soveltaen. Tehdyn arvioinnin mukaan tulisi harkita tilasiirtymistä seuraavan toiminnallisuuden toteuttamista muualla kuin tilasiirtymien skripteissä, esimerkiksi erillisessä .NET-ympäristöön toteutetussa komponentissa. Muut arvioidut päätökset olivat tehdyn arvioinnin mukaan hyviä. Järjestelmä ja projektin aikana hankittu kasvanut tietämys tarjoavat hyvän pohjan jatkokehitykselle.

## LÄHTEET

- [1] M-Files Oy. M-Files 9.0.3372.6 Documentation. [Viitattu 6.4.2013]. Saatavissa: Toimitetaan M-Filesin mukana.
- [2] M-Files Oy. M-Files 9.0.3372.6 API Documentation. [Viitattu 6.4.2013]. Saatavissa: Toimitetaan M-Filesin mukana.
- [3] M-Files Oy. M-Files UI Extensibility Framework Documentation. [WWW]. [Viitattu 6.4.2013]. Saatavissa: [http://www.m-files.com/UI\\_Extensibility\\_Framework/webframe.html](http://www.m-files.com/UI_Extensibility_Framework/webframe.html).
- [4] Microsoft. VBScript. [WWW]. [Viitattu 6.4.2013]. Saatavissa: <http://msdn.microsoft.com/en-us/library/t0aew7h6.aspx>.
- [5] SAP AG. SAP .NET Connector Architecture. [WWW]. [Viitattu 6.4.2013]. Saatavissa: [http://help.sap.com/saphelp\\_crm700\\_ehp02/helpdata/EN/4a/097ee1449836eae10000000a421937/content.htm](http://help.sap.com/saphelp_crm700_ehp02/helpdata/EN/4a/097ee1449836eae10000000a421937/content.htm).
- [6] SAP AG. SAP .Net Connector Programmer's Reference. [WWW]. [Viitattu 6.4.2013]. Saatavissa: [http://help.sap.com/saphelp\\_crm700\\_ehp02/helpdata/EN/4a/09830c43f4088ce10000000a421937/content.htm](http://help.sap.com/saphelp_crm700_ehp02/helpdata/EN/4a/09830c43f4088ce10000000a421937/content.htm).
- [7] SAP AG. SAP History: 1972-1981: the early years. [WWW]. [Viitattu 6.4.2013]. Saatavissa: <http://www.sap.com/corporate-en/our-company/history/1972-1981.epx>.
- [8] Adshhead, A. SAP: the climb gets easier. Computer Weekly (2002)12/12/2002, s. 26.
- [9] SAP AG. Content Repositories. [WWW]. [Viitattu 6.4.2013]. Saatavissa: [http://help.sap.com/saphelp\\_nw70/helpdata/en/ed/cefa3882cd8229e10000000a11402f/content.htm](http://help.sap.com/saphelp_nw70/helpdata/en/ed/cefa3882cd8229e10000000a11402f/content.htm).

- [10] SAP AG. The RFC API. [WWW]. [Viitattu 6.4.2013]. Saatavissa: [http://help.sap.com/saphelp\\_46c/helpdata/en/22/04280f488911d189490000e829fbbd/frameset.htm](http://help.sap.com/saphelp_46c/helpdata/en/22/04280f488911d189490000e829fbbd/frameset.htm).
- [11] SAP AG. RFC Purpose. [WWW]. [Viitattu 6.4.2013]. Saatavissa: [http://help.sap.com/saphelp\\_nw04/helpdata/en/6f/1bd5b6a85b11d6b28500508b5d5211/content.htm](http://help.sap.com/saphelp_nw04/helpdata/en/6f/1bd5b6a85b11d6b28500508b5d5211/content.htm).
- [12] KGS Software. SAPALink KGS ContentServer Release 4.5 Installation Guide. [WWW]. [Viitattu 6.4.2013]. Saatavissa: [http://www.kgs-software.com/index.php/en/downloads/doc\\_download/21-sapalink-45.html](http://www.kgs-software.com/index.php/en/downloads/doc_download/21-sapalink-45.html).
- [13] KGS Software. SAPALINK SDK Implementation Reference Guide. [WWW]. [Viitattu 6.4.2013]. Saatavissa: [http://www.kgs-software.com/index.php/en/downloads/doc\\_download/22-sapalink-framework-api.html](http://www.kgs-software.com/index.php/en/downloads/doc_download/22-sapalink-framework-api.html).
- [14] KGS Software. KGS Activator for SAP R/3 Reference Guide. [WWW]. [Viitattu 6.4.2013]. Saatavissa: [http://www.kgs-software.com/index.php/en/downloads/doc\\_download/18-kgs-activator.html](http://www.kgs-software.com/index.php/en/downloads/doc_download/18-kgs-activator.html).
- [15] Eloranta, V., Koskimies, K. Aligning Architecture Knowledge Management with Scrum. Proceedings of Seventh Workshop on SHARing and Reusing architectural Knowledge (SHARK 2012), ECSA/WICSA 2012, Helsinki, Finland, 21.8.2012. New York, NY, USA 2012, ACM. s. 112-115.
- [16] van Heesch, U., Eloranta, V., Avgeriou, P., Koskimies, K., Harrison, N. DCAR – Participants and roles. [WWW]. [Viitattu 6.4.2013]. Saatavissa: [http://www.dcar-evaluation.com/?page\\_id=64](http://www.dcar-evaluation.com/?page_id=64).
- [17] van Heesch, U., Eloranta, V., Avgeriou, P., Koskimies, K., Harrison, N. DCAR – Decision-Centric Architecture Review: Introduction. [WWW]. [Viitattu 6.4.2013]. Saatavissa: <http://www.dcar-evaluation.com/wp-content/uploads/2012/10/dcar-introduction.pdf>.

## LIITE1: DCAR-ARVIOINNIN PÄÄTÖSKUVAUKSET

<b>Päätöksen nimi</b>	Tilasiirtymistä seuraava toiminnallisuus toteutetaan M-Filesin tilasiirtymien skripteissä.
<b>Ongelma</b>	Kun käyttäjä tekee tilasiirtymän M-Filesissa, tätä tulee seurata erilaista tilanteesta riippuvaa toiminnallisuutta. Tilasiirtymien skriptit ovat luonnollinen paikka tälle toiminnallisuudelle.
<b>Ratkaisu / päätöksen kuvaus</b>	Toiminnallisuus toteutetaan VBScriptillä tilasiirtymien skripteihin.
<b>Harkitut vaihtoehtoiset ratkaisut</b>	Toteutetaan monimutkaisempi toiminnallisuus .NET-ympäristöön toteutetussa komponentissa, ja pidetään skriptit mahdollisimman yksinkertaisina.
<b>Päätöstä puoltavat argumentit</b>	<ul style="list-style-type: none"> <li>Ei tarvetta erillisen komponentin toteuttamiselle</li> </ul>
<b>Päätöstä vastustavat argumentit</b>	<ul style="list-style-type: none"> <li>Skriptien toteuttamiseen ei ole hyvää kehitysympäristöä</li> </ul>
<b>Lopputulos</b>	Vaihtoehtoista ratkaisua tulisi vakavasti harkita
<b>Lopputuloksen perustelu</b>	Kehittäminen .NET ympäristössä on nopeampaa hyvän kehitysympäristön vuoksi (Visual Studio).

<b>Päätöksen nimi</b>	KGS Activatoria käytetään laskujen tallentamiseen M-Filesistä SAP:iin.
<b>Ongelma</b>	Skannatut tilaukselliset laskut pitää automaattisesti tallentaa M-Filesista SAP:iin.
<b>Ratkaisu / päätöksen kuvaus</b>	Laskujen tallentamiseen käytetään KGS Activatorin DocumentPostExt-funktiota.
<b>Harkitut vaihtoehtoiset ratkaisut</b>	.NET-ympäristöön itse toteutettu komponentti tallentaa laskut SAP:iin kutsumalla suoraan SAP:in rajapintafunktioita SAP .NET Connectorin avulla.
<b>Päätöstä puoltavat argumentit</b>	<ul style="list-style-type: none"> <li>• Käytön tueksi löytyy valmiita esimerkkejä</li> <li>• Mahdollisuus pyytää KGS:ltä apua ongelmatilanteissa</li> </ul>
<b>Päätöstä vastustavat argumentit</b>	<ul style="list-style-type: none"> <li>• Riippuvuus KGS:n tuesta ongelmatilanteissa</li> <li>• Vuosittaiset lisenssimaksut</li> </ul>
<b>Lopputulos</b>	Päätös on hyvä
<b>Lopputuloksen perustelu</b>	Valmista komponenttia käyttämällä säästetään aikaa ja vältetään potentiaalisia ongelmia. Päätös ei estä oman komponentin toteuttamista tarvittaessa.



<b>Päätöksen nimi</b>	SAP Connectoria käytetään tiedon tuomiseen SAP:ista M-Filesiin.
<b>Ongelma</b>	Ostotilausten tiedot pitää tuoda M-Filesiin hyväksyntää varten.
<b>Ratkaisu / päätöksen kuvaus</b>	SAP Connector tuo jokaisen tarvittavan rivin SAP:ista kohteeksi M-Filesiin.
<b>Harkitut vaihtoehtoiset ratkaisut</b>	.NET-ympäristöön itse toteutettu komponentti tuo tiedot SAP:ista ajastetusti M-Filesiin juuri halutussa muodossa.
<b>Päätöstä puoltavat argumentit</b>	<ul style="list-style-type: none"> <li>• Käyttää valmista laajasti testattua rajapintaa</li> <li>• Valmis komponentti, joka vaatii hyvin vähän muutoksia</li> </ul>
<b>Päätöstä vastustavat argumentit</b>	<ul style="list-style-type: none"> <li>• Kohteita tulee paljon, koska tieto on hajautunut SAP:issa moneen tauluun</li> </ul>
<b>Lopputulokset</b>	Päätös on hyvä
<b>Lopputuloksen perustelu</b>	Valmista komponenttia käyttämällä säästetään aikaa. SAP Connectorin kehittäminen on mahdollista tarvittaessa.