

TP 12.1 Open Source -kirjastojärjestelmien kartoitus

Siirry [työpakkettikuvaukseen](#).



Tämä selvitys on toteutettu kirjallisuusselvityksenä ilman arvioitavien ohjelmistojen koekäyttöä tai lähdekoodin katselmointia. Ohjelmistojen dokumentaatio on osittain ristiriitaista ja jättää monilta osin tulkinnan varaa. Näiden seikkojen vuoksi vahvistuksen saaminen kaikkiin selvityksessä esille tulleisiin asioihin ja niiden perusteella tehtyihin tulkintoihin vaatisi perusteellisempaa lähdekoodin katselmointia sekä ohjelmistojen testaamista käytännössä.

Tässä kartoituksessa on tarkoituksena selvittää tarjolla olevien avoimen lähdekoodin kirjastojärjestelmien soveltuvuutta UKJ-projektin tarpeisiin. Kartoitukseen otetaan mukaan vain sellaisia järjestelmiä, joita kehitetään edelleen ja jotka omaavat UKJ-hin sisältyvää toiminnallisuutta. Selvitys toteutetaan kahdessa vaiheessa siten, että ensin tuotetaan listaus tarjolla olevista järjestelmistä, jonka jälkeen järjestelmien joukosta valitaan kaksi tai kolme järjestelmää lähempään tarkasteluun. Lähempään tarkasteluun otettavat järjestelmät valitaan järjestelmien sisältämän toiminnallisuuden sekä käyttäjä- että kehittäjäyhteisöjen laajuuden ja aktiivisuuden perusteella. Selvitys toteutetaan kokonaisuudessaan kirjallisuusselvityksenä eli arvioitavia järjestelmiä ei koekäytetä eikä asenneta testipalvelimelle lähempää tarkastelua varten.

Vaihe 1

Taulukossa 1 on esitetty listaus edelleen kehitetyistä keskeisimmistä avoimen lähdekoodin kirjastojärjestelmistä. Listaukseen on otettu mukaan ainoastaan sellaisia järjestelmiä, joita kehitetään edelleen aktiivisesti.

Järjestelmän nimi	Kehittäjä	1. julkinen versio	Vertailtu versio	Kotisivu	Koodi	Ohjelmointikielet	Tietokanta	Indeksi /hakumoottori	Lisenssi
Koha	Koha Community	2000	3.12 (03/2013)	www	Git	Perl	MySQL	Zebra	GPL
Evergreen	Gorgia Public Library Service (GPLS)	2006	2.3.5 (03/2013)	www	Git	Perl, JavaScript, Python, C	PostgreSQL	Oma	GPL
Kuali OLE	Kuali Foundation	2013	0.8 (06/2013)	www	SVN	Java, JavaScript	MySQL / Oracle	Oma	GPL
OpenBiblio	OpenBiblio-kehitystiimi	2002	0.7.1 (03/2012)	www	Mercurial	PHP	MySQL	Oma	GPL
NewGenLib	Verus Solutions	2005	3.0.4u2 (03 /2013)	www	SVN	Java	PostgreSQL	Lucene, Solr	GPL

Taulukko 1. Keskeisimmät avoimen lähdekoodin kirjastojärjestelmät.

Taulukossa 2 on listattu valittujen avoimen lähdekoodin kirjastojärjestelmien moduuleita sekä järjestelmien tukemia standardeja ja rajapintoja.

Järjestelmä	Käytön hallinta	Luettelointi	Hankinta	Kausijulkaisujen hallinta	Admin	Raportointi ja tilastointi	Standardit ja rajapinnat
Koha	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Marc21, UNIMARC, Marc export/import, Z39.50, SIP2, SRU, OAI-PMH, unAPI, OpenSearch, LDAP, Mozilla Persona, EDI, EZproxy, OpenURL
Evergreen	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Marc21, MODS, DC, SIP2, Z39.50, SRU, EDI, EZproxy, OpenURL, unAPI
Kuali OLE	Kyllä	Kyllä	Kyllä	Kyllä	Osittain	Osittain	Marc21, MarcXML, DC Tulossa: NCIP, SIP2, SRU/SRW, Z39.50, SUSHI, COUNTER, ONIX, EDI (V 1.0)
OpenBiblio	Kyllä	Kyllä	Ei	Ei	Kyllä	Ei	Marc21 & MarcXML export
NewGenLib	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Marc21 & MarcXML import/export, Z39.76, MODS 3.0, AGRIP AP, ISBD, OAI-PMH, SRU/W, Z39.50

Taulukko 2. Järjestelmien moduulit sekä tuetut standardit ja rajapinnat.

Vertailuun mukaan otetuista järjestelmistä monipuolisimman toiminnallisuuden tarjoavat Koha, Evergreen ja Kuali OLE. Koha ja Evergreen ovat tunnetuimmat sekä myös käytetyimmät avoimen lähdekoodin kirjastojärjestelmät, joilla molemmilla on laajat kansainväliset käyttäjä- ja kehittäjäyhteisöt. Kumpaankin järjestelmää käytetään eri kokoisissa ja eri tyypisissä kirjastoissa ympäri maailman, mikä tekee niistä kiinnostavia vaihtoehtoja myös tämän kartoituksen kannalta. Kuali OLE puolestaan eroaa Kohasta ja Evergreenistä monellakin tavoin, sillä se on vasta aivan elinkaarensa alkupäässä, eikä kaikkia taulukossa 2 esitettyjä toiminnallisuuksia ei ole vielä edes toteutettu, koska järjestelmä on edelleen kehitysvaiheessa. Valmistuessaan Kuali OLE:n kuitenkin pitäisi sisältää erittäin kattava toimintokokonaisuus sekä standardien ja rajapintojen tuki, joka yhdessä sen modernin arkkitehtuurin kanssa tekee siitä erittäin kiinnostavan vaihtoehdon. Myös NewGenLib-järjestelmä sisältää varsin kattavan toiminnallisuuden sekä standardien ja rajapintojen tuen, mutta jää jatkotarkastelun ulkopuolelle käyttäjä- ja kehittäjäyhteisöjen pienen koon vuoksi. OpenBiblio puolestaan karsiutuu suppean toiminnallisuutensa sekä käyttäjä- ja kehittäjäyhteisöjen pienen koon vuoksi.

Vaihe 2

Erityisesti Kohan ja Evergreenin kohdalla on tehty monia selvityksiä ja vertailuja, joissa keskitytään niiden tarjoamien toimintojen tarkasteluun ja vertailuun. Tässä tapauksessa tällaisen vertailun teko ei ole perusteltua, koska uudelle kirjastojärjestelmälle asetetut vaatimukset eivät vielä tätä selvitystä kirjoitettaessa ole riittävällä tarkkuudella tiedossa, jonka vuoksi vertailussa ei pystyttäisi kiinnittämään huomiota UKJ:n kannalta keskeisiin asioihin. Lisäksi Kuali OLE -järjestelmän keskeneräisyys estää sen täysipainoisen vertailun muiden tarkasteltavien järjestelmien kanssa. On myös erittäin epätodennäköistä, että yksikään vertailtavista järjestelmistä täyttäisi yksinään riittävällä tasolla uudelle kirjastojärjestelmälle asetettuja vaatimuksia. Edellä mainittujen seikkojen perusteella tarkemmassa vertailussa keskitytään järjestelmien toimintojen sijaan tarkastelemaan niiden arkkitehtuuria ja ennen kaikkea sitä, kuinka helposti niiden yksittäiset osat ovat vaihdettavissa tai vaihtoehtoisesti siirrettävissä muiden järjestelmien käyttöön. Uuden kirjastojärjestelmän keskeisenä ajatuksena on modulaarisuus eli järjestelmän rakentaminen useista toisiinsa löyhästi sidoksissa olevista moduuleista. Valittujen järjestelmien lähemmässä tarkastelussa onkin siis tarkoituksena paneutua nimenomaan siihen, onko valittujen järjestelmien yksittäisten osien hyödyntäminen UKJ:ssa mahdollista.

Koha

Koha koostuu Perl-ohjelmointikielellä toteutetuista moduuleista ja pakkauksista, jotka vastaavat järjestelmän kaikkien eri osa-alueiden toimintojen toteuttamisesta. Niin tietojen käsittely, tietokantayhteydet kuin käyttäjille näytettävien HTML-sivujen generoiminen on toteutettu edellä mainittujen moduulien ja pakkausten avulla.

Kohan toiminnallisuus on jaettu neljään ylätason moduuliin, jotka sisältävät useita eri tasoisia toisistaan riippuvaisia alatasen moduuleita. Osa alatasen moduuleista on nimenomaan Kohaa varten toteutettuja ja osa taas yleisesti käytössä olevia Perl-ohjelmointikielen kirjastoja, joiden tarjoamaa toiminnallisuutta Kohan omissa toteutuksissa on hyödynnetty.

Kohan neljä ylätason moduulia ovat : OPAC, Intranet, Daemons ja DB. OPAC-moduuli sisältää asiakkaiden käyttöön tarkoitetun järjestelmän osan eli hakuliittymän toteuttavan koodin sekä sen ulkoasun määrittelevät HTML-sivupohjat. Koodi ja sivupohjat on sijoitettu omiin alatasen moduuleihinsa. Intranet-moduuli sisältää puolestaan virkailijoiden käyttöön tarkoitetut järjestelmän osat eli käyttäjienhallinnan, lainauksen, aineiston hallinnan, hankinnan, järjestelmän ylläpidon, raportoinnin sekä tietueiden tuonti-/vientitoiminnot. Eri toimintoihin liittyvät osat on jaettu omiin alatasen moduuleihinsa, jonka lisäksi käyttöliittymän ulkoasun määrittelevät sivupohjat on sijoitettu omaan alatasen moduuliinsa varsinaisesta toimintalogiikasta erilleen. Toimintojen välillä navigointi sekä käyttäjien tunnistus on myös sijoitettu omaan moduuliinsa, joka toimii linkkinä muiden moduulien välillä. Daemons-moduuli sisältää järjestelmän taustaprosessit ja DB-moduuli puolestaan tietokantaoperaatioiden suorittamiseen tarvittavat osat. OPAC-, Intranet- ja Daemons-moduulit käyttävät DB-moduulin tarjoamia palveluita tietokantaoperaatioiden suorittamiseen.

Moduulien välinen tiedonvaihto ja kommunikointi on toteutettu niiden ohjelmallisten rajapintojen kautta. Käytännössä tämä tarkoittaa sitä, että tietyn moduulin tarjoamia palveluja käytettäessä on kyseinen moduuli sisällytettävä siihen ohjelman osaan, jossa palvelua halutaan käyttää. Ohjelmiston eri osien vaihtaminen toisiin on mahdollista, mutta osien toteutuksessa käytettävä ohjelmointikieli on sidottu Perl:iin, sillä sisällyttäminen on mahdollista vain samalla ohjelmointikielellä toteutettujen osien välillä.

Evergreen

Evergreen on rakennettu Open Service Request Framework (OpenSRF) –kehiksen ympärille, joka mahdollistaa ohjelmiston koostamisen useista toisiinsa löyhästi kytketyistä palveluista. OpenSRF-kehys toteuttaa palvelujen ja asiakasohjelmien välisen kommunikoinnin kehyksen sisältämän viestiväylän (messaging bus) kautta. Väylän sisällä liikkuvat viestit kulkevat JavaScript Object Notation (JSON) -muodossa Extensible Messaging and Presence Protocol (XMPP) –protokollaa käyttäen. Kehys mahdollistaa sen avulla toteutettujen palvelujen ja asiakasohjelmistojen sijoittamisen yhdelle palvelimelle tai hajauttamisen useille maantieteellisesti toisistaan erillään oleville palvelimille, jotka lisäksi myös tiettyä toiminnallisuutta tarjoavia palveluita voi olla sijoitettuna useille eri palvelimilla. Eri palvelimilla sijaitsevat palvelut kommunikoivat keskenään viestiväylän kautta.

OpenSRF-kehiksen viestiväylä käyttää viestien välittämiseen XMPP-protokollaa ja käytännössä viestiväylän toiminta rakentuu XMPP-viestejä välittävän ejabberd-palvelinsovelluksen ympärille. XMPP-protokollaa käytetään yleisesti erityisesti pikaviestintään tarkoitetuissa sovelluksissa. XMPP on XML-pohjainen protokolla, joten OpenSRF:n JSON-pohjaiset viestit on paketoitu XMPP:n XML-viestien sisälle. OpenSRF käyttää JSON-formaattia XML-formaatin sijaan, koska välitettävät tiedot pystytään esittämään tiiviimmässä muodossa, jonka lisäksi JSON-muotoisten tietojen käsittely on huomattavasti XML-pohjaisen tiedon käsittelyä kevyempää.

OpenSRF-kirjastoja on tarjolla useilla eri ohjelmointikielillä. Asiakasohjelmien toteutuksen mahdollistavia kirjastoja on tarjolla C-, Java-, JavaScript-, Perl- ja Python-ohjelmointikielillä. Palvelujen toteuttamiseen tarkoitettuja kirjastoja on sen sijaan tarjolla C-, Perl- ja Python-kielillä.

OpenSRF-sovellukset koostuvat yhdestä tai useammasta palvelusta, jotka kukin sisältävät yhden tai useamman metodin. Käytännössä palvelun sisältämät metodit toteuttavat kyseisen palvelun tarjoaman toiminnallisuuden. Metodien vaatimien parametrien tai niiden palauttamien tulosten määrää ei ole rajoitettu ja muodoltaan ne voivat olla merkkijonoja, numeroita, boolean-arvoja, taulukoita tai assosiatiivisia taulukoita (hash). Myös muiden tietorakenteiden käyttö parametreinä ja/tai paluuarvoina on mahdollista, mutta tämä edellyttää muutamien lisämäärittysten tekemistä.

Palveluita kutsuvissa asiakasohjelmissa sekä itse palveluiden toteutuksessa käytetään käytettävän ohjelmointikielen mukaisia tietotyyppejä, jotka OpenSRF-kehys muuntaa JSON-muotoon siirron ajaksi. Käytettäessä edellä listattuja perustietotyyppejä tapahtuu siirrettävien tietojen muuntaminen eri formaattien välillä OpenSRF:n toimesta automaattisesti. Käytettäessä monimutkaisempia tietorakenteita, kuten esimerkiksi Evergreenin omia luokkia edustavia olioita joudutaan muunnokset sen sijaan määrittelemään erillisessä konfiguraatiotiedostossa. Tiedosto määrittää kuinka Evergreenin luokkia edustavat oliot esitetään JSON-muodossa, jotta OpenSRF osaa tehdä tarvittavat muunnokset kyseisiin olioita sisältävien viestien välittämistä varten. Konfiguraatiotiedoston perusteella OpenSRF kykenee muuntamaan oliot JSON-muotoon sekä vastaavasti luomaan JSON-formaatista luettuja tietoja vastaavat oliot.

Kuali OLE



Kuali OLE:n kartoitus on tehty version 0.8 perusteella. Tällä hetkellä tuorein versio on 1.5-M1 (25.2.2014). Tämän dokumentin sisältö ei anna oikeaa kuvaa järjestelmän tämän hetkisistä ominaisuuksista.

Kuali Open Library Environment (Kuali OLE) –kirjastojärjestelmä on yksi voittoa tavoittelemattoman Kuali-säätiön koordinoimista järjestelmistä. Säätiön tavoitteena on tuottaa korkeakoulujen tarpeita vastaavia avoimen lähdekoodin järjestelmiä yhteistyössä korkeakoulujen kanssa. Kirjastojärjestelmän lisäksi Kuali-järjestelmien perheeseen kuuluu mm. taloushallinto-ohjelmisto, tutkimuksenhallintaohjelmisto, henkilöstöhallintajärjestelmä sekä joukko erilaisia middleware-ohjelmistoja.

Kuali OLE –kirjastojärjestelmä on oma itsenäinen järjestelmänsä, mutta monien sen toimintojen toteuttamisessa hyödynnetään muita Kuali-perheen järjestelmiä. Hyödynnettäviä Kuali-järjestelmiä ovat mm. KFS (Kuali Financial System), KIM (Kuali Identity Management), KRAD (Kuali Rapid Application Development), KEW (Kuali Enterprise Workflow) ja KRMS (Kuali Rules Management System), joiden avulla toteutetaan mm. käyttäjätietojen hallinta sekä työprosessien ja toimintojen suorittamiseen liittyvien ehtojen määrittely ja hallinta. Eri sovellukset on liitetty toisiinsa **Kuali Rice** –middleware ohjelmistopakettiin kuuluvaa KSB (Kuali Service Bus) –sovellusta käyttäen, joka toimii palveluväylänä eri sovellusten välillä. Käytännössä tämä tarkoittaa sitä, että eri sovellukset kommunikoivat keskenään palveluväylän kautta lähetettävien viestien välityksellä. KSB mahdollistaa eri Kuali-ohjelmistojen sekä myös muiden kuin Kuali-ohjelmistojen integroimisen järjestelmäkokonaisuuteen ilman ohjelmistoihin tehtäviä muutoksia. Käytännössä muiden kuin Kuali-ohjelmistojen liittäminen KSB:hen liittyy kuitenkin erinäisiä rajoituksia, jonka lisäksi vain pieni osa KSB:n tarjoamista palveluista on niiden käytettävissä. KSB:hen kytketyt ohjelmistot voivat sijaita samalla palvelimella tai ne voi olla hajautettu useille eri palvelimille. Taulukossa 3 on esitetty Kuali OLE:n toiminnassa hyödynnettäviä muita Kuali-perheen ohjelmistoja.

Lyhenne	Nimi	Kuvaus
KFS	Kuali Financial System	Taloushallinto-ohjelmisto.
Rice	Kuali Rice	Middleware ohjelmistopaketti.
KIM	Kuali Identity Management	Kuali Rice -ohjelmistopakettiin kuuluva käyttäjien, käyttäjäryhmien, roolien ja käyttöoikeuksien hallintaohjelmisto.
KEW	Kuali Enterprise Workflow	Kuali Rice -ohjelmistopakettiin kuuluva työprosessien hallintaohjelmisto.
KRMS	Kuali Rules Management System	Kuali Rice -ohjelmistopakettiin kuuluva toimintalogiikan määrittelyyn käytettävä ohjelmisto.
KSB	Kuali Service Bus	Kuali Rice -ohjelmistopakettiin kuuluva palveluväylä-ohjelmisto.
KRAD	Kuali Rapid Application Development	Kuali Rice -ohjelmistopakettiin kuuluva ohjelmistokehys Ricea käyttävien asiakasohjelmien kehittämiseen.
GOKb	Global Open Knowledge Base	Avoin e-aineistojen tietämyskanta.

Taulukko 3. Kuali OLE:n hyödyntämiä muita Kuali-perheen ohjelmistoja.

Kuali OLE:n käyttäjien ja käyttöoikeuksien hallinta on toteutettu Kuali-ohjelmistojen perheeseen kuuluvalla KIM-ohjelmistolla. Ohjelmisto sisältää mm. käyttäjien hallintaan, autentikointiin, käyttäjärooleihin ja käyttöoikeuksien hallinnointiin tarvittavan toiminnallisuuden. Erityisen kiinnostavaksi ohjelman tekee se, että sen tarjoamia toimintoja on mahdollista käyttää myös vain osittain. Käytännössä tämä tarkoittaa sitä, että esimerkiksi käyttäjän tunnistaminen voidaan hoitaa jollakin ulkoisella järjestelmällä ja jättää roolien sekä käyttöoikeuksien hallinta KIM:n vastuulle.

Kuali OLE:n toteutukseen on haettu vaikutteita palvelukeskeisestä arkkitehtuurista. Järjestelmän moduulit on liitetty palveluväylään, jonka kautta niiden tarjoamat palvelut ovat kaikkien järjestelmän osien käytettävissä. Myös muut OLE:n toiminnassa hyödynnettävät Kuali-perheen ohjelmistot on liitetty kokonaisuuteen palveluväylän välityksellä. Yhdessä eri osien tarjoamat palvelut muodostavat järjestelmän palvelurajapinnan, jonka kautta ulkoiset järjestelmät pääsevät tarvittaessa käsiksi OLE:n tarjoamiin palveluihin. Käytännössä palveluiden kuluttaminen tapahtuu HTTP-protokollan avulla kuljetettavien SOAP-viestien tai REST-palveluiden kautta. Palveluväylään on mahdollista liittää suoraan vain Java-pohjaisia Kuali-perheeseen kuuluvia palveluita, mutta muilla tekniikoilla toteutettujen palveluiden liittäminen on mahdollista välikappaleiden (*connector*) avulla. Välikappaleet toimivat eräänlaisina konverttereina palveluväylän ja siihen liitettävien palveluiden välillä.

Kuali OLE:n käyttöliittymien toteutuksessa on käytetty KRAD-ohjelmistokehystä, joka on tarkoitettu Kuali Ricea käyttävien asiakasohjelmien toteuttamiseen. Ohjelmistokehys sisältää mm. käyttöliittymän suunnittelua ja toteutusta sekä muiden Ricen tarjoamien palveluiden käyttöä helpottavia komponentteja. Kehyksen ansiosta OLE:n asiakasohjelmien käyttöliittymät ovat ilmeeltään ja toimintalogiikaltaan yhdenmukaiset, mutta käytettävyyden puolella on sen sijaan toivomisen varaa runsain mitoin.

Kuali OLE:n Document Store -nimellä kulkeva tietokantaratkaisu on periaatteessa formaattiriippumaton, sillä OLE ei aseta rajoituksia tietokantaan tallennettavan datan formaatille. Myös binääri-muotoisen tiedon tallentaminen onnistuu, mutta tällöin hakujen kohdistaminen varsinaiseen tietosisältöön ei ole mahdollista. Document Storen tallennusajustana toimii joko Oracle- tai MySQL-tietokanta. OLE:n suunnittelussa on nimenomaan lähdetty siitä, että järjestelmä ei ole sidottu tiettyihin formaatteihin ja että uusien formaattien lisääminen on mahdollista. Tietokantaan tallennetun datan käyttäminen järjestelmässä kuitenkin edellyttää, että se on järjestelmän ymmärtämässä muodossa. Käytännössä tämä tarkoittaa sitä, että datan tallentaminen tietokantaan on kyllä mahdollista erilaisissa formaateissa, mutta datan käyttäminen järjestelmässä edellyttää uutta formaattia tukevien datan käsittelystä vastaavien osien toteuttamista järjestelmään. Ilman uusien osien toteuttamista järjestelmä ei osaa käsitellä uudessa formaatissa tallennettua dataa.

Kuali OLE:n käsittelemät tietokokonaisuudet, kuten esimerkiksi bibliografiset tietueet, asiakastiedot ja lisenssit, on tallennettu dokumenttivarastoon eri tyyppisinä dokumentteina, joiden rakenne sekä tallennusformaatti vaihtelee dokumenttityypeittäin. Lisäksi saman tyyppisten dokumenttien tallentaminen erilaisissa formaateissa on myös mahdollista. Tietojen tallennuksen yhteydessä ei suoriteta tietojen muuntamista eri formaattien välillä, vaan tiedot tallennetaan suoraan siinä muodossa, kun ne tallennuksen yhteydessä vastaanotetaan. Käytännössä tietojen tallentamisessa on kuitenkin käytettävä tiettyjä ennalta määriteltyjä formaatteja, jotta tietojen haku ja käsittely järjestelmässä olisi mahdollista. Bibliografisten tietueiden tallennus on esimerkiksi mahdollista MarcXML-, Dublin Core sekä Simple (Unqualified) Dublin Core -formaateissa. Varastotietueiden tallennus on puolestaan mahdollista joko MarcXML- tai OLEML (OLE Markup Language) -formaateissa ja nidetietojen tallennus OLEML-formaatissa. Varasto- ja nidetiedot linkittyminen bibliografiaan tietueisiin on toteutettu OLE:ssa Instance-tyyppisten dokumenttien kautta, jotka tallennetaan OLEML-formaatissa. Uuden bibliografisen tietueen tallennuksen yhteydessä luodaan aina uusi Instance-dokumentti, joka sisältää sekä varasto- että nidetietueen. Koko tietuehierarkia luodaan siis aina riippumatta siitä, tallennetaanko kaikille tasolle jotain tietoa.

Palveluväylä

Suuret ohjelmistokokonaisuudet koostuvat monista eri ohjelmistoista, joiden välinen kommunikointi on usein erittäin monimutkaista johtuen ohjelmistojen käyttämistä erilaisista rajapinnoista ja tietorakenteista. Tämän seurauksena yhden ohjelmiston kommunikointitavoissa tapahtuvat muutokset voivat vaikuttaa suoraan monen muun eri ohjelmiston toimintaan ja voivat pahimmillaan estää koko ohjelmistokokonaisuuden toiminnan. Samasta syystä myös yksittäisen ohjelmiston korvaaminen toisella tai uuden ohjelmiston lisääminen kokonaisuuteen on erittäin työlästä aiheuttaen pahimmillaan muutoksia useisiin muihin järjestelmiin.

Palveluväylän keskeisenä ajatuksena on toimia ohjelmistokokonaisuuteen kuuluvien ohjelmistojen välisten viestien välittäjänä, jonka ansiosta ohjelmistojen välille ei synny suoria keskinäisiä riippuvuuksia. Käytännössä tämä tapahtuu toteuttamalla ohjelmistojen välinen kommunikointi ja tiedonvaihto palveluväylää pitkin lähetettävien viestien avulla, jolloin ohjelmistojen välille ei synny suoria keskinäisiä riippuvuuksia. Palveluväylän sisällä kulkevat viestit noudattavat väylän omaa tietomallia, josta tiedot muunnetaan aina kunkin ohjelman ymmärtämään muotoon. Käytännössä viestien muuntaminen tapahtuu väylän ja ohjelmiston välissä olevan muuntimen avulla, jolloin yksittäisessä ohjelmassa tapahtuvat muutokset vaikuttavat vain väylän ja kyseisen ohjelmiston väliseen muuntimeen. Palveluväylän ja ohjelmistojen välinen tiedonvaihto voidaan toteuttaa useilla eri tekniikoilla, kuten esimerkiksi Web Services tai REST-rajapinnan tai tiettyyn toteutustekniikkaan sidotun ohjelmointirajapinnan kautta. Tämä tekee järjestelmäkokonaisuuden ylläpidosta joustavaa ja mahdollistaa yksittäisten järjestelmien vaihtamisen ja uusien järjestelmien lisäämisen huomattavasti aiempaa vaivattomammin.

Palveluväylä on oma itsenäinen ohjelmistonsa, jonka sisältämät toiminnallisuudet ja ominaisuudet ovat ohjelmistokohtaisia ja saattavat näin ollen vaihdella suurestikin eri ohjelmistojen välillä. Valtaosa ohjelmistoista esimerkiksi tukee Web Services -rajapintaan perustuvaa liitäntäteknikkaa, mutta tiettyyn toteutustekniikkaan sidottujen ohjelmointirajapintojen tuen laajuudessa on sen sijaan suuriakin eroja. Yleisiä palveluväylään liittyviä ominaisuuksia ovat esimerkiksi viestien suodattaminen ja reitittäminen, työprosessien määrittelemisen ja koosteisten palvelujen luominen. Lisäksi myös ohjelmistojen hallintaan ja konfigurointiin käytettävissä työkaluissa on suuriakin eroja, joka osaltaan vaikuttaa ohjelmistojen käytettävyyteen.

Palvelulähtöisessä arkkitehtuurissa palveluväylä voi toteuttaa koko järjestelmäkokonaisuuden julkisen palvelurajapinnan, jonka tarjoamat julkiset palvelut voivat koostua useista yksittäisistä palveluista, jotka väylään kytketyt eri ohjelmistot toteuttavat. Tällaisia palveluita kutsutaan koosteisiksi palveluiksi ja samoja yksittäisiä palveluita voidaan helposti hyödyntää monissa koosteisissa palveluissa. Yksittäisten palvelujen ei tarvitse sijaita samalla palvelimella, vaan ne voi olla hajautettu useille toisistaan riippumattomille palvelimille. Palveluväylän toteuttaman julkisen palvelurajapinnan ansiosta rajapintaa käyttävien sovellusten ei tarvitse tietää millä palvelimella ja missä IP-osoitteessa yksittäiset palvelut sijaitsevat, vaan pelkkä palvelurajapinnan osoitteen tunteminen riittää. Palveluväylä vastaa palvelurajapintaan tulneiden palvelukutsujen välittämisestä pyydetyn palvelun toteuttaville yksittäisille palveluille ja niiden palauttamien vastausten välittämisestä kutsujalle.

Yhteenveto

Koha ja Evergreen ovat jo asemansa vakiinnuttaneita vuosia käytössä olleita järjestelmiä, joilla kummallakin on laaja kansainvälinen käyttäjäkunta. Kuali OLE on puolestaan uusi edelleen rakenteilla oleva järjestelmä, joka ei keskeneräisyytensä vuoksi ole vielä käytössä yhdessäkään kirjastossa. [Chicago yliopiston kirjasto](#) on kuitenkin jo ilmoittanut ottavansa OLE:n tuotantokäyttöön vuoden 2014 kesäkuussa ja käyttöönottoon liittyvät valmistelut ovat jo meneillään. Chicagon käyttöönotettava versio tulee olemaan versio 1.5, jonka arvioitu julkaisu-aika on vuoden 2014 ensimmäisellä neljänneksellä. Asiakkaiden hakuliittymänä Chicagon yliopiston kirjastolla on tarkoituksenaan käyttää avoimen lähdekoodin VuFind-ohjelmistoa. Lisäksi myös Englantilainen [Bloomsbury Colleges](#) -konsortio on [ilmoittanut](#) valinneensa Kuali OLE:n uuden kirjastojärjestelmänsä pohjaksi. Tarkkaa aikataulua tuotantokäyttöön siirtymisestä ei toistaiseksi ole, mutta ensimmäisen konsortion käyttöön tarkoitettua testiversion pitäisi valmistua vuoden 2013 lopulla. Tavoitteena vaikuttaisi olevan yhden keskitetyn ylläpidetyn Kuali OLE -instanssin pystyttäminen kaikkien konsortion jäsenten käyttöön. Asiakkaiden hakuliittymänä konsortio aikoo Chicagon yliopiston kirjaston tapaan käyttää VuFind-ohjelmistoa.

Koha, Evergreen ja Kuali OLE ovat kaikki rakenteeltaan modulaarisia ainakin siinä mielessä, että ne kaikki rakentuvat useista eri osista, jotka kommunikoivat keskenään erilaisten rajapintojen kautta. Kohan tapauksessa rajapinta on puhtaasti ohjelmallinen ja on näin ollen sidoksissa tiettyyn toteutustekniikkaan, Perl-ohjelmointikieleen. Evergreenin tapauksessa järjestelmän eri osat eivät ole suoraan yhteydessä toisiinsa ohjelmallisen rajapinnan kautta, vaan niiden välinen kommunikointi on toteutettu viestiväylän avulla. Tässäkin tapauksessa sekä palvelujen että asiakasohjelmien toteutus on kuitenkin sidottu tiettyihin tekniikoihin ja ohjelmointikieliin viestiväylän toteutustekniikasta johtuen. Kuali OLE on puolestaan rakennettu siten, että sen eri osien välillä ei periaatteessa pitäisi olla suoria riippuvuuksia, vaan osat kommunikoivat keskenään palveluväylän välityksellä. Käytännössä järjestelmän eri osien välillä on kuitenkin riippuvuuksia, sillä OLE:n käyttämä KSB-palveluväylä on toteutettu siten, että siihen kytkettyjen järjestelmien on oltava toteutettu Java-ohjelmointikielellä ja sisällettävä tiettyjä Kuali Rice -järjestelmän osia. OLE:n toteutuksessa on lisäksi käytetty myös KRAD-ohjelmistokehystä, jonka johdosta OLE on teknisesti sidoksissa sekä Riceen KRAD:iin. OLE:n sisäiset moduulit vaikuttavat kuitenkin sen sijaan dokumentaation perusteella itsenäisiltä kokonaisuuksilta, mutta täyden varmuuden saaminen edellyttäisi lähdekoodin katselmoitua.

Koha ja Kuali OLE tarjoavat puhtaasti www-pohjaiset virkailijakäyttöliittymät eli työasemakohtaisia asennuksia ei tarvita. Evergreenin virkailijakäyttöliittymät on sen sijaan asennettava työasemakohtaisesti. Kohan tapauksessa käyttöliittymä ja sen hyödyntämät palvelut ovat tiiviisti sidoksissa toisiinsa. Palvelut on suunniteltu tietyn käyttöliittymän tarpeisiin ja vaihtoehtoisen käyttöliittymän toteuttaminen edellyttäisi myös palvelujen tuplaamista ja mahdollisesti myös muokkaamista vaihtoehtoisen käyttöliittymän tarpeita vastaaviksi. Evergreenin ja Kuali OLE:n tapauksessa käyttöliittymä ja sen hyödyntämät palvelut on puolestaan erotettu toisistaan, jonka ansiosta useat vaihtoehtoiset käyttöliittymät voivat hyödyntää samoja järjestelmän tarjoamia palveluita. Koha ja Evergreen sisältävät virkailijaliittymien lisäksi myös asiakaskäyttöön tarkoitettua hakuliittymää, OPAC:it. Kuali OLE ei sen sijaan sisällä erillistä asiakaskäyttöön tarkoitettua hakuliittymää, vaan OLE tarjoaa tarvittavat rajapinnat, jotka mahdollistavat erillisen asiakasliittymä-ohjelmiston integroinnin sen kanssa. Ainakin avoimen lähdekoodin VuFind-ohjelmistoa ollaan rakentamassa yhteensopivaksi OLE:n kanssa ja OLE:n ensimmäisenä tuotantokäyttöön ottava Chicagon yliopiston kirjasto aikoo käyttää sitä asiakasliittymään toteuttamiseen. Myös Kuali OLE:n käyttöönottoa valmistelevalle Englantilainen Bloomsbury Colleges -konsortio tulee käyttämään VuFindia asiakasliittymänään.

Vertailussa mukana olleista järjestelmistä sekä Kuali OLE:n että Evergreenin toteutustapa on palvelukeskeinen eli järjestelmät tarjoavat palvelurajapinnan, jonka kautta asiakasohjelmat käyttävät niiden tarjoamia palveluita. Käytännön toteutuksien osalta järjestelmät kuitenkin eroavat toisistaan. Kuali OLE:n palvelut ovat KSB:n kautta käytettävissä Java-ohjelmointikielellä toteutetuista järjestelmistä käsin sekä internetin kautta SOAP-protokollan ja mahdollisesti myös REST-palveluiden avulla. KSB:n kautta tarjolla olevien palveluiden täysimääräinen hyödyntäminen on kuitenkin mahdollista vain Java-ohjelmointikielellä toteutetuista Riceen ohjelmistokirjastoja käyttävistä järjestelmistä käsin. Evergreenin toteutustapa sen sijaan sitoo sen palveluja käyttävät järjestelmät OpenSRF-kehiksen ja sitä tukevien ohjelmointikielten käyttöön. OpenSRF-kehys on avoimen lähdekoodin toteutus, mutta sen avulla toteutettujen palvelujen hyödyntäminen edellyttää asiakasohjelmille tarkoitettua OpenSRF-ohjelmistokirjaston käyttöä, joka on saatavilla tietyille ohjelmointikielille.

Kuali OLE:n palveluväylään perustuva toteutustapa on periaatteessa joustavin ja tarjoaa parhaat edellytykset modulaarisen ohjelmistokokonaisuuden rakentamiseen. Toteutustapa mahdollistaa ohjelmistokokonaisuuden koostamisen useista eri osista siten, että osien vaihdolla tai uusien lisäämisellä ei tarvitse olla vaikutusta muihin osiin. Kuali OLE:n käytännön toteutus kuitenkin rajoittaa tämän joustavuuden Kuali-järjestelmien piiriin, sillä Riceen ohjelmistokirjastoja käyttämättömien järjestelmien saumaton integrointi KSB:hen ei ainakaan tällä hetkellä näyttäisi olevan mahdollista. Käytännössä OLE:n omien moduulien korvaaminen on siis mahdollista, kunhan korvaavassa toteutuksessa on käytetty Riceen ohjelmistokirjastoja. Toisena vaihtoehtona on ohjelmistojen liittäminen KSB:hen Riceen ohjelmistokirjastoihin perustuvien välikappaleiden avulla, jolloin liittäminen voidaan tehdä ilman liitettävään ohjelmistoon tehtäviä muutoksia. OLE:n moduulien hyödyntäminen muissa järjestelmissä pitäisi myös olla mahdollista, mutta tämä edellyttää moduuleihin tehtäviä muutoksia. Tarvittavien muutosten ja niistä aiheutuvan työn määrä on arvioitava erikseen moduulikohtaisesti.

Yksikään arvioiduista järjestelmistä ei täytä UKJ:lle asetettuja kriteereitä siinä määrin, että sitä voitaisiin käyttää UKJ:n toteutuksessa sellaisenaan. Kuali OLE:n palveluväylään perustuva arkkitehtuuri vastaa periaatteellisella tasolla UKJ:n vaatimuksia, mutta OLE:n käytännön toteutukseen liittyvät erilaiset teknologiavalinnat asettavat tiettyjä kysymyksiä liittyen OLE:n käyttämisen UKJ:n pohjana. UKJ:n kehityksen ja ylläpidon kannalta olisi joustavampaa käyttää erillistä palveluväyläratkaisua, joka ei aseta siihen liitettävälle järjestelmälle toteutustekniikkaan liittyviä rajoituksia KSB:n tapaan. Lisäksi monet erilliset palveluväyläohjelmistot ovat toiminnallisuudeltaan KSB:tä monipuolisempia, jonka lisäksi niillä on myös laajemmat käyttäjä- ja kehittäjäyhteisöt. Erillisen palveluväyläohjelmiston käyttäminen ei sulki pois mahdollisuutta hyödyntää OLE:n tai jonkun muun arvioidun järjestelmän moduuleita tai niiden sisältämää toiminnallisuutta UKJ:n toteutuksessa. Olennaista on, että kokonaisuuteen liitettävät järjestelmät sisältävät toimivat ja hyvin dokumentoidut rajapinnat palveluväylään liittämistä varten.

Linkkejä:

<http://opensourceils.com/> (Open Source tietoportaali kirjastonhoitajille. Enimmäkseen KOHAa ja EverGreeniä tuntuu käsittelevän)

<http://opensource.califa.org/> (Open Source Open Libraries. Kalifornian kirjastojen? sivusto avoimen lähdekoodin kirjastojärjestelmistä. Sisältää mm. toiminnallisen määrittelyn kirjastojärjestelmälle. Tarkemmin [TP13 -kuvailussa](#))

<http://foss4lib.org/package-type/Integrated-Library-System> (Sivusto listaa eri OS kirjastosysteemejä)

<http://features.galecia.com/> (Koha-Evergreen-vertailu)

http://en.wikipedia.org/wiki/List_of_next-generation_library_catalogs (Wikipedia, vertailu)