



1000 Ljubljana - P.p. 639
tel: (01) 366-42-50
fax: (01) 366-43-00

Portal eSodstvo Demo podportal

Tehnična dokumentacija

KONTROLA VERZIJ

ZADNJA VERZIJA:

Verzija	0.3
Datum	02/09/14
Avtor	Renata Kišek, Gregor Lah, Kristjan Cočev
Odgovornost	Renata
Zaupnost	Interno gradivo
Datoteka	/ Users/renata/Documents/Razno/eSodstvo_DemoPodport al_09_02_2014_18_56.odt

ZGODOVINA:

Verzija	Datum	Avtor	Opis
0.1	02/02/14	Renata Kišek	Osnutek
0.2	02/07/14	Gregor Lah	Dokumentacija
0.3	02/09/14	Kristjan Cočev	Dokumentacija

REVIZIJE:

Revizija	Datum	Avtor	Opis

ZAŠČITA DOKUMENTA

© 2014 Vrhovno sodišče Republike Slovenije
Vse pravice pridržane. Reprodukcijska po delih ali v celoti na kakršni koli način in na katerem koli mediju ni dovoljena brez pisnega dovoljenja avtorja. Omejitve ne veljajo za državne organe Republike Slovenije. Vsaka kršitev se lahko preganja v skladu z Zakonom o avtorski in sorodnih pravicah in Kazenskim zakonikom Republike Slovenije

Kazalo vsebine

1. Uvod.....	4
2. Portal eSodstvo.....	4
2.1. Visokonivojska arhitektura trenutne rešitve.....	4
2.3. Dobre prakse trenutne rešitve portala eSodstvo.....	8
2.4. Pomanjkljivosti trenutne rešitve portala eSodstvo	8
3. Popis portalskih servisov in zahtev portala eSodstvo	8
3.1. Uporaba zunanjih servisov	8
3.3. Skupni servisi in funkcionalne zahteve portala eSodstvo	9
4. Predlog nove arhitekture portala eSodstvo	11
4.1. Uvod.....	11
4.2. Nova arhitektura portala eSodstvo.....	11
4.2.1. Integracija nove arhitekture v produkcijsko okolje obstoječe arhitekture....	12
4.2.2. DTO-common.....	12
4.2.3. Services-common.....	12
4.2.5. ESB	13
4.2.6. Subportal-base.....	13
4.2.7. VS-podportal.....	13
4.2.8. Avtenitkacija in avtorizacija uporabnikov podportala.....	14
4.3. Razvojno okolje za gradnjo podportalov.....	15
4.3.1. Uvod	15
4.3.2. Priprava razvojnega okolja.....	15
4.3.3. Priprava VS-podportal ogrodja.....	16
4.3.4. Zagon VS-podportal ogrodja.....	16
4.3.5. Predstavitveni nivo (Presentation Layer)	17

1. Uvod

Dokument podaja tehnične specifikacije in navodila za nameščanje ogrodja portala eSodstvo, ki sestoji iz trenutne verzije portala eSodstvo (Varnostna shema) in VS-podportal ogrodja (ogrodje za lažjo izgradnjo novih podportalov).

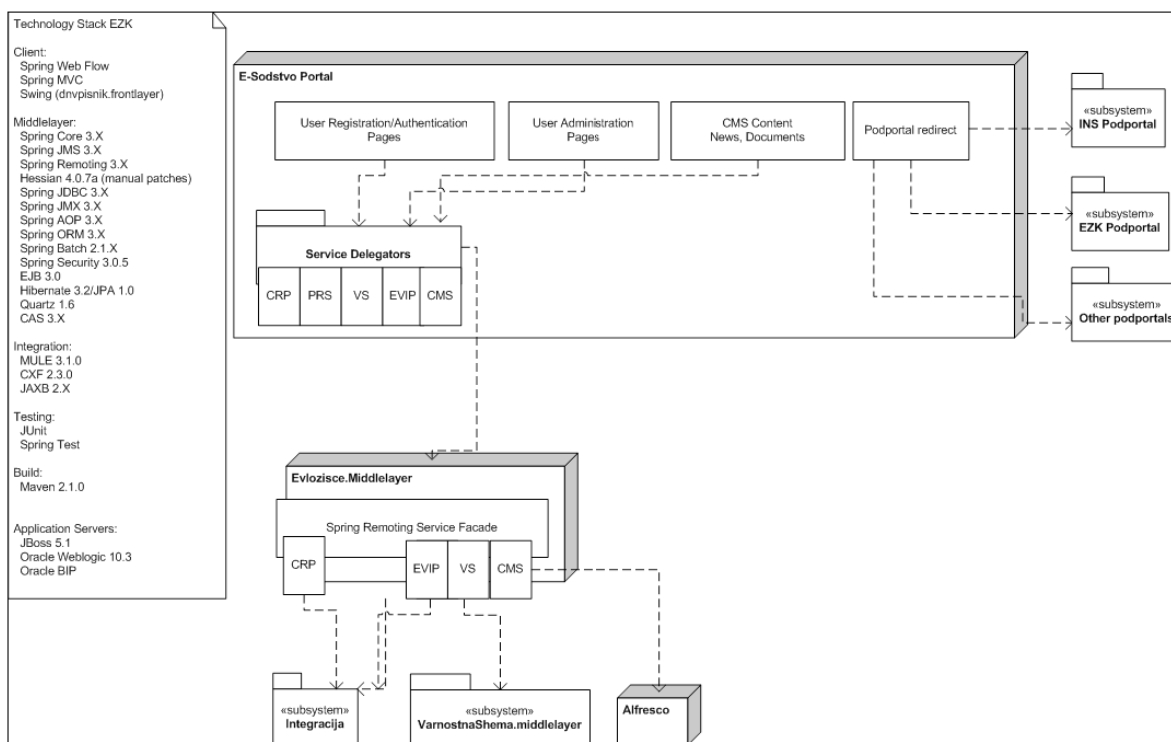
Dokument najprej podaja visokonivojsko arhitekturo trenutne rešitve portala eSodstvo ter popis skupnih funkcionalnosti oziroma servisov.

V nadaljevanju dokument podaja predlog nove arhitekture portala eSodstvo in poda navodila za vzpostavitev razvojnega okolja ter izgradnjo in opis VS-podportal ogrodja.

2. Portal eSodstvo

2.1. Visokonivojska arhitektura trenutne rešitve

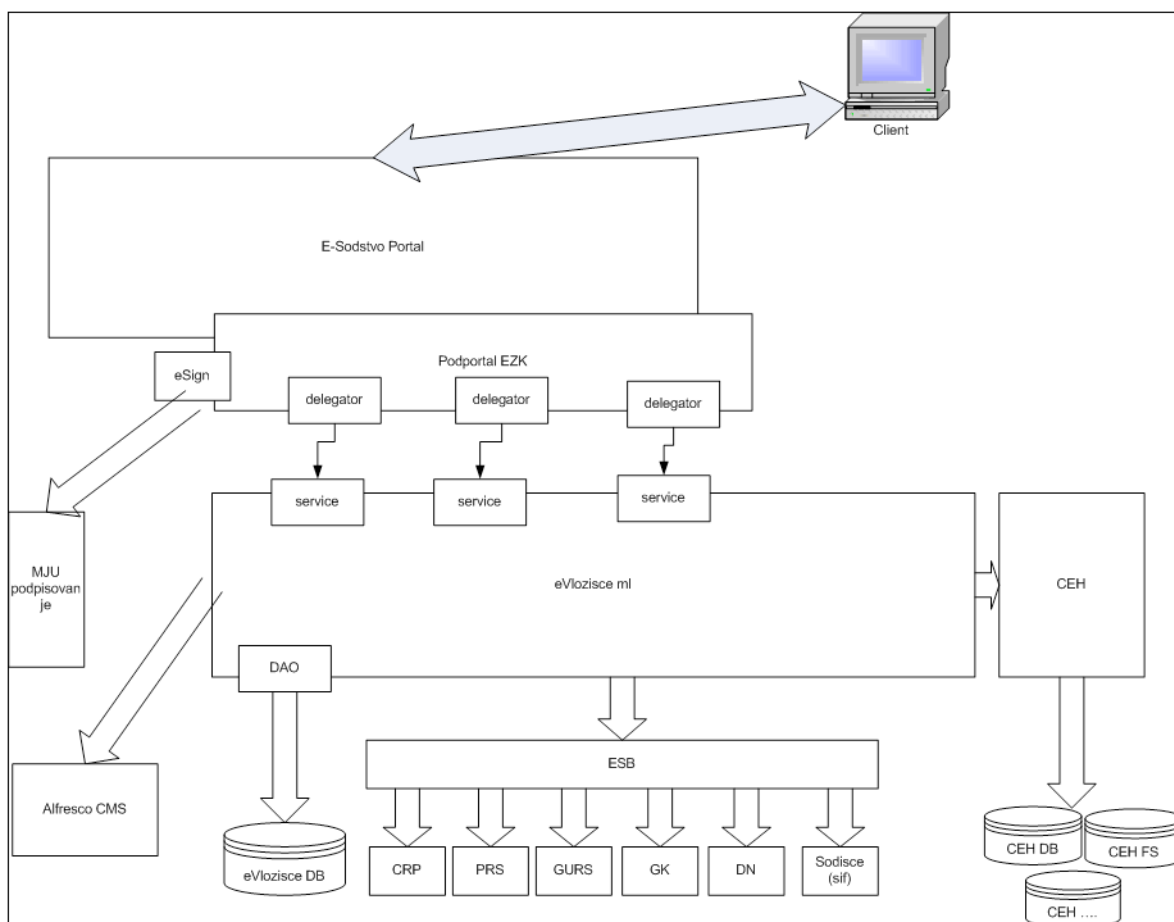
Spodnja slika predstavlja trenutno visokonivojsko arhitekturo portala eSodstvo in njegovih podportalov:



Rešitev je bila zasnovana in izvedena hkrati z uvedbo Elektronske zemljiške knjige (eZK) v letu 2011. Ena od osnovnih zahtev projekta eZK je bila, da je sistem tako zasnovan, da je na enostaven način možno dodajati podportale tako, da se uporabi skupne servise. Hkrati z uvedbo Elektronske zemljiške knjige je bil objavljen podportal eINS, kasneje pa še elzvršba. Žal rešitev ni idealna, saj obe rešitvi

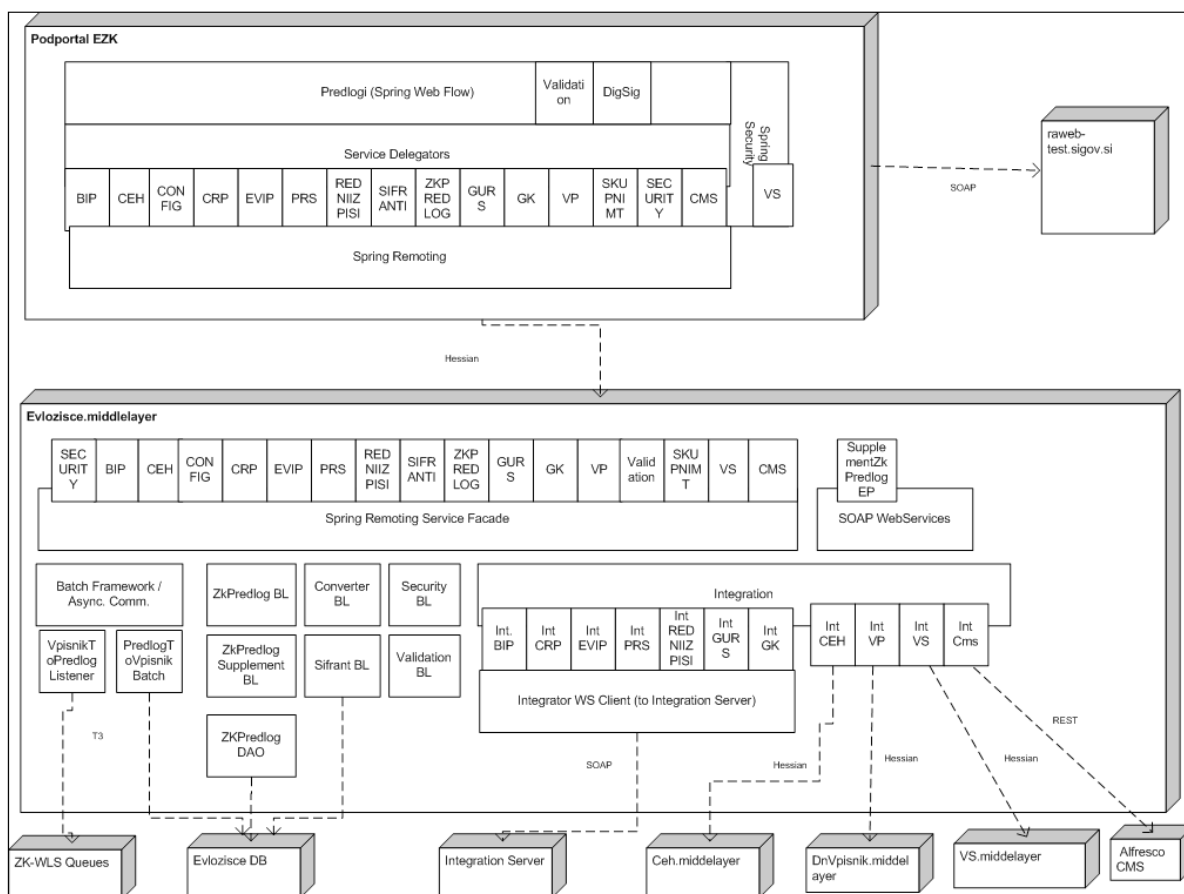
uporabljata eZK-jev Middlelayer oz. ni natančne ločnice, kaj sodi pod podportal eZK, kaj pa so skupne funkcionalnosti oz. servisi.

Spodnja slika prikazuje visokonivojsko arhitekturo podportala eZK:

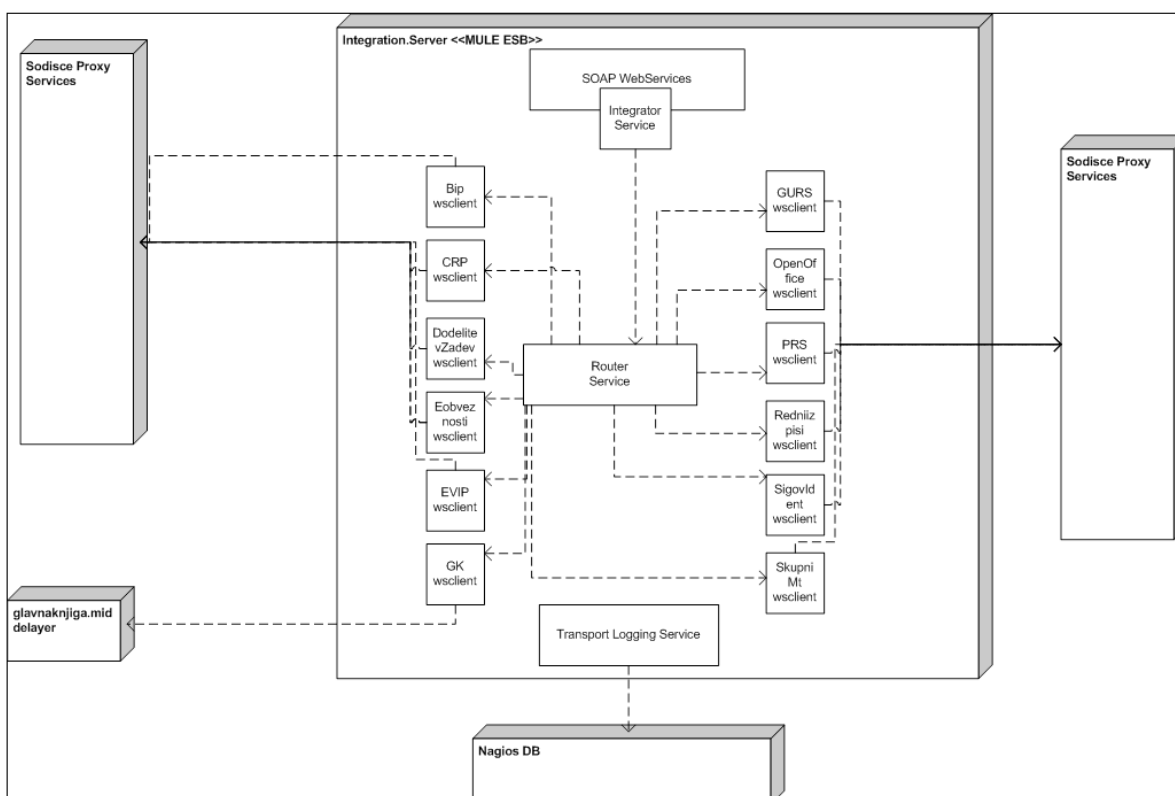


Ostala dva podportala uporabljata iste dostope do zunanjih servisov, isti CEH, iste komponente za podpisovanje, časovno žigosanje ter Alfresco CMS za novice.

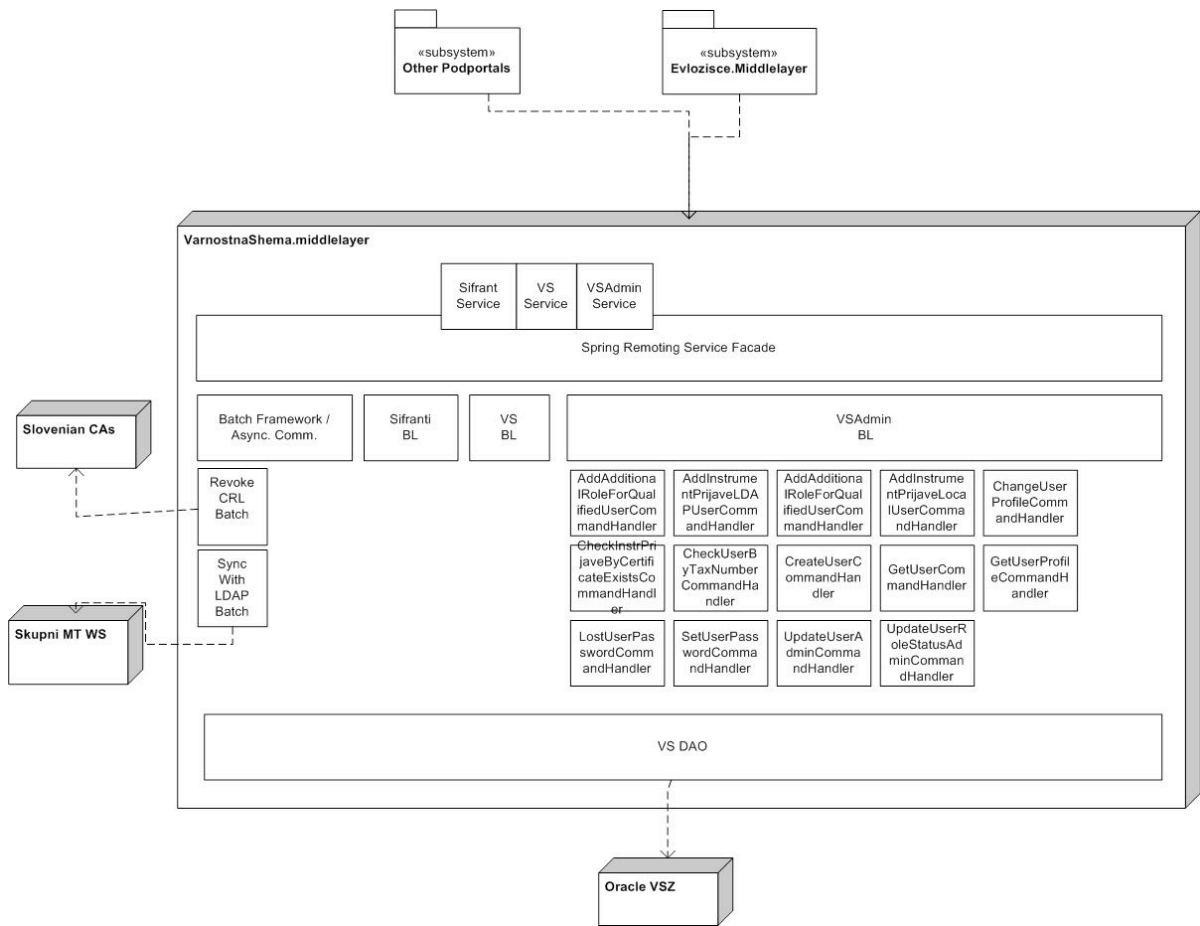
Struktura podportala eZK je predstavljena na naslednji sliki:



Integration server predstavlja spodnja slika:



Strukturo varnostne sheme opisuje spodnja slika:



2.2. Dobre prakse trenutne rešitve portala eSodstvo

1. Dostop do podportalov je omogočen preko enotne vstopne točke (<https://evlozisce.sodisce.si/>)
2. Vsi podportali uporabljajo eno (isto) Varnostno shemo (VaSh) - to pomeni, da je uporabnik enkrat vnesen v VaSh, lahko v eno uporabniško skupino ali več različnih.
3. Vsi podportali uporabljajo iste zunanje servise (CRP, PRS, elektronski podpis, ...) ter nekatere interne (VIP, eObveznosti,...)
4. Podportala eZK in eINS uporabljata enake strukture predlogov, isto hrambo predlogov in prilog – dokumentov, preden jih prevzame ustrezen vpisnik
5. Podportala eZK in eINS imata enako uporabniško izkušnjo.

2.3. Pomanjkljivosti trenutne rešitve portala eSodstvo

1. Izgled podportalov ni usklajen
2. Oblika uporabniških navodil ni enaka.
3. Manjka hitri pregled čez navodila (navodila v html obliki z iskalnikom, online pomoč,...)
4. Pri postopkih, kjer je potrebno pripeti priloge, ni alternativne poti (če je, na primer, priloga obsežen dokument, bi bilo smiselno asinhrono glede na predlog / vlogo ta dokument oddati po alternativni poti).
5. Nekateri podportali uporabljajo programske pakete, ki so označeni z eZK (taki, ki so skupni, bi se morali imenovati skupni ali common ali po nazivu in ne po podportalu).

3. Popis portalskih servisov in zahtev portala eSodstvo

Portalske skupne servise in funkcionalnosti lahko razdelimo na:

- zunanji servisi (portal eSodstvo omogoča enoten dostop do teh servisov za vse podportale in tudi druge rešitve - vpisniki)
- skupni servisi portala eSodstvo.

3.1. Uporaba zunanjih servisov

1. Kvalificirana digitalna potrdila (preverjanje veljavnosti)
2. CRP
3. PRS
4. GURS
5. Elektronsko podpisovanje in časovno žigosanje elektronskih dokumentov

3.2. Skupni servisi in funkcionalne zahteve portala eSodstvo

1. Portal eSodstvo
 1. enotna vstopna točka za zunanje in notranje uporabnike
 2. varnost (security)
 3. dostop do Varnostne sheme
 4. redirekcija na podportale (z izbrano uporabniško skupino), menjava med podportali
 5. GUI prilagojen za mobilne naprave (tablični računalniki, pametni telefoni)
2. Varnostna shema
 1. registracija uporabnikov
 2. avtorizacija in avtentikacija uporabnikov
 3. administracija uporabnikov
 4. izbira uporabniške skupine za dostop do podportalov
3. CMS
 1. vsebina
 2. novice
 3. dokumenti (navodila ipd)
4. Podportali
 1. enoten izgled uporabniškega vmesnika
 1. .css (glava, noga, menijska struktura, gumbi Naprej, Nazaj, Shrani, Predogled predloga / vloge, ...)
 2. uporaba t.i. „business rules“, kjer je to smiselno
 3. Pripenjanje dokumentov
 1. sinhrono
 2. asinhrono
 3. možnost vnosa id-ja dokumenta (na primer, če je priloga Sklep o izvršbi z opravilno številko XY),
 4. preverjanje obstoja dokumenta z določeno opravilno številko
 4. online pomoč
 1. enoten izgled
 2. hitro iskanje
 3. pretvorba v pdf na osnovi osniutkov.
 5. enotna shema za predloge / vloge
 6. enotna hramba predlogov / vlog
 7. poenoten dostop do vpisnikov
 8. poenoten dostop do šifrantov
 9. takse (nalogi, plačilo z moneto, karticami, spletne banke, preverjanje plačila)
 10. Pregledi (predlogi / vloge v delu, oddani predlogi / vloge, status predloga / vloge, povratne info)
 11. Listanje po Spisu (notarji, odvetniki,...)
 12. Vpogled v ekoledar
5. Podpora zunanjim storitvam, sistemom (v prihodnosti)
 1. „Upload“ predloga / vloge, paketni prenos več predlogov / vlog
 2. Integracija z zunanjimi sistemi, storitvami (spletni servisi)
 3. Validacija sheme predloga / vlog
 4. Potrjevanje prevzema (podpora procesu)

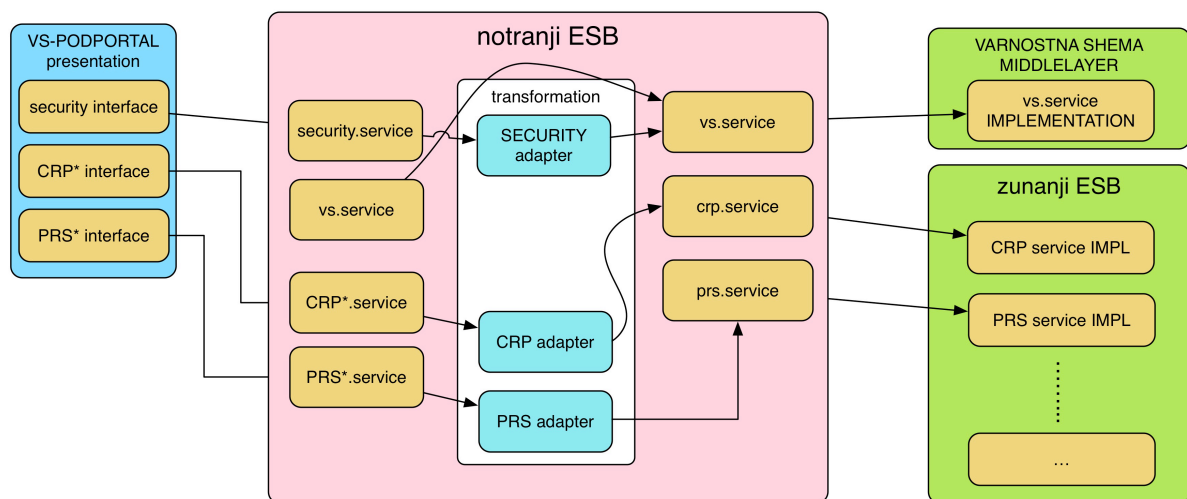
6. Mobilna aplikacija eSodstvo (promocija sodstva)
 1. obveščevalnik (predvsem za notranje uporabnike)
 1. dogodki iz eKoledarja
 2. razne novice (različnim uporabniškim skupinam)
 3. prijava na izobraževanja, tečaje
 4. itd
 2. Pregledi svojih predlogov / vlog (kot podportali)

4. Predlog nove arhitekture portala eSodstvo

4.1. Uvod

Nova arhitektura portala eSodstvo je osnovana okoli t.i. ESB (Enterprise Service Bus), ki omogoča uporabo skupnih servisov. ESB omogoča predstavitev večine integracijske kode in logike iz podportalov nanj, kar posledično pomeni večjo fleksibilnost celotne arhitekture ter omogoča hitrejši razvoj posameznih podportalov in storitev. Poleg dodanega ESB spremembe arhitekture vključujejo tudi implementacijo demo podportala, ki bo služil kot osnova za implementacijo novih podportalov.

4.2. Nova arhitektura portala eSodstvo



Nova arhitektura je sestavljena iz naslednjih modulov:

- DTO-common
- Services-common
- ESB
 - Transformers*
 - Adapters*
 - Stubs*
 - WAR*
- Subportal-base
- Vs-podportal
- Vrhovno (master)

*Zaradi lažjega razvoja trenutno še niso moduli, ampak paketi.

4.2.1. Integracija nove arhitekture v produkcijsko okolje obstoječe arhitekture

Za razvojno okolje uporabljamo modul ESB, ki kliče stub implementacije adapterjev posameznih storitev. Za produkcijsko okolje se lahko uporabi aplikacija istega modula, le da se stub implementacije zamenja s konkretnimi implementacijami. V nadaljnjih fazah lahko modul ESB zamenja katerakoli druga implementacija, saj je komunikacija med podportali in modulom ESB definirana preko vmesnikov v modulu Services-common.

4.2.2. DTO-common

Vsebuje definicije objektov, ki se uporabljajo za komunikacijo med ESB in novimi podportali. Nove DTO objekte smo vpeljali, da se vzpostavi neodvisnost od programskih paketov EZK, kar omogoča pričetek razvoja novih podporalov brez dostopa do obstoječe kode EZK. Za kompatibilnost z obstoječo arhitekturo skrbijo transformatorji na ESB.

V verziji 1.0.0 so implementirani DTO-ji za naslednje funkcionalnosti:

- PRS
- CRP
- Security

4.2.3. Services-common

Vsebuje nove definicije storitev. Nove definicije so tako kot DTO objekti neodvisne od obstoječe kode EZK. Za kompatibilnost z DTO skrbijo adapterji v modulu ESB.

V verziji 1.0.0 so definirani servisi za naslednje funkcionalnosti:

- PRS
- CRP
- Security

4.2.4. ESB

Vsebuje: implementacije adapterjev med obstoječimi servisi in novimi definicijami storitev, implementacije trasformatorjev podatkovnih objektov (DTO ↔ obstoječi objekti). V prvi verziji, katere osnovni namen je podpora za razvoj novih podportalov, ESB vsebuje tudi Stub implementacije servisov z namenom, da lahko razvijalci čim lažje začnejo z razvojem.

Adapterji:

- CRPServiceAdapter
- PRSServiceAdapter
- SecurityServiceAdapter

Mocki:

- User
- UserGroup

Stubi:

- AlfrescoServiceStub
- CRPServiceStub
- PRSSearchStub
- PRSServiceStub
- AdminSecurityServiceStub
- SecurityServiceIVSStub

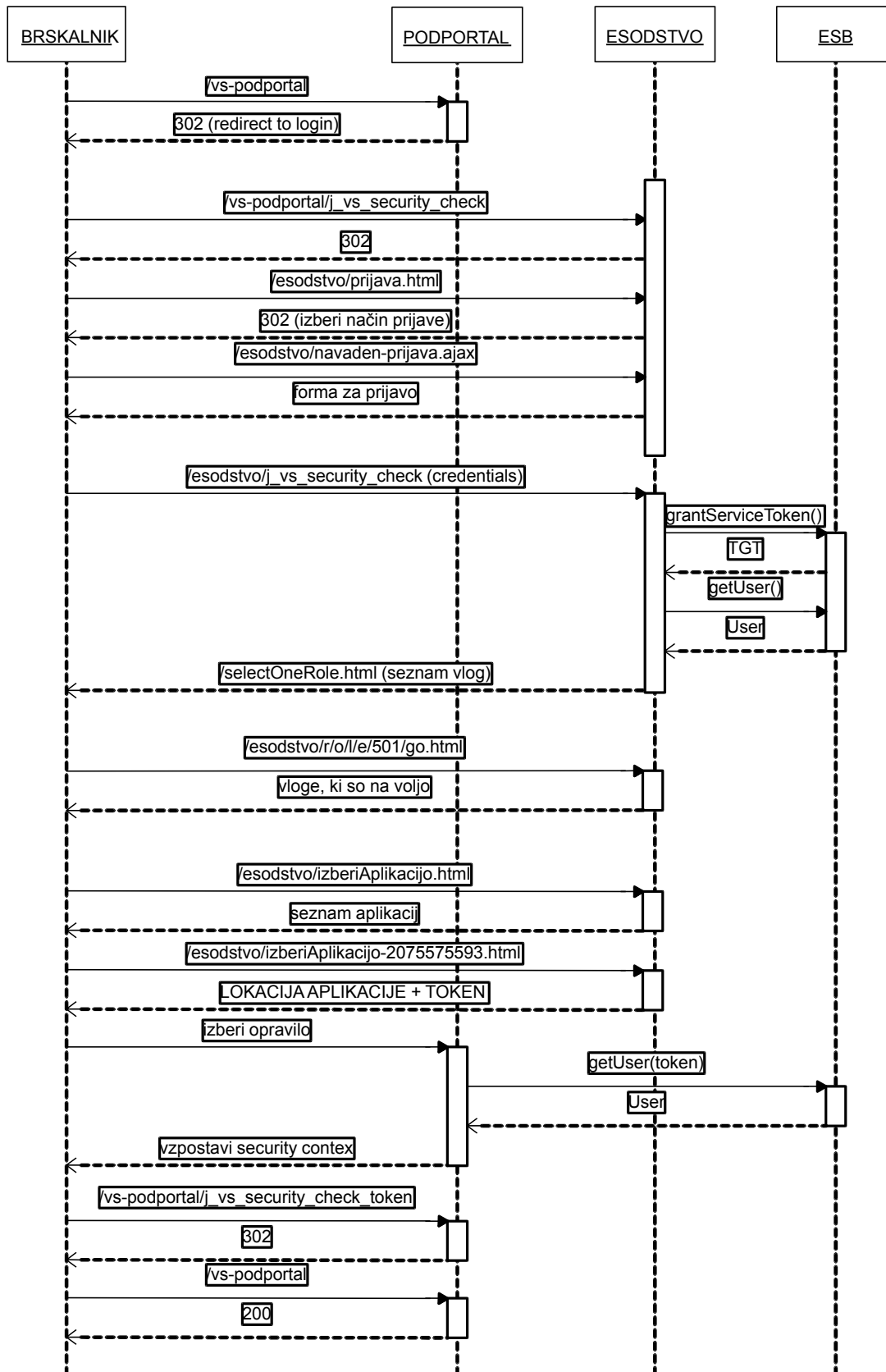
4.2.5. Subportal-base

Modul vsebuje osnovne razrede, ki so specifični za nivo pogleda in so skupni vsem novim podportalom. Npr. podpora za avtentikacijo z uporabo varnostne sheme.

4.2.6. VS-podportal

Implementacija demo podportala oz ogrodje za nove podportale, ki služi kot primer uporabe nove arhitekture ter best practice način razvoja novih podportalov. Podroben opis modula je na voljo v poglavju 4.3 Razvojno okolje.

4.2.7. Avtenitkacija in avtorizacija uporabnikov podportala



4.3. Razvojno okolje za gradnjo podportalov

4.3.1. Uvod

Razvojno okolje je razdeljeno na dva nivoja. Prvi nivo razvojnega okolja omogoča razvoj vseh modulov (dto-common, services-common, esb, subportal-base in VS-podportal), drugi nivo pa le delo na VS-podportal. Ta razdelitev sili razvijalce podportalov, da se držijo API-jev, ki so definirani v moduli iz prvega nivoja. Prvi nivo je tudi odvisen od obstoječih EZK knjižnic, medtem ko je drugi nivo odvisen samo od knjižnic na prvem nivoju.

4.3.2. Priprava razvojnega okolja

Za razvoj sta potrebna Java SDK 7.x ter Maven 3.1.x.

Visokonivojska datotečna struktura programskega paketa prvega nivoja je naslednja:

- Vrhovno: (m)
 - dto-common (m)
 - services-common (m)
 - esb (m)
 - subportal-base (m)
- environment (r)
- VS-podportal (m)

(m) – maven modul

(r) – datoteka z resurci in konfiguracijskimi datotekami

Environment vsebuje datoteke, ki so potrebne za zagon razvojnega paketa. Ob gradnji modula vrhovno se posodobijo jar datokete v environment.

Za uporabo prvega nivoja, mora imeti razvijalec na voljo obstoječe pakete ezk. Paket lahko zgradi z zagonom mvn clean install na master modulu vrhovno.

Programski paket drugega nivoja je sestavljen samo iz modula VS-podportal in podportalskih datotek, ki se nahajajo v environment-u. Zip arhiv programskega paketa drugega nivoja se ustvari ob gradnji modula vrhovno na njegovem target direktoriju.

Razvijalec na drugem nivoju mora iz paketa prekopirati konfiguracijske datoteke in namestiti jar knjižnice, ki so priloženi v paketu. Podrobna navodila so na voljo v paketu v README-podportal.txt.

VS-podportal ogrodje omogoča hitrejši razvoj podportalov v VS okolju.

Temeljno ogrodje, uporabljeno v razvojnem okolju je Spring IO platforma.

Uporabljeni moduli Spring IO platforme:

1. Spring Framework
2. Spring Security
3. Spring Web MVC
4. Spring Web Flow

Druge uporabljene tehnologije:

1. Thymeleaf
2. Twitter Bootstrap
3. JQuery

Definicije posebnih izrazov, uporabljenih v priročniku:

CODE_ROOT: oznaka korenskega direktorija izvorne kode razvojnega okolja

WAR: datoteka s končnico .war, vsebuje zapakirano obliko spletne aplikacije, pripravljeno za namestitev na aplikacijski strežnik

4.3.3. Priprava VS-podportal ogrodja

Za izgradnjo podportalskega ogrodja sta potrebni Java SDK 7.x ter Maven 3.1.x.

Poleg tega je za izgradnjo razvojnega okolja potrebno namestiti module, ki jih potrebuje podportal, prekopirati konfiguracijske datoteke ter zgraditi modul vs-podportal. Podrobna navodila so na voljo v programskem paketu v datoteki README-podportal.txt.

4.3.4. Zagon VS-podportal ogrodja

Za zagon VS-podportal ogrodja na vgrajenem aplikacijskem strežniku (jetty) se v ukazni lupini (command prompt) vašega sistema postavite na CODE_ROOT ter zaženite naslednji ukaz:

```
mvn jetty:run
```

Počakajte, da se strežnik postavi, v ukazni lupini boste videli naslednjo vrstico:

```
[INFO] Started Jetty Server
```

Nato v brskalniku pojdite na naslov:
<http://localhost:8080/vs-podportal/>

4.3.5. Predstavitveni nivo (Presentation Layer)

Predstavitveni nivo je osnovan na dveh tehnologijah: Spring Web MVC ter Spring Web Flow. Za templating potrebe je uporabljeno Thymeleaf programsko ogrodje. Kot varnostno programsko ogrodje je v uporabi Spring Security Framework.

V naslednjih poglavjih sledi opis posameznih komponent, ki so v uporabi na predstavitvenem nivoju (presentation tier).

Spring Web MVC

Spring Web MVC je programsko ogrodje, ki omogoča razvoj model-pogled-kontroler aplikacij, v bodoče označeno kot MVC (Model View Controller).

Prednost tovrstnega ogrodja je, da ločuje razvoj predstavitvenega sloja na tri nivoje (3-TIER), kjer ima vsak nivo točno določeno vlogo.

Aplikacije, ki so grajene s pomočjo MVC programskega ogrodja, posledično omogočajo veliko lažji nadaljni razvoj ter lažje vzdrževanje.

Pregled Nivojev MVC

1. Nivo Pogleda (View Tier)
skrbi za prezentacijo modela. Model je predpripravljen v kontroler nivoju. Tako dosežemo, da se nivo pogleda ukvarja izključno s prezentacijo modela.
2. Nivo Kontrolerja (Controller Tier)
Glavna naloga kontrolerja je koordinacija komunikacije med posameznimi nivoji. To vsebuje procesiranje uporabnikovih vnosov, validacijo, gradnjo modela za uporabo v nivoju pogleda ter komunikacijo z domenskim slojem.
3. Nivo Modela (Model Tier)
hrani vsebino podatkov, potrebnih v nivoju pogleda.

V nadaljevanju si bomo podrobneje ogledali vsakega od treh nivojev.

Nivo Pogleda (View Tier)

Na nivoju pogleda je uporabljena tehnologija Thymeleaf. Thymeleaf omogoča čisto izvirno HTML kodo, tako da z najmanj spremembami lahko privedemo statični prototip (HTML) do končne aplikacije (dinamični HTML). Thymeleaf koda je popolnoma čitljiva s strani HTML razvijalca ter vsebuje izključno standardne HTML elemente. Na tak način je podprta popolna razdvojenost nivoja pogleda.

Izvorna koda

Izvorna koda nivoja pogleda se nahaja na lokaciji:
CODE_ROOT/src/main/webapp/WEB-INF/pages

Statični viri (resources) kot so JavaScript, CSS in slike se nahajajo na lokaciji:
CODE_ROOT/src/main/webapp/WEB-INF/resources

Opomba: Ločitev statičnih virov omogoča optimizacijo, saj jih lahko ločimo od dinamičnih virov tako, da jih namestimo na glavni spletni strežnik (Apache) ter razbremenimo aplikacijski strežnik.

Zaradi večje souporabe izvirne kode, nam Thymeleaf omogoča t.i. fragmente, ki se lahko uporabijo na več mestih. Fragmenti so nedvoumno definirani izseki izvirne kode.

Če pogledamo datoteko:

```
.../pages/index.html
```

bomo zelo hitro opazili uporabo fragmentov. Na primer, naslednji kos kode:

```
<head th:replace="static/layout.html :: head"/>
```

pomeni uvoz fragmenta z imenom "head", ki se nahaja v datoteki static/layout.html v trenutno datoteko.

Osnovni fragmenti za strani podportala so definirani v datoteki:

```
.../pages/static/layout.html
```

Primer definicije fragmenta ("head"):

```
<head th:fragment="head">
```

```
...
```

```
</head>
```

Dodatna literatura

Seveda nam tehnologija Thymeleaf omogoča veliko več kot le fragmente. Ker to presega obseg tega priročnika, sledijo viri, kjer si lahko podrobneje pogledate celotno funkcionalnost Thymeleaf tehnologije:

1. Thymeleaf domača stran:

<http://www.thymeleaf.org/>

2. Uporaba Thymeleaf:

<http://www.thymeleaf.org/doc/html/Using-Thymeleaf.html>

3. Uporaba Thymeleaf v Spring Framework okolju:

<http://www.thymeleaf.org/doc/html/Thymeleaf-Spring3.html>

Nivo Kontrolerja (Controller Tier)

Na nivoju kontrolerja sta uporabljeni dve tehnologiji, Spring Web MVC, ki služi za predele podportala, ki ne vsebujejo uporabniških vnosov, ter Spring Web Flow, ki prevzame kontrolo pri predelih podportala, kjer se uporabljajo uporabniški vnosi, ki potrebujejo korake ter validacijo uporabniških vnosov.

Spring Web MVC

Glavna konfiguracija MVC kontrolerja se nahaja v naslednji datoteki:

CODE_ROOT/src/main/webapp/WEB-INF/config/webmvc-config.xml

Izvorna koda

VS-podportal ogrodje uporablja MVC kontroler z imenom MainController. Izvorna koda MainController se nahaja v naslednji datoteki:

CODE_ROOT/src/main/java/si/rettro/vs/MainController.java

MainController služi kot kontroler za vstopno stran (index.html).

Spring Web Flow

Glavna konfiguracija Spring Web Flow se nahaja v naslednji datoteki:

CODE_ROOT/src/main/webapp/WEB-INF/config/webflow-config.xml

Spring Web Flow kontroler prevzame vlogo, ko pričnemo s postopkom vnosa novega predloga. Torej ko nas aplikacija naslovi na naslednji URL:

<http://localhost:8080/vs-podportal/postopek1>

bo Spring Web Flow kontroler prevzel kontrolo nad aplikacijo.

Spring Web Flow kontroler programiramo preko konfiguracije postopkov (web flow) v *-flow.xml datotekah.

Postopek je razdeljen na tri korake: vnos predlagateljev, vnos priponk ter predogled. Web Flow konfiguracija vsebuje glavni postopek, ki je prav tako razdeljen na tri pod-postopke (subflow).

Konfiguracija postopka vnosa predloga (web flow) ter ustrezna HTML izvorna koda se nahaja v naslednjih datotekah:

1. Glavni postopek (main flow):

CODE_ROOT/src/main/webapp/WEB-INF/pages/postopek1/postopek1-flow.xml

2. Postopek vnosa predlagateljev (subflow):

CODE_ROOT/src/main/webapp/WEB-INF/pages/postopek1/predlagatelji-flow.xml

CODE_ROOT/src/main/webapp/WEB-INF/pages/postopek1/step1.html

3. Postopek vnosa priponk (subflow):

CODE_ROOT/src/main/webapp/WEB-INF/pages/postopek1/priponke-flow.xml

CODE_ROOT/src/main/webapp/WEB-INF/pages/postopek1/step2.html

4. Postopek predogleda (subflow):

CODE_ROOT/src/main/webapp/WEB-INF/pages/postopek1/predogled-flow.xml

CODE_ROOT/src/main/webapp/WEB-INF/pages/postopek1/step3.html

Dodatna literatura

Razlaga celotne funkcionalnosti Spring Framework, Spring Web MVC ter Spring Web Flow presega obseg tega priročnika.

Sledijo viri, ki vsebujejo celotno dokumentacijo prej navedenih ogrodij:

1. Spring Framework dokumentacija:
<http://docs.spring.io/spring/docs/4.0.0.RELEASE/spring-framework-reference/html/>
2. Spring Web MVC dokumentacija:
<http://docs.spring.io/spring/docs/4.0.0.RELEASE/spring-framework-reference/html/mvc.html>
3. Spring Web Flow dokumentacija:
<http://docs.spring.io/spring-webflow/docs/2.3.x/reference/htmlsingle/spring-webflow-reference.html>

Nivo Modela (Model Tier)

Nivo modela vsebuje podatke, ki se pripravljajo na nivoju kontrolerja (Controller Tier) ter uporabljajo na nivoju pogleda (View Tier).

Spring Web MVC

V primeru Spring Web MVC, se izgradnja in podajanje modela nahaja v MainController razredu.

Spring Web Form

V primeru Spring Web Form, se izgradnja in podajanje modela nahaja v konfiguracijah postopkov (*-flow.xml).