Република Србија
Агенција за привредне регистре

A Бранкова 25
11000 Београд (РС)

T +381 11 333 1 488
W www.apr.gov.rs

Услуге унапређења и развоја система Е-регистрација у Регистру привредних субјеката који води Агенција за привредне регистре

Конкурсна документација ЈНОП 14/11-18 – Прилог 1
ТЕХНИЧКЕ КАРАКТЕРИСТИКЕ (СПЕЦИФИКАЦИЈЕ)

Београд
новембар 2018. године

КОРИШЋЕНИ ТЕРМИНИ

Naziv	Opis
SOA	Service Oriented Architecture
MVP	Minimum Viable Product
АПР	Агенција за привредне регистре Републике Србије
РС	Република Србија

Садржај

Услуге унапређења и развоја система Е-регистрација у Регистру привредних субјеката

СВРХА

У складу са стратегијом развоја електронске управе у Републици Србији модернизовати постојећи начин рада Регистра привредних субјеката у АПР, успостављањем новог информатичког решење. Ново решење треба да унапреди постојеће процедуре рада, смањи број процедура које се спроводе у писаној форми и омогући њихово потпуно спровођење у електронској форми. Осим тога ново решење треба да омогући бољу комуникацију између државних органа и смањи број административних процедура на страни корисника.

ЦИЉ

Генерални циљ пројекта је реинжењеринг постојећег информационог система за спровођење свих пословних процеса у Регистру привредних субјеката (привредна друштва и предузетници), израда дизајна новог решења које ће омогућити пуну електронску регистрацију, на минимално функционалном производу верификовати предложен дизајн и израдити детаљну техничку спецификацију за имплементацију новог информационог система.

Будући информациони систем ће корисницима услуга АПР омогућити електронско подношење свих типова захтева, праћење статуса обраде захтева, преузимање аката у електронској форми, као и већу и једноставнију доступност података и докумената из регистра, посредством јединственог портала.

Са друге стране будући информациони систем ће запосленим у АПР омогућити ефикасније извршење своје функције, бољу комуникацију са корисницима, транспарентнији рад, већу сигурност, скраћење процедура и смањивање трошкова обраде захтева.

ОСНОВНЕ ИНФОРМАЦИЈЕ

Законски оквир

Регистар привредних субјеката почео је са радом 1. јануара 2005. године, чиме се са регистрације привредних субјеката у судском поступку прешло на административни поступак регистрације, са циљем убрзања поступка и постизања веће ефикасности кроз постојање јединственог, електронског система регистровања података, који су од значаја за привредне субјекте и који су неопходни у свакодневном правном и пословном промету.

Преласком на централизовани систем регистрације добијена је јединствена, централизована, електронска база података о привредним субјектима у Републици Србији, која садржи све податке који су према закону, а у складу са директивама ЕУ предмет регистрације, а која је путем интернета доступна свим заинтересованим лицима, без потребе да доказују свој правни интерес, на брз и једноставан начин.

Кључан захтев у реализацији будућег система је његова усклађеност са законским оквиром. У наставку је дат преглед самог законског оквира.

Рад Агенције за привредне регистре је уређене основним нормативним оквиром који чине пре свега *Закон о јавним агенцијама* и *Закона о Агенцији за привредне регистре*.

У односу на саме циљеве пројекта, кључни материјални нормативни оквир регистрације, представљају пре свега *Закон о привредним друштвима*, *Закон о јавним предузећима, Закон о задругама*, као и *Закон о електронском документу, електронској идентификацији и услугама од поверења у електронском пословању*.

Процесни елементи који дефинишу предмет и начин регистрације привредних субјеката пре свега чине *Закон о поступку регистрације у Агенцији за привредне регистре* као основним законом којим се уређује поступак регистрације и подзаконским актима. Такође је у процесном погледу, кључни елементу процесног оквира будућег система електронских пријава регистрације, поред *Закона о општем управном поступку* чине и *Уредба о канцеларијском пословању органа државне управе, Уредба о електронском канцеларијском пословању органа државне управе* и *Упутство о електронском канцеларијском пословању*, и прописи којима се уређује чување архивске грађе. Нови *Закон о електронској управи*, такође даје оквир функционалних захтева у делу поступака и обавеза у делу електронске управе (пријема, документа и нотификација).

Узимајући у обзир природу самог будућег апликативног система, морају се узети у обзир и законске регулативе везане за елементе електронског пословања, пре свега кроз *Закон о електронском документу, електронској идентификацији и услугама од поверења у електронском пословању* (и припадајућа подзаконска акта), конкретно у делу дигиталних потписа и електронског документа као и *Закон о информационој безбедности*, и припадајућа подзаконска акта.

Посебно је у делу прикупљања, чувања и објављивања података потребно узети у обзир *Закон о заштити података о личности*, али и потенцијално ГДПР2 регулативу.

У оквиру законских обавеза, али и дела потребе коришћења јединствених номенклатура општих података, укључујући адресне податке, потребно је у реализацији система узети у обзир *Уредбу о Адресном регистру*.

У наставку је листа која представља законски оквир у оквиру којег требају бити имплементиране функционалности будућег система:

- Закон о јавним агенцијама ("Сл. гласник РС", бр. 18/2005 и 81/2005 - испр.)
- Закон о Агенцији за привредне регистре („Службени гласник РС“, бр. 55/04, 111/09 и 99/11)
- Закон о поступку регистрације у Агенцији за привредне регистре ("Сл. гласник РС", бр. 99/2011 и 83/2014)
- Закон о привредним друштвима („Службени гласник РС”, бр. 36/2011, 99/11, 83/2014 - др. закон, 5/2015 и 44/2018)
- Закон о јавним предузећима („Службени гласник РС”, бр. 15/2016)
- Закон о задругама („Службени гласник РС”, бр. 112/2015)
- Правилник о садржини Регистра привредних субјеката и документацији потребној за регистрацију („Службени гласник РС”, бр. 42/2016)
- Одлука о накнадама за послове регистрације и друге услуге које пружа Агенција за привредне регистре ("Сл. гласник РС", бр. 119/2013, 138/2014, 45/2015, 106/2015, 32/2016 и 60/2016)
- Закон о електронском документу, електронској идентификацији и услугама од поверења у електронском пословању („Службени гласник РС“ бр. 94/2017)
- Правилник о условима које морају да испуњавају квалификовани електронски сертификати („Службени гласник РС“, бр. 34-18)
- Правилник о условима које мора да испуњава квалификовано средство за креирање електронског потписа односно печата и условима које мора да испуњава именовано тело („Службени гласник РС“ бр. 34-18)
- Закон о заштити података о личности („Службени гласник РС“ 97/2008, 104/2009-др.закон, 68/2012 -одлука УС и 107-12)

- Закон о заштити пословне тајне (“Сл. гласник РС”, број 72/2011)
- Закон о информационој безбедности ("Сл. гласник РС", бр. 6/2016 и 94/2017)
- Закон о електронској управи ("Сл. гласник РС", 27/2018)
- Закон о општем управном поступку („Службени гласник РС“, бр. 18/2016)
- Уредба о канцеларијском пословању органа државне управе („Службени гласник РС“, бр. 80/92, 45/2016 и 98/2016)
- Уредба о електронском канцеларијском пословању органа државне управе ("Сл. гласник РС" бр. 40/2010)
- Упутство о електронском канцеларијском пословању (“Сл. гласник РС“, број 102/2010
- Уредба о Адресном регистру („Службени гласник РС“, бр 63/2017)
- Закон о територијалној организацији Републике Србије ("Сл. гласник РС", бр. 129/2007, 8/2016 И 47/2018)
- Уредба (ЕУ) 2016/679 Европског парламента и Савета од 27. априла 2016. и заштити физичких лица у односу на обраду података о личности и о слободном кретању таквих података и о стављању директиве 95/46/ез ван снаге (општа уредба о заштити података) (ГДПР2).

Пословни контекст

Да би на прави начин дефинисали архитектуру софтверског решења за подршку раду регистара неопходно је да разумемо пословни контекст рада регистра.

Регистар је законом дефинисана јединствена, централна, електронска база података и докумената на основу којих је извршена регистрација.

Различити регистри успостављени су и регулисани различитим законским и подзаконскима актима која су под ингеренцијом различитих министарстава.

Радом регистра управља регистратор који има овлашћења и обавезе да:

1. се стара о законитом, систематичном и ажурном вођењу регистра;
2. ближе одређује начин вођења регистра;
3. доноси одлуке о упису у регистар, односно врши регистрацију података;
4. прописује стандардне обрасце регистрационих пријава и захтева у поступку вођења регистра;
5. предузима друге радње неопходне за несметано и правилно функционисање регистра у складу са законом;

Регистратор је независан у раду, у оквиру овлашћења утврђених законом и другим прописима.

Специфичним законским и подзаконскима актима дефинисан је предмет регистрације за сваки појединачни регистар док је процес регистрације делимично дефинисан законом о регистрацији.

Поступак регистрације покреће се подношењем пријаве.

По пријему пријаве регистратор проверава да ли су испуњени услови за регистрацију, и то:

1. да ли је надлежан за поступање по пријави;
2. да ли је пријаву поднело овлашћено лице;
3. да ли је податак или документ предмет регистрације;
4. да ли је податак или документ чија се регистрација захтева већ регистрован;
5. да ли пријава садржи податке и чињенице потребне за регистрацију;
6. да ли су уз пријаву приложени прописани документи;
7. да ли су чињенице из пријаве сагласне чињеницама из докумената приложених уз пријаву и подацима који су регистровани у регистру који поступа по пријави;

8. да ли је у регистру који поступа по пријави под истим називом већ регистровано друго правно лице или предузетник или је већ поднета пријава за регистрацију под истим називом или је назив већ резервисан у складу са овим законом, односно да ли је назив одређен у складу са законом;
9. да ли је уз пријаву приложен доказ о уплати накнаде за вођење поступка регистрације.

Специфичности, ризици, ограничења и изазови

Агенција на једном месту води више регистара који међусобно деле доста сличности и функционалности, међутим сваки регистар поседује и своје специфичности.

Архитектура софтверског решења за подршку рада регистра мора узети у обзир све релеватне специфицности, ризике, ограничења и изазове.

Грануларност и ниво апстрактности пословних правила дефинисаних законским и подзаконским актима је на много већем нивоу од нивоа грануларности и апстрактности потребне за спровођење конкретних регистрационих поступака у пракси. Велики број пословних правила практично се спроводи од стране лица заспослених на пословима обраде захтева у појединачним регистрима на основу интерне праксе, процедура и смерница која су врло често недовољно формализоване. Велики број и разноликост ових правила доводе у питању могућност њихове потпуне аутоматизације.

Аутоматизација ових правила смањује флексибилност и повећава комплексност софтверског решења, док на другој страни повећава ниво конзистентности у процесу обраде захтева смањујући утицај запослених лица на доношење и спровођење одлука. Смањење аутоматизације повећава флексибилност и смањује комплексност система, али на другој страни доводи до потенцијалне неконзистентности у процесу обраде захтева у зависности од запосленог лица које у овом случају има већи утицај на доношење и спровођење одлука. Проналажење праве мере у погледу аутоматизације ових правила у односу на комплексност решења велики је изазов.

Тренд модернизација и подизања нивоа ефикасности јавне управе веома често доводи до потребе за разменом података, интерно између регистара и екстерно са другим институцијама. Поред размене података, регистри неретко учествују и у интеграционим процесима како интерно у оквиру АПР, тако и екстрено са другим институцијама. Могућност једноставне размене података и имплементације интеграционих процеса велики је изазов коме треба посветити посебну пажњу.

Могућност брзе и једноставне измене и унапређење система услед промена законских и подзаконских аката је још један од изазова. Могућност брзог, једноставног и економичног успостављања нових регистара такође је један од изазова. Смањење трошкова експлоатације и одржавања је још један од изазова јер АПР ове трошкове финансира из сопствених средстава која су ограничена.

Подаци који се налазе у регистрима АПР од велике су важности како за државу тако и за привреду и грађање због чега је обезбеђење адекватне доступности и сигурности података још један је од изазова.

Континуитет рада регистара је императив, поготово када се узме у обзир чињеница да по закону сви захтеви морају бити обрађени у року од 5 дана.

Обезбеђење перформантности софтверског решења услед повећања обима посла је још један од изазова.

Дугорочна одрживост система је веома битна поготово у погледу стручних кадрова неопходних за имплеметацију, одржавање и унапређење система. Због ограничења зарада, АПР нема могућност да запошљава довољан број високо квалификованих стручњака. Пружаоци ИТ услуга су у сличном проблему, искусни и квалитетни стручњаци су дефицитарни на тржисту, док су

постојећи капацитети углавном ангажовани на уноснијим пројектима. Ове чињенице представљају велико ограничење и озбиљан ризик и изазов.

Једноставност коришћења и лаког учења софтверског решења је императив и представља још један од изазова.

ЗАХТЕВИ У ПОГЛЕДУ АРХИТЕКТУРЕ И ДИЗАЈНА

Архитектурни принципи

1. Основне функционалности регистра које садрже специфичне функционалности регистра треба изоловати у издвојене и независне софтверске компоненте;
2. Заједничке функционалности које не садрже специфичне функционалности и логику регистра треба изоловати у издвојене софтверске компоменте које би користили сви регистри;
3. Интеграциону логику треба изоловати у две засебне интеграционе компоненте, компоненту за размену порука и компоненту за оркестрацију процеса;
4. Компонента за размену порука треба да буде јединствена на нивоу АПР;
5. Компоненту за оркестрацију процеса мора обезбедити могућност миграције инстанци процеса на нове верзије процеса;
6. Компонента за оркестрацију процеса мора имати могућност претплате на поруке које се размењују преко компоненте за размену порука;
7. Сваки регистар је одговоран за сопствене интеграционе процесе и у логичком смислу треба да поседује независну компоменту за оркестрацију регистарских процеса;
8. Све компоненте морају релеватне пословне догађаје учинити доступним осталим компонентама система као и спољњим системима преко компоненте за размену порука;
9. Процеси различитих регистара треба да се интегришу искључиво разменом порука преко јединствене компонента за размену порука;
10. Све компоненте морају своје функционалности изложити преко стандардизованих API сервиса који у потпуности морају бити документовани;
11. Архитектура специфичних регистарских компоненти треба да буде базирана на конзистентној референтној архитектури;
12. Доменски модели и модели података специфичних регистарских компоненти морају међусобно бити конзистентни;
13. Доменски модели и модели података специфичних регистарских компоненти морају бити у складу са доменским језиком специфичног регистра;
14. Конзистентност података различитих пословних домена треба обезбедити разменом порука и интеграционим процесима;
15. Складишта података софтверских компоненти морају бити оптимизована за спровођење пословних трансакција. За аналитичке потребе и потребе извештавања треба формирати засебна складишта података оптимизована за те намене са одговарајућим моделом података;
16. Интеграцију са екстерним системима односно доменима искључиво треба имплементрати преко специфичне интеграционе компоненте која неће дозволити преливања пословне логике и детаља техничке имплементације;
17. Архитектура специфичне регистарске компоненте за обраду захтева мора бити у складу са CLEAN архитектуром;
18. Све софтверске компоненте морају бити скалабилне;
19. Све софтверске компоненте морају бити високо доступне;
20. Све софтверске компоненте морају бити безбедне;
21. Све софтверске комепоненте морају користити постојеће централизовано решење за ауторизација и аутентификацију корисника;
22. Све софтверске компоненте морају имати имплементиран детаљан записник грешака и апликативних догађаја;

23. Све компоненте морају бити усклађене са релеватним законима и прописима;
24. Све софтверске компоненте морају имати скрипте за аутоматизовано функционално тестирање преко сервиснога слоја;
25. Све софтверске компоненте морају имати скрипт за аутоматизовано постављање у рад;
26. Све софтверске компоненте морају бити документоване;
27. Све софтверске компоненте намењене корисницима морају бити једноставне и интуитивне за коришћење и лаке за учење;
28. Све софтверске компоненте намењене корисницима морају подржавати рад у вишејезичном окружењу;
29. Решење мора поседовати online систем помоћи и корисничку упутство.

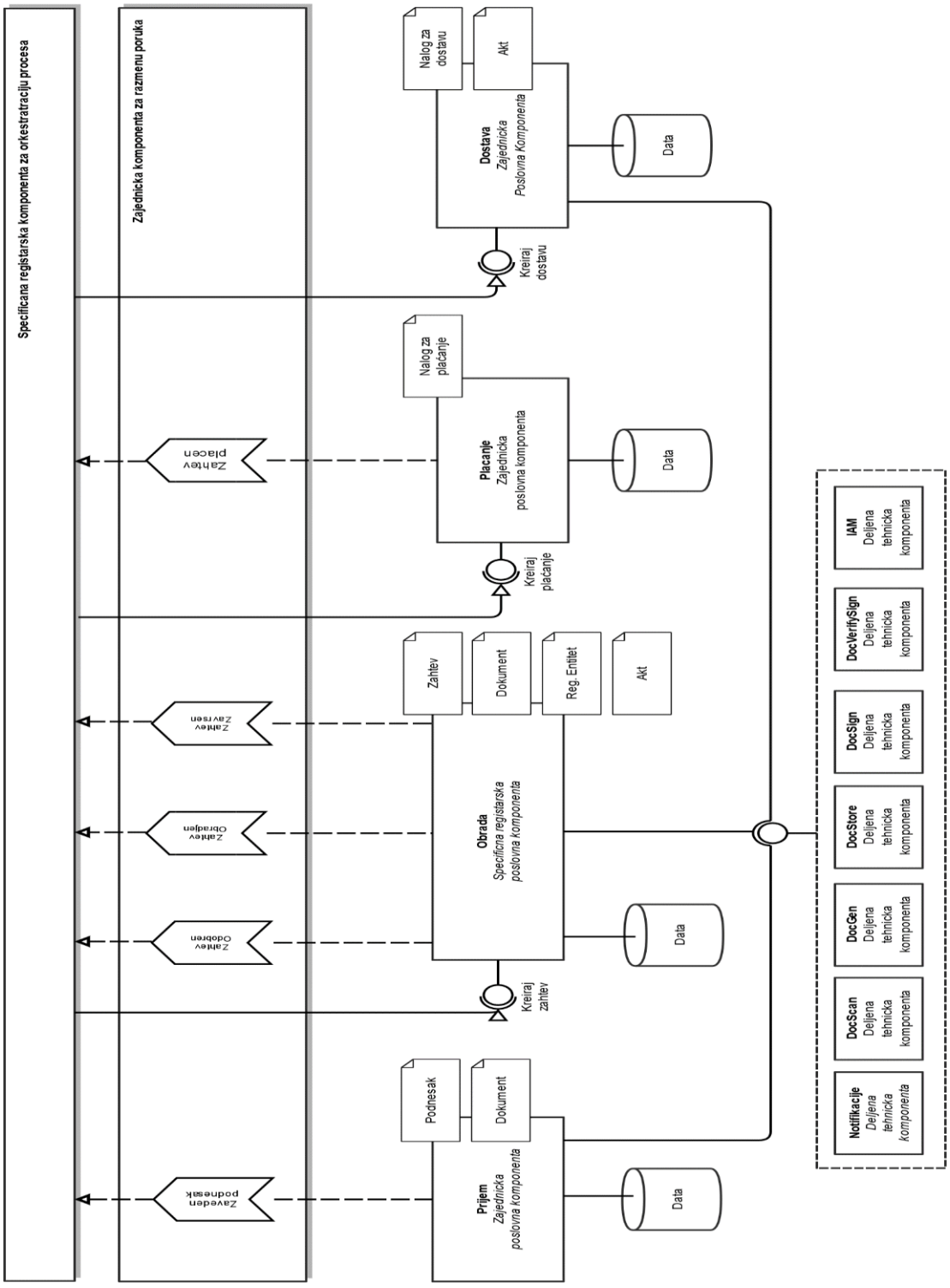
Опис адекватног архитектурног модела

Израда засебног софтверског решења за сваки појединачни регистар је предуг и прескуп приступ. Израда јединствене платформе за подршку регистрацији која би се користила за све регистре превише је комплексан приступ за имплементацију и одржавање. Адекватно архитектурно решење је негде између ова два приступа.

1. Архитектурно решење је базирано на сервисној архиктектури (SOA) и архитектури базираној на размени порука односно догађаја;
2. Систем је декомпонован на више пословних домена са јасно дефинисаним одговорностима;
3. Регистар је базиран на више пословних домена са јасно дефинисаним одговорностима од којих су неки пословни домени заједнички за све регистре;
4. Сваки пословни домен има одговорајући доменски модел података који јасно и недвосмислено моделује одговарајуће концепте из пословног домена и њихове међусобне релације;
5. Сваки пословни домен има одговорајући процесни модел који моделује пословне процесе из одговарајућег пословног домена;
6. Пословни домени се састоје из једне или више компоненти;
7. Поред компоненти које припадају различитим пословним доменима, систем садржи и низ дељених техничких компоненти чије се функционалности могу користити у различитим пословним доменима;
8. Компоненте су базиране на сервисној архитектури;
9. Комуникација између различитих компоненти остварује се оркестрацијом процеса и разменом порука.

Специфичности једног регистра смештене су у специфичан регистарски домен и имплементиране у три кључне компоненте:

1. Специфичан регистарски оркестратор процеса;
2. Специфична регистарска компонента за обраду захтева;
3. Специфична регистарска компонента за јавну претрагу регистрских података.



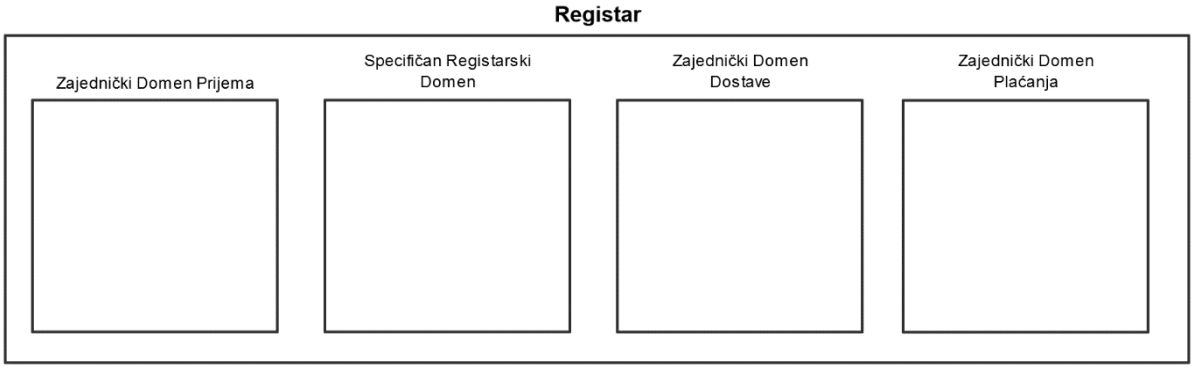
Пословни домени

Регистар је базиран на више пословних домена са јасно дефинисаним одговорностима од којих су неки пословни домени заједнички за све регистре.

Заједнички домени:

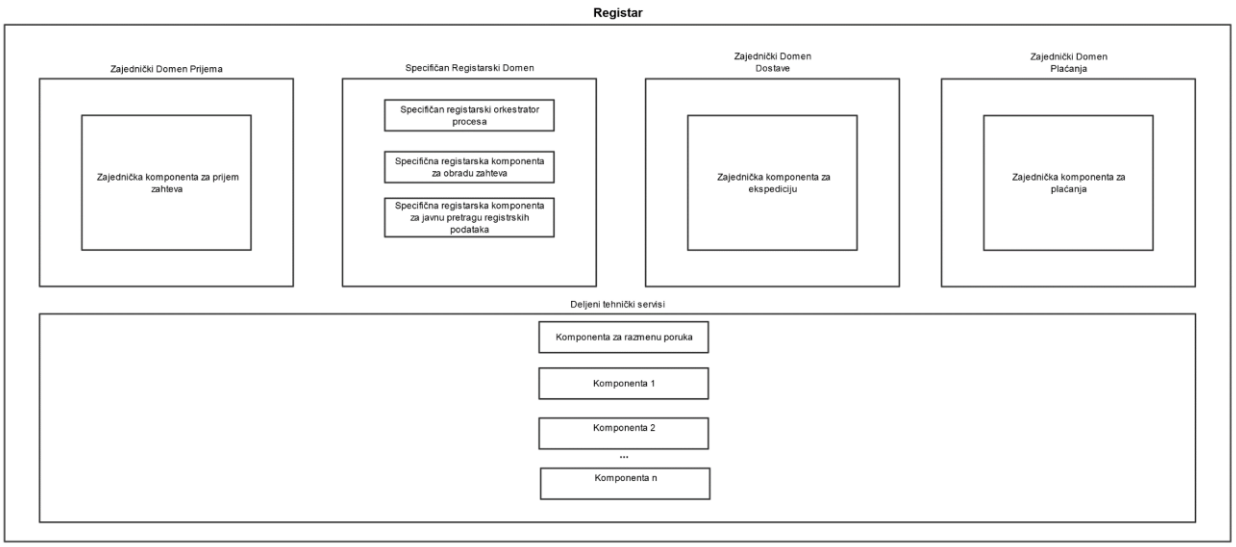
1. Пријем - Домен одговоран за пријем поднесака

2. Плаћања - Домен одговоран за евиденцију и управљање налозима за плаћање и уплатама
3. Достава - Домен одговоран за доставу и слање излазних докумената из регистарских процеса



Компоненте система

Функције пословних домена имплементиране су преко сервисних компоненти које припадају тим доменима. Најчешће је један пословни домен имплементиран кроз једну сервисну компоненту, али то не мора бити строго правило. Пословне компоненте као и пословни домени могу се поделити на заједничке и специфичне.



Заједничке пословне компоненте су:

1. Заједничка компонента за пријем захтева
2. Заједничка компонента за плаћања
3. Заједничка компонента за доставу

Специфичне компоненте регистра су:

1. Специфичан регистарски оркестратор процеса
2. Специфична регистарска компонента за обраду захтева
3. Специфична регистарска компонента за јавну претрагу регистрских података

Појединачне компоненте су на нивоу података базиране на доменским моделима који јасно и недвосмислено моделују одговарајуће концепте из пословног домена којем компоненте припадају.

Поред пословних компоненти, систем садржи и низ дељених техничких компоненти чије се функционалности могу користити у различитим пословним доменима.

Дељене техничке компоненте имају јасно дефинисане функције и одоговорности и ослобођене су сваког вида пословне, односно доменске логике.

Дељене техничке компоненте:

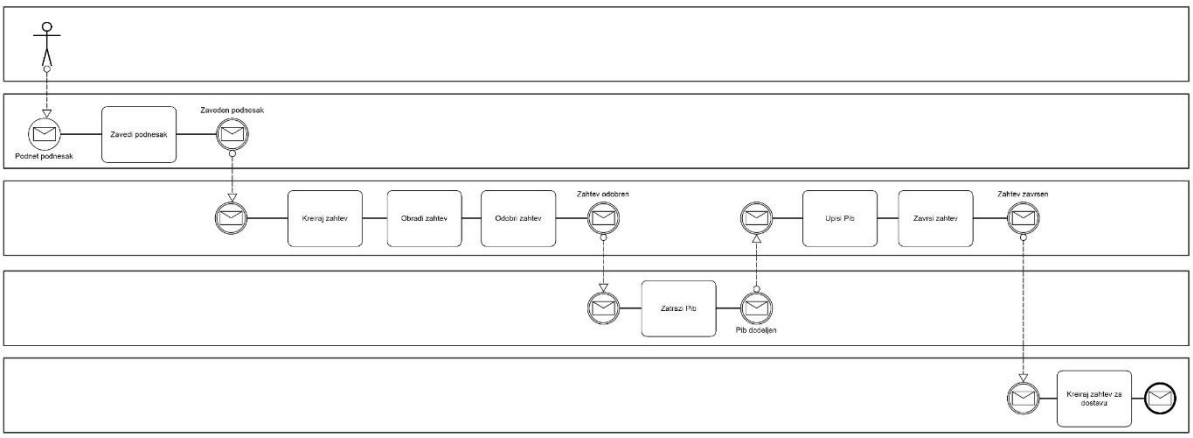
1. Нотификације
2. Скенирање Докумената
3. Генерисање Докумената
4. Складистење Докумената
5. Електронско Потписивање Докумената
6. Провера Електронски Потписаних Докумената
7. Компонента за размену порука

Оркестрација процеса и размена порука

Комуникација између сервисних компоненти у различитим пословним доменима остварује се оркестрацијом процеса и разменом порука. Процеси обраде захтева припадају неком од регистара и никада не оркестрирају операције у другим регистрима нити другим пословним доменима. Комуникација између процеса различитих регистара или процеса различитих пословних домена увек се остварује разменом порука. На овај начин границе и одоговрности процеса јасно су дефинисане. Да би се овај концепт у потпуности разумео најбоље га је сагледати на конкретним примерима.

Пример 1

Предпоставимо да постоји пословни захтев да се у Регистар привредних субјеката, аутоматски по службеној дужности упише податак о ПИБ-у који додељује други државни орган (Пореска управа) када се у Регистру привредних субјеката оснује нов привредни субјекат. Ова активност је део пословног процеса One Stop Shop (OSS) који је у АПР успостављен пре више година и који обухвата активности које се спроводе у три државне институције (АПР, Пореска управа и ЦРОСО).



Решење да се конзистентност података између регистара обезбеди некаквом синхорнизацијом на нивоу података између две институције изгледа као једноставно решење, али је оно потпуно неприхватљиво из разлога надлежности и информационе безбедности.

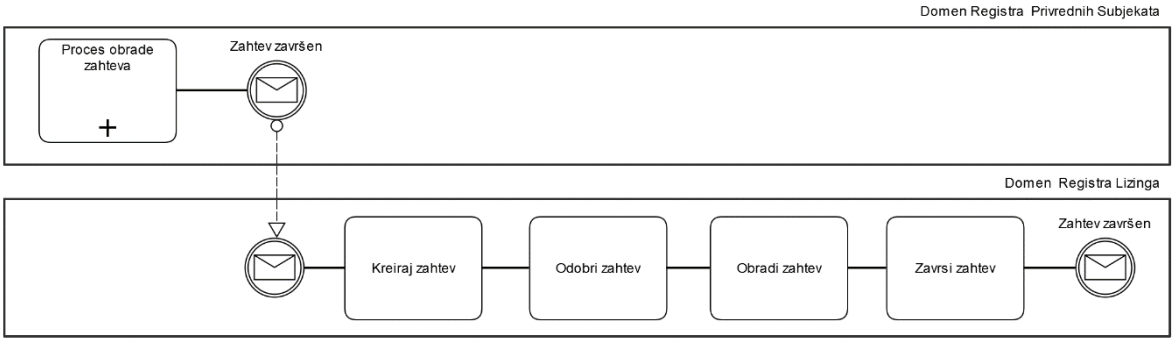
Решење да се у логику Регистра привредних субјеката имплементира додатна логика промене података у другом државном органу потпуно је неприхватљива са становишта софтверске архитектуре јер се мешају одговорности компоненти. Поред тога долази до непотребног спрезања Регистра привредних субјеката и информационог система другог државног органа. Замисимо како би на крају изгледала Регистарска компонента Регистра привредних субјеката ако би сличан захтев проследили и други државни органи.

Право решење је управо базирано на горе описаном концепту. Регистар привредних субјеката приликом уписа новог привредног субјекта дистрибуира одговарајућу поруку о томе преко заједничке компоненте за размену порука. Оркестратор процеса система OSS претплаћен је преко заједничке компоненте за размену порука на ову врсту порука. Када се порука о промени појави, оркестратор процеса OSS покреће један или више процеса промене података по службеној дужности који аутоматски мењају податке у одговарајућим регистрима других државних органа. На овај начин одговорности регистара и државних органа су јасно подељене, нема мешања одговорности нити непотребног спрезања.

Регистар привредних субјеката не зна ништа о пословним правилима и процесима у другом државном органу нити други државни орган зна било какве детаље о пословним правилима и процесима у Регистру привредних субјеката. На идентичан начин је могуће исти захтев имплементирати и у другим регистрима при чему у Регистру привредних субјеката не би било никаквих измена. Број заинтересованих компоненти које би биле претплаћене на специфичан пословни догађај из Регистра привредних субјеката ни на који начин не би утицао на логику Регистра привредних субјеката.

Пример 2

Предпоставимо да постоји пословни захтев да се у Регистру лизинга аутоматски по службеној дужности ажурирају подаци о пословном имену примаоца лизинга у свим регистрованим уговорима када у Регистру привредних субјеката дође до промене пословног имена привредног субјекта који има регистрован уговор о лизингу.



Решење да се конзистентност података између регистара обезбеди некаквом синхорнизацијом на нивоу података изгледа као једноставно решење, али је оно потпуно неприхватљиво јер би у Регистру лизинга дошло до промене података без икаквог трага. Решење да се у логику Регистра привредних субјеката имплементира додатна логика промене података у Регистру лизинга потпуно је неприхватљива са становишта софтверске архитектуре јер се мешају одговорности компоненти. Поред тога долази до непотребног спрезања Регистра привредних субјеката и Регистра лизинга. Замисимо како би на крају изгледала Регистарска компонента Регистра привредних субјеката ако би сличан захтев проследили и други регистри АПР.

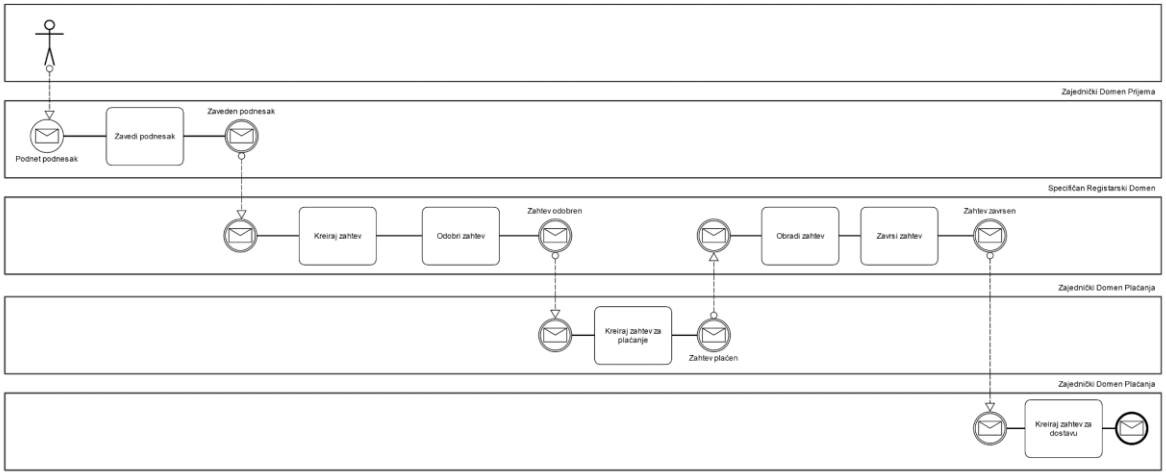
Право решење је управо базирано на горе описаном концепту. Регистар привредних субјеката приликом промене пословног имена дистрибуира одговарајућу поруку о томе преко заједничке компоненте за размену порука. Оркестратор процеса Регистра лизинга претплаћен је преко заједничке компоненте за размену порука на ову врсту порука. Када се порука о промени појави, оркестратор процеса Регистра лизинга покреће један или више процеса промене података по

службеној дужности који аутоматски мењају податке у одговарајућим регистрованим уговорима о лизингу.

На овај начин одговорности регистара су јасно подељене, нема мешања одговорности нити непотребног спрезања. Регистар привредних субјеката не зна ништа о пословним правилима и процесима у Регистру лизинга нити Регистар лизинга зна било какве детаље о пословним правилима и процесима у Регистру привредних субјеката. На идентичан начин је могуће исти захтев имплементирати и у другим регистрима при чему у Регистру привредних субјеката не би било никаквих измена. Број заинтересованих компоненти које би биле претплаћене на специфичан пословни догађај из Регистра привредних субјеката ни на који начин не би утицао на логику Регистра привредних субјеката.

Пример 3

Додатан пример може бити процес комплетне обраде једног захтева од пријема до доставе решења који се одвија у у оквиру једног регистра.



Након пријем и завођења поднеска компонента, која припада заједничком пословном домену пријема, дистрибуира одговарајућу поруку преко заједничке компоненте за размену порука. За ову специфичну поруку заинтересовани су сви регистри.

На основу података у поруци одговарајући регистарски оркестратор процеса иницира процес обраде захтева у надлежном регистру. Оркестратор процеса преко сервисног слоја издаје специфичној регистарској компоненти наредбу за креирање новог захтева на основу података из поднеска.

Процес се даље одвија у специфичној регистарској компоненти где запослени на пословнима обраде предмета доносе одлуке и извршавају одређене апликативне акције. Након одобрења специфична регистарска компонента дистрибуира одговарајућу поруку која покреће процес даље при чему оркестратор процеса заједничкој компоненти за плаћање преко сервисног слоја издаје наредбу за креирање захтева за плаћање.

Када заједничка компонента за плаћање региструје да је одговарајући захтев плаћен дистрибуира одговарајућу порука преко заједничке компоненте за размену порука која даље покреће процес у специфичну регистарску компоненту где запослени на пословнима приводе крају процес обраде захтева.

Након завршетка обраде захтева дистрибуира се одговарајућа поруку преко заједничке компоненте за размену порука која даље покреће процес при чему оркестратор процеса заједничкој компоненти за доставу преко сервисног слоја издаје наредбу за креирање захтева за доставу.

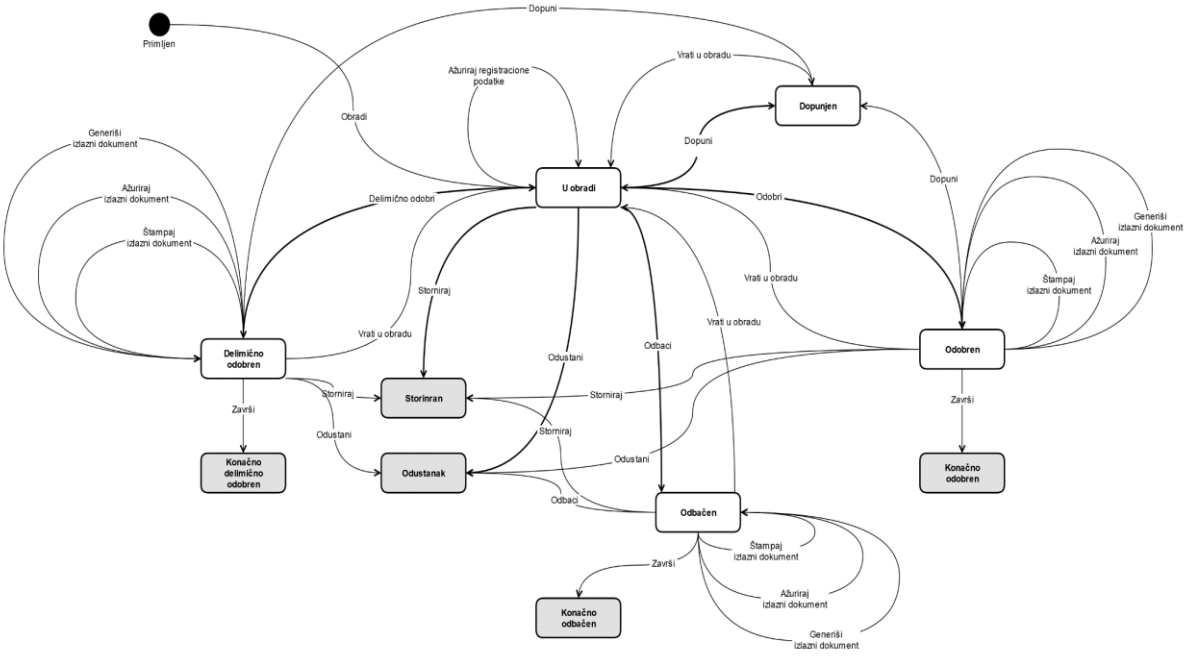
Из наведеног примера јасно се може закључити да су одговорности компоненти из различитих домена јасно подељене, да је избегнуто мешање одговорности и непотребно спрезање. Заједничке компоненте такође могу бити базиране на процесима. Одговорност за имплементацију ових процеса налази се у одговарајућим пословним доменима. У техничком смислу то значи да конкретан заједнички пословни домен мора имати свој процесни оркестратор који је одговоран за оркестрацију ових послова. Недопуистиво је да регистарски оркестратори процеса оркестрирају ове процесе.

Архитектура специфичне регистарске компоненте за обраду захтева

Специфична регистарска компонента за обраду захтева имплементира специфичности појединачног регистра пре свега у поступку обраде захтева. У целокупном процесу обраде захтева обрада захтева је за сваки регистар специфична. Поступак обраде у регистарској компоненти за обраду захтева може се сматрати једним кораком у целокупном процесу обраде захтева.

Основна јединица у процесу обраде је **захтев** који у завиности од врсте захтева може имати различита стања. Поступак обраде захтева састоји се од низа акција које је могуће извршити над захтевом. Могућност извршења појединих акција над захтевом зависи од стања у којем се захтев налази. Поједине акције доводе до промене стања захтева. Нека од дефинисаних стања су коначна и захтев који се налази у овим стањима више не може мењати свој статус. Приликом извршења акција над захтевом, емитују се одговарајуће поруке о насталим акцијама.

Зависност стања и акција дефинисана је машином стања. Машина стања је генерално веома слична за све врсте захтева, али не и потпуно идентична.



Одлуке о редоследу акција над захтевом доноси лице запослено на пословима обраде захтева на основу чињеница до којих долази увидом у предмет захтева и пратећу документацију. Пословна правила се у највећој мери спроводе од стране лица заспослених на пословима обраде захтева на основу интерне праксе, процедура и смерница која су врло често недовољно формализована. Велики број и разноликост ових правила доводе у питању могућност њихове потпуне формализације и аутоматизације.

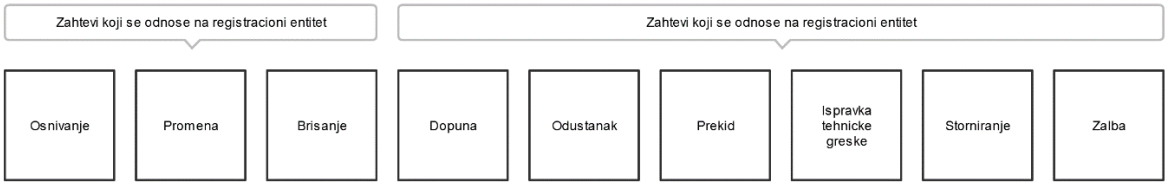
Аутоматизација ових правила смањује флексибилност и повећава комплексност система, док на другој страни повећава ниво конзистентности у процесу обраде захтева смањујући утицај запослених лица на доношење и спровођење одлука. Смањење аутоматизације повећава флаксибилност и смањује комплексност система, али на другој страни доводи до потенцијалне неконзистентности у процесу обраде захтева у зависности од запосленог лица које у овом случају има већи утицај на доношење и спроводјење одлука.

Питање архитектуре увек је питање компромиса између бенефита и недостатака. Смањење комплексности и повећање флексибилности по цену потенцијалне неконзистентности у процесу обраде захтева иницијално је мање ризичан концепт који је једноставнији за имплементацију. Дугорочно овај концепт је бољи избор у погледу флексибилности, једноставности и трошкова одржавања.

Поред лица запосленог на пословима обраде захтева акције над захтевом може преко сервисног интерфејса извршавати и регистарски оркестратор процеса.

Захтеви се могу поделити у две групе:

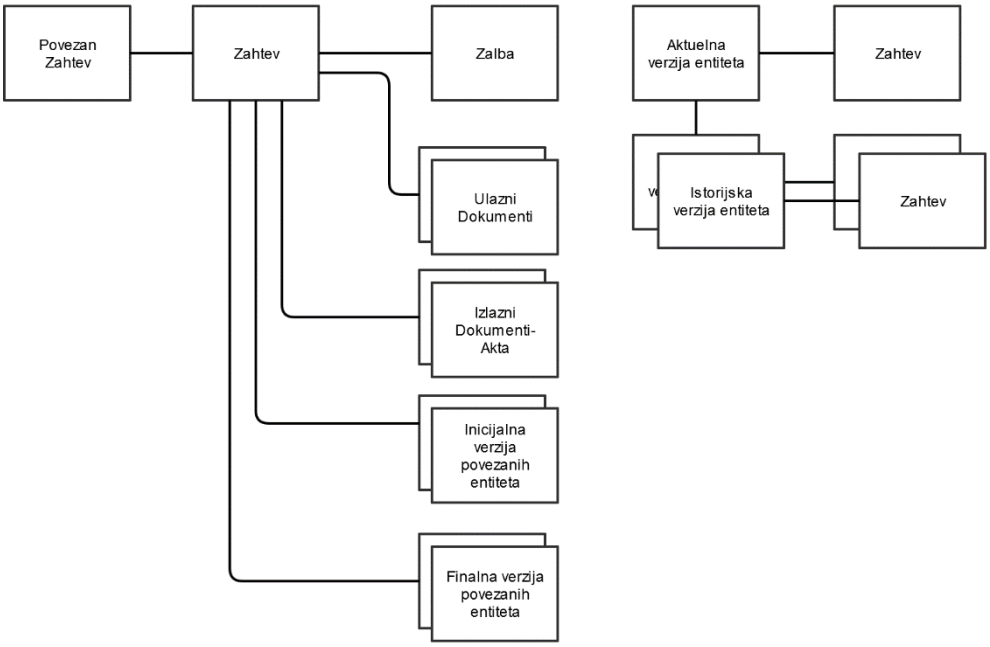
1. Захтеви који се односе на регистрациони ентитет
2. Захтеви који се односе на постојеће захтеве



Захтеви који се односе на регистрациони ентитет, чија обрада доводи до уписа, промене или брисања података једног или више регистрационих ентитета који су предмет регистрације у регистру.

Захтеви који се односе на постојеће захтеве су захтеви чија обрада у техничком смислу доводи до промене повезаног захтева. На пример ово могу бити захтеви за допуну, сторнирање, прекид или одустанак од постојећег захтева.

Приликом промене регистрационих ентитета чува се историјска верзија регистрационог ентитета која је једнозначно повезана са захтевом који је довео до промене регистрационог ентитета.



Као један од резултата обраде захтева у регистру, најчешће се генерише излазни документ Акт. Садржај и начин генерисања излазног документа зависи од врсте захтева и врсте исхода обраде захтева.

Захтеви иницијлано могу бити подносиоцу достављени у папирном и електронском облику. Имплементација специфичне регистарске компоненте у потпуности мора бити индиферентна у односу на начин иницијалне доставе што се постиже процесом обраде базираним на документима.

Папирни документи се скенирају у електронску форму док се електронски директно прослеђују специфичној регистарској компоненти. На овај начин постигнута је поменута индиферентност.

Софтверска Архитектура

Поред тога што регистри деле компоненте из заједничких домена, специфичне регистарске компоненте за обраду захтева треба да деле и заједничку архитектиру.

Предложена архитектура базирана је на архитектури познатијој као “Clean Architecture”. У питању је вишеслојна архитектура где сваки слој има јасно дефинисану одговорност.

У слоју апликативне логике налазе се доменски ентитети, доменски догађаји, имплементација случајева коришћења (Use Case) и интерфејси различитих инфраструктурних сервиса чије се конкретне имплементације налазе у слоју инфраструктурних сервиса.

У слоју инфраструктурних сервиса налазе се имплементације сервиса за приступ различитим инфраструктурним компонентама. Типичан пример инфраструктурних сервиса су репозиторијуми за приступ и перзистенцију података. Поред имплементације репозиторијума за приступ и перзистенцију података могу се наћи сервиси за логовање, слање порука и сл.

Слој за кориснички интерфејс имплементира механизме којима корисници могу приступати апликативној логици имплементираној у слоју аплиактивне логике. Апликације могу имплементирати различите механизме приступа. Механизми приступ могу бити web апликације, soap сервиси, json rpc сервиси, десктоп апликација, конзолна апликација, мобилне апликације и сл. У зависности од типа механизма за приступ овај слој садржи различите ентитете. У случају web апликације базиране на MVC архитектури ентитети који припадају овом слоју су контролери, view модели и view-ови.

Конкретно у случају специфичне регистарске компоненте предвиђено је да апликација има два механизма приступа, један реализован као web апликација намењена корисницима и други релазован као сервисни API намењен интеграцијама са другим компонентама система.

Референце везане за “Clean Architecture”:
https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html
https://jeffreypalermo.com/2008/07/the-onion-architecture-part-1/
https://www.youtube.com/watch?v=Nsjsiz2A9mg&t=2540s

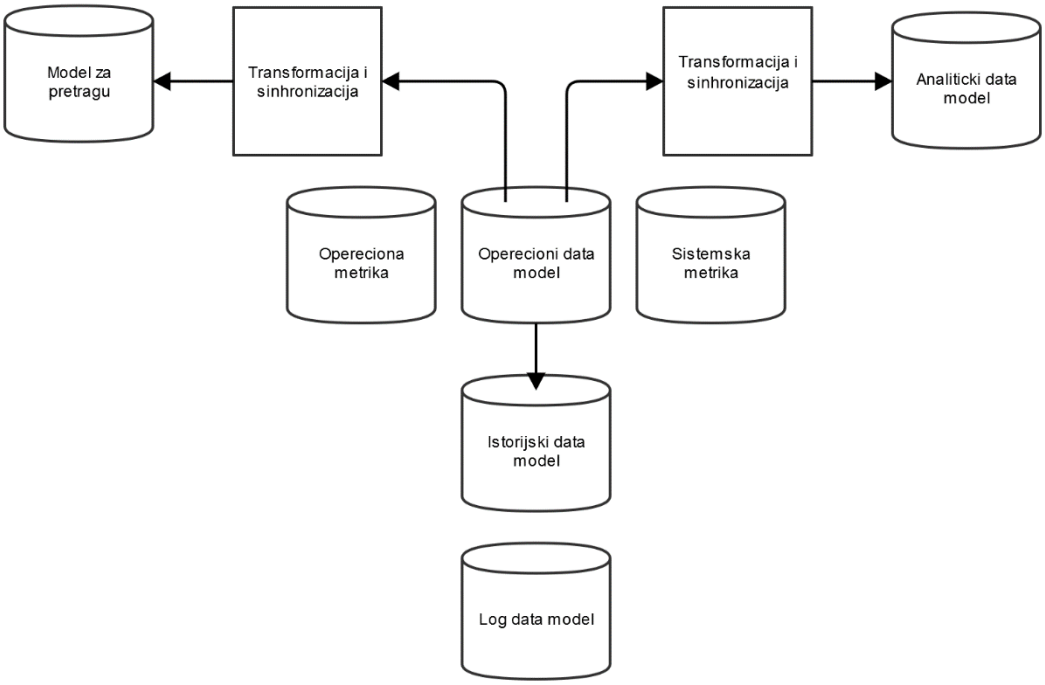
Примери имплементације “Clean Architecture”:
https://github.com/ivanpaulovich/clean-architecture-manga
https://github.com/ardalis/CleanArchitecturež

Архитектура података - управљање подацима

Веома је битно обезбедитити конзистентост и семантичку интероперабилност на нивоу архитектуре података специфичних регистарских компоменти. Регистри врло често у процесу обраде захтева користе идентичне врсте ентитете које у различитим регистрима треба моделовати на идентичан начин. На овај начин део регистарског доменског модела могуће је уз одређене измене користити у већини регистара. Овде се пре свега мисли на ентиете који егзистирају у

сваком регистру као што су на пример лице, адреса, документ, захтев и слично. Коришћење интероперабилних модела података поједностављује и формирање агрегираних аналитичких сетова података који могу послужити за доношење комплексних одлука на државном нивоу, али који могу бити и одлична основа за креирање комерцијалних АПР сервиса.

Сваки регистарски домен треба да има независно складиште података. Конзистентност података између регистара треба обезбедити интеграционим процесима. Складишта података специфичних регистарских компоменти треба да буду оптимизована за спровођење пословних трансакција односно процеса регистрације. За аналитичке потребе треба формирати засебна складишта података оптимизован за те намене. Модел података специфичних регистарских компоменти треба да буде у складу са доменским језиком самог регистра. На овај начин представници регистара, домен експерти, аналитичари и програмери лакше ће комуницирати у процесима имплементације, одржавања и унапређења компоненти система.



На дијаграму се види пример организације слакдишта података у једном регистру.

1. Операциона база података је централно место за перзисенцију података у процесу обраде захтева;
2. База података за претрагу настаје трансформацијом података из операционог модела у модел оптимизован за претрагу;
3. Аналитичка база података настаје трансформацијом података из операционог модела у модел оптимизован за аналитику. Ова база података може садржати податке из више различитих регистара;
4. Структура базе података за чување стања процеса диретно зависи од избора компоненте за оркестрацију процеса;
5. Историјски подаци због своје величине измештени су у засебну базу подата чиме систем добија на перформантности и скалабилности;
6. База података за оперативну метрику задржи релевантне податаке о кључним пословним показатељима као што су број поднетих захтева, број завршених захтева и слично;
7. База података за системску метрику садржи релевантне податаке о кључним системским параметрима као што су заузеће меморије, оптерећеност процесора и слично;
8. База података за логове садржи податке о гречкама и битним системским догађајима.

Нефункционални захтеви

Скалабилност и перформантност

Скалабилност и перформантност система се постиже хоризонталним скалирање компоненти система и партиционисањем.

Висока доступност и континуитет пословања

Висока доступност се постиже употребом редудантних компоненти на апликативном, системском и инфраструктурном нивоу.

Од суштинског је значаја да Агенција обезбеди континуитет пословања наорочито у сегменту чувања регистрованих података, због тога је неопходна да сви подаци буду покривени одговарајућим резервним копијама.

Безбедност информационог система

Безбедност система обезбеђена је на различитим нивоима употребом различитих система и технологија.

Физички ниво:

1. Неопходно је да сервери буду инсталирани у наменским просторијама које се налазе под физичком заштитом приступа чиме се онемогућује приступ систему од стане неовлаштених лица.

Инфраструктурни ниво:

1. Неопходно је да сервери буду обезбеђени специјализованим софтвером против малициозних програма.
2. Неопходно је да сервери буду редовно ажурирани новим сигурносим закрпама.
3. Све безбедносно релевантне догашаје на системском нивоу треба
4. Системске компонетне морају користи механизме аутентификације и ауторизације корисника.

Комуникациони ниво:

1. Неопходно је да сервери на мрежном нивоу буду заштићени од неовлашћеног приступа одговарајућим уређајим (FireWall, IPS is l.).
2. Комуникација између компоненти система одвија преко криптографски заштићених канала.
3. Комуникација између корисника и серверских web компонети одвија се преко заштићених канала (SSL)
4. Сва комуникациона опрема мора користи механизме аутентификације и ауторизације корисника.
5. Сви догађаји на комуникационом нивоу који потенцијално могу указивати на малициозно понашање морају се бележити и пратити.

Апликативни ниво:

1. Компоненте система на апликативном нивоу морају користи систем корисничких налога и рола.
2. Принцип непорецивости може се обезбедити бележењем свих релевантних акција чиме се повећава одговорност и смањује могућност злоупотребе. Највећи степен непорецивост и индивидуалне одговорности може се постићи употребом електронског потписа.
3. Системске лозинке морају бити одређене комплексности и морају се мењати периодично.
4. Лозинке се морају чувати криптоване.
5. Сви догађаји који потенцијално могу указивати на малициозно понашање морају се бележити и пратити. Један од примера је покушај више узастопних неуспешних приступа.

6. Системске компоненте морају бити тестирање на типичне безбедносне пропусте (OWASP TOP 10)

Ниво података:

1. Осетљиви подаци као што су лозинке и слично морају бити криптовани на нивоу података.
2. Непорецивост и аутеничност података тамо где је то неопходно треба обезбедити употребом електронског потписа.
3. Резервне копије података треба обезбедити крипто заштитом.
4. Неовлаштени приступи подацима морају се бележити и пратити.

Неопходно је обезбедити да компоненте система користе за ауторизација и аутентификацију корисника постојећи систем за аутентификацију и ауторизацију корисника чиме би се задовољава Single SignOn концепт.

Одрживост

Правила и процедуре које се спроводе у регистрима доминантно диктирају прописи. Због честих измена закона и подзаконских аката, могућност брзе и једноставне измене информационог система су од виталног значаја. Могућност брзог, једноставног и економичног успостављања нових регистара је такође веома важно. Смањење трошкова експлоатације и одржавања је још један од важних захетва, јер АПР ове трошкове финансира из сопствених средстава која су ограничена.

Висок ниво одрживости система треба обезбедити:

1. Добром архитектуром и дизајном решења;
2. Квалитетним кодирањем и документовањем;
3. Аутоматизованим тестирањем и испоруком;
4. Раздвојеним окружењем за тестирање и продукцију;
5. Изворним кодом у власништву АПР;
6. Обуком запослених у АПР.

Процеси, методе и алати за развој софтвера

Процеси, методе и алати који се користе при развоју система на пројекту треба да обезбеде уређен приступ у изради софтвера.

Процеси развоја софтвера који се примењују на пројекту треба да:

- дефинишу радне оквире и редослед активности
- дефинишу како се спроводе активности прикупљање захтева, дизајн, развој, тестирање...

Методе које се примењују на пројекту треба да буду:

- испробане технике за спровођење одређених активности
- методе за анализу, моделовање, дизајн, тестирање итд...

Алати који се примењују на пројекту треба да:

- Обезбеде аутоматизацију активности
- помажу у систематској примени софтверског инжењерства

Пожељан процесни оквир развоја система је РУП процес (Rational Unified Process), који подразумева специјализоване процесе итеративног развоја и инкременталне испоруке, са наглашеним активностима у појединим фазама. Процес је вођен ризицима, процес развоја је вођен случајевима коришћења (Use Case) и архитектурално је центричан.

Што се метода тиче пожељно је да се у процесу развоја користи објектно орјентисана анализа и дизајн (Object Oriented Analysis and Design – OOAD) uz korišćenje UML dijagrama i BPMN dijagrama kroz 4+1 View Model.

Пројектовање и моделовање система мора бити спровођено коришћењем алата Enterprise Architect, а испоруке треба да буду, осим у форми докумената и у формату еар фајла.

Пожељно је да аутоматизовани тестови компоненти система буду обезбеђени коришћењем SOAP-UI алата (SmartBear).

Пожељно је да за време трајања пројекта извођач обезбеди online колаборациони портал, ради лакше координације и заједничког рада тимова на пројекту и праћења реализације активности у оквиру пројекта.

ОПИС ЗАДАТАКА

Обухват

Услуге које извођач треба да пружи АПР кроз пројекат су: израда детаљне функционалне и нефункционалне спецификације за развој новог информационог система за Регистар привредних субјекат, израда детаљног дизајна система, израда минимално одрживог производа на коме ће бити потврђен или коригован предложени дизајн и израда детаљне техничке спецификације за пуну имплементацију новог информационог система за Регистра привредних субјекат и миграцију података, докумената и процеса из постојећег у нов систем Регистра привредних субјеката. Наведене услуге извођач мора да пружи кроз неколико фаза.

Прикупљање и анализа захтева

1. Упознавање са релевантним прописима и упутствима;
2. Упознавање са доменом проблема кроз анализу: садашње организације и начина рада Регистра привредних субјеката, тренутних пословних процеса и постојећих апликација за спровођење послова регистрације;
3. Идентификовање свих заинтересованих страна;
4. Разговор са свим заинтересованим странама у циљу прикупљања свих потребних информација за припрему спецификације захтева за израду будућег софтверског решења, кроз следеће активности:
    a. Прикупљање пословних захтева (опис пословни циљеви високог нивоа које треба постићи израдом будућег софтверског решења);
    b. Прикупљање корисничких захтева (попис функционалности из перспективе заинтересованих страна):
        i. Опис пословних процеса и активности у оквиру процеса;
        ii. Опис учесника у пословним процесима и њихових права;
        iii. Опис ентитета који учествују у пословним процесима (попис основних класа и њихових релација);
        iv. Опис пословних правила у току спровођењу пословних процеса;
        v. Опис случајева коришћења;
        vi. Нацрт корисничких форми.
    c. Дефинисање функционалних захтева који садрже корисничке захтеве и екстерна понашања система, који нису видљиви крајњем кориснику:
        i. Како систем интерагује са окружењем (идентификација интеграционих сервиса);
        ii. Које излазе систем генерише (извештаји, претраге);

iii. Статички модел високог нивоа;

iv. Шта није у опсегу реализације система.

d. Дефинисање нефункционалних захтева везаних за квалитет и ограничења:

i. Атрибути квалитета: перформансе, поузданост, безбедност;

ii. Технолошка ограничења (избор технологија);

iii. Оквир захтева у погледу избора адекватног процеса и метода за развоја будућег софтверског решења (РУП, Агиле, Линерни итд.);

e. Дефинисање имплементационих захтева:

i. Захтеви везани за миграција из постојећег система у будући систем;

ii. Опис захтева везаних за документовање будућег софтверског производа;

iii. Опис захтева везаних за припрему тренинг материјала;

iv. Опис захтева везаних за обуке;

5. Валидација захтева са наручиоцем по следећим критеријумима:
   a. Исправност захтева
   b. Доследност захтева
   c. Недвосмисленост захтева
   d. Свеобухватност захтева
   e. Релевантност захтева
   f. Могућност тестирања захтева
   g. Могућност праћења захтева
6. Израда и верификација Спецификације софтверских захтева са наручиоцем.

Резултат фазе анализе је **Спецификација софтверских захтева** (Software Requirements Specification – SRS) на основу које ће бити даље дизајниран система и реализована сама имплементација система. Спецификација софтверских захтева може бити израђена као више докумената или као један интегрални документ и обавезно мора бити додатно испоручена у форми пројекта израђеног применом Enterprise Architect алата (Sparx System, **eap** формат фајла).

Пројектовање решења

На основу Спецификација софтверских захтева пројектни тим извођача мора израдити **Пројекат за израду софтвера**, који се састоји од два дела:

1. Конептуални пројекат – који описује функције система на начин који је разумљив клијенту и описује шта систем ради;
2. Технички пројекта – који ће омогућити тиму који треба да развије софтвер да дефинише хардверске и софтверске потребе, компоненте и њихову комуникацију, улазе и излазе система, архитектуру система и друго. Овај документ треба да описује како систем ради.

Пројектовање је потребно спроводити посматрањем система одозго на доле кроз две фазе:

- Пројектовање архитектуре система
- Детаљно пројектовање система

При пројектовању је неопходно држати се свих смерница које су наведене у одлељку **Захтеви у погледу архитектуре и дизајна**.

Пројекат за израду софтвера обавезно мора бити у форми пројекта израђеног применом Enterprise Architect алата (Sparx System, **eap** формат фајла) и треба да буде у истом пројекту кроз који је израђена Спецификација софтверских захтева.

Развој минимално функционалног производа

На основу Пројекта за израду софтвера потребно је реализовати развој минимално функционалног производа (MVP). Овај производ треба да садржи све пројектоване компоненте и само кључне функционалности (нпр. по један упис, промену и брисање) које су довољне да би производ могао да се испоручи на окружење корисника и оцени. Он треба да послужи да се провери рано задовољство корисника, прикупе повратне информације, потврди или евентуално коригује предложени дизајн решења.

Фазадизајна и фаза развоја не морају временски бити оштро раздвојене, тако да ће Пројекат за израду софтвера и сам софтвер који се развија, до краја фазе развоја бити континуирано и конзистентно развијани и ажурирани до своје коначне верзије.

Израда детаљне техничке спецификација

На бази коначне верзије Пројекат за израду софтвера извођач ће израдити **Детаљну техничку спецификацију** за пуну имплементацију система Регистра привредних субјеката, у оквиру које ће бити дата и процена: буџета, ресурса и временског оквира неопходних за пуну имплементацију система, миграцију, припрему за продукцију и пуштање система у продукцију.

Обавезе, одговорности и кључни догађаји

Иницијална фаза

- **Почетак пројекта**– Уводни састанак извођача са релевантним представницима АПР са следећим темама:
  - Дискусија о циљевима пројекта, тимовима који ће учествовати у пројекту, плану комуникације, на бази чега ће бити израђена Пројектна повеља;
  - Прикупљање основних информација о активностима на пројекту, временском распореду активности, предметима испоруке и ризицима на пројекту, на бази чега ће бити израђен Пројектни план.

Извођач ће сачинити белешку са састанка и доставити га претставницима АПР.

- **Пројектна повеља и Пројектни план**

На бази разговора са уводног састанка извођач ће израдити и доставити Пројектну повељу и Пројектни план. Пројектни план треба да садржи активности са временским распоредом, учесницима са улогама и одговорностима на страни извођача и АПР, ризике и управљање ризицима.

Извођач ће доставити представницима АПР Пројектну повељу и Пројектни план. Представници АПР ће извршити ревизију докумената и усвојити их.

Фаза анализе

- **Спецификација захтева**

У складу са Пројектним планом, а на основу прикупљених захтева и анализе захтева, извођач ће израдити спецификацију захтева и доставити је представницима Агенције за привредне регистре.

- **Валидација и верификација захтева**

Представници АПР ће валидирати и верификовати спецификацију захтева. Консултант ће помоћи да се разјасне све недоумице до којих дође у процесу верификације захтева.

- **Израда и верификација Спецификације софтверских захтева**

На основу верификованих захтева извођач ће израдити документ Спецификација софтверских захтева.

Представници АПР ће извршити ревизију и усвојити Спецификацију софтверских захтева.

Фаза пројектовања

- **Пројектовање архитектуре система**

Извођач ће на основу Спецификације софтверских захтева дефинисати архитектуру система која мора описати:

- Како су објекти груписани у компоненте на системском нивоу;
- Како су ове компоненте структуиране;
- Како су интерфејси компоненти дизајнирани;
- Како ове компоненте интерагују;
- Како ове компоненте задовољавају нефункционалне захтеве.

Дизајн архитектуре треба да прикаже структуру или структуре система, које садрже софтверске елементе, спољa видљиве особине тих елемената и релације између њих (SEI дефиниција)

Дизајн архитектуре треба да прикаже основну организацију система која је отеловљена у својим компонентама, њиховим међусобним релацијама и релацијама са окружењем и принципи који су водиља за дизајн и развој (IEEE дефиниција)

Дизајн архитектуре треба да дефинише структуру система у смислу архитектурално релевантних компоненти, дефинише понашање система у смислу интеракције између архитектурно релевантних компоненти и реши већину нефункционалних захтева.

Архитектурно релевантне компоненте:

- Утичу на укупну структуру и понашање (пословно и технички критични делови система);
- Утичу на атрибуте квалитета (нпр: перформансе, безбедност, поузданост итд...);
- Утичу на технолошка ограничења или ограничења окружења (нефункционални захтеви).

Важно је да архитекта донесе одлуке који погледи у дизајну ће бити коришћени. Уобичајена пракса је да се користе следећа четири погледа:

- Логички поглед;
- Процесни поглед;
- Физички поглед;
- Развојни поглед.

Препорука је да се у овом пројекту користе и додатни поглед:

- Информациони поглед.

Декомпозицију система треба наставити док се не постигне довољан ниво детаљности и практичности, који ће омогућити дизајнерима да квалитетније и ефикасније спроведу детаљно пројектовање система. Потребно је препоручити стилове за које се утврди да су релевантни и помоћу којих ће захтеви бити решени на најбољи начин (нпр: Pipe-and-Filtering, Publish-and-Subscribe, Client/Server, N-Tier, SOA, REST итд.).

Ограничења у смислу процеса развоја софтвера и алата могу се решити касније у фази развоја. Реузлтат ове фазе треба приказати у документу **Опис архитектруре система**.

- **Детаљно пројектовање система**

У овој фази ће бити донете важне одлуке у погледу технологија, метода и алата који ће се користити у фази развоја. Ту се мисли на избор програмских језика за развој компоненти и

модула, дифинисање модела ентитета, избор система за чување података и докумената, дефинисање начина и правила обраде податак, избор middleware компоненти и друго.

Задатак ове фазе је пројектовање на нижим нивоима апстракције:

- Пројектовање саставних делова архитектурно релевантних и нерелевантних компоненти;
- Пројектовање класа;
- Дефинисање структуре и понашања конститутивних елемената и компоненти.

Пројектовање треба радити кроз погледе које је дефинисао архитекта.

Логички поглед

Дефинише структуру и понашање система:

- садржи компоненте релевантне за архитектуру система;
- њихове међусобне релације;
- одговорности сваке компоненте система;
- њихове интерфејсе;
- интеракцију токова података и токова контроле
  - поруке (one-way, two-way, callback...);
  - како компоненте сигнализирају једна другој да треба извршити неки задатак.

Једна од нотација за приказивање логичког погледа је коришћење UML компонентних дијагарама

Процесни поглед

Дефинише конкурентност и синхронизацију:

- Опције дизајна у погледу перформанси и скалабилности;
- Које компоненте се могу извршавати конкурентно;
- Како се та конкурентна извршавања синхронизују.

Могуће је креирати разне моделе:

- Како су системски радни токови партиционисани кроз процесе;
  - Структура процеса на нивоу комплетног система;
- процесне токове у оквиру сваког процеса;
- мапирање конкурентних функција у процесима и токовима;
- механизми комуникације између процеса;
  - како конкурентне компоненте кроз одвојене процесе могу интераговати (нпр. RPCs, Named Pipes)?
- стања, транзиције и догађаји конкурентних компоненти у тренутку извршавања;
- синхронизација између конкурентних токова;
  - како спречити конкурентне токове да не доведу до корупције података (нпр. дељена меморија, приступ подацима о стране конкурентних токова);
  - како закључати и откључати ове податке.

Процесни погледа могуће је моделовати коришћењем различитих UML dijagrama: компонентног, секвенци, машине стања, активности, као и BPMN дијаграма.

Физички поглед

Дефинише физичко окружење на коме ће се систем извршавати:

- опис серверске, мрежне и друге опреме;
- спецификација хардвера и мреже неопходна за задовољавање планираних капацитета и доступности система;
- мапирање софтверских компоненти на опрему.

Физички поглед могуће је моделовати коришћење UML deployment дијаграма или цртежа израђених помоћу неког алата (нпр. Microsoft Visio).

Развојни поглед

Дефинише развојно окружење:

- Интегрисана развојна окружења која ће се користити у развоју;
- Структуру изворног кода и начин његовог верзионирања;
- Алат за планирање и аутоматизацију таскова;
- Алат за тестирање и аутоматизацију тестирања;
- Алат за изградњу извршног кода и испоруку на продукционо окружење.

Веома је важно да се у фази развоја примењују принципи и правила писања доброг кода и пожељно је да то у овој фази буде дефинисано. Овде су набројани неки од принципа и правила:

- тежити једноставном и читљивом коду;
- пратити стандарде кодирања (именовање, распоред кода, коментари...);
- пратити најбоље праксе у дизајну;
- измене кода су неизбежне - писати код који се може накнадно проширити;
- користити технике специфичне за језик и платформу (нпр. атрибути и рефлексије у C#);
- код изолован од инфраструктуре и окружења;
- користити абстракције и енкапсулацију;
- писати код који може бити поново употребљен;
- користити друге изворне кодове (готове компоненте, делове кода);
- изабрати алате за верзионирање, тимски рад и аутоматизацију.

У овој фази је важно дефинисати и моделе и методе тестирања које ће бити примењене у фази развоја. Начин тестирања доста зависи од изабраног модела процеса развоја софтвера али би свакако требало спроводити следећа тестирања:

- Unit testing;
- Component testing;
- Integration testing.

Добра пракса је обезбедити што више аутоматизованих тестова.

Свакако би требало дефинисати и моделе и методе тестирања које би требало спроводити у развоју будућег система.

Информациони поглед

Дефинише структуре и понашање података релевантних за архитектуру система:

- главне ентитете података које конзумирају или производе компоненте система;
- структуру ентитета и релације међу њима;
- како информације теку између компоненти (кроз интерфејсе дефинисане у логичком погледу);
- власништво над податцима (системе или складишта података која садрже валидну верзију података);
- управљање подацима (безбедност, креирање, архивирање, враћање).

Нотације које се користе су најчешће дијаграми класа (структурни погледи) и дијаграми секвенци и дијаграми машине стања (динамички погледи). За дефинисање управљања подацима могу се користити и друге нотације, дијаграми или само текст.

На основу Спецификација софтверских захтева пројектни тим извођача ће израдити прву верзију документа **Пројекат за израду софтвера** који се састоји од Описа архитектуре система и детаљног дизајна система**.**

Извођач ће доставити представницима АПР прву верзију Пројекта за израду софтвера. Представници АПР ће извршити ревизију докумената и усвојити их.

Фаза развоја

У фази развоја потребно је превести дизајн у извршни програмски код минимално функционалног производа (MVP). Релација између дизајна и развоја зависи од изабраног модела процеса развоја софтвера (итеративни, инкрементални, агилни приступ).

Када минимално функционалан модел буде развијен и тестиран, извођач ће на њему проверити задовољство корисника, прикупити повратне информације, потврдити или евентуално кориговати предложени дизајн решења.

Фаза пројектовања и фаза развоја не морају временски бити оштро раздвојене, тако да ће Пројекат за израду софтвера и сам софтвер који се развија, до краја фазе развоја бити континуирано и конзистентно развијани и ажурирани до своје коначне верзије.

Коначне верзије извршног и комплетног изворног кода MVP, као и коначну верзију Пројекта за израду софтвера извођач ће предати АПР на крају ове фазе.

Финална фаза

На бази коначне верзије Пројекат за израду софтвера извођач ће израдити **Детаљну техничку спецификацију** за пуну имплементацију система Регистра привредних субјеката, као и документ **Студија изводљивости** са проценом: буџета, ресурса и временског оквира неопходних за пуну имплементацију система, миграцију, припрему за продукцију и пуштање система у продукцију.

Извођач ће доставити представницима АПР Детаљну техничку спецификацију и Студију изводљивости. Представници АПР ће извршити ревизију докумената и усвојити их.

У овој фази биће извршена и коначна примопредаја пројекта.

Извођач је одговоран за следеће испоруке:

	Испоруке	**Рокови**
1	Пројектна повеља и Пројектни план	Пре почетка фазе анализе
2	Спецификација софтверских захтева	На крају фазе анализе
3	Опис архитектруре система	У фази пројектовања, на крају пројектовање архитектуре система
4	Пројекат за израду софтвера - прва верзија	На крају фазе пројектовања
5	Извршни и изворни код MVP и Пројекат за израду софтвера – коначна верзија	На крају фазе развоја
6	Детаљна техничка спецификација и Студија изводљивости	На крају пројекта

Плаћање по завршетку фаза:

	Фазе пројекта	**Плаћања**
1	Иницијална фаза	0%
2	Фаза анализе	30%
3	Фаза пројектовања	30%
4	Фаза развоја	20%
5	Завршна фаза	20%

ТЕХНИЧКА ПОНУДА

Техничка понуда треба да садржи опис процеса развоја софтвера, метода и алата које ће извођач користити у пројекту, пројектну методологију и опис процеса управљања пројектним активностима које извођач намерава да примени, као и предлог пројектног плана са временаским распоредом активности и мапирањем активности на чланове предложеног пројектног тима. Важно је да се у пројектном плану предвиде и нагласе активности за чију реализацију се очекује ангажовање извршилаца из АПР, са описом активности, врстом извршилаца и процењеним временом за њихову реализацију.