

		Verzija:	1.0
Projekt:	MOPE-IS Dovoljenja	Datum:	01.06.2026
Vsebina:	IS Dovoljenja - Krovni arhitekturni dokument		

Reference

Smernice MDP za razvoj informacijskih rešitev
Dokument Generične Tehnološke Zahteve (GTZ)
Posebne tehnološke zahteve SLO4D

Kazalo vsebin

SEZNAM KRATIC	5
1. Uvod	7
2. Seznam gradnikov z opisi	7
2.1. Makro arhitektura	7
2.2. Makro arhitekturni gradniki	10
2.3. Gradniki postavitve	10
2.4. Mikro arhitekturni gradniki	13
3. Popis uporabljenih tehnologij	14
3.1. Predstavitveni nivo	15
3.2. Poslovni in integracijski nivo	16
3.2.1. Kontejnerji	18
3.3. Podatkovni nivo	19
3.4. Dokumentni nivo	19
4. Specifikacije aplikacije za prikaz podrobnosti delovanja vseh vključenih komponent	20
4.1. Nadzor nad integracijami z zunanji sistemi	20
4.2. Produktni nadzorni sistemi	20
4.3. Nadzor nad stanjem sistema v celoti	21
5. Arhitektura sistema za implementacijo	21
5.1. Avtentikacija uporabnikov	23
5.1.1. Tehnološka zasnova avtentikacije	23
5.2. Obdelave podatkov na zahtevo uporabnika	25
6. Varnostni in zaščitni mehanizmi	26

6.1.	Arhitekturni nivo.....	26
6.1.1.	Zaščita komunikacijskih kanalov.....	26
6.2.	Sistemski nivo	27
6.2.1.	Omejevanje dostopa.....	27
6.2.2.	Zaščita pred tretjimi osebami.....	27
6.2.3.	Avtentikacija servisov	30
6.3.	Podatkovni nivo	31
7.	Integracije z zunanjimi sistemi.....	32
7.1.	Izhodne integracije z zunanjimi sistemi.....	32
7.1.1.	MS Sharepoint.....	32
7.1.2.	KRPAN in CEH.....	33
7.1.2.1.	Integracija KRPAN.....	36
7.1.2.1.1	Primer klica servisa KRPAN – dodajanje dokumenta.....	37
7.1.2.1.2	Primer klica servisa KRPAN – pridobitev stanja dokumenta	37
7.1.2.2.	Integracija CEH	38
7.1.3.	AJPES.....	39
7.1.3.1.	Prevzem podatkov iz AJPES	40
7.1.3.1.1	Prvi (inicialni) prenos	41
7.1.3.1.2	Periodičen prenos sprememb.....	41
7.1.4.	SI-CES	41
7.1.5.	GURS.....	42
7.1.5.1.	Prevzem podatkov iz GURS	43
7.1.5.1.1	Prvi (inicialni) prenos	43
7.1.5.1.2	Periodičen prenos sprememb.....	43
7.1.6.	IS Monitoring	43
7.2.	Vhodne integracije iz drugih zunanjih sistemov	44

Kazalo slik

Slika 1 - Storitvene povezave med notranjimi in zunanji moduli s protokoli povezav, reverse proxy nivoji so predmet odločitve upravljalca	8
Slika 2 - Sistem v distribuirani postavitvi omogoča višjo razpoložljivost in zanesljivost sistema kar ustreza raznolikim varnostnim omejitvam	9
Slika 3 - Arhitektura komponent sistema po nivojih	15
Slika 4 - Avtentikacijski tok uporabnika	23
Slika 5 - Dokazilo o pristnosti zagotovi SI-CAS; velja za uporabnike v HKOM in DMZ	24
Slika 6 - Osveževanje dostopnega žetona	24
Slika 7 - Izvajanje asinhronih nalog	25
Slika 8 - Izmenjava dokazil med uporabnikovim brskalnikom in elementi sistema IS Dovoljenja.	28
Slika 9 - Storitve zavračajo zahteve brez veljavnega pristnega žetona (v odgovoru ni podrobnosti o razlogu zavrnitve)	29
Slika 10 - Tok zahtevkov in odgovorov ob običajnem dostopu uporabnika.	30
Slika 11 - Tok zahtevkov in odgovorov ob avtentikaciji servisov	31
Slika 12 - Oddaja vloge v Krpan in CEH	34
Slika 13 - Oddaja dovoljenja v Krpan in Ceh	35
Slika 14 - Prenos datotek iz CEH-a	36
Slika 15 - Integracije s sistemom AJPES	40
Slika 16 - Izmenjave s sistemom SI-CES	41
Slika 17 - Integracije s sistemom GURS	42
Slika 18 - Izmenjave podatkov z IS Monitoring	44

Kazalo tabel

Tabela 1 - Seznam kratic.....	6
Tabela 2 - Makro arhitekturni gradniki	10
Tabela 3 - Gradniki postavitve	13
Tabela 4 - Mikro arhitekturni gradniki	14

SEZNAM KRATIC

Kratika	Izvorni naziv (opis)	Slovenski prevod
API	Application programming interface	Vmesnik za aplikacijsko programiranje
CRUD	Create, read, update, delete functions	Stavki za kreiranje, branje, posodobitev, brisanje
CSV	Comma-separated values	Vrednost, ločene z vejico
DB	Database	Podatkovna zbirka
DMZ	DeMilitarized Zone	Javno dostopno informacijsko okolje na MDP
DRO	Državni računalniški oblak	
DS	Dokumentni sistem (KRPAN).	
EJB	Enterprise Java beans	
eZK	Elektronska zemljiška knjiga	
FTS	Full-text search	Iskanje po vsem besedilu
GIS	Geographic information system	Geografski informacijski sistem
GURS	Geodetska uprava Republike Slovenije	
GTZ	Generične tehnološke zahteve za razvoj informacijskih sistemov	
HKOM	Hitro komunikacijsko omrežje	
HTTP	Hypertext transfer protocol	Protokol za prenos hiperteksta
HTTPS	Hypertext transfer protocol secure	Protokol za prenos hiperteksta prek SSL
IS	Information system	Informacijski sistem
JDBC	Java database connectivity	Javanska povezljivost podatkovnih baz

JSON	Javascript object notation	Objektni javascript zapis
LDAP	Lightweight Directory Access Protocol	Preprost protokol za dostop do imenika
MDP	Ministrstvo za digitalno preobrazbo	
MOPE	Ministrstvo za okolje, podnebje in energijo	
NAT	Network address translation	Prevajanje omrežnih naslovov
PTZ SLO4D	Posebne tehnološke zahteve SLO4D	
REST	Representational state transfer	Predstavitveni prenos stanj
RLS	Row level security	Varnost na ravni vrstice
RS	Republika Slovenija	
SI-CES	Centralni sistem za strežniško e-podpisovanje	
SSL	Secure socket layer	Sloj varnih vtičnic
SSO	Single sign-on	Enkratni vpis
SQL	Structured query language	Strukturirani jezik za poizvedovanje
URL	Uniform Resource Locator	Naslov vira v enotni obliki.
WAR	Web archive	
XML	Extensible Markup Language	Razširljivi označevalni jezik
XSD	XML Schema Definition	Definicija sheme razširljivega označevalnega jezika

Tabela 1 - Seznam kratic

1. Uvod

Dokument vsebuje opis in razlago arhitekture IS Dovoljenja ter popis funkcionalnosti in poslovnih procesov, ki jih sistem podpira.

IS Dovoljenja predstavlja sodobno informacijsko rešitev za podporo postopkom izdaje/spremembe okoljevarstvenih dovoljenj in drugih odločb s področja varstva okolja in pregled nad okoljskimi podatki na Direktoratu za okolje v okviru Ministrstva za okolje, podnebje in energijo (v nadaljevanju MOPE).

IS Dovoljenja je varen in učinkovit enotni sistem, ki bo ponujal informacijsko podporo poslovnim procesom, ki se nanašajo na okoljevarstvena dovoljenja in druge odločbe s področja varstva okolja ter omogočal celovit pregled nad podatki o predmetnih postopkih ter sledenje izvedbi le teh. Sistem uporabnikom zagotavlja izpise v obliki priročnih poročil v tekstovni in grafični obliki ter izvoze podatkov. V IS Dovoljenja so vzpostavljene naslednje povezave z zunanjimi sistemi:

- MS Sharepoint,
- Dokumentni sistem KRPAN (skupaj s sistemom CEH),
- AJPES,
- GURS (kataster nepremičnin in register prostorskih enot),
- IS Monitoring na ARSO,
- gradniki SLO4D v okviru skupnih gradnikov na MDP (SI-CAS, Varnostna shema, SI-CES),
- GIS pregledovalniki (Atlas okolja, Atlas voda (vključno z vsemi hidrografskimi podatki za posamezne lokacije), iSlovenia, Kakovost podzemnih voda, Kakovost površinskih voda itn. Integracija se implementira v obliki povezave (linka) iz IS Dovoljenja do posameznega GIS pregledovalnika v obsegu, kot ga obstoječi GIS pregledovalnik omogoča (npr. odpiranje na točno določeno lokacijo in/ali samodejni vklop določenih grafičnih slojev). Če posamezen GIS pregledovalnik ne podpira teh funkcionalnosti, lahko IS Dovoljenja ponudi le povezavo (link) do vstopne strani posameznega GIS pregledovalnika.

Podatki so dostopni na enem mestu, povezave na zunanje vire podatkov omogočajo pravočasno in konsistentno izmenjavo podatkov med sistemi, kar uporabnikom omogoča pregled nad izdanimi odločbami ter poslovnimi procesi v fazi izvedbe, ki se nanašajo na postopke izdaje/spremembe okoljevarstvenih dovoljenj in drugih odločb s področja varstva okolja.

2. Seznam gradnikov z opisi

Tehnološka platforma je načrtovana na podlagi dokumentov Referenčna arhitektura, Smernice MDP za razvoj informacijskih rešitev in Generične tehnološke zahteve za razvoj informacijskih sistemov (GTZ).

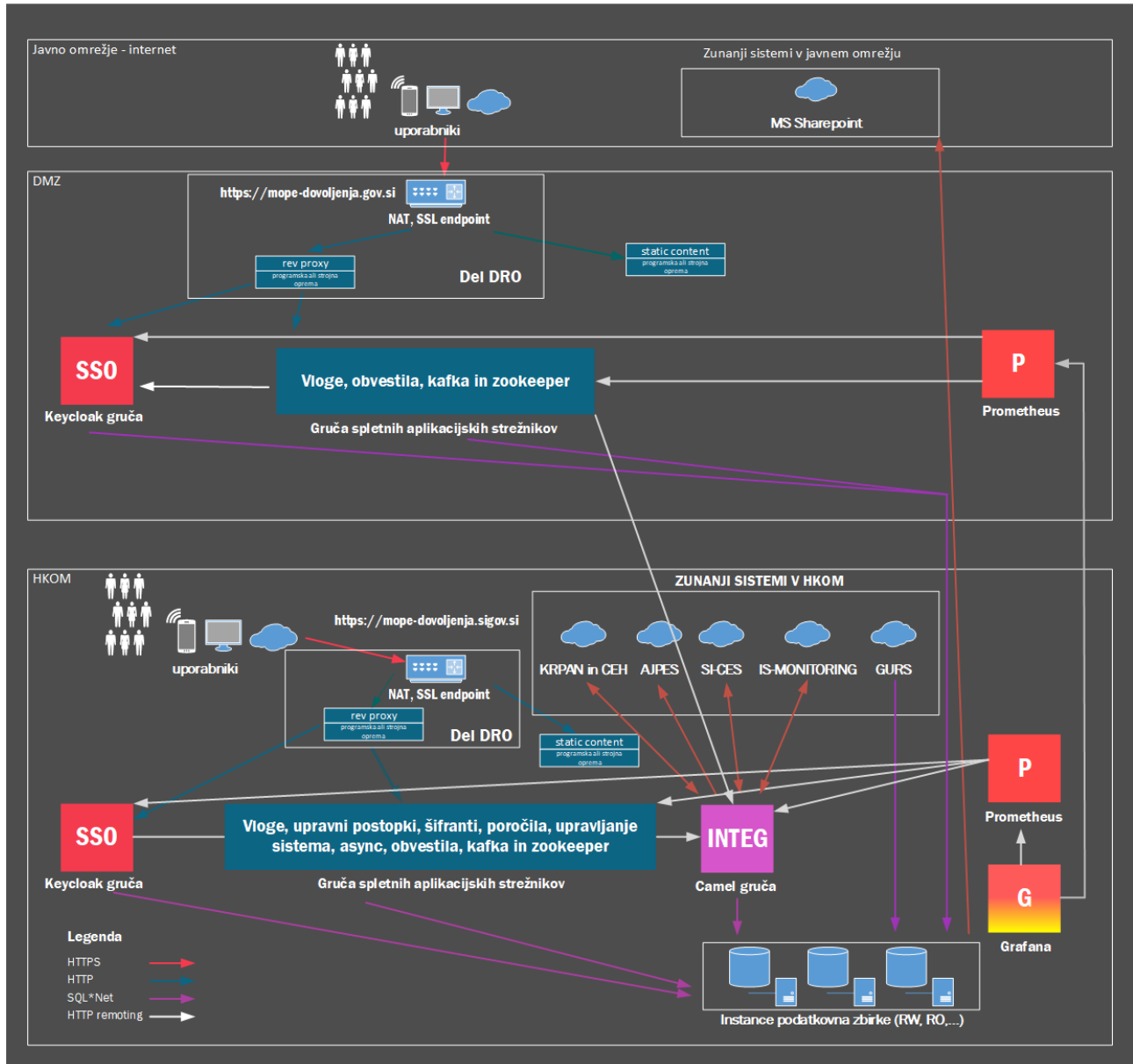
2.1. Makro arhitektura

Makro arhitektura sistema temelji na organizacijski urejenosti državne uprave in trenutnem stanju (arhitekturi) sistema na MDP. Obstaja ena postavitvev sistema z možnostjo postavitve sekundarne postavitve (kolokacija).

Upravljalavec sistema se lahko sam odloči ali bo sistem postavljen v 2n+1 obliki ali ne. Takšna postavitvev je lahko v obliki enega Docker grozda, ali se lahko postavijo ločeni grozdi. Sistem je pripravljen na redundantno postavitvev, pri tem morajo strežniki, ki so v delujočem stanju (online) ohranjati povezavo z ostalimi strežniki v delujočem stanju zaradi zagotavljanja odvisnih storitev in distribuiranega sistema predpomnilnika. Operativno to pomeni, da če je ena postavitvev v Ljubljani in

druga v Mariboru, mora ena izmed njiju biti offline ali pa mora med njimi biti vzpostavljena aktivna povezava. Več o tem v nadaljevanju poglavja.

Makro arhitektura sledi načelu centralizacije nadzora nad delovanjem sistema in evidentiranimi podatki tako v smislu lokacije kakor v smislu implementacije.



Slika 1 - Storitvene povezave med notranjimi in zunanji moduli s protokoli povezav, reverse proxy nivoji so predmet odločitve upravljalca

Sistem je postavljen distribuirano. To je mogoče razumeti kot mikrostoritve v smislu ločevanja odgovornosti, visoke razpoložljivosti in skalabilnosti. Od klasične definicije mikrostoritev se sistem loči zaradi centralizacije podatkov. Edine povezave posameznih storitev so:

- Vhodne:
 - REST storitev
- Izhodne:
 - Izdajatelji žetonov za dostop (SSO) - Teh je več, ker so praviloma postavljeni v clustered načinu,

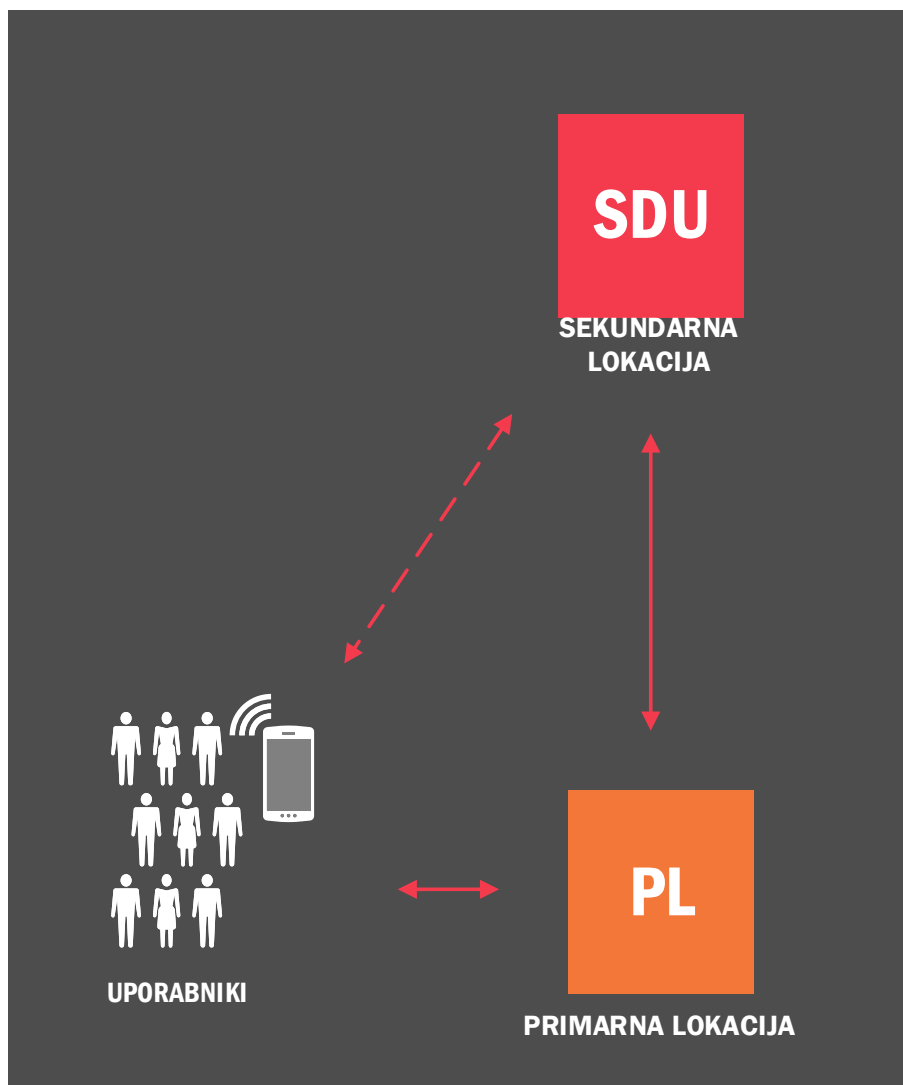
- Podatkovne zbirke (množina, ker se lahko ločijo transakcijske, analitične in logging zbirke – tako so pripravljene povezave na podatkovne zbirke)

Če delujejo navedene komunikacijske točke, je storitev mogoče uporabljati. Tako je storitev na voljo v največji mogoči razpoložljivosti in največji mogoči odpornosti (resilience).

Sistem je ločen na storitve po pravilih domenskega ločevanja. Torej na podlagi poslovnih procesov in dela uporabnikov. Tako posamezen uporabnik lahko veliko večino dnevnega dela opravi znotraj ene storitve. S tako delitvijo je zmanjšan pritisk na prehajanje med storitvami in s tem zmanjša odvisnost podpore poslovnemu procesu od večjega števila dejavnikov.

Komunikaciji s sistemi izven IS Dovoljenja je namenjen aplikacijski modul Integ, ki nima neposredne komunikacije z uporabniki. Na tak način je doseženo najboljše varovanje zunanjih sistemov in IS Dovoljenja pred zunanjimi vplivi.

Primarne lokacije, kolokacije in drugi mehanizmi varovanja stabilnosti strojne in systemske opreme so predmet upravljavca. Sistem upravljavcu ne postavlja posebnih omejitev. Na primer mogoč je preklop med lokacijami »v živo«, izbris stanja transakcije (zaradi preklopa podatkovne baze), kar povzroči razveljavitev celotne transakcije in stabilen rezultat.



Slika 2 - Sistem v distribuirani postavitvi omogoča višjo razpoložljivost in zanesljivost sistema kar ustreza raznolikim varnostnim omejitvam

2.2. Makro arhitekturni gradniki

Gradnik	Opis
Postavitev sistema v DMZ okolju	Postavitev aplikacije v DMZ okolju na MDP. To postavitev uporabljajo zunanji uporabniki sistema (podrobnosti v nadaljevanju).
Interni uporabniki	Uporabniki, ki so v omrežju HKOM.
Zunanji uporabniki	Uporabniki, ki niso v omrežju HKOM.

Tabela 2 - Makro arhitekturni gradniki

2.3. Gradniki postavitve

Gradnik	Opis
NAT, SSL endpoint	<p>Naprava ali skupina naprav, ki predstavlja ponor (endpoint) varne komunikacije med brskalnikom uporabnika in izbrano aplikacijo IS Dovoljenja. Predvidoma implementirana v visoki razpoložljivosti in izvaja naloge:</p> <ol style="list-style-type: none">1. SSL endpoint (HTTPS). Ima nameščen strežniško kvalificirano digitalno potrdilo, za posamezne spletne naslove zahteva uporabniško strežniško digitalno potrdilo, katerega elemente posreduje v obliki glave sporočila (Header) zalednim sistemom.2. Reverse proxy in load balancer. Posamezne spletne naslove prevaja in posreduje ustreznim zalednim sistemom (aplikacijskim strežnikom). Podpira tudi WebSocket nadgradnjo protokola.3. Strežba statičnih vsebin. Izbrane spletne naslove streže z lokalne hrambe (diska). Predvsem gre za javascript, html, css in png datoteke. V manjši meri tudi PDF (navodila itd.). Podpira samodejno preusmerjanje zahtev na privzet objekt.
NGINX	<p>Reverse proxy in strežnik za statično vsebino. Uporabi se le izjemoma, v okoljih, kjer teh storitev ni mogoče izvesti na strojnem nivoju. Praviloma postavljen v obliki gruče (visoka razpoložljivost). Praviloma se v produkciji ne uporablja.</p> <p>Statična vsebina se namešča z uporabo Jenkins postopkov.</p>
Keycloak gruča	<p>Centralni avtentikacijski podsistem. Lahko se uporabi tudi kot skupni gradnik MDP. Vsi elementi aplikacij IS</p>

	<p>Dovoljenja zaupajo Keycloak gruči. Vsak uporabnik (tudi klienti API storitev) se avtentificirajo na Keycloak. Keycloak izda žeton, ki je podpisan, kar je veljavno dokazilo za vse aplikacijske strežnike. Keycloak je redko samostojen v izvedbi avtentikacije (le kadar bi bil uporabljen mehanizem user/password). Praviloma zahtevo za avtentikacijo posreduje tretjim gradnikom (v primeru IS Dovoljenja je to sistem SI-CAS (v ozadju samodejna povezava z Varnostno shemo)). Sistem je namenjen registriranim uporabnikom, uporabniki so torej v sistem vpisani pred prvo prijavo. Registracija novega uporabnika je ločen proces, ki poteka preko Varnostne sheme.</p> <p>Keycloak uporablja ločeno povezavo na podatkovno zbirko (ob namestitvi in posodobitvi keycloak-a ta sam namesti in posodobi bazne tabele, zato potrebuje poleg DML tudi DDL pravice za shemo). Za branje in urejanje uporabnikov, shranjevanje zgodovine prijav in dodeljenih pravic uporablja CRUD stavke (nad pogledi).</p> <p>Keycloak je postavljen kot gruča aplikacijskih strežnikov (razpoložljivost in razširljivost kapacitet).</p> <p>Spremembe Keycloak se nameščajo z uporabo Jenkins postopkov.</p>
Camel gruča	<p>Apache Camel podpira vso interakcijo aplikacije Integ s sistemi izven aplikacije Integ. Sistem podpira sinhrono in asinhrono obdelavo zahtevkov, kar je definirano v okviru posamezne povezave.</p> <p>Izpad tega podsistema ne pomeni zaustavitev IS Dovoljenja, temveč izpad nalog izmenjav z zunanjimi sistemi.</p> <p>Apache Camel je nabor knjižnic, ki med drugim temelji na Apache CXF arhitekturi, podpira pa širok nabor vmesnikov in varovalnih mehanizmov. Dobavljen je kot WAR paket (prebuilt).</p> <p>Način dostopa do integracijske točke je določen za vsako storitev posebej. Zunanja storitev lahko dostopa preko skupne dostopne točke (enako kot uporabniki) ali neposredno do integracijske točke ali preko posebne dostopne točke, namenjene zgolj integracijskim storitvam.</p> <p>Apache Camel uporablja ločeno povezavo na podatkovno zbirko (brez pravic za spreminjanje sheme). Uporablja izključno shranjene procedure.</p> <p>Apache Camel je postavljen kot gruča aplikacijskih strežnikov (razpoložljivost in razširljivost kapacitet).</p>
Gruča spletnih aplikacijskih strežnikov	<p>Aplikacijski strežniki so stateless storitve, ki tečejo v Wildfly strežniku, ki je postavljen kot docker container. Storitve</p>

	<p>sicer ni omejena na docker in lahko teče v klasični postavitvi aplikacijskega strežnika ali z uporabo drugačne vrste kontejnerizacije. Navodila so pripravljena in testiranje se izvede nad docker okoljem.</p> <p>Uporabljen bo Docker (verzija 17.12.0-ce ali višja) nad OS Oracle Linux Server release 7.4 ali višja (ali CentOS7 primerljive verzije). Kontejnerji znotraj Dockerja bodo zgrajeni nad zadnjim uradnim Wildfly 29 Dockerjem.</p> <p>Storitve so stateless, kar pomeni, da je upravljavalec prost pri razporejanju in številu storitev (testiranje fail-over in podobno).</p> <p>Posamezna storitev je pripravljena kot WAR aplikacija, ki je zgrajena v okolju naročnika ali upravljavca (glej Jenkins avtomatizacija nameščanja). Teče nad Wildfly 29 aplikacijskim strežnikom.</p> <p>Storitve vsebuje izdatne možnosti beleženja (vsaka zahteva, vsak SQL ukaz, merjenje odzivnosti), kar upravljavalec nastavlja preko aplikacijskega strežnika.</p> <p>Storitve ne uporabljajo Hibernate ali drugih ORM orodij. Do podatkovne zbirke dostopajo preko shranjenih procedur (branje in shranjevanje dokumentov, postopki) ali preko pogledov (izključno v primeru iskalnikov, kjer so pogledi omejeni s stolpci in vrsticami).</p> <p>Storitve ne vsebujejo spletnih strani. Vsa interakcija s storitvami poteka preko REST protokola z uporabo JSON objektov.</p> <p>Storitve so varovane z uporabo dostopnih žetonov. Veljavnost žetona se preverja na podlagi podpisa žetona. Veljavnost podpisa se preverja na podlagi podpisnega ključa, ki je del nastavitev storitve. Dostopni žeton izdaja izključno Keycloak.</p> <p>Uporaba HTTPS med brskalnikom in SSL zaključno točko je obvezna. Aplikacija ne deluje po protokolu HTTP.</p>
Podatkovna zbirka	<p>Sistem vsebuje eno podatkovno zbirko, ki je center postavitve. Ta zbirka ima lahko kopije (za namene varovanja podatkov ali izvajanja analiz). Nabor kopij določa upravljavalec neodvisno (recimo, ali bo kopija na isti lokaciji, kopije sploh ne bo itd.). Predvidoma bo sistem postavljen z eno kopijo na isti lokaciji in eno na oddaljeni (oboje z namenom varovanja podatkov).</p> <p>Vsi elementi podatkovne zbirke se posodablajo samodejno (glej Jenkins avtomatizacija nameščanja).</p>
SVN repozitorij	<p>Sistem ima svojo postavitev repozitorija izvorne kode in polizdelkov (javnih knjižnic).</p>

Jenkins avtomatizacija nameščanja	<p>Odprtokodna rešitev za avtomatizacijo postopkov, tudi izgradnjo izdelkov na podlagi izvirne kode, polizdelkov (javno dostopne knjižnice) in navodil za izdelavo (v obliki skript in nastavitvenih datotek). Izvorno kodo in polizdelke prevzema iz repozitorija in pripravi končne izdelke.</p> <p>Rešitev tudi izvaja nameščanje izdelkov glede na nastavitve sistema (na primer ura, verzija itd.). Namestitvev izvede na treh mestih: podatkovna zbirka, strežba statičnih vsebin in docker kontejnerjev (REST API). Docker kontejnerje preda Docker swarm manager-ju (tudi master).</p>
Docker swarm manager/master	<p>Osnovni del sistema je Docker swarm, ki upravlja z Docker kontejnerji. Vsak kontejner si je najlažje predstavljati kot majhen virtualni strežnik, namenjen izvajanju enega aplikacijskega procesa. Vsak aplikacijski proces lahko v sistemu teče večkrat (paralelno) z namenom zagotavljanja višje prepustnosti in neobčutljivosti na izpade. Docker swarm manager upravlja z razmestitvijo kontejnerjev na gostitelje, s številom kontejnerjev in verzijami kontejnerjev v izvajanju. Po potrebi kontejnerje seli, zažene ali ugasne. Nadgradnje izvaja tako, da stare kontejnerje ugasne in zažene nove (politika prehoda je nastavljiva).</p>

Tabela 3 - Gradniki postavitve

2.4. Mikro arhitekturni gradniki

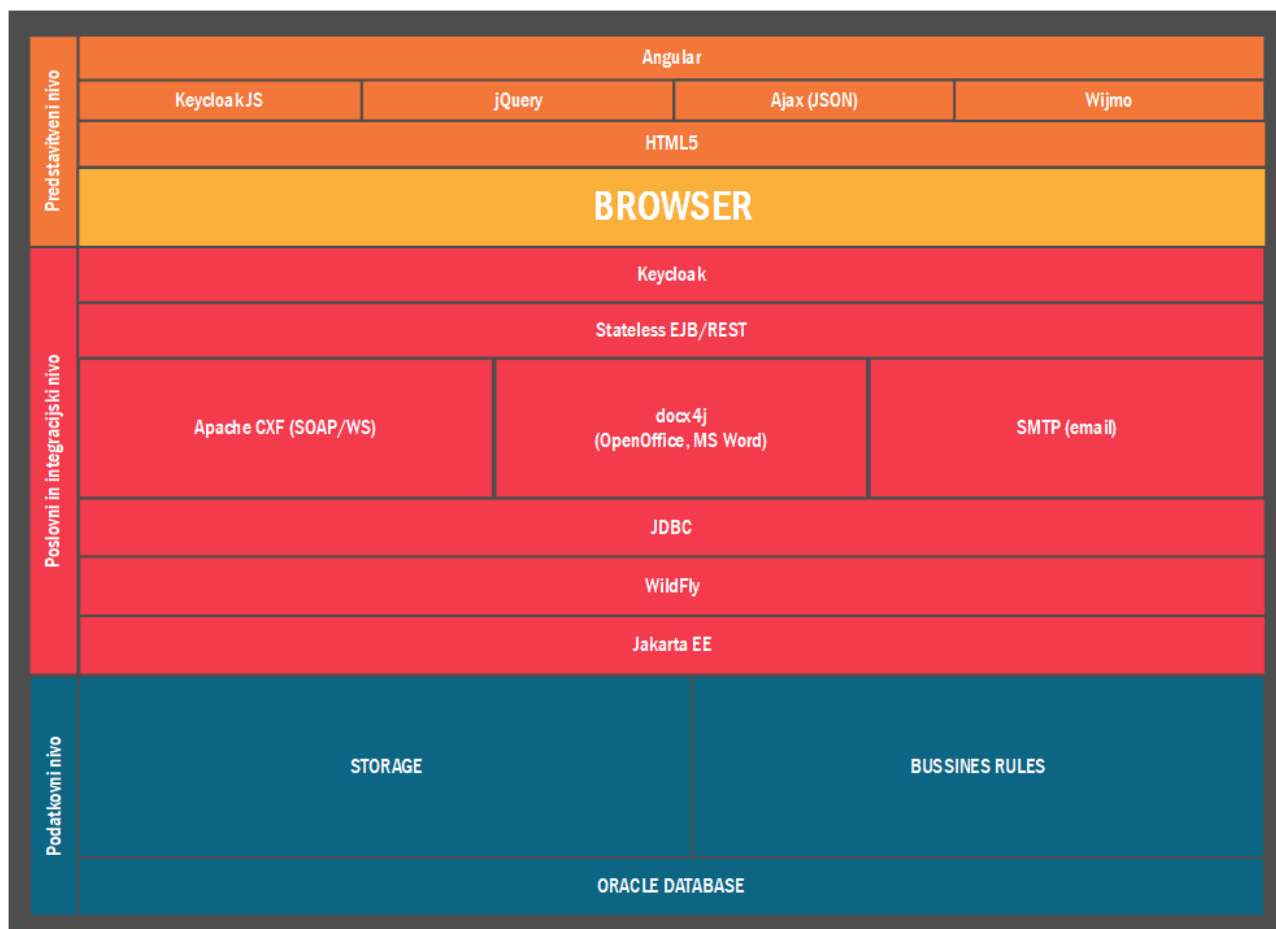
Gradnik	Opis
Oracle Database	<p>Podatkovna zbirka sistema je Oracle Database 19.0.0.0.0 z zadnjim PSU, ki je nameščena v DRO.</p> <p>Zaradi lažjega upravljanja z večjo količino podatkov bo uporabljena opcija Partitioning. Opcija se uporabi na področju večjih sklopov podatkov smiselno tako, da zmanjša pritisk na pregledovanje velikih tabel in (ob večjih brisanjih) na podatkovne dnevnike, ki se pretakajo na oddaljeno lokacijo. Dodatno ima podatkovna zbirka nameščene opcije Spatial, Oracle Text.</p> <p>Način zaščite v okolju DRO je predmet odločitev upravljavca in skrbnika aplikacij IS Dovoljenja.</p>
Oracle Spatial	<p>IS Dovoljenja na podatkovni zbirki shranjuje in analizira geografske podatke. Komponenta Oracle Spatial vsebuje funkcije in procedure za analiziranje geografskih podatkov.</p>

Diskovni podsistem	IS Dovoljenja nima posebnih zahtev glede diskovnega podsistema. V sistemu se hranijo transakcijski podatki v skladu z zakonskimi določili hranjenja podatkov. Na aplikacijskih strežnikih se ne hranijo podatki, ki bi zahtevali varovanje pred izgubo.
Poslovna pravila	Poslovna pravila se preverja z baznimi procedurami. Administracijo poslovnih pravil se izvaja preko šifranta poslovnih pravil, kjer je mogoče nastaviti tekst in proženje poslovnega pravila. Poslovna pravila se lahko prožijo pred shranjevanjem zapisov ali pri prehodih med različnimi statusi delovnega toka.
Jakarta EE	IS Dovoljenja temeljijo na Jakarta EE 10 (predvsem področja Servlet, JAXRS, Security Realm).
Wildfly/JBoss	IS Dovoljenja je pripravljen za izvajanje nad spletnim aplikacijskim strežnikom Wildfly 29, ga pa je mogoče z manjšimi prilagoditvami prenesti tudi na druge Java strežnike.
Kafka / Zookeeper	Kafka in Zookeeper sta dve komponenti, ki se pogosto uporabljata v okoljih mikrostoritev za omogočanje zanesljive, razširljive in učinkovite komunikacije med storitvami.

Tabela 4 - Mikro arhitekturni gradniki

3. Popis uporabljenih tehnologij

Arhitekturno gre za spletno več-nivojsko aplikacijo brez gradnikov na strani odjemalca z izjemo brskalnika (podprti so vsi glavni brskalniki, brez posebnih vtičnikov).



Slika 3 - Arhitektura komponent sistema po nivojih

3.1. Predstavitveni nivo

Uporabljena je tehnologija Angular 17+ ki predstavlja preizkušeno rešitev posebej primerno za zahtevne informacijske sisteme. Gre za tehnologijo, ki v celoti sloni na delovanju brskalnika (HTML5). Deluje v celoti znotraj brskalnikov (podprti so najbolj popularni brskalniki zadnjih različic), zato je sistem bolj neodvisen od operacijskega sistema končnega uporabnika oziroma naprave in zahteva manj (nič) nastavitvev okolja uporabnika. Takšen sistem je mogoče v polni funkcionalnosti uporabiti tudi na sodobnih mobilnih napravah.

Aplikacija deluje kot aplikacija v brskalniku (single page application – SPA), kar predstavlja boljši izkoristek strežniške opreme, saj strežniška oprema aplikacijo posreduje uporabniku kot statično vsebino, podatke pa aplikacija pridobi preko REST protokola. Sporočila s strežnika proti aplikaciji (na primer o zaključeni obdelavi, novih podatkih in podobno) sistem posreduje z uporabo WebSocket (nadgradnja protokola http). Sporočila niso vrste zanesljive dostave (reliable messaging), zato ni potrebna namestitev strežniških sporočilnih vrst. Takšna zasnova skupaj z nalaganjem modulov na zahtevo (lazy load) omogoča uporabo kompleksne aplikacije tudi v brskalnikih, ki imajo močno omejene kapacitete (starejši računalniki). Kljub vsemu je treba poudariti, da zaradi zastarelih šifrirnih mehanizmov brskalnikov na zastarelih operacijskih sistemih (na primer Windows XP, Windows Vista), ki jih proizvajalci ne posodablajo več, sistem na takšnih operacijskih sistemih ne bo deloval.

Več o varnostnih mehanizmih v poglavju 6, tu le povzetek: Aplikacija ob zagonu preveri, ali ima uporabnik na voljo žeton za varovanje sistema. Praviloma takšnega žetona nima (razen, ko odpre

nov listič v brskalniku), zato ga aplikacija preusmeri na centralni avtentikacijski sistem Keycloak (ki tako služi tudi kot SSO). Po zaključeni avtorizaciji (bodisi interno v Keycloak ali po posredovanju na zunanjo storitev) Keycloak uporabnikov brskalnik preusmeri nazaj na aplikacijo z uporabo globokih povezav (deep link) in z žetonom za varovanje sistema. Tako uporabnik lahko transparentno dostopa do konkretne vsebine v sistemu (in ne izključno do začetne strani). Aplikacija ob vsakem zahtevku do poslovnega nivoja poda žeton kot del zahtevka. Ker ima žeton izredno kratko veljavnost (5 minut), aplikacija pred vsakim dostopom preveri starost žetona in po potrebi (če je starejši od štirih minut) na Keycloak izda zahtevek za osvežitev žetona (z uporabo REST klica). Tako uporabniku ni treba izvajati ponovne prijave vsakih 5 minut, sistem pa je varovan z vedno svežim žetonom. Zaradi časovne omejenosti trajanja žetona je pomembna časovna usklajenost uporabnikovega sistema s centralnim sistemom. Uporabnika ob odstopanju ure o tem obvestimo.

Poslovni nivo in seje na podatkovni zbirki bosta v celoti izdelana kot stateless (poslovni nivo ne hrani stanja seje), kar olajša razširljivost sistema in bistveno zmanjša pomnilniški odtis uporabnika na strežniku.

Oblikovanje aplikacije temelji na responsive design, tako da se aktivno prilagaja napravi uporabnika.

Sistem za analitiko in SSO (Keycloak) so odprtokodne rešitve, katerih uporabniški vmesnik temelji na JSF tehnologiji.

Uporabljeni so standardi http, REST, HTML5, WebSocket, Angular 17+, TypeScript, JSON, Javascript.

Specifikacije uporabljenih tehnologij:

- http: <https://tools.ietf.org/html/rfc2068>
- http vsebina (REST): <https://tools.ietf.org/html/rfc7231>
- WebSocket: <https://tools.ietf.org/html/rfc6455>
- HTML5: <https://www.w3.org/TR/html5/>
- Angular 17: <https://angular.io/docs/ts/latest/>
- TypeScript: <https://www.typescriptlang.org/docs/>
- JSON: <https://tools.ietf.org/html/rfc7159>
- Javascript: <https://www.w3.org/standards/webdesign/script>

3.2. Poslovni in integracijski nivo

Poslovni nivo temelji na Jakarta EE standardu, izvaja se nad aplikacijskim strežnikom družine JBoss (WildFly 29). Predvidena je implementacija delilnika bremen na strojni stikalni opremi. Upravljavalec bo lahko dinamično povečal in zmanjšal število zalednih implementacij istega sklopa sistema z uporabo Docker Swarm storitev. Strežba statičnih vsebin (html, javascript in slik) ni predmet arhitekture in jih upravljavalec izvaja sam na podlagi navodil. Prav tako usmerjanje prometa na zaledne API sisteme na podlagi url (context). Zato Apache ali NGINX strežniki v okolju naročnika niso predvideni (v primeru, da na posamezni postavitvi takšna implementacija ni mogoča, je predvidena uporaba NGINX). Takšen sistem omogoča izvedbo nadgradnje med delovanjem brez izpadov, tako imenovan rolling update (odvisno tudi od vrste spremembe).

Poslovna logika je izvedena kot stateless EJB storitev z REST vmesnikom in zaščito (avtentikacijo) z žetoni. Žetone s kratkim rokom trajanja (5 minut) izdaja centralni avtentikacijski sistem Keycloak, ki je lahko postavljen v obliki grozda. Keycloak v produkcijski postavitvi le izjemoma sam izvaja avtentikacijo, čeprav je to v celoti podprto. Praviloma posreduje avtentikacijski zahtevek zunanji storitvi, ki bo v tem primeru sistem SI-CAS (v ozadju samodejna povezava z Varnostno shemo). Poslovna logika ne izvaja nobenega prometa v smeri Keycloak za namene preverjanja sej. Veljavnost žetona se preverja izključno na podlagi veljavnosti podpisa v žetonu, ki ga od Keycloak prejme brskalnik uporabnika. Tako ima Keycloak privatni ključ (nastane ob namestitvi Keycloak in ga je mogoče spremeniti). Vsaka postavitev poslovne logike ima med nastavitvami tudi javni ključ s

katerim preverja ustreznost žetona in veljavnost. Zato je izrednega pomena usklajenost ur (čas) med vsemi instancami poslovnega dela sistema (nujna uporaba NTP ali primerljivega sistema). Pri tem je razumno odstopanje do 15 sekund. Uporabniški vmesnik je zavezan, da ne posreduje žetonov, ki jim bo veljavnost potekla čez manj kot 60 sekund. Za pot do poslovnega nivoja je preostalih 45 sekund dovolj. Trajanje samega zahtevka na poslovnem nivoju na to nima vpliva, saj se veljavnost žetona na poslovnem nivoju preverja takoj na začetku sprejema zahtevka. EJB storitev preverja veljavnost žetona implicitno (brez programske kode izvajalca), vloge pa deklarativno. Tako REST zahtevek, ki ni ustrezno avtentificiran in avtoriziran, ne pride do metode, ki jo naslavlja, temveč je zavržen na nivoju JBoss modula Keycloak.

EJB storitev za dostop do podatkovne zbirke uporablja JDBC (z uporabo Oracle JDBC gonilnika), pred uporabo JDBC sistem za izbrane podatke (pogosto uporabljeni šifranti, avtorizacijske informacije o uporabniku) poskusi pridobiti podatke iz predpomnilnika EJB storitve. Kadar podatkov v predpomnilniku ni, izvede sistem klic pripravljenega ukaza (prepared statement) brez lepljenja SQL stavkov (uporabljen je parameter binding). To zagotavlja varnost pred SQL injection. Za namene transakcijske obdelave podatkov so uporabljene PL/SQL procedure in funkcije. Za CRUD operacije (branje in pisanje) bazni uporabnik ne potrebuje dostopa do tabel ali pogledov. Za iskanje in analitiko so uporabljene metode gradnje poizvedbe na podlagi resource datotek s predpripravljenimi poizvedbami in segmenti poizvedb katerim se iskalne podatke podaja izključno v obliki parametrov, kar preprečuje SQL injection. Te poizvedbe nimajo neposrednega dostopa do baznih tabel, ampak dostopajo le do baznih pogledov z omejenim naborom stolpcev in vrstic.

Storitev ima minimalen nabor pravic za dostop do PL/SQL paketov in v primeru iskalnikov in analitike tudi pogledov. PL/SQL paketi in pogledi so domensko izdelani, kar pomeni, da so namenjeni posamezni funkcionalnosti, ki jo tudi ustrezno ščitijo (minimalen nabor stolpcev in vrstic). Omejevanje vrstic se izvaja na podlagi prijave na podatkovno zbirko (s tem je enolično določeno za katero storitev gre in iz katerega varnostnega območja je vzpostavljena povezava). DMZ modul vidi samo vrstice »zunanjih« uporabnikov. Omejevanje je izvedeno tudi na podlagi uporabnikovih vlog. Ta zaščita se izvaja na podatkovni zbirki vendar na podlagi informacije, ki jo posreduje poslovni nivo. Če ta informacija ni podana, bazni uporabnik ne vidi podatkov, čeprav ima dostop do metode.

Integracijski del (storitev Integ) za izhodne klice je implementiran nad standardom Apache Camel (nad množica Apache CXF), ki podpira obilico dostopnih točk, med drugim tudi spletne storitve, SOAP, JMS, SMTP. Gre za zrel sistem integracij, ki omogoča spremljanje prek različnih metod vpogleda (podatkovna zbirka, JMX, REST, EJB itd.) in gradnjo integracij v obliki obdelovalne linije (kakor izgradnja avtomobila na proizvodni liniji). Ta sistem je namenjen namenskim integracijam, ko sistem IS Dovoljenja kliče druge organizacije, ki za določen namen in vsebino pripravijo sporazum in skupni standard povezave.

Vhodne integracije, ko drugi informacijski sistemi kličejo IS Dovoljenja, se izvajajo preko REST klicev na mikrostoritve sistema IS Dovoljenja. Zunanjim sistemom bo dodeljen Keycloak uporabnik, kateremu bodo preko Varnostne sheme dodeljene vloge s katerimi bo lahko dostopal samo do metod, ki jih potrebuje za integracijo. Keycloak uporabniku kreira žeton za brezpristopno uporabo, ki ga uprabi za kreiranje aktivnega žetona za dostop, ki ima omejeno veljavnost. Dostopne točke, ki bodo namenjene dostopu zunanjih sistemov, bodo ustrezno dokumentirane s standardom Microprofile OpenAPI.

Za namene komunikacije med mikrostoritvami, ko morajo sporočilo prejeti vse instance mikrostoritve in ne le naključno izbrana, se uporablja Kafka v kombinaciji z Zookeeper-jem, ki je potreben za delovanje Kafke.

Apache Kafka je odprtokodni sistem za obdelavo pretoka podatkov, ki nudi visoko prepustnost za realnočasovno obdelavo podatkov. Kafka je platforma, ki omogoča publikacijo in naročanje na tokove (streams) zapisov, podobno kot sporočilni sistem. Kafka je prilagodljiva in omogoča visoko stopnjo zaupanja pri dostavi sporočil, kar je ključno v sistemu, kjer je treba obvestiti vse instance posamezne storitve.

Zookeeper je koordinacijski servis za razdeljene aplikacije, ki omogoča visoko zanesljivost. Omogoča funkcionalnosti, kot so shranjevanje konfiguracij, imenovanje servisov, distributirane sinhronizacije in skupin, omogoča pa tudi predvolilne mehanizme. Kafka za svoje delovanje potrebuje Zookeeperja za upravljanje in koordiniranje porazdeljenih operacij. Zookeeper na primer skrbi, da so podatki pravilno replicirani med Kafko, prav tako pa tudi usklajuje in nadzira porazdeljene operacije znotraj Kafke.

Za monitoring storitev se uporabi klijžnico MicroMeter (<https://micrometer.io/>), ki je del strežnika Wildfly in bo za vsako posamezno instanco storitev pripravljala statistične podatke o uporabi in času trajanja. Statistične podatke instanc se bodo zbirale v sistemu Prometheus (<https://prometheus.io/>), ki je del infrastrukture DRO. Iz storitve Prometheus bomo potem lahko podatke prikazovali na strani za administracijo, ki bo del sistema IS Dovoljenja, ali pa v storitvi Grafana (<https://grafana.com/>), ki je prav tako del infrastrukture DRO.

Za namene izdelave dokumentov na podlagi predlog bo uporabljeno orodje Docx4j.

Za namen izdelave PDF dokumentov bo uporabljena odprto kodna knjižnica itext (<https://itextpdf.com/>).

Obvestila na elektronski naslov bodo posredovana z uporabo SMTP protokola.

Uporabljeni so standardi http, REST, JSON, WebSocket, Jakarta EE, MicroProfile, EJB, JAX-RS 2.1, SQL.

Specifikacije uporabljenih tehnologij:

- http: <https://tools.ietf.org/html/rfc2068>
- http vsebina (REST): <https://tools.ietf.org/html/rfc7231>
- WebSocket: <https://tools.ietf.org/html/rfc6455>
- JSON: <https://tools.ietf.org/html/rfc7159>
- Jakarta EE 10: <https://jakarta.ee/specifications/>
- MicroProfile 6.0: <https://download.eclipse.org/microprofile/microprofile-6.0/microprofile-spec-6.0.pdf>
- EJB 3.1: <https://jcp.org/en/jsr/detail?id=318>
- JAX-RS 2.1: <https://jcp.org/en/jsr/detail?id=370>
- SQL-92: <http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt>

3.2.1. Kontejnerji

Uporaba kontejnerjev (v konkretnem primeru Docker Swarm) je ključen element arhitekture IS Dovoljenja. Kljub temu sistem lahko deluje tudi brez kontejnerjev. Vendar šele z uporabo kontejnerjev pridejo do izraza odločitve pri gradnji arhitekture kot so stateless storitve, sistem enotne prijave (SSO), delitev bremena (load balancing), skalabilnost (razširljivost), odpornost (resilience), tekoče nadgradnje (rolling updates) itd.

Kapaciteta sistema (procesor, pomnilnik, disk) se združi (ne glede na število fizičnih ali virtualnih strežnikov) v eno storitveno gručo (swarm cluster). V to gručo glede na izbrano politiko upravljaavec namešča storitve. Posamezna storitev vzpostavi vsaj eno instanco slike (image) implementacije, po potrebi pa tudi več in to je mogoče dinamično prilagajati obremenitvam sistema (scaling). Upravljaavec določi katera verzija posamezne storitve naj bo v uporabi in v koliko instancah, sistem poskrbi za ostalo.

V primerjavi s klasičnimi metodami upravljanja s sistemi ima kontejner poleg avtomatizacij še eno prednost. Kontejnerji so bistveno manjši od virtualnih strežnikov. Tako danes virtualni strežnik v produkcijskem okolju dosega 16GB pomnilnika. Kontejnerji lahko živijo s 4GB, dokler upravljalavec po potrebi (začasno) doda še kakšno instanco.

Ta dinamičnost se zdi v prvi vrsti zabavna in v drugi ekološka. V resnici pa omogoča boljše uporabniško izkušnjo. Za primer vzemimo monolitno aplikacijo, ki prav tako podpira razširljivost (scaling). Takšna aplikacija zaseda 16GB pomnilnika in vzpostavitev dodatne instance zahteva dodatnih 16GB. Ko je obremenjen le posamezen del sistema (na primer obdelava vlog), ne pa ostali deli, smo še vedno le pri dveh instancah. Poleg tega je moral upravljalavec pripraviti še en virtualen strežnik (ali spremeniti obstoječega) in dopolniti nastavitve spletnega strežnika. Kontejnerji se tu izkažejo (če so pravilno implementirani). Dodatna instanca vzame do 4GB pomnilnika (privzeto 1GB), kar pomeni, da v dodatnih 16GB pomnilnika lahko namestimo še vsaj 4 instance in jih imamo tako 5 v primerjavi z dvema v monolitni izvedbi. In to le z izvedbo ukaza ali klikom na uporabniškem vmesniku. Storitve ima namreč en skupen spletni naslov, ne glede na število instanc, ki storitvi strežejo.

Kontejnerji poskrbijo tudi za omrežno ločitev v primeru velikih podatkovnih centrov. To omogoča, da MDP v DRO namesti en Docker swarm grozd, ki ga uporabi za različne sisteme. Omrežje lahko popolnoma loči s čimer zagotovi varnost sistemov pred vdori v sisteme nameščene v isti grozd.

Kontejnerji pa prinašajo tudi izzive. Ker ima vsako okolje svoje posebnosti in ker je nameščanje virtualnih strežnikov (tudi kontejnerjev), ki jih bo izdelal zunanji izvajalec, varnostno sporno, je potrebno zagotoviti, da se slika (image) pripravi transparentno. Torej, da vsebuje le tiste datoteke in nastavitve, ki so v danem primeru smiselne in ne vnašajo varnostnih ranljivosti ali nestabilnosti. Končno sliko (image) zato izdela upravljalavec sam, na podlagi besedilne specifikacije (Docker datoteka). Tako lahko prilagaja posamezne nastavitve kontejnerja, nameščene knjižnice in pakete, predvsem pa ohranja nadzor nad končno izdelano sliko. Takšna slika ne vsebuje specifičnih omrežnih nastavitvev ali uporabniških imen in gesel. Še vedno je dovolj generična, da jo je mogoče namestiti v testno okolje in po potrditvi ustreznosti tudi v produkcijsko. Razlike med navedenimi okolji so podane v fazi priprave storitve.

3.3. Podatkovni nivo

Za potrebe IS Dovoljenja se na infrastrukturi MDP postavi ločena podatkovna zbirka v kateri bodo shranjeni vsi podatki, ki jih IS Dovoljenja potrebuje (razen tistih, do katerih bo IS Dovoljenja dostopal preko spletnih servisov).

Podatkovna zbirka sistema je Oracle Database 19.0.0.0.0 z zadnjim PSU, ki je nameščena v DRO. Zaradi lažjega upravljanja z večjo količino podatkov bo uporabljena opcija Partitioning. Opcija se uporabi na področju večjih sklopov podatkov smiselno tako, da zmanjša pritisk na pregledovanje velikih tabel in (ob večjih brisanjih) na podatkovne dnevnike, ki se pretakajo na oddaljeno lokacijo. Dodatno ima podatkovna zbirka nameščene opcije Spatial in Oracle Text.

Uporabljen je standard SQL.

SQL-92: <http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt>

3.4. Dokumentni nivo

IS Dovoljenja bo končne verzije dokumentov odlagal v dokumentni sistem KRPAN, ki je že v uporabi na MOPE. Gre za horizontalna rešitev, ki podpira evidentiranje in vodenje splošnih in upravnih zadev ter dokumentnih seznamov (sistem za pisarniško poslovanje). Dejanska hramba pa je v CEH - Centralna hramba gradiva, ki omogoča zakonsko skladno in revizijsko varno dolgoročno hrambo elektronskega gradiva.

Integracija med sistemom KRPAN in IS Dovoljenja bo speljana preko aplikacijskega modula Integ z uporabo SOAP protokola.

Delovne verzije dokumentov bodo shranjene na MS Sharepoint, ki je že uporabljen na MOPE. Za branje in pisanje dokumentov na MS Sharepoint bo narejena integracija preko Sharepoint REST API.

V IS Dovoljenja bo implementirana funkcionalnost elektronskega podpisovanja dokumentov, za kar bo uporabljen sistem SI-CES (spletno podpisovanje).

Uporabljeni sta standarda XML in ISO 19005-1 (PDF/A):

- XML 1.1: <https://www.w3.org/TR/2006/REC-xml11-20060816/>
- ISO 19005-1 (PDF/A): <https://www.iso.org/standard/38920.html>

4. Specifikacije aplikacije za prikaz podrobnosti delovanja vseh vključenih komponent

IS Dovoljenja je z vsebinskega, tehnološkega in infrastrukturnega vidika relativno kompleksen projekt ker med ostalim vsebuje dvojno postavitvev (interna in zunanja postavitvev) ter več različnih integracij z zunanjimi sistemi. Kompleksnost sistema poleg načrtovanja prinaša izzive tudi na področju upravljanja s sistemom. Upravljanje se pogosto razume predvsem na infrastrukturnem področju kar v primeru IS Dovoljenja ne drži.

Nadzor nad delovanjem vseh vključenih komponent je razdeljen na nadzor nad povezavami z zunanjimi storitvi, nadzor nad delovanjem produktov (odprto-kodnih) ter nadzor stanja sistema v celoti (vseh gradnikov).

4.1. Nadzor nad integracijami z zunanjimi sistemi

IS Dovoljenja se integrira z več različnimi zunanjimi sistemi preko klicev spletnih storitev (popis storitev je v poglavju 7).

Za posamezno integracijo se implementira sistem za spremljanje izvanja v obliki evidentiranja vseh izvedenih klicev in prejetih odgovorov (brez vsebine) na baznem nivoju. Za pregled evidentiranih podatkov se lahko implementira pregledovalnik v okviru nadzornega modula IS Dovoljenja (modul Upravljanje sistema).

4.2. Produktni nadzorni sistemi

V IS Dovoljenja so uporabljeni ali vgrajeni produkti (odprto-kodni), ki imajo svoje nadzorne sisteme:

- Wildfly Application Server: web console
- Keycloak: Keycloak Admin Console.

Vgrajene produkte se ne prilagaja po nepotrebnem (le v okviru zahtev postavitve v okolje naročnika). Na ta način je mogoče lažje:

- Načrtovati in izvajati nadgradnje in
- Izpostavljati elemente kot nove horizontalne gradnike.

Vsi nadzorni sistemi so nameščeni tako, da so dosegljivi izključno iz internega omrežja HKOM. Keycloak nadzorni sistem je dosegljiv preko skupnega naslova IS Dovoljenja, Wildfly Application Server je dosegljiv izključno iz sistemskega omrežja.

Specifikacija nadzornih sistemov produktov je izven domene tega projekta. Uporabljena je ustrezna različica nadzornega sistema glede na različico produkta.

4.3. Nadzor nad stanjem sistema v celoti

Za potrebe spremljanja delovanja instanc bomo uporabili sistema Prometheus (<https://prometheus.io/>) in Grafano (<https://grafana.com/>), ki sta del infrastrukture DRO v povezavi s knjižnico MicroMeter (<https://micrometer.io/>) in HealthCheck-i posameznih storitev.

Vsaka instanca spletnih storitev bo vsebovala HealthCheck metode, ki bodo preverjale, če je instanca delujoča in lahko dostopa do virov, ki so nujno potrebni za njeno delovanje (podatkovna baza, avtentikacija, ...). Te bodo dostopne na točno določenem endpointu, ki bo enak na vseh storitvah. HealthCheck metode bodo uporabljene s strani Docker Swarm za nadzor in rešart posamezne instance storitve, če ta ni več »živa«, pobirali pa jih bomo tudi s sistemom Prometheus, tako da jih bo mogoče prikazati na nadzorni plošči.

Poleg samega HealthCheck-a bomo na vsaki spletni storitvi uporabili tudi knjižnico MicroMeter, ki je že vključena v strežnik Wildfly. Poleg osnovnih metrik, ki jih izvaja MicroMeter na strežniku Wildfly (poraba pomnilnika, CPU poraba, ...) bomo implementirali še vsaj naslednje:

- Števec za posamezen zahtevek (request),
- Čas trajanja zahtevka,
- Števec za posamezen bazni klic znotraj zahtevka,
- Čas trajanja baznega klica.

Metrike za posamezno storitev bodo na voljo na "/metrics".

Za zbiranje metrik in zdravja instanc storitev bomo uporabili sistem Prometheus. Ta bo za instance vseh storitev zbiral metrike in zdravje storitve, ki je implementirano s HealthCheck metodo. V primeru nedelovanja instanc določene storitve bo v sistemu Prometheus mogoče skonfigurirati obveščanje administratorjev sistema. Metrike, ki jih bo zbiral Prometheus bo mogoče prikazati neposredno v spletni aplikaciji sistema IS Dovoljenja (modul Upravljanje sistema) ali pa za prikaz le teh uporabiti sistem Grafana.

Zunanja postavitve bo zbirala metrike iz instanc storitev, ki bodo nameščene na zunanji postavitvi, notranja pa bo poleg instanc storitev iz notranje postavitve pobirala tudi metrike iz Prometheusa, ki bo nameščen na zunanji postavitvi, tako da bomo v sistemu za prikaz imeli na razpolago metrike iz celotnega sistema, kot je razvidno na sliki arhitekture (Slika 1 - Storitvene povezave med notranjimi in zunanjimi moduli s protokoli povezav, reverse proxy nivoji so predmet odločitve upravljalca).

5. Arhitektura sistema za implementacijo

Arhitektura IS Dovoljenja je prilagojena obstoječi informacijski arhitekturi na MDP z upoštevanjem dokumentov Smernice MDP za razvoj informacijskih rešitev in Generične tehnološke zahteve za razvoj informacijskih sistemov (GTZ).

Postavitev sistema temelji na izhodiščih:

- Visoka razpoložljivost
- Dinamična zmogljivost (skalabilnost)
- Varnost (šifrirane povezave)

-
- Odpornost (resilience)
 - Brez stanja (stateless)
 - Prilagodljiva uporabniška izkušnja (responsive design).

Sistem je zasnovan na način, da ima ločeno postavitve v internem okolju (HKOM) ter ločeno postavitve za zunanji del (DMZ). Podatkovna zbirka bo v internem okolju (HKOM) in bo skupna za vse sklope (module) postavljene v internem in v okolju DMZ.

Sistem je funkcionalno razdeljen na 9 sklopov (modulov), ki jih upravljavec samostojno namesti v različne dele omrežja in imajo glede na uporabniške skupine tudi ločene prijave na podatkovno zbirko za isto funkcionalnost. Povedano drugače: funkcionalni sklop, ki se uporablja v javnem omrežju, je nameščen ločeno od iste implementacije, ki se uporablja znotraj HKOM. Obe namestitvi imata ločeni prijavi na podatkovno zbirko. Na podatkovni zbirki je implementirana omejitev, ki glede na prijavo (uporabnika na podatkovni zbirki) omejuje dostop in spreminjanje podatkov v grobem smislu tako, da lahko v najslabšem primeru uporabnik z javnega omrežja vidi in spreminja zgolj podatke, ki so dosegljivi uporabnikom z javnega omrežja.

Predvideni so ločeni API-ji za posamezne funkcionalne sklope in sicer:

- Vloge (HKOM in DMZ)
- Upravni postopki (HKOM)
- Šifranti (HKOM)
- Poročila (HKOM)
- Upravljanje sistema (HKOM)
- Integ (HKOM)
- Async (HKOM)
- Sistem za obveščanje (HKOM in DMZ)
- Kafka (HKOM in DMZ)
- Zookeeper (HKOM in DMZ)

Predvideni so naslednji vstopni naslovi v sistem (za vsakega je potrebno opraviti prijavo):

- HKOM:
 - <https://mope-dovoljenja.sigov.si>
- DMZ:
 - <https://mope-dovoljenja.gov.si/>

Administrativni sklopi (moduli) za upravljanje sistema so nameščeni zgolj v HKOM omrežju.

Zaradi stroge ločitve med internim (HKOM) in zunanjim (DMZ) okoljem velja omejitev, da uporabnik, ki ima dodeljeno uporabniško vlogo za dostope znotraj internega okolja, ne more, v primeru dostopa do sistema iz DMZ okolja (npr. od doma brez VPN povezave), uporabljati funkcionalnosti (npr. administratorske sklope), ki so dovoljene za uporabo le znotraj internega (HKOM) okolja.

Poleg varovanja na predstavitvenem in poslovnem nivoju je zadnja ovira omejevanje dostopa na podatkovni zbirki, ki bo izvedena na podlagi prijave (uporabnika na podatkovni zbirki). Za vsak aplikacijski modul (API) bo kreiran svoj datasource, ki bo imel pravice samo do nabora baznih paketov in pogledov, ki jih potrebuje za svoje delovanje. Sistemski podatki uporabnika se bodo ob inicializaciji povezave posredovali na podatkovno zbirko. Na pogledih, kjer bo potrebna omejitev vpogleda glede na pravice uporabnika, se bo ta omejeval s tehnologijo Oracle Row-Level Security.

Predvidena je implementacija delilnika bremen na strojni stikalni opremi. Zaledni sistemi so implementirani v stateless obliki. Upravljavec bo lahko dinamično povečal in manjšal število zalednih

implementacij istega sklopa sistema. Strežba statičnih vsebin (html, javascript in slik) ni predmet arhitekture in jih upravljaavec izvaja sam na podlagi navodil. Prav tako usmerjanje prometa na zaledne API sisteme na podlagi url (context). Zato Apache ali NGINX strežniki v testni in produkcijski postavitvi niso predvideni.

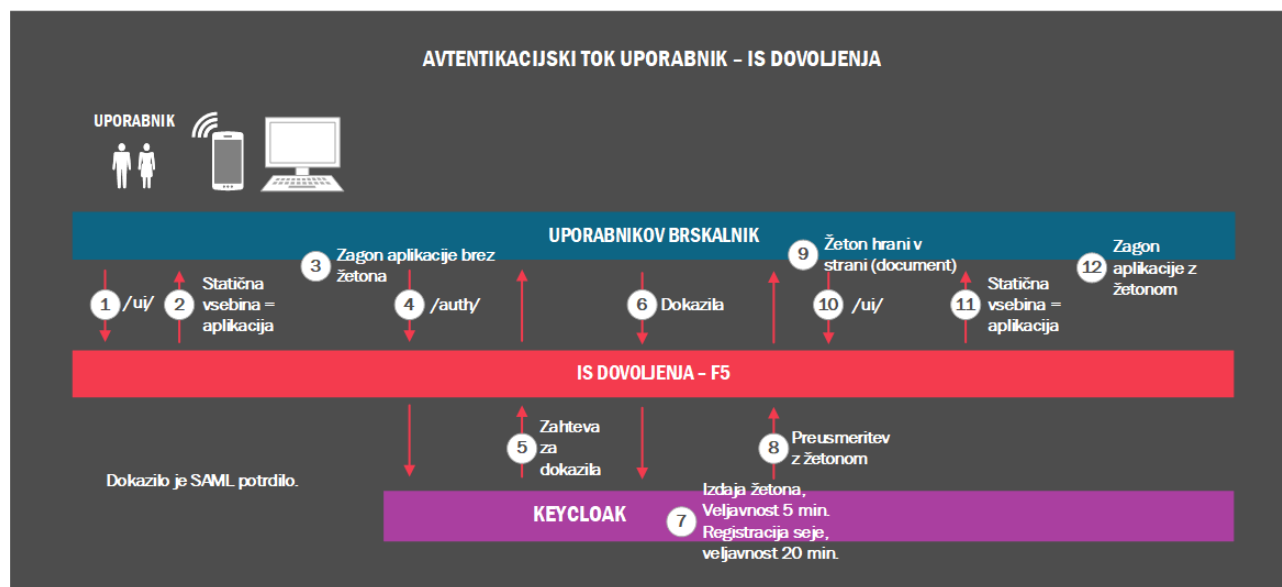
Predvidena je uporaba JBoss Wildfly 29 v obliki Docker kontejnerjev. Ne glede na to, rešitev deluje tudi brez Docker kontejnerjev, vendar zahteva ročno nameščanje in upravljanje z viri. Pri tem bodo izdelki izdelani v okolju MDP (uporaba Jenkins), tako bo MDP imel končno kontrolo nad vsebino izdelkov. Posebej to velja za docker kontejnerje, kjer velja, da bo definicija kontejnerja del SVN, kontejner izgradi Jenkins v okolju MDP in upravljaavec bo lahko docker definicijo dopolnil ali popravil in to oddal v SVN. Na tak način bo produkcijsko, testno in razvojno okolje bolj usklajeno. Varnostno občutljivi podatki (na primer gesla, ključi ipd.) niso predmet docker definicije, temveč jih upravljaavec določa ob namestitvi. Enako velja za vse mrežne nastavitve in načrtovane razlike med okolji (testno, šolsko, produkcijsko).

5.1. Avtentikacija uporabnikov

5.1.1. Tehnološka zasnova avtentikacije

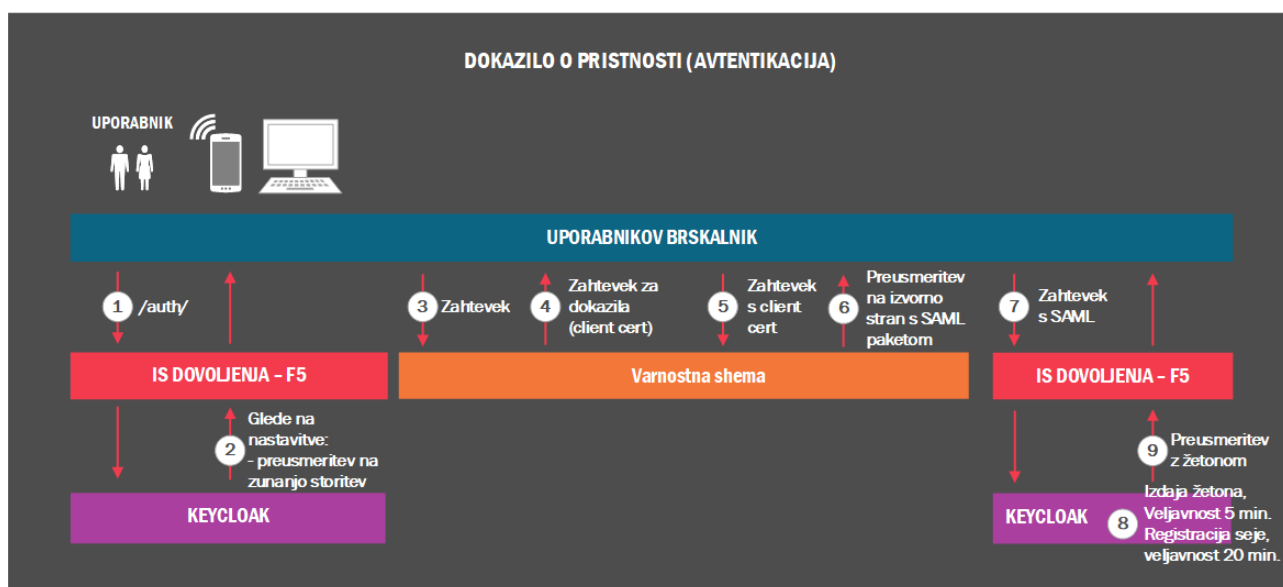
Avtentikacijski sistem temelji na izključni uporabi sistema SI-CAS (v ozadju samodejna povezava z Varnostno shemo) in ne predvideva možnost prijave v sistem z uporabniškim imenom (ali elektronskim naslovom) in geslom.

Potek prijave (authentication flow) v razmerju med uporabnikovim brskalnikom in IS Dovoljenja je prikazan na spodnji sliki.



Slika 4 - Avtentikacijski tok uporabnika

Glede na zgornjo sliko se po koraku 5 izvede preusmeritev na sistem SI-CAS (v ozadju samodejna povezava z Varnostno shemo), kjer uporabnik izkaže svojo identiteto, v zgornjo sliko se vrne v koraku 6. Na spodnji sliki tem točkam ustrezata točki 1 in 7.



Slika 5 - Dokazilo o pristnosti zagotovi SI-CAS; velja za uporabnike v HKOM in DMZ

Veljavnost žetona je omejena na 5 minut. Daljše trajanje bi onemogočalo centralno upravljanje s sistemom, ker na primer uporabniku lahko odvzamete vse pravice, vendar bi uporabnik, ki je že prijavljen z aktivno uporabo sistema izvajal naloge v nedogled. Žeton se zato v aplikaciji (v ozadju) osvežuje, dokler je njegova prijava veljavna. Proces je prikazan na spodnji sliki.



Slika 6 - Osveževanje dostopnega žetona

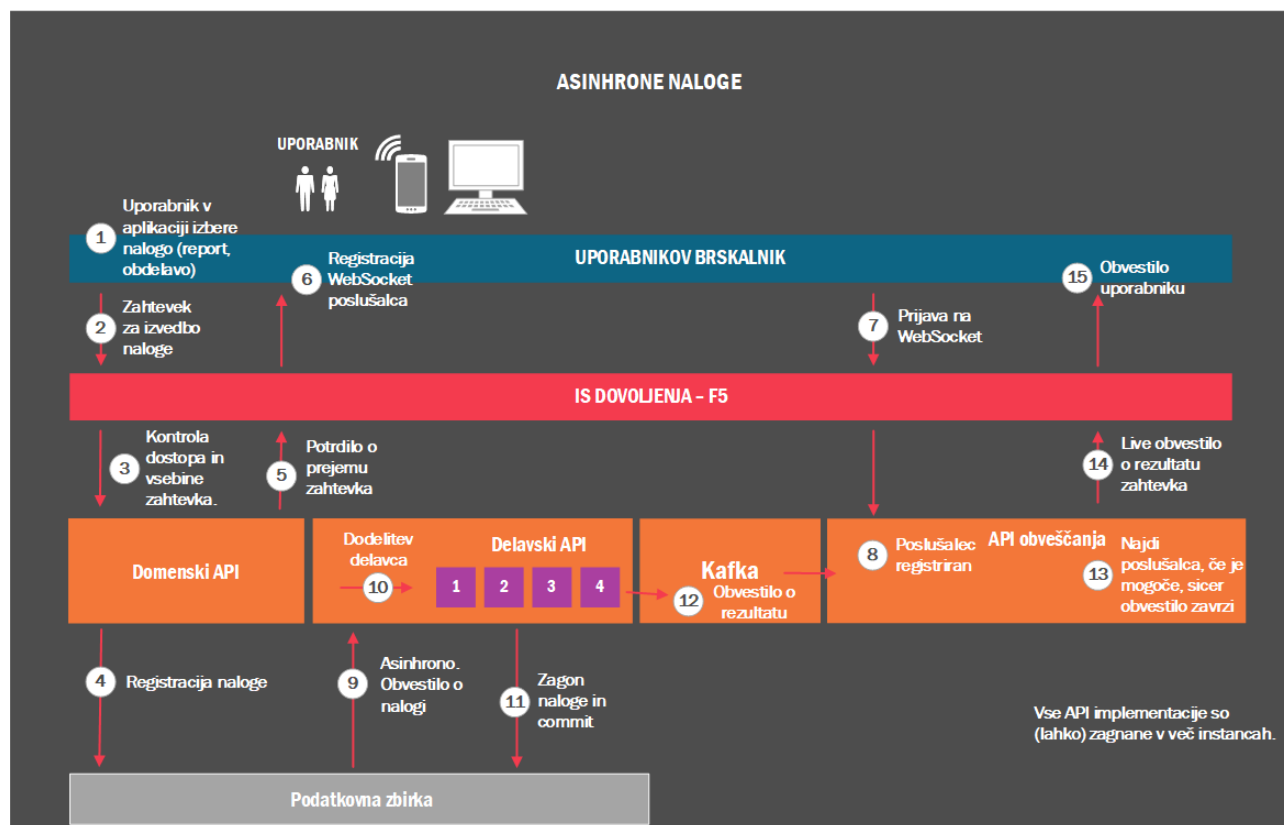
V okviru IS Dovoljenja se implementira ločena statična aplikacijska stran na katero je uporabnik preusmerjen v primeru, ko ob poskusu prijave sistem ugotovi, da ni registriran v sistemu (SI-CAS/Varnostni shemi) oziroma nima ustreznih vlog za dostop do sistema. Na tej strani uporabnik prejme obvestilo, da ni registriran ter navodila oz. povezavo do strani za registracijo v sistem.

5.2. Obdelave podatkov na zahtevo uporabnika

V IS Dovoljenja se izvaja množica obdelav. Večina je neodvisna od zahtev uporabnikov (časovno proženje), veliko pa je takšnih, kjer uporabnik sproži akcijo, ta akcija pa traja dlje od nekaj sekund. V vseh takšnih primerih je implementiran asinhron postopek.

V primerih, kjer je posledica akcije sporočilo enkratne vrednosti (na primer poročilo), se rezultat takšne akcije ne shrani na podatkovno zbirko, temveč se posreduje uporabniku. V drugih primerih (na primer primerjava sprememb v podatkih zunanjega vira) je obvestilo uporabniku manj pomembno, bistven je rezultat obdelave. V obeh primerih je posredovanje obvestila uporabniku operacija, ki jo je dovoljeno zavreči. Zato se implementira lažja (tanjša) oblika komunikacije, ki manj obremenjuje sistem. Kadar bo obvestilo treba dostaviti s preverljivo gotovostjo (potrditev uporabnika o prebranem obvestilu), bo to obvestilo posredovano preko elektronske pošte ali v aplikaciji v profil uporabnika. WebSocket ne izvaja poizvedovanja (polling), temveč le čaka na naslednje obvestilo (če obvestil ni, ni prometa).

Za namene asinhronne obdelave je sestavljen sistem delavskih API-jev, od katerih ima vsak (če so nameščeni v več instancah) 4 delavske procese. Tem delo dodeljuje usmerjevalnik, ki je del takšnega API. Vsak delavski proces je sposoben izvesti katerokoli nalogo (večina je implementirana v obliki stored procedur). Takšna razdelitev omogoča omejitev pritiska obdelav na celoten sistem. Obdelave niso implementirane v API-jih, ki strežejo uporabniškemu vmesniku. Rezultat je prikazan na spodnji sliki.



Slika 7 - Izvajanje asinhronih nalog

6. Varnostni in zaščitni mehanizmi

Varnostni in zaščitni mehanizmi so bistveni del vsakega sistema, tako tudi IS Dovoljenja, ker zagotavljajo varnost na vseh nivojih (arhitekturni, sistemski, podatkovni).

Opisi varnostnih in zaščitnih mehanizmov so navedeni tudi v poglavjih 2, 3 in 5. V nadaljevanju so popisani vsi mehanizmi, ki se uporabljajo znotraj IS Dovoljenja.

6.1. Arhitekturni nivo

Sistem je zasnovan na način, da ima ločeno postavitve v internem okolju (HKOM) ter ločeno postavitve za zunanji del (DMZ). Podatkovna zbirka bo v internem okolju (HKOM) in bo skupna za za sklope (module) postavljene v internem in v okolju DMZ.

Sistem je funkcionalno razdeljen na več sklopov (modulov), ki jih upravljavec sistema samostojno namesti v različne dele omrežja in imajo, glede na uporabniške skupine, tudi ločene prijave na podatkovno zbirko za isto funkcionalnost. Povedano drugače: funkcionalni sklop (API), ki se uporablja v zunanjem okolju (DMZ), je nameščen ločeno od iste implementacije sklopa (API), ki se uporablja znotraj internega okolja (HKOM). Obe namestitvi API-ja imata ločeni prijavi na podatkovno zbirko. Na podatkovni zbirki je implementirana omejitev, ki glede na prijavo (API iz katerega se dostopa do podatkov) omejuje dostop in spreminjanje podatkov v grobem smislu tako, da lahko v najslabšem primeru uporabnik iz zunanjega okolja (DMZ) vidi in spreminja zgolj podatke, ki so dosegljivi uporabnikom zunanjega okolja (DMZ).

Administrativni sklopi (moduli) za upravljanje sistema so nameščeni zgolj v internem okolju (HKOM). Poleg varovanja na predstavitvenem in poslovnem nivoju je zadnja ovira omejevanje dostopa na podatkovni zbirki, ki bo izvedena na podlagi prijave (uporabnika na podatkovni zbirki).

Zaradi stroge ločitve med internim (HKOM) in zunanjim (DMZ) okoljem velja omejitev, da uporabnik, ki ima dodeljeno uporabniško vlogo za dostope znotraj internega okolja, ne more, v primeru dostopa do sistema iz DMZ okolja (npr. od doma brez VPN povezave), uporabljati funkcionalnosti (npr. administratorske sklope), ki so dovoljene za uporabo le znotraj internega (HKOM) okolja.

6.1.1. Zaščita komunikacijskih kanalov

IS Dovoljenja ne implementira zaščite komunikacijskih kanalov, ampak to funkcijo izvaja DRO. Uporabljena je šifrirana varna povezava (SSL) med uporabnikovim brskalnikom in dostopno točko v okviru DRO. Sistem uporablja enotno dostopno točko (zmanjševanje profila za napad).

Povezave med dostopno točko in storitvami niso šifrirane, lahko pa se upravljavec sistema ob postavitvi odloči drugače ter s tem seznani naročnika.

Uporaba sodobnih sistemov šifriranja (algoritmi in dolžina ključev) onemogoča uporabo starejših brskalnikov, ki niso več vzdrževani s strani proizvajalcev.

6.2. Sistemski nivo

Vsi elementi IS Dovoljenja zaupajo Keycloak gruči. Vsak uporabnik (tudi klienti storitev API) se avtentificira na Keycloak. Keycloak izda žeton, ki je podpisan, kar je veljavno dokazilo za vse aplikacijske strežnike. Keycloak je redko samostojen v izvedbi avtentikacije (le kadar bi bil uporabljen mehanizem username/password). Praviloma zahtevo za avtentikacijo posreduje tretjim gradnikom (v primeru IS Dovoljenja sistemu SI-CAS (v ozadju samodejna povezava z Varnostno shemo)). Sistem je namenjen registriranim uporabnikom, uporabniki so torej v sistem vpisani pred prvo prijavo. Registracija novega uporabnika je ločen proces, ki ga pokriva ločen administratorski sklop (modul) znotraj IS Dovoljenja.

Keycloak uporablja ločeno povezavo na podatkovno zbirko. Za branje in urejanje uporabnikov, shranjevanje zgodovine prijav in dodeljenih pravic uporablja CRUD stavke (nad pogledi).

Aplikacija ob zagonu preveri, ali ima uporabnik na voljo žeton za varovanje sistema. Praviloma takšnega žetona nima (razen, ko odpre nov listič v brskalniku), zato ga aplikacija preusmeri na centralni avtentikacijski sistem Keycloak (ki tako služi tudi kot SSO). Po zaključeni avtorizaciji (bodisi interno v Keycloak ali po posredovanju na zunanjo storitev) Keycloak uporabnikov brskalnik preusmeri nazaj na aplikacijo z uporabo globokih povezav (deep link) in z žetonom za varovanje sistema. Tako uporabnik lahko dostopa do konkretne vsebine v sistemu (in ne izključno do začetne strani). Aplikacija ob vsakem zahtevku do poslovnega nivoja poda žeton kot del zahtevka. Ker ima žeton izredno kratko veljavnost (5 minut), aplikacija pred vsakim dostopom preveri starost žetona in po potrebi (če je starejši od štirih minut) na Keycloak izda zahtevek za osvežitev žetona (z uporabo klica REST). Tako uporabniku ni treba izvajati ponovne prijave vsakih 5 minut, sistem pa je varovan z vedno svežim žetonom. Zaradi časovne omejenosti trajanja žetona je pomembna časovna usklajenost uporabnikovega sistema s centralnim sistemom. Uporabnika ob odstopanju ure o tem obvestimo.

6.2.1. Omejevanje dostopa

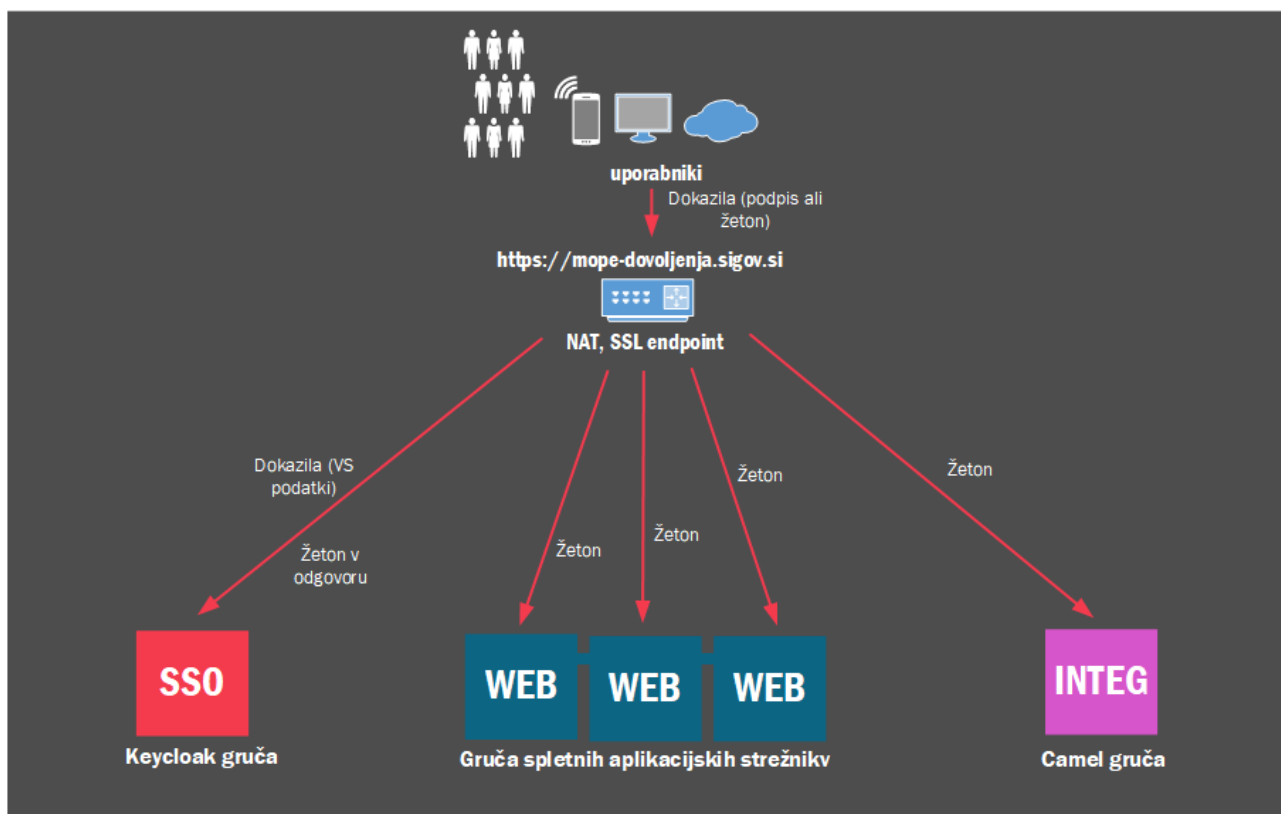
Uporabnik vidi le funkcionalnosti, ki jih ima na voljo glede na dodeljene pravice. To velja tako za menijsko strukturo, kakor funkcionalne elemente (gumbi, meniji,...) znotraj posameznega konteksta.

Prav tako nima na voljo dostopa in s tem prikaza do zapisov, do katerih nima pravic.

Vizualno omejevanje je implementirano večinoma v aplikaciji v brskalniku, zato je le dodaten element, ki pa ne predstavlja varnostnega mehanizma v smislu varovanja podatkov pred zlorabo. Omogoča pa, da uporabniku ne izpostavimo pogleda na funkcionalnosti, ki jih ne sme uporabljati in tako skrijemo tudi del sistema, ki bi ga lahko napadel.

6.2.2. Zaščita pred tretjimi osebami

Zaščita pred tretjimi osebami je zaščita pred osebami, ki niso uporabniki IS Dovoljenja.



Slika 8 - Izmenjava dokazil med uporabnikovim brskalnikom in elementi sistema IS Dovoljenja.

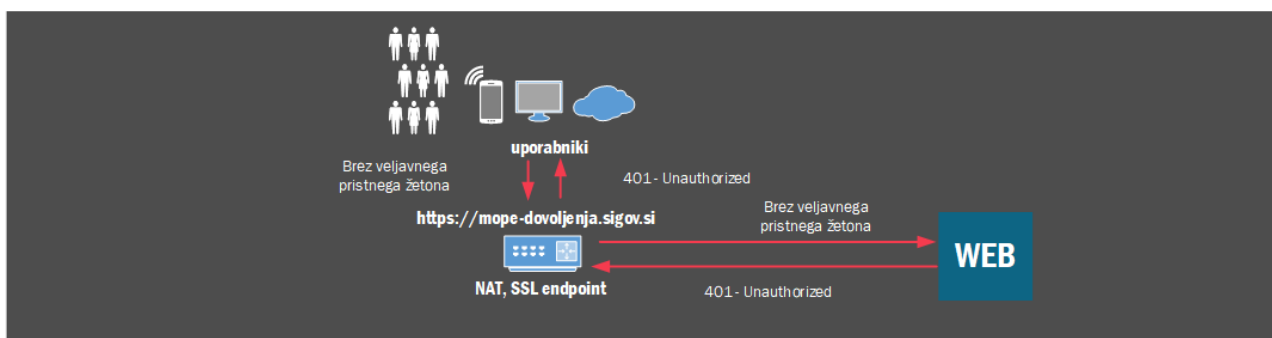
Vsaka izmed storitev IS Dovoljenja pričakuje, da uporabniška zahteva s seboj prinaša dokazilo o pristnosti. Vse storitve razen SSO podsistema Keycloak zahtevajo, da je takšno dokazilo o pristnosti veljaven OpenId žeton, ki ga izda SSO podsistem.

SSO podsistem Keycloak izda OpenId žeton na podlagi dokazil uporabnika. Dokazilo je lahko obstoječ veljaven žeton (uporabnik je že prijavljen) ali dokazilo, ki je določeno v fazi načrtovanja sistema. Keycloak podpira več načinov avtentikacije uporabnika (LDAP, Active Directory, uporabniško ime/geslo, uporabniško kvalificirano digitalno potrdilo, zunanja SSO storitev itd.). Ob tem je lahko aktivnih več načinov avtentikacije hkrati. Trenutno izbran način avtentikacije je uporaba sistema SI-CAS (v ozadju samodejna povezava z Varnostno shemo).

Žeton, ki ga izda SSO, je podpisan s privatnim ključem, ki je poznan izključno Keycloak SSO. Tako je zagotovljena avtentičnost dokazila. Žeton ima tudi izredno kratko veljavnost (5 minut). V obdobju veljavnosti žetona lahko uporabnikov brskalnik zahteva izdajo novega žetona, ki ima prav tako veljavnost 5 minut. Tako od uporabnika v času aktivnosti sistem ne zahteva ponovne prijave, po drugi strani pa je mogoče omejiti veljavnost seje v primeru sprememb ali potrebe po odklopu uporabnikov.

Vsaka izmed storitev pričakuje, da uporabniška zahteva v glavi zahtevka (Authorization header) poda žeton. Pristnost žetona preverja z uporabo javnega ključa Keycloak SSO. Storitve tako ne izvajajo nobene komunikacije s SSO. Vsebina žetona storitvi prenaša identiteto uporabnika (v obliki tehničnega identifikatorja - GUID), poleg tega pa še nastavljive podatke; privzeto ime, priimek, e-mail in vloge (roles). Če žeton ni podan ali ni pristen ali mu je potekla veljavnost, storitev odgovori s 401 – Not Authorized, brez podrobnosti. Storitve ne izdaja nikakršnih preusmeritev (redirect). To je tudi ključni element zaščite pred tretjimi osebami.

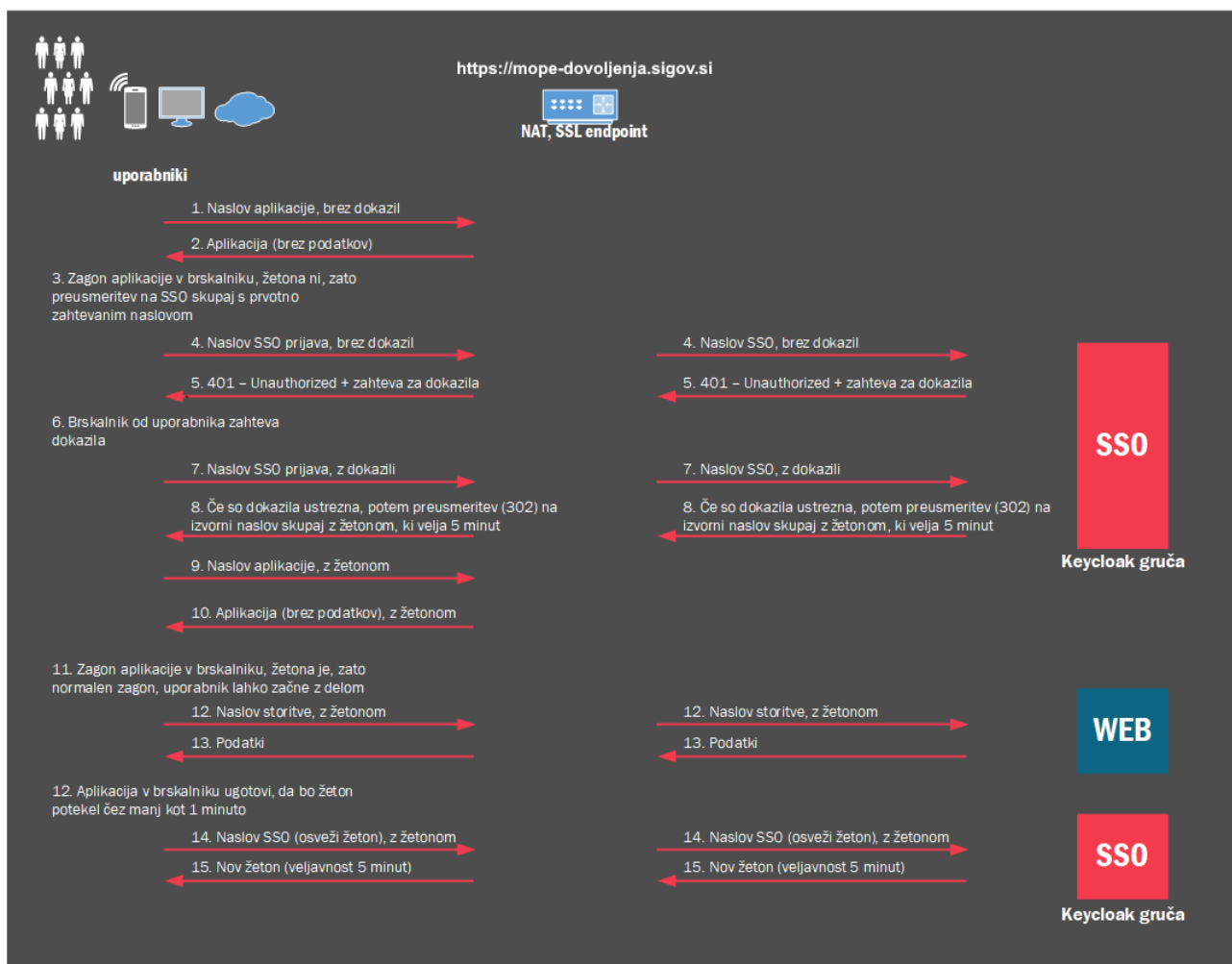
Storitev za posamezno povezavo ne kreira seje na strežniškem nivoju. Vse podatke pridobi iz žetona ali podatkovne zbirke (te podatke sistem hrani tudi v predpomnilniku). Na ta način zmanjšamo pomnilniški odtis, omogočimo tudi pravi stateless storitveni nivo.



Slika 9 - Storitve zavračajo zahteve brez veljavnega pristnega žetona (v odgovoru ni podrobnosti o razlogu zavrnitve)

Pošiljanje žetona je tako naloga aplikacije, ki teče v brskalniku uporabnika. Ta aplikacija prav tako preverja veljavnost žetona (v tem primeru brez javnega ključa – z namenom zaščite pred brute-force napadi na mehanizem podpisa). V primeru, da žetona aplikacija nima ali je žeton tik pred potekom (manj kot 1 minuta do poteka), aplikacija izvede zahtevek na SSO za podaljšanje in dobi nov žeton. Operacija je za uporabnika neopazna. Pogoji so usklajeni med storitvami in SSO sistemom.

Uporabnikov brskalnik lahko aplikacijo pridobi brez dokazila. Ob zagonu aplikacije ta ugotovi, da žetona nima (tega hrani v pomnilniku brskalnika, ob zagonu aplikacije je ta prazen) in uporabnikov brskalnik preusmeri na SSO stran. S tem se aplikacija zapre, uporabnik je na SSO strani, kjer mora (glede na nastavitve sistema) izkazati svojo pristnost. Ko dokaže svojo pristnost, SSO brskalniku vrne navodila za preusmeritev na aplikacijo skupaj z žetonom. Tokrat aplikacija ob zagonu ugotovi, da ima žeton in lahko nadaljuje delo.



Slika 10 - Tok zahtevkov in odgovorov ob običajnem dostopu uporabnika.

Centralna pozicija sistema SSO Keycloak predstavlja enotno točko varovanja sistema. Potvarjanje žetonov (bodisi z vdorom v SSO sistem ali s krajo podpisnega ključa) predstavlja največjo grožnjo sistemu. Posamezna storitev je prav tako lahko tarča vdora, vendar je v tem primeru nabor podatkov, ki so ogroženi, ustrezno manjši. Varovanje SSO sistema (tudi Varnostne sheme) je zato ključnega pomena za varnost sistema. To pomeni tako nadzor delovanja kakor sprotno posodabljanje! Naročnik mora zato načrtovati redna naročila za posodabljanje varnostnih mehanizmov sistema (običajno nekajkrat letno, ob objavljenih ranljivostih pa tudi izredno).

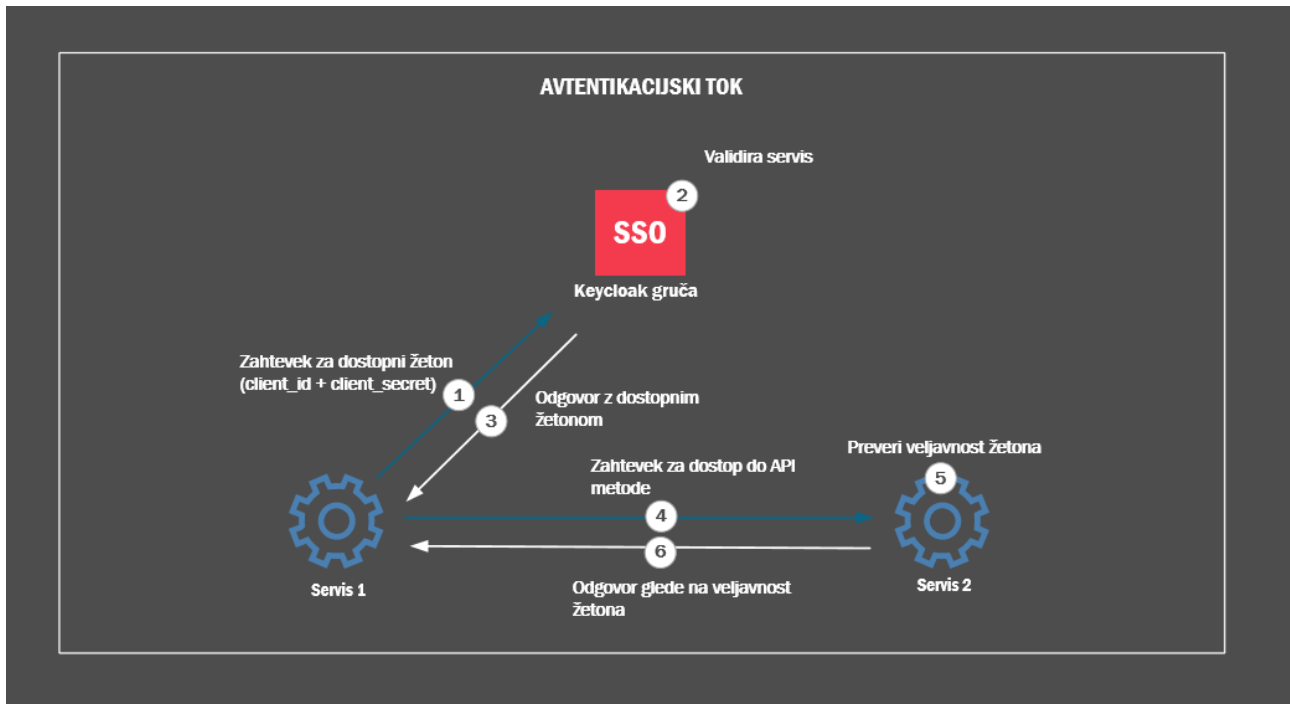
V primeru vdora v sistem je treba zagotoviti, da je ogrožen čim manjši del podatkov in postopkov, zato je sistem ločen na dva logična sklopa. Storitve, ki so postavljene v DMZ in strežejo uporabnikom izven HKOM, uporabljajo račune na podatkovni zbirki, ki imajo dostop le do minimalnega nabora operacij:

- Ne morejo dostopati do podatkov aplikacij, ki niso dostopne izven HKOM.
- Ne morejo dostopati do operacij, ki niso potrebne izven HKOM (na primer administracija sistema in podobno).

6.2.3. Avtentikacija servisov

Servisi, ki samodejno izvajajo procedure (brez prisotnosti uporabnikov) in potrebujejo dostop do API-jev drugih servisov, se pred pošiljanjem zahtevka na drug servis avtentificirajo na Keycloak strežniku.

Vsak servis prejme svoj `client_id` in `client_secret`, s katerima lahko od Keycloak zahteva žeton za dostop do API-ja na drugem servisu. API-ji delujejo enako kot pri uporabnikih – preverijo ali ima zahteva veljaven žeton in ustrezno odgovorijo.



Slika 11 - Tok zahtevkov in odgovorov ob avtentikaciji servisov

6.3. Podatkovni nivo

Vsi dostopi do podatkovne zbirke so implementirani preko klicev namenskega nivoja spletnih storitev.

Dodatno so vse operacije sprememb podatkov in večina operacij branja podatkov izvedene z uporabo PL/SQL paketov (stored procedure). Izjema so:

1. funkcionalni sklopi, ki so izvedeni v obliki produktov (Keycloak) in
2. funkcionalni sklopi, kjer je zahtevana dinamična priprava poizvedb (analitika, iskanje z uporabo filtrov).

V vsakem primeru je dostop omejen preko posebnih pogledov, ki uporabnika na podatkovni zbirki omejujejo pri dostopu do podatkov tako, da pod nobenimi pogoji (tudi select brez where kriterijev) ne more dostopati do tabel, vrstic in stolpcev, do katerih zunanji uporabniki nimajo dostopa (z uporabo Oracleovega RLS ali where kriterijev ali oboje).

Na podatkovni zbirki so izdelani namenski uporabniki za dostop posamezne storitve. Ti uporabniki imajo minimalen nabor konkretnih pravic, nimajo možnosti spreminjanja modela podatkovne zbirke in so dodatno ločeni na uporabnike, ki so namenjeni storitvam, objavljenim v DMZ, in tistim znotraj HKOM. Zadnja linija zaščite je uporaba Oracleovega RLS, ki po omejitvah (stolpci in vrstice), ki jih ima prijavljeni bazni uporabnik, aplicira še omejitve končnega uporabnika (vloge in dostopi). Brez informacije o končnem uporabniku metode in pogledi na podatkovni zbirki ne vračajo podatkov.

7. Integracije z zunanjimi sistemi

Izmenjave podatkov v sistemu v grobem delimo na naslednje: izhodne integracije in vhodne integracije.

7.1. Izhodne integracije z zunanjimi sistemi

Pri izhodnih integracijah z zunanjimi sistemi gre za izmenjave, kjer je pobudnik IS Dovoljenja in ta kliče druge zunanje sisteme preko različnih protokolov (REST, SOAP, DB LINK). Protokol je v veliki meri odvisen od implementacije izmenjave na sistemu, ki ga kličemo. Vse zunanje integracije se izvajajo preko modula Integ, ki je postavljen znotraj HKOM-a in predstavlja izhodno točko za klic zunanjih sistemov.

Pričakovane izmenjave z zunanjimi sistemi, ki bodo speljane preko modula Integ so naslednje:

- MS Sharepoint
- KRPAN
- AJPES
- SI-CES
- GURS
- IS Monitoring

7.1.1. MS Sharepoint

Delovne verzije dokumentov, ki se bodo generirale oz. uporabljale v okviru IS Dovoljenja bodo hranjene na MS Sharepoint, ki je že uporabljen na MOPE in znotraj katerega bo kreirano ločeno mesto (site) za namen IS Dovoljenja. Poleg delovnih verzij dokumentov bodo na MS Sharepoint hranjene tudi predloge (vzorci) dokumentov, ki se bodo uporabljali za kreiranje osnutkov odločb in ostalih dokumentov. Predloge bodo iz IS Dovoljenja dostopne skozi šifrant predlog v katerem bo za posamezno predlogo shranjena povezava (link) do lokacije znotraj MS Sharepoint.

Za branje in pisanje dokumentov na MS Sharepoint bo narejena integracija preko Sharepoint REST API.

Koncept integracije IS Dovoljenja z MS Sharepoint:

1. Ob kreiranju novega projekta v okviru IS Dovoljenja bo potrebno istočasno kreirati tudi ustrezno mapo znotraj MS Sharepoint-a v kateri bodo hranjeni vsi dokumenti povezani s tem projektom (delovna mapa, ki lahko vsebuje osnutke in pomožne dokumente).
 - Kreiranje nove mape projekta v MS Sharepoint bo možno sprožiti preko klica API vmesnika (MS Sharepoint) neposredno iz IS Dovoljenja.
2. Struktura map in podmap za namene hranjenja dokumentov posameznega projekta je stvar odločitve naročnika.
 - Pravice dostopa IS Dovoljenja do vseh map projektov bodo enake. Krmiljenje pravic urejanja dokumentov posameznega projekta bo urejano na nivoju projekta znotraj IS Dovoljenja. Če uporabnik ne bo imel pravice videti ali urejati podatke projekta znotraj IS Dovoljenja, ne bo videl tudi povezave (linka) do dokumentov shranjenih v MS Sharepoint.
3. Dostop do MS Sharepoint bo urejen za:

-
- **Administratorje MS Sharepoint**, ki bodo imeli pravice kreirati/urejati/brisati mape namenjene za IS Dovoljenja.
 - **IS Dovoljenja (aplikacijskega uporabnika)**, ki bo imel pravice dodajanja/spreminjanja/brisanja posameznih predlog (vzorcev) in ostalih word dokumentov znotraj map IS Dovoljenja.
 - **Posamezne uporabnike IS Dovoljenja**, ki bodo uporabljali funkcionalnosti povezane z MS Sharepoint.

V razvojnem okolju bo uporabljena postavitve MS Sharepoint v okolju izvajalca.

V testnem in produkcijskem okolju bo uporabljena produkcijska postavitve MS Sharepoint v okolju MDP. Za potrebe testnega okolja bo v produkcijski postavitvi MS Sharepoint kreirana ločena »testna« map v kateri bodo hranjene vsebine povezane s testnim okoljem IS Dovoljenja. Za potrebe produkcijskega okolja bo kreirana ločena »produkcijska« map. Na ta način bosta na isti postavitvi MS Sharepoint na voljo obe okolji za IS Dovoljenja.

Testno okolje:

- V testnem okolju bo uporabljena produkcijska postavitve, ki je dosegljiva na url-ju:
 - <https://mnz.sharepoint.com/>

Produkcijsko okolje:

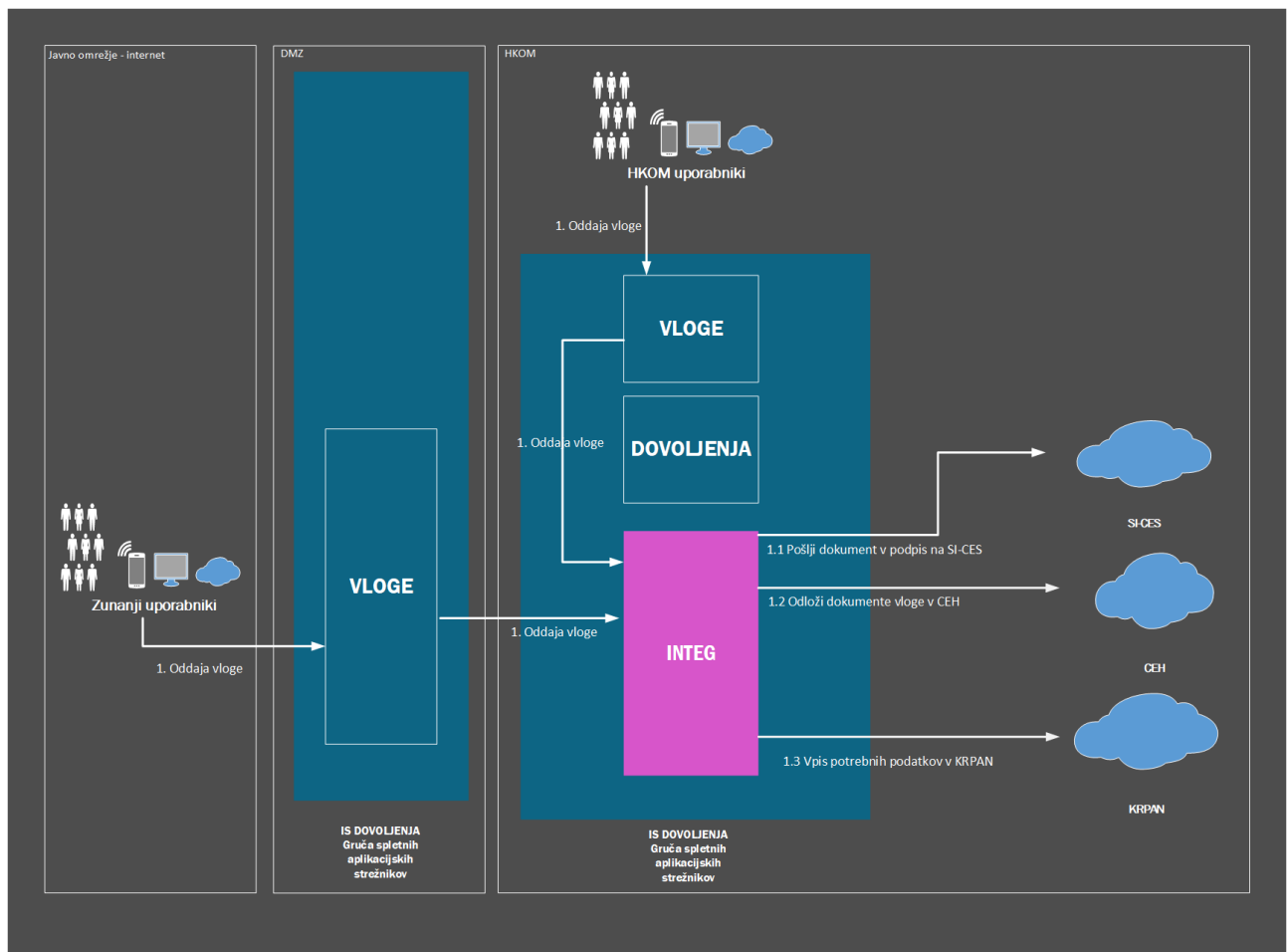
- Produkcijska postavitve je dosegljiva na url-ju:
 - <https://mnz.sharepoint.com/>

7.1.2. KRPAN in CEH

IS Dovoljenja se bo s sistemi KRPAN in CEH integriral z namenom kreiranja nove zadeve v sistemu KRPAN, odlaganja dokumentov v KRPAN/CEH, podpisovanja in vročanja dokumentov ter branja dokumentov iz sistema KRPAN/CEH. Izmenjave s sistemom KRPAN/CEH se bodo izvajale, ko bo uporabnik sprožil akcijo, ki zahteva zapis ali branje podatkov na/iz sistema KRPAN/CEH (prenos datoteke, potrditev dokumenta v končno stanje, ...).

Primeri integracije IS Dovoljenja s sistemom KRPAN/CEH:

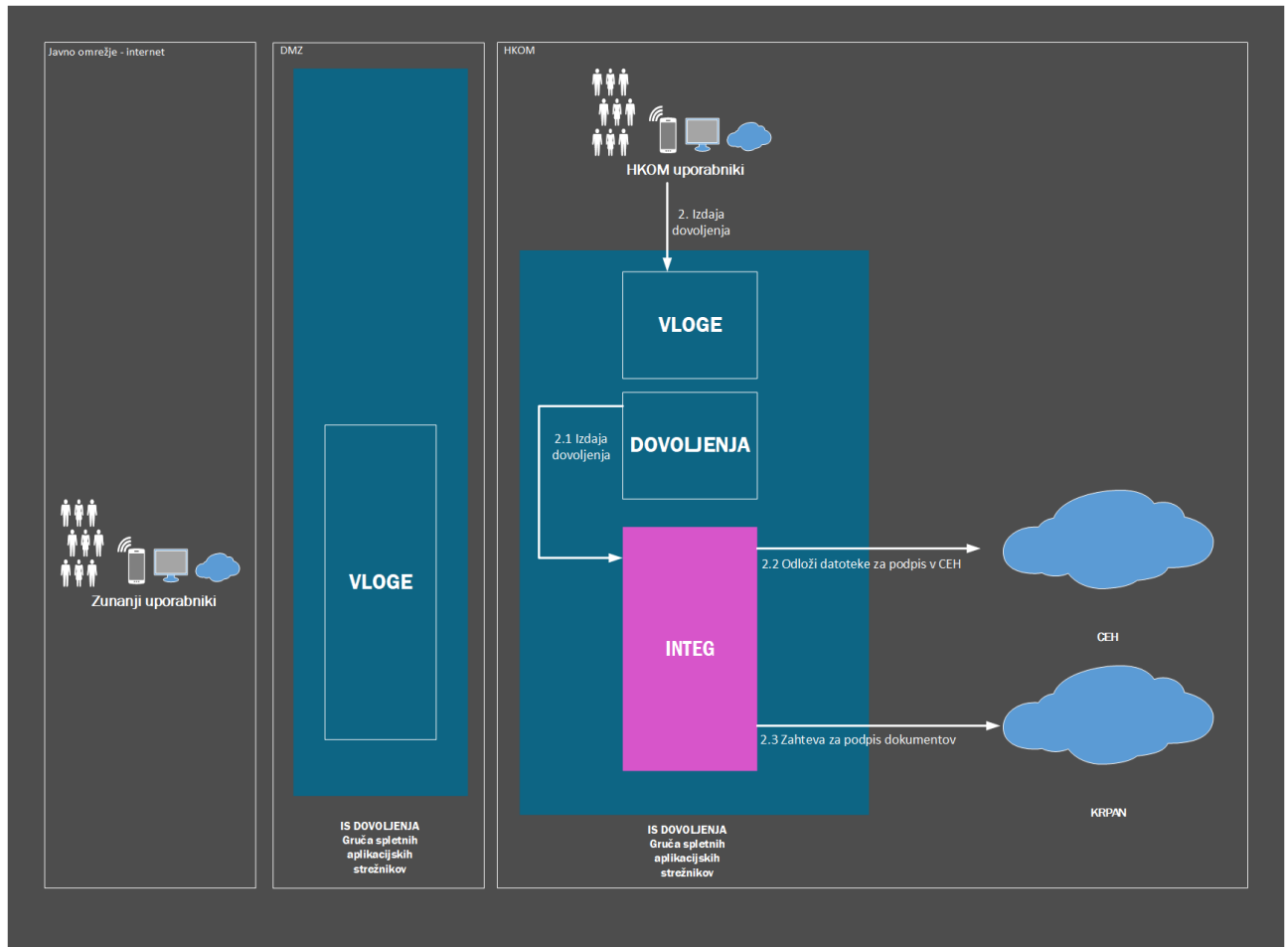
1. Vlogo lahko oddajajo uporabniki znotraj ali izven HKOM-a. Ob postopku oddaje se preko modula Integ izvede naslednje korake izmenjave s sistemoma KRPAN in CEH:



Slika 12 - Oddaja vloge v Krpam in CEH

- 1.1. Kreirane dokumente vloge (pdf) se pošlje v digitalno podpisovanje, uporabnika se preusmeri na SI-CES, kjer uporabnik dokumente podpiše s svojim digitalnim potrdilom, po podpisu IS Dovoljenja prejme podpisan dokument iz SI-CES-a.
- 1.2. Podpisan dokument odložimo v CEH in kot odgovor prejmemo ID dokumenta.
- 1.3. Vpis potrebnih podatkov v KRPAN:
 - 1.3.1. Številka zadeve: V kolikor gre za vlogo za izdajo novega dovoljenja, se v KRPAN-u kreira novo številko zadeve, v kolikor gre dopolnitev obstoječe vloge, se dokument vstavi na obstoječo zadevo.
 - 1.3.2. ID dokumentov iz CEH-a.
 - 1.3.3. Ostali metapodatki (ob kreiranju nove zadeve še klasifikacijska številka, nosilec zadeve ipd).

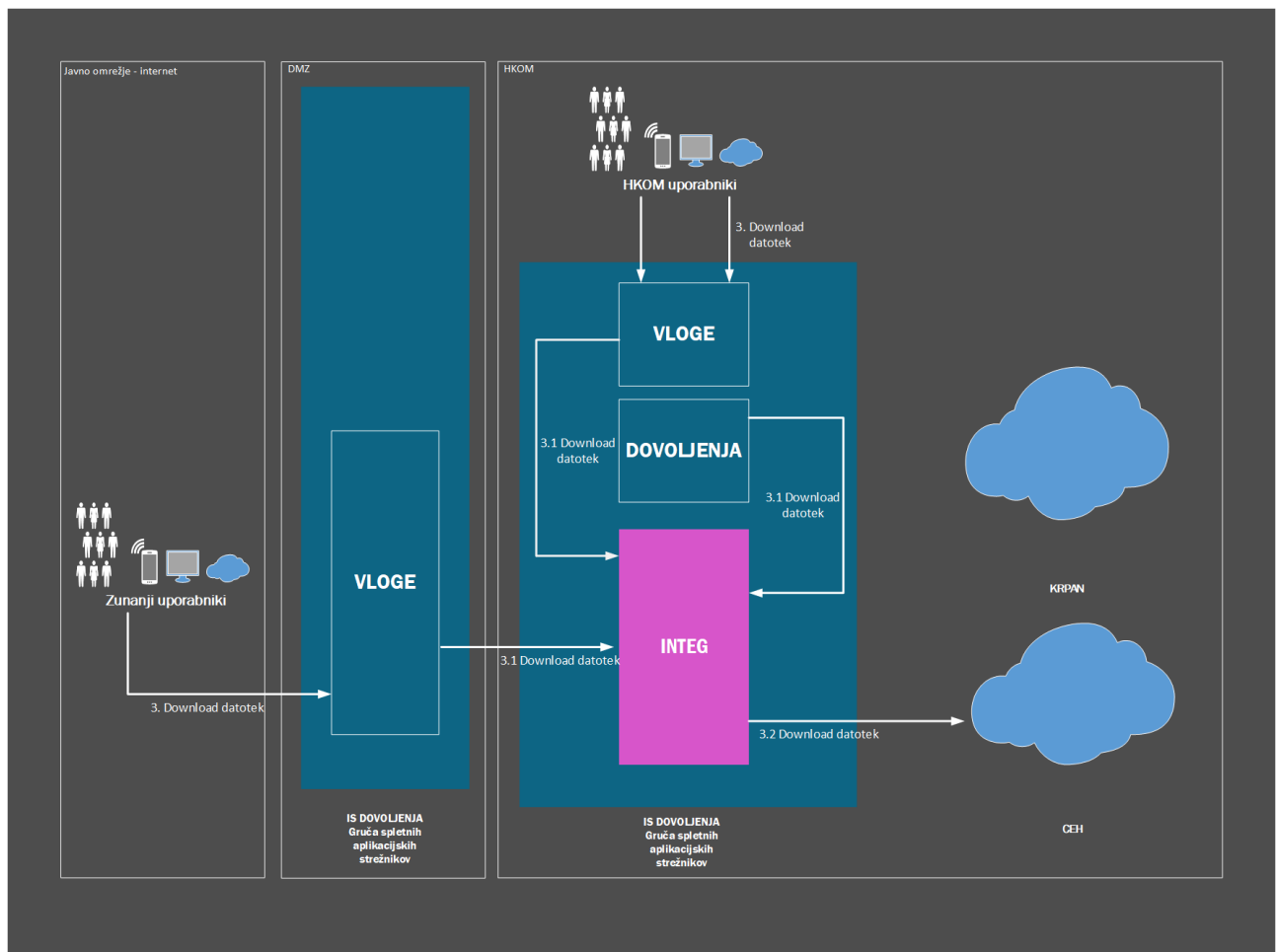
2. Izdaja dovoljenja poteka samo znotraj omrežja HKOM po naslednjih korakih:



Slika 13 - Oddaja dovoljenja v Krpan in Ceh

- 2.1. Uporabnik MOPE preko aplikacije IS Dovoljenja potrdi izdajo dovoljenja. Zahtevek aplikacija posreduje modulu Integ.
- 2.2. Modul Integ odloži datoteke v CEH in prejme ID dokumentov iz CEH-a.
- 2.3. Modul Integ kliče KRPAN z zahtevo za podpis dokumentov in vsemi potrebnimi podatki, ki jih KRPAN potrebuje za nadaljnje postopke (vročitev dokumentov). Po podpisu KRPAN sporoči IS Dovoljenja reference do podpisanih dokumentov v CEH-u. Za premik dokumentov v strukturirani del CEH poskrbi KRPAN.

3. Prenos datotek iz dokumentnega sistema bo potekal po naslednjih korakih:



Slika 14 - Prenos datotek iz CEH-a

- 3.1. Zahteva uporabnikov za prenos datotek se posreduje modulu Integ.
- 3.2. Modul Integ datoteke prenese iz CEH-a.

7.1.2.1. Integracija KRPAN

Izmenjave s sistemom KRPAN bodo potekale preko SOAP servisov (tesna integracija), ki so dosegljivi na spodnjih url-jih:

Razvojno okolje:

- Razvojni servis je dosegljiv na url-ju:
 - <https://integracije.krpan.marg.si/IntegrationMJU/ExternalServices/CkeServis.asmx>
- Tehnični opis servisa se nahaja na url-ju:
 - <https://integracije.krpan.marg.si/IntegrationMJU/ExternalServices/CkeServis.asmx?WSDL>

Testno okolje:

- Testni servis je dosegljiv na url-ju:
 - <https://krpan-integracije-test.testad.sigov.si/IntegrationMJU2/ExternalServices/CkeServis.asmx>

-
- Tehnični opis servisa se nahaja na url-ju:
 - <https://krpan-integracije-test.testad.sigov.si/IntegrationMJU2/ExternalServices/CkeServis.aspx?wsdl>
 - Testno okolje KRPAN-a je dostopno na:
 - <https://krpan-test.sigov.si>

Produksijsko okolje:

- Produksijski servis je dosegljiv na url-ju:
 - <https://krpan-integracije.ad.sigov.si/IntegrationMJU2/ExternalServices/CkeServis.aspx>
- Tehnični opis servisa pa je na voljo na naslovu:
 - <https://krpan-integracije.ad.sigov.si/IntegrationMJU2/ExternalServices/CkeServis.aspx?wsdl>
- Produksijsko okolje KRPAN-a je dosegljivo na naslovu
 - <https://krpan.sigov.si/>

7.1.2.1.1 Primer klica servisa KRPAN – dodajanje dokumenta

Za potrebe dodajanja novega dokumenta v KRPAN, ki bo obenem preko sistema KRPAN tudi digitalno podpisan ter vročen, se uporabi klic metode **CreateNewDocument**.

Poleg ostalih podatkov se v klic metode posredujejo naslednji pomembni podatki:

- **DocumentType**: Draft
- **SignProcessGUID**: ID procesa zagotovi sistem KRPAN
- **Responsible**: Signirna oznaka uporabnika, ki dodaja dokument
- **ShortContent**: Opis dokumenta
- **MainContent**: Vsebina dokumenta
- **DocumentDate**: Datum dokumenta
- **Caseld**: ID zadeve v katero se bo kreiran dokument uvrstil
- **BackendID**: ID dokumenta iz sistema KRPAN. S tem podatkom se bo ustvarila zaledna oznaka novega dokumenta.
- **SendToSigning**: true
- **Approvers**: seznam podpisnikov
 - Seznam vseh možnih podpisnikov prejmemo s klicem metode GetApprovers.
- **LetterTypeID**: vrsta pošiljke
 - Seznam vrst pošiljk prejmemo s klicem metode GetLetterTypeList.
- **ServiceMethod**: želeni način vročitve
- **Shipment**: sklop podatkov o prejemniku ali več njih

Lastni dokument (draft) se v tem primeru, ob obdelavi v sistemu KRPAN, spremeni v izhodni dokument (outgoing).

Metoda ob uspešni izvedbi med ostalim vrne tudi podatek **DocumentID**, ki predstavlja identifikator dokumenta znotraj sistema KRPAN.

7.1.2.1.2 Primer klica servisa KRPAN – pridobitev stanja dokumenta

Za potrebe pridobitve stanja dokumenta (npr. če je bil ta digitalno podpisan) se uporabi klic metode **GetDocumentData**.

V klic metode se poleg apiKey pošlje še podatek **DocumentID**, ki smo ga od sistema KRPAN prejeli ob uspešnem dodajanju dokumenta.

Metoda ob uspešni izvedbi med ostalim vrne tudi podatek **DocumentStatusUri**, ki predstavlja status tega dokumenta. Na podlagi tega podatka lahko IS Dovoljenja preveri, če je dokument digitalno podpisan ter pripravljen za prevzem. Iz klica metode se lahko pridobi tudi podatek o datumu odpreme dokumenta (**DocumentDate**).

Metoda ne omogoča pridobitev podatka o uspešnosti vročanja dokumenta.

7.1.2.2. Integracija CEH

Integracija s sistemom CEH poteka preko namenskih vmesnikov na strani sistema CEH.

Vse zadeve in dokumenti povezani s temi zadevami se kreirajo izključno skozi sistem KRPAN. Sistem KRPAN je edini dokumentni sistem na MOPE, ki se bo uporabljal v okviru IS Dovoljenja, in bo vsa uradna dokumentacija evidentirana izključno skozi KRPAN.

Do gradiva shranjenega v CEH bo neposredno dostopal le IS Dovoljenja (modul Integ).

Tehnične podrobnosti integracije IS Dovoljenja s sistemom CEH bodo usklajene z vzdrževalci sistema CEH.

Klasifikacijski razredi, ki se uporabljajo znotraj CEH, so naslednji:

- 35451 - Odpadki – dovoljenje za odstranjevanje azbesta,
- 35452 - Odpadki - okoljevarstvena dovoljenja za vnos zemljin, blata iz ČN, komposta, digestata (zemljine, blato ČN, kompost),
- 35460 - Odpadki – okoljevarstvena dovoljenja za obdelavo odpadkov,
- 35459 - Odpadki - drugi upravni akti (druge odločbe),
- 35461 - Odločbe o vpisu v evidenco načrtov ravnanja z odpadki,
- 35456 - Odpadki – odlagališča – okoljevarstvena dovoljenja,
- 35455 - Odpadki - okoljevarstvena dovoljenja za zapiranje in zaprta odlagališča,
- 35458 - Odpadki – prevozniki, posredniki, trgovci,
- 35457 - Odpadki – zbiralci,
- 35447 - Vloga za izdajo/spremembo OVD (skupni OVD),
- 35432 - IED vloga za spremembo okoljevarstvenega dovoljenja,
- 35433 - IED okoljevarstvena dovoljenja,
- 35434 - IED - odvzem dovoljenja in odločbe o prenehanju veljavnosti,
- 35436 - HOS - vpis v evidenco naprav in spremembe,
- 35440 - Skrbniki varstva okolja (IED, SEVESO, obdelava odpadkov),
- 35444 - Zrak - okoljevarstvena dovoljenja za emisije v zrak, sprememba, odvzem, odločba o prenehanju,
- 35450 - Hrup - OVD za obratovanje vira hrupa, sprememba, odvzem, odločba o prenehanju,
- 35453 - Elektromagnetno sevanje - odločbe o spremembi obratovalnega monitoringa,
- 35467 - SEVESO splošno,
- 35468 - SEVESO okoljevarstveno dovoljenje, sprememba, odvzem, odločbe o prenehanju,
- 35469 - SEVESO odločbe o verižnih učinkih,
- 35448 - Odpadne vode - okoljevarstveno dovoljenje/sprememba,
- 35466 - Vode - odločbe o monitoringu in emisijah,
- 35445 - Pooblastila za obratovalni monitoring (za vse sektorje).

Tipi zadev, ki se bodo hranili v CEH, so naslednji:

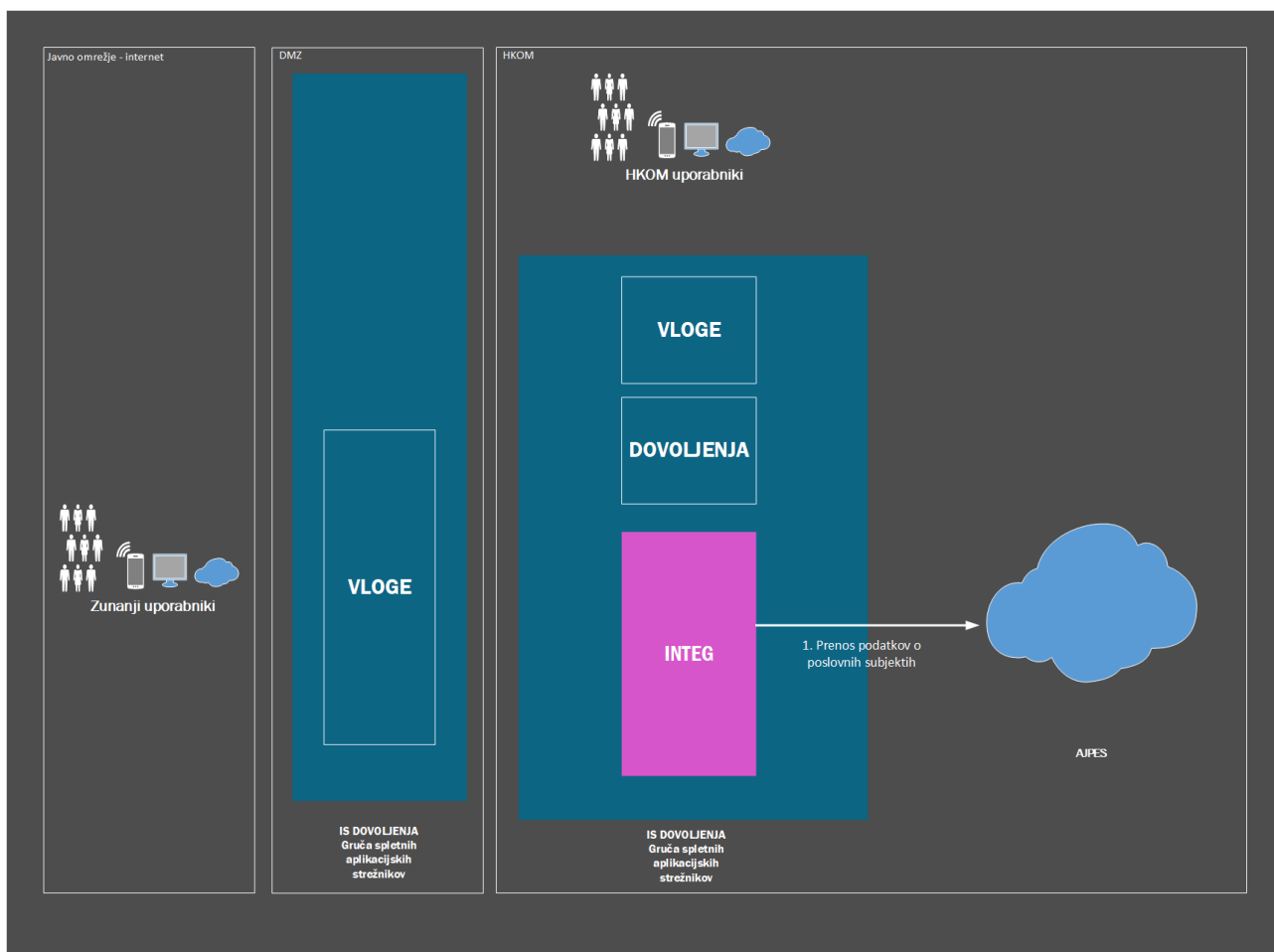
- Okoljevarstvena dovoljenja (več različnih vrst),
- Spremembe, odvzem, prenehanje okoljevarstvenih dovoljenj (več različnih vlog),
- Potrdila o vpisih v evidenco (več različnih vrst),
- Pooblastila za izvajanje obratovalnih monitoringov (več različnih vrst).

Za vse tipe zadev (in dokumentov) se v sistemu CEH hranijo podatki o:

- Vrsti zadeve (VARCHAR2(200)),
- Številki zadeve (VARCHAR2(40)),
- Upravljavcu oz. stranki v postopku (VARCHAR2(200)),
- Datumu prejema (DATE),
- Datumu izdaje (DATE),
- Datum dokončnosti (DATE)
- Datum pravnomočnosti (DATE),
- Roku hrambe (NUMBER(2)),
- Datum, kdaj je bila zadeva rešena (DATE).

7.1.3. AJPES

Sistem IS Dovoljenja podatke o poslovnih subjektih Poslovnega registra Slovenije iz AJPES-a prenaša na tedenski ravni v nočnem času preko FTP strežnika na strani AJPES (vse tedenske spremembe se prevzamejo z eno datoteko, ki je odložena na FTP strežniku). Za proženje pridobivanja podatkov skrbi modul Integ, kot je prikazano na spodnji sliki.



Slika 15 - Integracije s sistemom AJPES

Za integracije se uporabi FTP strežnik na AJPES, ki bo dostopen iz vseh okolij IS Dovoljenja (razvojno, testno, produkcijsko).

FTP dostop:

- Spletni naslov za prevzem podatkov:
 - <ftp://direktoko@prenos.ajpes.si/prs>
- Mape s podatki: XRB, XRT, SIF

Za pripravo integracije se izvajalec sklicuje na naslednja navodila:

https://www.ajpes.si/Doc/AJPES/Za_razvijalce/Struktura_podatkov_PRS_082020.pdf

7.1.3.1. Prevoz podatkov iz AJPES

Prevoz podatkov Poslovnega registra Slovenije (PRS) iz AJPES je implementiran na dva načina in sicer:

7.1.3.1.1 Prvi (inicialni) prenos

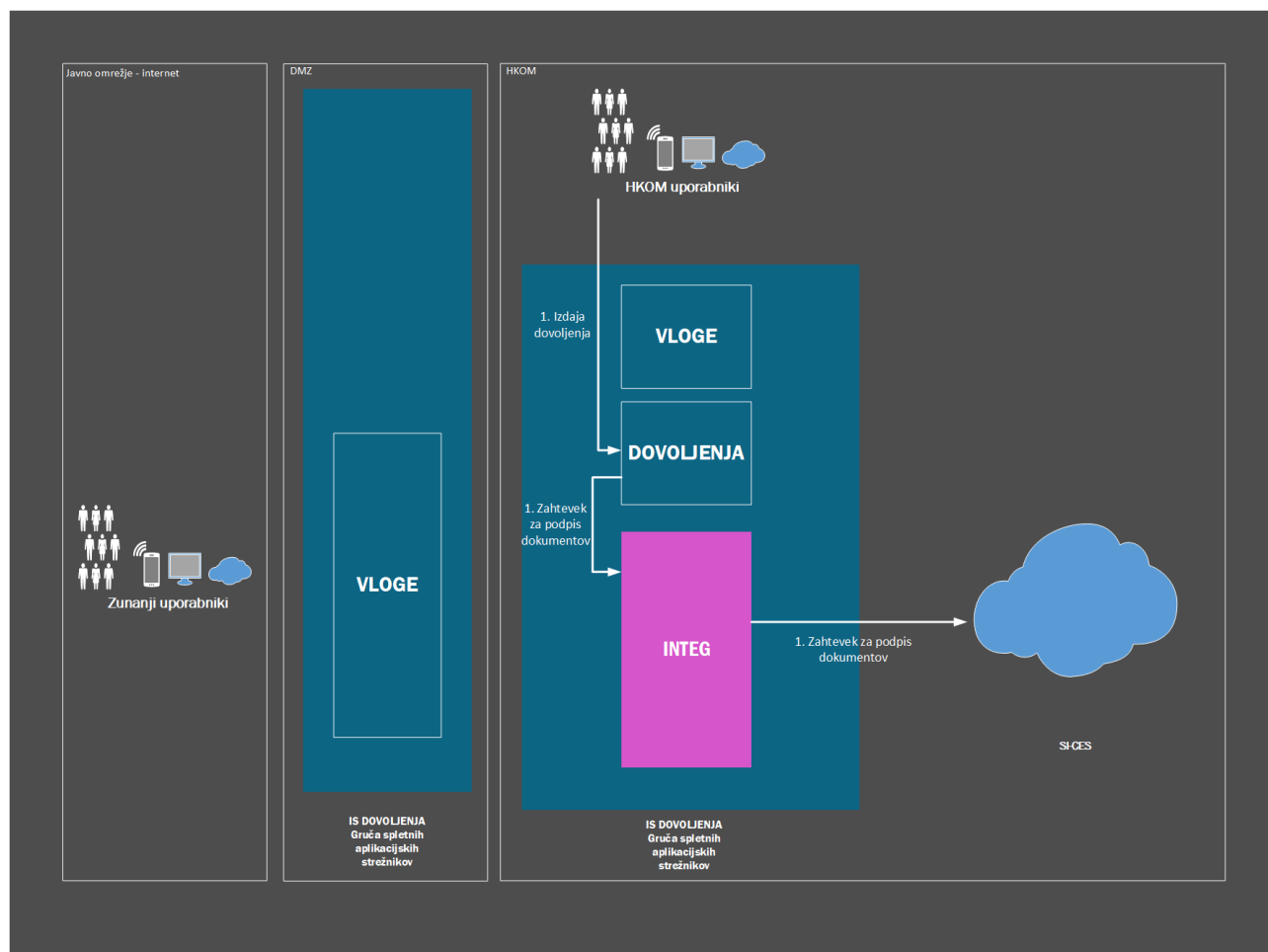
Za prvi (inicialni) prenos celotnega stanja PRS je bil uporabljen izvoz celotnega stanja PRS, ki je na voljo na FTP strežniku na AJPES.

7.1.3.1.2 Periodičen prenos sprememb

Po prvem (inicialnem) prenosu IS Dovoljenja periodično (predvidoma enkrat tedensko) iz FTP strežnika na AJPES prevzema le spremembe v stanju PRS.

7.1.4. SI-CES

Storitev SI-CES je uporabljena za podpisovanje dokumentov v IS Dovoljenja. Izmenjava se sproži, ko bo uporabnik sprožil akcijo, ki bo zahtevala njegov podpis na dokumentu. Izmenjava poteka z uporabo SOAP protokola.



Slika 16 - Izmenjave s sistemom SI-CES

Razvojno okolje:

- V razvojnem okolju je bil uporabljan testni servis, ki je dosegljiv na url-ju:
 - <https://sicas-test.sigov.si/CES-Sign/SI-CESSign?wsdl>

Testno okolje:

- Testni servis je dosegljiv na url-ju:
 - <https://sicas-test.sigov.si/CES-Sign/SI-CESSign?wsdl>

Produksijsko okolje:

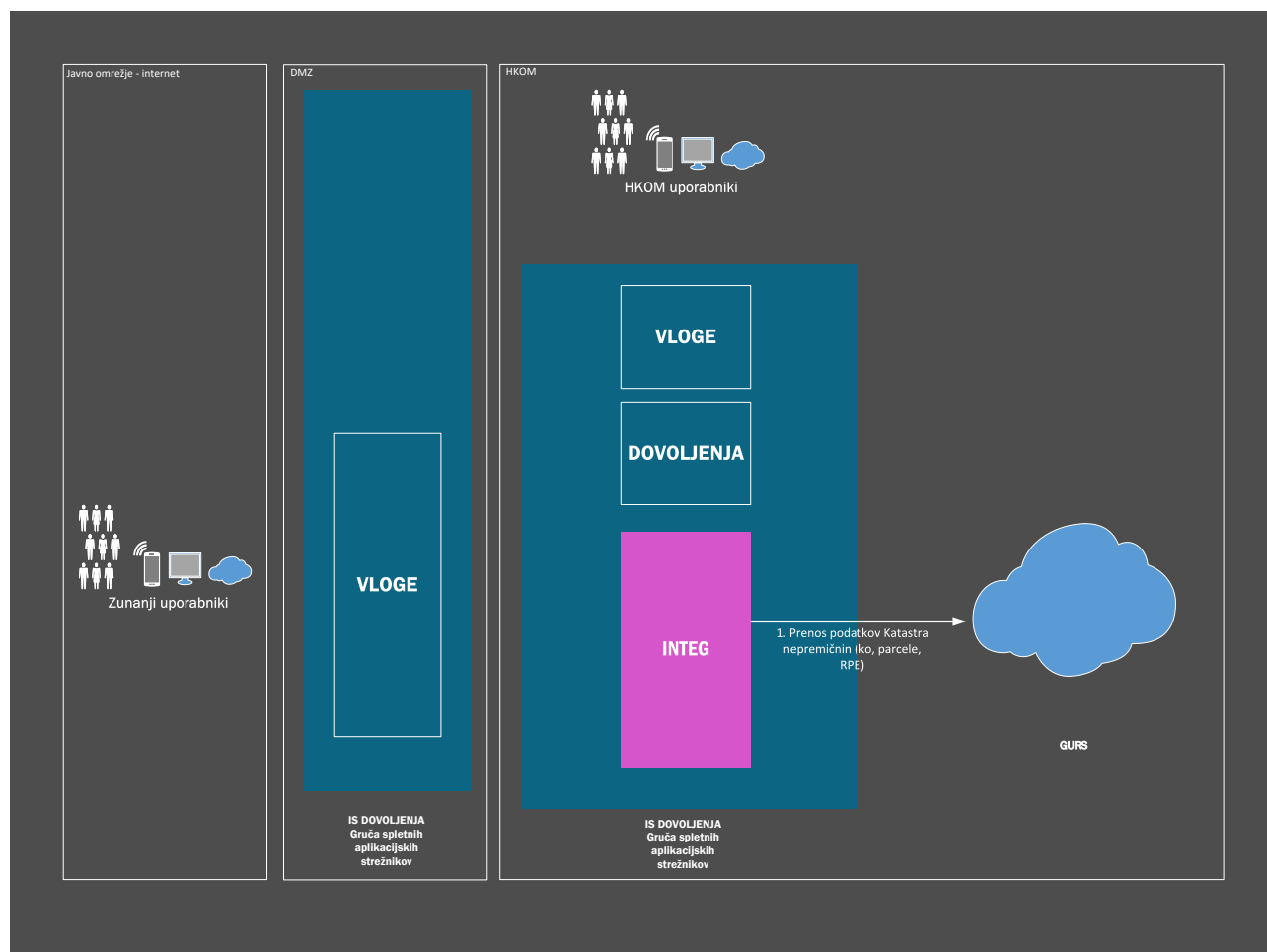
- Produkcijski servis je dosegljiv na url-ju:
 - <https://sicas.gov.si/CES-Sign/SI-CESSign?wsdl>

7.1.5. GURS

Sistem IS Dovoljenja potrebuje iz GURS-a podatke o katastrskih občinah in parcelah ter podatke iz registra prostorskih enot (RPE), ki se vodijo v Katastru nepremičnin.

Zaradi potrebe po izvajanju presekov podatkov shranjenih v IS Dovoljenja (npr. centriodi ali poligoni kompleksov in naprav) s podatki iz GURS (parcele katastra) je bila sprejeta odločitev, da se iz GURS prenaša celotno stanje parcel katastra (za celo Slovenijo). Iz tega razloga je potrebno izvajati prevzem podatkov neposredno v IS Dovoljenja preko spletnih servisov GURS namesto občasnih poizvedovanj (preko spletnih servisov) za posamezno parcelo.

Sistem IS Dovoljenja podatke o katastrski občinah, parcelah katastra ter RPE iz GURS-a prenaša na tedenski ravni v nočnem času z uporabo REST klicev na javno dostopne servise GURS. Za proženje pridobivanja podatkov skrbi modul Integ, kot je prikazano na spodnji sliki.



Slika 17 - Integracije s sistemom GURS

Razvojno okolje:

- V razvojnem okolju je uporabljen produkcijski servis GURS, ki je dosegljiv na url-ju:
 - <https://ipi.eprostor.gov.si/wfs-si-gurs-kn/ogc/features/collections?f=text%2Fhtml> (<https://ipi.eprostor.gov.si/wfs-si-gurs-kn/ogc/features/collections?f=text%2Fhtml>)

Testno okolje:

- V testnem okolju je uporabljen produkcijski servis GURS, ker ponuja osveženo stanje podatkov (testni servisi GURS ponujajo testne podatke, ki za uporabo v IS Dovoljenja niso ustrezni). Produkcijski servis GURS je dosegljiv na url-ju:
 - <https://ipi.eprostor.gov.si/wfs-si-gurs-kn/ogc/features/collections?f=text%2Fhtml> (<https://ipi.eprostor.gov.si/wfs-si-gurs-kn/ogc/features/collections?f=text%2Fhtml>)

Produkcijsko okolje:

- Produkcijski servis je dosegljiv na url-ju:
 - <https://ipi.eprostor.gov.si/wfs-si-gurs-kn/ogc/features/collections?f=text%2Fhtml> (<https://ipi.eprostor.gov.si/wfs-si-gurs-kn/ogc/features/collections?f=text%2Fhtml>)

7.1.5.1. Prezem podatkov iz GURS

Prezem podatkov o katastrski občinah, parcelah katastra ter RPE iz GURS je implementiran na dva načina in sicer:

7.1.5.1.1 *Prvi (inicialni) prenos*

Za prvi (inicialni) prenos celotnega stanja je bil uporabljen servis naveden zgoraj v načinu za prenos inicialnega (celotnega) stanja.

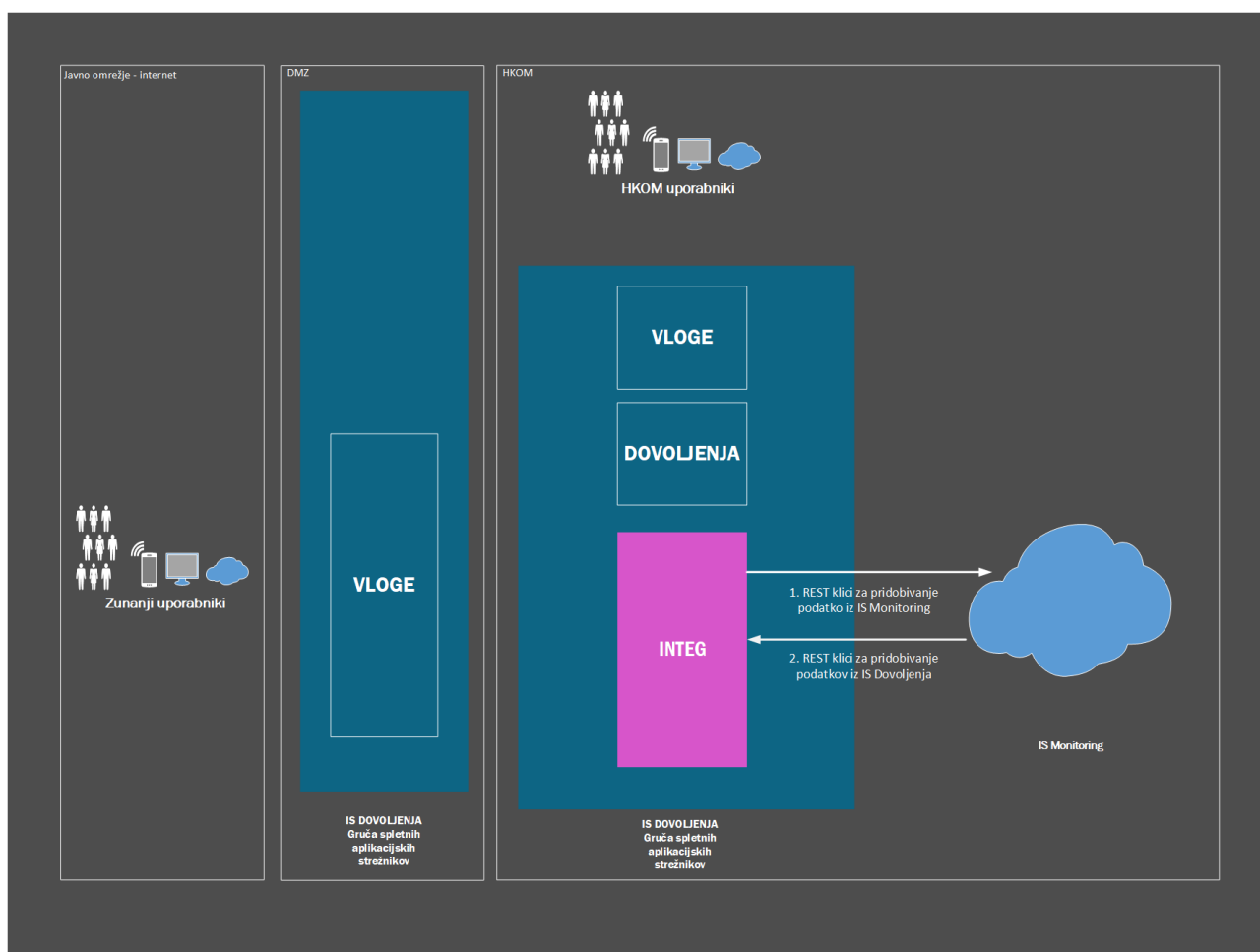
Po uspešnem prenosu je bil v sistemu shranjen datum inicialnega prenosa.

7.1.5.1.2 *Periodičen prenos sprememb*

Po prvem (inicialnem) prenosu IS Dovoljenja periodično (predvidoma enkrat tedensko) iz GURS prevzema le spremembe v stanju podatkov od datuma zadnjega uspešnega prenosa. Za prevzem sprememb je klican servis naveden zgoraj v načinu za prenos sprememb v podatkih.

7.1.6. IS Monitoring

Vzpostavljena je dvosmerna integracija s sistemom IS Monitoring. Integracija poteka z uporabo REST storitev. Pri izhodni integraciji, kjer IS Dovoljenja pridobiva podatke iz IS Monitoringa, se prenaša podatke katerih skrbnik je IS Monitoring.



Slika 18 - Izmenjave podatkov z IS Monitoring

7.2. Vhodne integracije iz drugih zunanjih sistemov

V primerih, kjer ima IS Dovoljenja vlogo čakajočega na klic, se implementira integracijo v vsebinsko primeren API z uporabo REST protokola. V ta namen se lahko uporabi obstoječe metode API-ja, ki so že narejene za potrebe UI aplikacije IS Dovoljenja ali pa se kreira nove, če obstoječih ni ali pa nimajo primernih podatkov. Zunanji sistemi bodo dostopali preko dostopnih točk na modulu Integ, ta pa jih bo prevsmeril na ustrezno vsebinsko mikrorstitev. Vse vmesnike, ki bodo namenjeni za vhodne integracije z zunanjimi sistemi se opremi z ustreznimi anotacijami po specifikaciji MicroProfile OpenAPI tako, da se bo za te metode samodejno zgenerirala specifikacija, ki jo bodo zunanji sistemi lahko uporabili za izdelavo vmesnikov.

Avtentikacija se izvede enotno z ostalimi točkami dostopa z uporabo OIDC. V primeru dostopa s strani storitev (drugih aplikacij) se v IS Dovoljenja SSO ročno izdela uporabnika za posamezno zunanjo storitev in pripravi trajen žeton. Storitve mora (skladno z OIDC) žeton, ob dostopu, uporabiti tako, da pridobi začasen »access« žeton, s katerim nato izvede klic na storitev znotraj IS Dovoljenja. Podobno kakor ostalim uporabnikom, je potrebno znotraj IS Dovoljenja tudi takšnemu uporabniku urediti/dodeliti ustrezne pravice.

Trenutno je na takšen način predvidena integracija s sistemom ARSO IS Monitoring. Podatke, ki jih IS Monitoring prevzema od IS Dovoljenja, ARSO lahko po potrebi posreduje naprej svojim drugim sistemom (npr. IS Odpadki).