



Sistem Archeia

Visokonivojski opis arhitekture

Kazalo vsebine

O DOKUMENTU	3
NAMEN	3
UPORABNIKI DOKUMENTA	3
KRATICE, POJMI IN DEFINICIJE	3
SPLOŠNO	3
ALFRESCO CONTENT SERVICES	5
ARHITEKTURA	5
ARHITEKTURNI DIAGRAM NAMESTITVE SISTEMA	5
KLJUČNE KOMPONENTE SISTEMA	6
SKLADIŠČNI IN SPLETNI STREŽNIKI (REPOSITORY AND SHARE SERVERS)	6
STREŽNIK BAZ PODATKOV (DATABASE SERVER)	6
SHRAMBA VSEBINE (CONTENT STORE)	7
STORITVE ISKANJA ALFRESCO – LUCENE - INDEKSNI STREŽNIK (ALFRESCO SEARCH SERVICES – LUCENE - INDEX SERVER)	7
STORITVE UPRAVLJANJA ALFRESCO (UPRAVLJANJE DOKUMENTOV – RECORDS MANAGEMENT - RM)	7
SPECIFIKACIJE STROJNE OPREME	7
SPECIFIKACIJE PROGRAMSKE OPREME	8
PROGRAMSKA OPREMA IN UPORABLJENE TEHNOLOGIJE	8
PREDPOMNILNIK	9
ALFRESCO CACHE STORE - SHRANJEVANJE VSEBINE V PREDPOMNILNIKU (CCS)	11
LASTNOSTI CACHINGCONTENTSTORE (CACHINGCONTENTSTOREPROPERTIES)	12
ALFRESCO ENCRYPTED CONTENT STORE (ECS)	13
LASTNOSTI SHRANJENE ŠIFRIRANE VSEBINE (PROPERTIES)	15
MESSAGE QUEUE (MQ) KLIENT	15
REVIZIJSKA SLED	16
MIGRACIJA IN REPLIKACIJA DOKUMENTOV	17
HEMA SESTAVE PROGRAMSKE OPREME IN POVEZAV MED NJENIMI DELI	17
VARNOST REST VMESNIKOV TER AVTENTIKACIJA IN AVTORIZACIJA UPORABNIKOV	18

1 O DOKUMENTU

1.1 NAMEN

Dokument opisuje arhitekturno in tehnološko zasnovo sistema Archeia – *sistem za dolgoročno hrambo gradiva v elektronski obliki* (v nadaljevanju Archeia). Vključuje opis sheme sestave programske opreme in povezav med njenimi deli, popis uporabljenih tehnologij in dodatne opreme ter varnostne in zaščitne mehanizme, ki bodo uporabljeni.

1.2 UPORABNIKI DOKUMENTA

Dokument je namenjen članom projektne skupine na strani izvajalca in naročnika.

1.3 KRATICE, POJMI IN DEFINICIJE

Kratice/Pojem	Celoten naziv
PZI	Projekt za izvedbo
Archeia	Archeia – Sistem za dolgoročno hrambo gradiva v elektronski obliki
VSRS	Vrhovno sodišče Republike Slovenije (naročnik projekta)

2 SPLOŠNO

Sistem Archeia je visoko razpoložljiv sistem, ki zagotavlja hranjenje in pridobivanje dokumentov v obliki servisnih storitev. Sistem je skladen z vsemi predpisi za dolgoročno hrambo dokumentov (ZVDAGA).

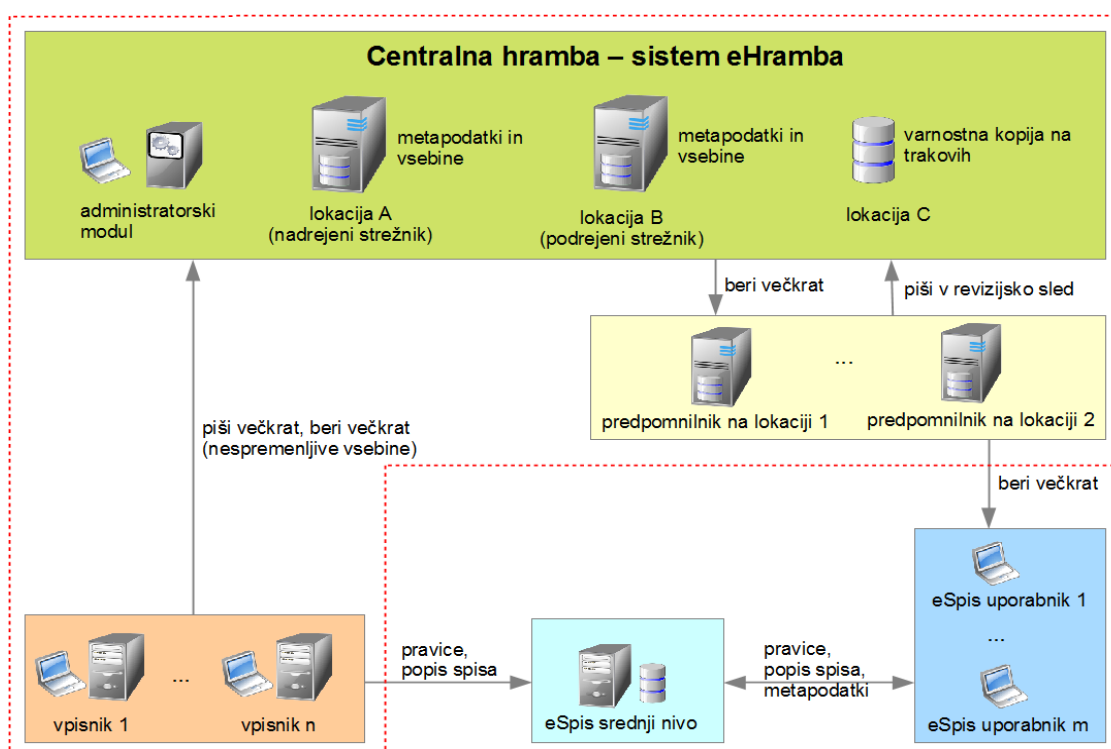
Osnovni namen Archeiaee je visoko odzivno hranjenje in pridobivanje različnih tipov dokumentov v vseh razmerah, ne glede na število uporabnikov sistema, količino hranjenih dokumentov, obremenitev sistema s klici servisov in razvejanost računalniškega omrežja, v katerem deluje.

Značilnosti računalniškega omrežja so:

- **zvezdna topologija omrežja**, s centralnim voliščem in več (tipično >100) krajevnimi vozlišči;
- **prenosne hitrosti** (*bandwidth*) med vozlišči in centralnim sistemom so različne (od slabih: 10 Mb/s do zelo dobrih: >100Mb/s);
- vsako vozlišče ima lasten (krajevni) **strežnik**;
- na vsak krajevni strežnik v vozlišču je priključenih **več uporabnikov** na delovnih postajah, prenosne hitrosti med vozliščem in delovno postajo so solidne (>=100MB/s);
- **dinamično izbiranje dostopnega vozlišča**: zaradi mobilnosti zaposlenih se lahko uporabniki (njihovi prenosni računalniki ali mobilne naprave) v omrežje priklapljajo iz poljubnega vozlišča;
- **dinamično dodajanje, spreminjanje in odstranjevanje vozlišč**: sodišča oz. oddelki so lahko razpršeni po različnih lokacijah (vozliščih), na eni lokaciji (vozlišču) pa je lahko tudi več oddelkov; oddelki se lahko združujejo, razdružujejo in ukinjajo; sistem Archeia mora biti zasnovan tako, da dodajanje in spreminjanje naslovov vozlišč poteka avtomatično s spreminjanjem konfiguracije; odstranjevanje vozlišč se nikdar ne sme odražati v izgubi dokumentov;

- **dinamično nameščanje novih verzij – vozlišč:** zaradi večjega števila vozlišč mora nameščanje novih različic aplikacij potekati tako, da se aplikacijo namesti samo na eno »izbrano« vozlišče, na vseh ostala vozlišča pa se namestitve izvede avtomatizirano
- **množica različnih naprav,** s katerimi končni uporabniki dostopajo do dokumentov: uporabniki lahko za delo od doma ali pri delu na drugem sodišču uporabljajo različne naprave za delo (delovno postajo, prenosni računalnik, mobilno napravo);
- **centralna hramba dokumentov:** končni dokumenti morajo biti dostopni preko centralnega sistema za nadaljnjo obdelavo kot so elektronsko vročanje, strojno kuvertiranje, arhiviranje, časovno žigosanje, itd.;
- **varnost:** centralni sistem je postavljen v fizično varovanem okolju, kjer so dostopi do podatkov in binarnih vsebin zaščiteni in beleženi, medtem ko so lokalna vozlišča nižje stopnje zaščite; zaradi tega mora biti programska oprema v lokalnih vozliščih narejena na način, da so dokumenti in metapodatki v vsakem trenutku (ob hrambi ali prenosu) zaščiteni pred nepooblaščenimi dostopi, spreminjanjem ali posredovanjem.

Sistem Archeia zagotavlja zakonsko skladno dolgoročno hrambo dokumentov na način, ki ga prikazuje naslednja slika:



3 ALFRESCO CONTENT SERVICES

Alfresco Content Services je platforma za upravljanje vsebin (angleško Enterprise Content Management – v nadaljevanju ECM), ki omogoča upravljanje vseh vrst vsebin: od pisarniških dokumentov, skeniranih slik, fotografij in celo velikih video datotek. Vgrajena funkcionalnost delovnih tokov (angleško Workflow) uporabnikom omogoča avtomatizacijo poslovnih procesov, ki zahtevajo veliko dokumentov, s čimer prihranijo čas in denar.

Alfresco Content Services vključuje hibridne, krajevne, oblačne in mobilne možnosti namestitve.

Dodatne informacije o Alfresco Content Services so na voljo na spletni strani Alfresco (<http://www.alfresco.com>) in v dokumentaciji Alfresco (<http://docs.alfresco.com>).

4 ARHITEKTURA

Jedro arhitekture storitve Alfresco Content Services je skladišče (angleško repository) za shranjevanje vsebine, ki jo podpira strežnik baz podatkov, ki ohranja vsebino in metapodatke. T.i. »out of the box« aplikacije zagotavljajo standardne rešitve, kot so upravljanje dokumentov, upravljanje zapisov in upravljanje spletnih vsebin. Programski vmesniki podpirajo več jezikov in protokolov, na katerih lahko razvijalci izdelajo aplikacije in rešitve po meri.

Alfresco Content Services je javanska aplikacija, ki se izvaja v aplikacijskem strežniku Tomcat.

Obstajajo trije glavni nivoji:

- Skladišče (Repository tier) - se uporablja za shranjevanje vsebine in metapodatkov
- Spletni nivo – (web tier) - Alfresco Share ali drug uporabniški vmesnik
- Lucene indeks (Index tier) - Zagotavlja napredne storitve iskanja

Obstajajo tudi druge stopnje uporabe ali komponente za servisiranje določenih vrst potreb.

V sistemu Archeia bo prišlo v poštev:

- Transformacija dokumentov (Transformation tier) - obravnava določene vrste sprememb dokumentov
- Iskalnik po vsebini dokumenta (Solr6) – podpira t.i. »full text search«

Več informacij o glavnih sistemskih komponentah se nahaja v dokumentaciji Alfresco:

<http://docs.alfresco.com/6.1/concepts/alfresco-arch-about.html>

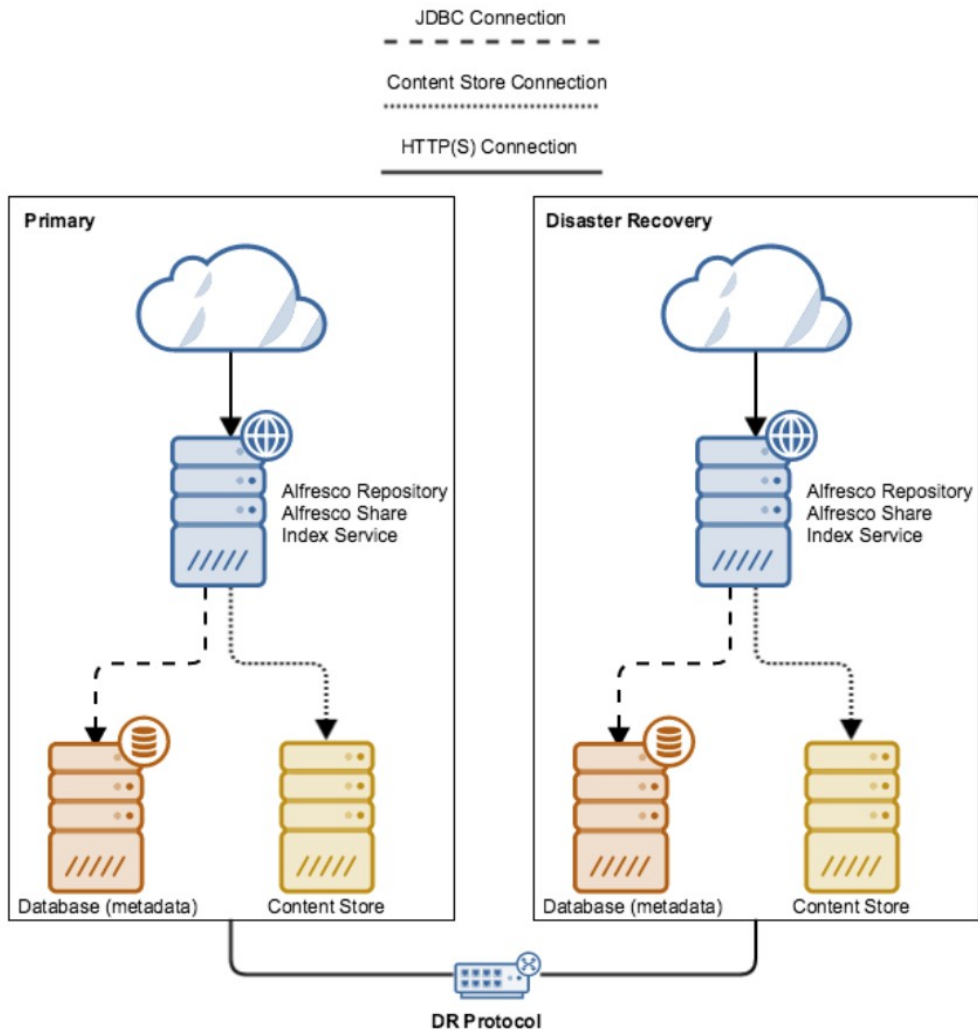
Alfresco Content Services podpira združevanje vozlišč (angleško clustering of nodes). To ponuja možnost horizontalnega skaliranja sistema. Dodatne informacije o vzpostavitvi grozdenja se nahajajo v dokumentaciji Alfresco:

<http://docs.alfresco.com/6.1/concepts/ha-intro.html>

4.1 ARHITEKTURNI DIAGRAM NAMESTITVE SISTEMA

Izbrana je t.i. Business arhitektura, kot prikazuje spodnja slika (glej opis na strani 8 v dokumentu https://www.alfresco.com/sites/www.alfresco.com/files/alfresco_content_services_5.2_reference_architecture.pdf):

Business



4.2 KLJUČNE KOMPONENTE SISTEMA

4.2.1 SKLADIŠČNI IN SPLETNI STREŽNIKI (REPOSITORY AND SHARE SERVERS)

V referenčni arhitekturi obstaja vsaj en strežnik, ki izvaja aplikacijo Alfresco Share. Kjer velikost razmestitve zahteva več vozlišč, dostop do vozlišč opravi naprava za uravnavanje obremenitve (Load balancer). To zagotavlja, da so omrežne zahteve (kot so HTTP (s), FTP in CIFS) enakomerno porazdeljene med več vozlišči.

Opomba: Specifična konfiguracija tega LB je odvisna od protokolov, za katere se pričakuje, da jih bo uporabljal.

4.2.2 STREŽNIK BAZ PODATKOV (DATABASE SERVER)

Vozlišča Alfresco in Share ter indeksni strežniki so konfigurirani za komunikacijo s samostojnim strežnikom baz podatkov. Vsi metapodatki za vsebino so shranjeni v tej bazi podatkov.

4.2.3 SHRAMBA VSEBINE (CONTENT STORE)

Za referenčno arhitekturo je predvidena privzeta shramba vsebine (Content Store). To je shranjevanje vsebine datoteke (File Content Store). Alfresco podpira številne vrste skladišč vsebin, ki so podrobno opisane v uradni dokumentaciji Alfresco.

Shranjevanje vsebine je abstrakcija in uporaba različnih shramb vsebine ima malo vpliva na referenčno arhitekturo.

Alfresco podpira tudi več shramb z vsebinami. Kadar gre za več shranjenih vsebin, mora biti vsaka shramba povezana z vsemi vozlišči skladišča v gruči, kot tudi z vsemi nameščenimi indeksnimi strežniki.

V sistemu Archeia bodo dokumenti shranjeni na lokalnem datotečnem sistemu.

4.2.4 STORITVE ISKANJA ALFRESCO – LUCENE - INDEKSNI STREŽNIK (ALFRESCO SEARCH SERVICES – LUCENE - INDEX SERVER)

Lucene je odgovoren za indeksiranje vsebine in metapodatkov, obdelavo zahtev za iskanje in vračanje rezultatov iskanja pa bo omogočal Solr6, ki je opcijski. Indeksiranje se izvaja na asinhron način in ne vpliva na performance Alfresca. Solr strežnik, ki izvaja indeksiranje, se izvaja na drugem strežniku kot Alfresco repozitorij. Iskanje po besedilu (angl. full text search) bo upošteval slovenski jezik.

Podrobne informacije o Solr in njegovi uporabi v uvajanju Alfresco Content Services so na voljo v uradni dokumentaciji Alfresco:

<http://docs.alfresco.com/5.1/concepts/solr-home.html>

<http://docs.alfresco.com/5.2/concepts/search-home.html>

4.2.5 STORITVE UPRAVLJANJA ALFRESCO (UPRAVLJANJE DOKUMENTOV – RECORDS MANAGEMENT - RM)

Alfresco Government Services – Records Management (RM) zagotavlja funkcionalnost upravljanja zapisov o dokumentih.

Ta komponenta ne spremeni arhitekture namestitve storitev Alfresco Content Services, lahko pa vpliva na zahteve glede shranjevanja, modele dovoljenj in druge vidike uporabe platforme. Način, kako vpliva na shranjevanje, indeksiranje in spreminjanje velikosti podatkovne baze, je odvisen od tega, kako so konfigurirane in uporabljene storitve upravljanja.

4.3 SPECIFIKACIJE STROJNE OPREME

V tabeli spodaj so opisane zahteve za strojno opremo za referenčno arhitekturo.

Server	CPU	RAM (GiB)	Storage	Service	Notes
Alfresco / Share / Index Server	Quad core, Intel Xeon @2.6Ghz (exact model may vary) or better	32	1TB SSD	Alfresco Repository, Share, ActiveMQ, eHramba Web UI	Local disk is used for application files, temporary files, and index.
SOLR6 Server	Quad core, Intel Xeon @2.6Ghz (exact model may vary) or better	32	1TB SSD	SOLR6 Server	Local disk is used for index.
Elastic search stack	Quad core, Intel Xeon @2.6Ghz (exact model may vary) or better	32	1TB SSD	Elastic search, Kibana, APM server	Local disk is used for index.
Cache stack – predpomnilnik	Quad core, Intel Xeon @2.6Ghz (exact model may vary) or better	32	1TB SSD	Predpomnilnik Server	Local disk is used for application files, temporary files, and index.
Postgres	Quad core, Intel Xeon @2.6Ghz (exact model may vary) or better	32	1TB SSD	Postgres DB	Local disk is used for database files.
Content Store	N/A	N/A	Mounted storage, local, SAN, or NAS		Size and performance depends on size of repository. See the Sizing Guide for formulas.

Glede na stanje strežnikov priporočamo zamenjavo diskov (SSD), minimalno naj bodo 1TB, minimalno naj bo na vsakem strežniku 32 GB RAM-a, da bo predpomnilnik zadovoljivo hiter.

Predlagamo pa, da v testnem obdobju izvedemo performančne teste na različnih (tudi manjših) konfiguracijah.

Za testiranje funkcionalnosti sistema Archeia zadošča en strežnik s SSD 1TB ter 32 GB RAM-a.

Za performančno testiranje pa je smiselno imeti enako postavitev kot v produkciji (lahko strežniki z manj kapacitete):

- strežnik za Postgres bazo
- strežnik za Alfresco repozitorij, Share in Active MQ, Archeia admin aplikacija
- strežnik za SOLR
- strežnik za Elastic search, Kibana in APM strežnik
- strežnik za Predpomnilnik.

4.4 SPECIFIKACIJE PROGRAMSKE OPREME

Ta razdelek navaja posebno programsko opremo, potrebno za izdelavo referenčne arhitekture.

4.4.1 PROGRAMSKA OPREMA IN UPORABLJENE TEHNOLOGIJE

Izbran je operacijski sistem Linux.

Podatkovna baza je Postgres.

Aplikacijski strežnik je Tomcat

Programska platforma je Java

Grafični vmesnik bo razvit s spletno platformo VUE, zahteve glede brskalnika: grafični vmesnik mora delovati vsaj na brskalniku Firefox ESR 52.

Avtomatski testi za grafični vmesnik je Selenium + Cucumber, testiranje se bo izvajalo v brskalniku Firefox ESR 52.

Unit testi bodo implementirani z uporabo platforme JUnit. Poleg JUnit testov se uporablja tudi testno ogrodje od produkta Postman, s katerim bo izvedeno določeno funkcijsko in integracijsko testiranje, razen če se izkaže, da Postman glede na zahteve naročnika glede števila uporabnikov in količin dokumentov ni primerna izbira; v tem primeru bo potrebno poiskati in uporabiti drugo primerno orodje.

4.5 PREDPOMNILNIK

Predpomnilnik predstavlja ločen strežnik, ki omogoča hitrejši dostop do datotek in njihovih metapodatkov na oddaljenih lokacijah, kjer je hitrost dostopa do centralnega strežnika Archeia lahko omejena. Predpomnilnik bo implementiran z uporabo obstoječega Alfresco strežnika, ki pa bo vseboval izključno funkcionalnosti, ki so potrebne za delovanje predpomnilnika. Razlog za izbiro "okrnjene" različice Alfresco strežnika je ta, da že v osnovi ponuja vse potrebne funkcionalnosti persistence datotek in metapodatkov na fizični datotečni sistem, beleženje dostopov do fizičnih datotek in metapodatkov, možnost hranjenja v RAM-u ipd. Izvedene pa bodo ustrezne razširitve, ki bodo omogočale komunikacijo s centralnim Archeia strežnikom in posledično zagotavljanje konsistentne med predpomnilnikom in centralnim Archeia strežnikom.

Vsi dokumenti in njihovi metapodatki se vedno pišejo in spreminjajo direktno v centralnem strežniku Archeia. S tem zagotovimo, da se v centralnem strežniku Archeia vedno nahaja zadnja verzija vseh dokumentov in metapodatkov, pa tudi vse morebitne različice dokumentov in metapodatkov.

Pri branju dokumentov iz sistema Archeia je pristop različen. Medtem ko uporabniki vpisnikov do dokumentov v sistemu Archeia dostopajo neposredno, bodo uporabniki aplikacije eSpis do sistema Archeia dostopali preko predpomnilnika. Glavna vloga predpomnilnika je funkcija medpomnilnika (*cache*) za dokumente in metapodatke. Za branje dokumentov v lokalnem omrežju tako ne bo potrebno vsakič dostopati do centralne hrambe, ampak le takrat, ko iskanega dokumenta še ne bo v predpomnilniku. V primeru zahtevka za pridobivanje (branje) novega dokumenta, predpomnilnik le-tega, in njegove metapodatke, ob prvem dostopu pridobi iz sistema Archeia. Pri tem se dokument in metapodatki shranijo v predpomnilniku za vse nadaljnje dostope. Vsi lokalno hranjeni dokumenti bodo shranjeni na dototečnem sistemu v kriptirani obliki. V primeru poizvedovanja za dokumentom ali njegovimi metapodatki, ki so že v predpomnilniku, se centralni sistem Archeia ne kliče. V primeru, da se kateri od dokumentov ali njegovih metapodatkov v centralnem strežniku sistema Archeia spremeni, se za dotični dokument zapiše v »čakalno vrsto« za vse predpomnilnike v sistemu navodilo za izbris dokumenta in njegovih metapodatkov iz predpomnilnika. Navodilo za izbris dobijo vsi predpomnilniki v sistemu. Če je bil določen predpomnilnik v »offline« načinu delovanja, se morajo ob ponovni vzpostavitvi povezave najprej obdelati vse sporočila iz njegove čakalne vrste na centralnem strežniku in šele nato lahko predpomnilnik spet začne streči zahteve klientov.

Čakalna vrsta se nahaja na centralnem strežniku. V tej vrsti se bodo nabirala sporočila, ki jih mora predpomnilnik obdelati, in običajno pomenijo, da mora predpomnilnik, zaradi kakršnihkoli sprememb na datoteki ali metapodatkih, invalidirati vsebino, ki jo hrani. Predpomnilnik bo seveda vedel, kdaj bo izgu-

bil povezavo s čakalno vrsto, in v tistem trenutku bo interno prešel v "offline" način delovanja. V tem režimu delovanja bo lahko še vedno klientom stregel dokumente, ki jih ima trenutno v svojem predpomnilniku. Ko pa se bo predpomnilnik uspel spet uspešno povezati na čakalno vrsto v centralnem strežniku, bo prebral število vseh sporočil v čakalni vrsti in bo prešel v novo stanje npr. "sinhronizacija", v katerem bo obdelal vsa "stara" sporočila iz čakalne vrste, tako da se čim bolj zmanjša možnost, da bi uporabniki aplikacije eSpis delali na stari verziji. Ko se vsa stara sporočila iz čakalne vrste obdelajo, lahko preide predpomnilnik v "normalno" stanje predpomnilnika.

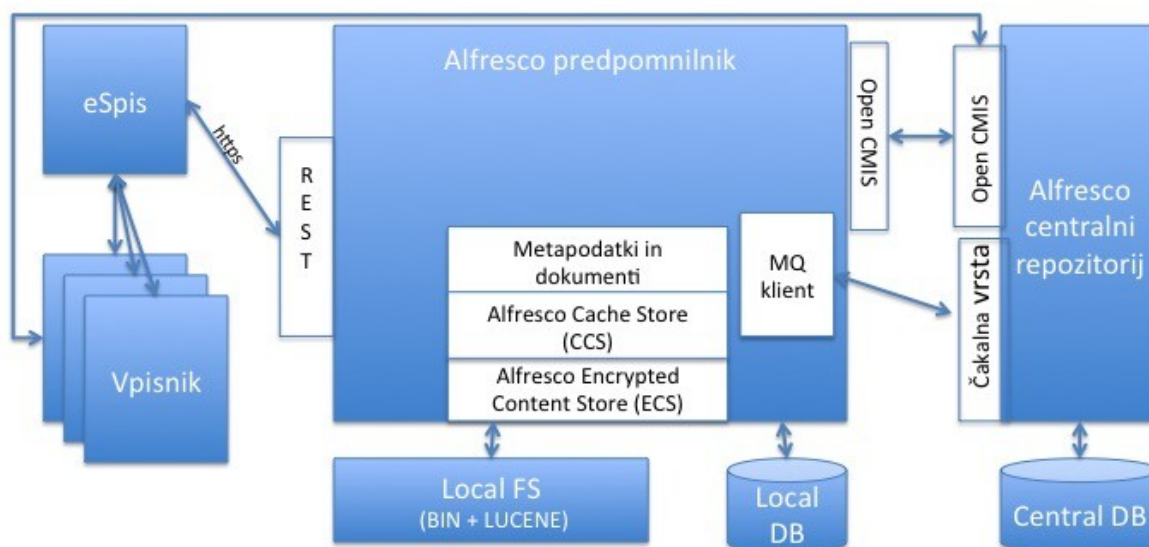
Zaradi narave hrambe dokumentov se ne pričakuje pogostih sprememb na vsebini dokumentov in metapodatkov, zato se tudi ne pričakuje, da bi bilo teh sporočil v čakalnih vrstah veliko. Tako da bo čas, ko bo predpomnilnik v stanju "sinhronizacija", relativno kratek.

Predpomnilnike bo v sistem Archeia moral registrirati administrator sistema z enkratnim identifikatorjem (UUID – namenjen predvsem za potrebe logiranja), ključem (shared key) in geslom. V sistem se bodo lahko prijavili izključno en klient s točno določenim UUID-jem. V primeru poizkusa prijave več klientov z istim UUID-jem se bo generiral alarm, ki bo posredovan administratorjem sistema Archeia.

Predpomnilnik bo fizično implementiran na krajevnih strežnikih z operacijskim sistemom Linux.

Predpomnilnik mora delovati tudi v offline načinu; to je stanje, ko predpomnilnik izgubi internetno povezavo s centralnim strežnikom Archeia. Čas za opozorila (čas brez povezave) bo definiran v administratorskem modulu.

Spodnja slika prikazuje delovanje predpomnilnika z eSpisom in centralnim repozitorijem:



Predpostavljamo, da bo za dostopne pravice uporabnikov do predpomnilnika poskrbel eSpis (popis spisa bo eSpis dobil od ustreznega vpisnika, ki bo poskrbel, da bo vrnil le seznam dokumentov, do katerih

ima uporabnik pravico vpogleda). Logika eSpisa bo poskrbela, da bo uporabnik lahko vpogledoval preko eSpisa le v tiste dokumente, za katere mu je pravice dovoli ustrezen vpisnik. Povezava med eSpisom in predpomnilnikom bo varna (https). Rešitev, kako bo eSpis aplikacija vedela, preko katerega predpomnilnika pridobivati informacije, bo implementirana v sistemu Archeia, dodane bodo fiksne nastavitve v konfiguracijski datoteki. Podrobneje bo opisano v namestitvenem dokumentu.

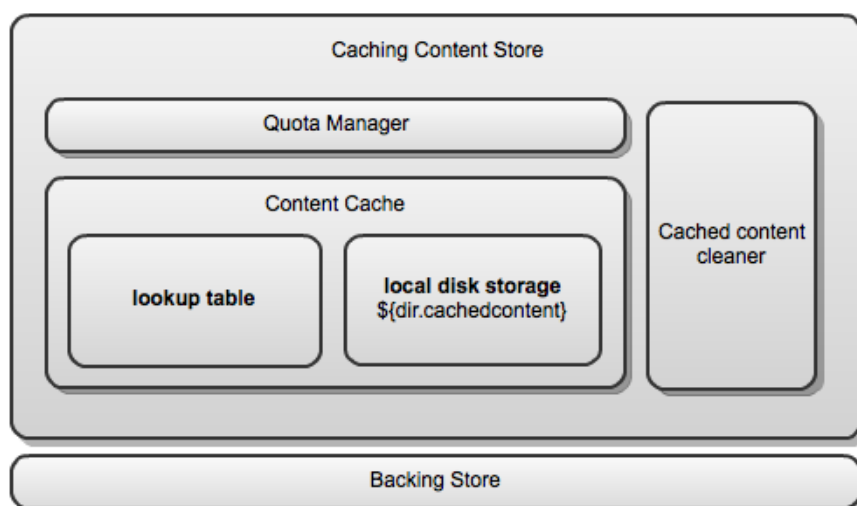
Varnostni mehanizmi so opisani v poglavju Varnost REST Vmesnikov ter Avtentikacija in avtorizacija uporabnikov.

4.5.1 ALFRESCO CACHE STORE - SHRANJEVANJE VSEBINE V PREDPOMNILNIKU (CCS)

CSS omogoča hitri dostop do dokumentov in njegovih metapodatkov. Dokumenti znotraj Alfresco Cache Store sistema niso v kriptirani obliki, ampak se nahajajo izključno v notranjem pomnilniku strežnika (RAM). Zato je dostop do datotek in metapodatkov, ki se nahajajo v pomnilniku strežnika izredno hiter, saj vsebine datotek ni potrebno dešifrirati, preden se posredujejo uporabniku.

Uporaba razreda CachingContentStore (CachingContentStoreCl Class) doda pregledno predpomnjenje vsakemu izvajanju ContentStore ter izboljšuje hitrost dostopa.

Spodnji Diagram prikazuje arhitekturo CCS:



Glavni razredi in vmesniki, ki tvorijo vsebino CCS, so:

- **CachingContentStore:** To je glavni razred, ki implementira vmesnik ContentStore in ga je zato mogoče uporabiti v poljubni kombinaciji z drugimi ContentStore-i. CachingContentStore obravnava vso logiko visokega nivoja interakcije med predpomnilnikom in fizičnim datotečnim sistemom (Backing Store), medtem ko je predpomnjenje omogočeno s pomočjo sodelujočega objekta ContentCache.
- **ContentCache:** Ta razred je odgovoren za umeščanje elementov in pridobivanje predmetov iz predpomnilnika. Posamezna implementacija (ContentCacheImpl) za ta razred uporablja iskalno tabelo (lookup table) za sledenje datotek, ki jih upravlja predpomnilnik, in direktorij v lokalnem datotečnem sistemu za shranjevanje datotek predpomnjenih vsebin.

- **QuotaManagerStrategy:** Ta vmesnik implementirajo t.i. upravitelji kvot (Quota managers) in nadzorujejo kako je porabljen disk za predpomnjeno vsebino. Za to Alfresco ponuja dve izvedbi: **UnlimitedQuotaStrategy** (ne omejuje uporabe diskov in s tem onemogoča funkcijo kvote) in **StandardQuotaStrategy** (poskuša ohraniti uporabo pod določenim maksimumom, podanim v bajtih ali MB).

Razred **CachingContentStore** je zelo konfigurabilen in mnoge njegove komponente bi lahko zamenjali za druge izvedbe, če bi se to izkazalo kot smiselno ali potrebno.

Opravo za brisanje predpomnjene vsebine (**CachedContentCleaner**) periodično pregleduje strukturo imenika, ki vsebuje datoteke predpomnjenih vsebin, in izbriše datoteke vsebine, ki jih predpomnilnik ne uporablja. Datoteke predpomnilnika se ne uporabljajo, če nimajo vnosa v pregledni tabeli, ki jo implementira **ContentCacheImpl**. Opravo za brisanje predpomnilnika vsebine ni del arhitekture, temveč je pomožni objekt za **ContentCacheImpl** in mu omogoča učinkovitejše delovanje.

4.5.2 LASTNOSTI **CACHINGCONTENTSTORE** (**CACHINGCONTENTSTOREPROPERTIES**)

Za razred **CachingContentStore** je možno nastaviti številne lastnosti (properties).

Naslednje lastnosti se uporabljajo v vzorčni nastavitveni datoteki, **caching-content-store-context.xml.sample** in jo je možno nastaviti v datoteki **alfresco-global.properties**. Njihove privzete vrednosti so na voljo v datoteki **repository.properties**. Po dogovoru z naročnikom bodo posamezne vrednosti prilagojen naročnikovim zahtevam.

- **system.content.caching.cacheOnInbound = true**

Omogoči predpomnjenje s pisanjem. Če je »true«, poskus zapisovanja vsebine na datotečni sistem povzroči, da je predmet v predpomnilniku. Ko je element prvič prebran (pod pogojem, da element v tem času ni bil odstranjen iz predpomnilnika), je datoteka že lokalno shranjena za hitrejši dostop. Priporočljivo je, da je ta lastnost nastavljena na »true« za večino scenarijev uporabe.

- **system.content.caching.maxDeleteWatchCount = 1**

Določa, kolikokrat mora datoteka zadostiti pogoju izbrisa iz predpomnilnika, preden se dejansko lahko pobriše. Privzeta vrednost je vedno nastavljena na 1, vendar se lahko poveča, če bralcev (readers), pridobljenih iz predpomnilnika, ni mogoče uporabiti zaradi izbrisa osnovne datoteke.

- **system.content.caching.contentCleanup.cronExpression = 0 0 3 * * ?**

Določa, kako pogosto se bo izvajalo opravilo za čiščenje predpomnjene vsebine. Priložena vrednost je *quartz* izraz in je podobna izrazu *Unix cron*. V tem primeru bo opravilo za brisanje predpomnjene vsebine delal vsako jutro ob 3. uri.

- **system.content.caching.timeToLiveSeconds = 0**

Določa čas v sekundah, po katerem se element briše iz predpomnilnika. Po preteku tega časa element ne bo več predpomnjen in zahteva za URL vsebine bo povzročila, da bo element pridobljen iz datotečnega sistema in ponovno shranjena. Vrednost 0 pomeni, da elementi ne bodo imeli parametra TTL in iz predpomnilnika ne bodo brisani.

- **system.content.caching.timeToldleSeconds = 60**

Določa najdaljši čas, ko lahko element v predpomnilniku obstaja brez zahteve zanj. Vsakič, ko se dostopa do elementa, se parameter Time To Idle osveži in element ostane v predpomnilniku.

- `system.content.caching.maxElementsInMemory = 5000`

Velja za iskalno tabelo v ContentCache. Vsak URL vsebine potrebuje dva vnosa v iskalni tabeli, tako da lahko vrednost 5000 omogoči shranjevanje 2500 postavk vsebine v pomnilnik za iskalno tabelo.

- `system.content.caching.maxElementsOnDisk = 10000`

Velja za iskalno tabelo v ContentCache. Vsak URL vsebine potrebuje dva vnosa v iskalni tabeli, tako da lahko vrednost 10000 omogoči shranjevanje 5000 elementov na disku.

- `system.content.caching.minFileAgeInMillis = 2000`

Določa, da morajo biti datoteke vsaj te starosti, preden so označene za brisanje. To tudi ustavi nepotrebna preverjanja, kot je nalaganje in pregledovanje datoteke z lastnostmi.

- `system.content.caching.maxUsageMB = 4096`

Določa največjo porabo diska v MB, ki bi jo vsebovala predpomnjena vsebina. Z drugimi besedami, ta lastnost določa kvoto prostora na disku, dodeljeno imeniku `$ {dir.cachedcontent}`. Uporablja ga razred `StandardQuotaStrategy`, kot je konfiguriran v datoteki `caching-content-store-context.xml.sample`.

- `system.content.caching.maxFileSizeMB = 0`

Določa največjo velikost posamezne datoteke predpomnjene vsebine v MB. Vsebina, ki je večja od te velikosti, je še vedno mogoče pridobiti z razredom `CachingContentStore`, vendar vsebina ne bo predpomnjena. Če je ta lastnost nastavljena na nič, potem ne velja omejitev velikosti za posamezne datoteke. To lastnost uporablja razred `StandardQuotaStrategy`, kot je konfigurirana v datoteki `caching-content-store-context.xml.sample`.

Konfiguracija `CachingContentStore` je opisana v namestitvenem dokumentu.

Podrobneje o CCS in o primerih uporabe je na voljo na povezavi:

<https://docs.alfresco.com/5.1/concepts/ccs-overview.html>

4.5.3 ALFRESCO ENCRYPTED CONTENT STORE (ECS)

ECS se uporablja za šifriranje vsebine v fazi zapisovanja na fizični datotečni sistem. To se izvede s premešanjem navadnega besedila v šifrirano besedilo (šifriranje) in nato spet nazaj (dešifriranje) s pomočjo simetričnih in asimetričnih ključev.

Z uporabo dodatka `Alfresco Encrypted Content Store` nepooblaščenim osebam preprečujemo neposreden vpogled v vsebino datotek, ki se nahajajo na datotečnem sistemu.

Predn je dokument zapisan v shrambo šifrirane vsebine, ECS uporabi simetrično šifriranje za šifriranje dokumenta, predn ga zapiše v shranjeno vsebino. Nov simetrični ključ se ustvari vsakič, ko se dokument zapiše v shrambo vsebine. To pomeni, da je vsak dokument v sistemu šifriran z drugačnim simetričnim ključem. Še več, asimetrično šifriranje (kot je RSA) se uporablja za šifriranje / dešifriranje

teh simetričnih šifrirnih / dešifrirnih ključev. Asimetrično šifriranje uporablja glavni ključ, ki je izbran iz niza konfiguriranih glavnih ključev.

ECS šifrira vsebino z glavnim ključem, ki je naključno izbran iz skupine glavnih ključev. Ni na voljo nadzora za uporabo določenega glavnega ključa, saj bi sicer morebitni napadalci lahko ciljali na določene glavne ključke, ko poskušajo dostopati do vsebine ali jo spreminjati.

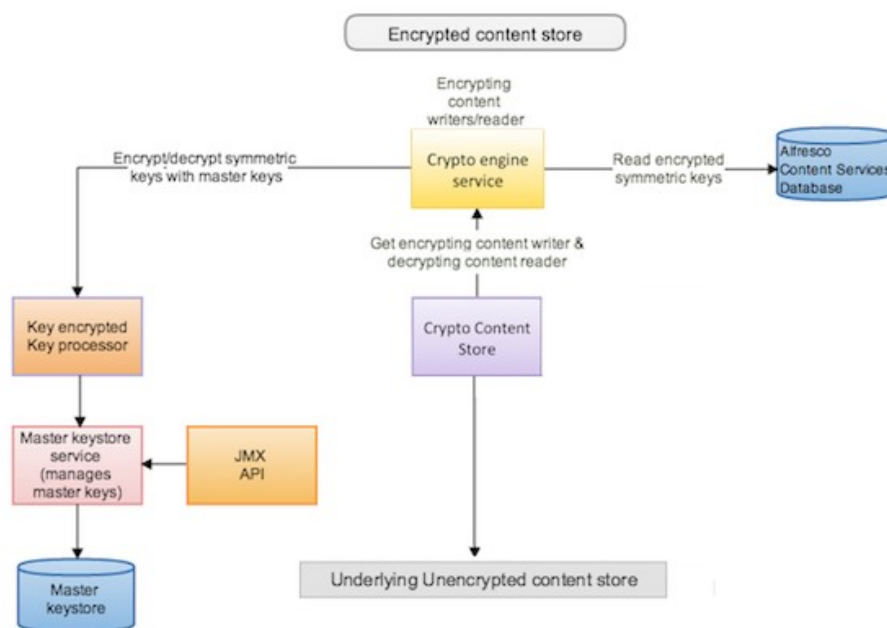
Alfresco Content Services uporablja niz glavnih ključev, ki so:

- naključno izbrani
- shranjene z geslom v t.i. "key store"-u
- se lahko "umaknejo (retired)" v primeru kraje ključev ali kot del standardnega procesa umika.

Skladišče (repozitorij) ve, kateri glavni ključ je bil uporabljen za šifriranje danega simetričnega ključa, tako da lahko, ko uporabnik želi prebrati določen dokument, skladišče dešifrira simetrični ključ (z uporabo tega ključa) in nato uporabi dešifrirano simetrično tipko za dešifriranje vsebine dokumenta.

Pomembno: Alfresco Content Services ne shranjuje navedenega / izbranega glavnega ključa. Glavni ključi so zakriptirani v shrambi ključev, do katerega lahko dostopamo. Če ni mogoče dostopati do tega ključa, vsebine ni mogoče dešifrirati.

Naslednji diagram prikazuje izvajanje šifriranja vsebine s šifrirano shrambo vsebine nad privzeto shrambo vsebine.



Pomembno: Če je shramba šifrirane vsebine omogočena na obstoječi ali nadgrajeni namestitvi storitve Alfresco Content Services, bo šifrirana samo nova vsebina, obstoječa vsebina ne bo šifrirana.

Namestitev ECS in upravljanje glavnih ključev je del namestitvenega dokumenta.

4.5.4 LASTNOSTI SHRANJENE ŠIFRIRANE VSEBINE (PROPERTIES)

Za shrambo šifrirane vsebine je treba nastaviti številne lastnosti. Te lastnosti nastavite v datoteki `alfresco-global.properties`.

- `filecontentstore.subsystem.name`
Omogoča podsistem ECS, na primer, `EncryptedContentStore`.
- `cryptodoc.jce.providerName`
Določa ime Java Security ponudnika. Če je prazno, označuje uporabo privzetega ponudnika. Izbrati je mogoče tudi svojega ponudnika tako, da se to lastnost nastavi na ime razreda ponudnika. Če določeno ime ponudnika ni nastavljeno, sistem izbere najprimernejšega ponudnika.
- `cryptodoc.jce.keystore.type`
Podaja tip shrambe ključev, na primer `jceks`.
- `cryptodoc.jce.keystore.path`
Podaja pot do shrambe ključev, ki vsebuje glavne ključe, na primer `/opt/alfresco/my_key.jks`.
- `cryptodoc.jce.keystore.password`
Določa geslo ključa, na primer `'geslo'`.
- `cryptodoc.jce.keyaliases`
Podaja seznam z vzdevki ločenih imen / imen glavnih ključev v glavnem ključu, na primer `mkey1`, `mkey2`. To so vzdevki, ki se uporabljajo z orodjem `keygen`, na primer `encstore`.
- `cryptodoc.jce.key.passwords`
Določa seznam gesel, ločenih z vejicami, ki se uporabljajo za nalaganje ključev iz shrambe glavnega ključa. Položaj gesla se ujema s položajem ustreznega vzdevka ključa v lastnostih `cryptodoc.jce.keyaliases`. To geslo se uporablja z ukazom `keytool` in se lahko razlikuje od glavnega gesla. Na primer `geslo`, `geslo`.
- `cryptodoc.jce.keygen.defaultSymmetricKeySize`
Določa velikost ključa za simetrične ključe, ki se uporabljajo za šifriranje / dešifriranje vsebine dokumenta.
- Opomba: Privzeta velikost simetričnega ključa je 128 bitov. Uporabniki, ki želijo boljšo ključno moč, bi morali prenesti in namestiti datoteke s pravili neomejene moči JPS Cryptography Extension (JCE) za JRE.
- `cryptodoc.jce.keygen.defaultSymmetricAlgorithm`
Določa algoritem simetričnega ključa.

Naslednje lastnosti se uporabljajo za ponovno šifriranje simetričnih ključev (za preklic glavnega ključa).

- `cryptodoc.symmetricKey.reencryption.batch.size`
Določa število simetričnih ključev, ki so ponovno šifrirani v vsakem paketu, na primer 200.
- `cryptodoc.symmetricKey.reencryption.numThreads`
Podaja število niti, ki naj bodo uporabljene za izvedbo ponovnega šifriranja, na primer 4.

Pot do gesel, gesla, vzdevki in njihova gesla so splošne lastnosti, ki jih je mogoče predpisati za konfiguriranje šifrirane vsebine z datoteko `alfresco-global.properties`.

Vmesnik JMX razkrije (exposes) te lastnosti in omogoča uporabniku, da jih spremeni za tekoči sistem. Glej tudi <https://docs.alfresco.com/tasks/encrypted-jmx.html>.

Več o ECS je na voljo na naslednji povezavi:

<https://docs.alfresco.com/5.2/concepts/encrypted-cs-home.html>

4.5.5 MESSAGE QUEUE (MQ) KLIENT

Message Queue (MQ) klient se uporablja za dvosmerno komunikacijo med predpomnilnikom in centralnim Archeia strežnikom. V smeri od predpomnilnika proti centralnemu Archeia strežniku se bodo pošiljale informacije o dostopih do dokumentov in njihovih metapodatkih na predpomnilniku.

(revizijska sled). V smeri od centralnega Archeia strežnika proti predpomnilniku se bodo pošiljali podatki o spremembah na datotekah in njihovih metapodatkih.

Za MQ se bo uporabil obstoječi ActiveMQ strežnik na centralnem Archeia strežniku, saj je že integriran v Archeia strežnik. Spremljanje delovanja vrst bo možno v administratorski aplikaciji.

4.6 REVIZIJSKA SLED

Modul za revizijsko sled je del sistema Archeia. S svojimi funkcionalnostmi mora zadostiti zahtevam predpisov (ZVDAGA, ETZ, ZVOP-2).

Revizijska sled vpogledov v dokument bo zagotavljala nespremenljivost zapisov ter onemogočala njihovo brisanje in vrivanje med obstoječe zapise.

V revizijski sledi bodo beleženi vsi vpogledi v dokumente. Ob neposrednem dostopu do dokumenta (brez predpomnilnika) bo hkrati z zahtevo po dokumentu v sistem Archeia posredovan tudi podatek o konkretnem uporabniku, ki vpogled v dokument zahteva. Te rešitve Alfresco privzeto ne omogoča in jo je potrebno dograditi. Enak podatek bo posredovan tudi pri dostopu do dokumenta preko predpomnilnika, vendar pa je pri tej vrsti dostopa potrebno upoštevati še to, da se ob podani zahtevi dokument prenese v predpomnilnik, kasneje pa do predpomnilnika in do dokumenta v njem lahko dostopajo različni uporabniki. Tudi ti dostopi do dokumenta v predpomnilniku se morajo zabeležiti v revizijsko sled. Te vpoglede bo sistem Archeia shranjeval v ločeno revizijsko sled (tudi zaradi zagotavljanja offline načina delovanja).

Iz predpomnilnikov se prenaša revizijska sled v ločene tabele (ki pa so vsebinsko enake centralnim revizijskim tabelam) v centralni Alfresco strežnik. Vsi dostopi in vse akcije znotraj predpomnilnika pa se obnašajo povsem enako kot znotraj centralnega Alfresco strežnika.

Iskanje po revizijski sledi (po centralni in predpomnilniški) bo potrebno implementirati glede na zahteve, ki jih vsebinsko še dorekamo, zato v tem trenutku risanje slike ni smiselno. Predvidevamo, da bo iskanje vedno vrnilo tudi podatek, ali se sled nahaja v centralni ali v predpomnilniški revizijski sledi.

Ko je predpomnilnik v offline načinu (se pravi, nima povezave do centralnega Alfresco strežnika), se vsi dostopi beležijo lokalno. Ko se spet vzpostavi povezava s centralnim Alfresco strežnikom, se podatki prenesejo v centralni Alfresco strežnik.

Ob ponovni vzpostavitvi povezave se sproži invalidacija (razveljavitev) podatkov v predpomnilniku za tiste dokumente in metapodatke, pri katerih je v offline načinu predpomnilnika v centralnem Alfresco strežniku prišlo do sprememb (izbrisi dokumentov, spremembe metapodatkov ipd). V praksi to pomeni, da bo tak dokument z metapodatki iz predpomnilnika zbrisan in ob ponovni zahtevi po pregledu tega dokumenta se bo le-ta prebral iz centralnega Alfresco strežnika in shranil v predpomnilnik.

Smiselno pa je tudi definirati čas, v katerem se dokument v predpomnilniku hrani oz. kdaj se tak dokument samodejno izbriše iz predpomnilnika. Primer: Uporabnik dela na zadevi, za katero potrebuje nekaj dokumentov te ali katere druge zadeve. Ker v njih vpogleduje, se ti dokumenti shranijo v predpomnilniku. Uporabnik po nekem času z zadevo zaključi in do teh dokumentov ni več vpogledov x dni. Smiselno je, da se taki dokumenti po x dneh brišejo iz predpomnilnika. X je lahko poljubno število.

4.7 MIGRACIJA IN REPLIKACIJA DOKUMENTOV

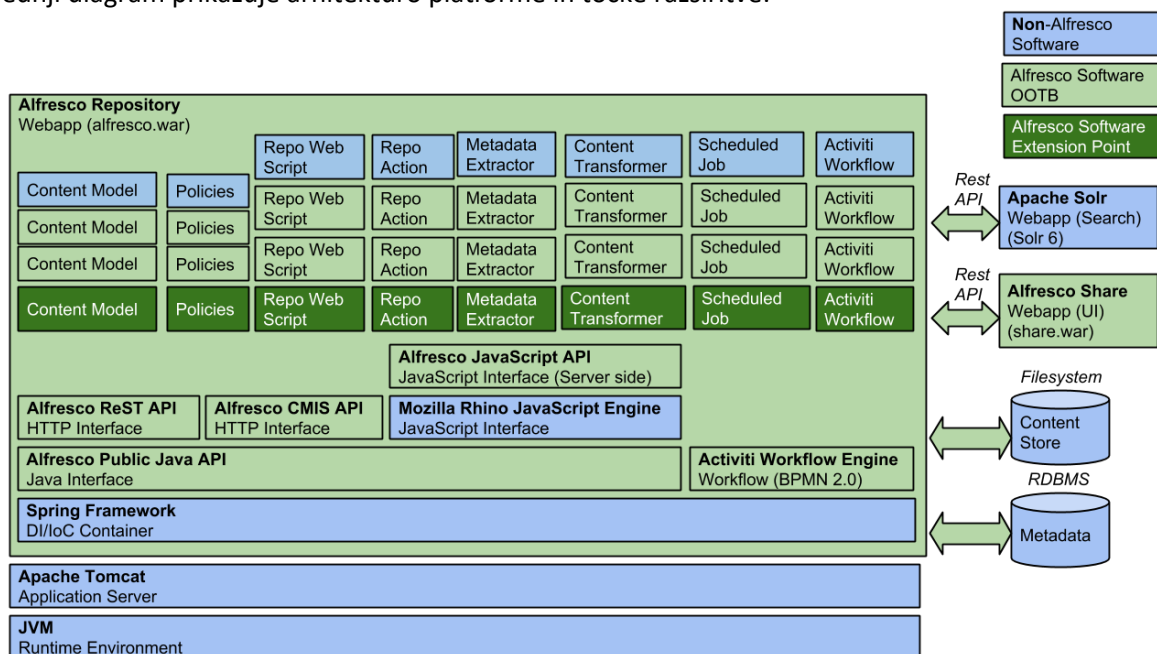
V sodstvu je za hrambo dokumentov v povezavi z nekaterimi aplikacijami in vpisniki že v uporabi sistem Alfresco. Te aplikacije oziroma vpisniki uporabljajo različna načina za naslavljanje dokumentov: nekatere jih naslavlja po ID-jih, druge po logični poti.

Pri migraciji dokumentov in podatkov iz obstoječega Alfresca v sistem Archeia bo treba poskrbeti, da se v primeru aplikacij, ki uporabljajo naslavljanje po ID-jih, ti ID-ji ne izgubijo, če vsak sistem za hrambo dodeli hranjenemu dokumentu interni ID, in to vsak sistem (če jih je več) drugačnega. Migracija dokumentov in podatkov bo morala biti izvedena tako, da bo replikacija podatkov in dokumentov celovita in pravilna.

5 SHEMA SESTAVE PROGRAMSKE OPREME IN POVEZAV MED NJENIMI DELI

Arhitektura platforme je sestavljena iz skladišča (repozitorija) in povezanih storitev. Platforma vsebuje ključne razširitvene točke za izgradnjo lastnih razširitev.

Naslednji diagram prikazuje arhitekturo platforme in točke razširitve:



Platformo sestavljajo:

- repozitorij in vse storitve
- razširitvene točke razvijalcev (<https://docs.alfresco.com/dev-platform-extension-points.html>)
- in API-ji (<https://docs.alfresco.com/dev-api-intro.html>).

Skladišče zagotavlja shranjevanje dokumentov in druge vsebine. Metapodatki dokumentov oz. vsebine so shranjeni v relacijski bazi podatkov, vsebina pa je shranjena neposredno v datotečnem sistemu. Razmerja med postavkami vsebine in njihovimi različnimi lastnostmi (metapodatki) so definirana v enem ali več modelih vsebine (glej tudi <https://docs.alfresco.com/./references/dev-extension-points-content-model.html>).

6 VARNOST REST VMESNIKOV TER AVTENTIKACIJA IN AVTORIZACIJA UPORABNIKOV

Varnost med vpisniki in sistemom Archeia bo zadoščena z naslednjimi ukrepi:

- uporaba reverse-proxy strežnika pred Archeia strežnikom. Reverse proxy bo nastavljen, da bo do Archeiae strežnika možen promet samo iz IP-jev, kjer se bodo izvajali posamezni vpisniki. Na reverse-proxy-ju bodo zaprte tudi vse direktne končne točke (endpoint-i), ki niso potrebni za OpenCMISprotokol
- uporaba varne povezave (https)
- vsak vpisnik se bo moral prijaviti v sistem Archeia
- vpisniki bodo uporabljali za "priklop" na Archeiao OpenCMIS protokol

V primeru, da kateri od vpisnikov ne bo mogel uporabljati OpenCMIS protokola, bo možna komunikacija tudi preko REST vmesnikov, ki pa bodo ščiteni z uporabo Keycloak strežnika in OpenID Connect protokola.

Aplikacija eSpis se bo priklapljala na REST vmesnik predpomnilnika. Vsi ti REST vmesniki bodo ščiteni z uporabo Keycloak strežnika in OpenID Connect protokola. Omejevanja pravic dostopa do dokumentov ta verzija rešitve Archeia ne omogoča, bo pa potrebno to zagotoviti pri integraciji z aplikacijo eSpis.

Konfiguracija za vse varnostne nastavitve in nginx reverse-proxyju bo navedena v namestitvenem dokumentu.

Avtorizacija uporabnikov bo realizirana preko LDAP, ki ga ureja aplikacija Razpored in ki je v domeni naročnika. Avtentikacija uporabnikov (tako vpisnikov kot fizičnih uporabnikov) bo preko Oauth2 protokola in OpenIDConnect, ki je vmesni protokol med aplikacijo Razpored in LDAP.

Dogovorjeno je, da se pripravi protokole in razvije konektorje znotraj KeyCloak in slednje poveže z aplikacijo Razpored.

Keycloak je Open Source Identity and Access Management Server, ki je skladen z OAuth2 and OpenID Connect(OIDC) protokoli.

Dokumentacija Keycloak predlaga tri načine, kako zavarovati REST APIje, izbran je način uporabe OpenID Connect (OIDC) + Oauth2. Več o Keycloak se nahaja na povezavi <https://medium.com/@bcarunmail/securing-rest-api-using-keycloak-and-spring-oauth2-6ddf3a1efcc2>.