AiP Beroun s.r.o., http://www.aipberoun.cz, tel.: +420 311 611 237, fax: +420 311 611 238
Talichova 807, Beroun, 266 01; IČO: 25778943, DIČ: 25778943

REGISTR DIGITALIZACE HISTORICKÝCH FONDŮ

Softwarové řešení projektu

Technická specifikace

verze 1.0

Ing. Karel Kučera, AiP Beroun s.r.o.

Obsah

1 Úvod o dokumentu

1.1 Účel

Tento dokument je součástí technické dokumentace softwarového řešení projektu „Registr digitalizace historických fondů". Dokument by měl dále sloužit jako jeden z výchozích informačních zdrojů pro další rozvoj tohoto systému a také při rozvoji dalších již existujících systémů v oblasti zpřístupnění historických fondů, jako je např. Manuscriptorium. Dokument jistě také pomůže i při návrhu a vývoji dalších kooperujících softwarových systémů v této oblasti.

1.2 Předpokládaný čtenář

Tento dokument je určen především pro zadavatele (Národní knihovna České republiky). Dále je určen všem, kteří se budou podílet na dalším rozvoji systému „Registr digitalizace historických fondů" a také těm, kteří se podílejí na rozvoji projektu Manuscriptorium a s ním souvisejících projektů v oblasti historických fondů.

1.3 Termíny a konvence

Základní termíny jsou popsány v dokumentu [1]. Další termíny jsou vysvětleny v této kapitole.

Server RDHF – serverová část softwarového řešení projektu „Registr digitalizace historických fondů"

Běžný uživatel – uživatel, který využívá služby RDHF prostřednictvím veřejného uživatelského prostředí

Správce databáze (Správce) – uživatel, který je zodpovědný za vytváření a správu záznamů v některé z databází RDHF (digitální kopie, konkordance)

Administrátor systému (Administrátor) – osoba zodpovědná za chod systému RDHF, zajišťuje např. import dat do systému z externích zdrojů (Manuscriptorium), zálohování dat RDHF a další s tím spojené činnosti. V první fázi projektu zajišťuje také správu datových úložišť.

DigCopyId – persistentní identifikátor digitální kopie fyzického dokumentu. Tento identifikátor je zároveň unikátním klíčem v tabulce digitálních kopií rdhf_main.

1.4 Reference

V dokumentu se odkazujeme na následující literaturu:

[1] AiP Beroun, „Vývoj registru digitalizace pro historické dokumenty, analýza projektu, v. 1.0," Beroun, 2015.

[2] AiP Beroun, „Vývoj URI resolveru pro historické dokumenty, analýza projektu, v.1.0," Beroun, 2015.

[3] NKČR, AiP Beroun, http://www.manuscriptorium.com/sites/default/files/docs/manuscriptorium_visk6_definice.pdf.

[4] „The Open Archives Initiative Protocol for Metadata Harvesting," [Online]. Available: http://www.openarchives.org/OAI/openarchivesprotocol.html.

1.5 Seznam obrázků

2 Úvod

Tento dokument popisuje stav systému Registru digitalizace historických fondů (dále RDHF) v okamžiku jeho uvádění do poloprovozu. Nejsou zde proto zohledněny následně realizované úpravy na serverové části a v klientských aplikacích.

Pro účely poloprovozu byla zrealizována základní verze systému, která umožňuje provádět nejdůležitější činnosti Registru digitalizace historických fondů s dostupnými daty. RDHF byl v této fázi projektu naplněn importem metadat k digitálním kopiím fyzických historických dokumentů z projektu Manuscriptorium.

V této verzi zatím není k dispozici kompletní správa uživatelů. Nebyl také ještě realizován mechanismus pro jejich diferencovaný přístup k datům organizace, kterou reprezentují. V pilotním projektu se předpokládá manipulace s daty (digitální kopie, konkordance) úzkou skupinou uživatelů, kteří mají přístup ke všem datům. Ti zároveň v rámci poloprovozu ověří navržené postupy, funkčnost celého systému a zhodnotí také uživatelskou přívětivost aplikací pro správu dat obsažených v Registru digitalizace historických fondů.

Při přípravě realizace se autoři setkali s řadou více či méně neočekávaných potíží. Problematické jsou především chybějící informace o digitálních kopiích a jejich umístění na datových úložištích. Metadatové záznamy Manuscriptoria neobsahují technická metadata o digitálních kopiích. Získat tyto informace od zahraničních přispěvatelů Manuscriptoria v reálném čase bylo mimo možnosti realizačního týmu. Bylo možno získat pouze technická metadata digitálních kopií, které vznikly na území ČR především v rámci projektu VISK 6. V dalších fázích projektu bude RDHF v případě potřeby o tato metadata doplněn.

Mimo jiné i z důvodu nedostupnosti technických metadat bylo přistoupeno k zjednodušené identifikaci datových úložišť pouze podle URL adres digitálních kopií na nich umístěných. Digitální kopie k jednomu fyzickému dokumentu na jednom datovém úložišti mohou být proto rozlišeny pouze pořadovým číslem v rámci tohoto datového úložiště.

Přesto i přes některé kompromisy je současná verze systému RDHF plně schopná nasazení v poloprovozu. Od něj si autoři slibují kromě důkladného otestování provozuschopnosti systému také odhalení všech při návrhu nepředvídaných situací, které mohou případně způsobit provozních problémy.

3 Koncepce Registru digitalizace historických fondů

3.1 Základní požadavky na systém

Zásadní rozdíl mezi stávajícím Registrem digitalizace a řešením pro registraci digitálních kopií historických dokumentů vychází ze skutečnosti, že ve druhém případě se vždy jedná o vždy o sběr a správu informací o digitalizaci konkrétních fyzických exemplářů. Stávající Registr digitalizace má mimo jiné za cíl vyvarovat se vícenásobné digitalizaci jedné bibliografické jednotky na jednom nebo více digitalizačních pracovištích. Naproti tomu snahou vznikajícího Registru digitalizace historických fondů v tomto ohledu je naopak soustředit informace o všech dostupných digitálních kopiích jednoho fyzického historického dokumentu.

Digitální kopie takového fyzického dokumentu musí kromě obsahové stránky předlohy také přesně zachovat i informaci o jejím fyzickém stavu. Tato informace může mít především v oblasti velmi starých a vzácných exemplářů velkou důležitost. V případě některých typů historických dokumentů musí digitální kopie také zachovat informaci o odlišnostech digitalizovaného fyzického dokumentu od dalších exemplářů téže bibliografické jednotky. Těmito odlišnostmi mohou být např. provenienční znaky jako razítka, exlibris, supralibros, specifické vazby, vpisky aj. Na rozdíl od běžné digitalizace dokumentů jako bibliografických jednotek, registr digitálních kopií jednotlivých fyzických exemplářů musí brát v úvahu také fakt, že v některých případech může být pro jeden konkrétní exemplář žádoucí existence více digitálních kopií (různých technických parametrů), zhotovených pro různé účely a použití. Důležité jsou proto také technické parametry digitalizace i výsledné digitální kopie obsažené v technických metadatech digitální kopie dokumentu. Registr digitalizace historických fondů musí proto zajistit kompletní vstup a správu informací o všech dostupných digitálních kopiích exemplářů historických dokumentů.

Neméně důležitou pro koncepci RDHF je také nutnost zachování informace o vztahu mezi konkrétním exemplářem a jeho digitální kopií (kopiemi). Jak bylo podrobně popsáno v [1], nejspolehlivější metodou identifikace fyzického dokumentu (byť ne stoprocentně spolehlivou) je vytvoření identifikátoru exempláře pomocí lokačních údajů obsažených v metadatech, existujících k danému dokumentu. Zachování jednoznačné identifikace dokumentu v čase (persistentní identifikace) je pak možno zajistit vytvořením mechanismu pro sledování změn umístění (a tím také změn lokačních údajů v metadatech). Tento mechanismus byl nazván „Digitální konkordance". Digitální konkordance umožňuje identifikovat konkrétní historický dokument na základě znalosti jakýchkoliv lokačních údajů, které jsou pro tento dokument známy (a uloženy v systému konkordancí RDHF).

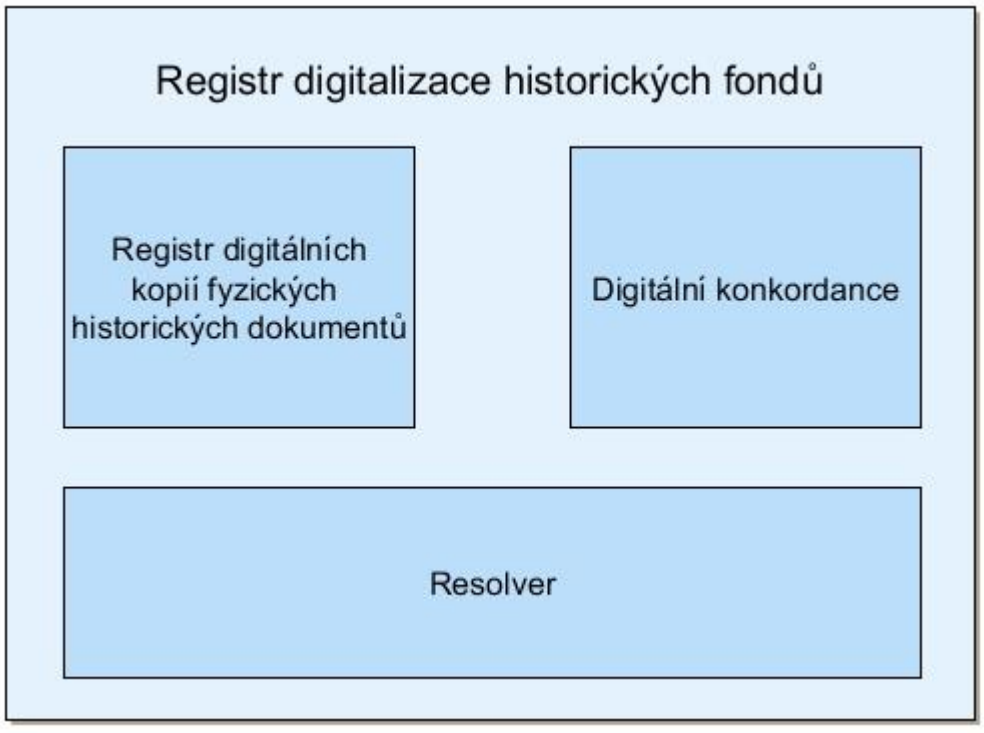
Konečně URI resolver musí být nedílnou součástí Registru digitalizace historických fondů, pokud si stanovíme za cíl nejen evidenci digitálních kopií historických dokumentů, ale také jejich zpřístupnění. Úkolem resolveru je tedy

umožnit uživateli přístup ke každé nalezené digitální kopii každého dokumentu zapsanému v RDHF.

Předpokladem pro správné fungování Registru digitalizace historických fondů, tak jak byl navržen, je existence základních popisných a strukturálních metadat k digitálním kopiím. Důležitá je pak také možnost využití technických metadat o digitalizaci a digitalizačních pracovištích.

3.2 Konceptuální model systému RDHF

Na základě předchozího rozboru byl stanoven takový koncept, že Registr digitalizace historických fondů bude sestávat ze tří relativně samostatných funkčních celků, které jsou ovšem spolu úzce provázány. Základním a nejdůležitějším celkem je vlastní databáze digitálních kopií fyzických historických dokumentů. Ta má za účel udržovat záznamy o všech digitálních kopiích dokumentů. Pro udržení persistentní identifikace fyzických dokumentů a tím také persistentní identifikace jejich digitálních kopií slouží tabulka konkordancí. Ta je vztažena k záznamům o digitálních kopiích obsaženým v hlavní databázi. Poslední částí RDHF je resolver, sloužící k zpřístupnění (zobrazení) vlastní digitální kopie dokumentu.



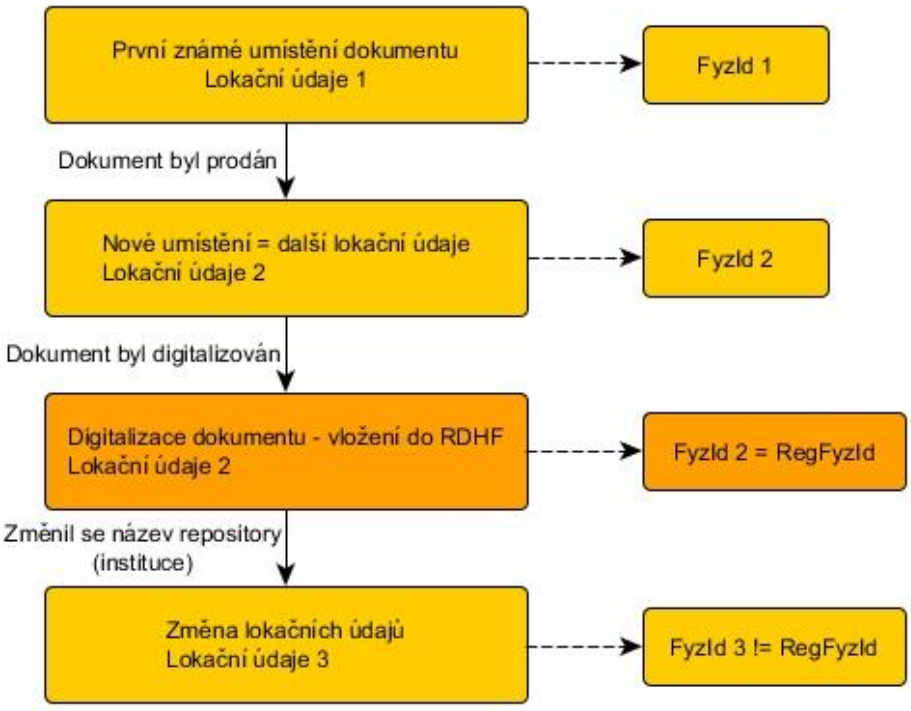
Obr. 1 Konceptuální schéma Registru digitalizace historických fondů

3.2.1 Databáze digitálních kopií fyzických historických dokumentů

Databáze digitálních kopií fyzických dokumentů soustřeďuje veškeré dostupné informace o jednotlivých digitálních kopiích všech známých digitalizovaných exemplářů historických dokumentů. Tato databáze vzniká jednak automatizovaně importem záznamů z Manuscriptoria a dalších dostupných zdrojů, jednak ručním vkládáním záznamů správci databáze. Správci mají kromě vkládání záznamů o digitálních kopiích historických dokumentů také možnost jim svěřené záznamy opravovat či mazat. Databáze je organizována tak, že každé jedné registrované digitální kopii dokumentu přísluší jeden záznam, přičemž pro jeden fyzický dokument může existovat více digitálních kopií a tedy i více záznamů v databázi.

3.2.2 Digitální konkordance

Tato součást systému RDHF soustřeďuje veškeré známé informace o změnách lokačních údajů historických dokumentů. Lokační údaje hrají zásadní roli při automatizované tvorbě jednoznačných identifikátorů fyzických dokumentů FyzId. Změna umístění a tím i lokačních údajů dokumentu prakticky znemožňuje jeho persistentní identifikaci, což názorně ilustruje Obr. 2. Jako perzistentní identifikátor (RegFyzId) digitalizovaného dokumentu se proto bere identifikátor, který byl vytvořen z lokačních údajů fyzického dokumentu platných právě v okamžiku zápisu záznamu o jeho první digitální kopii do databáze RDHF.



Obr. 2 Příklad změny FyzId při změně lokačních údajů dokumentu

V tabulce digitálních konkordancí jsou proto uvedeny všechny známé změny v umístění **digitalizovaného** exempláře (jehož záznam existuje v RDHF) – tedy jeho lokační údaje, ať vznikly před vložením záznamu do RDHF nebo až po něm.

Záznamy v tabulce konkordancí pro daný dokument

	Persistentní identifikátor	Identifikátor odpovídající lokačním údajům
1	RegFyzId (= FyzId 2)	FyzId 1
2	RegFyzId (= FyzId 2)	FyzId 3

Obr. 3 Záznam konkordancí pro modelový příklad

Z principu tohoto přístupu tedy jasně vyplývá, že tabulka konkordancí obsahuje pouze informace o změnách v umístění exemplářů, tedy o změnách jejich lokačních údajů v čase. Podrobněji je mechanismus digitálních konkordancí popsán v dokumentu [1], kapitola 4.2. Na Obr. 3 jsou znázorněny zápisy v tabulce konkordancí odpovídající modelovému příkladu na Obr. 2.

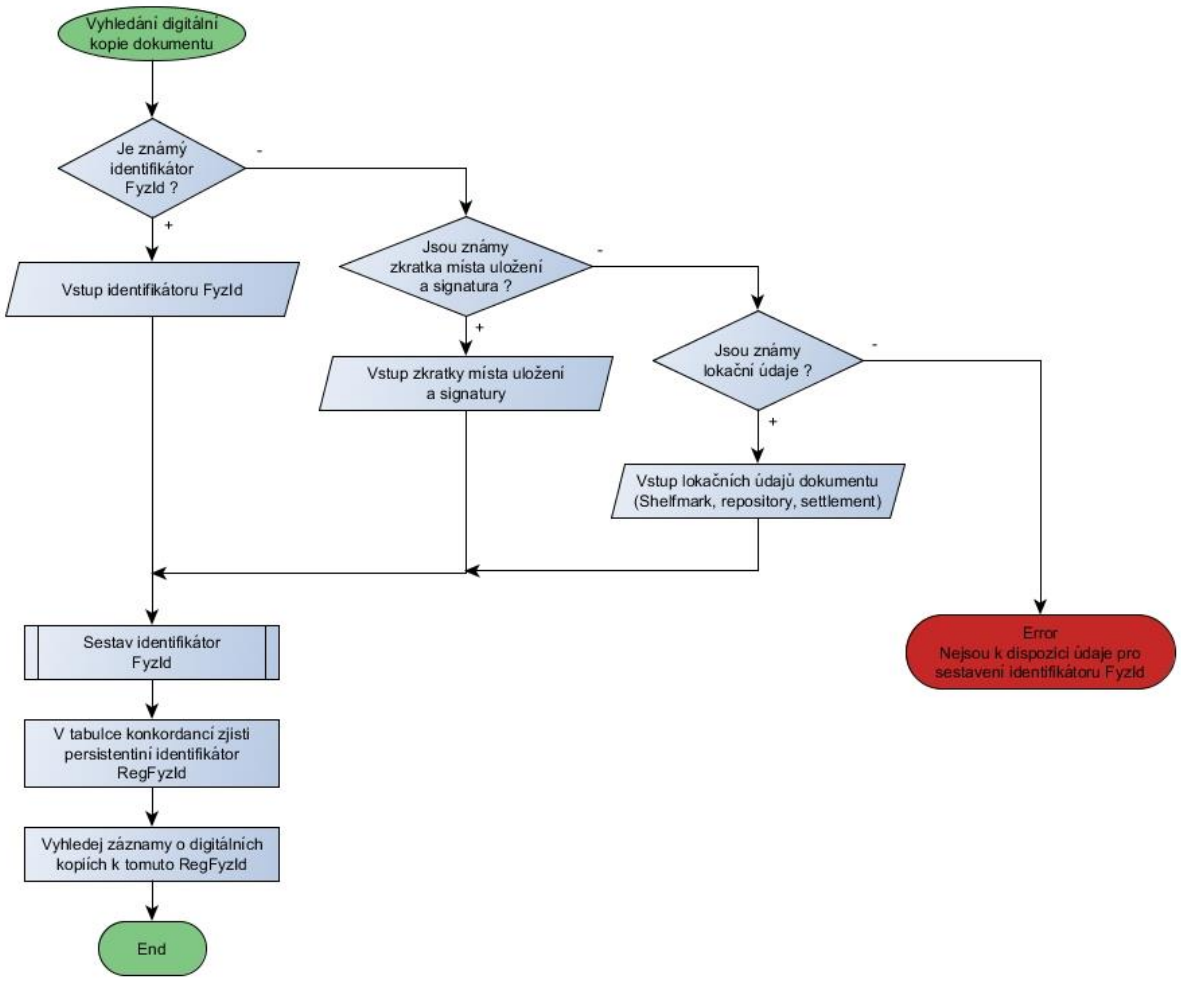
Tabulka konkordancí hraje v RDHF velmi důležitou roli při zjišťování persistentního identifikátoru fyzického dokumentu ze známých lokačních údajů dokumentu platných v nějakém čase. Jak již bylo uvedeno, persistentním identifikátorem RegFyzId se stává identifikátor FyzId v okamžiku vstupu prvního záznamu o jeho digitální kopii do RDHF.

Jako příklad uveďme hledání digitální kopie určitého dokumentu: Nejprve se uživateli známé lokační údaje (nemusí odpovídat právě platným lokačním údajům) použijí k vytvoření „pracovního" identifikátoru FyzId. Tento identifikátor potom bude sloužit jako klíč k prohledání tabulky konkordancí. V případě, že byl v tabulce nalezen záznam o změně s tímto klíčem, použije se RegFyzId uvedené v tomto záznamu, v případě, že nebyla nalezena změna lokačních údajů, jako RegFyzID se použije „pracovní" FyzId. Pomocí takto zjištěného RegFyzId se potom v databázi digitálních kopií vyhledají všechny digitální kopie příslušného fyzického dokumentu.

3.2.3 Resolver

Problematika resolveru pro digitální kopie historických dokumentů je podrobně popsána v dokumentu [2]. Úkolem resolveru obsaženého v RDHF je zpřístupnit uživateli digitální kopii(e) daného fyzického exempláře historického dokumentu na základě vložení jeho známých lokačních údajů jako je signatura, repository a settlement. K vyhledání digitální kopie je použit mechanismus zjištění persistentního identifikátoru její fyzické předlohy prostřednictvím konkordanční tabulky, jak bylo

popsáno v předchozí části. Algoritmus zpřístupnění vyhledání digitální kopie je podrobně popsán v kapitole 4 dokumentu [2] a pro ilustraci je uveden na Obr. 4.



Obr. 4 Postup vyhledání digitálních kopií k historickému dokumentu podle [2]

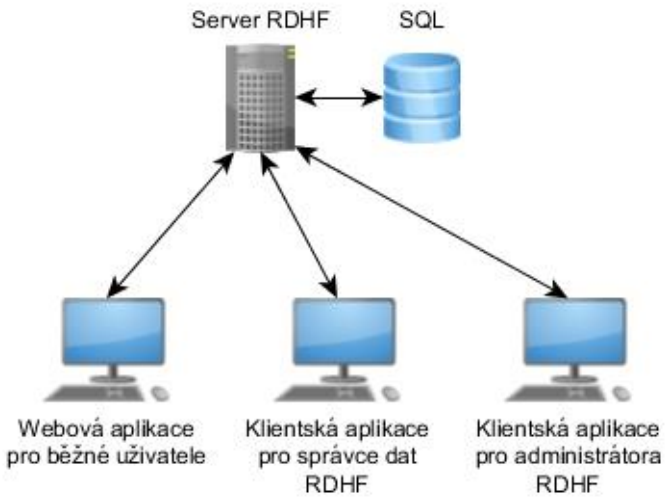
4 Architektura systému RDHF

4.1 Server a klientské aplikace

Pro realizaci systému „Registr digitalizace historických fondů" byla zvolena architektura klient/server, která umožňuje oddělit zabezpečení všech funkčních požadavků na systém RDHF od vlastního uživatelského rozhraní.

Serverová část zajišťuje veškeré služby Registru digitalizace historických fondů pro všechny typy jeho uživatelů tak, jak byly popsány výše. Pro ukládání a správu dat RDHF využívá relační databázový systém SQL. Vzhledem k jednoduchosti databázové architektury se zde nabízí použití některého z open sourcových projektů.

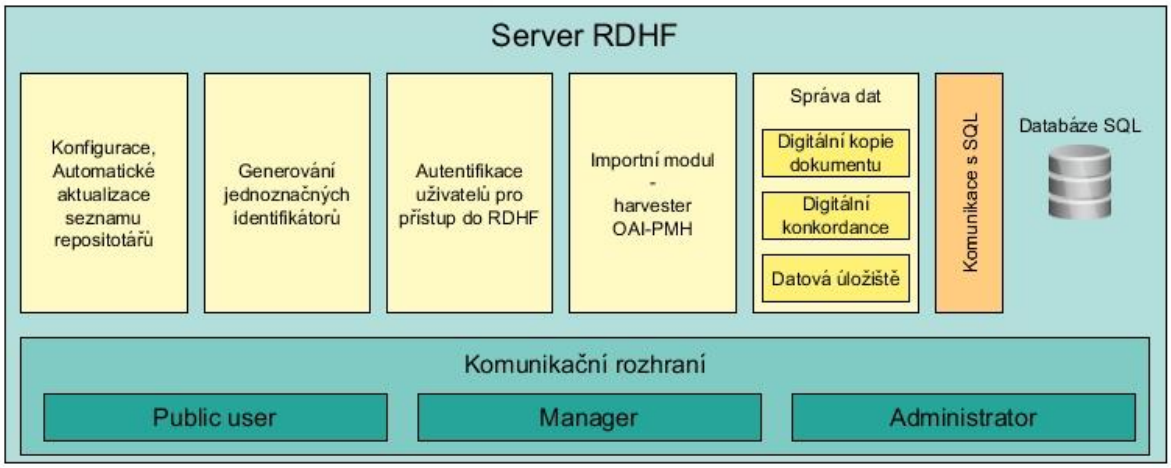
Jednotlivé klientské aplikace zpřístupňují služby serveru uživatelům. Aplikace jsou řešeny buď jako webové nebo nativní. Webová aplikace je určena pro obecné využívání Registru digitalizace historických fondů odbornou veřejností. Uživatelská rozhraní správců dat RDHF a administrátorů systému zajišťují nativní aplikace, které poskytují uživateli větší komfort při práci s daty RDHF. Všechny uživatelské aplikace komunikují se serverem prostřednictvím protokolu HTTP s architekturou REST. Komunikace mezi klientem a serverem je tedy bezestavová, jako formát výměny dat byl zvolen formát JSON.



Obr. 5 Základní schéma RDHF

4.2 Architektura serveru

Serverová část RDHF je tvořena několika základními funkčními celky, které zajišťují kompletní funkcionalitu systému. Vlastní architektura serveru RDHF je naznačena na Obr. 6.



Obr. 6 Schéma serverové části systému

4.2.1 Správa dat a komunikace s SQL databází

Tento modul zajišťuje veškerou komunikaci s databází, je zodpovědný za tvorbu příslušných SQL dotazů do tabulek databáze a převzetí výsledků. Zpracovává data digitálních kopií dokumentů, digitálních konkordancí i datových úložišť. Protože databázové systémy od různých výrobců mají odlišnosti v syntaxi i bohatosti jazyka SQL, bylo rozhraní modulu navrženo tak, aby v případě potřeby bylo možné snadno přejít na jiný databázový systém pouhou výměnou části tohoto modulu.

4.2.2 Importní modul

Importní modul má na starosti automatizované vkládání metadatových záznamů o digitálních kopiích fyzických historických dokumentů z definovaných zdrojů dat do systému. Obecně je možný import ze všech zdrojů, které jsou schopné poskytovat požadované informace o digitálních kopiích prostřednictvím standardního vytěžovacího protokolu OAI-PMH (Open Archives Initiative - Protocol for Metadata Harvesting), podmínkou je ale existence lokačních údajů o fyzických dokumentech v importovaných metadatových záznamech. Pro pilotní provoz byla realizována implementace pro zpracování záznamů ze systému Manuscriptorium. Importní modul kromě počátečního naplnění RDHF daty z požadovaných zdrojů zajišťuje také jejich aktualizaci. Ta může být automatizovaná v určitých časových intervalech nebo na vyžádání obsluhou – administrátorem systému.

4.2.3 Uživatelé systému RDHF

Obecné uživatelské aplikace či spolupracující knihovnické systémy využívají služeb Registru digitalizace historických fondů anonymně, bez nutnosti jakéhokoliv přihlašování či registrace. Naproti tomu aplikace, které umožňují uživateli nějakým

způsobem manipulovat s daty RDHF (vkládat, upravovat či mazat záznamy) vyžadují přihlášení uživatele, aby bylo možno zjistit jeho oprávnění k požadovaným činnostem nad určitými daty (digitální kopie, konkordance, …). Z tohoto hlediska systém rozlišuje tři typy uživatelů:

- **Veřejný uživatel** – klientská aplikace nebo kooperující systém, který využívá služeb RDHF, ale nemá žádnou možnost manipulace s daty. Využívá k volání služeb serveru rozhraní *Search*, které nepožaduje autentifikaci uživatele.
- **Správce** – klient, který má oprávnění nějakým způsobem nakládat s daty různých součástí RDHF, jako jsou správa digitálních kopií dokumentů, správa digitálních konkordancí, datových úložišť apod. Tyto klientské aplikace pro komunikaci se serverem využívají rozhraní *Manage*. Toto rozhraní již vyžaduje pro provedení požadavků klienta autentifikaci uživatele, proto požadavky obsahují povinný parametr UserId.
- **Administrátor** – uživatel oprávněný k manipulaci s daty na úrovni jejich zálohování, je zodpovědný za údržbu databáze, databázového systému, a za údržbu serveru. Zajišťuje také import metadat o digitálních kopiích dokumentů z definovaných zdrojů.

4.2.4 Autentifikace uživatelů

Systém RDHF nemá vlastní správu uživatelů. Za přihlášení do systému je zodpovědná klientská aplikace, která zajistí přihlášení a ověření uživatele v externím modulu pro správu uživatelů, který využívá mimo jiné i Manuscriptorium, a odtud také získá id a práva uživatele v systému RDHF. Při volání požadavku na službu serveru (requestu) klient vytvoří spolu s kontrolním kódem, který obdrží od autentifikačního modulu serveru RDHF, identifikátor uživatele. Ten potom pošle na server jako jeden z parametrů requestu. Autentifikační modul serveru jednak generuje kontrolní kód pro sestavení platného identifikátoru uživatele klientem a dále kontroluje platnost identifikátoru uživatele, předaného jako parametr příchozího požadavku, a oprávnění uživatele k provedení požadované operace.

4.2.5 Generování identifikátoru fyzického dokumentu a digitální kopie

Systém pro generování identifikátorů fyzických dokumentů je stěžejním modulem serveru RDHF. Od identifikátorů fyzických dokumentů se odvíjí tvorba identifikátorů digitálních kopií dokumentů, vygenerované identifikátory také slouží prostřednictvím digitálních konkordancí k nalezení perzistentních identifikátorů dokumentů v RDHF. Modul pro svou práci využívá databázi míst uložení dokumentů, která obsahuje jednoznačné zkratky (identifikátory) všech těchto míst, ve kterých jsou umístěny fyzické dokumenty, k nimž existují digitální kopie registrované v RDHF. Systém s pomocí vložených lokačních údajů v metadatech (např. repository a settlement) zjistí jednoznačnou zkratku místa uložení fyzického dokumentu a spolu se známou signaturou potom vytvoří jeho jednoznačný identifikátor FyzId. Principy

tvorby identifikátoru jsou popsány např. v části „Názvové konvence" v dokumentu [3], přesný algoritmus tvorby identifikátoru fyzického dokumentu je popsán dále v kapitole 5.3.

4.2.6 Automatické aktualizace informací pro tvorbu identifikátorů a konfigurace

Jak bylo uvedeno výše, pro tvorbu jednoznačných identifikátorů fyzických dokumentů je nezbytná existence databáze jednoznačných zkratek míst uložení fyzických dokumentů. Tato databáze je v současnosti realizována datovým souborem ve formátu XML a speciálními aplikacemi pro jeho správu. Soubor se jmenuje knihovny.xml. Protože tato databáze je spravována naprosto odděleně a nezávisle na systému RDHF (využívá se např. v projektech VISK 6 nebo Manuscriptorium), server RDHF stejně jako ostatní systémy používá pracovní kopii tohoto XML souboru. Pro správnou funkci modulu generování FyzId v RDHF je nutno zajistit pravidelnou aktualizaci pracovní kopie souboru knihovny. xml. Ta se provádí pravidelně s daným intervalem nebo na vyžádání administrátorem systému. Pro pilotní provoz byl zvolen interval aktualizace jedna hodina. Aktualizaci pracovní kopie zajišťuje modul pro konfiguraci a automatické aktualizace. Modul také umožňuje měnit některé vlastnosti systému (jako je například právě interval aktualizace souboru knihovny.xml) za běhu serveru bez nutnosti jeho restartu. Změny v konfiguraci serveru lze vzdáleně provádět z klientské aplikace administrátora.

4.3 Komunikační rozhraní

Komunikační rozhraní serveru zprostředkovává vlastní komunikaci mezi klienty a serverovou částí systému. Je rozděleno na tři části, kde každá komunikuje s jiným typem klienta. Pro veřejné uživatele je k dispozici rozhraní *Search*, pro správce rozhraní *Manage* a administrátorská aplikace využívá rozhraní *Admin*. Podrobně jsou jednotlivé části rozhraní popsány v samostatném dokumentu.

4.4 Klientské aplikace

Klientské aplikace slouží ke komunikaci uživatelů nebo jiných spolupracujících systémů se serverem RDHF. Jedná se o aplikace určené pro veřejné uživatele, správce dat a administrátory. Každá klientská aplikace bude podrobně popsána v samostatném dokumentu.

4.4.1 Webová aplikace pro běžné uživatele

Webová aplikace je určena pro využívání služeb RDHF odbornou veřejností. Je realizována technologií HTML5/jQuery. Podrobněji bude popsána v samostatném dokumentu.

4.4.2 Aplikace pro správce dat

Klientská aplikace pro správu dat umožňuje pověřeným uživatelům vytvářet a spravovat záznamy v tabulkách RDHF. Tito uživatelé spravují data o digitálních kopiích a digitálních konkordancích. Zároveň mají také k dispozici služby veřejného rozhraní.

Aplikace je napsána jako nativní ve vývojovém prostředí Delphi 10.1 a primárně je kompilována pro operační systém Windows 10. Díky flexibilitě a komplexnosti vývojového prostředí lze tuto aktualizaci v případě potřeby převést také pro operační systém Mac OSX nebo pro mobilní platformy Android a iOS. Aplikace bude podrobněji popsána v samostatném dokumentu.

4.4.3 Aplikace pro administrátora systému RDHF

Klientská aplikace určená pro správu systému RDHF administrátorem byla v první fázi také realizována jako spustitelná aplikace pro OS Windows a rovněž byla vyvinuta ve vývojovém prostředí Delphi 10.1. Aplikace umožňuje administrátorovi systému provádět kromě aktualizace pracovního souboru knihovny.xml a konfiguračního souboru serveru také správu datových úložišť a import metadat digitálních kopií. Lze také provést odpojení serveru například z důvodu údržby (zálohování či obnovy dat) nebo při probíhajícím importu metadat. Podrobně je aplikace administrátora popsána v samostatném dokumentu.

5 Technické řešení serverové části

5.1 Použité technologie

Serverová část systému RDHF je řešena jako Java servlet. V rámci pilotního řešení je servlet provozován na open source servlet kontejneru Apache Tomcat v. 8.0.

Servlet byl napsán v jazyce Java v.1.8 ve vývojovém prostředí Eclipse. Pro realizaci projektu byla použita řada open source technologií:

Rozhraní REST – technologie Jersey - https://jersey.java.net/

Harvester OAI-PMH – OCLC Research - http://www.oclc.org/research/themes/data-science/oaicat.html

Práce s dokumenty XML – technologie JAXB - http://www.oracle.com/technetwork/articles/javase/index-140168.html

Logování – Appache logging services (log4j) - http://logging.apache.org/log4j/2.x/

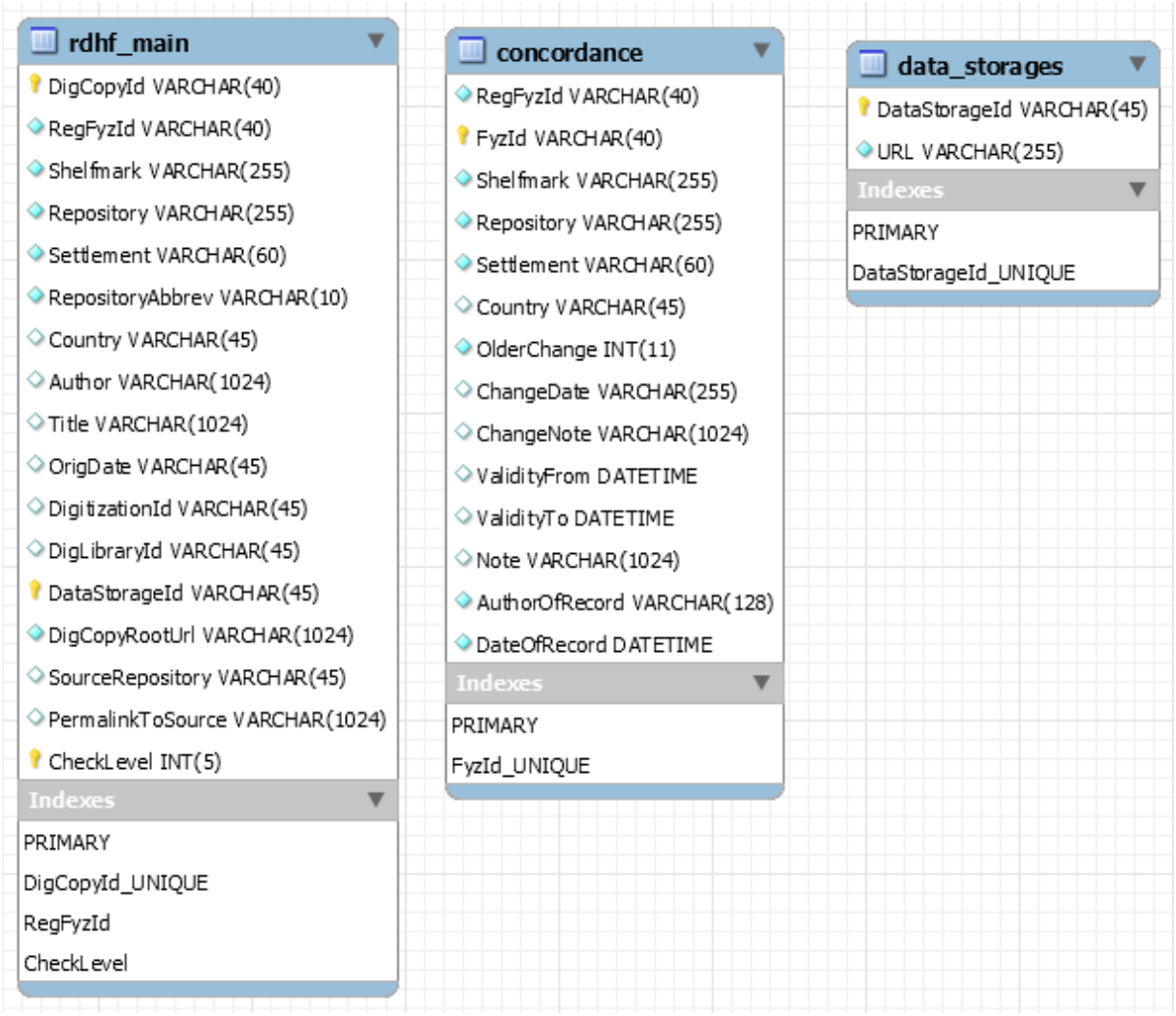
FTP klient a utility – Apache Commons - https://commons.apache.org/

Jako relační databázový systém byl použit MySQL (http://www.mysql.com/), se kterým RDHF server komunikuje prostřednictvím standardního rozhraní JDBC. S menšími úpravami lze v případě potřeby použít libovolný jiný open source nebo komerční SQL databázový systém.

Komunikace klienta se severem je výhradně asynchronní. Doba čekání na odezvu serveru se nastavuje v konfiguraci klienta, u webové aplikace je daná nastavením browseru. Webová aplikace může se serverem komunikovat přes rozhraní *Search* metodami GET a POST, aplikace pro správce a administrátora komunikují prostřednictvím rozhraní *Manage* a *Admin* pouze metodou POST.

5.2 Datový model

Databáze systému RDHF je velmi jednoduchá. Sestává se ze tří samostatných tabulek. První tabulka s názvem *rdhf_main* udržuje informace o všech digitálních kopiích historických dokumentů, které jsou registrovány v RDHF. Jak již název napovídá, je to hlavní a nejdůležitější tabulka, nezbytná pro funkci všech částí systému. Tabulka *concordance* udržuje změnové záznamy, vypovídající o změnách lokačních údajů fyzických dokumentů. Jak již bylo zmíněno, tyto údaje jsou nezbytné k identifikaci fyzického dokumentu, nemusí být ale trvalé v čase. Konečně tabulka *data_storages* udržuje informace o datových úložištích, na kterých jsou digitální kopie dokumentů uloženy. Datový model RDHF je zachycen na Obr. 7.



Obr. 7 Datový model RDHF

5.2.1 Tabulka rdhf_main

Tato tabulka soustřeďuje informace o všech digitálních kopiích dokumentů v RDHF. Jeden záznam v tabulce obsahuje informaci o jedné digitální kopii fyzického dokumentu a je tvořen těmito poli:

- **DigCopyId** – jednoznačný perzistentní identifikátor digitální kopie dokumentu – je zároveň unikátním primárním klíčem do tabulky
- **RegFyzId** – perzistentní identifikátor fyzického dokumentu, k němuž existuje tato digitální kopie. Identifikátor digitální kopie je odvozen právě od identifikátoru RegFyzId.
- **Shelfmark** – signatura fyzického dokumentu
- **Repository** – název místa uložení nebo název vlastníka fyzického dokumentu
- **Settlement** – název místa Repository (zpravidla město)

- **RepositoryAbbrev** – jednoznačná zkratka místa uložení dokumentu – zjišťuje se z Repository a Settlement v souboru knihovny.xml a jednoznačně identifikuje místo (příp. vlastníka), ve kterém je fyzický dokument umístěn
- **Country** – země, ve které je fyzický dokument umístěn.
- **Author** – autor (nebo autoři) dokumentu, pokud jsou známi
- **Title** – název (nebo názvy) dokumentu, pokud jsou známy
- **OrigDate** – datace vzniku dokumentu (pokud je známa)
- **DigitizationId** – identifikátor digitalizačního pracoviště – v této fázi projektu není k dispozici
- **DigLibraryId** – identifikátor digitální knihovny – v této fázi projektu není k dispozici
- **DataStorageId** – identifikátor datového úložiště, na kterém je digitální kopie dokumentu uložena
- **DigCopyRootUrl** – URL na root digitální kopie dokumentu
- **SourceRepository** – identifikátor zdroje, ze kterého byl záznam o digitální kopii převzat (Manuscriptorium)
- **PermalinkToSource** – permanentní link na zdrojový záznam
- **CheckLevel** – servisní informace – udává stav záznamu

Všechna pole této tabulky se vyplňují při importu ze zdrojového systému metadat nebo při ručním vkládání dat z klientského prostředí správce databáze digitálních kopií. Pole *CheckLevel* je servisní, udává například, zda byl záznam vložen ručně nebo automatizovaným importem, zda byla provedena jeho kontrola, zda je či není platný a podobně.

5.2.2 Tabulka concordance

Záznamy tabulky *concordance* obsahují informace o změnách umístění fyzických dokumentů a tedy jejich lokačních údajů v jejich metadatech. Tyto změny lokačních údajů fyzického dokumentu způsobují i změnu jeho identifikátoru *FyzId*, který se automatizovaně vytváří právě z těchto lokačních údajů. Záznamy v tabulce *concordance* potom zachycují vztah nově vytvořeného *FyzId* k perzistentnímu identifikátoru *RegFyzId*, který vznikl z lokačních údajů fyzického dokumentu při vstupu záznamu o jeho první digitální kopii do RDHF.

- **RegFyzId** – perzistentní identifikátor fyzického dokumentu, ke kterému se vztahuje změna lokačních údajů
- **FyzId** – identifikátor fyzického dokumentu vytvořený z nových lokačních údajů. Je zároveň unikátním primárním klíčem v tabulce *concordance*
- **Shelfmark** – nová signatura fyzického dokumentu
- **Repository** – nová repository (údaj v metadatech) fyzického dokumentu
- **Settlement** – nový settlement (údaj v metadatech) fyzického dokumentu
- **Country** – nová země umístění fyzického dokumentu
- **OlderChange** – příznak, že změna lokačních údajů dokumentu vznikla před vstupem záznamu o digitální kopii dokumentu do RDHF (ale do tabulky

konkordancí mohla být zapsána až PO vložení záznamu o digitální kopii do RDHF)

- **ChangeDate** – v případě že se jedná o „starou“ změnu lokačních údajů, doba (alespoň přibližná), kdy ke změně došlo.
- **ChangeNote** – doplňující poznámka ke „starší“ změně lokačních údajů
- **ValidityFrom** – doba, od které změna lokačních údajů platí
- **ValidityTo** – doba, do kdy změna lokačních údajů platí
- **Note** – poznámka k záznamu
- **AuthorOfRecord** – identifikátor autora záznamu nebo jeho poslední změny
- **DateOfRecord** – datum a čas vložení nebo poslední změny záznamu

5.2.3 Tabulka data_storages

Tabulka *data_storages* udržuje všechny známé informace o datových úložištích, na kterých jsou umístěny digitální kopie dokumentů registrované v RDHF. V současné době nejsou pro datová úložiště k dispozici žádné podrobnější údaje a jsou prakticky známy pouze UR adresy rootů těchto úložišť. Tabulka datových úložišť je proto velmi jednoduchá a obsahuje pouze dvě pole:

- **DataStorageId** – identifikátor datového úložiště, je unikátním primárním klíčem do tabulky *data_storages*
- **URL** – URL datového úložiště

5.3 Jednoznačný identifikátor fyzického dokumentu

Jednoznačný identifikátor fyzického dokumentu se sestává ze tří částí. Jsou to: Jednoznačná zkratka místa uložení fyzického dokumentu, normalizovaná signatura a kontrolní znaková sekvence. Normalizovaná signatura je obecně vytvořena z běžně používaného jednoznačného identifikátoru fyzického dokumentu (zpravidla signatury). Podmínkou při tom je, že tento identifikátor je jedinečný v rozsahu daného místa uložení dokumentu a zároveň je také uveden v popisných metadatech jako identifikátor fyzického dokumentu (například ve formátu TEI P5 je to obsah elementu <idno>).

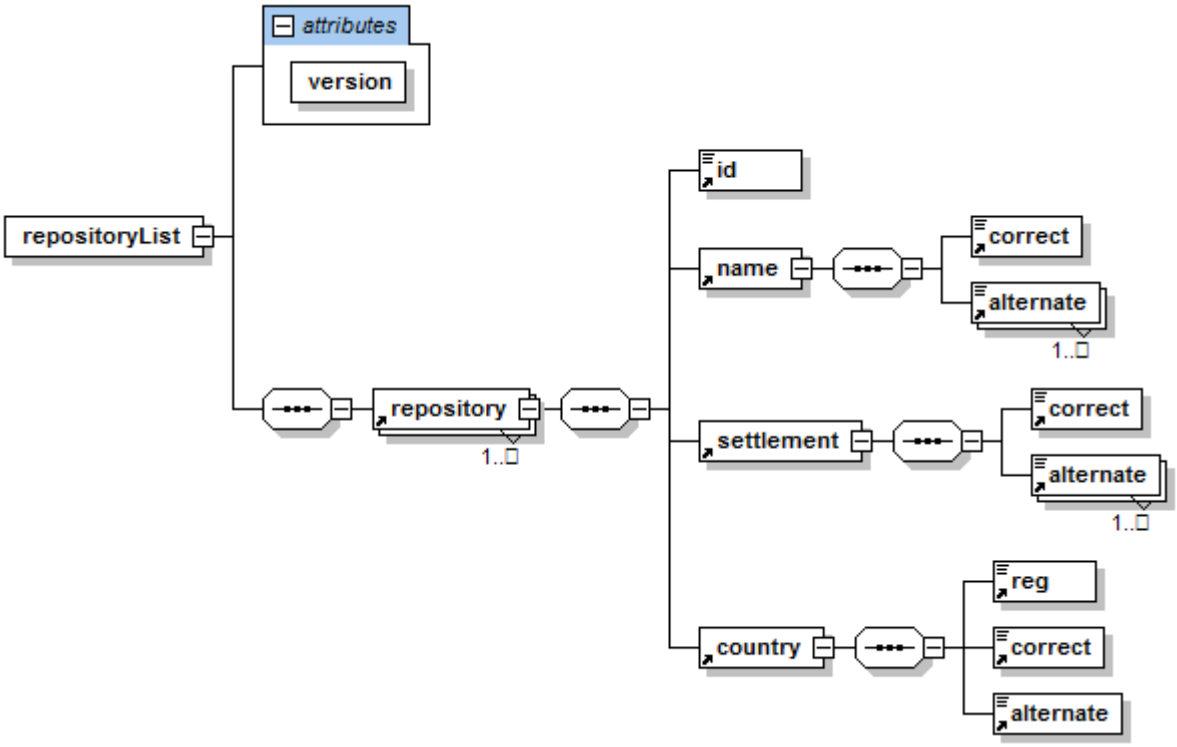
5.3.1 Identifikátor místa uložení fyzického dokumentu

V metadatových záznamech spojených s fyzickými dokumenty jsou názvy uložení fyzických dokumentů uvedeny mnoha různými způsoby i s mnoha chybami a „překlepy“. Automatizovaně identifikovat místo uložení fyzického dokumentu z těchto informací je tak v podstatě nemožné. Pro ilustraci „rozmanitosti“ zápisu informací o umístění fyzických dokumentů v jejich popisných metadatech je v příloze A uveden záznam z databáze míst uložení popisující Národní knihovnu České republiky se všemi

jejími alternativními názvy zjištěnými v dostupných metadatových záznamech (elementy <repository> a <settlement> v TEI P5).

Z nejednotné formy zápisu těchto lokačních údajů i díky mnoha chybám při jejich zápisu v metadatových záznamech k fyzickým dokumentům vyplynula nutnost jednoznačné identifikace míst uložení a vytvoření jejich jednoznačných identifikátorů. Ty jsou zásadní pro tvorbu jednoznačného identifikátoru fyzického dokumentu FyzId, jehož jsou nedílnou součástí (jak je popsáno v [3]). Tato zkratka je zároveň jednoznačným identifikátorem místa uložení fyzického dokumentu v databázi míst uložení.

Současnou formu databáze míst uložení, jejich alternativních názvů a jednoznačných zkratek (identifikátorů) reprezentuje XML soubor knihovny.xml. Grafické znázornění definice XML schématu (XSD) tohoto souboru je uvedena na Obr. 8. Ze schématu je zřejmé, že ke každému místu uložení (element <repository>), reprezentovanému jeho zkratkou (element <id>) existuje korektní (oficiální) název a k němu řada alternativních názvů, tak jak byly uvedeny v popisných metadatech dokumentů. Tento název je uveden v elementu <name>. Další část názvu místa uložení (adresa, město) je obsažena v elementu <settlement>. I v něm je kromě jeho správného názvu uvedena řada alternativních názvů. Z informace v polích <name> a <settlement> se potom v databázi zjišťuje identifikátor místa uložení, kterým je 6-tipísmenná zkratka, umístěná v elementu <id>. Ke každému názvu místa uložení je ještě volitelně uvedena země uložení dokumentu (element <country>) také s oficiálním a alternativními názvy. V elementu <reg> je uveden kód státu podle ISO 3166.



Obr. 8 Definice XML schématu souboru knihovny.xml

Jednoznačný identifikátor místa uložení smí obsahovat znaky:

- Velká písmena bez diakritiky "A" až "Z" (`0x41..0x5A`)
- Číslice "0" až "9" (`0x30..0x39`)
- Znak podtržítko "_" (`0x5F`)

Identifikátor má vždy šest znaků. Pokud je zkratka umístění kratší (musí mít nejméně dva znaky), je doplněna zprava na tuto délku oddělovacími znaky "_" (0x5F).

5.3.2 Sestavení jednoznačného identifikátoru fyzického dokumentu (FyzId)

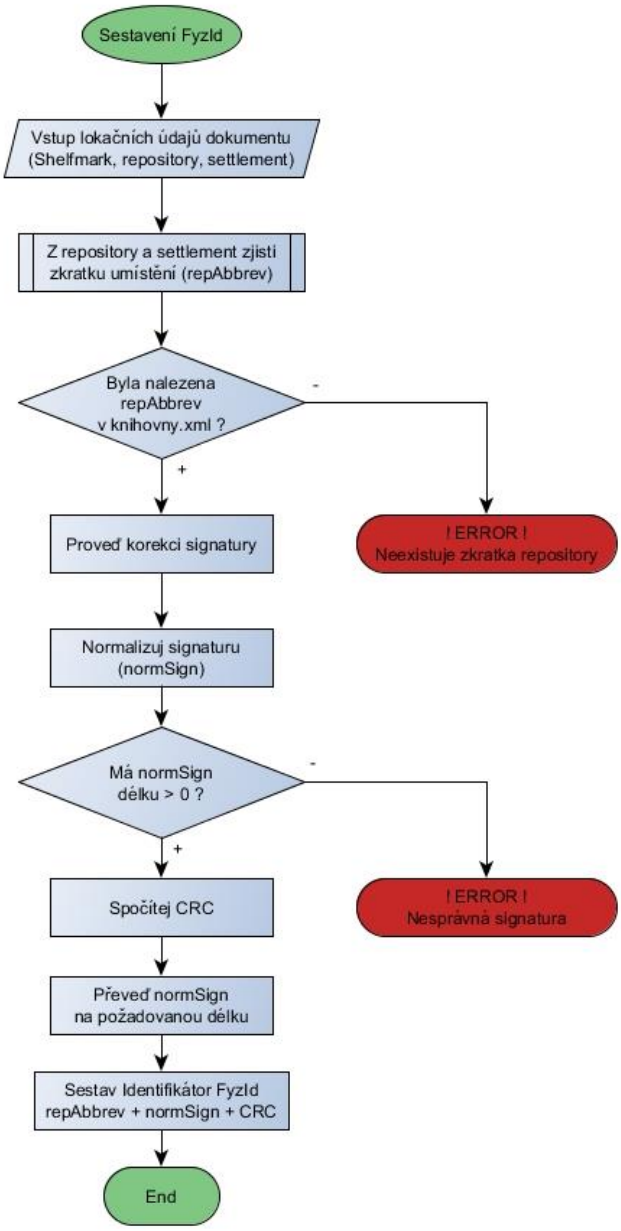
Stejně jako identifikátor místa uložení tak i identifikátor fyzického dokumentu FyzId smí v souladu s definicí v [3] obsahovat pouze tyto znaky:

- Velká písmena bez diakritiky "A" až "Z" (`0x41..0x5A`)
- Číslice "0" až "9" (`0x30..0x39`)
- Znak podtržítko "_" (`0x5F`)

Sestavení identifikátoru fyzického dokumentu FyzId probíhá podle algoritmu uvedeného na Obr. 9.

Nejprve se z vložených údajů (repository a settlement) zjistí unikátní zkratka místa uložení (repositoryAbbrev). V případě, že byla zkratka nalezena, pokračuje se v sestavování FyzId.

Dalším krokem je korekce signatury. Tato korekce souvisí se vstupní podobou signatury. Stává se například, že obsahuje poměrně dlouhý text, stejný pro několik dokumentů a v závěru je rozlišena pouze řeckými písmeny. Standardním algoritmem podle [3], který předpokládá pouze existenci znaků latinky, by došlo po normalizaci signatury ke vzniku několika identických identifikátorů FyzId pro různé dokumenty. Obecně se mohou vyskytovat signatury, přicházející do Manuscriptoria (a tím i do RDHF), které obsahují kromě latinky znaky v různých znakových sadách, přičemž vyjmutí těchto znaků při normalizaci signatury může mít za následek vznik neplatných, nejednoznačných a už vůbec ne unikátních identifikátorů FyzId. Proto je potřeba do postupu tvorby FyzId zařadit korekční člen, který bude reagovat na nepředvídané tvary signatur především cizokrajných dokumentů. Současná korekce spočívá pouze v nahrazení vybraných řeckých znaků, které byly identifikovány v signaturách, znakovými sekvencemi, např. α -> ALPHA, β -> BETA, γ -> GAMA, δ -> DELTA, ε -> EPS, ζ -> ZETA, η -> ETA. Pro signatury v jiných abecedách (cyrilice, azbuka) lze v rámci korekce provést např. transliteraci do latinky. Tato korekce samozřejmě není optimální. Proto se také v současné době intenzivně pracuje na vytvoření nových algoritmů, umožňujících převést libovolné znakové sady do tvaru vyžadovaného při vstupu signatury do procesu tvorby identifikátoru FyzId.



Obr. 9 Algoritmus sestavení identifikátoru FyzId

Po projití signatury korekčním mechanismem se provede její normalizace. Ta spočívá nejprve v odstranění diakritiky, potom v nahrazení znaků, které neodpovídají definici povolených znaků (jsou uvedeny na začátku této kapitoly) oddělovacím znakem (podtržítko) a převedením zbývajících povolených znaků na velká písmena. Dále se odstraní oddělovací znaky ze začátku a konce signatury a odstraní se vícenásobné výskyty oddělovacích znaků. Pokud je délka takto normalizované signatury nulová, znamená to, že signatura obsahuje pouze nekorigované nepřípustné znaky.

Z šestimístného identifikátoru místa uložení a normalizované signatury (která může být a často také je mnohem delší než definovaných dvanáct znaků) se vypočítá

kód CRC. To je textový řetězec složený ze sedmi znaků vytvořených pomocí algoritmu CRC32. Jeho smyslem je spolehlivé rozlišení zkrácených normalizovaných signatur. Nejprve se algoritmem CRC32 vypočte číslo z textu, složeného z šestimístné zkratky místa uložení dokumentu a normalizované (ale ještě nezkrácené signatury). Z tohoto čísla (0 až 4294967295) sestavíme kontrolní řetězec tímto postupem:

1. Číslo se celočíselně dělí 16, zbytek po dělení se převede na znaky "0" až "9", "A" až "F". Tak vznikne první znak ZPRAVA
2. Výsledek z předchozího dělení se celočíselně dělí 36, zbytek po dělení se převede na znaky "0" až "9", "A" až "F", vznikne druhý znak ZPRAVA
3. Výsledek z předchozího dělení se celočíselně dělí 36, zbytek po dělení se převede na znaky "0" až "9", "A" až "F", vznikne třetí znak ZPRAVA
4. Výsledek z předchozího dělení se celočíselně dělí 36, zbytek po dělení se převede na znaky "0" až "9", "A" až "F", vznikne čtvrtý znak ZPRAVA
5. Výsledek z předchozího dělení se celočíselně dělí 36, zbytek po dělení se převede na znaky "0" až "9", "A" až "F", vznikne pátý znak ZPRAVA
6. Výsledek z předchozího dělení se celočíselně dělí 36, zbytek po dělení se převede na znaky "0" až "9", "A" až "F", vznikne šestý znak ZPRAVA
7. Výsledek z předchozího dělení se převede na znaky "0" až "4", vznikne sedmý znak

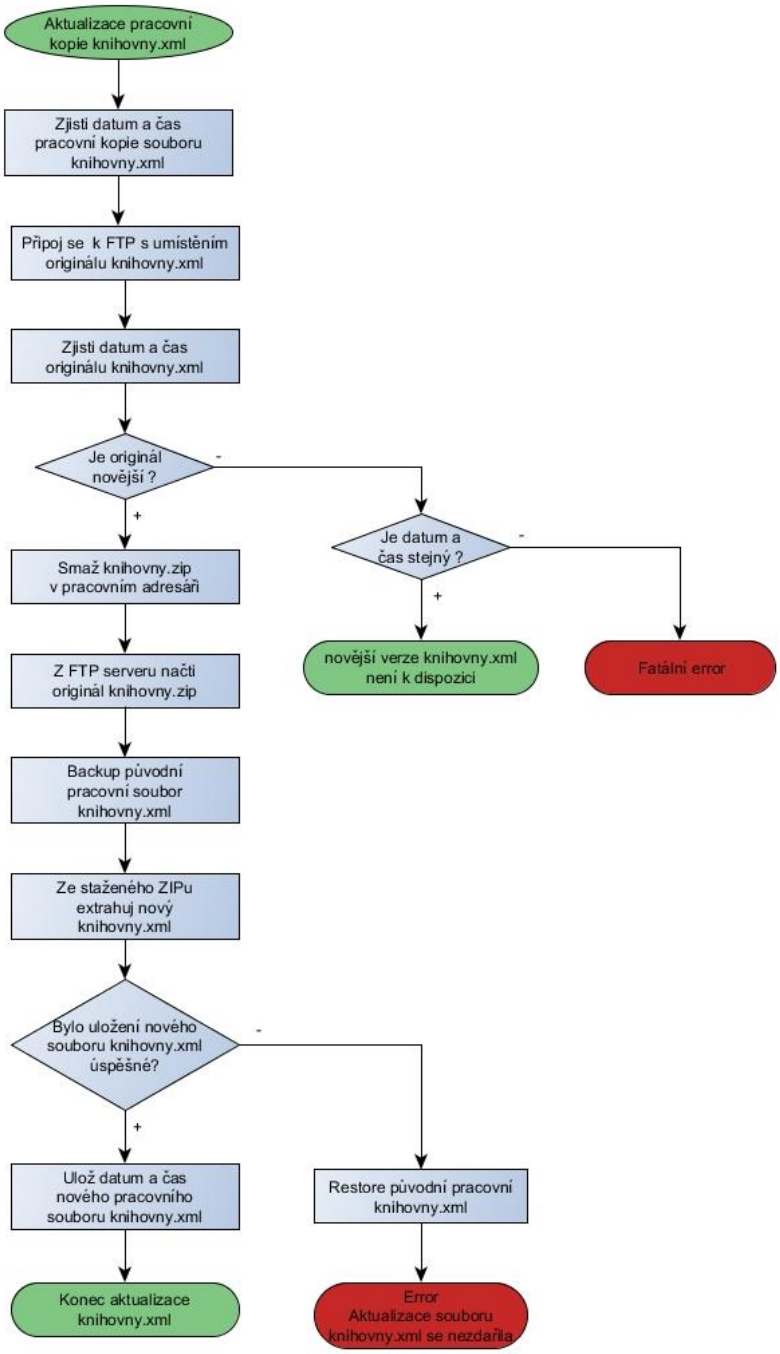
Nyní převedeme normalizovanou signaturu na řetězec o délce 12 znaků. Pokud je normalizovaná signatura delší než dvanáct znaků, nejprve se z ní odstraní všechny oddělovací znaky. Pokud je i pak delší, zkrátí se na dvanáct znaků. Pokud je naopak vzniklý text kratší než dvanáct znaků, doplní se zprava oddělovacími znaky (podtržítko).

Jednoznačný identifikátor fyzického dokumentu FyzId o celkové délce dvacet pět znaků vznikne spojením zkratky místa umístění (6 znaků), normalizované signatury (12 znaků) a řetězce CRC (7 znaků).

5.3.3 Synchronizace databáze místa uložení

Jak bylo uvedeno výše, pracovní kopii souboru knihovny.xml je třeba synchronizovat s jejím originálem v Manuscriptoriu. Toto probíhá jednak automaticky ve zvoleném časovém intervalu (v našem případě jedna hodina) nebo na vyžádání administrátorem systému RDHF. Originál souboru je dostupný prostřednictvím protokolu FTP v komprimované podobě metodou ZIP.

Po spuštění procesu aktualizace program nejprve zjistí datum a čas pracovní kopie souboru knihovny.xml. Poté se prostřednictvím FTP klienta připojí k FTP serveru, na kterém je k dispozici originální (aktuální) verze souboru a zjistí datum a čas této verze. Pokud jsou datum a čas originálu i pracovní kopie stejné, znamená to, že na serveru není k dispozici novější aktualizace souboru knihovny.xml. Pokud je snad aktuální verze na serveru starší, než její pracovní kopie, znamená tom že někde došlo k neočekávanému problému s verzemi souboru knihovny.xml a administrátor RDHF musí tuto situaci řešit společně se správcem databáze míst uložení historických dokumentů.



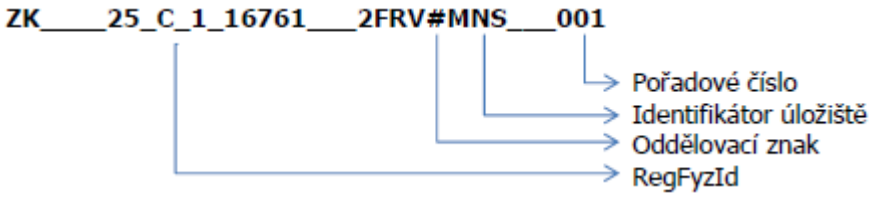
Obr. 10 Postup aktualizace souboru knihovny.xml

Pokud je na FTP serveru umístěn soubor knihovny.xml, který je novější, než je pracovní verze serveru RDHF, algoritmus již pokračuje standardně. V pracovním adresáři serveru RDHF smaže naposledy stažený komprimovaný soubor knihovny.zip. Poté z FTP serveru stáhne nový komprimovaný soubor. Před dalším krokem provede zálohování aktuální pracovní kopie souboru knihovny.xml. Poté se pokusí o extrahování nové aktuální verze souboru knihovny.xml do pracovního adresáře serveru RDHF. Pokud vše proběhlo v pořádku, v pracovním adresáři serveru RDHF je

nyní aktualizovaná pracovní verze souboru knihovny.xml. V případě, že došlo v průběhu stahování či extrahování k nějaké chybě, tj. nepodařilo se stáhnout nebo rozbalit soubor knihovny.zip, jako pracovní soubor se použije předchozí záloha souboru knihovny.xml.

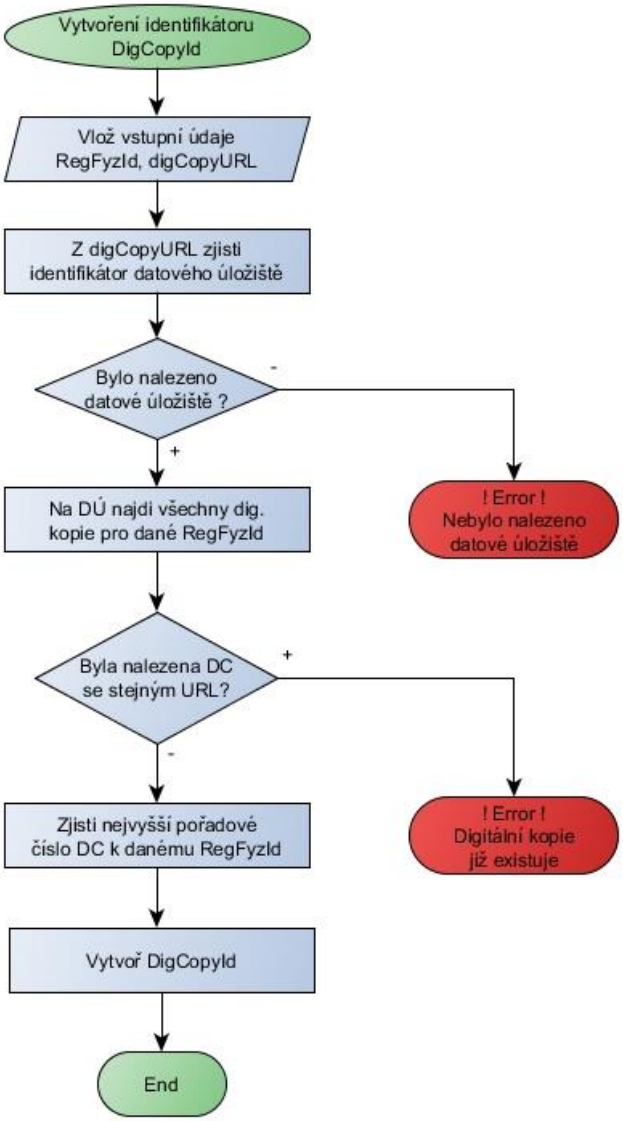
5.4 Identifikátor digitální kopie dokumentu

Jak je popsáno např. v kapitole 5.1 dokumentu [1], jednoznačný a perzistentní identifikátor digitální kopie fyzického dokumentu (DigCopyId) je vytvořen pomocí perzistentního identifikátoru tohoto fyzického dokumentu. Identifikátor digitální kopie je odvozen od persistentního identifikátoru předlohy, tj. fyzického dokumentu RegFyzId. Za oddělovacím znakem "#" (0x23) následuje šestipísmenná zkratka datového úložiště a za ní třímístné pořadové číslo digitální kopie na tomto datovém úložišti. Příklad struktury identifikátoru digitální kopie pro dokument s RegFyzId ZK____25_C_1_16761___2FRV je uveden na Obr. 11.



Obr. 11 Struktura identifikátoru DigCopyId

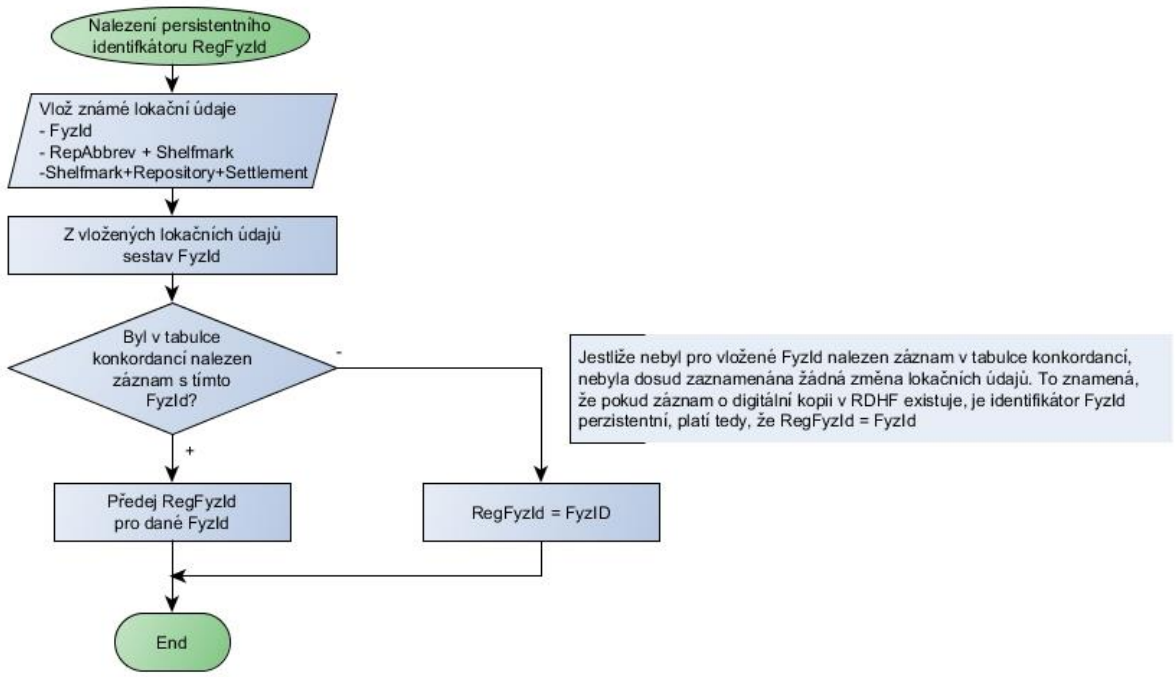
Aby se zamezilo vkládání více záznamů o téže digitální kopii do RDHF, byl použit pro vyváření identifikátoru DigCopyId algoritmus, který ze vstupních dat zjistí identifikátor datového úložiště, na kterém je digitální kopie dokumentu uložena. Poté ověří, zda na tomto datovém úložišti již není tato digitální kopie uložena. Při ověřování vychází z předpokladu, že pokud mají dvě digitální kopie ke stejnému fyzickému dokumentu na stejném datovém úložišti stejnou URL adresu, jedná se o tutéž digitální kopii. Není-li tento předpoklad splněn, jedná se o novou digitální kopii dokumentu a přistoupí se k vytvoření jejího identifikátoru DigCopyId. Ten je složen z identifikátoru datového úložiště, na kterém je digitální kopie uložena, a pořadového čísla, které bude o jednu vyšší, než je nejvyšší pořadové číslo již existující digitální kopie k danému fyzickému dokumentu.



Obr. 12 Princip vytvoření identifikátoru DigCopyId

5.5 Algoritmus pro vyhledání perzistentního identifikátoru fyzického dokumentu

Jednou z klíčových funkcí Registru digitalizace historických fondů je vyhledání digitální kopie konkrétního požadovaného fyzického dokumentu na základě známých, byť třeba v současné době již neplatných lokačních informací. Nezbytným krokem k vyhledání nalezení digitální kopie je nalezení persistentního identifikátoru její předlohy – fyzického dokumentu. Algoritmus je principiálně velice jednoduchý, jak je zřejmé z Obr. 13.

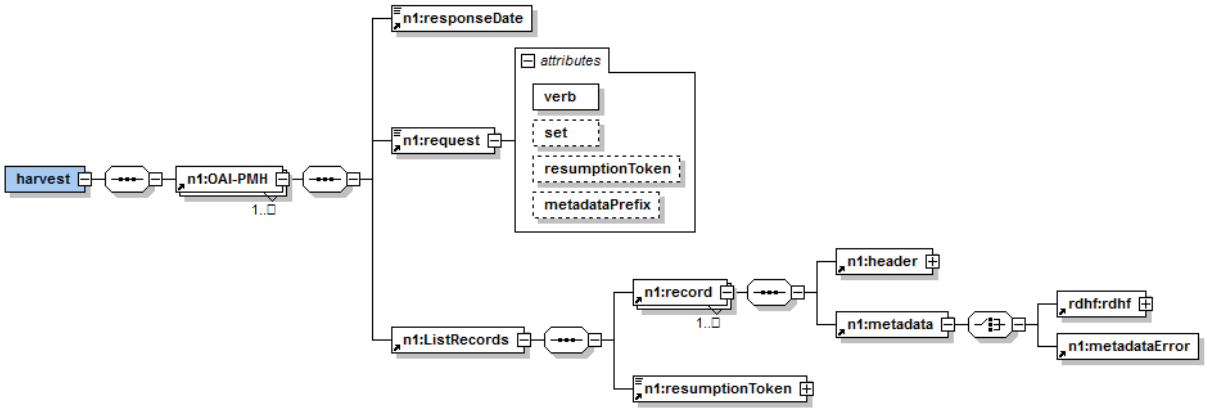


Obr. 13 Algoritmus určení persistentního identifikátoru RegFyzId

5.6 Import metadat do RDHF

Import metadat o digitálních kopiích dokumentů do systému RDHF probíhá prostřednictvím standardního rozhraní pro vytěžování metadat OAI-PMH. Protože rozhraní OAI-PMH je implementováno na protokolu http, lze přes něj velmi snadno vytěžovat zdroje, které svůj obsah poskytují prostřednictvím protokolu OAI-PMH. V terminologii OAI se zdroj metadat, poskytující svůj obsah prostřednictvím tohoto protokolu nazývá OAI-PMH data repository. Zařízení, které metadata z data repository „sklízí", se nazývá OAI-PMH harvester. Vlastní komunikační protokol je velmi jednoduchý, popisuje pouze šest základních operací, které jsou pro sklízení metadat nezbytné. Z data repository lze těžit jednotlivé záznamy požadavkem (verb v terminologii OAI) GetRecord nebo celý požadovaný zdroj či jeho část požadavkem ListRecords. Tento požadavek poskytuje metadatové záznamy v určitých předem specifikovaných kvantech. Pomocí požadavku ListIdentifiers lze také vytěžit seznam samotných identifikátorů metadatových záznamů, který může být na straně příjemce dat dále zpracován a později použit ke sklízení požadovaných metadat jednou ze dvou z výše uvedených metod. Protokol je podrobně popsán v [4].

Pro účely naplnění RDHF daty a jeho aktualizací byla použita metoda sklízení metadat příkazem ListRecords. Protože je OAI-PMH realizován na protokolu HTTP, je odezva na požadavek formátována v XML. Pro příklad je schéma odezvy pro požadavek ListRecords uvedeno na Obr. 14. Zde je také vidět umístění vytěženého záznamu, který se nachází v elementu <n1:metadata> (element <rdhf:rdhf>).

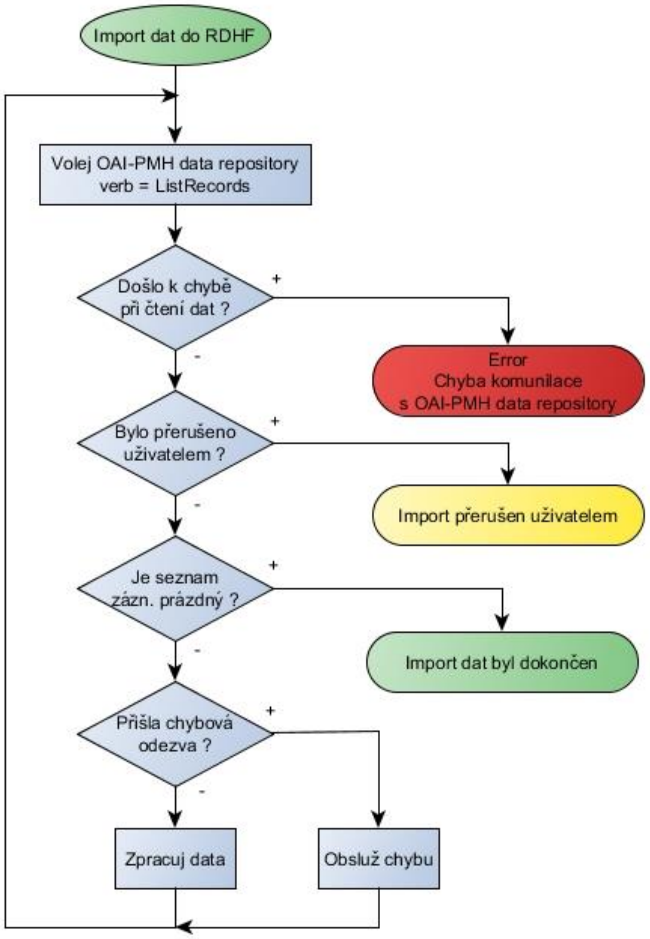


Obr. 14 Schéma standardní XML odezvy protokolu OAI-PMH

5.6.1 Proces sklízení metadat z OAI-PMH data repository

Import metadat běží na serveru RDHF ve vlastním vláknu jako samostatný podproces, který může být kdykoliv přerušen klientskou aplikací (uživatelem Administrátor). Po inicializaci modulu a připojení k databázi se zahájí vytěžování zvoleného zdroje metadat. Celý proces probíhá v cyklu, který začíná přečtením první dávky vstupních záznamů ze zdrojové OAI-PMH data repository a končí v okamžiku, kdy buď byla předána poslední dávka záznamů k zpracování, nebo bylo požadováno přerušení procesu uživatelem, nebo došlo k fatální chybě v komunikaci s OAI-PMH data repository. Celý proces je znázorněn na Obr. 15. U každého záznamu v převzaté dávce se provede kontrola jeho OAI-PMH identifikátoru, poté se záznam přečte a zpracuje (Obr. 16), přičemž algoritmus zpracování jednoho záznamu a jeho uložení do RDHF je uveden na Obr. 17.

Prvním krokem při zpracování příchozího metadatového záznamu je stanovení jednoznačné zkratky místa uložení fyzického dokumentu z elementů <repository> a <settlement>. Pokud byla odpovídající zkratka nalezena v souboru knihovny.xml, ze zkratky a signatury se sestaví identifikátor FyzId (Obr. 9). Poté se sestaví identifikátor digitální kopie DigCopyId (Obr. 12) a prověří se jeho (ne)existence v RDHF. Pokud v tabulce digitálních kopií (*rdhf_main*) již existuje záznam s tímto DigCopyID, znamená to, že záznam o této digitální kopii již v RDHF existuje. Pokud do této chvíle proběhlo všechno v pořádku, sestaví se záznam do tabulky *rdhf_main*. Do polí *Repository*, *Settlement* a *Country* se nevkládají automaticky informace z vstupního záznamu, ale jejich korektní hodnoty ze souboru knihovny.xml, resp. obsahy elementů <correct> (viz Obr. 8). Vytvořený záznam se potom uloží do databáze.



Obr. 15 Postup při vytěžování zdrojové OAI-PMH data repository

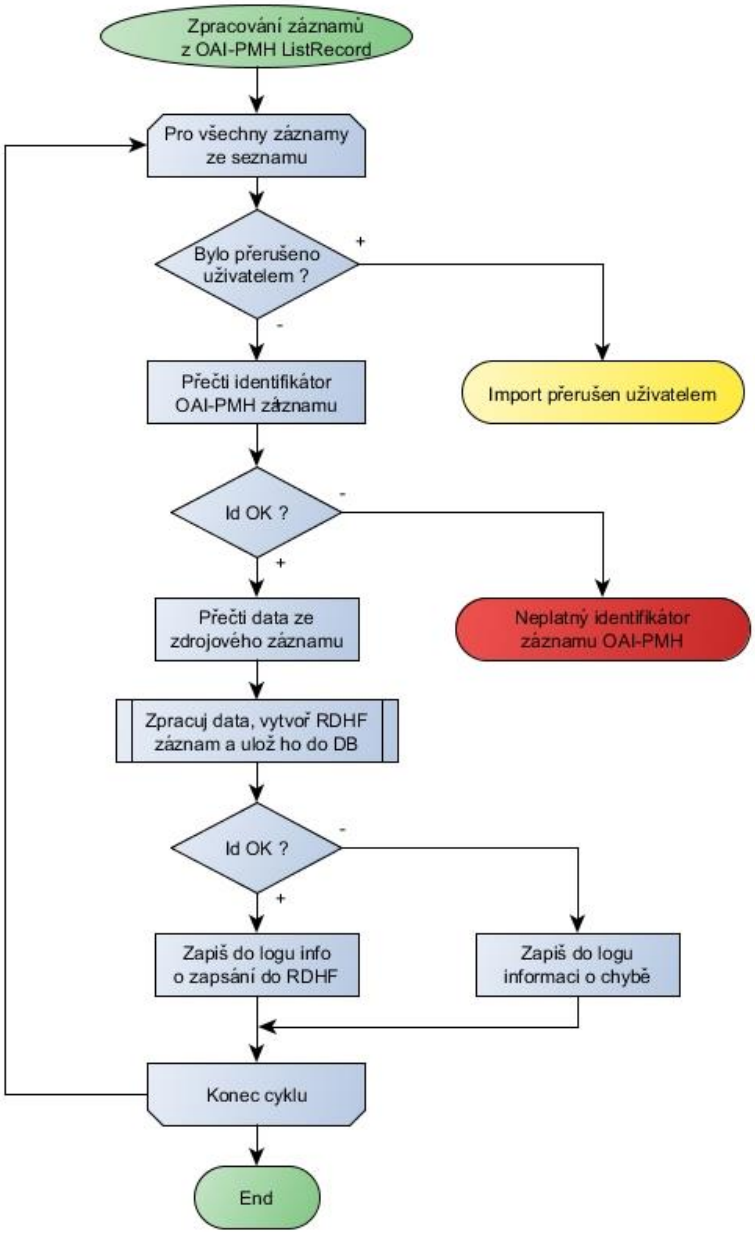
5.6.2 Zpracování vstupních záznamů v různých formátech

I když byl zpočátku realizován import pouze z jednoho zdroje, kterým je Manuscriptorium, je zřejmé, že pro účely registrace digitálních kopií historických fondů budou vytěžovány i další dostupné zdroje metadat. V knihovním prostředí je naprosto běžné, že metadata jsou uchovávána v různých systémech a také v různých formátech. Každý ze zdrojů metadat tak může předávat záznamy v jiném formátu (TEI P5, MARC, DC apod.). Řešení tohoto problému jsou dvě:

První řešení předpokládá účast poskytovatele metadat. Spočívá ve vytvoření dalšího metadatového formátu na straně OAI-PMH data repository, který bude speciálně určen pro vytěžování této repository harvesterem systému RDHF.

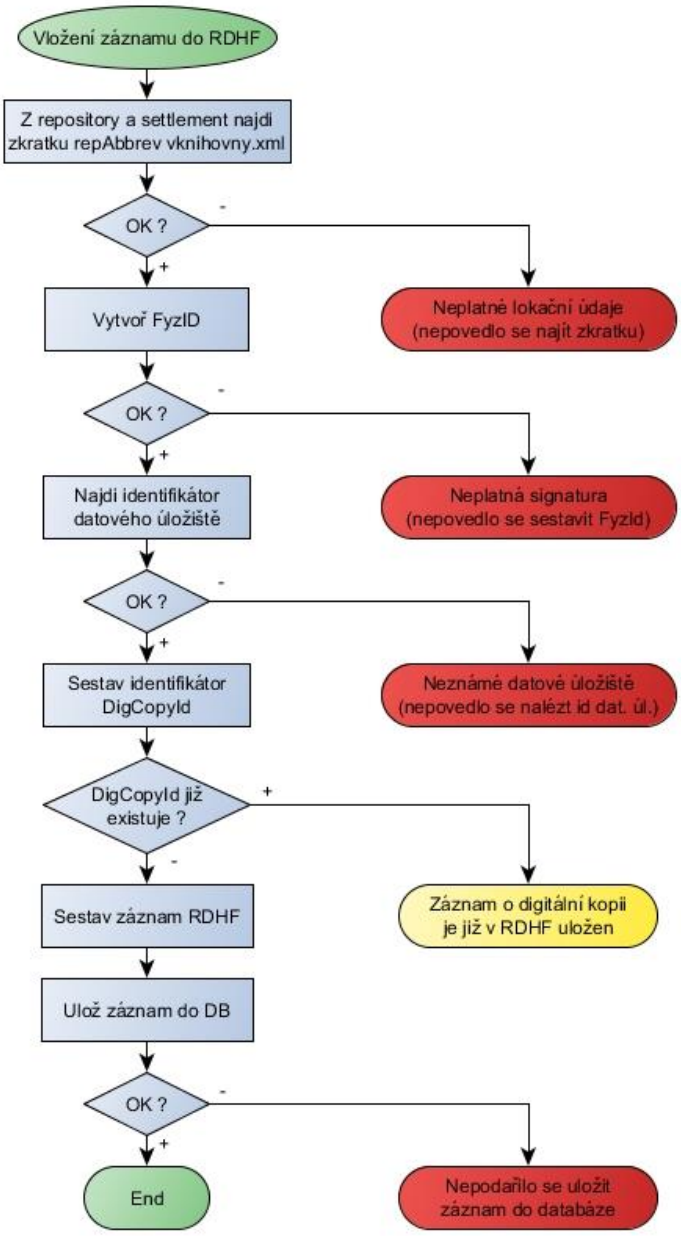
Naproti tomu druhé řešení akceptuje existenci různých formátů metadat na vstupu do RDHF. Importní modul je proto navržen tak, aby byl schopen zpracovat metadata přicházející v různých formátech.

Pro účely RDHF byla zvolena druhá varianta. Importní modul proto disponuje univerzálním programovacím rozhraním (interface) pro objekty (implementace tříd)



Obr. 16 Zpracování dávky vytěžených záznamů ze seznamu ListRecords

zpracovávající záznamy z jednotlivých zdrojů. Třídy pro zpracování vstupního záznamu musí implementovat toto jednoduché rozhraní. Názvy tříd pro zpracování metadat z konkrétních zdrojů (a formátů) jsou uvedeny v konfiguračním souboru serveru. Definice interface je uvedena ve Výpis 1. Metoda setInputData() zajišťuje přečtení vstupního záznamu a zbývající metody potom slouží k předání z něj extrahovaných informací k dalšímu zpracování.



Obr. 17 Postup vložení záznamu do RDHF

```
public interface RecordInput {
	void setInputData(Node metadataNode) throws TransformerException,
DOMException, SAXException, IOException, ParserConfigurationException;
	String getIdno();
	String getSettlement();
	String getRepository();
	String getCountry();
	String getTitle();
	String getAuthor();
	String getOrigDate();
	String getDocType();
	String getDigCopyUrl();
	String getSourceRepository();
	String getSourceRecPermalink();
	String getDataStorageURL();
}
```

Výpis 1 Interface pro zpracování vstupního záznamu

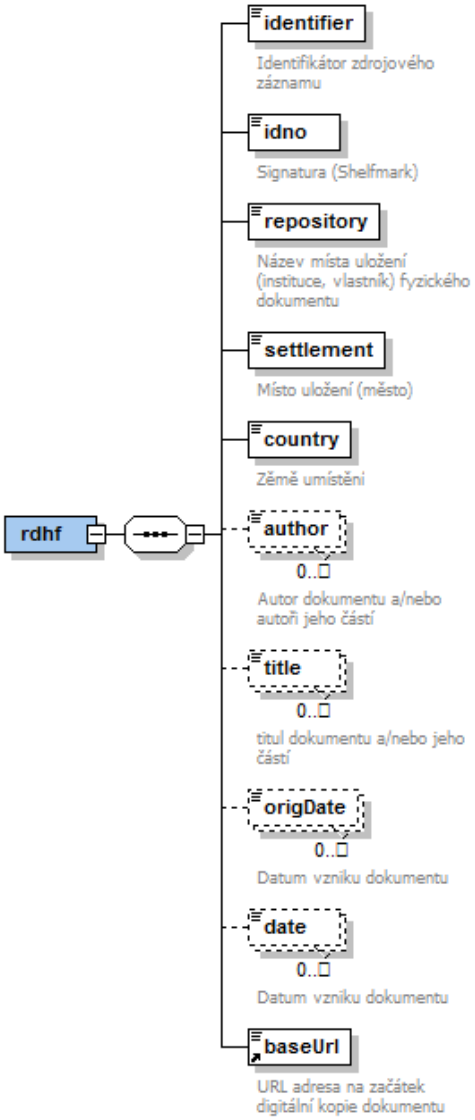
5.6.3 Formát metadat pro import z Manuscriptoria

Jak bylo zmíněno výše, současné řešení importního modulu akceptuje různé formáty metadat na vstupu. Přesto byl pro co nejjednodušší import dat z OAI-PMH data repository Manuscriptora vytvořen speciální formát metadat. Jeho podoba v okamžiku realizace pilotního projektu je uvedena na Obr. 18.

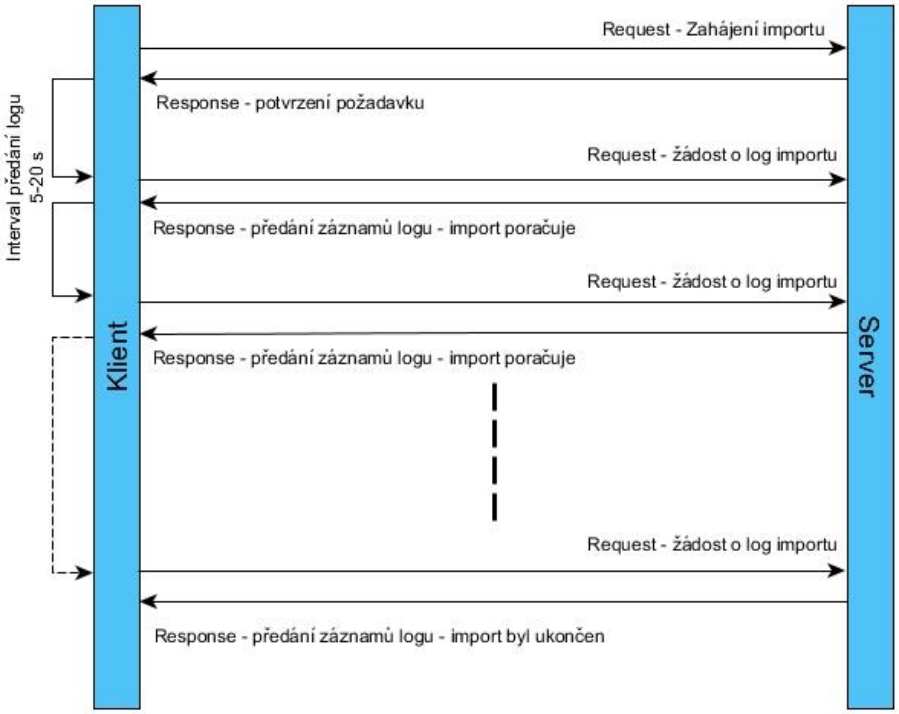
5.6.4 Logování importu metadat

Import metadat přes rozhraní OAI-PMH je podrobně logován. Logování probíhá na straně serveru, který vytváří v pracovním adresáři logovací soubor. V něm se soustřeďují informace o veškeré činnosti serveru od jeho spuštění po celou dobu běhu. Ukládá se sem také kompletní průběh importu metadat.

Informace o importu se předávají také klientské aplikaci. Importní modul serveru si k tomuto účelu při zahájení importu vytváří zvláštní buffer, do kterého se informace určené pro klientskou aplikaci ukládají. Klient nejprve pošle serveru požadavek na zahájení importu. Po potvrzení zahájení importu serverem posílá klient v pravidelných intervalech (5 – 20s) požadavek na informace o stavu importu (*importStatus*). Server v odezvě na každý požadavek *importStatus* vrátí klientu informace uložené v bufferu, poté ho vyprázdní a do prázdného bufferu pokračuje v zápisu stavových informací. Klient vysílá požadavek na status importu do doby, než v odezvě na předchozí požadavek dostane ze serveru informaci o ukončení importu. Získané informace ze serveru klient potom zapisuje do jednoho či více souborů, které slouží obsluze ke kontrole importu a analýze chyb při importu. Logují se také informace o těch záznamech, které byly importovány, ale v době importu již byly v RDHF obsaženy. Diagram komunikace klienta se serverem při importu metadat je uveden na Obr. 19.



Obr. 18 Schéma formátu pro import dat z Manuscriptoria



Obr. 19 Diagram komunikace klienta se serverem při importu metadat

Každý záznam o importu pro klienta má tři části:

- Status – obsahuje kód události nebo číslo chyby
- Text – vlastní textová informace o události při importu
- Info – doplňující informace, např. identifikátor vstupního záznamu

Jednotlivé stavové kódy jsou následující:

1 - START – zahájení importu
2 - END – ukončení importu
3 - ABORT – import byl přerušen uživatelem
100 - SUCCESS – záznam byl úspěšně uložen
101 - ERR_READ – chyba při čtení záznamu
102 - ERR_INSERT – chyba při zápisu záznamu do databáze
103 - ERR_EXISTS – vkládaný záznam již v RDHF existuje
111 - ERR_LOCATION – nesprávné lokační údaje
112 - ERR_OAI_ID – nesprávný identifikátor OAI-PMH záznamu
113 - ERR_REPOSITORY – nelze najít jednoznačnou zkratku místa uložení
114 - ERR_DATASTORAGE – nelze najít dané datové úložiště
128 - ERR_FATAL – katastrofální selhání

5.7 Autentifikace oprávněného uživatele

Registr digitalizace historických fondů nemá vlastní systém pro správu uživatelů a jejich oprávnění, pro účely autentifikace uživatelů využívá správu uživatelů v systému Manuscriptorium. Zde jsou pro jednotlivé uživatele uložena oprávnění také pro činnosti v RDHF.

Autentifikace oprávněného uživatele RDHF spočívá v předání identifikačního kódu uživatele (dále UserId) jako parametru každého požadavku na server, který autentifikaci uživatele vyžaduje. Server potom analýzou struktury UserId rozhodne o oprávnění uživatele k vykonání dané operace. Aby nebylo možno cizími prostředky zachytit a zneužívat identifikátory oprávněných uživatelů, je každý UserId doplněn kontrolním kódem s omezenou platností. Ta je defaultně nastavena na pět minut a lze ji v případě potřeby změnit v konfiguraci serveru. Celý mechanismus autentifikace funguje tak, že před požadavkem na službu serveru klient nejprve požádá server o kontrolní kód. Po jeho převzetí sestaví z tohoto kódu a oprávnění uživatele získaných při přihlášení identifikátor UserId a ten odešle jako jeden z parametrů spolu s požadavkem na server. Identifikátor UserId se skládá, ze čtyř částí:

- Role - role uživatele, například správce digitálních kopií, správce konkordancí, supervisor, administrátor
- Oprávnění – oprávnění daného uživatele jako je zápis záznamu, jeho editace či mazání
- Kontrolní kód – kód předaný serverem na vyžádání klienta, do serveru se vrací ke kontrole platnosti jako součást UserId
- Kód uživatele v Manuscriptoriu slouží např. jako identifikátor autora záznamu v digitálních konkordancích apod.

Struktura identifikátoru:

```
RRRCTRLCTPRMuuuuuuuuu – 8 znaků = kód uživatele z Mns
  |      |   |______ 3 znaky = oprávnění uživatele
  |      |__________ 6 znaků = kontrolní kód
  |_________________ 3 znaky = role uživatele
```

Role uživatele – udává, se kterými typy dat smí nakládat (RDHF, konkordance, datová úložiště), každá role ke každému typu dat je vyjádřena jedním znakem, které lze podle role i kombinovat.
1 – správce, 2 - supervisor, 999 – administrátor

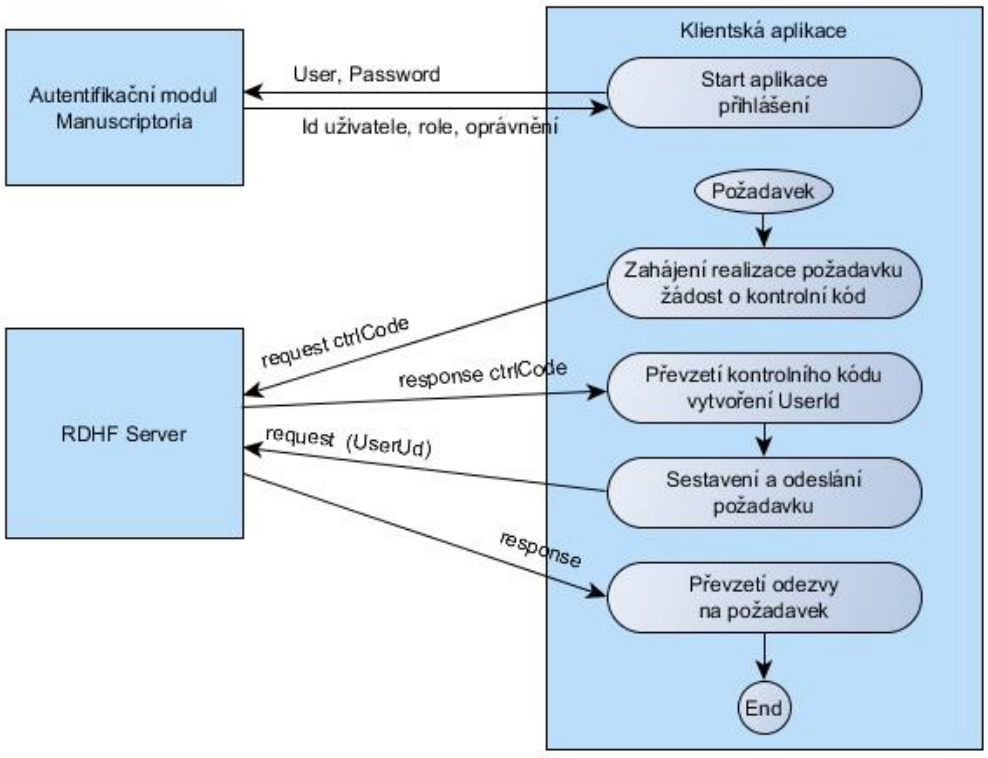
```
RRR
|||__ digitální kopie
||___ konkordance
|____ datová úložiště
```

Oprávnění uživatele – určuje činnosti, které je uživatel oprávněn s daty provádět.

```
PRM
|||__ digitální kopie
||___ konkordance
|____ datová úložiště
```

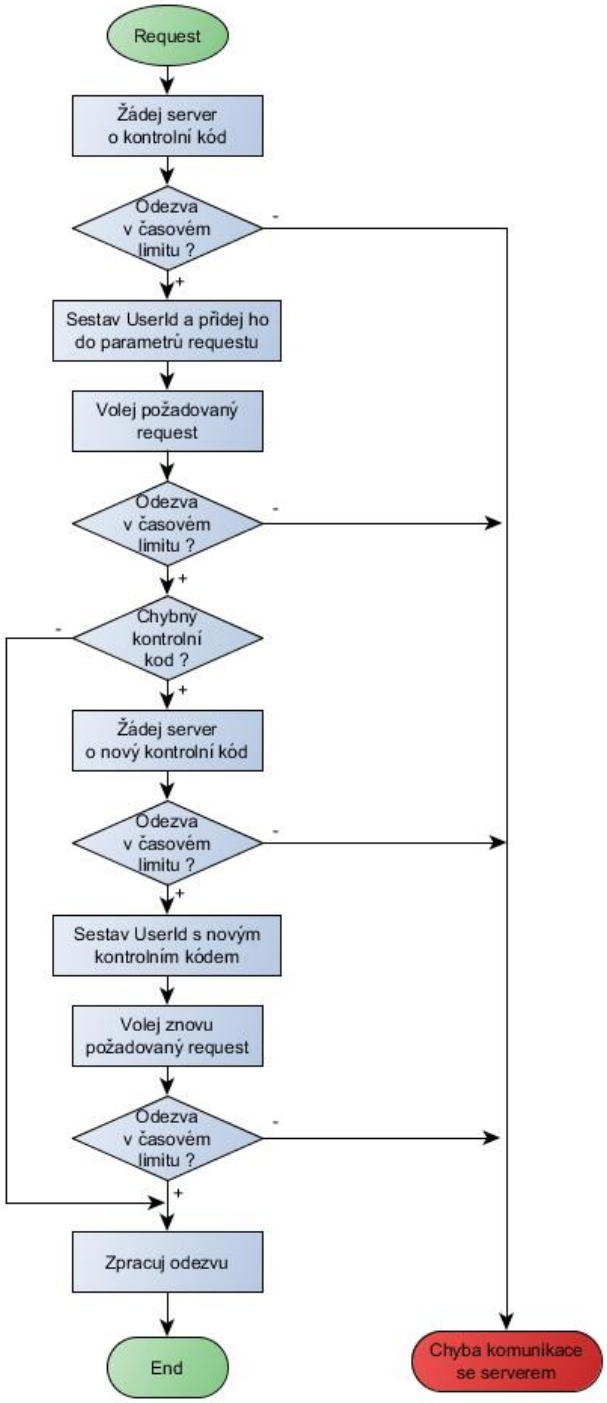
Oprávnění pro každou tabulku je vyjádřeno jedním znakem 0 – 4.
0 – pouze čtení, 1 – zápis, 2 – zápis a editace, 3 – mazání, 4 – úplný přístup

Vlastní proces autentifikace uživatele při provádění požadavku na server je na Obr. 20.



Obr. 20 Průběh zpracování požadavku s autentifikací uživatele

Celý mechanismus ovšem musí vzít v úvahu také situaci, že platnost kontrolního kódu skončila právě v průběhu realizace požadavku. Pokud odezva na požadavek předává chybu, klient nejprve zjistí, zda se nejedná o chybu „Invalid ctrl code" a pokud ano, zopakuje žádost o kontrolní kód. Ze získaného kontrolního kódu vytvoří nový UserId a zopakuje požadavek s tímto novým UserId. Zjednodušeně je algoritmus volání requestu ze strany klienta naznačen na Obr. 21.



Obr. 21 Algoritmus volání požadavku na server klientem

5.8 Správa dat uživatelem

Kromě automatizovaného importu metadatových záznamů do systému mohou pověření uživatelé prostřednictvím klientských aplikací vytvářet data v RDHF také

ručně. Prakticky se jedná o vložení, aktualizaci nebo smazání záznamů v tabulce digitálních kopií (*rdhf_main*), konkordancí, nebo datových úložišť.

5.8.1 Vložení záznamu o digitální kopii dokumentu

Zpracování požadavku na přidání záznamu o digitální kopii dokumentu probíhá v několika krocích.

- Nejprve se podle identifikátoru UserId ověří, zda má daný uživatel oprávnění vkládat záznamy do této databáze.
- Ze zadaných lokačních údajů (signatura, repository, settlement) se sestaví identifikátor FyzId a v tabulce konkordancí se ověří, zda nebyly změněny lokační údaje, tedy že záznam do RDHF vstoupil již dříve, ale s jinými lokačními údaji (pod jiným FyzId).
- Z databáze knihovny.xml se převezmou oficiální lokační údaje (repository, settlement, country)
- Z adresy URL začátku digitální kopie dokumentu se zjistí datové úložiště a jeho identifikátor (DsId)
- Vygeneruje se identifikátor digitální kopie dokumentu, který je zároveň i klíčem do tabulky *rdhf_main* a záznam se uloží do databáze.

Postup víceméně odpovídá způsobu vložení záznamu při automatizovaném importu, viz Obr. 17.

5.8.2 Aktualizace a mazání záznamů o digitálních kopiích dokumentů

Záznam, který má být aktualizován, je v tabulce digitálních kopií určen identifikátorem digitální kopie dokumentu DigCopyId. Před aktualizací či smazáním záznamu se podle identifikátoru UserId provede ověření oprávnění daného uživatele k aktualizaci či mazání záznamu. Při aktualizaci lze měnit všechny vložené údaje, kromě lokačních (repository a settlement). V signatuře lze měnit pouze formální nevýznamné znaky, které neovlivní tvorbu identifikátoru FyzId ale jen upraví jeho formální podobu (například mezery v signatuře jsou nahrazeny tečkou). Lze nahrazovat znaky 0x20..0x2F, 0x3A..0x40, 0x5C..0x60, 0x7B..0x7F a případně další znaky, které podle mechanismu popsaného v kapitole 5.3. neovlivní výslednou tvorbu identifikátoru FyzId a tím ani od ní odvozený identifikátor digitální kopie DigCopyId. Stejně tak při případné editaci adresy URL začátku digitální kopie nesmí dojít k takové její změně, při které dojde ke změně identifikátoru datového úložiště DsId. To by rovněž vedlo ke změně identifikátoru DigCopyId. Pokud tedy dojde k „přestěhování" digitální kopie fyzického dokumentu z jednoho datového úložiště na jiné, je třeba původní záznam v tabulce *rdhf_main* smazat a následně vytvořit nový. DigCopyId nového záznamu bude odrážet umístění na novém datovém úložišti. Mazání záznamů o digitálních kopiích je nevratné, smazaný záznam již nelze obnovit.

5.8.3 Vložení konkordančního záznamu

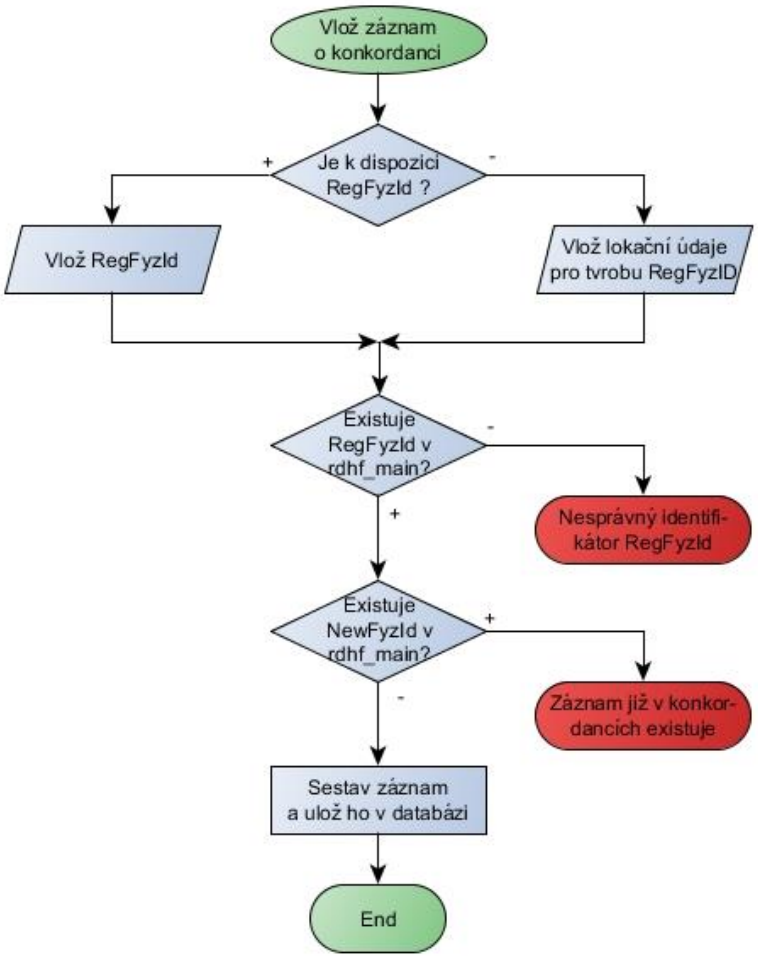
Záznam v tabulce konkordancí vypovídá o změně lokačních údajů jednoho fyzického dokumentu. Změna lokačních údajů dokumentu má za následek vznik jeho nového identifikátoru FyzId. V tabulce konkordancí je každý změnový záznam vztažen vždy ke konkrétnímu fyzickému dokumentu. K němu musí v RDHF existovat záznam o digitální kopii. Tento fyzický dokument je v tabulce digitálních kopií určen perzistentním identifikátorem RegFyzId, který je sestaven v okamžiku zápisu prvního záznamu o jeho digitální kopii do RDHF. Pokud chceme vložit záznam do tabulky konkordancí, musíme tedy nejprve určit identifikátor RegFyzId a ověřit existenci záznamu s tímto RegFyzId v tabulce *rhf_main*. Postup při vkládání záznamu do tabulky konkordancí je následující:

- Nejprve se podle identifikátoru UserId ověří, zda má daný uživatel oprávnění vkládat záznamy do této databáze.
- Ze zadaných lokačních údajů (signatura, repository, settlement) se sestaví identifikátor FyzId a ověří se jeho existence (jako RegFyzId) v tabulce *rdhf_main*.
- Z nových lokačních údajů (odlišných od těch, co jsou uvedeny v záznamu o digitální kopii) se sestaví nový identifikátor fyzického dokumentu (NewFyzId)
- Ověří se, zda záznam o této změně lokačních údajů již není v tabulce konkordancí zaznamenán (v tabulce konkordancí již toto FyzId existuje).
- Z databáze knihovny.xml se převezmou oficiální lokační údaje (repository,settlement,country) podle zkratky nového místa uložení.
- Sestaví se záznam o konkordanci a ten se uloží do tabulky *concordance*. Unikátním klíčem v této tabulce je nový FyzId.

Algoritmus vložení záznamu o konkordanci je uveden na Obr. 22.

5.8.4 Aktualizace a mazání konkordančního záznamu

Konkordanční záznam, který má být aktualizován nebo smazán, je v tabulce *concordance* určen unikátním klíčem FyzId (na rozdíl od RegFyzId, který se může v tabulce vyskytovat opakovaně). Před aktualizací či smazáním záznamu se podle identifikátoru UserId provede ověření oprávnění daného uživatele k aktualizaci či mazání záznamu v této tabulce. Podobně jako u digitálních kopií lze editovat všechny vložené údaje, kromě lokačních. Pro editaci signatury platí stejná pravidla jako u záznamu v tabulce *rdhf_main*. Rovněž mazání záznamů v této tabulce je nevratné.



Obr. 22 Uložení záznamu o konkordanci

5.8.5 Vložení, editace a mazání záznamu o datovém úložišti

Záznam v tabulce digitálních kopií *data_storage* je velmi jednoduchý. V současné době sestává pouze z identifikátoru datového úložiště DsId, který je zároveň unikátním klíčem v tabulce, a adresy URL tohoto datového úložiště. Záznam se vytváří vložením obou informací o datovém úložišti, přičemž editovat lze pochopitelně pouze adresu datového úložiště. Smazání záznamu v této tabulce je nevratné.

5.9 API pro klientské aplikace

Klientské aplikace komunikují se serverem Registru digitalizace historických fondů prostřednictvím definované množiny požadavků (requestů). Tato komunikace probíhá na protokolu http, je bezestavová a používá architekturu REST. Pro

komunikaci klienta se serverem byla navržena tři rozhraní (REST resources). Každé z nich obsahuje specifickou množinu requestů pro určitý typ klientské aplikace. Rozhraní *Search* je určené pro veřejné aplikace, *Manage* pro správce a supervizory systému, *Admin* pro administraci systému.

REST rozhraní k serverové části projektu bude mít vždy následující strukturu:

```
http://<adresa_serveru>:<port>/<typ_rozhraní>/<request>?<parametry>
```

kde:

- <adresa_serveru> - je URL adresa, na které je k dispozici server RDHF
- <port> - je port pro server RDHF
- <typ_rozhraní> - (resource) definuje činnost, kterou bude server provádět
- <request> - je požadavek, který má být proveden

Pro komunikaci je defaultně použit port 8080. Všechna volání služeb serveru jsou asynchronní, tzn., že klient není blokován čekáním na odezvu. Služby rozhraní *Search* jsou dostupné metodami GET a POST, služby *Admin* a *Manage* pouze metodou POST. Rozhraní *Admin* a *Manage* obsahují navíc request *ctrlCode*, který vrací právě aktuální kontrolní kód následně použitý klientem pro sestavení platného identifikátoru uživatele UserId.

Kompletní popis aplikačního rozhraní serveru RDHF je předmětem samostatného dokumentu.

5.9.1 Rozhraní Search

Rozhraní *Search* na straně serveru obsluhuje třída PublicUser. Rozhraní se z klientské aplikace se volá podle následujícího příkladu.

```
http://<rdhf_server>:8080/search/<request>?<parametry>
```

Rozhraní *Search* tvoří následující množina požadavků:

Digitální kopie dokumentu

- **digCopyIds** - vrátí v odezvě seznam identifikátorů všech digitálních kopií, vyhledaných podle zadaných parametrů volání
- **digCopyRec** - vrátí v odezvě kompletní záznam(y) o digitální kopii (nebo kopiích) daného fyzického dokumentu jako výsledek hledání podle zadaných parametrů volání
- **allDigCopyIds** - vrátí všechny identifikátory digitálních kopií, které odpovídají zadaným lokačním údajům
- **allDigCopyRec** - vrátí všechny záznamy o digitálních kopiích, které odpovídají zadaným lokačním údajům
- **getFyzId** - z identifikačních údajů sestaví identifikátor fyzického dokumentu FyzId a předá ho v odezvě

- **verifyRepository** - ověří, zda vložené údaje o repository (repository a settlement) mají odpovídající zkratku repository v souboru knihovny.xml
- **getRepositoryList** - vrátí seznam všech zkratek a názvů knihoven (repositories), jejich umístění a zemi

Konkordance

- **concordRec** - předá záznam z tabulky konkordancí určený parametrem fyzId
- **getRegFyzId** - z lokačních údajů sestaví FyzId a ověří ho v tabulce konkordancí. V odezvě předá RegFyzId a jemu odpovídající identifikační údaje. Pokud v tabulce konkordancí nebyl nalezen záznam s tímto FyzId, předpokládá se, že FyzId je zároveň perzistentní identifikátor RegFyzId.
- **docLocationHistory** - vyhledá v tabulce konkordancí veškeré záznamy o změnách umístění požadovaného fyzického dokumentu a předá tyto záznamy v odezvě

Resolver

- **gotoDigCopy** - vrátí všechny dostupné informace potřebné k zobrazení digitální kopie dokumentu
- **gotoDocAllDigCopies** - předá všechny přístupové parametry ke všem digitálním kopiím požadovaného fyzického dokumentu

5.9.2 Rozhraní Manage

Rozhraní *Manage* na straně serveru obsluhuje třída Manager. Toto rozhraní požaduje pro všechny své požadavky autentifikaci uživatele, tzn., že v parametrech volání musí být identifikátor uživatele UserId. Komunikace mezi klientem a serverem přes rozhraní *Manage* probíhá metodou POST. Volání požadavku klient provede podle následujícího příkladu.

```
http://<rdhf_server>:8080/manage/<request>?userId=xxxx&<parametry>
```

Rozhraní *Manage* tvoří následující množina požadavků:

Digitální kopie dokumentu

- **addRdhfRecord** - vloží nový záznam do hlavní tabulky registru digitalizace RDHF (tabulka digitálních kopií dokumentů *rdhf_main*)
- **updateRdhfRecord** - provede aktualizaci požadovaného záznamu v hlavní tabulce RDHF
- **deleteRdhfRecord** - smaže požadovaný záznam v hlavní tabulce RDHF

Konkordance

- **addConcordRecord** - přidá záznam do tabulky konkordancí
- **updateConcordRecord** - provede aktualizaci požadovaného záznamu v tabulce konkordancí
- **deleteConcordRecord** - smaže požadovaný záznam v tabulce konkordancí

Datová úložiště

- **getDataStorageList** - předá v odezvě kompletní seznam datových úložišť (nebo jeho požadovanou část). Tento request je možno volat také z rozhraní *Admin*
- **addDataStorageRecord** - přidá záznam do tabulky datových úložišť. Tento request je možno volat také z rozhraní *Admin*
- **updateDataStorageRecord** - provede aktualizaci požadovaného záznamu v tabulce datových úložišť. Tento request je možno volat také z rozhraní *Admin*
- **deleteDataStorageRecord** - smaže požadovaný záznam v tabulce datových úložišť. Tento request je možno volat také z rozhraní *Admin*.

5.9.3 Rozhraní Admin

Stejně jako rozhraní *Manage*, požaduje i *Admin* pro zpracování požadavků autentifikaci uživatele a proto pro všechny požadavky je povinný parametr UserId. Rovněž komunikace mezi klientem a serverem přes rozhraní *Admin* probíhá metodou POST. Na straně serveru služby tohoto rozhraní zajišťuje třída Admin.

```
http://<rdhf_server>:8080/admin/<request>?userId=xxxx&<parametry>
```

Rozhraní *Admin* tvoří následující množina požadavků:

- **dataImport** - provede import dat do tabulky digitálních kopií z požadovaného zdroje
- **stopDataImport** - zastaví provádění právě spuštěného importu
- **importStatus** - předá informace o stavu importu od začátku nebo od okamžiku posledního předání informací requestem „importStatus" do okamžiku nového volání tohoto requestu
- **reloadConfig** - zajistí znovunačtení konfiguračního souboru serveru za běhu
- **reloadRepositoryList** - zajistí okamžitou synchronizaci souboru knihovny.xml
- **version** - předá informaci o verzi serveru a datu jejího vydání
- **serverAccess** - povolí nebo zakáže přístup všem uživatelům (kromě administrátora) k běžícímu serveru
- **getServerAccess** - zjistí aktuální stav serveru RDHF, tj. zdali je či není server RDHF přístupný uživateli

- **serverInfo** - předá dostupné informace o nastaveních (konfiguraci) serveru RDHF

5.10Inicializace serveru

RDHF server je realizován jako servlet, pracující bezestavově. Nastavení vlastností serveru a čtení dalších informací potřebných pro realizaci požadavků klientů se provádí při spuštění serveru. Změnu konfigurace lze také provést na vyžádání za běhu servletu.

Veškerá potřebná data pro běh servletu jsou ukládána v jeho kontextu (třída ServletContext). Jsou to:

- Název kontextového parametru v souboru web.XML (obsah elementu <context-param><param-name) – standardně „configFile"
- Singleton ConfigManager pro správu konfigurace serveru
- Kompletní mapa míst uložení fyzických dokumentů, získaná zpracováním souboru knihovny.xml
- Právě aktuální kontrolní kód pro sestavení UserId
- Instance třídy zpracovávající stavové informace při importu dat pro klientskou aplikaci
- Hlavní stránka projektu (html). Může sloužit jako úvodní stránka projektu nebo pro testovací účely apod.
- Stavová informace (semafor) „serverEnabled", která říká, zda je server právě přístupný uživatelům či nikoliv

Počáteční inicializaci i konečnou finalizaci servletu zajišťuje třída RdhfContextListener, která implementuje rozhraní ServletContextListener. K tomuto účelu obsluhuje události contextInitialized() a contextDestroyed(). Třída je deklarována v souboru web.xml servletu takto:

```
<listener>
      <listener-class>
            cz.aipberoun.rdhf.RdhfContextListener
      </listener-class>
</listener>
```

V souboru web.xml je také definována cesta ke konfiguračnímu souboru servletu:

```
<context-param>
      <param-name>configFile</param-name>
      <param-value>D:\workspace\RdhfSrv\work\config.xml</param-value>
</context-param>
```

V obsluze události contextInitialized() třída RdhfContextListener nejprve ze souboru web.xml zjistí cestu ke konfiguračnímu souboru a vytvoří singleton ConfigManager, který spravuje konfiguraci serveru. Dále spustí dva časovače -

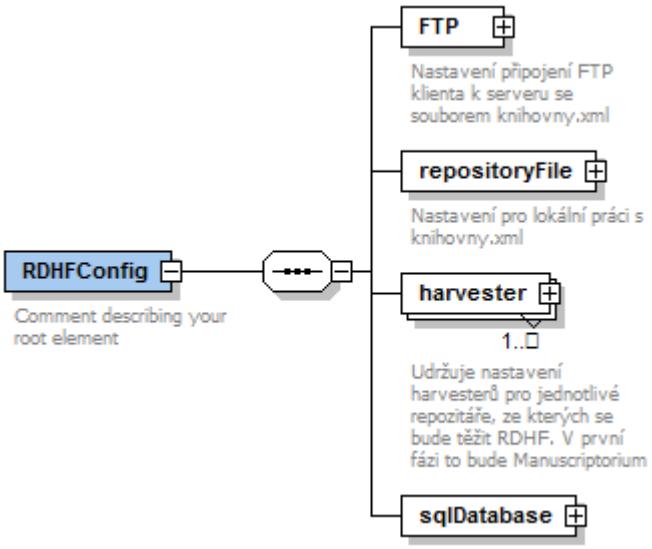
démony. První z nich zajistí provádění automatické aktualizace pracovní verze souboru knihovny.xml v definovaném intervalu. Ten byl přednastaven na jednu hodinu. V dalších verzích bude tento čas definován v konfiguračním souboru a bude ho možno dle potřeby měnit. Celý proces aktualizace pracovní verze souboru knihovny.xml byl podrobně popsán v 5.3.3 a je uveden na Obr. 10. Druhý časovač zajišťuje automatické generování kontrolního kódu pro generování identifikátoru UserId v intervalu pěti minut.

V obsluze události contextDestroyed() třída RdhfContextListener uvolní démony spuštěné při inicializaci.

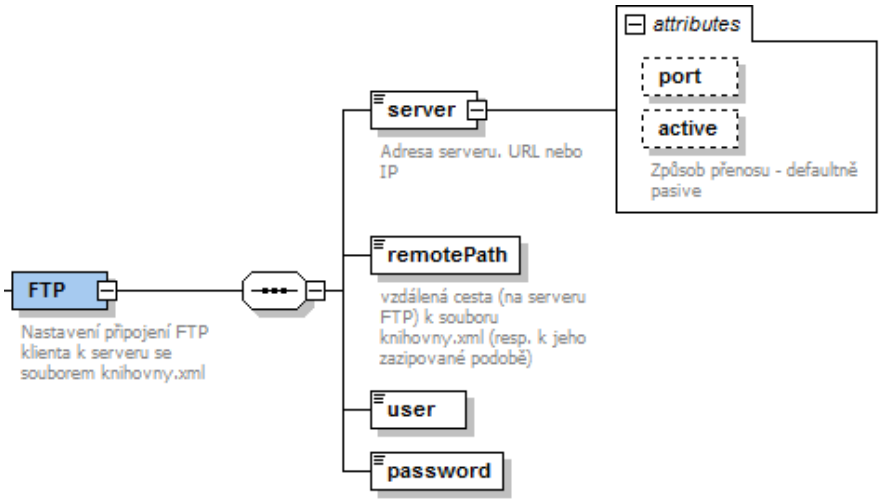
Konfigurační soubor nemusí být načten jen při spouštění servletu. Uživatelské rozhraní *Admin* má k dispozici požadavek *reloadConfig*, který umožní uživateli provést změnu parametrů serveru bez nutnosti jej restartovat. Stejně tak má administrátor systému možnost na vyžádání provést synchronizaci pracovní verze souboru knihovny.xml mimo definovaný interval jedné hodiny, a to přímým voláním požadavku *reloadRepositoryList*.

5.11 Konfigurační soubor

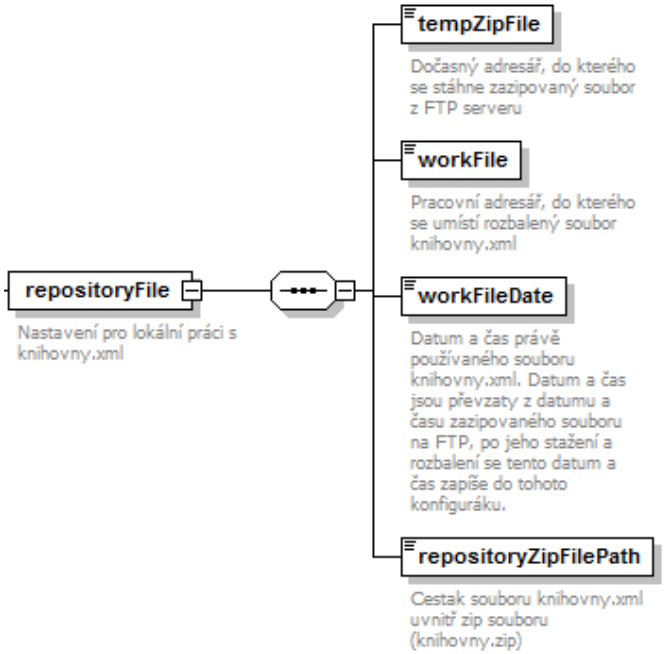
Konfigurační soubor umožňuje nastavit základní vlastnosti RDHF serveru. V této verzi se skládá ze čtyř částí. První sekce - element <FTP> - udržuje informace o FTP serveru, ke kterému se RDHF server připojuje pro zajištění synchronizace pracovního souboru knihovny.xml. V části konfigurace obsažené v elementu <repositoryFile> jsou informace potřebné k zpracování převzatých dat z FTP serveru. Element <harvester> uchovává informace o jednotlivých zdrojích pro import metadat a jejich vlastnostech. V sekci <sqlDatabáze> jsou uvedeny parametry připojení k SQL databázi. Základní struktura konfiguračního souboru je uvedena na Obr. 23.



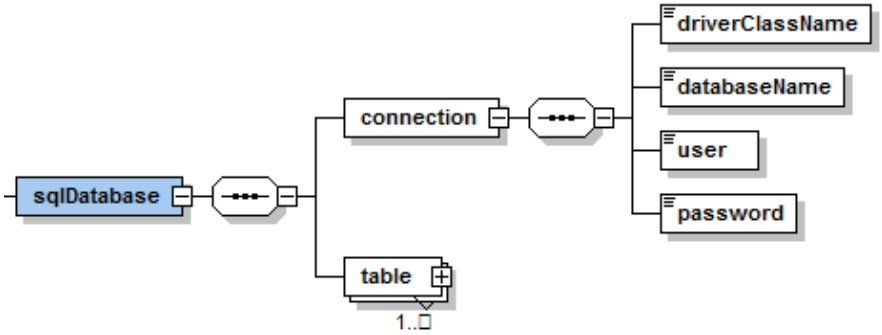
Obr. 23 Základní struktura konfiguračního souboru



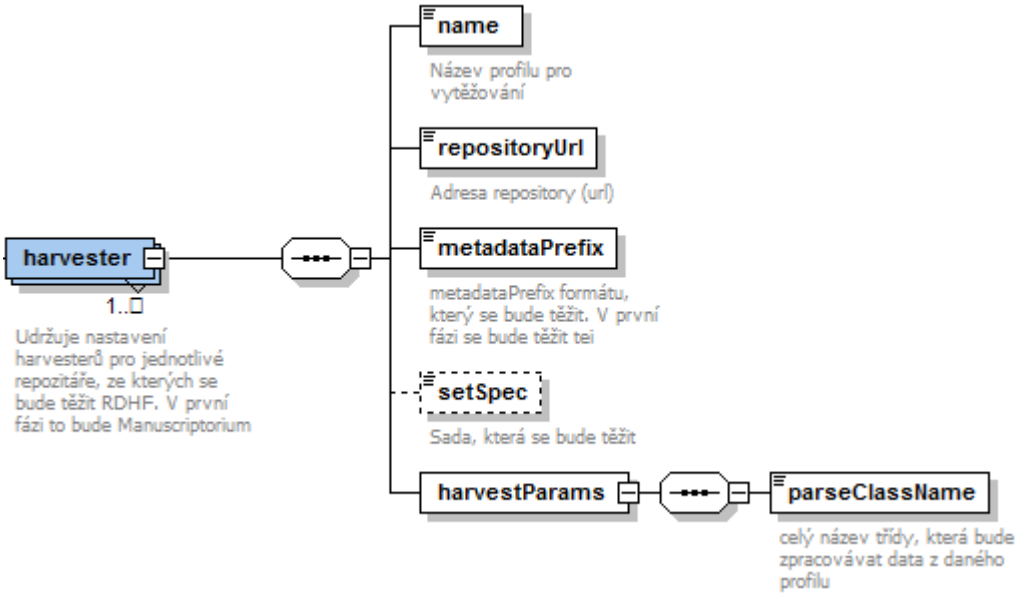
Obr. 24 Připojení k FTP serveru



Obr. 25 Informace pro synchronizaci knihovny.xml



Obr. 26 Informace pro připojení k relační databázi



Obr. 27 Návrh struktury informací pro OAI-PMH harvester

Na obrázcích Obr. 24, Obr. 25 a Obr. 26 jsou ukázány jednotlivé sekce konfiguračního souboru. Obr. 27 představuje návrh struktury informací potřebných pro úspěšné vytěžování dostupných OAI-PMH data repositories za účelem importu do RDHF.

6 Další rozvoj systému RDHF

Jak bylo zdůrazněno již v úvodu tohoto dokumentu, pro účely poloprovozu byla podle předchozí analýzy ([1] a [2]) zrealizována pilotní verze systému. Ta splňuje základní nároky na RDHF a plně umožňuje spuštění RDHF v režimu poloprovozu. Po dokončení této verze systému však autoři spatřují celou řadu dalších možností rozvoje RDHF.

Pro účely poloprovozu byl učiněn předpoklad, že v této fázi projektu bude několik pověřených uživatelů spravovat celý RDHF. Proto byla správa uživatelů velmi zjednodušena. Je zřejmé, že především v oblasti správy digitálních kopií dokumentů bude potřeba diverzifikovat správce záznamů s kompetencemi pouze v rozsahu metadatových záznamů určitého vlastníka či správce fyzických dokumentů či jejich digitálních kopií. To by mělo zabránit neoprávněné manipulaci s metadatovými záznamy napříč různými zdroji metadat a vlastníků či správců dokumentů. Konkrétní strukturu kompetencí správců ve vztahu k vlastníkům záznamů, digitálních kopií nebo jejich fyzických předloh stanoví další analýza tohoto problému.

Pro pilotní řešení byly nativní aplikace správců a administrátorů vytvořeny pro operační systém Windows 32/64bit. V případě potřeby lze pokračovat ve vývoji těchto aplikací pro operační systém OSX či mobilní platformy (Android, iOS).

Dalším tématem rozvoje RDHF je doplnění technických metadat k popisným metadatům digitálních kopií dokumentů. Ukazuje se, že tato technická metadata lze získat poměrně snadno k digitálním kopiím dokumentů, jejichž digitalizace proběhla v projektu VISK6. Naproti tomu u ostatních zdrojů digitálních kopií bude získání technických metadat dáno především úspěšností při jednání s jejich vlastníky či správci. Spolu se získáním těchto dat bude potřeba vytvořit mechanismy pro jejich vložení do stávajícího RDHF.

Současná verze systému RDHF umožňuje hromadný import dat pouze prostřednictví vytěžovacího rozhraní OAI-PMH. Lze předpokládat, že do budoucna budou dostupná data z různých dalších zdrojů, které nemusí vždy disponovat rozhraním OAI-PMH. Bude proto potřeba navrhnout a vytvořit další importní moduly pro odlišné způsoby hromadného vstupu metadatových záznamů do RDHF.

Při návrhu systému také nebyla řešena situace, kdy dojde k hromadnému stěhování digitálních kopií z jednoho úložiště na jiné. Změna datového úložiště pro část na něm uložených digitálních kopií z organizačních důvodů nebo například po změně jejich správce může mít za následek nesoulad částí jejich identifikátorů, které jsou odvozeny právě od identifikátoru datového úložiště. Tuto situaci je potřeba analyzovat a následně navrhnout způsob řešení.

V této fázi projektu není také brána v úvahu možnost, že dojde k hromadným změnám v uložení fyzických dokumentů, jež tvoří předlohy k digitálním kopiím registrovaným v RDHF. Tím dojde také ke změně lokačních údajů v metadatech velkého množství fyzických dokumentů což následně vygeneruje stejné množství záznamů v tabulce konkordancí. K hromadným změnám může dojít například při změně oficiálního názvu organizace, její reorganizaci či přemístění fondů a podobně. Tento případ lze řešit návrhem a vytvořením služby pro hromadnou a

automatizovanou tvorbou záznamů konkordancí reflektující změnu lokačních údajů dotčených dokumentů.

Z realizace projektu RDHF také vyplynula nutnost revize technického řešení databáze míst uložení fyzických dokumentů. Současný stav, kdy existuje jeden centrálně udržovaný XML soubor a jednotlivé systémy používají (read-only) pracovní kopie tohoto souboru, se jeví dále neudržitelný. Je to právě z důvodu narůstajícího počtu systémů a aplikací, které tuto databázi využívají. Kromě RDHF je to samozřejmě Manuscriptorium a systémy a aplikace v projektu VISK6. Ukazuje se proto nutnost tuto databázi přetvořit do podoby serveru zajišťujícího jak bezpečnou údržbu a správu databáze pověřenými uživateli, tak zpřístupnění databáze dalším systémům a uživatelským aplikacím. Součástí serveru by měl být také centrální generátor identifikátorů fyzických dokumentů FyzId, který je nejvýznamnějším uživatelem databáze míst uložení fyzických dokumentů. Centralizace generování jednoznačných identifikátorů fyzických dokumentů zajistí kromě možnosti identifikace všech druhů duplicit v Manuscriptoriu také mnohem vyšší jednoznačnost propojení metadatových záznamů fyzických dokumentů Manuscriptoria s jejich digitálními kopiemi umístěnými na datových úložištích. Reorganizace databáze míst uložení fyzických dokumentů a vytvoření centrálního generátoru FyzId je svou důležitostí a svým rozsahem námětem na samostatný projekt.

Dalším námětem vyplývajícím ze současného řešení Registru digitalizace historických fondů je zavedení digitální konkordance nejen pro digitalizované dokumenty, ale pro všechny fyzické dokumenty, jejichž metadatové záznamy jsou obsaženy v katalogu Manuscriptoria. Podobně jako v RDHF by lokační údaje z příchozích metadatových záznamů procházely tabulkou konkordancí. Identifikace více metadatových záznamů s různými lokačními údaji (a tím i různými identifikátory FyzId) k jednomu fyzickému dokumentu by spolu s centrálním generátorem jejich identifikátorů FyzId zároveň eliminovala další zdroj duplicit v Manuscriptoriu. Takovéto komplexní řešení digitálních konkordancí v rámci celého systému Manusciriptorium a Registru digitalizace historických fondů by ovšem znamenalo významnou rekonstrukci stávajícího systému pro správu dat Manuscriptoria. Velkým přínosem upgradu správního systému by však bylo dokončení identifikace a odstranění duplicit z katalogu Manuscriptoria a tím také jednoznačné navázání záznamů katalogu Manuscriptoria na digitální kopie dokumentů na datových úložištích. Rozšířené možnosti provázání katalogu s dalšími souvisejícími informacemi o fyzických historických dokumentech by dále pomohly zkvalitnit a rozšířit badatelské prostředí Manuscriptoria.

Přílohy

A. Záznam ze souboru knihovny.xml pro NKČR

```
<repository>
  <id>NKCR__</id>
  <name>
   <correct>Národní knihovna České republiky</correct>
   <alternate>Národní Knihovna</alternate>
   <alternate>Narodní knihovna České republiky (National Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna ČR (National Library of the
CR)</alternate>
   <alternate>N8rodní knihovna České republiky (National Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
the Czech Library)</alternate>
   <alternate>Národní- knihovna České republiky (National library of
the Czech republic)</alternate>
   <alternate>Nk ČR</alternate>
   <alternate>NKČR</alternate>
   <alternate>Národní knihovna v Praze</alternate>
   <alternate>Národní knihovna České Republiky</alternate>
   <alternate>National Library of the Czech Library</alternate>
   <alternate>Národní knihovna České republiky</alternate>
   <alternate>Národní knihovna české republiky</alternate>
   <alternate>Národní knihovna Českér republiky</alternate>
   <alternate>NK ČR</alternate>
   <alternate>National library of Czech Republic</alternate>
   <alternate>National Library of the Czech Republic</alternate>
   <alternate>National Library the Czech Republic</alternate>
   <alternate>National library of the Czech Republic</alternate>
   <alternate>National Library of the Czech Republic,
Prague</alternate>
   <alternate>National Library of the Czech Republik</alternate>
   <alternate>National Library of the Czech Republic
Prague</alternate>
   <alternate>National Library of Czech Republic</alternate>
   <alternate>National Library The Czech Republic</alternate>
   <alternate>The National Library of the Czech Republic</alternate>
   <alternate>National Library of the Czech Republic</alternate>
   <alternate>Natiönal Library of Czech Republik</alternate>
   <alternate>Natioanal Library of the Czech Republic</alternate>
   <alternate>National library of the Czech Republic,
Prague</alternate>
```

```
   <alternate>National Library of Czech republic</alternate>
   <alternate>Národní knihovna České republiky (National library of
the Czech republic)</alternate>
   <alternate>Národní knihovna České republiky (National library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky v Praze (National
library of the Czech republic in Prague)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
the Czech republic)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
the Czech Republic</alternate>
   <alternate>Národní knihovna Česká republiky (National Library of
the Czech republic)</alternate>
   <alternate>Národní knihovna ČR (National Library of the Czech
Republic)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky )National Library of
the Czech Republic)</alternate>
   <alternate>Národní kihovna ČR (National Library of the Czech
Republic)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
the Czech Rebublic)</alternate>
   <alternate>Národní knihovna české republiky (National Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna české republiky (National library of
the Czech republic)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky (The National Library
of the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
Czech Republic)</alternate>
   <alternate>Národní knihovna ČR (National Republic of the Czech
Republic)</alternate>
   <alternate>Národní knihovna České republiky (National Libratry of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky (National Library of
Czech republic)</alternate>
   <alternate>Národní knihovna České republiky (National library of
the Czech republic</alternate>
```

```
   <alternate>Národní knihovna České republiky (National Library of
the Prague)</alternate>
   <alternate>Národní knihovna České republiky (National LIbrary of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky National Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky (National lIbrary of
the Czech Republic)</alternate>
   <alternate>Národní- knihovna České republiky (National Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna České republiky (Nationaly Library of
the Czech Republic)</alternate>
   <alternate>Národní knihovna</alternate>
   <alternate>Národní knihovna ČR</alternate>
  </name>
  <settlement>
   <correct>Praha</correct>
   <alternate>Prague</alternate>
   <alternate>Praha (Prague)</alternate>
  </settlement>
  <country>
   <reg>CZ</reg>
   <correct>Česká republika</correct>
   <alternate>Česko</alternate>
  </country>
 </repository>
```

B. Soubor web.xml projektu

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID"
version="3.1">
      <display-name>RdhfSrv</display-name>
      <listener>
            <listener-class>cz.aipberoun.rdhf.RdhfContextListener
            </listener-class>
      </listener>
      <servlet>
            <servlet-name>Jersey REST Service</servlet-name>
            <servlet-class>
                  org.glassfish.jersey.servlet.ServletContainer
            </servlet-class>
            <init-param>
```

```
                <param-name>jersey.config.server.provider.packages
                </param-name>
                <param-value>cz.aipberoun.rdhf</param-value>
            </init-param>
            <load-on-startup>1</load-on-startup>
        </servlet>
        <servlet-mapping>
            <servlet-name>Jersey REST Service</servlet-name>
            <url-pattern>/*</url-pattern>
        </servlet-mapping>
        <context-param>
            <param-name>configFile</param-name>
            <param-value>D:\workspace\RdhfSrv\work\config.xml
            </param-value>
        </context-param>
        <context-param>
            <param-name>homePage</param-name>
            <param-value>WEB-INF/index.html</param-value>
        </context-param>
        <welcome-file-list>
            <welcome-file>index.html</welcome-file>
            <welcome-file>index.htm</welcome-file>
            <welcome-file>index.jsp</welcome-file>
        </welcome-file-list>
</web-app>
```