

Nmet - Standardizace přenosu informací mezi software, které slouží k popisu archiválií, a digitálním archivem

Konečný uživatel výsledků: **Ministerstvo vnitra České republiky**
Nad Štolou 3, 170 34 Praha 7

Název projektu: Vytvoření standardů pro komunikaci informačního systému digitálního archivu
s jeho okolím

Číslo projektu: TITSMV301

Řešitel projektu: LightComp v.o.s., Marbes s.r.o.

Doba řešení: 1. 10. 2023 – 30. 9. 2024

Důvěrnost a dostupnost: veřejně přístupný

Informace o autorském týmu:

Ing. Petr Pytelka (LightComp v.o.s.)
Ing. Karel Žáček (LightComp v.o.s.)
Jan Vejskal (Marbes s.r.o.)
PhDr. Jakub Doležal (LightComp v.o.s.)
Ing. Tomáš Pytelka (LightComp v.o.s.)

**Další informace o projektu:**

Metodika definuje postupy a technického standardy (definice výměnných balíčků dat), jejíž cílem je předávání informací mezi digitálním archivem a systémem, který slouží k popisu archiválií (např. ELZA). Předávání informací se rozumí předávání digitálních archiválií (replik) a jejich metadat a také metadat analogových archiválií ve směru z DA, ale také metadat, které vznikají při popisu archiválií ve směru do DA.

Komunikační rozhraní mezi software pro popis archiválií a digitálními archivy

LightComp v.o.s., Marbes s.r.o.

25.11.2024

1	Účel a užití standardu	1
1.1	Zkratky	1
1.2	Termíny a pojmy	2
1.3	Standardy a specifikace	3
1.4	Případy užití	5
1.5	Architektura propojení	8
1.6	Právní okolí a souvislosti	9
2	Princip komunikace	13
2.1	Způsob komunikace	13
2.2	Princip a účel REST rozhraní	18
2.3	Princip a účel přenosového rozhraní	18
3	Přístup k balíčkům	19
3.1	Verze balíčků	19
3.2	Základní informace o balíčku	20
3.3	Archivní metadata	21
3.4	Obsah balíčku	21
3.5	Soubory a komponenty	22
4	Změnové balíčky	25
4.1	Účel změnových balíčků	25
4.2	Struktura změnových balíčků	26
4.3	Přidání metadatového souboru	27
4.4	Archivní popis	27
5	REST rozhraní	29
5.1	Zabezpečení	29
5.2	Popis rozhraní	31
5.3	Implementační doporučení	35
6	Přenosové rozhraní	39
6.1	Zabezpečení	39
6.2	FileTransfer	39
6.3	Stažení balíčků a souborů	40
6.4	Změna balíčků	40

7	Zobrazení komponent	43
7.1	Zabezpečení	43
7.2	Popis rozhraní	44

Účel a užití standardu

Digitální archiválie uchovávané v digitálních archivech je nutné zpracovávat za účelem jejich budoucího zpřístupnění a využití.

Cílem tohoto standardu je definice komunikačního rozhraní mezi digitálním archivem dle referenčního modelu OAIS a aplikací pro zpracování archiválií. Standard umožňuje archivům zpracovávat archivní informační balíčky (AIP) pomocí k tomu určených aplikací. Pro zpracování balíčků definuje standard metody přístupu k jejich datovému a metadatovému obsahu. Standard umožňuje uložení výsledného archivního popisu zpět do digitálního archivu.

Významným přínosem standardu je oddělení způsobu uložení dat v digitálním archivu a metod jejich zpracování. Je tak umožněn relativně nezávislý rozvoj vlastního digitálního archivu a metod zpracování v něm uložených digitálních archiválií.

1.1 Zkratky

Seznam zkratek použitých ve standardu.

AIP	<i>archivní informační balíček</i> dle referenčního modelu OAIS
CORS	Cross-origin resource sharing
DA	<i>digitální archiv</i>
ESSL	elektronický systém spisové služby
LTP	Long Term Preservation / <i>dlouhodobé uchovávání</i>
NAD	Národní archivní dědictví
NSESSS	Národní standard pro elektronické systémy spisové služby
NDA	<i>Národní digitální archiv</i>
OAIS	<i>Open Archival Information System</i>
PREMIS	standard pro zápis metadat, viz <i>PREMIS</i>
SW	software
UML	Unified Modeling Language
UUID	Universally Unique Identifier
ZP	<i>Základní pravidla pro zpracování archiválií</i>

1.2 Termíny a pojmy

Seznam významných termínů a pojmů používaných v rámci metodiky.

administrativní metadata

metadata pro zajištění *dlouhodobého uchovávání* archiválií a jejich přístupnosti; zahrnují informace o původu archiválií, formě a způsobu jejich uchování; často jsou uložena ve formátu *PREMIS*, *anglický ekvivalent: administrative metadata*

aplikace pro zpracování

viz *software pro zpracování*

archív

paměťová instituce pečující a uchovávající archiválie dle platné legislativy (zákon č. 499/2004 Sb. a další zákony)

archivní informační balíček

metadata a data uložená v *digitálním archivu* v jednom informačním balíčku; je složen z *informačního obsahu* a přidružených informací o uchovávání, *anglický ekvivalent: Archival Information Package* nebo také *AIP*

archivní popis

strukturovaný metadatový záznam k dané archiválii nebo skupině archiválií, v *archivním informačním balíčku* je uložen ve formě *popisných metadat*, *anglický ekvivalent: archival description*

archivní zpracování

třídění, zpracování a popis archiválií

balíček

obvykle je myšlen *archivní informační balíček*

digitální archiv

archiv s oprávněním ukládat *digitální archiválie* dle platné legislativy (zákon č. 499/2004 Sb.)

digitální archiválie

vzniká na základě výběru k trvalému uchování vzhledem k její trvalé hodnotě dané politickým, hospodářským, právním, historickým, kulturním, vědeckým nebo informačním významem; je zařazena do evidence archiválií a je uchovávána v *digitálním archivu*; může mít charakter čistě digitální (*anglický ekvivalent: born-digital*) nebo podobu metadat analogové archiválie, a případně i jejich digitalizátů

dlouhodobé uchování

viz *dlouhodobé uchovávání*

dlouhodobé uchovávání

trvalý proces zajišťující přístupnost a srozumitelnost *informačního obsahu* archiválií po dlouhou dobu; termín dle (ČSN ISO 14721/OAIS), *anglický ekvivalent: long-term preservation* nebo také *LTP*

informační obsah

veškeré informace uložené a *dlouhodobě uchovávané* prostřednictvím *archivního informačního balíčku* uloženého v *digitálním archivu*; informační obsah zahrnuje *digitální archiválie*, informace potřebné pro jejich interpretaci a metadata k nim se vztahující, *anglický ekvivalent: content information*

inherentní archivní popis

archivní popis zpravidla automatizovaně vytvořený na základě *popisných metadat* předaných původcem (*popis od původce*), případně je tento popis obohacen archivářem při převjímce archiválií; vztahuje se vždy výhradně k obsahu samotného *balíčku*

komponenta

počítačový soubor tvořící *digitální archiválii*, je uložena v *archivním informačním balíčku*

kontextuální archivní popis

archivní popis vytvářený při *zpracování* archiválií za účelem zpřístupnění a využití archiválií. Tento popis odpovídá relevantní části archivního popisu v archivní pomůcce a je vytvářen dle metodiky *Základní pravidla pro popis archiválií*

Národní digitální archiv

digitální archiv, jehož provoz zajišťuje Národní archiv dle zákona č. 499/2004 Sb. §15 a dalších

pečující archiv

archiv v jehož péči jsou předmětné archiválie; archiv v rámci péče o archiválie mimo jiné provádí jejich *archivní zpracování*

počítačový soubor

uspořádaná a pojmenovaná sekvence bytů známá operačnímu systému (definice dle standardu *PRE-MIS*), *anglický ekvivalent: file*

popisná metadata

část metadat popisující archiválie, za účelem jejich efektivního vyhledání a identifikace, a usnadňující porozumění jejich obsahu, *anglický ekvivalent: descriptive metadata*; v *archivních informačních balíčcích* jsou rozlišovány tři druhy popisných metadat: *popis od původce*, *inherentní archivní popis*, *kontextuální archivní popis*

popis od původce

strukturované metadatové záznamy, které jsou převzaty od původce dokumentů během výběru archiválií; obsahují informace o dokumentech, které byly vytvořeny nebo přijaty původcem

reprezentace

seskupení *počítačových souborů* tvořících datovou část *balíčku*, které mají obvykle logickou či časovou souvislost; součástí reprezentace mohou být i doplňující metadata, *anglický ekvivalent: representation*

software pro zpracování

specializovaná aplikace umožňující *zpracování* archiválií, při tomto zpracování je vytvářen jejich metadatový popis, tzv. *archivní popis*

technická metadata

metadata, která popisují technické vlastnosti jednotlivých digitálních objektů a *komponent*, jako jsou formát, struktura, rozlišení a další technické aspekty, *anglický ekvivalent: technical metadata*

zpracování

viz *archivní zpracování*

1.3 Standardy a specifikace

Soupis obsahuje související standardy a technické specifikace, které jsou východiskem pro tento standard, jsou jím dále rozvíjeny či jiným způsobem s ním souvisí.

1.3.1 OAIS

Open Archival Information System / Otevřený archivační informační systém.

ISO 14721 / ČSN ISO 14721

Norma je konceptuální model dlouhodobého archivu, který popisuje základní komponenty dlouhodobého archivu, definuje jejich postavení a model archivních balíčků (SIP, AIP, DIP).

1.3.2 NMET1: Výměnný formát digitálních archiválií

Úplný název: *Standard výměnného formátu digitálních archiválií verze 1.0 mezi digitálními archivy v České republice*

1.3.3 METS

Metadata Encoding & Transmission Standard

Verze: 1.12.1

Ke stažení: `mets.xsd`

Odkaz na stránky se specifikací: <http://www.loc.gov/standards/mets/>

1.3.4 PREMIS

Datový slovník PREMIS pro metadata uchovávání je mezinárodní standard pro metadata, který podporuje uchovávání digitálních objektů a zajišťuje jejich dlouhodobou použitelnost. PREMIS byl vyvinut mezinárodním týmem odborníků a je implementován v projektech ochrany digitálních dat po celém světě. Podpora systému PREMIS je začleněna do řady komerčních i open-source nástrojů a systémů pro ochranu digitálních dat. Redakční výbor PREMIS koordinuje revize a implementaci standardu, který se skládá z datového slovníku, schématu XML a podpůrné dokumentace.

Verze: 3.0

Ke stažení: `premis-v3-0.xsd`

Odkaz na stránky se specifikací: <https://www.loc.gov/standards/premis/>

Řízený slovník typů událostí v PREMIS

Typy událostí jsou akce prováděné v rámci úložiště nebo mimo něj, které mají vliv na *dlouhodobé uchovávání* digitálních objektů. Seznam typů akcí a jejich základních definic je předmětem tohoto řízeného slovníku.

Ke stažení: `preservation-events-revision1.pdf`

Odkaz na stránky s typy událostí: <http://id.loc.gov/vocabulary/preservation/eventType.html>

1.3.5 EAD

Výměnný formát pro uložení popisných metadat ve verzi 3.

Profil EAD pro ČR

Způsob uložení popisných metadat v souladu se schválenými pravidly zpracování archiválií.

Odkaz na stránky se specifikací: <https://stands.nacr.cz/ead/>

1.3.6 E-ARK

Soustava specifikací podoby balíčků v digitálním archivu, resp. jejich dílčích variant a částí. Tento standard využívá konkrétní verze specifikací E-ARK dostupných v době jeho přípravy.

Odkaz stránky se specifikací E-ARK: <https://dilcis.eu/>

Common Specification for Information Packages

Společná část specifikace E-ARK.

Verze: 2.2.0 (May 17, 2024)

Ke stažení: PDF

Odkaz na stránku specifikace: <https://dilcis.eu/specifications/common-specification>

Archival Information Package

Specifikace pro AIP dle E-ARK.

Verze: 2.2.0 (May 17, 2024)

Ke stažení: PDF

Odkaz na stránku specifikace: <https://dilcis.eu/specifications/aip>

1.3.7 Základní pravidla pro popis archiválií

Základní pravidla pro zpracování archiválií ver. 3.1

Verze: 3.1 (Metodický pokyn č. 4/2022 odboru archivní správy a spisové služby,)

Ke stažení: PDF

Odkaz na stránku pravidel: <https://stands.nacr.cz/zp>

1.4 Případy užití

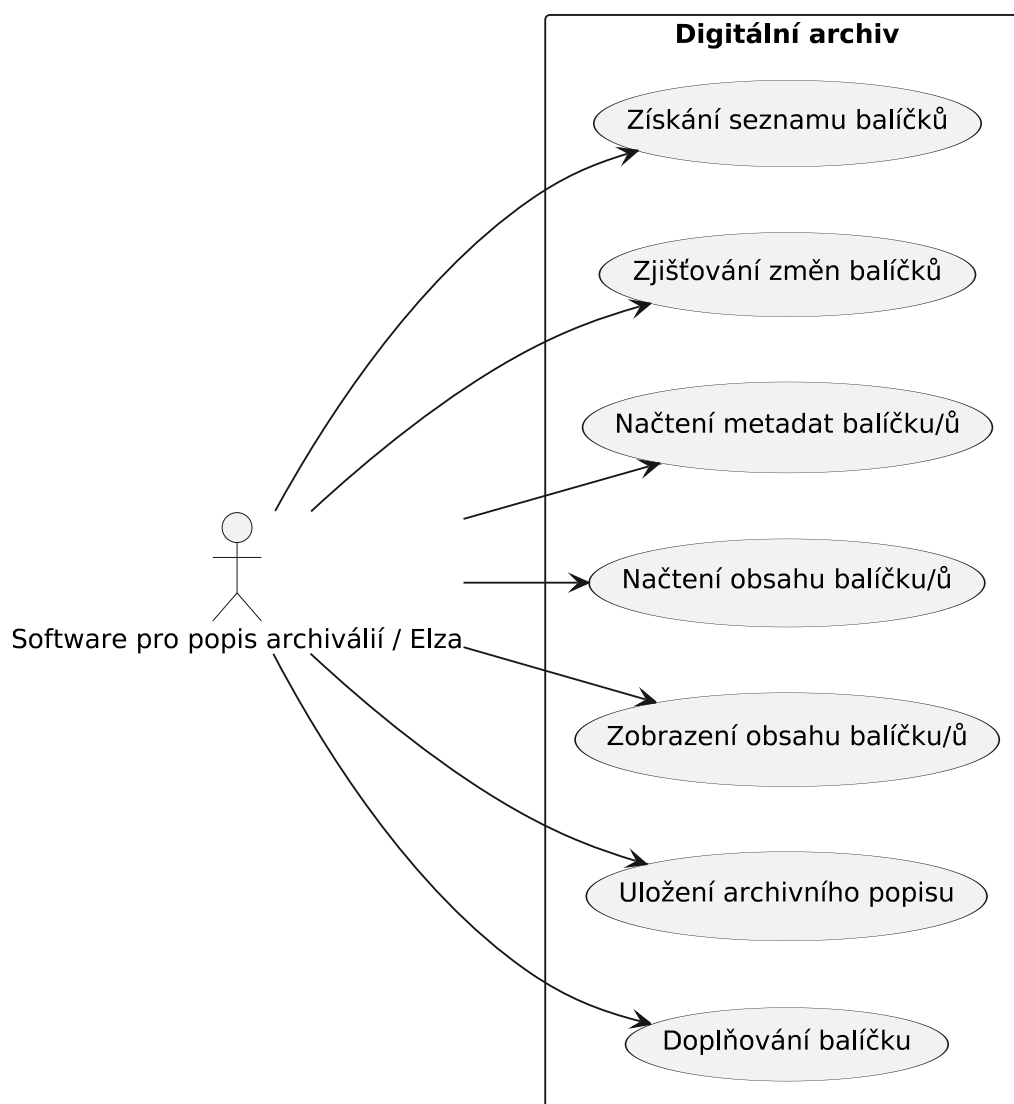
Digitální archiv primárně zajišťuje trvalé uchování obsahu digitálních archiválií a umožňuje k nim přístup. Úlohou *software pro zpracování* archiválií je umožnit archiváři archiválie zpracovat a vytvořit k nim *archivní popis* za účelem jejich zpřístupnění.

Pro naplnění uvedených cílů musí mít software pro zpracování archiválií přístup k archiváliím uloženým v DA, přistupovat k jejich datovému a metadatovému obsahu. Jednotlivé identifikované scénáře jsou zachyceny v následujícím diagramu případů užití (*use case model* dle UML).

Jednotlivé scénáře jsou dále stručně popsány.

1.4.1 Získání seznamu balíčků

Software pro zpracování archiválií musí mít možnost zobrazit seznam relevantních balíčků uložených v digitálním archivu. Archivář má možnost na základě seznamu balíčků vybírat ty, které budou dále zpracovány. Jednotlivý balíček je vždy zařazen do určitého archivního souboru v souladu s evidencí NAD.



Obr. 1: Případy užití

1.4.2 Zjišťování změn balíčků

Do digitálního archivu formou přírůstků v rámci akviziční činnosti archivu přibývají archiválie. Informaci o těchto nových archiváliích, resp. archivních balíčcích je nutné poskytovat do *software pro zpracování*.

V rámci *dlouhodobého uchovávání* a zajišťování formátové ochrany uložených archiválií dochází v digitálním archivu k formátovým migracím, tj. např. převodu ze starších formátů do novějších. Tyto formátové migrace jsou zapisovány do balíčků a nové podoby jednotlivých komponent se tak stávají příomou součástí balíčků. V důsledku těchto změn do balíčku mohou přibývat i celé nové *reprezentace*.

Shodně může docházet k rozšiřování uchovávaných metadat a jejich změnám. Například z důvodu zjištění správné verze formátu *komponenty*, změn v archivním popisu (ať již *inherentního* nebo *kontextuálního*).

Posledním významným příkladem typu změny v balíčku je jeho delimitace do jiného archivu, případně vnitřní skartace. V obou případech se může jednat o úplnou nebo jen částečnou změnu. Pro další zpracování již nemusí být dostupný celý balíček nebo vybrané části.

Výše uvedené příklady vyžadují, aby aplikace pro zpracování mohla zjišťovat změněné balíčky uložené v digitálním archivu a vhodným způsobem poskytovat informaci o jeho aktuálním obsahu.

1.4.3 Načtení metadat balíčku/ů

Za účelem zobrazení souhrnných informací o balíčcích je nutné mít možnost načíst jejich metadata. Na úrovni celého balíčku se jedná například o původce, číslo jednací příslušné přejímky, ID přejímky, typ uloženého obsahu, časový rozsah archiválií a další.

Metadatový popis, obzvlášť tzv. inherentní archivní popis, je významný pro hromadné efektivní zpracování archiválií. Tento inherentní popis pocházející obvykle přímo od původce archiválií umožňuje archiváři porozumění obsahu archiválií bez nutnosti jejich dalšího podrobného zkoumání.

Pro dosažení obou cílů musí digitální archiv poskytovat možnost načtení metadat balíčků v definovaném rozsahu za účelem zpracování archiválií.

1.4.4 Načtení obsahu balíčku/ů

Načtením obsahu balíčku/ů rozumíme možnost poskytování datového obsahu uloženého v rámci balíčku. Důvody pro vyžádání tohoto obsahu mohou být dány potřebami při jejich zpracování, přípravou archiválií pro nahlížení ze strany badatele či jinými formami zpřístupnění. Aplikace pro zpracování tak poskytují archiváři možnost obsah zobrazit a dále analyzovat.

Je možné vyžádat obsah celého balíčku či jen vybraných komponent.

1.4.5 Zobrazení obsahu balíčku/ů

Při zpracování je vhodné mít možnost rychlého nahlédnutí obsahu konkrétní komponenty. Digitální archiv má možnost za tímto účelem poskytovat příslušné rozhraní.

1.4.6 Uložení archivního popisu

Při zpracování archiválií dochází ke vzniku příslušného archivního popisu a tím k jejich zasazení do kontextu. Výsledný archivní popis musí být uložen do digitálního archivu, aby se stal součástí archivního balíčku s popisovanou archiválií.

Digitální archiv musí poskytovat službu umožňující přidání a změnu archivního popisu archiválií v balíčku.

1.4.7 Doplnování balíčku

Doplňováním balíčku rozumíme obvykle přidání dalších komponent v důsledku archivního zpracování. Při doplňování balíčku může být nutné také doplnění a úprava obsahu logické strukturální mapy.

Digitální archiv musí poskytovat rozhraní umožňující doplnění obsahu balíčku.

Poznámka

Speciálním případem doplňování a změny balíčku je odstranění (delimitace) vybraných komponent. Jedná se o zvláštní případ změnového balíčku, jehož definice je nad rámec této specifikace.

Dalšími příklady doplňování balíčku je digitalizace analogových archiválií, formátová migrace a jiné. Definice takovýchto změnových balíčků je nad rámec této specifikace.

1.5 Architektura propojení

Metodika nabízí způsob propojení *digitálního archivu* a *software pro zpracování* archiválií. Technicky se jedná o soustavu komunikačních rozhraní, která jsou popsána v jednotlivých kapitolách (*REST rozhraní*, *Přenosové rozhraní*).

Způsob realizace je ovlivněn vztahem mezi digitálním archivem, rozsahem jeho funkcí, rychlostí zpracování jednotlivých požadavků, typem a rozsahem uložených dat.

Z hlediska architektury jsou uvažovány tři modelové scénáře:

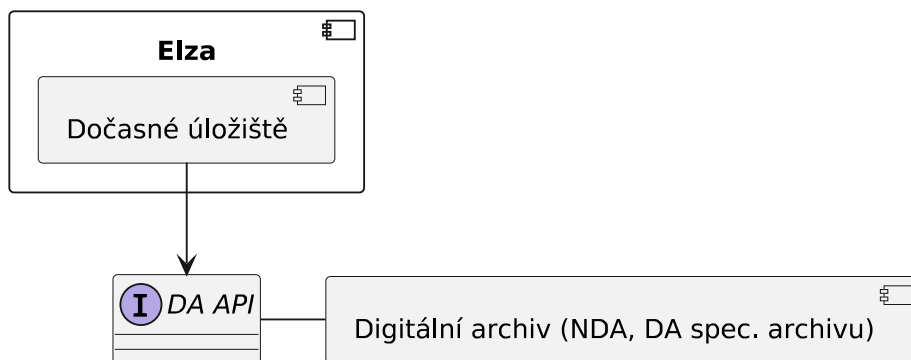
Tabulka 1: Modelové scénáře digitálních archivů

modelový scénář	popis
multiinstitucionální archiv	digitální archiv fungující pro více institucí, předpokládáme delší odezvy a spíše dávkový charakter interakce, příkladem je technické řešení NDA II.
specializovaný archiv	digitální archiv realizovaný v rámci archivu s možností rychlých odpovědí na dotazy, tj. s možností středně až velmi rychlého vyžádání konkrétních souborů Data v cache budou spíše menšího rozsahu.
integrované řešení archivu	aplikace pro zpracování současně plní funkce modulu <i>Správy dat</i> dle modelu OAIS, tj. aplikace je přímo neoddělitelnou součástí řešení digitálního archivu. Data v cache by měla být zcela minimální až žádná. Je možný přímý přístup k potřebným datům.

1.5.1 Vztah mezi aplikačními komponentami

Součástí *digitálního archivu* (či jeho vnitřního úložiště) musí být realizace rozhraní dle této metodiky. Rozhraní musí poskytovat všechny nezbytné metody pro získávání informačních balíčků, včetně jejich změn, a zabezpečit přístup k souborům tvořícím vlastní balíček. *Software pro zpracování* potřebuje pro svou činnost metadata z každého balíčku, a to minimálně v rozsahu metadat o uchovávání a příslušný stávající *archivní popis*.

Předané informace o *archivním informačním balíčku* je nutné uložit v aplikaci pro zpracování po nezbytně nutnou dobu. Jejich rozsah a hloubka již záleží na typu realizovaného scénáře. Ve všech scénářích bude nutná existence vrstvy označované jako *Dočasné úložiště*. Vrstva zajistí uchování předaných metadat a případně předaných *komponent*. Doba uchování záleží jak na technickém řešení aplikace, tak i na metodice práce zpracování digitálních archiválií v daném archivu. Pro daný scénář je možné zvolit i delší dobu jejich uchování. Důvodem pro takové řešení může být přístup k datům v DA s dlouhou latencí, omezené množství dat v lokální databázi a v případě potřeby uživatele schopnost zajištění rychlé odpovědi, zajištění samotné evidence a další.



Obr. 2: Vztah aplikačních komponent

1.6 Právní okolí a souvislosti

Na základě této metodiky je definována v jednotlivých kapitolách sada API funkcí. Tyto funkce mají za úkol umožnit archivářům zpracování archiválií uložených v digitálním archivu a definovat technické metody přístupu pro archiváře pečujícího archivu. Tato metodika blíže neupravuje právní postavení a vztah mezi digitálním archivem, případně jeho provozovatelem, a archivářem přistupujícím k obsahu uloženému v digitálním archivu.

Pro zvažování právních souvislostí je významný vztah mezi archivářem přistupujícím k archiváliím v digitálním archivu, archivem (v němž působí), a provozovatelem digitálního archivu, kde jsou digitální archiválie uchovávány.

Tabulka 2: Vztah mezi archivářem, archivem a digitálním archivem

typ vztahu	popis
vlastní digitální archiv	<i>Archiv</i> je provozovatelem akreditovaného <i>digitálního archivu</i> . Toto je obvyklá situace pro specializované archivy zřizující vlastní digitální archiv.
digitální archiv v jiném archivu	<i>Archiv</i> sám není provozovatelem <i>digitálního archivu</i> , ale na základě smlouvy ukládá <i>digitální archiválie</i> v jiném akreditovaném <i>archivu</i> , který je provozovatelem <i>digitálního archivu</i> . Toto je možná budoucí situace pro specializované <i>archivy</i> .
Národní digitální archiv	Každý archiv, který není provozovatelem digitálního archivu nebo nemá smlouvu s jiným digitálním archivem, musí <i>digitální archiválie</i> ukládat do <i>Národního digitálního archivu</i>

1.6.1 Související právní předpisy

Výběr přímo souvisejících právních předpisů:

Tabulka 3: Přímou související právní předpisy:

předpis	název
Zákon č. 499/2004 Sb.	Zákon o archivnictví a spisové službě a o změně některých zákonů
Zákon č. 121/2000 Sb.	Zákon o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon)
Zákon č. 365/2000 Sb.	Zákon o informačních systémech veřejné správy a o změně některých dalších zákonů
Zákon č. 110/2019 Sb.	Zákon o zpracování osobních údajů
Zákon č. 181/2014 Sb.	Zákon o kybernetické bezpečnosti a o změně souvisejících zákonů (zákon o kybernetické bezpečnosti)
Vyhláška č. 645/2004 Sb.	Vyhláška, kterou se provádějí některá ustanovení zákona o archivnictví a spisové službě a o změně některých zákonů
Vyhláška č. 259/2012 Sb.	Vyhláška o podrobnostech výkonu spisové služby
Vyhláška č. 317/2014 Sb.	Vyhláška o významných informačních systémech a jejich určujících kritériích

Spolupráce mezi digitálním archivem a archivem, do jehož péče archiválie v digitální podobě náleží, je upravena zejména v zákoně č. 499/2004 Sb. § 15 odst. 3 (archiválie v digitální podobě), § 18c (Správa metadat archiválií), § 40 (možnost poskytnutí repliky), § 46 odst. 3 (Národní archiv na úseku péče o ar-

chiválie v digitální podobě), § 60a (Oprávnění k ukládání archiválií v digitální podobě), případně § 60b (Odnětí oprávnění k ukládání archiválií v digitální podobě).

1.6.2 Technické důsledky

U dokumentů uložených v *Národním digitálním archivu* nebo v *digitálním archivu*, který zajišťuje uložení archiválií v digitální podobě pro jiný akreditovaný archiv, je úkolem provozovatele *digitálního archivu* zachování neporušitelnosti jejich obsahu a čitelnosti. Na straně jednotlivých *pečujících archivů* a aplikací přistupujících k obsahu digitálního archivu je nutné přijmout právní a technická opatření, aby byl zajištěn plný soulad řešení s aktuální legislativou, a to zejména s ohledem na ochranu osobních údajů, autorských a jiných práv.

Při *zpracování* digitálních archiválií a přístupu k nim dochází k jejich přenosu do *aplikace pro zpracování*. Tento přenos je považován za dočasný a data v aplikaci pro zpracování by měla být uchovávána po nezbytně nutnou dobu, aby nedocházelo k vytváření dodatečných lokálních úložišť duplikujících funkci trvalého uložení v digitálním archivu.

Princip komunikace

Kapitola popisuje způsob komunikace mezi digitálním archivem a softwarem pro zpracování archiválií. Nejprve jsou definovány jednotlivé případy užití, tj. situace, kdy ke komunikaci dochází.

Následně je popsán technický způsob komunikace s odkazy na jednotlivá komunikační rozhraní a popis jejich funkcí a přenosových možností.

2.1 Způsob komunikace

Komunikaci na základě *případů užití* lze rozdělit na několik hlavních oblastí:

- *informace o balíčcích*
- *přístup k balíčkům a jejich obsahu*
- *zobrazení komponent balíčku*
- *ukládání změn v balíčcích*

Jednotlivé oblasti jsou pokryty k tomu určenou sadou funkcí. Tyto funkce jsou realizovány pomocí dvou typů rozhraní:

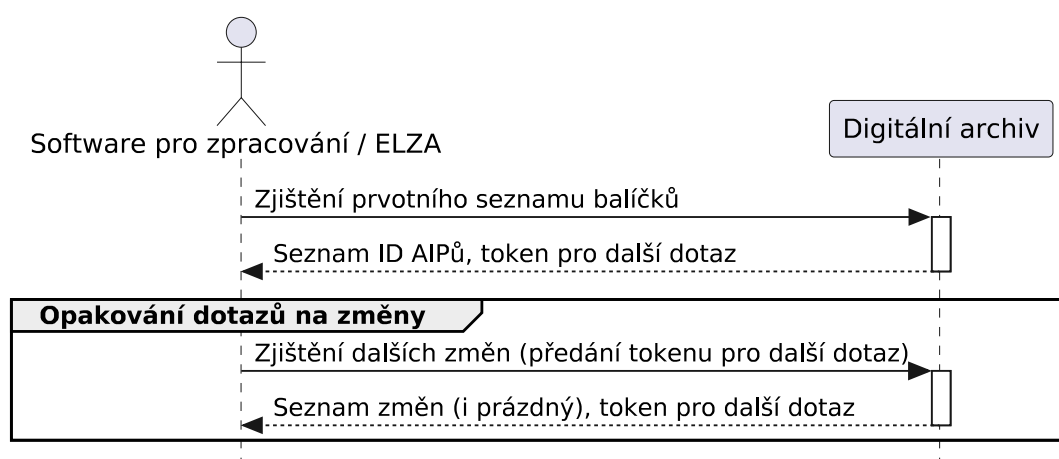
- *Princip a účel REST rozhraní*
- *Princip a účel přenosového rozhraní*

Komunikace u obou typů rozhraní vychází vždy ze strany softwaru pro zpracování archiválií a digitální archiv reaguje na jednotlivé příchozí požadavky.

2.1.1 Informace o balíčcích

Informacemi o *balíčcích* rozumíme předání seznamu dostupných balíčků v *digitálním archivu* do *software pro zpracování* archiválií. Jedná se o balíčky, které mají nebo mohou být pomocí softwaru *zpracovávány* a k nimž má uživatel (technický účet), resp. *archiv* přístup.

Druhou částí tohoto rozhraní je možnost periodického dotazování na změny v balíčcích. Změnami rozumíme nové přírůstky (přidané informační balíčky) a také informační balíčky, v nichž došlo ke změně. Změna balíčku je indikována jeho novou verzí, podrobněji viz *Verze balíčků*.



Obr. 1: Zjištění informací o balíčcích

2.1.2 Přístup k balíčkůům a jejich obsahu

Přístupem k *balíčkůům* a jejich obsahu je chápána sada funkcí, které umožňují zjistit souhrnné informace o balíčku. Těmi jsou například informace o dataci uložených archiválií, příslušnost k archivnímu souboru, identifikace přejímky a další.

Následně je možné začít zpracovávat jeden nebo více vybraných balíčků. Za tímto účelem je nutné načíst inherentní popisná metadata, zjistit jaké *komponenty* jsou uloženy v archivním balíčku a jakým způsobem jsou popsány v archivním popisu.

Archivář v závislosti na povaze uložených dat v balíčku má možnost vyžádat nejen metadata balíčku, ale i obsah vybraných nebo všech datových komponent. Vyžádané komponenty lze dále prohlížet a na základě jejich obsahu archiválie zpracovávat.

Následující diagramy zobrazují uvedené tři dílčí funkcionality:

Zjištění souhrnných metadat balíčku

Cílem funkcionality je získat pro jeden nebo více balíčků soubor `PACKAGE-INFO.xml`. Ten obsahuje souhrnná metadata balíčku.

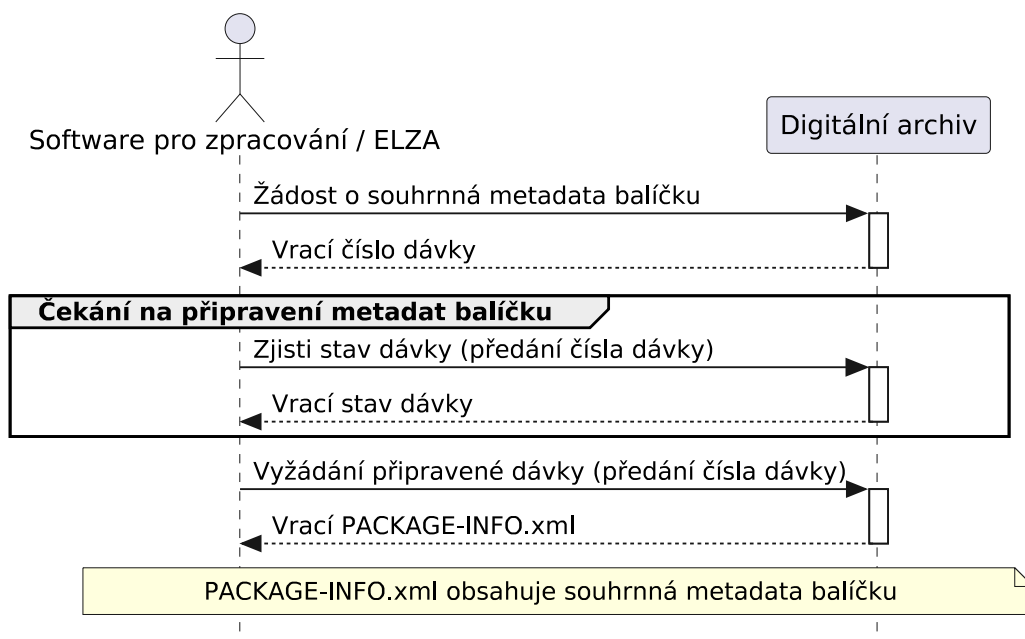
Zjištění popisných metadat balíčku

Cílem funkcionality je získat pro jeden nebo více balíčků jejich metadata, a to minimálně v rozsahu: `METS.xml`, popisná metadata (EAD), metadata o uchovávání (PREMIS). Na základě znalosti těchto metadat je možné zobrazit náhled na obsah balíčku.

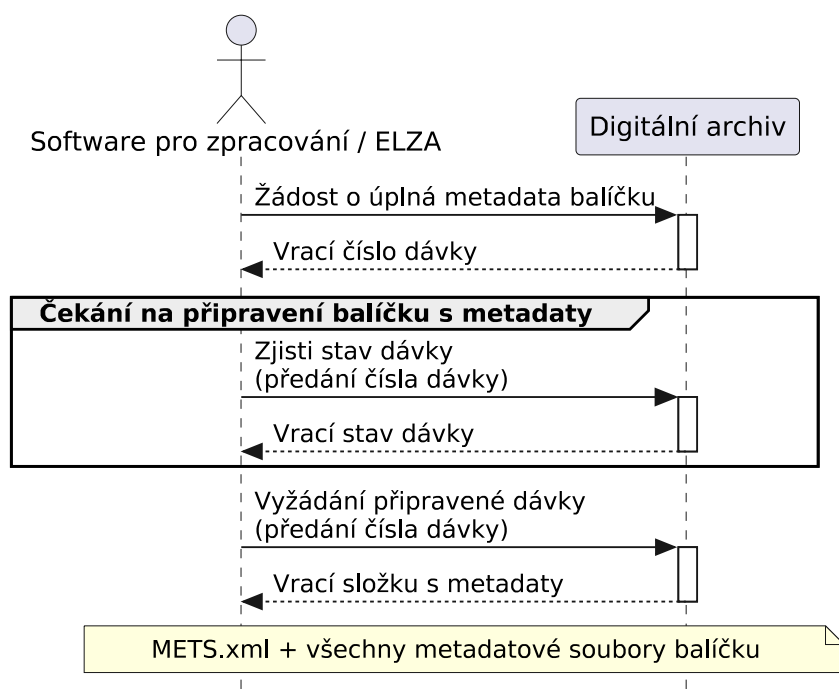
Stahování datových komponent balíčku

V uživatelském rozhraní je možné vyžádat vybranou (vybrané) *komponenty* pro daný *balíček*, případně vyžádat sadu balíčků se všemi komponentami.

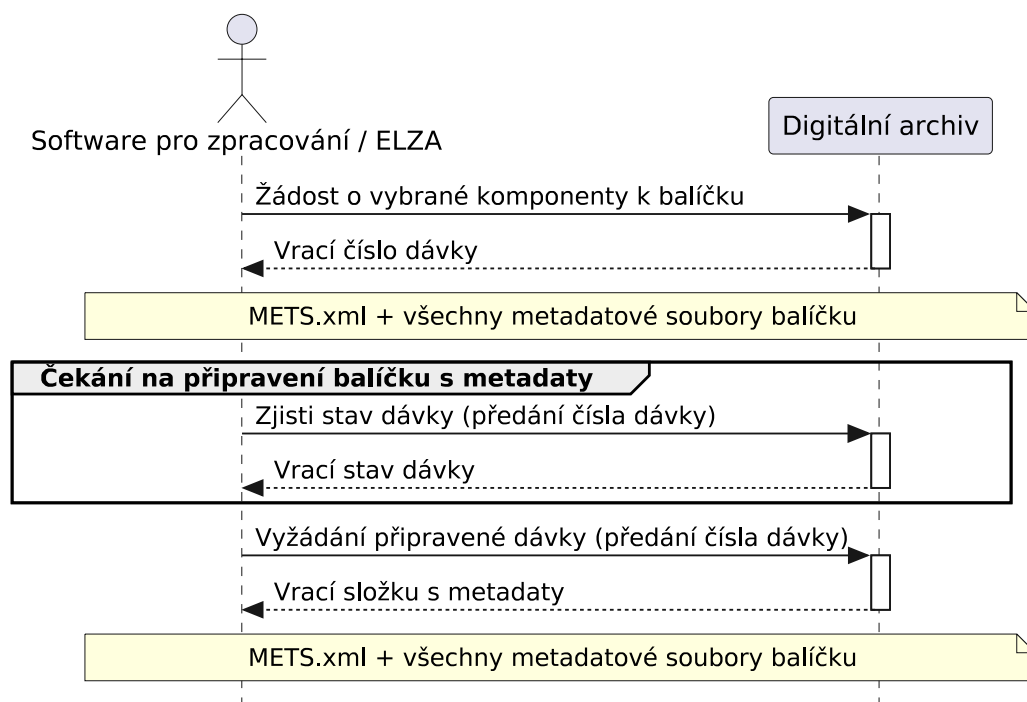
Pro nahlížení obsahu datových komponent je určeno samostatné API, viz *Zobrazení komponent balíčku*. Stažení je primárně určeno pro možnost přímého zkoumání komponent, nikoli jejich nahlížení.



Obr. 2: Zjištění souhrnných metadat balíčku



Obr. 3: Zjištění popisných metadat balíčku



Obr. 4: Stažení datových komponent balíčku

2.1.3 Zobrazení komponent balíčku

Za účelem *zpracování* archiválií je nutné mít k dispozici možnost zobrazení vybrané *komponenty* formou náhledu (např. PDF dokument). *Digitální archiv* může přímo poskytovat funkce pro nahlížení obsahu vybraných komponent bez nutnosti jejich přenosu do aplikace. Jedná se o preferovanou variantu způsobu přístupu ke komponentám oproti *Stažení datových komponent balíčku*.

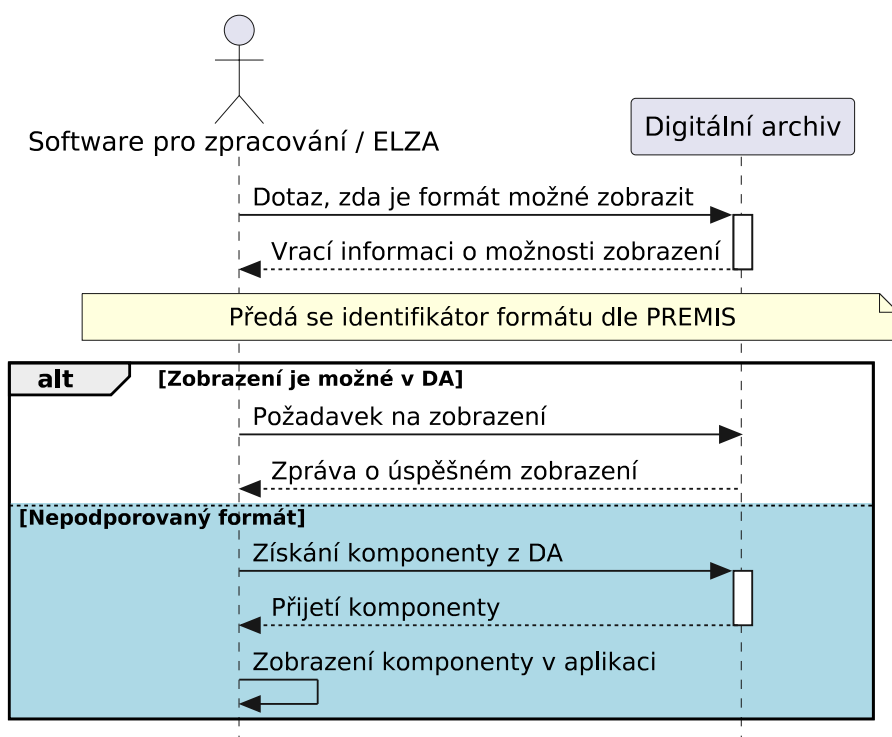
Tímto způsobem je řešeno nahlížení jen vybraných datových formátů a digitální archiv nemá možnost zprostředkovat náhled na všechny typy obsahu. Při zobrazení náhledu komponenty je nutné zjistit, zda je náhled podporován a následně je možné zobrazení. Zobrazení některých komponent může trvat delší dobu, v určitých případech dochází k zpřístupnění až po delší době (například z důvodu aktivace emulace).

2.1.4 Ukládání změn v balíčcích

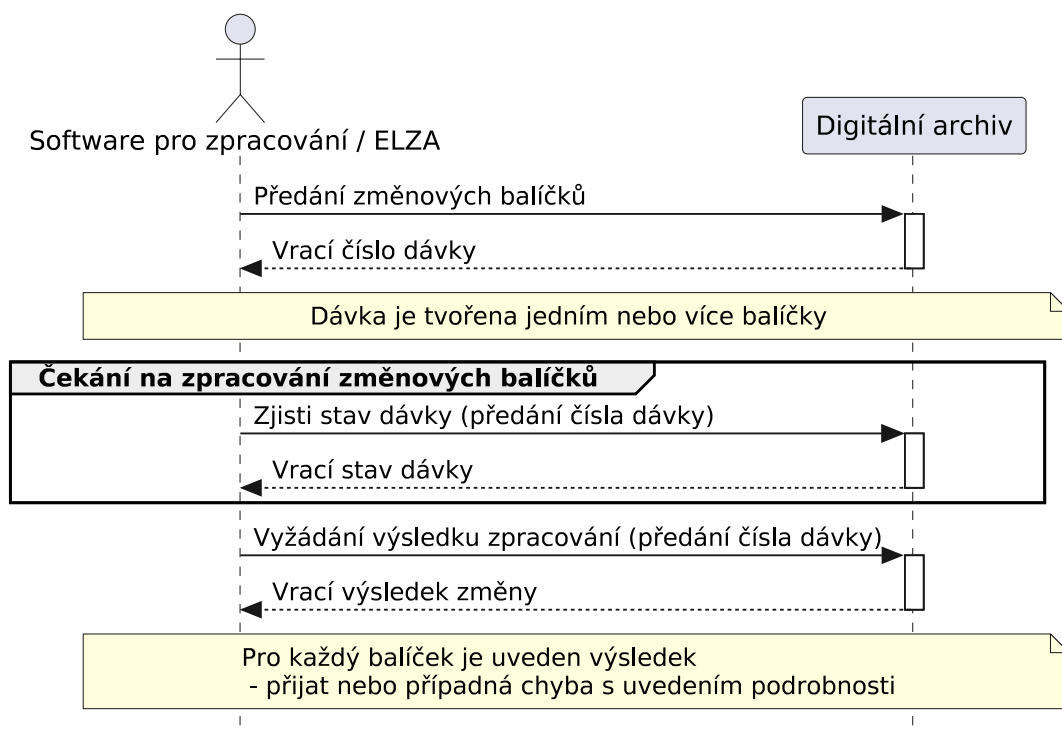
Při *zpracování* archiválií dochází k tvorbě *archivního popisu* a také v některých případech k přidávání komponent (souborů) do *balíčku*. Výsledek zpracování (*kontextuální archivní popis*, případně i přidání *komponenty*) lze zaslat zpět do *digitálního archivu*.

Z pohledu digitálního archivu je příjem změn do balíčku poměrně komplexní proces, kdy změna má obvykle charakter změnového balíčku. Tento musí být řádně zkontrolován a v případě akceptace budou změnové informace zapsány do datového úložiště a do příslušného modulu správy dat.

Balíčky jsou do digitálního archivu předány formou dávky, avšak jejich zápis do dalších procesů obvykle probíhá po jednom balíčku. Předání dávky balíčků do digitálního archivu není obvykle atomickou transakcí, ale jedná se o sadu dílčích požadavků, kdy některé mohou být provedeny a jiné odmítnuty. Důvody odmítnutí je nutné předat zpět do aplikace pro zpracování.



Obr. 5: Zobrazení komponent balíčku



Obr. 6: Ukládání změn v balíčcích

2.2 Princip a účel REST rozhraní

Účelem REST rozhraní je zajištění dobře definovaného technického způsobu komunikace digitálního archivu a aplikace pro zpracování. Rozhraní obsahuje funkce umožňující naplnění a realizaci *případů užití*.

Součástí REST rozhraní je návrh řešení *způsobu autentizace*. Konkrétní implementace se ve způsobu realizace zabezpečení mohou odlišovat, avšak zvolená metoda musí být vždy detailně zdokumentovaná, aby bylo možné její kvalitativní posouzení a současně byla možná její implementace v konkrétní aplikaci.

REST rozhraní se používá pro naplnění všech způsobů interakce s digitálním archivem, konkrétně:

- předávání informací o dostupných balíčcích v digitálním archivě
- notifikace o změnách v uložených balíčcích
- vložení požadavku na poskytnutí balíčků
- vložení požadavku na poskytnutí konkrétních komponent z balíčku
- zjišťování stavu zpracování požadavků
- rychlé stažení menších výsledků
- zjišťování stavu zpracování změn
- získání výsledku zpracování změn

Podrobný popis rozhraní a jeho funkcí je v kapitole *REST rozhraní*.

2.3 Princip a účel přenosového rozhraní

Účelem přenosového rozhraní je zajištění důvěryhodného přenosového kanálu pro získání požadovaných dat z digitálního archivu a možnost tímto kanálem zasílat zpět změny formou *změnových balíčků*.

Přenosové rozhraní je určeno pro přenos větších i menších objemů dat. Po přenesení celé dávky balíčků dojde k jejímu zpracování a přípravě odpovědi. Realizace *přenosového rozhraní* nemusejí být atomické z pohledu provádění transakcí. Například je možné zpracování balíčků v samostatných dílčích transakcích. Důsledkem této možnosti je, že je možné i provedení částečné změny či poskytnutí jen části dat. Z těchto důvodů je nutné při implementaci uvažovat s variantou jen částečného zpracování přenesené dávky. Způsob předávání chyb je vymezen u jednotlivých metod.

Dávkové rozhraní se používá v následujících případech:

- přenos vyžádaných informací o balíčcích
- přenos metadatového a datového balíčku z digitálního archivu
- přenos vybraných souborů (komponent) z digitálního archivu
- přenos změnových balíčků do digitálního archivu

Podrobný popis rozhraní a jeho funkcí je v kapitole *Přenosové rozhraní*.

Přístup k balíčkům

Kapitola popisuje podrobněji způsob a možnosti přístupu k metadatovému a datovému obsahu balíčků. Zejména jsou popsány vybrané aspekty, jako je verzování balíčků, zjišťování souhrnných informací, varianty přístupu k obsahové části balíčků.

3.1 Verze balíčků

Za účelem zjišťování změn v balíčcích je nutné mít možnost identifikovat konkrétní datovou podobu balíčku, tj. verzi balíčku. Každá verze je identifikována pomocí řetězce. Pro tento řetězec reprezentující konkrétní verzi se neuvažuje z pohledu tohoto protokolu jakákoliv sémantická interpretace. Pokud je verze balíčku v digitálním balíčku odlišná od verze balíčku v softwaru pro zpracování, tak z čísla verze se neodvozují žádné další doplňující informace, vyjma skutečnosti, že v digitálním archivu existuje novější podoba balíčku.

Současně platí, že v případě změny balíčku, která je významná a má dopad na archivní zpracování, MUSÍ digitální archiv zajistit vznik samostatné verze balíčku. Dopad na zpracování mají změny přímo v archivním popisu, v uchovávaných datech balíčku (například přidání nové reprezentace), či relevantní změny v metadatech.

3.1.1 Verze při zjišťování změn

Verze balíčku se uplatňuje při zjišťování změn v balíčcích, a to v několika situacích. Informace o verzi je součástí informací o změně balíčku. Software pro zpracování má možnost posoudit, zda došlo ke změně balíčku od jemu naposledy známé verze. O balíček je nutné žádat, jen pokud se jedná opravdu o novou verzi.

V případě ztráty synchronizace změn mezi digitálním archivem a softwarem pro zpracování je možné vždy vyžádat kompletní seznam balíčků od počátku jejich uložení v digitálním archivu. Díky poskytování čísla verze balíčku je možné efektivně určit, které balíčky jsou odlišné.

3.1.2 Verze při zasílání změn

Součástí zasílaných změn do digitálního archivu (např. aktualizace archivního popisu) je informace o verzi balíčku, s níž pracovala aplikace pro zpracování. Digitální archiv při příjmu změnového balíčku MUSÍ určit, zda změna byla realizována nad poslední platnou podobou balíčku. V případě změny realizované nad starší podobou balíčku je vhodné, aby byla taková změna ze strany digitálního archivu odmítnuta. Uživatel by při archivním zpracování měl pracovat nad aktuální podobou balíčků, nikoli nad jejich zastaralými a neplatnými verzemi. Například v důsledku delimitace by již data v daném balíčku nemusela být dále platná. Dalším příkladem je naopak situace, kdy do balíčku přibudou další reprezentace původních dat, které teprve umožní správnou interpretaci datového obsahu a tím i tvorbu archivního popisu.

3.2 Základní informace o balíčku

Základní informace o balíčku jsou předávány v metadatovém souboru označovaném `PACKAGE-INFO.xml`. Tento soubor je předáván jako součást metadat celého balíčku. V případě požadavku na poskytnutí jen základních informací o balíčcích je předáván pro každý balíček pouze samostatně tento soubor a žádný další.

Samostatně předaný soubor MUSÍ splňovat všechny požadavky kladené na tento soubor v rámci metodiky *NMET1: Výměnný formát digitálních archiválií*. Nad rámec těchto výchozích požadavků MUSÍ obsahovat:

- informaci o celkové velikosti balíčku
- informaci o aktuální verzi balíčku v digitálním archivu

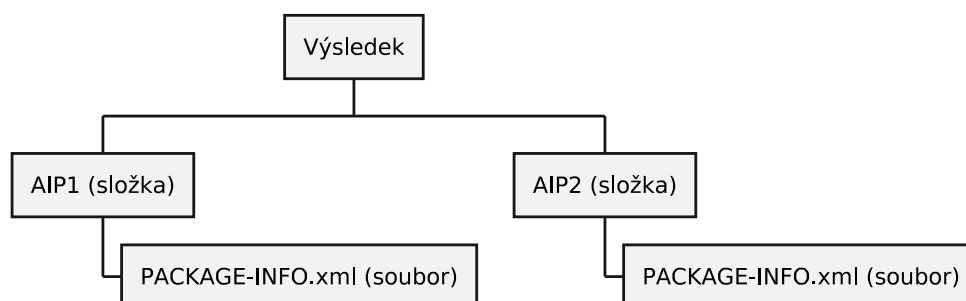
Způsob zápisu těchto informací do datového souboru je popsán v *NMET1: Výměnný formát digitálních archiválií*. Uvedené rozšíření obsahu `PACKAGE-INFO.xml` se uplatní vždy, když je předáván do softwaru pro zpracování, tj. i v případě zpřístupnění obsahu balíčku v podobách: *Archivní metadata* a *Obsah balíčku*.

3.2.1 Způsob získání základních informací

Pro získání základních informací je nutné zaslat příslušný požadavek pomocí *REST rozhraní*, konkrétně pomocí metody `POST /download/aips`. Parametrem volání je seznam vyžádaných balíčků a parametr `dipType="package_info"` určující výslednou podobu dat.

3.2.2 Formát výsledných dat

Pro každý požadovaný balíček je předána vždy jedna složka. Název této složky odpovídá identifikátoru balíčku. Složka obsahuje vždy jeden soubor pojmenovaný `PACKAGE-INFO.xml`.



Obr. 1: Příklad odpovědi pro balíčky s identifikátory AIP1 a AIP2.

3.3 Archivní metadata

V některých případech je postačující poskytnutí jen tzv. archivních metadat. Ta jsou tvořena soubory:

- `PACKAGE-INFO.xml` - obsahuje základní informace o celém balíčku
- `EAD-INHERENT.xml` - inherentní archivní popis
- `EAD-CONTEXT.xml` - kontextuální archivní popis

V případě požadavku na poskytnutí jen archivních metadat jsou pro každý balíček samostatně předány tyto tři uvedené soubory (pokud existují).

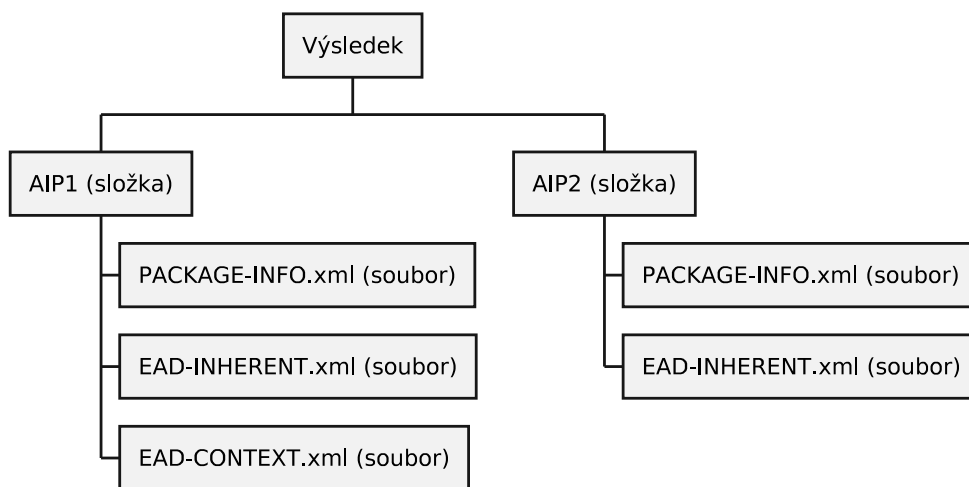
Formát těchto souborů odpovídá jejich definici v *NMET1: Výměnný formát digitálních archiválií*.

3.3.1 Způsob získání archivních metadat

Pro získání archivních metadat je nutné zaslat příslušný požadavek pomocí *REST rozhraní*, konkrétně pomocí metody `POST /download/aips`. Parametrem volání je seznam vyžádaných balíčků a parametr `dipType="archdesc"` určující výslednou podobu.

3.3.2 Formát výsledných dat

Pro každý požadovaný balíček je předána vždy jedna složka. Název této složky odpovídá identifikátoru balíčku. Složka vždy obsahuje soubory s pevnými názvy (pokud existují): `PACKAGE-INFO.xml`, `EAD-INHERENT.xml` a `EAD-CONTEXT.xml`.



Obr. 2: Příklad odpovědi pro balíčky s identifikátory AIP1 a AIP2.

3.4 Obsah balíčku

Balíček v digitálním archivu je tvořen metadaty a vlastním uloženým datovým obsahem. Způsob a podoba uložení metadat i dat v balíčku vždy vychází z konkrétního technického řešení daného digitálního archivu. Digitální archiv však MUSÍ být schopen předávat data ve výměnném formátu v souladu s metodikou *NMET1: Výměnný formát digitálních archiválií*. Software pro zpracování archiválií může využít stejný standard pro přístup k datovému a metadatovému obsahu balíčku.

Za účelem poskytování datového obsahu balíčku jsou definovány čtyři rozsahy poskytovaných dat. V průběhu zpracování archiválií se mohou proměňovat potřeby na rozsah zpřístupněných dat. Digitální archiv

musí umožňovat vyžádání jiného datového rozsahu při zachování platnosti předaných identifikátorů jednotlivých součástí balíčku.

Varianty zpřístupnění obsahu:

označení rozsahu	popis rozsahu
meta-data_base	metadatový DIP obsahující: METS.xml, PACKAGE-INFO.xml, EAD (inherentní, kontextuální), PREMIS (uchování)
aip_base	úplný DIP obsahující metadata_base a navíc binární data komponent
aip_raw	nativní podoba balíčku, podoba je dána implementací digitálního archivu

3.4.1 Metadata

V případě rozsahu metadata_base (metadatový DIP) a aip_base (výměnný DIP - data i metadata) se předávají základní metadata o celém balíčku (PACKAGE-INFO.xml) a všechna metadata v souladu s jejich definicí dle *NMET1: Výměnný formát digitálních archiválií*. V rámci metadat se nepředávají interní metadata digitálního archivu, která slouží pro jeho vnitřní potřebu, či jsou nad rámec uvedené specifikace.

3.4.2 Vyžádání balíčku jen s metadaty

Pro vyžádání balíčku s metadaty je nutné zaslat příslušný požadavek pomocí *REST rozhraní*, konkrétně pomocí metody POST /download/aips. Parametrem volání je seznam vyžádaných balíčků a parametr dipType="metadata_base".

3.4.3 Vyžádání balíčku s daty a metadaty

Pro vyžádání úplného balíčku je nutné zaslat příslušný požadavek pomocí *REST rozhraní*, konkrétně pomocí metody POST /download/aips. Parametrem volání je seznam vyžádaných balíčků a parametr dipType="aip_base".

3.4.4 Formát výsledných dat

Pro každý požadovaný balíček je předána vždy jedna složka. Název této složky odpovídá identifikátoru balíčku. Složka obsahuje buď jen metadatovou část balíčku nebo úplný balíček v souladu s *NMET1: Výměnný formát digitálních archiválií*.

3.5 Soubory a komponenty

V některých případech je vhodné mít možnost při zpracování vyžádat obsah konkrétního datového nebo metadatového souboru, který je součástí balíčku. Příkladem je situace, kdy je v aplikaci pro zpracování k dispozici jen metadatová část a vzniká potřeba prohlédnout konkrétní obsahovou komponentu.

3.5.1 Vyžádání souborů

Za účelem vyžádání datových (či i metadatových) souborů tvořících balíček je součástí *REST rozhraní* funkce POST /download/files. Tato funkce umožňuje pro vybraný balíček stáhnout určené soubory. Jedná se o dávkovou operaci. Na základě předaného požadavku je připravena dávka ke stažení a tu je možné následně přenést. Požadavek obsahuje identifikátor AIPu a seznam identifikátorů požadovaných souborů.

3.5.2 Formát výsledných dat

Pro každý požadovaný soubor je předána vždy jedna složka. Název této složky odpovídá identifikátoru souboru. Složka obsahuje vždy jeden soubor, jeho název by měl odpovídat názvu uvedenému v METS.xml, v části `<filesec>`.

Změnové balíčky

V kapitole je popsán účel změnových balíčků, způsob jejich tvorby, formát změnových balíčků a metody přenosu do digitálního archivu.

4.1 Účel změnových balíčků

V důsledku *archivního zpracování* a zajišťování *dlouhodobého uchovávání* archiválií vzniká potřeba přidávání doplňujících informací a dat do *archivních informačních balíčků*. Důvodů pro doplňování obsahu archivních informačních balíčků existuje více, a to i nad rámec vlastního zpracování informačního obsahu balíčku a tvorbu souvisejícího *archivního popisu*, které jsou primárním předmětem zájmu této specifikace.

Příklady důvodů pro přidávání informací a dat do balíčků jsou:

- ukládání vytvořeného archivního popisu (*kontextuální archivní popis*)
- další změny metadat (např. o uchování, detekce formátů komponent apod.)
- přidávání komponent z důvodu digitalizace
- migrace formátů komponent z důvodu zajištění dlouhodobého uchovávání (LTP), či jejich zpřístupňování
- dělení balíčků na dílčí balíčky
- provádění skartací a delimitací (částečných i úplných).

V této specifikaci je primárně popsána možnost ukládání vytvořeného archivního popisu. Technický návrh je však připraven tak, aby bylo možné realizovat i další uvedené operace.

Přidávání dat a informací do digitálního archivu je navrženo formou tzv. změnových balíčků. Tyto změnové balíčky mají podobu SIP balíčků ve smyslu referenčního modelu *OAIS*. Každý změnový balíček musí projít komponentou pro příjem nových balíčků do digitálního archivu. Ta zajistí provedení příslušných kontrol, ověření příslušnosti navrhovaných rozšíření dat v informačním balíčku, kontrolu oprávnění a následně realizaci úpravy obsahu balíčku a jeho podoby.

Poznámka

Technické rozhraní dále popsané je obecnější a umožňuje kromě změnových balíčků také vkládání zcela nových balíčků.

4.2 Struktura změnových balíčků

Formát změnových balíčků vychází ze specifikace *E-ARK* a jejího českého profilu pro výměnné balíčky dle *NMET1: Výměnný formát digitálních archiválií*.

Ve změnovém balíčku se uvádějí jen soubory, které do balíčku nově přibývají a nezbytně nutná průvodní metadata. Změnový balíček je samostatným SIP balíčkem, a to včetně vlastního identifikátoru balíčku. Každý vkládaný SIP by měl mít unikátní identifikátor. Určení, který AIP balíček je daným změnovým balíčkem doplňován, se provede pomocí metadat uvedených ve vkládaném změnovém SIP balíčku.

4.2.1 Základní podoba balíčku

Základní požadavky na podobu SIP balíčku a jeho identifikátory:

CZDAP-IPF0101: Změnový balíček MUSÍ odpovídat specifikaci výměnného balíčku dle *NMET1: Výměnný formát digitálních archiválií*.

CZDAP-IPF0102: Identifikátor změnového balíčku MUSÍ být řetězec s délkou maximálně 50 znaků.

CZDAP-IPF0103: Identifikátor změnového balíčku BY MĚL mít podobu UUID dle **RFC 9562**¹, verze 4.

CZDAP-IPF0104: Následující pole v kořenovém METS .xml MUSÍ mít konkrétní uvedené hodnoty dle následující tabulky.

Tabulka 1: Hodnoty vybraných polí v METS.xml

Název elementu	Cesta v METS	Hodnota
Content Category czdax-pmt0103	<code>mets/@TYPE</code>	Dataset
Content Information Type Specification czdax-pmt0105	<code>mets/@csip:CONTENTINFORMATIONTYPE</code>	OTHER
Other Content Information Type Specification czdax-pmt0106	<code>mets[@csip:CONTENTINFORMATIONTYPE='OTHER']</code> <code>@csip:OTHERCONTENTINFORMATIONTYPE</code>	change_request_v1_0

4.2.2 Vazba na měněný AIP

Souhrnné informace o SIP balíčku jsou uvedeny v souboru `PACKAGE-INFO.xml`.

CZDAP-IPF0201: Soubor `PACKAGE-INFO.xml` MUSÍ obsahovat:

- definici intelektuální entity zdrojového balíčku, a to včetně uvedení jeho verze
- definici intelektuální entity pro samotný SIP balíček (identifikátor `_THIS`)

¹ <https://datatracker.ietf.org/doc/html/rfc9562.html>

- definici agenta - osoba provádějící změnu
- definici agenta - použitý software
- uvedení události *ing*, resp. *ingestion* dle řízeného slovníku: <https://id.loc.gov/vocabulary/preservation/eventType/ing.html>, zdrojem pro tuto událost je zdrojový balíček

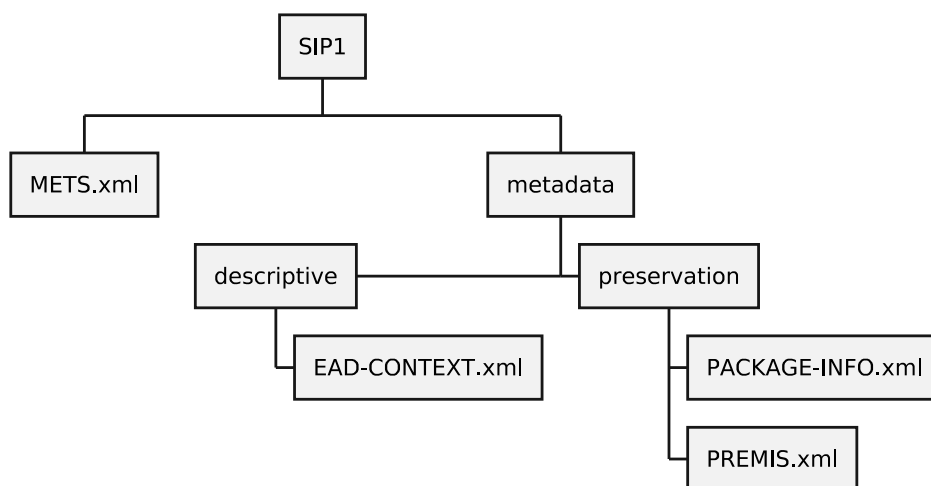
Informace uvedené v souboru `PACKAGE-INFO.xml` jsou zásadní z hlediska připojení změny k požadovanému balíčku.

CZDAP-IPF0202: Pokud by došlo k významné změně balíčku oproti poslední známé verzi balíčku uvedené v souboru `PACKAGE-INFO.xml`, NESMÍ digitální archiv takovou změnu akceptovat, resp. MUSÍ dojít k jejímu odmítnutí s chybou.

4.3 Přidání metadatového souboru

Přidání metadatového souboru do AIPu znamená jeho uvedení ve změnovém AIPu. Pokud je metadatový soubor aktualizací některé předchozí verze, je nutné původní soubor označit jako odstraněný.

Typickým příkladem užití je přidání metadatového souboru s kontextuálním archivním popisem při dokončení zpracování dané skupiny archiválií.



Obr. 1: Příklad struktury změnového balíčku s ID: SIP1.

CZDAP-IPM0001: Nové metadatové soubory MUSÍ být předány ve složce `metadata` a být uvedeny v příslušných sekcích v souboru `METS.xml`.

CZDAP-IPM0002: Vznik každého přidávaného kontextuálního archivního popisu do cílového balíčku MUSÍ být zdokumentován formou události v odpovídajícím souboru ve formátu PREMIS v3. Součástí popisu souboru MUSÍ být uvedení softwaru, pomocí něhož je soubor vytvořen.

4.4 Archivní popis

Pomocí změnových balíčků je možné upravovat kontextuální i inherentní archivní popis. Předmětem této specifikace je možnost změny kontextuálního archivního popisu. Změnu inherentního popisu je možné realizovat obdobně jako kontextuálního archivního popisu, avšak má formu individuálních rozšíření dané implementace digitálního archivu.

CZDAP-IPM0101: Vkládaný archivní popis musí mít formát dle *NMET1: Výměnný formát digitálních archiválií*, a to včetně platných odkazů na archivní balíčky a jejich části.

CZDAP-IPM0102: Vkládaný kontextuální archivní popis MUSÍ plně popisovat obsah archivního balíčku, k němuž se váže. Balíček je považován za plně popsáný, pokud je archivní popis napojen buď na celý balíček, nebo na jeho jednotlivé části tak, že popis zahrnuje všechny datové části balíčku.

CZDAP-IPM0103: Kontextuální popis MŮŽE popisovat jednotlivé části balíčku s různou mírou a hloubkou podrobnosti.

CZDAP-IPM0104: Soubor obsahující vkládaný kontextuální archivní popis BY MĚL mít název `EAD-CONTEXT.xml`.

Je ponecháno na implementaci a rozhodnutí digitálního archivu, zda bude při nahrání nového popisu předchozí verze uloženého archivního popisu ponechána nebo odstraněna.

Rozhraní je tvořeno souborem REST služeb. Ty jsou určeny pro zjišťování balíčků v digitálním archivu, jejich změn a vkládání požadavků na stahování balíčků.

Na toto základní rozhraní navazuje *přenosové rozhraní*, které umožňuje hromadný přenos výsledků z digitálního archivu. Rozhraní také umožňuje zasílat zpět do digitálního archivu změny v archivním popisu, resp. obecněji v celém balíčku.

5.1 Zabezpečení

Funkce REST API vyžadují autentizovaného uživatele. Na základě autentizace je provedena autorizace k provedení dané operace.

Jednotlivé implementace digitálního archivu mohou podporovat více metod zabezpečení přístupu a autentizace. Následují dva příklady způsobu zabezpečení přístupu.

- *Autentizace na Národním archivním portálu*
- *Autentizace pomocí API Key*

5.1.1 Autentizace na Národním archivním portálu

Přístup k digitálnímu archivu zajišťuje Národní archivní portál. Uživatelé Portálu mají přiřazeny role, které určují možnost využívat jednotlivé služby. Pokud se uživatel přihlásí na Portál a má odpovídající roli, tak může volat požadované funkce.

V případě autentizace pomocí Portálu je každý požadavek směřující na digitální archiv, resp. příslušné funkce obohacen o JWT token s informacemi o přihlášeném uživateli.

Specifikace technického řešení zabezpečení Portálu je součástí jeho dokumentace, viz https://frnk.lightcomp.cz/download/nacr/ndais/doc/bezpecnost/portal_jwt.html.

5.1.2 Autentizace pomocí API Key

K autentizaci každého požadavku je možné použít API Key. Jedná se o dvojici identifikátoru klíče a tajné hodnoty klíče. Tajná hodnota slouží pro podepisování požadavků. Každý přijatý požadavek s tímto typem autentizace musí obsahovat potřebné informace ve svých hlavičkách.

API Key je tvořen dvojicí hodnot:

- *identifikátor klíče* - UUID identifikující daný klíč
- *hodnota klíče* - 40 znaků, které tvoří klíč, znaky jsou v rozsahu [0-9][A-Z][a-z]

Ke každému uživatelskému účtu je možné přiřadit více přístupových api klíčů.

Hlavičky požadavku

Zaslaný požadavek musí mít povinně uvedeny dvě hlavičky:

- `Authorization` - hlavička s informacemi o oprávnění
- `X-NDA-Date` - čas vzniku požadavku

Hlavička X-NDA-Date

Hodnota obsahuje čas vzniku požadavku. Uvádí se čas UTC.

Formát: `yyyymmddHHMMSS`

Příklad: `20230915215620`

Hodnota vstupuje do výpočtu hodnoty podpisu (viz *Způsob výpočtu podpisu požadavku*) a čas vzniku požadavku se nesmí příliš odlišovat od přesného času. Maximální povolená odchylka je 2 minuty.

Hlavička Authorization

Hodnota obsahuje typ autentizace a sadu hodnot.

Příklad:

```
NDA-HMAC-SHA256 KeyId=29ca33ec-46bc-402d-b3bd-8d00d387842d,  
→Signature=IT+NIJmiqm2QnB3nW3dqOwCip4z6TBYb9/y9SKv0xTc
```

Jako typ autentizace se uvádí: `NDA-HMAC-SHA256`.

Po typu autentizace za mezerou následují čárkou oddělené hodnoty: `KeyId=...` a `Signature=...`. `KeyId=...` je identifikátor daného klíče, tj. jeho UUID. `Signature=...` je pomocí *base64* zakódovaný podpis požadavku.

Způsob výpočtu podpisu požadavku

Podpis je vygenerován podepsáním řetězce funkcí HMAC-SHA256 a zakódováním podpisu pomocí Base64.

Do výpočtu podpisu požadavku vstupuje:

1. hodnota z hlavičky `host` (cílový server)
2. typ HTTP požadavku (`GET`, `POST`, ...)
3. relativní část URL požadavku (cesta)

4. parametry požadavku uváděné v URL
5. hodnota z hlavičky X-NDA-Date

Části jsou zřetězeny jedna po druhé do řetězce bez oddělovačů. Tento řetězec vstupuje do podepisovací funkce. Pořadí zřetěžení je pevně dané.

Jako token:secret pro funkci HMAC-SHA256 slouží hodnota klíče.

Příklad

API Key:

- KeyId=29ca33ec-46bc-402d-b3bd-8d00d387842d
- hodnota: Pr3fxFN4dB5kMtqdRUzj5lHfJS61eATb5wCqUveb

Dotaz na další změněné balíčky:

- URL: `https://portalvyvoj.nacr.cz/da/updates`
- Metoda: GET
- Čas položení dotazu: 2023-09-15 21:56:20

Řetězec pro podepsání:

HOST	cesta	X-NDA-Date
portalvyvoj.nacr.czGET/da/updates20230915215620		

Výsledek: OPpkPjL19zbGVzN3UOV9ul1TTau5YxGmXbpMzv1Yv5Q

Příklad 2

Dotaz obsahující parametry se podepisuje tak, že se tyto přidávají do podepisovaného řetězce bez oddělovače.

Pro dotaz metodou GET na URL:

<code>https://portalvyvoj.nacr.cz/da/updates-from?pageSize=100& ↪nextQuery=352220190915215620</code>
--

Je podepsaný řetězec:

<code>portalvyvoj.nacr.czGET/cam/api/v1/da/updates-frompageSize=100& ↪nextQuery=352220190915215620</code>

5.2 Popis rozhraní

REST rozhraní je soubor jednotlivých služeb. Tyto služby umožňují:

- zjišťování seznamu balíčků v digitálním archivu
- zjišťování změn v balíčcích
- vkládání požadavků na stažení balíčků
- vkládání požadavků na stažení souborů (komponent) z balíčků

- zjišťování stavu požadavků na stažení
- stahování připravených výsledků
- zjišťování stavu požadavků na změny balíčků
- stahování výsledků požadovaných změn.

Pro přenos výsledků a vkládání požadavků na změny do digitálního archivu je určeno primárně navazující *přenosové rozhraní*.

V samostatné kapitole jsou uvedena *doporučení pro implementaci*.

5.2.1 OpenAPI

Podrobná definice rozhraní je popsána pomocí OpenAPI. Definice obsahuje specifikaci metod a podobu návratových *JSON* objektů.

Specifikace rozhraní ke stažení: `openapi.yaml`

POST /download/aips

Stahování balíčků v poslední verzi.

Metoda umožňuje zahájení stahování jednoho nebo více balíčků v požadované podobě. Stahování balíčků je asynchronní a nejprve je nutné vložit tento požadavek na stažení.

Příprava dat pro stažení může trvat delší dobu a stav přípravy dat je možné zjistit pomocí metody `batchStatus`. Připravenou dávku lze následně stáhnout, a to buď pomocí přímého volání nebo obzvlášť v případě většího objemu dat pomocí `FileTransfer`.

Výsledkem volání je identifikátor dávky, která bude na serveru připravena. Dávka je tvořena vždy sadou adresářů pojmenovaných shodně jako identifikátor daného balíčku. V adresáři jsou umístěny soubory vztahující se k danému balíčku.

Status Codes

- **200 OK²** – Vrací identifikátor dávky ke stažení. Výslednou dávku je možné přenést službou `FileTransfer`.
- **403 Forbidden³** – Chyba stažení balíčku způsobená nedostatečným oprávněním. Případné upřesnění chybějícího oprávnění je uvedeno v chybovém objektu.
- **404 Not Found⁴** – Chyba způsobená uvedením neplatné hodnoty parametru, obvykle chybným uvedením identifikátoru balíčku nebo požadovaným typem balíčku. V detailním popisu chyby jsou uvedeny další podrobnosti.

POST /download/files

Stahování jednoho nebo více souborů z balíčku.

Metoda připraví dávkové stažení jednoho nebo více souborů z jednoho AIPu. Stahování souborů je asynchronní a nejprve je nutné vložit tento požadavek na jejich stažení.

Příprava dat pro stažení může trvat delší dobu a stav přípravy dat je možné zjistit pomocí metody `batchStatus`. Připravenou dávku lze následně stáhnout, a to buď pomocí přímého volání nebo obzvlášť v případě většího objemu dat pomocí `FileTransfer`.

Výsledkem volání je identifikátor dávky, která bude na serveru připravena.

² <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1>

³ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.4>

⁴ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5>

Status Codes

- **200 OK**⁵ – Vrací identifikátor dávky ke stažení. Výslednou dávku je možné přenést službou FileTransfer.
- **403 Forbidden**⁶ – Chyba stažení balíčku způsobená nedostatečným oprávněním. Případné upřesnění chybějícího oprávnění je uvedeno v chybovém objektu.
- **404 Not Found**⁷ – Chyba způsobená uvedením neplatné hodnoty parametru, obvykle chybným uvedením identifikátoru balíčku nebo požadovaným typem balíčku. V detailním popisu chyby jsou uvedeny další podrobnosti.

GET /download/result/{batchId}

Stažení připravené dávky.

Metoda slouží pro stažení připravené dávky. Dávku je možné stahovat, až když metoda getStaus vrátí, že je dávka dokončena (stav FINISHED).

V případě velkého objemu dat nebo i z jiných implementačních důvodů může být nutné výsledek stáhnout pomocí služby FileTransfer, nikoli pomocí této metody. Tento stav je signalizován návratovým kódem 413.

Parameters

- **batchId** (*string*) – Identifikátor dávky, která bude stažena

Status Codes

- **200 OK**⁸ – Dojde k zahájení stahování výsledku. Ten je předán ve formě ZIP souboru, v němž jsou uloženy jednotlivé soubory výsledku.
- **403 Forbidden**⁹ – Chyba stažení balíčku způsobená nedostatečným oprávněním. Případné upřesnění chybějícího oprávnění je uvedeno v chybovém objektu.
- **404 Not Found**¹⁰ – Chyba způsobená uvedením neplatné hodnoty parametru, obvykle chybným uvedením identifikátoru balíčku nebo požadovaným typem balíčku. V detailním popisu chyby jsou uvedeny další podrobnosti.
- **413 Request Entity Too Large**¹¹ – Chyba způsobená příliš velkým datovým objemem odpovědi.

K chybě může dojít při požadavku na přímé stažení balíčku nebo souboru. Pro získání požadovaných dat musí být v takovém případě využito dávkové API, které umožňuje přenos větších objemů dat a navazování přerušeného stahování.

GET /download/status/{batchId}

Zjištění stavu zpracování dávky.

Metoda umožňuje po vložení požadavku, který vrací BatchId, zjistit stav zpracování dávky. Dle stavu lze následně provést stažení výsledku.

Parameters

⁵ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1>

⁶ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.4>

⁷ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5>

⁸ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1>

⁹ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.4>

¹⁰ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5>

¹¹ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.14>

- **batchId** (*string*) – Identifikátor dávky, jejíž stav je zjišťován

Status Codes

- **200 OK**¹² – Vrací informaci o stavu zpracování dávky. Připravený výsledek je možné stáhnout pomocí služby FileTransfer. V případě vrácení chybových stavů 403 a 404 se jedná o trvalé selhání požadavku a jeho příprava je již přerušena.
- **403 Forbidden**¹³ – Chyba stažení balíčku způsobená nedostatečným oprávněním. Případné upřesnění chybějícího oprávnění je uvedeno v chybovém objektu.
- **404 Not Found**¹⁴ – Chyba způsobená uvedením neplatné hodnoty parametru, obvykle chybným uvedením identifikátoru balíčku nebo požadovaným typem balíčku. V detailním popisu chyby jsou uvedeny další podrobnosti.

GET /ingest/result/{batchId}

Převzetí výsledku zpracování dávky

Metoda slouží pro převzetí výsledku zpracování dávky balíčků. Výsledek je možné převzít, až když metoda getStatus vrátí, že je dávka dokončena (stav FINISHED).

Parameters

- **batchId** (*string*) – Identifikátor dávky pro převzetí výsledku

Status Codes

- **200 OK**¹⁵ – Výsledek zpracování jednotlivých předaných balíčků.
- **403 Forbidden**¹⁶ – Chyba vložení způsobená nedostatečným oprávněním. Případné upřesnění chybějícího oprávnění je uvedeno v chybovém objektu.
V případě této chyby nedošlo ke zpracování žádné části z vkládané dávky.
- **404 Not Found**¹⁷ – Chyba způsobená uvedením neplatné podoby vstupní dávky.
V detailním popisu chyby jsou uvedeny další podrobnosti.
V případě této chyby nedošlo ke zpracování žádné části z vkládané dávky.
- **503 Service Unavailable**¹⁸ – Výsledek zpracování dávky není dostupný.

K chybě může dojít v případě předčasného volání metody getResult. Metoda by měla být volána až po dokončení zpracování vstupní dávky, indikovaném pomocí metody getStatus.

GET /ingest/status/{batchId}

Zjištění stavu zpracování požadavku na aktualizaci balíčků.

Metoda umožňuje po vložení změnových balíčků zjistit stav jejich zpracování. Podle zjištěného stavu lze následně provést stažení výsledku.

Parameters

¹² <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1>

¹³ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.4>

¹⁴ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5>

¹⁵ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1>

¹⁶ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.4>

¹⁷ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5>

¹⁸ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.5.4>

- **batchId** (*string*) – Identifikátor dávky, jejíž stav je zjišťován.

Status Codes

- **200 OK**¹⁹ – Vrací informaci o stavu zpracování dávky. Připravený výsledek je možné stáhnout pomocí metody getResult. V případě vrácení chybových stavů 403 a 404 se jedná o trvalé selhání požadavku a jeho další zpracování je přerušeno.
- **403 Forbidden**²⁰ – Chyba vložení způsobená nedostatečným oprávněním. Případné upřesnění chybějícího oprávnění je uvedeno v chybovém objektu.
V případě této chyby nedošlo ke zpracování žádné části z vkládané dávky.
- **404 Not Found**²¹ – Chyba způsobená uvedením neplatné podoby vstupní dávky. V detailním popisu chyby jsou uvedeny další podrobnosti.
V případě této chyby nedošlo ke zpracování žádné části z vkládané dávky.

GET /updates

Zjištění seznamu dostupných balíčků a jejich změn v digitálním archivu.

Metoda umožňuje získat úplný seznam balíčků v digitálním archivu dostupných pro uživatele a sledovat změny v těchto balíčcích.

Povinným parametrem je maximální počet záznamů vrácených v odpovědi. Volitelným parametrem je identifikátor pro zjištění dalších balíčků a jejich změn.

Query Parameters

- **nextQuery** (*string*) – Volitelný identifikátor dalšího dotazu. Jeho hodnota je vrácena jako výsledek předcházejícího dotazu. Při prvním dotazu se parametr neuvádí.
- **pageSize** (*integer*) – Maximální počet vrácených záznamů. (Required)

Status Codes

- **200 OK**²² – Vrací seznam dostupných balíčků. Počet vrácených balíčků je na nejvyšší počet uvedený v parametru pageSize. Server může vrátit nižší počet, než je hodnota pageSize. Pokud nejsou žádné balíčky a jejich změny dostupné, je vrácen prázdný seznam.
- **404 Not Found**²³ – Chyba způsobená uvedením neplatné hodnoty parametru je signalizována návratovým kódem 404 a chybovým objektem s detailním popisem.

5.3 Implementační doporučení

Při implementaci komunikačního rozhraní je vhodné zvážit aspekty popsané v této kapitole. Jedná se o několik doporučení k možnému způsobu implementace.

¹⁹ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1>

²⁰ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.4>

²¹ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5>

²² <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.2.1>

²³ <https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.5>

5.3.1 Dostupné balíčky a zjišťování změn

V *REST rozhraní* je pro zjištění dostupných balíčků v digitálním archivu funkce `GET /updates`. Funkce se volá opakovaně s parametrem `nextQuery` pro zjištění dalších změn. Hodnota parametru je vrácena vždy z předchozího volání uvedené funkce. Jedná se o řetězec a jeho vnitřní struktura a obsah jsou plně na implementujícím digitálním archivu.

Požadavky na parametr `nextQuery`:

- hodnota platná bez časového omezení
- není určen pro interpretaci na klientovi
- musí zaručovat poskytnutí všech změnových informací z DA

Funkce `GET /updates` nepodporuje klasické stránkování. Informace o pozici, pokud je potřebná, musí být zakódována uvnitř parametru `nextQuery`, nebo ji pro daný **iterátor** musí uchovávat DA.

Dále jsou uvedeny příklady možných implementačních přístupů:

Digitální archiv s transakční historií

Předpoklad: V digitálním archivu je dostupná tabulka s transakční historií, tj. při každém zápisu či změně do digitálního archivu vzniká záznam o transakci, který má jednoznačný identifikátor.

Obsah `nextQuery` je potom možné sestavit jako kombinaci hodnot `from:<transakce od>`, `to:<transakce do>`, `pos:<pozice>`. Na základě hodnot *transakce od* a *transakce do* je možné sestavit **uspořádaný** seznam změněných balíčků. Uspořádanost seznamu změn je významná, aby bylo možné vracet informace o změnách postupně a nedocházelo k jejich přeskupování v rámci pořadí. Hodnota *pozice* umožňuje určit, kolik z daného rozsahu změn již bylo předáno, resp. která poslední změna byla předána.

Digitální archiv s notifikacemi o změnách

Předpoklad: V digitálním archivu je při změně balíčku vyvolána událost, na níž je možné reagovat v aplikační komponentě. Způsob předávání těchto notifikací není významný.

Informace o přijatých notifikacích je možné zapisovat do databáze a očíslovat je vzestupnou řadou čísel. Obsah `nextQuery` potom tvoří vždy uvedené pořadové číslo poslední předané změny.

Digitální archiv s iterátory

Předpoklad: Pro každého připojeného klienta bude vytvářen samostatný iterátor. Jeho hodnota bude svázána s účtem pro synchronizaci.

Hodnota `nextQuery` může mít charakter *iterátoru* a jeho hodnota bude nastavována vždy samostatně pro každého připojeného klienta. Způsob implementace iterátoru závisí na použité technologii při implementaci. Iterátor není platný pro více uživatelů, ale bývá samostatně vytvářen pro jednotlivé připojované systémy a jejich účty.

Implementačně se může jednat například o samostatnou tabulku v databázi, do níž jsou zaznamenávány jednotlivé záznamy o změnách. Tyto záznamy čekají na předání danému klientovi. Při příštím vyžádání dalších změn je možné odstranit předchozí již předané záznamy.

5.3.2 Změna rozsahu oprávnění

Na základě příslušného uživatelského oprávnění má uživatel dostupnou sadu balíčků prostřednictvím API uložených v digitálním archivu. V případě změny rozsahu oprávnění by měla existovat možnost, jak získat seznam nově zpřístupněných balíčků.

Příkladem takové situace může být umožnění přístupu k třídě balíčků, které v DA existují, ale doposud k nim uživatel neměl oprávnění. Podstatné je, že při zpřístupnění balíčků na základě změny oprávnění nedochází k jejich změnám a informace o nich nemusí být dostupná formou změn poskytovaných funkcí `GET /updates`.

Změna oprávnění se také může týkat dostupné podoby balíčků.

Přidání oprávnění k balíčkům

Přidání oprávnění k sadě balíčků je možné detekovat a reagovat na něj formou zneplatnění, resp. resetování parametru `nextQuery` používaného ve funkcí `GET /updates`.

V případě *digitálního archivu s transakční historií* je možné do parametru `nextQuery` přidat otisk oprávnění. Při žádosti o změny a detekci změny oprávnění je možné poskytnout úplný seznam balíčků od začátku, tj. shodně jako při nezádání parametru `nextQuery`.

Způsob řešení této situace je nutné zvážit v každé jednotlivé implementaci samostatně.

Zánik oprávnění k balíčkům

Funkce rozhraní umožňují předávat informaci o zániku balíčku jako takového. Pokud však dojde k situaci, kdy balíček již byl předán z digitálního archivu a druhá strana by s ním neměla dále nakládat, je nutné ji o tomto informovat. Klientské implementace MUSÍ umožnit odstranění takovýchto dříve zpřístupněných balíčků. Tento scénář není podporován v rámci API funkcí.

V případě zániku oprávnění ke skupině balíčků je nutné postupovat administrativní cestou.

Přenosové rozhraní

Přenosové rozhraní navazuje na *REST rozhraní* a umožňuje hromadný přenos balíčků a jejich částí z digitálního archivu. Rozhraní také umožňuje zasílat zpět do digitálního archivu změny v archivním popisu, resp. i obecněji v celém balíčku.

Toto rozhraní je založeno na webových službách určených pro bezpečný přenos velkých datových objemů označovaných *FileTransfer*.

Rozhraní využívá shodnou autentizaci se základním rozhraním, viz *Zabezpečení*.

6.1 Zabezpečení

Přenos dávky MUSÍ probíhat pomocí zabezpečeného kanálu a se zajištěním autentizace a autorizace jednotlivých požadavků. Autorizaci požadavků zajišťuje příslušný digitální archiv, který poskytuje přístup ke svým službám.

Pro autentizaci každého požadavku MŮŽE být použita technika podepisování těchto požadavků, a to shodně se zabezpečením REST API dle specifikace *Zabezpečení*. Jedná se o výchozí doporučení pro implementaci. Variantně MŮŽE být vyžadována například autentizace pomocí certifikátu, případně jiných ekvivalentních způsobů.

6.2 FileTransfer

Služba FileTransfer slouží pro důvěryhodný přenos adresářů a v nich uložených souborů. Služba je určena také pro přenos velkých objemů dat. Je možné, aby přenášený adresář/balíček obsahoval hlubší a košatější hierarchii. Stejně tak je možné pomocí služby efektivně přenášet větší množství menších souborů nebo i jednotlivé velké soubory.

Technická specifikace služby a její referenční implementace je dostupná na adrese <https://lightcomp.github.io/filetransfer>.

Služba je využívána ve variantě WSDL rozhraní:

- XSD schéma

- WSDL rozhraní

6.3 Stažení balíčků a souborů

Pomocí funkcí `POST /download/aips` a `POST /download/files` v *REST rozhraní* je možné vytvářet požadavky na přenos dat z digitálního archivu do návazné aplikace. Následně dochází k ověření stavu připravenosti dávky pomocí funkce `GET /download/status/{batchId}`. Pokud je dávka s výsledky připravena, může dojít k jejímu přenosu pomocí služby *FileTransfer* do aplikace pro zpracování archiválií.

6.3.1 Parametry přenosu

Parametry přenosu se rozumí údaje, které jsou zapotřebí pro jeho zahájení, tj. pro zahájení pomocí metody `Begin`.

- Typ přenosu: `download`
- Doplnující ID: číslo dávky, tj. hodnota `{batchId}`

Pokud je dávka připravena, je možné zahájit její stahování voláním dalších přenosových metod.

6.3.2 Formát dat

Dávkou se vždy rozumí sada adresářů. Způsob jejich pojmenování a obsah je dán původním požadavkem, tj. jeho typem a další parametrizací. Přehled podob stažených dat je v následující tabulce:

Požadavek	Parametry	Formát dat
<code>POST /download/aips</code>	<code>dipType=pe="package_info"</code>	<i>Základní informace o balíčku</i>
<code>POST /download/aips</code>	<code>dipType="archdesc"</code>	<i>Archivní metadata</i>
<code>POST /download/aips</code>	<code>dipType=pe="metadata_base"</code>	<i>Obsah balíčku</i>
<code>POST /download/aips</code>	<code>dipType="aip_base"</code>	<i>Obsah balíčku</i>
<code>POST /download/aips</code>	<code>dipType="aip_raw"</code>	dle specifikace digitálního archivu
<code>POST /download/files</code>	AIP ID, seznam ID souborů	<i>Soubory a komponenty</i>

6.4 Změna balíčků

Změna balíčků (přidání archivního popisu či komponent) je iniciováno pomocí služby *FileTransfer*. *Změnové balíčky* jsou nahrány ve formě dávky do digitálního archivu. Následně dochází k ověření stavu zpracování dávky pomocí funkce `GET /ingest/status/{batchId}`. Pokud je dávka zpracována, může dojít k převzetí výsledku pomocí funkce `GET /ingest/result/{batchId}`.

6.4.1 Parametry přenosu

Parametry přenosu se rozumí údaje, které jsou zapotřebí pro jeho zahájení, tj. pro zahájení pomocí metody `Begin`.

- Typ přenosu: `ingest`
- Volitelné ID: číslo dávky zasílajícího, určeno pro snazší dohledávání případných chyb komunikace

Následně proběhne nahrání dávky změnových SIPů. Na závěr přenosu zasílá klient na server zprávu `FinishRequest`, odpoví na úspěšné dokončení přenosu a jeho potvrzením je zpráva `FinishResponse`, která obsahuje následující údaje potřebné pro další zpracování:

- V atributu `type` je nastavena hodnota `ingest_id`
- V atributu `id` je uveden identifikátor přijaté dávky.

Identifikátor předaný v atributu `id` je možné následně použít pro čtení informací o stavu zpracování a následné stažení výsledku. Použije se jako parametr `batchId` v příslušných voláních.

6.4.2 Formát dat

Nahrávanou dávkou se vždy rozumí sada adresářů. Každý adresář je jednotlivý změnový SIP balíček. Identifikátory těchto balíčků určuje zasílající. Forma změnových balíčků je v části [Změnové balíčky](#).

Identifikátory předaných SIP balíčků je nutné uchovat do doby zpracování odpovědi. Jednotlivé části odpovědi jsou vázány na předané identifikátory vstupních balíčků.

Zobrazení komponent

Za účelem archivního *zpracování* archiválií uchovávaných v *digitálním archivu* je nutné mít možnost nahlížet obsah jednotlivých *komponent* (počítačových souborů) tvořících *balíček*. Digitální archiv by měl poskytovat rozhraní umožňující zobrazení jednotlivé komponenty. Zobrazení je realizováno formou klientského webového rozhraní, kdy je do digitálního archivu zaslán požadavek na zobrazení komponenty a digitální archiv provede její zobrazení v určeném okně.

7.1 Zabezpečení

Funkce pro zobrazení komponent vyžadují autentizovaného uživatele. Na základě autentizace je provedena autorizace k provedení dané operace.

Jednotlivé implementace digitálního archivu mohou podporovat více metod zabezpečení přístupu a autentizace. Následují příklady způsobu zabezpečení přístupu s uvedením výhod jednotlivých přístupů.

Varianty zabezpečení API pro zobrazení komponent:

- *individuální řešení*
- *řešení pomocí API gateway*
- *řešení pomocí API key*

7.1.1 Individuální řešení

Architektura zabezpečení přístupu je odlišná v jednotlivých implementacích digitálních archivů a vychází z požadavků daného řešení. Každé individuální řešení musí pokrývat řešení situace, kdy je zaslán požadavek na zobrazení příslušné komponenty a uživatel není autentizován. V takovém případě dojde k vyvolání autentizačního mechanismu a po jeho úspěšném dokončení pokračuje již započatý požadavek. Dochází k tomu bez nutnosti opětovného zaslání ze strany aplikace pro zpracování archiválií.

Všechna řešení by měla umožňovat také efektivní podporu pro opětovné zobrazení další komponenty, a to tak, aby se uživatel nemusel pro každý jednotlivý požadavek opětovně autentizovat.

7.1.2 Řešení pomocí API gateway

Aplikace pro zpracování archiválií může být součástí vnitřní infrastruktury digitálního archivu. Jedná se o poměrně obvyklé řešení pro *archivy*, které provozují vlastní *digitální archiv*. V těchto případech je součástí řešení obvykle API gateway (nebo jiná technologická obdoba), kde se realizuje autentizace všech dílčích částí řešení digitálního archivu. Jednotlivé komponenty dostávají informaci o platné autentizaci v hlavičce příchozího požadavku. Příklady takovýchto technologií: používání JWT tokenů či OpenID nebo distribuovaná autentizace pomocí SAML, Shibboleth a další.

V uvedených případech není součástí odesílaného požadavku na zobrazení komponenty žádná dodatečná autentizační informace. Vše musí být zajištěno mechanismy na straně API gateway. Viz také: *Autentizace na Národním archivním portálu*.

7.1.3 Řešení pomocí API key

Pro přístup k datům uloženým v rámci digitálního archivu je navržen přístup pomocí specializovaných API klíčů, viz *Autentizace pomocí API Key*. Tento mechanismus je možné využít i pro přístup z klientské aplikace pro zobrazení komponent. Mechanismus API klíčů může být využíván jak pro tzv. sdílené technické účty, tak pro individuální uživatelské účty. V případě využití tohoto mechanismu je nutné na straně klientské aplikace zajistit správné podepsání příslušného požadavku, a to se všemi požadavky z toho plynoucími.

Toto řešení je vhodné, pokud je již na straně digitálního archivu tento mechanismus využit pro autentizaci pro *REST rozhraní*. Je nutné posoudit, zda je možné využívat i sdílené technické účty pro zobrazování, nebo jen individuální. V případě využití sdílených technických účtů je nutné zajistit, aby nemohlo dojít k úniku klíčů, a takovéto klíče nepřenášet do klientské aplikace. Individuální klíče je možné využívat i přímo z klientské aplikace.

7.2 Popis rozhraní

API umožňuje zobrazení komponenty z archivního balíčku v samostatném rámci prohlížeče. API je navrženo pro integraci v prohlížeči formou interakce mezi stránkou z aplikace pro zpracování a stránkou poskytující zobrazení komponent digitálního archivu. Jedná se tak o přímou frontendovou integraci bez samostatného kanálu mezi serverovými (backendovými) částmi aplikací. Výhodou tohoto přístupu je nízká náročnost implementace. Za jistou nevýhodu je možné považovat nemožnost pokročilejších forem integrace, a to například interakce s jiným typem aplikací, než jsou webové aplikace, či hlubší propojení metod autentizace.

Služba pro zobrazení komponenty se inicializuje otevřením rámce (IFRAME) a předáním parametrů (požadavek typu GET). Dále se zasílají zprávy mezi rodičovským oknem a vnořeným rámcem, které umožňují dále řídit zobrazení a případně ho parametrizovat.

Typická interakce pomocí API mezi zpracovávající aplikací (klient) a implementací API na straně digitálního archivu (server):

1. (Klient → Server) Inicializace rámce (GET s parametry)
2. (Server → Klient) Potvrzení o dokončení inicializace, případně předání informace o chybě (oprávnění apod.)
3. (Klient → Server) Zaslání požadavku na zobrazení komponenty
4. (Server → Klient) Informace o aktuálně zobrazené komponentě

Oboustranná komunikace mezi klientem a serverem umožňuje parametrizovat zobrazení, dostávat informace o jeho stavu a vykreslovat informace o aktuálně zobrazované komponentě či případné chybě.

7.2.1 Inicializace rámce

Klient otevře URL pro zobrazení serveru v elementu IFRAME. Jako parametry URL se předává:

- (povinně) zdrojové URL, odkud je server volán, je nutné pro řešení CORS, parametr: `SOURCE_ORIGIN=...`

7.2.2 Společné vlastnosti zpráv

Zprávy jsou zasílány jako JSON objekty. Každá zpráva je určena svým typem v atributu `type`.

Zpráva může být rozšířena o volitelný atribut `messageId` umožňující její identifikaci na straně příjemce a zapsání do logu. Hodnota by měla být typu `string`.

7.2.3 Dokončení inicializace

Směr: Server → Klient

Zaslání zprávy ze zobrazovací aplikace DA (server) do rodičovského rámce (klient) o dokončení inicializace.

- Typ zprávy: `type: ViewInit`

Příklad:

```
{
  "type": "ViewInit"
}
```

7.2.4 Zobrazení komponenty

Směr: Klient → Server

Odeslání identifikátorů komponenty z klientské aplikace do aplikace pro zobrazení. Klient může zprávu zaslat až po obdržení zprávy o *dokončení inicializace*.

Parametry zprávy jsou:

- Typ zprávy: `type: ViewRequest`
- Volitelná identifikace požadavku, vrací se zpět po dokončení editace: `requestId: string`
- Identifikátor balíčku: `daoId: ...`
- Identifikátor komponenty v balíčku: `entityRef:`

Příklad:

```
{
  "type": "ViewRequest",
  "requestId": "itemId=423456",
  "daoId": "de450515-98c3-4710-9d0d-bf227bfa875f",
  "entityRef": "uuid-d68bc931-9d71-4188-8931-c5816099f7bd"
}
```

7.2.5 Zaslání notifikace o zobrazené komponentě

Směr: Server → Klient

Zobrazení komponenty z digitálního archivu může trvat určitou dobu, může skončit úspěšně i chybou. Aby bylo možné uživateli efektivně zobrazovat informaci o aktuálně zobrazeném balíčku, komponentě a jeho kontextu, je vždy zaslána zpět zpráva o aktuálně zobrazené komponentě.

Parametry zprávy jsou:

- Typ zprávy: `type: ActiveView`
- Volitelná identifikace požadavku, vrací se zpět, pokud byla předána s požadavkem na zobrazení: `requestId: string`
- Identifikátor balíčku: `daoId: ...`
- Identifikátor komponenty v balíčku: `entityRef:`

Příklad:

```
{
  "type": "ActiveView",
  "requestId": "itemId=423456",
  "daoId": "de450515-98c3-4710-9d0d-bf227bfa875f",
  "entityRef": "uuid-d68bc931-9d71-4188-8931-c5816099f7bd"
}
```

7.2.6 Zaslání zprávy o chybě

Směr: Server → Klient

Zpráva umožňuje předat informaci o chybě ze strany serveru do klienta. Scénáře pro použití: chybně předaná data z klienta/aplikace a další.

Parametry zprávy jsou:

- Typ zprávy: `type: ViewError`
- Chybová zpráva: `message: string`
- Volitelné upřesnění chyby: `description: string`

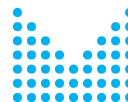
Příklad:

```
{
  "type": "ViewError",
  "message": "Chyba",
  "description": "Popis chyby"
}
```

T A
Č R

Tento projekt je financován se státní podporou
Technologické agentury ČR
v rámci programu BETA2

www.tacr.cz
Výzkum užitečný pro společnost

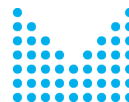


MINISTERSTVO VNITRA
ČESKÉ REPUBLIKY

T A
Č R

Tento projekt je financován se státní podporou
Technologické agentury ČR
v rámci programu BETA2

www.tacr.cz
Výzkum užitečný pro společnost



MINISTERSTVO VNITRA
ČESKÉ REPUBLIKY

LightComp
 **MARBES**