MASARYKOVA
UNIVERZITA

FAKULTA INFORMATIKY

Dokončení procesu vývoje serveru otevřených dat

Diplomová práce

BC. DOMINIK SKÁLA

Brno, podzim 2021

MASARYKOVA
UNIVERZITA

FAKULTA INFORMATIKY

Dokončení procesu vývoje serveru otevřených dat

Diplomová práce

BC. DOMINIK SKÁLA

Vedoucí práce: Ing. Leonard Walletzký, Ph.D.

Laboratoř servisních systémů

Brno, podzim 2021


FACULTAS ARTIS INFORMATICÆ
UNIVERSITAS MASARYKIANA

Prohlášení

Prohlašuji, že tato diplomová práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Bc. Dominik Skála

Vedoucí práce: Ing. Leonard Walletzký, Ph.D.

Poděkování

Na tomto místě bych rád poděkoval Ing. Leonardu Walletzkému, Ph.D., za cenné rady a vedení diplomové práce. Dále děkuji Františku Hánovi ze společnosti Operátor ICT, a.s. za kooperaci při řešení lokálního katalogu a migrační aplikace.

Shrnutí

Diplomová práce Dokončení procesu vývoje serveru otevřených dat se zabývá problematikou otevřených dat v České republice. Cílem práce je dokončit aplikaci lokálního katalogu tak, aby bylo snadné ji distribuovat a instalovat. Dále se zaobírá strukturou doporučení DCAT-AP-CZ, konkrétně verzemi 1.2 a 2.0.1 a jejich kompatibilitou.

V teoretické části práce jsou rozebírána otevřená data v České republice, proces publikace datových sad a jejich distribuce. V rámci distribuce jsou využívány lokální katalogy, pro které byl v kontextu České republiky využíván zejména CKAN. Ten má být nahrazen aplikací lokálního katalogu od společnosti Operátor ICT, a.s.

Praktická část práce se zaobírá analýzou DCAT-AP-CZ, aplikačních rozhraní aplikací CKAN a LKOD od Operátora ICT, a.s. V rámci analýzy je navržen postup pro migraci dat mezi doporučením DCAT-AP-CZ 1.2 a 2.0.1. Ten je pak naimplementován jako webová aplikace. Praktická část tak řeší celý vývoj této aplikace jako služby.

Praktickým výstupem této práce je dokončená aplikace lokálního katalogu s jasným instalačním návodem, postup pro migraci dat mezi jednotlivými verzemi doporučení DCAT-AP-CZ, webová aplikace která tento postup integruje a analýza osob, které takovou aplikaci mohou jednak spravovat a dále pak využívat.

Klíčová slova

LabSeS, Open Data, Open Source, Python, SeSLab, CKAN, LKOD, NKOD, Lokální katalog, Operátor ICT, Otevřená formální norma, datová sada, DCAT-AP-CZ 2.0.1, migrace dat

Obsah

Seznam tabulek

Seznam obrázků

Úvod

Předpokládaná diplomová práce má za úkol dokončit vývoj serveru otevřených dat, jež by měl být dostupný pro Ministerstvo vnitra České republiky, případně pro další instituce v České republice. Hlavní problém, který tato práce adresuje, je dokončení aplikace Lokálního katalogu otevřených dat a zjednodušení přenosu dat ze stávajících aplikací na novou platformu.

Primárním úkolem je tedy dokončení aplikace Lokálního katalogu otevřených dat pro možnost provozu v produkčním prostředí, a také v provozu v prostředí Masarykovy Univerzity. Pro provoz v prostředí Masyrykovy Univerzity je pro Fakultu Informatiky vytvořena instance aplikace. Pro zjednodušení přenosu dat ze stávajících aplikací je provedena analýza aktuální verze a starší verze standardu DCAT-AP-CZ, dále pak analýza aplikací CKAN a LKOD od ICT a na základě těchto analýz byla vystavěna webová migrační aplikace. Práce se dále zaobírá implementací samotného migračního nástroje, nasazením aplikace, využitými technologiemi a ukázkovým spuštěním aplikace. Součástí tvorby samostatného serveru je i tvorba profilu typického správce serveru.

Práce je rozdělena do 9 kapitol, kde úvodní kapitola pojednává o teorii otevřených dat, jejich dostupnosti a zpracování. Druhá kapitola se věnuje teorii datových sad z pohledu českých zákonů, definici hlavních pojmů a Národnímu katalogu otevřených dat. Třetí kapitola řeší datové sady z pohledu aplikačního, popisuje zainteresované strany v projektu, rozebírá jednotlivé aplikace lokálních katalogů a popisuje, co je lokální katalog vyvíjený Operátorem ICT. Čtvrtá kapitola shrnuje stav lokálního katalogu na počátku prací a stanovuje cíl pro dokončení aplikace. Kapitola číslo pět popisuje užité technologie v rámci diplomové práce. Šestá kapitola se již plně věnuje tématu migrace, ve kterém je provedena celková analýza migračního procesu a je navržena migrační aplikace jako služba. Sedmá kapitola popisuje návody na instalaci lokálního katalogu a migrační aplikace, definuje profil osob pro správu takových aplikací. V předposlední kapitole jsou rozebírána možná rozšíření, kterými by šlo navázat na stávající práci. Poté již následuje kapitola závěru, kde jsou zhodnoceny výsledky práce, zda dosáhla stanovených cílů a hodnocení nasazení aplikace.

1 Otevřená data

Otevřená data jsou všude kolem nás, ať už si to uvědomujeme či ne. Jejich vliv na náš život je v dnešní době obrovský, v mnoha případech nám mohou usnadit spoustu práce, v některých ohledech mohou nabídnout možnost tvorby nové služby, možností je nespočetně. [1] Data jsou sbírána pomocí zařízení spadající do IoT[1]. Tato data pomáhají v rozvoji otevřených dat, aplikací využívajících tato data a dále v aplikacích používaných ve Smart City konceptu.

1.1 Motivace

Motivací pro otevírání dat může být mnoho. Už úvod kapitoly částečně možnosti naznačuje. Ať už instituce státní (orgány, ministerstva, úřady), tak soukromé mohou nabízet otevřená data. Proč by to měly dělat? Jedním z důvodu může být vizualizace dat. Příkladem může být například seznam dopravních nehod vizualizovaných v mapě. [2] V mapě je jasně viditelné, na kterých místech se uskutečnila, jaká dopravní nehoda, z takové mapy pak lze relativně jednoduše vyčíst, které úseky jsou náchylnější k dopravním nehodám, takové úseky může odbor dopravy následně zpřehlednit.

Dalším vhodným příkladem může být otevřenost státní správy - již nyní město Brno nabízí otevřenou datovou sadu s hlasováním jednotlivých představitelů města Brna. [3] Vizualizace takového hlasování může dopomoci ukázat aktivitu jednotlivých představitelů a může pomoci rozhodnout voliči v jeho volbě. Lze na to pohlížet dvěma způsoby. Je nutné spoléhat na to, že data jsou opravdu správná a nikdo s nimi nemanipuloval. To ale není předmětem této diplomové práce.

Otevřená data lze využívat i pro firemní účely. Příkladem může být Ares - což je veřejně přístupná aplikace umožňující získání informací o podnikatelských subjektech. Tuto aplikaci zpravidla integrují účetní programy - ve kterých uživatel zadá většinou jen IČO nebo DIČ subjektu a program si již načte veškeré informace automaticky. Ta-

1. Internet of Things - internet věcí, snaha připojit co nejvíce zařízení k internetu za účelem sběru dat.

ková služba je dnes základem účetních programů a bez ní se program na trhu prakticky neprosadí. Systém Money S3 tuto integraci provedl již v roce 2009 v základním modulu Adresář. [4]

V neposlední řadě je na místě zmínit koronavirovou pandemii. Zde pomáhají data na mnoha frontách - nejpoužívanější oblastí jsou bez pochyb statistika očkování a kontrola dosažení dostatečné proočkovanosti. S otevřenými daty lze jednoduše cílit na konkrétní místa. Příkladem zde může být například lepší cílení očkovací kampaně v oblastech s horší proočkovaností. Druhým faktorem, který v koronavirové pandemii za pomocí otevřených dat sledujeme, je počet osob nakažených virem COVID-19. Sledováním takovýchto dat lze vytvářet predikce budoucího vývoje. Vláda tak čtením a predikcí z otevřených dat může zlepšit vývoj pandemie včasnými opatřeními. [5]

1.2 Otevřená data

Otevřená data můžeme definovat více způsoby. Pro účely této práce jsou však stežejní dvě primárními definice - prvním je zákonný pohled, který definuje česká legislativa. Tím druhým pak všeobecná definice otevřených dat, jež není zcela ucelená a může se lehce měnit.

1.2.1 Otevřená data dle českého zákona

Otevřená data dle §3 odst. 11 zákona č. 106/1999 Sb. o svobodném přístupu k informacím jsou "informace zveřejňované způsobem umožňujícím dálkový přístup v otevřeném a strojově čitelném formátu, jejichž způsob ani účel následného využití není omezen a které jsou evidovány v národním katalogu otevřených dat.". [6]

Ze zákona je patrných několik specifikací - otevřená data musí být dálkově přístupná, tzn. pokud možno online dostupná. Data také musí být otevřená a strojově čitelná - jejich struktura by měla být zpracovatelná nějakým algoritmem. Další podmínkou je, že jejich následné využití není nikterak omezeno a musí být zaregistrovány v národním katalogu otevřených dat, který je popsán v sekci 2.2.

1.2.2 Otevřená data z pohledu technologií

Dalším pohledem je pohled technologický. V tomto ohledu je definice obecnější, za to však jasnější. Publikace Open data and charities pro Nominet Trust říká, že datové sady (tedy data), jsou otevřené, pokud splňují tři kritéria:

- Data jsou dostupná online.
- Data jsou publikována v otevřeném, strojově čitelném, formátu.
- Licenční užití dat dovoluje třetím stranám užití a zpracování.

Dostupnost online je důležitá zejména pro komunikaci různých algoritmů, procesů nebo programů navzájem. Právě provázanost těchto programů a možnost komunikace přes otevřená data je to, co dělá otevřená data tak silným nástrojem. [7] Důležitým aspektem je však i strojová čitelnost a licenční užití. Právě strojově čitelná data v otevřeném formátu umožňují snadnou implementaci při zpracování dat. Nejdůležitějším aspektem v rámci dat je také licenční užití, jejich definicí se zaobírá podsekce 2.1.5.

2 Principy otevírání dat v ČR

2.1 Základní definice

2.1.1 Datová sada

Jakub Klímek z Ministerstva vnitra definuje datovou sadu jako množinu souvisejících údajů zpřístupněných prostřednictvím jedné či více distribucí. [8] Datová sada je tedy struktura obsahující informace o sobě a informace o datech samotných. Informace o datové sadě označujeme jako metadata datové sady. Metadatům datových sad v rámci České republiky se věnuje následující podsekce 2.1.2.

2.1.2 Metadata datové sady

V rámci definice od Jakuba Klímka nalezneme také definici metadat datové sady. Z hlediska české legislativy je definována následovně: "Data popisující datovou sadu, zejména její věcný obsah, časové, územní a další souvislosti.". [8] Metadata datové sady můžeme tedy připodobnit metadatům souborů. Tak jako soubory v počítači mají základní informace jako velikost, formát, datum a čas změny a vytvoření, vlastníka, aj., tak i metadata tedy obsahují základní informace. Těmi jsou například název, popisek, klíčová slova, odkaz na dokumentaci. Metadata se nám hodí zejména pro účely katalogizace. Zpracovávat datové sady samy o sobě a diferencovat na základě jejich obsahu, by bylo příliš náročné a dalo by se říci, až zbytečné. V rámci národního katalogu (blíže je národní katalog popsán v sekci 2.2) jsou navíc přidávány další parametry, které budou pomáhat katalogizaci datových sad a jejich vyhledatelnosti. [9]

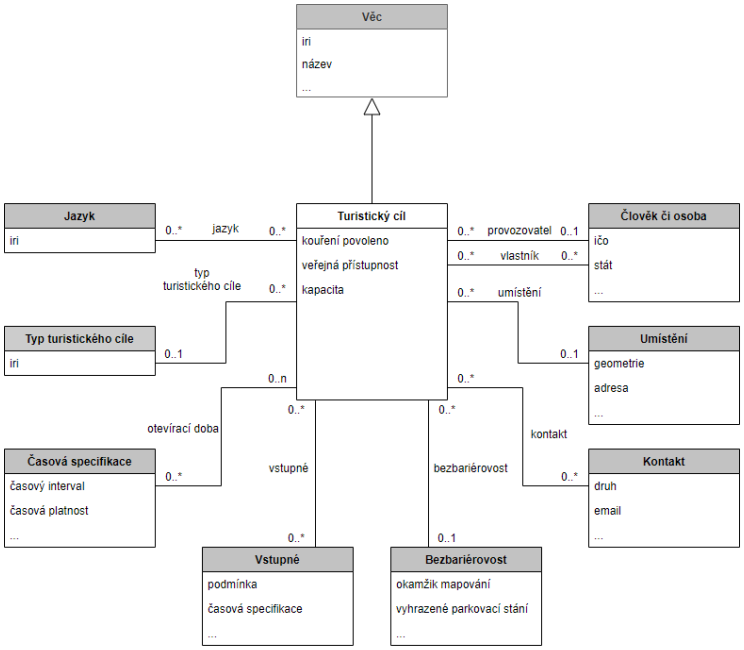
2.1.3 Otevřené formální normy

Součástí metadat datové sady budou od roku 2022 také položky specifikace. Ty obsahují definici otevřené formální normy (pozor, neplést s otevřenou normální formou). [10] Otevřená formální norma (dále jen OFN) je technickou specifikací, která udává, co by datová sada měla obsahovat. Důležité je zmínit, že užití formálních norem je pro

poskytovatele otevřených dat závazné dle zákona č. 106/1999 Sb. o svobodném přístupu k informacím. [6].

Typickým příkladem OFN jsou turistické cíle. Tato OFN je již finalizovaná a bez vážné diskuse a rozmýšlení do ní nebudou přibývat další aktualizace.

Pro definici OFN lze využít třídních diagramů nebo ER diagramů[1]. Na obrázku 2.1 je zaznamenán konceptuální model turistického cíle. Jednotlivé jsou propojeny vazbami s definovanou kardinalitou. Definice OFN navíc umožňuje znovu používat třídy již vytvořené. Na obrázku vidíme třídu: "Vstupné". Tato třída není vytvořena speciálně pro turistický cíl, nýbrž pro všechny OFN, které využívají určitého poplatku za vstup. Každá třída (pro příklad Vstupné) je tak navržena obecně, aby bylo možné ji použít v různých případech užití. Stejně tak je objekt **Vstupné** využit v OFN **Události**. [11]



Obrázek 2.1: Konceptuální model otevřené formální normy Turistického cíle

1. ER diagram - Entity Relation diagram - diagram popisující vztahy mezi objekty a objekty samotné. Využívá se zejména pro modelování relačních databází

2.1.4 Stupeň otevřenosti datové sady

Stupeň otevřenosti je stupnice, která na škále hodnot 1 - 5 definuje do jaké míry jsou využívány standardy HTML pro otevřená data. Vyšší stupeň na stupnici otevřenosti znamená, že do přípravy dat bylo investováno více zdrojů.

1. stupeň Data jsou distribuovaná online a s uvedenými licenčními podmínkami užití. Nemusí zde ale splňovat žádné formáty otevřených dat.

2. stupeň Ve druhém stupni jsou již data strojově čitelná a je jednoduše zpracovatelná programovacími jazyky a volně dostupnými nástroji. V tomto formátu se již jedná o tabulky ve formátu XLS, DOC soubory, tabulky uvtniř HTML dokumentů a další.

3. stupeň Tento stupeň je v kontextu veřejné správy České republiky nejnižším povoleným stupněm otevřenosti. Každá datová sada, která je označena jako otevřená, musí splňovat minimálně tento stupeň. [12]. Otevřená data musí být dodána ve formátu, který je otevřený. Otevřené formáty jsou například formáty typu XML, CSV. V rámci české legislativy sem patří i formát JSON.

4. stupeň Povinně je využíváno IRI[2] v rámci všech distribucí datové sady. Datové sady využívají pro obsah RDF modelů, který může být náročnější na porozumění a zpracování. [13].

5. stupeň Jednotlivé datové sady jsou vzájemně prolinkovány mezi sebou pomocí www odkazů. Je zde nutné kontrolovat stav a dostupnost jednotlivých datových sad a kvalitu prolinkování.

2.1.5 Licence otevřených dat

Tato práce se zaobírá pouze licencemi datových sad v prostředí České republiky. Každá datová sada je souborem dat, která byla někým získána. Při distribuci takové datové sady je vhodné určit omezení použití dat, aby případný uživatel věděl, jak s nimi může nakládat. Licenci

2. IRI - International Resource Indicator - jednoznačený identifkátor objektu (např. Autor)

otevřených dat lze definovat v rámci OFN datové sady pro každou distribuci konkrétní sady. Michal Kubáň dělí licence otevřených sad v kontextu České republiky do čtyř druhů práva. [14]

autorské právo - většina datových sad není chráněna dle §2 autorského zákona (121/2000 Sb.). Většinou se jedná o obyčejná data, která nejsou chráněna ani právy duševního vlastnictví. Může se však stát, že budou obsahovat autorskoprávně chráněná díla.

autorskoprávní ochrana databází jako díla - ochrana je v tomto případě omezena na celé databáze dle §ě odstavce 2 a 5 autorského zákona (121/2000 Sb.). Chráněna je v tomto případě celá struktura databáze - tj. jednotlivé vazby, struktura, řazení. Jedná se o tzv. **originální databázi**.

zvláštní práva pořizovatele databáze - V tomto případě se jedná o dodatečnou práci vloženou do publikace dat. Musí být kvalitativně nebo kvantitativně podstatný, a tedy doložitelný.

právo na ochranu osobních údajů - Neobsahují-li data osobní údaje, pak lze taková data zveřejnit bez další konfigurace. Obsahují-li osobní informace, pak musí být každý příjemce dat o této skutečnosti informován a s daty musí nakládat jako správce osobních údajů. Přechází na něj tedy zodpovědnost za splnění podmínek ochrany osobních údajů.

V rámci národního katalogu, který je popsán v následující sekci 2.2, by měly být vždy vyplněny veškeré hodnoty pro usnadnění užití datové sady při její distribuci. Martin Kubáň popisuje postup pro definici pro stanovení licence ve svém článku: "Stanovení podmínek užití otevřených dat". [14] Zde je doporučováno využití veřejných licencí CC BY 4.0[3] a nebo CC0[4]. Licence datových sad v rámci OFN datové sady je popsána v kapitole 6.

3. CC 4.0 - Creative Commons BY 4.0 - licence dovolující sdílení a úpravu dat za podmínky uvedení autora a licence dat - https://creativecommons.org/licenses/by/4.0/
4. CC0 - Creative Commons Zero - licence bez jakýchkoliv omezení na sdílení, kopírování a úpravy - https://creativecommons.org/publicdomain/zero/1.0/

2.2 NKOD

NKOD - Národní Katalog Otevřených Dat je jednotným katalogem otevřených dat České republiky, do kterého dle zákona musí přispívat všichni poskytovatelé dat, kteří vydávají jakékoliv informace označené jako "Otevřená data"dle §3 odst. 11 zákona č. 106/1999 Sb. Obsahem národního katalogu otevřených dat je tedy seznam všech datových sad, které jsou vydávány jednotlivými poskytovateli dat[5] Národní katalog je projektem Ministerstva vnitra, které má tento portál ve své správě. Důležitou vlastností tohoto portálu je efektivní vyhledávání mezi dostupnými otevřenými daty jednotlivých poskytovatelů. Katalog dále umožňuje filtraci podle jednotlivých poskytovatelů, podle témat datových sad a nebo dle klíčových slov. Jeho další funkcionalitou je dostupnost aplikačního rozhraní pomocí technologií SPARQL a GraphQL.

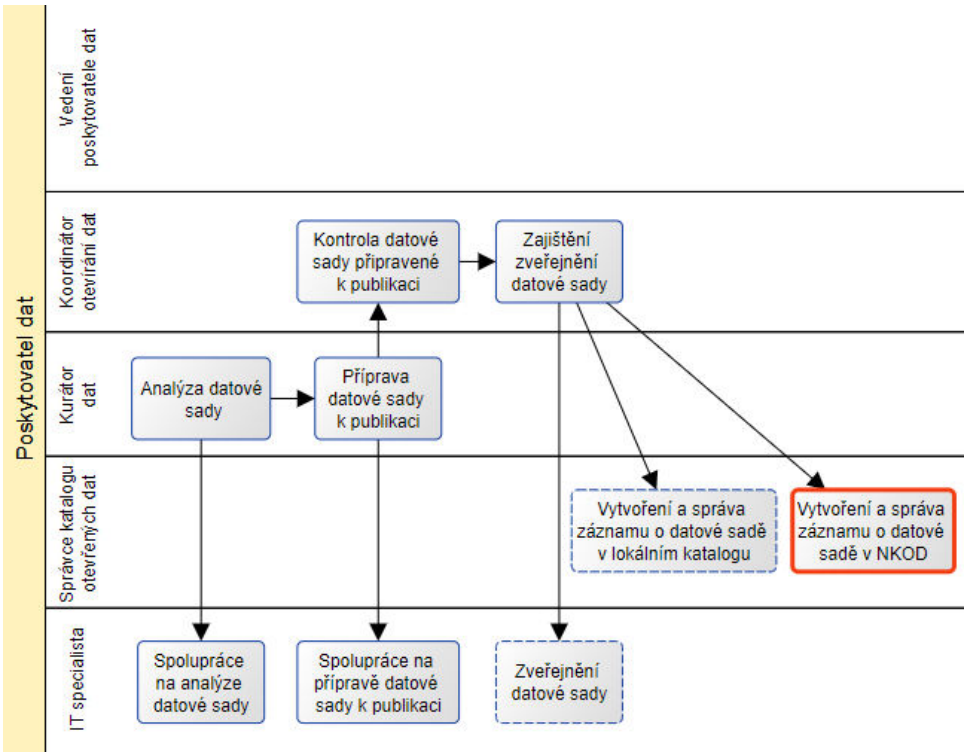
2.2.1 Zákonné povinnosti

Jak bylo zmíněno v předchozí sekci, do národního katalogu mají povinnost přispívat všichni poskytovatelé dat vydávající "otevřená data". Tím ale povinnosti nekončí. Působnost katalogu se od roku 2022 a od roku 2023 značně posiluje. [15] Od 1.2.2022 je povinností v rámci otevřených dat sdílet informace, které budou vyvěšeny v prostoru tzv. "Veřejné desky". [16] Druhým důležitým bodem úpravy je zejména aktualizace přístupu k otevřeným datům. Od stejného data by všechny registry, evidence a seznamy měly být zveřejňované s konceptem: "open by default". To znamená že všechna data, která dle §5 odst. 5 zákona č. 106/1999 Sb., o svobodném přístupu k informacím a nejedná se o zákonem omezenou výjimku (ochrana osobních údajů, bezpečnostní zájem, ochrana obchodního tajemství a další), mohou být zveřejněny. Význam Národního katalogu dat tak dále roste, nejen zásluhou legislativních úprav České republiky, ale i v rámci Evropské legislativy.

5. Správce informačního systému veřejné správy nebo jiný povinný subjekt, který poskytuje otevřená data dle zvláštního právního předpisu.

2.2.2 Registrace datové sady v NKOD

Pro registraci datové sady v NKOD je možné využít jednoho ze dvou hlavních postupů. Lze je označit za manuální a automatickou registraci. [17]



Obrázek 2.2: Registrace datové sady v NKOD

Manuální registrace nastává v situaci, kdy uživatel nemá vlastní lokální katalog. V takovém stavu musí poskytovatel dat registrovat datovou sadu ručně a to následujícím způsobem:

- Správce přistoupí na adresu národního katalogu – https://data.gov.cz/datov%C3%A9-sady.
- Na stránce vyhledá a otevře odkaz "**Registrovat novou datovou sadu**"– https://data.gov.cz/formul%C3%A1%C5%99/registrace-datov%C3%A9-sady.

- Na této stránce vyplní správce informace o distribuci a poskytování datových sad, které mu připravil kurátor[6].

- Úspěšně vyplněný formulář správce stáhne a následně odešle datovou schránkou na adresu: **m3hp53v** s předmětem: **NKOD**.

Proces není příliš jednoduchý a lze v něm udělat spoustu chyb - zejména při vyplňování parametrů datové sady.

Druhým způsobem, který je o dost jednodušší, je postup automatický. V tomto případě správce disponuje lokálním katalogem otevřených dat. V praxi tedy stačí datovou sadu publikovat v rámci lokálního katalogu (musí však dodržovat otevřenou formální normu DCAT-AP-CZ). Pro zaslání datové sady do NKOD je pak potřeba jediné - registrace lokálního katalogu. Registrovat lokální katalog je možné na adrese: https://data.gov.cz/formul%C3%A1%C5%99/registrace-lok%C3%A1ln%C3%ADho-katalogu.

Národní katalog se pak již sám postará o zpracování dat, buď z formátu statických souborů a nebo pomocí SPARQL koncového bodu. V případě SPARQL koncového bodu se národní katalog dotazuje na instance datových sad a na seznam atributů dle specifikace DCAT-AP-CZ a DCAT-AP, které byly popsány v podsekci 6.2.3. Pokud správce dat nemůže nebo nemá možnost distribuce přes SPARQL koncový bod, může využít statického souboru katalogu ve formátu JSON nebo RDF, v němž definuje seznam distribuovatelných datových sad.

2.2.3 Publikační plán

Publikační plán je proces a ucelený postup, podle kterého by mělo docházet k vytváření nových datových sad v rámci instituce. Součástí tohoto procesu je určení rolí jednotlivých stakeholderů a aktérů, určení odpovědností jednotlivých aktérů, výběr datových sad, které budou publikovány, vytvoření harmonogramu podle kterého budou datové sady publikovány, jeho schválení, a vyhodnocení stavu publikace. [18] V rámci tohoto procesu dojde k určení osob kurátora dat a správce lokálního katalogu.

6. Kurátor je osoba, která připravuje data do formátu k otevření. Jeho popis je v podsekci 2.2.4

2.2.4 Kurátor dat

Kurátor dat je osoba zodpovědná za kvalitu datových sad a jejich kontrolu. V procesu tvorby datových sad zodpovídá za analýzu datové sady samotné. Sám navrhuje sady, které mohou být zveřejněny. To činí v rámci vytvoření publikačního plánu. Při zveřejňování datové sady provádí její analýzu a přípravu k publikaci tak, aby odpovídala otevřené formální normě. [19] Publikaci datové sady může provádět správce lokálního katalogu, ale i kurátor dat, pokud k tomu má dostatečné zkušenosti.

3 Aplikační pohled na otevřená data v ČR

3.1 Zainteresované strany

3.1.1 Ministerstvo vnitra

Ministerstvo vnitra je v rámci projektu lokálního katalogu jedním z hlavních stakeholderů. Jakožto tvůrce národního katalogu a specifikace DCAT-AP-CZ určuje směrování otevřených dat v České republice. V rámci tvorby národního katalogu dochází ke tvorbě otevřených formálních norem, které spoluvytváří s ostatními ministerstvy a případnými externími spolupracovníky a nadšenci pro otevřená data. [20]

V projektu národního katalogu zajišťuje provoz služby národního katalogu, zálohování datových sad a jejich metadat. Všeobecně je ministerstvo vnitra gestorem otevřených dat v České republice. [21]

3.1.2 Operátor ICT

Společnost Operátor ICT, a.s. (dále jen OICT) je městskou akciovou společností. Ta zajišťuje zejména pro Prahu, ale nejen pro ni, služby informačních a komunikačních technologií. V gesci má tak koncepce a řešení Smart City, realizaci projektů, poradenství a mimo jiné třeba implementaci lokálního katalogu, který je popsán v podsekci 3.5. Lokální katalog je vyvíjen ve spolupráci s Ministerstvem vnitra.

OICT má ve své správě také Lítačku, zároveň se věnuje tématu otevřených dat - ať už formou projektů covid.praha.eu, který sleduje volné termíny testování, nebo pragozor.cz, kde dochází k agregaci dat a výpisu zajímavých poznatků o městě formou čísel. [22]

3.1.3 Fakulta informatiky

Fakulta informatiky, konkrétně Laboratoř servisních systémů (dále jen SESLAB), dlouhodobě spolupracuje na vývoji konceptu SmartCity, také se věnuje tématice otevřených dat. [1] SESLAB v rámci této práce využije dokončený server lokálního katalogu k dalšímu testování a pomůže tak dalšímu rozvoji otevřených dat v ČR. Rozvoj otevřených dat v České republice se tak přenese na akademickou půdu brněnské univerzity.

3.2 LKOD

LKOD - Lokální katalog otevřených dat je aplikace, která udržuje veškerá otevřená data, která byla publikováná organizací. Takovýto lokální katalog by v kontextu České republiky měl vždy veškeré publikované datové sady sdílet do národního katalogu otevřených dat. Typické aplikace, které můžeme označit jako lokální katalog, jsou například CKAN, DKAN, GeoServer a další. [23] GeoServer a jemu podobné jsou portály, které se zaměřují primárně na sdílení lokalizačních datových sad. [24] Těmito datovými sadami mohou být zejména turistické body nebo body zájmu. Předmětem této práce však není kontext dat, primární zájem je o metadata a distribuce datových sad.

3.3 CKAN

CKAN je aplikací, která až do nedávna byla jednou z hlavních poskytovatelů otevřených dat na české i světové scéně. [23]. Jejím primárním účelem je tvorba a možnost sdílení datových sad v příjemném uživatelském prostředí. Snahou CKANu bylo připodobnit a zjednodušit proces otevírání dat co nejblíže běžné tvorbě stránek. Užití nachází v mnoha institucích, organizacích a také samosprávách. [25]

Tato aplikace pracuje se 3 typy informací:

- datové sady,
- uživatelé,
- organizace / instituce.

.

První a zároveň tím nejdůležitější informací, jsou datové sady. Každá datová sada v CKANu umožňuje definici konkrétních informací o sobě - těmto informacím se říká metadata datové sady a informace o distribuci dat. Tyto pojmy jsou již známy z kapitoly 2.

Z pohledu CKANu mohou metadata obsahovat informace jako název datové sady, periodicitu aktualizace, poznámky, štítky, správce (organizace), časový rozsah platnosti dat a časové známky tvorby a aktualizace datové sady. Blíže je rozbor metadat datových sad CKANu popsán v podsekci 6.2.3. Každá datová sada dále definuje své zdroje,

těmi mohou být jakékoliv soubory typu XML, PDF, CSV a nebo JSON, které obsahují důležité informace - právě konkrétní položky datových sad.

Dalším důležitým prvkem jsou uživatelé. Každý uživatel je vytvořen pod určitou institucí, čímž CKAN umožňuje tvorbu autentizovaných účtů, které spravují pouze konkrétní instituce. Tento přístup je vhodný zejména pro tvorbu jednotných CKAN aplikací, které pod sebou sdružují více organizací/institucí. Typickým příkladem může být třeba Pražský CKAN [26].

Organizace samotné pod sebou sdružují jak uživatele, tak jednotlivé datové sady. Pro každou organizaci existuje seznam uživatelů a seznam datových sad.

3.4 DCAT-AP-CZ

DCAT-AP-CZ[1] je otevřenou formální normou definující tvorbu datových sad platných v rámci České republiky, která je postavena na standardu DCAT-AP. Standard DCAT-AP[2] je evropskou směrnicí postavenou na slovníku DCAT[3]. Aktuální vydání DCAT-AP je ve verzi 2.0.1, ze kterého vychází i české doporučení DCAT-AP-CZ. [27]

DCAT, neboli Data Catalog Vocabulary, je slovník, který vytvořilo W3C konsorcium[4]. Slouží k zaručení vzájemné komunikaci a kompatibilitě mezi různými datovými slovníky. Umožňuje tak snazší práci s datovými sadami, samotnými katalogy. [28]

Poslední verze standardu DCAT-AP je směrnicí vydanou Evropskou unií, která zaručuje kompatibilitu mezi katalogy zveřejněnými v zemích Evropské unie. [27] Na ní právě staví DCAT-AP-CZ celé doporučení pro otevřenou formální normu, která je závazná pro všechna data, která jsou označená jako "otevřená data"v rámci České republiky. Tuto povinnost ukládá zákon §3 odst. 11 zákona č. 106/1999 Sb. o svobodném přístupu k informacím. Doporučení DCAT-AP-CZ definuje

1. Data Catalog Vocabulary Application Profile Czech Republic
2. DCAT-AP - Data Catalog Vocabulary Application Profile
3. DCAT - Data Catalog Vocabulary
4. W3C konsorcium - World Wide Web konsorcium je mezinárodní konsorcium, které vyvíjí webové standardy

formát jak pro souborová data, tak pro SPARQL[5] endpointy. Více je o SPARQL endpointech popsáno v kapitole 6.

3.5 LKOD od Operátora ICT

Tento nástroj je odpovědí na neustále se měnící požadavky formátu otevřených dat. Jedná se o aplikaci kompletně vyvinutou OICT.

Nástroj byl vyvinut jako Open-source alternativa pro běžně dostupné lokální katalogy, mezi které patří výše zmíněný CKAN, DKAN a další. Motivací pro tvorbu nového lokálního katalogu byla nekompatibilita stávajících aplikací s aktuální verzi směrnice DCAT-AP-CZ. Verze 1.2 podporovala CKAN, aktuální verze 2.0.1 podporu CKAN a DKAN aplikací nenabízí.

5. SPARQL je databázový jazyk sloužící k získávání dat ze souborů ve formátech typu RDF

4 Dokončení procesu vývoje serveru otevřených dat

4.1 Stav lokálního katalogu

Úvodní stav projektu, ve kterém se lokální katalog od OICT nacházel v moment započetí této práce byl úctyhodný. Katalog byl schopen vkládat jednotlivé datové sady, umožňoval distribuci pomocí SPARQL endpointů. Je to vše, co je v základu od lokálního katalogu pro snadnou distribuci požadováno. Aplikace však měla několik zásadních nedostatků.

1. Informace o jednotlivých komponentách, jejich užití a nastavení konfiguračních souborů.
2. Chybějící postup pro nasazení aplikace.
3. Neexistující postup pro migraci dat ze zastaralých lokálních katalogů.

Pro aplikaci neexistoval žádný ucelený návod. Bez něj nebylo možné zjistit, jaké všechny komponenty aplikace na pozadí využívá, ať už pro distribuci datových sad a nebo pro cachování. Při instalaci bez jakéhokoliv know-how o konkrétní aplikaci bylo vždy nutné projít zdrojové kódy, pokusit se jednotlivé komponenty spustit a sledovat stav jednotlivých komponent. V případě problému při spuštění jednotlivé komponenty opravit. Takový postup není správně.

Druhý problém, který se v rámci lokálního katalogu vyskytoval byla nedostatečná specifikace jednotlivých komponent a jejich konfigurací. Lokální katalog se skládá z docker kontejnerů, konkrétně z backendové[1] části, frontendové části[2], redis aplikace[3] a SPARQL databáze[4] a PostgreSQL databáze[5]. Jak frontendová, tak backendová část a i sparql databáze mají svá specifika, která je nutná nastavit. Toto by měl adresovat nově vytvořený návod a demo aplikace, která byla vytvořena spolu s OICT.

---

1. backend aplikace vytvořená v JavaScript knihovně Node.js
2. frontend aplikace vytvořený v JavaScriptové knihovně React
3. redis - cachovací aplikace
4. SPARQL - databáze a jazyk pro čtení a úpravu RDF souborů
5. PostgreSQL - aplikace relační databáze postavená na systému SQL

Poslední problém, který aplikace neřeší a ani nemůže řešit, je migrace dat mezi starší verzí doporučení DCAT-AP-CZ 1.2 a verzí 2.0.1. Přesný postup pro přesun mezi doporučeními nebyl definován, krom definice jednotlivých doporučení. Pro tyto účely musí být vytvořena migrační aplikace, která provede konverzi mezi doporučeními a provede konverze jednotlivých atributů do nových formátů.

Došlo ke stanovení cílů, které tato práce musí adresovat. Těmi je tvorba instalačního procesu aplikace, který dopomůže jednoduchému a rychlému nasazení aplikace. Druhým cílem je pomoc při přesunu dat z aplikací, které nejsou doporučením DCAT-AP-CZ 2.0.1 podporovány. Musí tedy proběhnout analýza obou doporučení a musí být navržen postup pro přenos těchto dat.

4.2 Nasazení aplikace lokálního katalogu v prostředí FI MUNI

Lokální katalog od OICT je samostatnou aplikací, jak bylo zmíněno v sekci 3.5. Pro dokončení procesu vývoje bylo nutné instanci aplikace nainstalovat.

V rámci tvorby serveru byly navrženy 3 iterace jednotlivých instalací. Všechny iterace využívají instalace za pomocí docker kontejnerů, účelem tohoto postupu je co nejvíce zjednodušit proces samotné instalace. Aplikace lokálního katalogu je rozdělena na frontendovou a backendovou část. První verze instance lokálního katalogu žádný instalační proces nenabízela a spoléhala pouze na know-how ze strany OICT.

Na základě tohoto know-how byl vytvořen docker-compose soubor, který spouští obě části aplikace, vytváří databázový server. Druhá verze tuto již obsahuje i routovací komponentu Traefik[6]. Aplikace jsou v základní verzi (kterou je třetí iterace) spuštěny s porty, které jednoznačně identifikují konkrétní aplikaci z pohledu síťového prostředí. Vyhnout se tomuto směrování pomocí portů lze právě využitím Traefiku, jejíž užití je zobrazeno v příloze B. Aplikace backendu

6. Traefik - aplikace sloužící k spuštění jednotlivých kontejnerů na vlastních URL adresách. Umožňuje tak definic snadno zapamatovatelných adres namísto užití portů

běží na adrese: http://lkod.fi.muni.cz/lkod-api a frontend na adrese: https://lkod.fi.muni.cz/admin.

Třetí iterace instalačního procesu byla vytvořena na straně OICT za účelem zjednodušit celý proces instalace a nabídnout instalaci i méně zdatným osobám. Z toho důvodu existuje distribuce aplikace s již vygenerovaným **docker-compose.yml** souborem, která spouští jednotlivé aplikace na konkrétních portech. V této verzi je nutné si dokonfigurovat routovací aplikaci. Aplikaci backendu je bez routovací aplikace spuštěna na adrese: http://lkod.fi.muni.cz:3002/, aplikace frontendu oproti tomu na adrese: http://lkod.fi.muni.cz:3001.

5 Technologie

5.1 Docker

Docker je open source aplikací nabízející možnost spouštění aplikací formou tzv. kontejnerů. Kontejner je ucelená sada aplikací a konfigurace. Umožňuje tak jednorázovou konfiguraci aplikace, kterou je možné spustit stejným způsobem na různých zařízeních. Docker tedy nabízí formu zapouzdření, čímž šetří práci nejen programátorovi ale i DevOps týmům. [29]

Nejvíce lze docker připodobnit virtualizovaným strojům[1] s tím rozdílem, že hardwarové parametry jsou přímo sdíleny mezi operačním systémem a kontejnery, což je dělá efektivnějšími než jsou virtuální stroje. [29]

Docker je v rámci implementace využit pro nasazení lokálního katalogu od OICT a pro nasazení a aktualizaci migrační aplikace.

5.2 SPARQL

SPARQL[2] je protokol a dotazovací jazyk nad soubory RDF. Jeho syntaxe je silně podobná jazykům typu SQL, avšak jazyky SQL provádí dotazování nad relačními databázemi, kdežto SPARQL provádí dotazování nad soubory formátu RDF, JSON, relační data, XML a další.

SPARQL databáze je využívána aplikací lokálního katalogu pro publikování datových sad. Takto vypublikovanou datovou sadu, která je vypublikována za pomocí SPARQL, může národní katalog (sekce 2.2) automaticky stáhnout a zpracovat dle doporučení DCAT-AP-CZ. [30]

5.3 Python

Python je open source programovací jazyk, který je tvořen s myšlenkou pro přenositelnost a znovupoužitelnost. Jedná se o jeden z nej-

1. Virtualizovaný stroj je forma zapouzdření celého operačního systému. Takový systém může být nasazen v rámci dalšího operačního systému, od kterého je úplně odstíněn.
2. SPARQL - SPARQL Protocol and RDF Query Language

populárnějších jazyků, ne-li snad nejpopulárnější. Jeho popularita v posledních 15 letech roste. [31] V době tvorby této diplomové práce existuje již verze 3.10, avšak pro účely práce je využívána verze 3.7.

Python je často označován spíše jako skriptovací jazyk než programovací jazyk. To je zejména z důvodu, že jej lze využít v podobném ohledu jako skriptovací jazyk v prostředí shell[3] pro propojení různých aplikací či jazyků k tvorbě nových funkcionalit. [31]

Prakticky nedílnou součástí je dnes balíčkovací systém zvaný pip. Lze jej připodobnit balíčkovacímu apt[4] a nebo Composeru[5]. Jednotlivé balíčky jsou uloženy v repozitáři balíčků (tzv. index balíčků). Základním indexem je pro pip - PyPI. Po instalaci pip balíčkovacího systému lze jednotlivé balíčky doinstalovávat a importovat, pip pak při instalaci balíčku ověří závislosti a lokálně dostupné balíčky, pokud je nenalezne, hledá balíček v jednom z indexů, ať už v základním, nebo v dále definovaných uživatelem.

5.4 Flask

Flask je open source framework[6], který je vytvořen v jazyce Python a instaluje se právě s nástrojem pip. Někdy bývá označován až za microframework - jeho dvěma jedinými částmi je systém směrování (routing) a šablonovací systém.[32] V základu nabízí tvorbu jednoduchých statických stránek, avšak také poskytuje instalaci dodatečných balíčků pro snadnou tvorbu formulářů, úpravu šablon nebo třeba možnost využití databáze. Flask umožňuje stavět relativně jednoduše škálovatelné webové aplikace v relativně krátkém časovém úseku. Dalším důvodem, proč byl vybrán, je právě fakt, že je tvořen v Pythonu a je to tedy další Python komponenta zasazená do prostředí OICT.

Pro snadnější tvorbu aplikace jsou využívána rozšíření ze sady wtforms. Jak název napovídá, ta pomáhají k jednodušší tvorbě formulářů. Dále jsou pak z prostředí flasku využity rozšíření flash, redirect a session pro usnadnění práce při přesměrování, výpisu informač-

3. Uživatelské prostředí / skriptovací jazyk v prostředí systému Unix
4. Balíčkovací systém využívaný v systémech Linux založených na operačním systému Debian - https://packages.debian.org
5. Balíčkovací systém pro PHP - https://getcomposer.org
6. Framework - ucelená sada funkcí a tříd tvořící samostatnou aplikaci. Usnadňuje tvorbu nových aplikací.

ních hlášení a udržení stavu o uživateli. Konkrétní užití jednotlivých rozšíření je popsáno v praktické části v sekci 6.3.

5.5 Další nástroje

5.5.1 PyCharm

Při vývoji migrační aplikace byla využito vývojového prostředí PyCharm Community, které je vytvářeno studiem JetBrains. Společnost nabízí studentské licence, které lze využít právě pro školní projekty.

Hlavní výhodou vývojového prostředí je možnost integrace jak balíčkovacího systému pip, tak virtuálního prostředí, které umožňuje pracovat s nainstalovanými balíčky uvnitř pouze konkrétního projektu.

5.5.2 GitHub

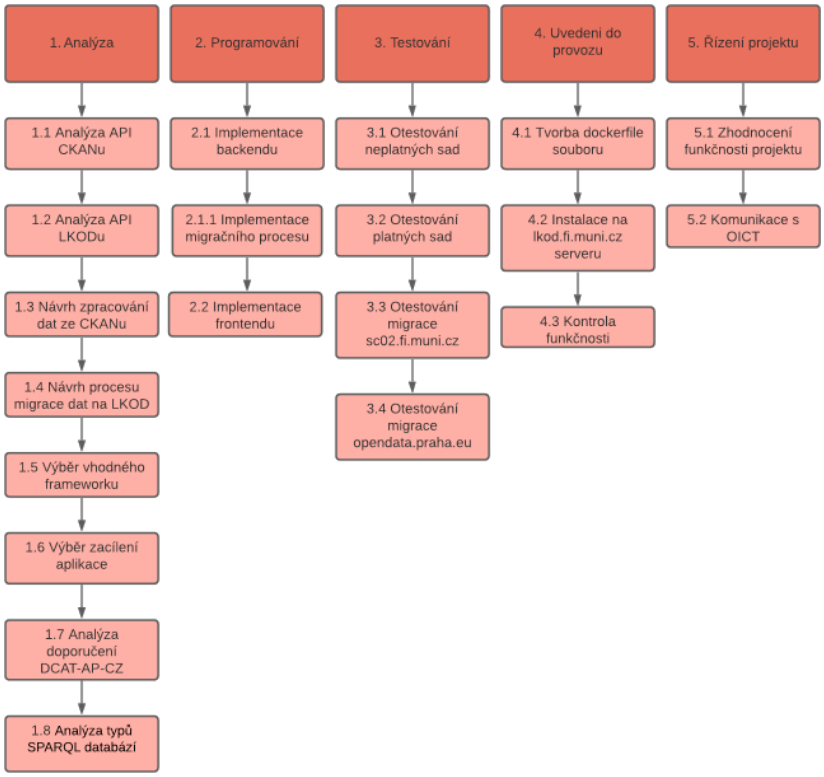
Samozřejmostí je využití verzovacího nástroje. Pro tyto účely vystačil Github. Ten nabízí jak placené, tak zdarma verze, avšak již v základní (zdarma) verzi služba nabízí neomezený počet soukromých i veřejných repozitářů.

6 Migrace dat

Stěžejní oblastí diplomové práce je migrace dat. V tomto případě přesun dat obnáší analýzu celého procesu, programování, testování, uvedení do provozu a projektové řízení. Všechny tyto části jsou zpracovány ve WBS, jejíž struktura je postupně rozebrána v nadcházející sekci.

6.1 Work Breakdown Structure

Work Breakdown Structure (WBS) je rozdělena do pěti sloupců, kde každý sloupec sdružuje témata zmíněná výše. Rozbor jednotlivých témat je proveden v navazujících sekcích.



Obrázek 6.1: Work Breakdown Structure

6.2 Analýza

Primárním cílem pro migraci dat byla snaha o usnadnění přenosu dat do aplikace Lokálního katalogu od OICT. S vývojovým oddělením na straně Operátora ICT jsme došli k závěru, že pro pomoc přesunu dat bude nejdůležitější se soustředit na CKAN. Právě ten je hlavní platformou pro sdílení a distribuci otevřených dat v Praze. Proto bylo logické pro přesun využít a zanalyzovat CKAN.

Při analýze serveru sc02.fi.muni.cz bylo nalezeno 84 datových sad (včetně neveřejných), které lze zmigrovat do serveru LKOD od ICT. Pokud by každou datovou sadu bylo nutné registrovat ručně, pak průměrně zpracování jedné datové sady potrvá 15 minut. To lze převést na 21 hodin čisté práce bez přestávky.

To je pouze jeden CKAN server, kterých je v ČR stovky. Samozřejmě každý CKAN server má jak veřejnou, tak neveřejnou část dat. Aby byla migrace úspěšná, je nutné zmigrovat obě části. Pro příklad lze uvést datový server Prahy[1]. Na tomto portálu existuje 336 veřejných datových sad (a 0 neveřejných - neveřejné datové sady jsou skryté v sekundárním CKANu, který není pro veřejnost dostupný). Přenášet a plnit ručně každou takovou sadu by zabralo více než týden.

Největším benefitem, který má migrační nástroj přinést, je bezpečná a rychlá možnost přenosu datových sad ze zastaralého CKAN serveru.

Dle konzultace s OICT a na základě analýzy potřeb, bylo rozhodnuto, že implementace musí proběhnout v jazyce Python. Hlavní důvody tohoto rozhodnutí byly dva - první - CKAN samotný je implementován v jazyce Python - a v případě, že někdo bude instalovat další aplikaci na server CKANu, je vhodné se vyhnout instalaci dalších komponent[2]. Druhým a tím stěžejnějším důvodem je, že pro OICT bylo jednodušší využívat Python aplikaci, protože dle informací od konzultanta jejich portfolio sestává i z Python aplikací, jde tedy o to udržet jednotný tech-stack.

Tím byl vyřešen bod 1.5 z WBS. Rozhodli jsme se pro využití Flask frameworku, o kterém je více popsáno v sekci 5.4, implementace jednotlivých části je popsána v 6.3.

---

1. opendata.praha.eu - CKAN aplikace Hl. města Praha
2. Komponenta je v tomto případě jakýkoliv balíček, programovací jazyk, kompilátor

6.2.1 Aplikační CKAN API rozhraní

CKAN nabízí aplikační rozhraní, pomocí kterého se lze na jednotlivé datové sady dotazovat. Pro účely migrace jsou dostupné a využitelné dva endpointy. Těmi jsou:

- package_list,
- package_show,
- package_search.

Pro přihlášení k CKAN API je možné využít speciálního přihlašovacího klíče, který uživatel nalezne v administraci CKANu. API je samozřejmě možné používat jak ve veřejném modu, tak v soukromém modu. To znamená, že při vynechání API klíče dojde k načtení pouze veřejných datových sad. V opačném případě se ve výpisu objevují i skryté datové sady. Avšak tato možnost funguje pouze v případě endpointu package_search spolu s query parametrem **include_private**.

Součástí výstupu všech dotazů CKAN API jsou atributy: **result** a **success**. Položka **success** obsahuje boolovské označení stavu dotazu. Pokud byl dotaz úspěšně proveden, výsledkem je: *true*, v opačném případně *false*. Atribut **result** obsahuje samotný výsledek dotazu.

Endpoint package_list, jak už název napovídá, zobrazuje výpis všech datových sad, které daný server registruje. Datové sady jsou označeny unikátním názvem, který musí být použitelný v URL adrese. Tento název se generuje přímo z názvu datové sady.

Datový endpoint package_show požaduje parametr: **id**, který musí obsahovat název datové sady, dle formátu zmíněného výše. Výsledkem jsou pak metadata datové sady, příklad výsledku je možné vidět v příloze A.

Endpoint package_search umožňuje vyhledávání datových sad podobně jako package_list. Rozdílem v tomto případě je možnost doplnit další parametry - maximální počet položek, zobrazení skrytých datových sad a možnost přímo vyhledávat podle názvu datových sad.

Při analýze všech dostupných endpointů pro přístup k datovým sadám vychází nejlépe využití bodu package_search. To je zejména z důvodu možnosti vypisovat i skryté datové sady, které přes endpoint package_list není možné získat.

6.2.2 Analýza LKOD API rozhraní

Aplikační rozhraní LKODu od ICT je oproti CKANU dosti strohé. Celkem nabízí jedenáct endpointů, z toho dva endpointy jsou nachystány pro informování o stavu aplikace, tři endpointy slouží k přihlašování se k aplikaci a zbylé dovolují správu datových sad. V průběhu dokončování této diplomové práce přibylo několik nových endpointů, které slouží hlavně pro správu datových souborů. [3]

Pro účely migrační aplikace je nutné využít definovanou sadu endpointů v pevně definovaném pořadí.

/login - za pomocí POST[4] požadavku uživatel pošle jméno a heslo, při úspěšném přihlášení vrátí server accessToken, který slouží k další komunikaci,

/datasets - požadavek slouží k vytvoření nové datové sady, v tomto bodě se datová sada pouze inicializuje prázdná, pro vytvoření datové sady je nutné mít accessToken a znát ID organizace z LKOD aplikace,

/sessions - endpoint zařizuje tvorbu session[5], tu vytváří pro konkrétní datovou sadu, určenou pro komunikaci mezi LKOD serverem a NKOD serverem. Vrací session ID, které je nutné v kombinaci s datovou sadou poslat v dalším bodě komunikace

/form-data - Tento bod slouží pro odeslání dat. Jedná se v podstatě o standardní formulářovou akci, na kterou lze pomocí metody POST poslat data, která LKOD zpracuje a registruje.

Aplikace neumí přes API vytvářet nové organizace a neumí vytvářet nové uživatele. Pro nahrání datové sady je tedy nutné znát ID organizace, pro kterou chceme datové sady nahrávat. Toto ID lze získat z výpisu dalšího datového bodu, tím je **/organizations**, který vrací ID organizace, její název a IČO.

Přesný postup komunikace, i s ukázkovými vstupy a výstupy jednotlivých adres, je popsán na obrázku 6.6.

---

3. Dokumentace LKOD API je dostupná z https://golemiolkodapidev.docs.apiary.io/
4. POST požadavek - typ požadavku, který určuje formát zasílání dat - v těle požadavku. Druhou variantou je GET požadavek, který má veškeré parametry v hlavičce nebo v URL.
5. Session - vytvoření "spojení", s pomocí kterého se lze identifikovat vůči aplikacím

6.2.3 Analýza verzí DCAT-AP-CZ

Ačkoliv to ze struktury v příloze A nemusí být patrné, tato struktura není platná dle směrnice DCAT-AP a také dle specifikace Ministerstva vnitra. Pro jejich splnění je nutné, aby v rámci ČR obsahovala také lokální označení RÚIAN, klíčová slova a označení doby platnosti dat. [9] Což znamená, že datová sada nesplňuje otevřenou formální normu datové sady, kterou definuje Ministerstvo vnitra. Otevřená formální norma nám definuje obsah metadat datové sady. Co jsou metadata datové sady již bylo vysvětleno v podsekci 2.1.2.

Samotná otevřená formální norma, která je založena na standardu DCAT-AP-CZ, využívá verzi 2.0.1 tohoto standardu, ta vyšla jako doporučení 11. ledna 2021.

Toto doporučení dnes obsahuje **8 povinných** a **9 nepovinných** atributů. Mezi povinné se řadí: název, popis, poskytovatel, téma, periodicita aktualizace, klíčová slova, související geografické území (již zmíněné označení RÚIAN) a položky distribuce datové sady. Striktně se vyhrazuje vůči CKAN API a jasně uvádí, že v této verzi již CKAN API není podporováno, což je dáno větším nesouladem s touto směrnicí. [33]

Oproti tomu verze vycházející ze standardu DCAT-AP-CZ 1.2, vydaná jako doporučení 4. dubna 2019, konkrétně její část popisující rozhraní CKAN API, definuje jako povinné následující položky: název, popis, periodicitu aktualizace, typ RÚIAN označení, kód prvku RÚIAN, klíčová slova a položky distribuce datové sady. Rozdíl je pro lepší orientaci uveden v tabulce 6.1

Tabulka 6.1: Rozdíl povinných atributů mezi verzemi DCAT-AP-CZ

Verze 1.2	Verze 2.0.1
název	název
popis	popis
periodicita aktualizace	poskytovatel
typ RÚIAN označení	téma
kód prvku RÚIAN	periodicita aktualizace
klíčová slova	klíčová slova
distribuce datové sady	RÚIAN IRI
	distribuce datové sady

Z tabulky je patrné, že k největší změně došlo u atributů označení RÚIAN. Ty dříve obsahovaly identifikátor typu a kód prvku. Tyto byly nahrazeny za IRI územního prvku v následujícím formátu: https://linked.cuzk.cz/resource/ruian/stat/1. Původní atributy obsahovaly hodnoty - identifikátor: "ST", kód prvku: 1.

Novými položkami standardu DCAT-AP-CZ jsou **poskytovatel** a **téma**. **Poskytovatel** musí obsahovat IRI OVM[6] z registru práv a povinností. Ač je toto označení zdlouhavé, v praxi se jedná o identifikační číslo osoby (IČO), v tomto případě instituce, které je ověřitelné na adrese: **rpp-opendata.egon.gov.cz**.[7] **Téma** je novou položkou, která se nevyskytuje v dřívější verzi. Jejím obsahem musí být seznam položek z evropského číselníku datových témat.[8]

Změna nastala i u položek, které jsou přítomny v obou verzích. Formát položky **periodicita aktualizace** ve verzi 1.2 využívala hodnot dle formátu ISO 8601[9], kdežto ve verzi 2.0.1 je periodicita definována jako IRI hodnota z evropského číselníku frekvence[10]. Důležitou informací je, že musí vzniknout mapovací funkce, která původní hodnotu transformuje do nového číselníku. Příkladem budiž hodnota **P1Y** určující frekvenci aktualizace 1 rok dle ISO 8601. Tuto hodnotu lze transformovat na **http://publications.europa.eu/resource/authority/frequency/ANNUAL**.

Součástí specifikace jsou i nepovinné položky. Jejich výčet je znázorněn v tabulce 6.2. Z ní je ještě více patrné, že verze nejsou vzájemně kompatibilní a v případě aktualizace musí dojít k ručnímu doplnění chybějících hodnot.

---

6. OVM - orgán veřejné moci
7. IRI OVM pro FI MUNI je následující: https://rpp-opendata.egon.gov.cz/odrpp/zdroj/orgán-veřejné-moci/00216224
8. Příkladem tak může být hodnota: http://publications.europa.eu/resource/authority/data-theme/OP_DATPRO, která označuje typ: "Předběžné údaje".
9. ISO 8601 - https://web.archive.org/web/20171020084445/https://www.loc.gov/standards/datetime/ISO_1.pdf
10. Evropský číselník frekvence - https://op.europa.eu/cs/web/eu-vocabularies/dataset/-/resource?uri=http://publications.europa.eu/resource/dataset/frequency

Tabulka 6.2: Rozdíl nepovinných atributů mezi verzemi DCAT-AP-CZ

Verze 1.2	Verze 2.0.1
EuroVoc klasifikace	související geografické území
e-mail kurátora dat	časové pokrytí
jméno kurátora dat	kontaktní bod
odkaz na dokumentaci	odkaz na dokumentaci
časové pokrytí od	odkaz na specifikaci
časové pokrytí do	EuroVoc klasifikace
distribuce datové sady	prostorové rozlišení
	časové rozlišení
	vazba součástí jiné datové sady

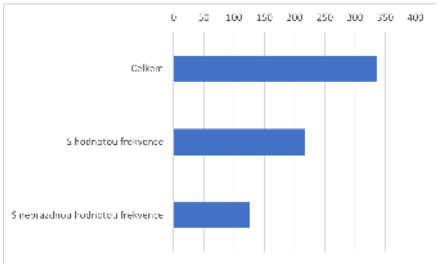
Novou položkou je zde specifikace, která definuje IRI z číselníku specifikací. Položkou lze zadefinovat otevřenou formální normu, kterou datová sada bude splňovat. Za pomocí specifikace je možné měřit kvalitu jednotlivých datových sad, což je tématem samostatné diplomové práce.

6.2.4 Analýza datových sad v reálném prostředí

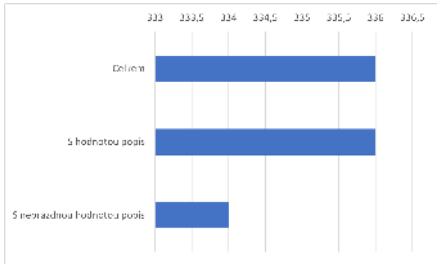
Směrnice DCAT-AP-CZ je pro poskytovatele otevřených dat závazná a datová sada (potažmo její metadata) musí tuto směrnici splňovat, pokud má být publikována. V reálném prostředí tomu tak ale vždy není a metadata datové sady směrnici neodpovídají. Při analýze reálného prostředí byly ověřeny veřejné datové sady přístupné z portálu otevřených dat Praha.

Analýza byla provedena nad všemi datovými sadami portálu otevřených dat pro Prahu[11] (aktuální počet při tvorbě této diplomové práce je 336). V grafech na obrázku 6.2 jsou znázorněny statistiky jednotlivých datových sad. Z grafu 6.2 (c) je patrné, že atribut RÚIAN je obsažen, ale nevyplněn u 62 datových sad. Dvě datové sady atribut vůbec neobsahují.

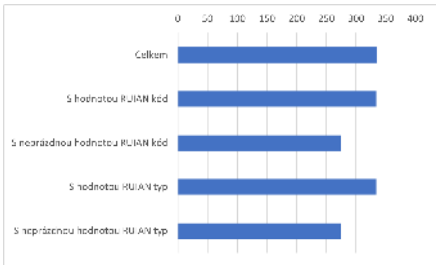
11. Portál otevřených dat Praha - https://opendata.praha.eu



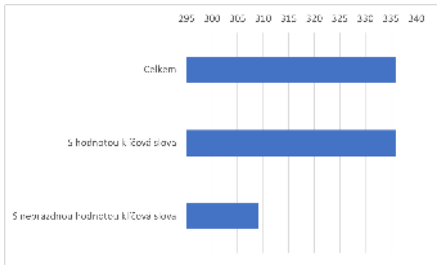
(a) Atribut **frequency**



(b) Atribut **notes**



(c) Atribut **ruian_code** a **ruian_type**



(d) Atribut **tags**

Obrázek 6.2: Statistika existence atributů v datových sadách dle doporučení DCAT-AP-CZ 1.2

Podobně je tomu i s dalšími atributy. Za zmínku stojí položka frekvence, která je obsažena v 217 sadách, ale v pouhých 125 datových sadách nalezneme nějaké informace. Absence atributů značně komplikuje proces migrace. Migraci nelze provést 1:1 a musí být nabídnuta nějaká forma předvyplnění atributů pro splnění doporučení DCAT-AP-CZ 2.0.1.

6.2.5 Analýza typů SPARQL databází

Součástí zprovoznění serveru je i tvorba SPARQL služby. Tato práce využívá pro zprovoznění služby lokálního katalogu SPARQL engine Fuseki. I přesto je ale provedena základní analýza běžně dostupných SPARQL aplikací, které pro tyto účely lze použít.

Fuseki open-source aplikace, kterou je možné spustit jako samostatnou službu s uživatelským rozhraním, nebo jako aplikaci uvnitř Docker kontejneru. Server poskytuje nejen SPARQL protokol

pro dotazování a úpravu dat, ale nabízí i uložiště těchto dat formou TDB. To nabízí perzistentní uložiště RDF souborů. [34] Aplikace je vývíjena komunitně, jako open-source. Přispívat do ní tedy může každý, proejkt spravuje Apache Software Foundation (stejná společnost zaštiťující webový server Apache).

RDF4J Java framework sloužící pro správu a zpracování RDF souborů. Mezi hlavní výhody patří jednoduché aplikační rozhraní. Tvorba RDF modelů je možná přímo v jazyce Java. Pro správu dat nabízí tzv. Repository API, které nabízí přístup přes SPARQL query jazyk. RDF4J nabízí datové uložiště formou ukládání dat v RAM paměti, na disku a pomocí ElasticSearch. [35] Framework je využíván dalšími SPARQL enginy, příkladem může být Graph Db Ontotext. Licence tohoto frameworku je open-source, znamená to tedy, že jej opět může upravovat a rozšiřovat kdokoliv a je distribuovatelný zdarma. Tvůrcem je společnost Eclipse Foundation, Inc. Tato společnost je známá tvorbou vlastního IDE prostředí Eclipse.

Virtuoso další open-soruce platforma pro tvorbu a distribuci SPARQL databází. Virtuoso lze označit za vysoko-výkonný SQL databázový server, který má podporu pro SPARQL. RDF soubory jsou uloženy přímo v databázi, tato aplikace je využívána národním katalogem. Virtuoso je pravidelně vyvíjeno a aktualizováno, v době tvorby této práce byla poslední verze 7.2.6, která byla vydána 22.6.2021. [36] Aplikace je vydávána v open-source a komerční verzi. Open-source verze je volně dostupná na Githubu a je tak možné ji pod licencí GNU-GPL používat. Komerční licence je placenou verzí. Tvůrcem aplikace je OpenLink Software, Inc.

Oxigraph databázový server implementující SPARQL protokol, jedná se o aplikaci, která je stále ve vývoji a SPARQL dotazy nejsou optimalizované. Implementace této aplikace probíhá v Rustu a v Pythonu. Oxigraph nenabízí veřejně dostupné docker balíčky, uživatel si tedy musí sám balíček vystavět, pokud má zájem o spuštění pod aplikací Docker. [37] Aplikace je distribuována pod licencemi Apache a MIT, tj. jedná se o open-source aplikaci. Aplikaci tvoří komunita vývojářů.

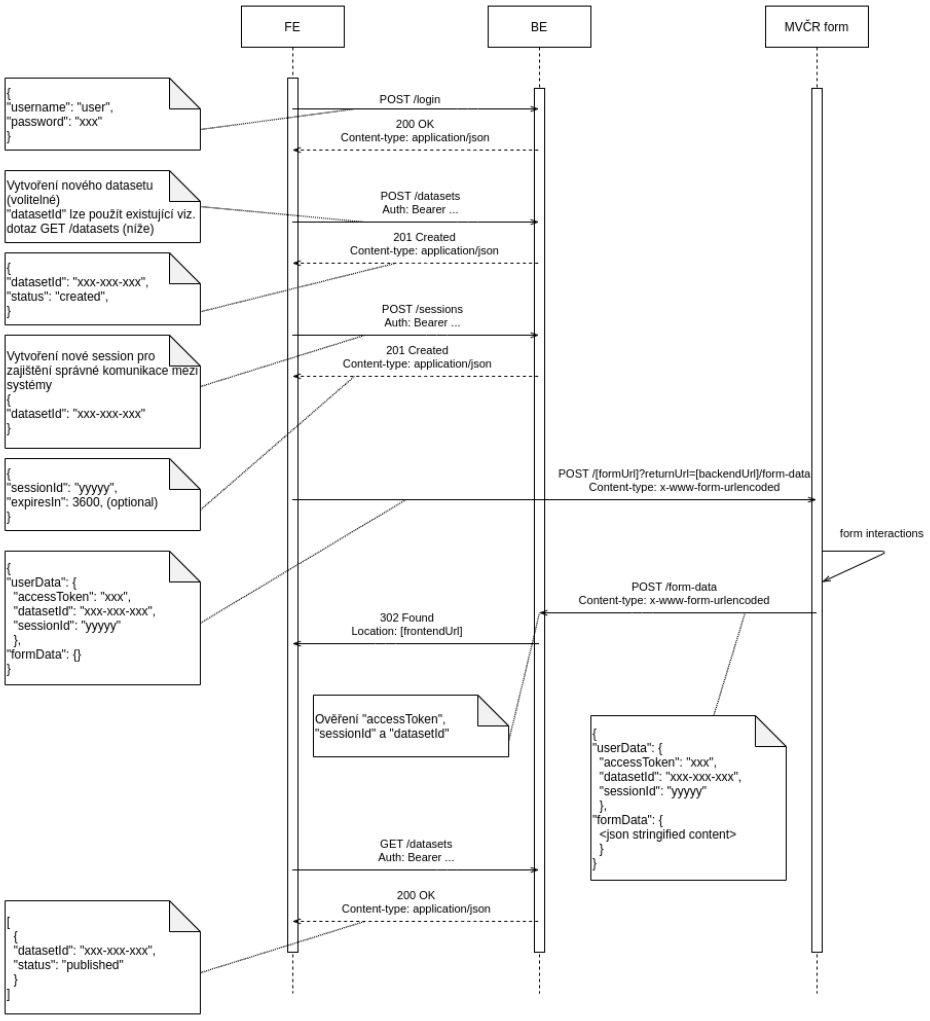
Graph DB Ontotext GraphDB je vysoce efektivní grafová databáze, která využívá RDF4J frameworku a aplikačního rozhraní pro ukládání a dotazování RDF souborů. Mezi podpoované RDF formáty patří RDF/XML, N3, Turtle. [38] Aplikace byla vyvinuta společností Ontotext USA, Inc., která se zaměřuje na správu dokumentů a databáze. Aplikace je dostupná ve třech edicích, přičemž jediná edice zdarma slouží k vyzkoušení aplikace a k provozu s menšími službami.

6.2.6 Migrační proces

Jednotlivé komponenty (CKAN API, LKOD API, otevřená formální norma datové sady v obou verzích) byly zanalyzovány. V rámci migrace existují dva zásadní problémy, které by migrace měla usnadnit a pokud možno odstranit. Problémem je nekonzistence mezi verzemi standardu DCAT-AP-CZ a nutnost vyplňovat metadata k jednotlivým datovým sadám ve formuláři národního katalogu. Ten byl zmíněn už v kapitole 6. Migrační proces lze rozdělit na dvě části - samotný proces z pohledu správce dat a aplikační proces migrace dat. Oba jsou popsány v následujících podsekcích.

6.2.7 Návrh procesu migrace dat na LKOD

Ze strany OICT proběhl návrh komunikace lokálního katalogu dat a národního katalogu. Komunikace je popsána na diagramu 6.3. Implementace migračního skriptu je vyvozena právě z tohoto diagramu. Je zde však několik rozdílů v rámci implementace. Diagram komunikace je pak viditelný na obrázku 6.6.

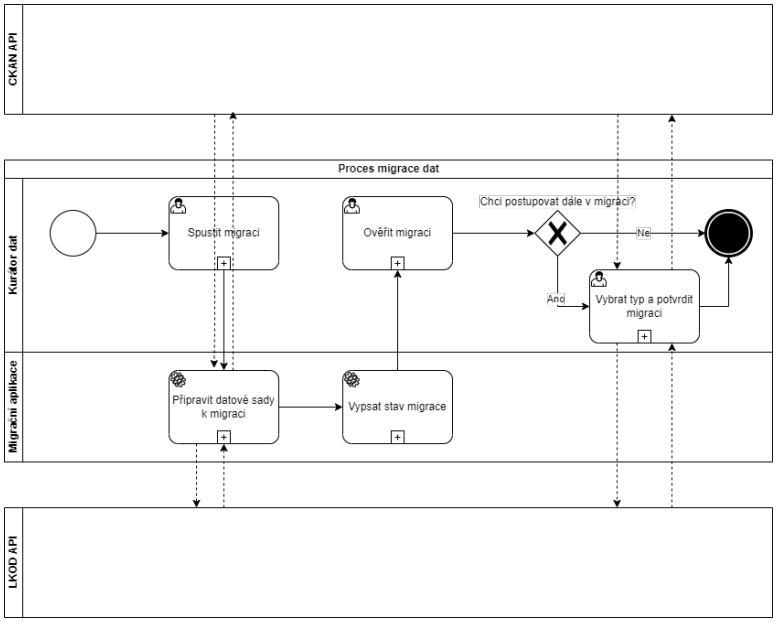


Obrázek 6.3: Průběh komunikace mezi LKOD

Proces začíná přihlášením ověřením dostupnosti serveru CKAN a LKOD, následně pokračuje přihlášením k aplikaci lokálního katalogu od OICT. Po úspěšném přihlášení musí proběhnout stažení všech datových sad ze serveru CKAN. V závislosti na vyplnění API klíče pro CKAN dojde k načtení buď veřejných nebo i neveřejných datových sad. Neveřejné jsou staženy pouze při vyplnění správného API klíče. Následně musí dojít ke stažení všech datových sad, každá datová sada musí být transformována do formátu otevřené formální normy datové

sady verze 2.0.1. Pokud jsou metadata datové sady dle OFN platná, následuje uložení do seznamu datových sad, které jsou **plně zmigrovatelné**. V opačném případě je sada přiřazena do seznamu **zmigrovatelné s úpravami** a pokračuje se další iterací datové sady, dokud existují nezmigrované datové saday. Výstupem tohoto procesu jsou tedy dva seznamy datových sad - **plně zmigrovatelné** a **zmigrovatelné s úpravami**.

V druhém kroku si uživatel musí sám vybrat, jaké datové sady chce zmigrovat. Tímto krokem může na vlastní zodpovědnost provést migraci datových sad, které je nutné upravit dále. Po potvrzení migrace dojde k migraci všech vybraných datových sad. Pro tyto účely dále slouží subproces popsaný v podsekci 6.2.8.



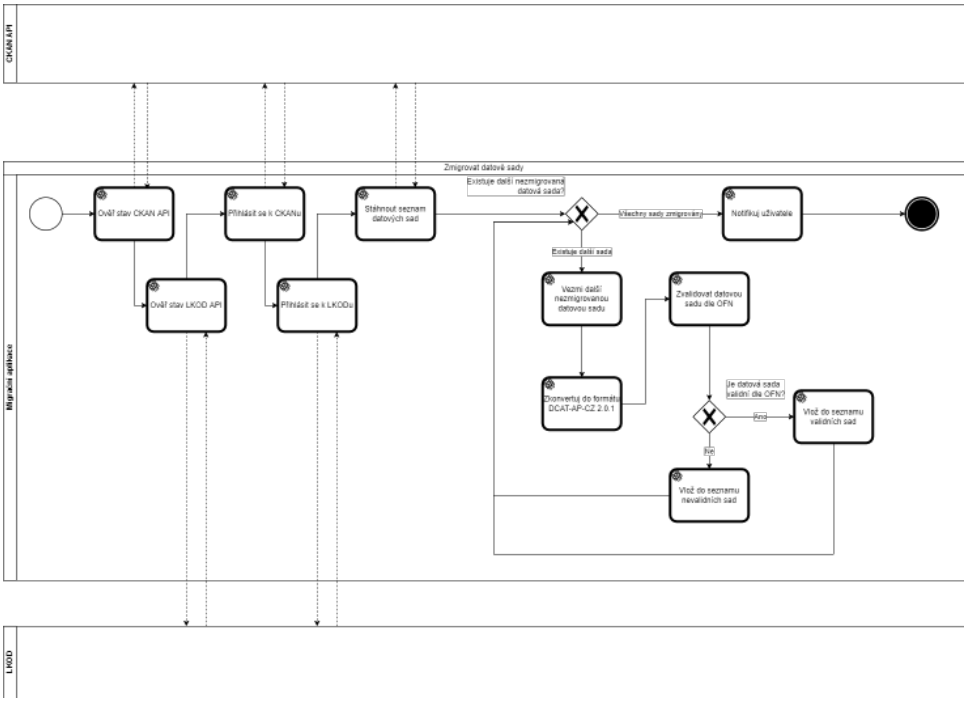
Obrázek 6.4: Proces migrace z pohledu správce dat

6.2.8 Subproces migrace datové sady do NKOD

Proces migrace datových sad do NKOD je složen z komunikace mezi servery migrační aplikace, LKOD backend a NKOD serverem. Model procesu je na obrázku 6.5.

Aplikace se nejprve přihlásí k lokálnímu katalogu, obdrží přihlašovací token **accessToken**. S tímto se dále autentizuje ve všech voláních

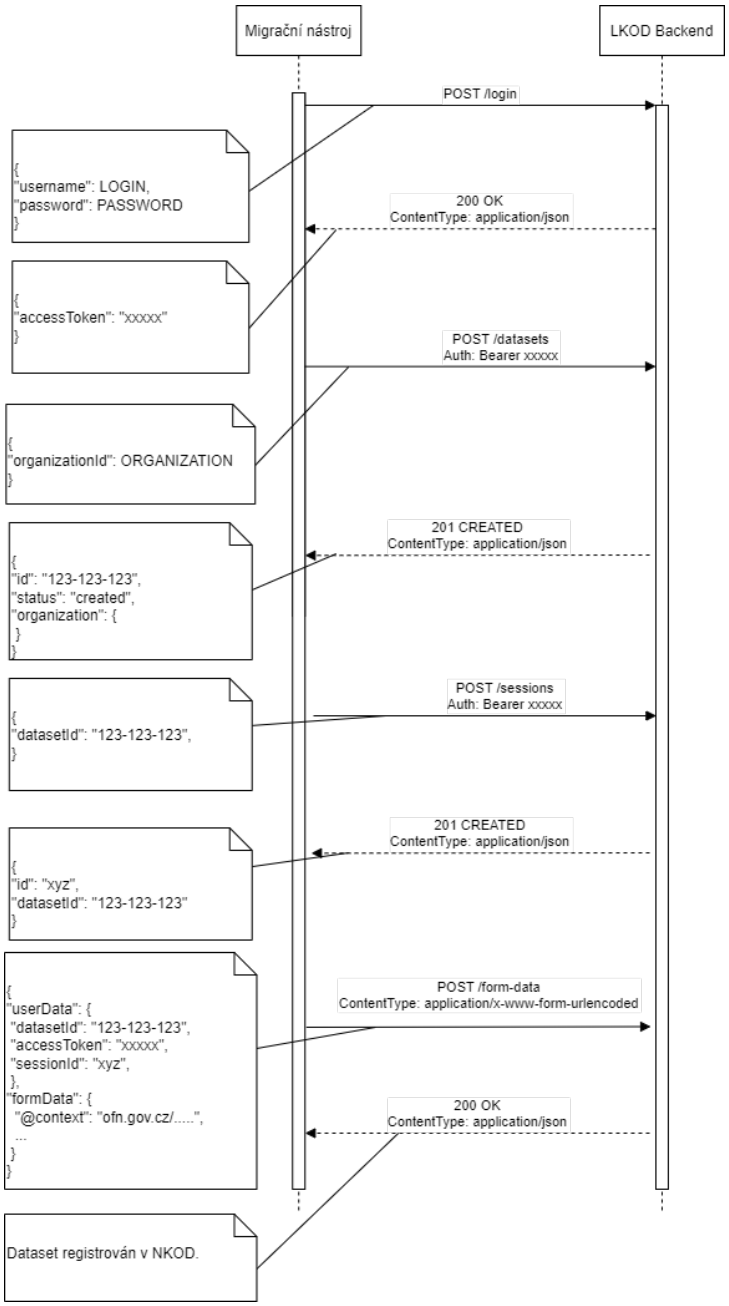
API. Nutné je poté vytvořit datovou sadu pod konkrétní organizací. Po jejím úspěšném vytvoření je stěžejní vytvořit sezení[12], která umožňuje komunikaci mezi aplikací migrace, LKOD serverem a NKOD serverem. Pro nahrání metadat datové sady je pak potřebné poslat samotná metadata spolu s autentizačními informacemi (**accessToken**, **datasetId** a **sessionId**) na server lokálního katalogu, který data ukládá a po publikaci distribuuje pomocí SPARQL endpointu. Po zpracování je výsledek stavu zaslán zpět a víme tak, zda byla migrace konkrétní datové sady provedena úspěšně.



Obrázek 6.5: Subproces migrace

12. sezení - session - navázané spojení pro autentizaci mezi servery

6.2.9 Komunikace migrace



Obrázek 6.6: Průběh komunikace mezi migračním nástrojem a lokálním katalogem od OICT

6.3 Implementace aplikace

Aplikace migračního skriptu byla navržena tak, aby byla spustitelná na co největším počtu zařízení. Z toho důvodu byla vytvořena webová aplikace na adrese, která je uvedena v příloze této diplomové práce. Aplikaci lze nainstalovat dvěma způsoby:

- manuální instalace,
- instalace přes docker.

Oba postupy instalace jsou popsány v podsekci 7.2.

Aplikace využívá frameworku Flask 5.4 k vytvoření přívětivého uživatelského rozhraní pro snadnou migraci dat. Webová aplikace je rozdělena mezi backendovou část a frontendovou část.

6.3.1 Backend

Backendová část aplikace implementuje proces celé migrace. Součástí migrace je přihlášení k jednotlivým lokálním katalogům, stažení informací o datových sadách. To zařizují tři hlavní třídy - konfigurační třída, třída pro stažení dat a mapovací třída. Backendová část vystavuje pouze potřebné metody, které frontendová část volá.

Konfigurační třída - Config - Tato komponenta slouží k vytvoření konfigurace pro migraci. V rámci konfigurace se udržují informace o CKAN serveru - **URL** a **accessToken**.

Třída pro stažení dat - Fetcher - Komponenta zapouzdřuje API požadavky pro CKAN do ucelených funkcí tak, aby nebylo nutné při každé komunikaci na CKAN nebo na LKOD nutné znovu vyplňovat parametry požadavků.

Třída pro konverzi dat - Migrator - Tato třída provádí mapování struktury DCAT-AP 1.2 na novou verzi 2.0.1. Při inicializaci třídy dojde k naplnění hodnot z MigrationForm formuláře, který je popsán v následující podsekci. Hlavní metodou této třídy je **prepare_dataset_json_object()**. Ta zkontroluje veškeré atributy z datové sady ze CKANu, které jsou relevantní pro DCAT-AP-CZ 2.0.1. Tyto atributy převede do nového formátu, pokud je

třeba, a přiřadí je do adekvátních polí v novém objektu. Výstupem je pak slovník v Pythonu, který lze snadno zkonvertovat na JSON objekt. Mezi další metody této třídy patří:

convert_ISO_8601_to_eu_frequency(frequency) - metoda provede konverzi hodnoty periodicity aktualizace z formátu 1.2 na formát dle Evropského číselníku hodnoty platny pro doporučení 2.0.1

get_ruian_type(uri, ruian_type, ruian_code) - metoda z parametrů metodě zadané automaticky vybere vhodný formát pro konverzi. Pokud je zadána URI, je zkonvertována původní adresa na formát:
https://linked.cuzk.cz/resource/ruian/OBLAST/ID, kde **OBLAST** je typ územního celku a **ID** je ID územního celku. V opačném případě je proveden pokus o konverzi atributů **ruian_type** a **ruian_code** do výše zmíněného URL formátu.

migrate_dataset(dataset) - metoda zajistí stažení dat z CKAN aplikace, konverzi do nového formátu a nahrání na LKOD backend.

migrate() - metoda zapouzdřuje předchozí metodu do cyklu tak, aby bylo možné jedním voláním možné provést migraci celé aplikace.

6.3.2 Frontend

Frontendová část aplikace přebírá podstatné prvky z konceptu MVC[13] frameworků. Její hlavní rozdělené je na: **templates**, **controllers** a **models**.

Adresář models

Složka **models** obsahuje všechny třídy (modely), které využívá webová aplikace - přítomny jsou dva formuláře, které provádí zpracování dat, validační třída pro JSON objekty. Formuláře přítomné jsou **LoginForm** a **MigrationForm**.

13. MVC - model view controller, paradigma pro tvorbu webových aplikací, kterým se řídí značná část webových frameworků

LoginForm třída slouží ke zpracování úvodního formuláře na hlavní stránce. Vystavuje sadu metod, které umožňují přihlášení k CKAN serveru. Níže je seznam veřejných metod této třídy a jejich popis:

- **process_data()** - hlavní metoda této třídy, po úspěšné validaci formuláře provede inicializaci migračního skriptu, stažení všech datových sad, konverzi do nového formátu a uložení do dvou seznamů rozlišujících validaci datové sady vůči OFN.

Třída MigrationForm provádí zpracování již zanalyzovaných dat. Jedná se o sekundární formulář, který uživatel musí vyplnit po úspěšném načtení datové sady. Mezi parametry konfigurace patří typ migrace a seznam datových sad. Tyto datové sady jsou výstupem předchozího formuláře. Typ migrace určuje, které datové sady se budou dále migrovat. Mezi hlavní metody této třídy patří:

- **process_data()** - jedná se o hlavní metodu této třídy, která zpracovává data z tohoto formuláře, provede načtení spojení ze session k CKAN serveru, LKOD serveru a provede finální migraci dle uživatele zadaných informací.
- **get_migration_datasets()** - vrátí seznam datových sad rozdělených dle validace vůči OFN.

Adresář controllers

Složka **controllers** obsahuje všechny controllery, které aplikace pro frontend vystavuje. Pod každým controllerem je registrován seznam akcí, každá akce je samostatnou podstránkou v rámci webové aplikace. Ve stávající verzi aplikace je dostupný controller **site**, který obsahuje následující akce:

- **index** - úvodní akce, toto je domovská stránka aplikace, na které je vypsán úvodní migrační formulář. Akce zpracovává jak GET tak POST požadavky, při odeslání GET požadavku se vypíše tato homepage. Při odeslání POST požadavku s daty formuláře se odešle úvodní formulář, po jehož zpracování dojde ke stažení datových sad a vypíše se druhý krok umožňující potvrzení migrace.

- **authors** - stránka popisující seznam autorů této práce a informace o Laboratoři servisních systémů, akce zpracovává pouze GET požadavky.
- **validate** - akce pro dynamickou validaci jednotlivých datových sad, jejím výstupem je seznam chyb datové sady vůči OFN datové sady, akce zpracovává pouze POST požadavky.
- **installation** - stránka popisující postup pro instalaci tohoto migračního nástroje, akce zpracovává pouze GET požadavky.

Adresář templates

Pro tvorbu šablon byla využita základní šablona Bootstrap, která nabízí jednoduchý a responzivní design, který lze ihned použít. Nabízí také sadu komponent s pevně definovaným chováním, která pro tyto účely byla maximálně vhodná.

O rozšíření funkce šablon se dále stará knihovna jQuery, která je nádstavbou Javascriptu. Nabízí možnost provádět AJAX volání[14]

Flask lze využívat vykreslováním standardního HTML, avšak nabízí i šablonovací systém, nazývající se Jinja2. Ten usnadňuje práci při tvorbě HTML zdrojových kódů, v rámci šablon lze zapisovat podmínky přímo do HTML. [39]

Šablony jsou rozděleny do složek: **layouts** a **site**. Ve složce layouts je definována obecná šablona, ze které dědí všechny ostatní šablony stránek. V rámci hlavní šablony je definován import CSS šablon, JavaScript knihovny jQuery a základní rozvržení, kterého se drží zbytek webu. Jmenovitě tak tvoří menu, záhlaví stránky.

Jinja2 nástroj je použit pro tvorbu jednotlivých dynamických prvků stránek. Na výpisu 6.1 je znázorněn for cyklus umožňující výpis všech varovných zpráv, které webová aplikace uživateli může vypsat.

```
{% with messages = get_flashed_messages(
   with_categories=True) %}
{% if messages %}
    {% for category, message in messages %}
```

14. AJAX volání - AJAX request - dynamické volání požadavků na webový server

```
        <div class="alert alert-{{category}}"
           role="alert">{{ message }}</div>
        {% endfor %}
    {% endif %}
    {% endwith %}
```

Výpis 6.1: Výpis varovných zpráv v aplikaci pomocí šablonovacího nástroje

6.4 Testování

6.4.1 Otestování datových sad

V rámci analýzy datových sad, která byla provedena v podsekci 6.2.4, bylo zjištěno, že prakticky žádné datové sady nebudou vyhovovat. To je dáno za prvé nekonzistencí doporučení DCAT-AP-CZ verze 1.2 a 2.0.1. Druhým důvodem proč datové sady nebudou vyhovovat je fakt, že v reálném prostředí nebyly datové sady naplněny všechny správně. Z toho je patrné, že nelze garantovat správnou migraci všech sad.

Pro účely testování byly provedeny migrace serverů: sc02.fi.muni.cz a opendata.praha.eu. Právě pomocí nich bylo provedeno ověření jak platných, tak neplatných datových sad dle obou směrnic. V průběhu testování bylo naraženo na problémy s chybějícími daty, které bude nutné předvyplnit tak, aby datová sada vyhovovala otevřené formální normě datové sady. Konzultací se zástupci z MVČR[15] jsme dospěli k závěru, že není možné hodnoty předvyplnit staticky nějakou pevnou hodnotou. Uživatel se sám musí rozhodnout, zda chce hodnotu předvyplnit. Pokud ano, musí být schopen vybrat si kterou hodnotu, v opačném případě je nutné uvést stav: "nespecifikováno". Podobný přístup je již v praxi využíván, příkladem budiž datová sada **Cenový barometr** od ČTÚ[16]. [40]

Aplikace v tomto ohledu splnila požadavky tak, aby odpovídaly podmínkám nastavenými MVČR. Praxe je tedy taková, že pokud datová sada licenci obsahuje, jsou licence konvertovány do nového formátu, pokud ne, aplikace nabízí předvyplnění globálně pro všechny

15. Ministerstvo vnitra České republiky
16. ČTÚ - Český Telekomunikační Úřad

sady stejnými hodnotami. Pokud uživatel odmítne předvyplnění, je taková sada nevalidní a může si na vlastní zodpovědnost vybrat, zda bude zkonvertována.

6.4.2 Testování aplikace

Pro účely testování byly provedeny manuální testy nad výše zmíněnými servery. Spolu s testováním migrační aplikace byla ověřena aplikace lokálního katalogu, do které byla data nahrány. V rámci aplikace lokálního katalogu se podařilo objevit tři významné chyby, které byly zdokumentovány ve verzovací aplikaci Gitlab, kterou OICT využívá pro vývoj aplikace lokálního katalogu.

6.5 Uvedení do provozu

6.5.1 Tvorba dockerfile souboru

Aplikaci migrační aplikace má samostatný dockerfile soubor, kterým lze aplikaci zabalit do docker balíčku. Takový balíček lze následně spustit jako samostatný kontejner v rámci Dockeru. Docker a jeho funkcionalita byly popsány v kapitole 5.

Dockerfile soubor je shell[17] skript, který definuje, jaké příkazy se mají provést, aby byla aplikace spuštěna. Pro migrační aplikaci byl nachystán dockerfile, který je na výpisu 6.2.

```
# syntax=docker/dockerfile:1

FROM python:3.7-slim-buster

LABEL org.opencontainers.image.authors="500352
   @mail.muni.cz, 133@muni.cz,
   dominikskala@seznam.cz"

COPY requirements.txt requirements.txt
```

16. https://gitlab.com/operator-ict/golemio/lkod/lkod-backend/-/issues/7
16. https://gitlab.com/operator-ict/golemio/lkod/lkod-backend/-/issues/8
16. https://gitlab.com/operator-ict/golemio/lkod/lkod-backend/-/issues/9
17. uživatelské prostředí / skriptovací jazyk v prostředí systému Unix

```
WORKDIR /app

COPY app/ /app
COPY requirements.txt /requirements.txt
RUN pip3 install -r /requirements.txt

COPY main.py /main.py

CMD ["python3", "/main.py"]
```

Výpis 6.2: Dockerfile skript pro migrační aplikaci

6.5.2 Instalace na lkod.fi.muni.cz serveru

Instalace na společném serveru již v rámci docker aplikace nezabrala příliš dlouho. Pro vytvoření nového kontejneru bylo potřebné vložit do docker-compose souboru pro obecnou aplikaci nový kontejner. Přesný popis ukázky instalace přes docker-compose je na výpisu 6.3. Aplikace se s touto konfigurací spustí na portu 5000 a je možné k ní přistoupit přes adresu: http://lkod.fi.muni.cz:5000. V příloze B této práce je uveden ukázkový docker-compose.yml soubor pro spuštění aplikace na vlastní doméně - konkrétně na http://lkod-migrace.cz.

```
  lkod-migration:
    container_name: lkod-migration
    image: skalincz/lkod-python-migration:latest
    networks:
      - lkod
    ports:
      - "5000:5000"
    depends_on:
      - "lkod-be"
```

Výpis 6.3: Část docker-compose souboru popisující instalaci migrační aplikace

Kontrola funkčnosti byla dále nad aplikací provedena pokusem o migraci serverů sc02.fi.muni.cz a opendata.praha.eu. Výsledkem je funkčně provedená migrace, která byla provedena nad oběma servery.

6.6 Řízení projektu

Komunikace s OICT probíhala na několika platformách. Úvodní komunikace byla vedena skrze e-maily, později byl zvolen nástroj Slack, konkrétně do komunikační skupiny **Otevřená data ČR**. V této komunikační skupině jsou přítomni specialisté z OICT, MVČR, Ministerstva financí a další externí řešitelé. Pro meetingy a jednání byl využíván nástroj Microsoft Teams, v rámci kterého probíhaly prezentace jednotlivých iterací aplikace a analýza zadání.

7 LKOD as a Service

7.1 Návod na instalaci LKOD

Instalační proces lokálního katalogu od ICT prošel značnou úpravou. Na začátku spolupráce neexistoval žádný jasný návod, jak aplikaci nainstalovat. Konkrétní proces iterací jednotlivých instalací je popsán v kapitole 4.

Aplikace LKOD je rozdělena mezi dvě části - frontendovou část a backendovou část. Každá má svůj repozitář na portálu Gitlab. Ukázková instalace je provedena na serveru Debian 10.

Instalace aplikace je nyní shrnuta do 5 kroků:

1. Instalace docker aplikace
2. Vystavění vlastní instance frontendové aplikace
3. Inicializace docker-compose souboru
4. Inicializace klíčů
5. Spuštění aplikace

7.1.1 Instalace docker aplikace

Pro nejjednodušší spuštění aplikace je třeba mít nainstalovanou aplikaci docker, který umožňuje instalaci přednastavených balíčků a jejich správu na uzavřeném prostředí. Jedná se o kontejnerové prostředí, kdy se kontejnery chovají podobně jako virtualizované stroje - tedy neovlivňují prostředí, ve kterém jsou nainstalovány. V rámci prostředí operačního systému Debian je možné použít příkaz:

```
sudo apt-get remove docker docker-engine
   docker.io containerd runc
```

Dále musí být provedena inicializace repozitáře z Gitlab verzovacího nástroje příkazem:

```
git clone https://gitlab.com/operator-ict/
   golemio/lkod/lkod-general.git
```

7.1.2 Vystavění vlastní instance frontendové aplikace

Frontendová část aplikace lokálního katalogu podporuje spuštění na samostatném serveru. Pro spuštění na vlastní URL adrese je ale nutné vytvořit vlastní verzi aplikace. Je potřeba provést následující:

1. Stáhnout repozitář frontendové aplikace.
2. Upravit soubor **.env** s proměnnou **PUBLIC_URL** na hodnotu: "/CESTA", kde **CESTA** je prefixovaná hodnota URL na serveru. Každá adresa pak bude prefixovaná tímto textem (např. /admin - URL pak budou ve formátu: /admin/prihlaseni).
3. Spustit kompilaci aplikace příkazem: `docker build . --tag=NAZEV`, kde **NÁZEV** je identifikátor kontejneru. Tento název bude později použit v docker-compose souboru jako reference zdroje kontejneru.

7.1.3 Instalace pomocí docker-compose souboru

Druhým krokem instalace je inicializace docker-compose souboru. Je vhodné vytvořit složku uvnitř složky `/home`. V rámci této složky je nutné vytvořit docker-compose.yml soubor příkazem:

```
touch docker-compose.yml
```

Ukázkový docker-compose pro aplikaci je uveden v příloze.

7.1.4 Inicializace klíčů

Dalším krokem je inicializace klíčů. Ty jsou využívány k vytváření a podepisování access tokenů. Je samozřejmě nutné vytvořit jak soukromý, tak veřejný klíč. Pro vygenerování klíčů v prostředí operačního systému Debian je možné provést následující příkazy:

```
ssh-keygen -t rsa -b 4096 -m PEM -f key.key
openssl rsa -in key.key -pubout -outform PEM
    -out key.pub
```

Oba klíče musí být pro aplikaci viditelné, to znamená, že je nutné nad nimi provést ještě přiřazení oprávnění pro čtení a zápis:

```
chmod 755 key.*
```

Následně musí být přichystána adresářová struktura pro SPARQL:

```
touch fuseki-users
mkdir fuseki-db
chmod 777 fuseki-db
chmod 777 fuseki-users
```

Soubor fuseki-users musí obsahovat uživatelské jméno a heslo k přihlášení k lkod serveru. Ty jsou ve formátu: "jmeno:heslo". Z těchto údajů je pak vygenerován Authorization bearer token uvnitř .env.be souboru, který obsahuje tento text zakódovaný v Base64 kódování.

7.1.5 Spuštění aplikace

Pro spuštění všech kontejnerů s aplikací je třeba provést příkaz: `docker-compose up` ve složce s vytvořeným docker-compose.yml souborem. Pro spuštění aplikací na pozadí je vhodné přidat parametr `-d`. Výsledný příkaz je pak `docker-compose up -d`.

Po provedeni příkazu dojde ke stažení všech kontejnerů, k jejich inicializaci a spuštění s parametry určenými v docker-compose.yml souboru.

S prvotním spuštění aplikace je nutné zinicializovat obsah databázového serveru. Pro účely Masarykovy univerzity je vytvořena výchozí organizace: **FI MUNI** spolu s výchozím uživatelem. Přístupové údaje jsou přiloženy v příloze této práce.

7.2 Návod na instalaci migračního nástroje CLM

Aplikaci CLM lze nainstalovat dvěma způsoby. Prvním, který je již využit v rámci instalace LKOD aplikace, je možnost využít docker-compose.yml souboru. Pro spuštění aplikace byl vytvořen Dockerfile, který automaticky stáhne aplikaci, nainstaluje potřebná rozšíření a spustí ji. Ukázkový dockerfile je dostupný na výpisu 6.2.

Dockerfile definuje, jak se má daný balíček vytvořit. Po stažení repozitáře tak může kdokoliv a kdykoliv vytvořit docker balíček pro vlastní potřeby. Vytvoření balíčku je pak zařízeno příkazem níže:

```
docker build . --tag skalincz/lkod-python-
   migration:latest
```

Po tvorbě balíčku jej můžeme kdykoliv v rámci docker prostředí spustit ručně přes příkaz: `docker run -d -p 80:5000 lkod-python-migration`. Tím aplikaci spustí veřejně na portu 80, v rámci kontejneru používá port 5000, na kterém je ve výchozím stavu webová aplikace spuštěna. Pro spuštění aplikace na konkrétní podstránce (bez označení portu aplikace) spolu s aplikaci LKOD je v příloze této diplomové práce uveden docker-compose.yml, který tuto logiku zařizuje.

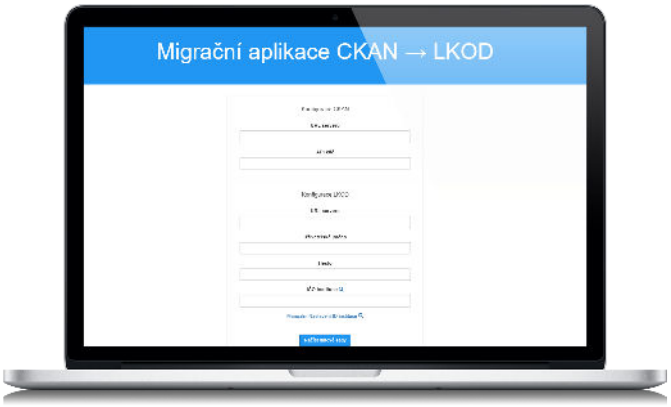
Druhý způsob instalace je bez možnosti docker aplikace. V takovém případě je žádoucí postupovat následovně:

- Naklonování repozitáře s migrační aplikaci:
  `git clone https://github.com/Skalin/CLM`
- Instalace Python 3.7 (operační systém Debian již v základní verzi má předinstalovanou verzi Python 3)
- Instalace pip balíčkovacího systému:
  `apt install -y python3-pip`
- Otevření adresáře s projektem:
  `cd CLM`
- Instalace povinných balíčků pro aplikaci:
  `pip3 install -r requirements.txt`
- Spuštění aplikace:
  `python3 ./main.py`

7.3 Návod na migraci ze CKANu

Z uživatelského hlediska můžeme návod definovat jako proces, ten je vizualizován na obrázku 6.4 a je definován v seznamu níže.

1. Příchod na stránku: **http://lkod-migrace.cz**.

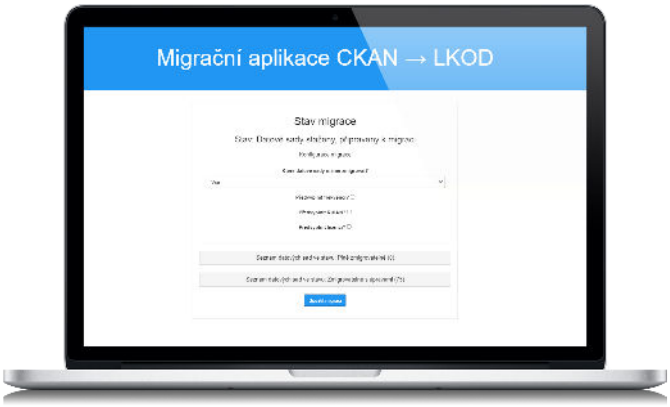


Obrázek 7.1: První krok migrace v aplikaci CLM

2. **Vyplnění údajů o CKAN serveru, LKOD serveru a instituci.**
   V tomto bodě musí uživatel znát adresu předchozí CKAN serveru, přístupové údaje k LKOD serveru a ID instituce. V neposlední řadě musí také znát IČO instituce, kterou chce zmigrovat. Ukázkové vyplnění je viditelné na obrázku 7.1. Po odeslání formuláře dojde k validaci adres obou serverů a následně k ověření dostupnosti API na obou serverech. V případě, že je API na serverech nedostupné, je uživatel upozorněn a požádán o ověření údajů. Po úspěšném ověření dojde k automatickému stažení všech datových sad ze CKANu. Uživatel může a nemusí vyplnit API klíč pro CKAN, čímž umožní stažení privátních datových sad.

3. **Načtení datových sad**
   Po úspěšném ověření obou serverů se provede načtení všech datových sad ze strany CKANu, ty jsou transformována do verze DCAT-AP-CZ 2.0.1 Datové sady jsou automaticky rozděleny do

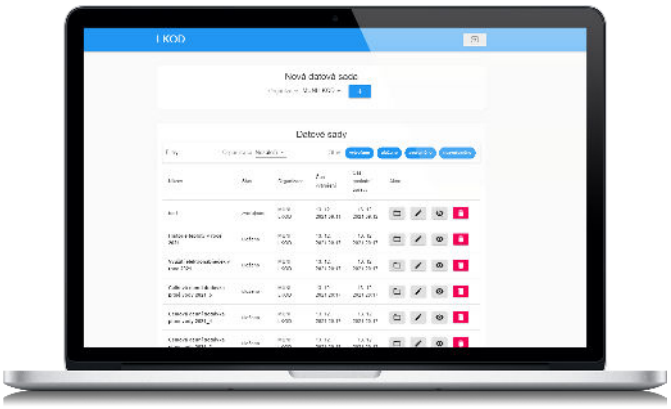
dvou typů: **Plně zmigrovatelné** a **Zmigrovatelné s úpravami**, jak je vidět na obrázku 7.2.



Obrázek 7.2: Druhý krok migrace - výpis datových sad pro potvrzení

4. **Výběr varianty migrace**
   V tomto bodě si uživatel může jednotlivé datové sady prověřit a zkontrolovat, v čem neodpovídají OFN datové sady. Seznam chyb je možné vidět přímo v modálním okně u každé datové sady po rozkliku tlačítka: "Chyby datové sady". V průběhu konfigurace lze vybrat parametry migrace. Uživatel může nechat předvyplnit licence, frekvenci aktualizace dat a RÚIAN. Tyto hodnoty budou nastaveny globálně pro všechny datové sady, kterým chybí některý z těchto parametrů. Pokud jejich předvyplnění nebude povoleno, dojde (v případě výběru migrace i nevalidních sad) k zmigrování ve stavu, v jakém byly získány ze CKANu.

5. **Potvrzení a spuštění migrace**
   Po potvrzení migrace se provede nahrání všech datových sad. Po úspěšném dokončení je uživatel přesměrován na úvodní stránku s výpisem potvrzovací hlášky o stavu migrace.



Obrázek 7.3: Výpis datových sad v lokálním katalogu

7.4 Typický uživatel

V rámci této diplomové práce jsou definování dva typičtí uživatelé. Pro správu a nastavení samotných aplikací (aplikace lokálního katalogu, migrační aplikace) je nutné definovat prvního uživatele. Pro samotnou údržbu datových sad a migrace je nutné definovat uživatele druhého.

V rámci správy a instalace obou aplikací je zapotřebí technického vzdělání a nebo zkušenost s technologiemi, které jsou zmíněné v této práci v kapitole 5. Takový uživatel by tedy měl být schopen vytvořit a sám spravovat kontejnery v rámci aplikace Docker, měl by být schopen řešit dílčí problémy se spuštěním. Rovněž by měl být schopen pracovat v systému Linux. Je nutné zmínit, že aplikace lokálního katalogu od ICT momentálně nepodporuje tvorbu dalších organizací a uživatelů v rámci grafického rozhraní. Pro tyto účely je nutné, aby byla osoba schopna napsat vlastní sadu SQL dotazů, které vytvoří jak uživatele, tak organizace v rámci aplikace lokálního katalogu.

Správce serveru můžeme definovat následujícími požadavky:

- Znalost anglického jazyka.
- Znalost operačního systému Linux (Debian, Ubuntu), instalace systému, správa a užívání systému.
- Alespoň základní znalost SQL (konkrétně PostgreSQL), schopnost tvorby vlastních SQL dotazů pro tvorbu nových záznamů.

- Znalost aplikace Docker (schopnost vytvořit a spravovat jednotlivé kontejnery, řešit provázání, nasazovat jednotlivé instance).

Pokud takováto osoba spravuje migrační aplikaci a řeší samotné migrace, měla by se orientovat v tématice OFN a být nápomocná, nebo se sama stát kurátorem otevřených dat pro danou instituci. To obnáší znalosti licencování otevřených dat, znalost tvorby datových sad a schopnost řešit kvalitu datových sad.

Z druhého pohledu, lze pak zobecnit správce samotného lokálního katalogu a migrační aplikace. Taková osoba musí být alespoň částečně technicky zdatná, musí být schopna základní práce na PC a rozumět alespoň základním pojmům v oblasti IT jako jsou FTP servery, přihlášení k aplikaci, datové typy.

Správce dat můžeme definovat následujícími požadavky:

- Znalost anglického jazyka.
- Pokročilá znalost práce na PC.
- Schopnost ovládat a pracovat s FTP, Amazon S3.
- Znalost DCAT-AP-CZ 2.0.1, orientace v problematice licencování datových sad, v jejich struktuře.

Když si oba seznamy porovnáme, je z nich patrné, že se ve své podstatě jedná o popisy pracovních pozic.

7.5 Demo Instalace

Demo instalace obou aplikací byla provedena a obě jsou provozuschopné. Lokální katalog pro potřeby Masarykovy univerzity pracuje na adrese: **http://lkod.fi.muni.cz/admin**, API běží na **http://lkod.fi.muni.cz/lkod-api**. Součástí aplikace je také sparql databázový server, který je spuštěn na adrese: **http://lkod.fi.muni.cz:3030/lkod**. Na této adrese je dostupný seznam všech datových sad, které si národní katalog může automaticky stahovat. V tomto souboru jsou distribuovány pouze publikované datové sady.

Popis procesu tvorby dockerfile souboru je popsán v kapitole 4.

7.6 Migrace sc02.fi.muni.cz a opendata.praha.eu

7.6.1 Migrace sc02.fi.muni.cz

Migrace serveru sc02 proběhla dle předchozí analýzy. Původní datové sady bez předvyplnění hodnot nelze zmigrovat. To znamená, že z 75 veřejně dostupných datových sad nelze ani zkonvertovat bez dalších úprav. Datové sady však do lokálního katalogu zmigrovány byly a jsou nachystány k úpravě kurátorem nebo správcem dat. Po zapracování kurátorem mohou být datové sady dále používány v souladu s otevřenou formální normou datové sady. Při migraci byly doplněny výchozí parametry frekvence hodnotou **IRREG**[1]. Hodnoty licence byly naplněny s informacemi, že datové sady neobsahují osobní údaje, RÚIAN hodnota nebyla vyplněna u datových sad, které ji neobsahovaly.

V tomto stavu lze migraci považovat za úspěšnou. Data byla konvertována do specifikace DCAT-AP-CZ 2.0.1 a jsou uložena v lokálním katalogu od OICT.

7.6.2 Migrace opendata.praha.eu

Z celkových 336 veřejně dostupných datových sad se podařilo úspěšně zmigrovat bez větších úprav 125 datových sad. S nutnými úpravami je zbylých 211 datových sad, u kterých je nutné doplnit alespoň povinné parametry dle specifikace DCAT-AP-CZ. U většiny datových sad se jedná o chybějící atributy klíčových slov, popis, zaktualizovat odkazy k jednotlivým souborům. V případě datových sad pro Prahu by bylo možné předvyplnit označení RÚIAN pro kraj Praha, ale i přesto by bylo nutné hodnoty zkontrolovat. V rámci frekvence byla opět předvyplněna hodnota **IRREG**.

Nepodařilo se zmigrovat dvě datové sady, které obsahují více než 100 distribucí datové sady. Migrace takovýchto sad nesmí probíhat formátem přímé konverze, ale měla by být provedena konverzí distribucí datové sady do samotné datové sady. Ta je v plánu v rámci prvního rozšíření, které je popsáno v podsekci 8.1.1.

V tomto stavu je nutné migraci manuálně zkontrolovat a projít hodnoty RÚIAN a licence datových sad. Migrace se v tomto případě také dá považovat za úspěšnou, téměř všechny datové sady byly zkon-

1. IRREG - neurčeno přesně datum aktualizace

vertovány do nového formátu a byly přeneseny do aplikace lokálního katalogu.

8 Možná rozšíření

8.1 Rozšíření migrační aplikace

8.1.1 Transformace distribucí datových sad na datové sady

Při migraci datových sad ze serveru opendata.praha.eu bylo zjištěno, že ve velmi výjimečných případech existují sady s více než desítkou distribucí konkrétní datové sady. Taková datová sada je sice dle OFN validní, dle směrnice DCAT-AP-CZ to v pořádku není. [41]

Při konzultaci se zástupci z Ministerstva vnitra došlo ke shodě, že datová sada obsahující množství distribucí by měla být zkonvertována do formátu, kdy existuje jedna datová sada zastřešující všechny distribuce. Tyto distribuce by měly být přetvořeny do formátu samostatných datových sad.

S OICT proběhla domluva o implementaci této části v rámci další údržby. Formát podmínek pro migraci takovýchto datových sad je momentálně v řešení a bude zadokumentován v příslušném "issue"[1] v aplikaci Github[2].

8.1.2 Výchozí nastavení pro jednotlivé datové sady

Aplikace momentálně nabízí předvyplnění různých hodnot pro splnění normy DCAT-AP-CZ. Na výběr je z atributů: periodicita aktualizace, licence distribucí datových sad a hodnota RÚIAN. Nastavení je v tento moment pouze globální pro co nejjednodušší a co možná nejrychlejší přesun dat.

V budoucnu by migrační aplikace mohla nabízet předvyplnění hodnot v rámci konkrétních datových sad - u konkrétních sad by tak bylo možné vybrat o jakou licenci při distribuci se má jednat a nebo pro jaké geografické území je datová sada platná.

---

1. Issue - forma připomínek v rámci verzovací aplikace GitHub
2. https://github.com/opendata-mvcr/lkod/issues/8

8.1.3 Migrace z jiných aplikací

Nynější verze aplikace podporuje migraci z aplikace CKAN. V případných budoucích rozšířeních by aplikace mohla podporovat další lokální katalogy, mezi které patří DKAN, GeoSearch a další.

8.2 Rozšíření lokálního katalogu

8.2.1 Možnost přidávání jednotlivých organizací přes GUI

Lokální katalog je open-source aplikací, což znamená, že do ní může přispívat kdokoliv. Pro rozšíření lokálního katalogu by bylo vhodné navrhnout systém vytváření nových organizací přes GUI rozhraní. Momentálně lze do lokálního katalogu přidat organizaci pouze manuálně pomocí tří SQL dotazů typu INSERT. Takové řešení spoléhá na práci správce serveru a na jeho schopnosti SQL jazyka. V rámci provozuschopnosti aplikace by bylo vhodnější, kdyby správa organizací byla tvořena přes GUI. Takovou organizaci by mohl vytvořit pouze speciální uživatel, tím by mohl být právě například správce.

Rozšíření by tedy muselo obsahovat - rozšíření systému práv o tzv. super uživatele, který má možnost vytvářet organizace. Dále pak možnost přiřadit oprávnění uživateli pro tvorbu organizací ve struktuře organizací. V neposlední řadě možnost tvorby organizací přes uživatelské rozhraní.

8.2.2 Instalační proces přes webovou aplikaci

Toto rozšíření je ve své podstatě nádstavbou rozšíření z předchozí sekce. Aplikace lokálního katalogu musí být nyní nainstalována správcem za pomocí docker kontejnerů. Dále pak musí ručně vytvořit organizaci a přiřadit k ní uživatele. Za účelem zjednodušení úvodní instalace by bylo na místě utvořit instalační formulář. Takový formulář by obsahoval identifikační údaje o organizaci a registrační formulář pro prvotního uživatele. Správce by tak webovou aplikaci nainstaloval vyplněním jednoho formuláře. Proces instalace by tak byl zjednodušen a byl by schopen instalaci provést i méně zdatný technik (případně správce dat po řádném zaučení).

9 Závěr

Spuštění aplikace a dokončení procesu se podle mého názoru zdárně povedlo. Aplikace je v prostředí Fakulty informatiky Masarykovy univerzity připravena, spuštěna a je nachystána na případné další projekty a práce.

LKOD od OICT spolu s migračním skriptem posloužil k částečně úspěšné migraci serverů sc02.fi.muni.cz a opendata.praha.eu. Jeho funkčnost byla úspěšně ověřena, byla ověřena funkčnost lokálního katalogu a byly odladěny objevené chyby. Migrační proces pomůže s přenosem dat z aplikace CKAN do aplikace LKOD od OICT. Nemůže ale nahrazovat data, která vznikla nově s aktualizovaným doporučením DCAT-AP-CZ. Pro tyto účely migrační aplikace nabízí možnost předvyplnění hodnot. Aplikace LKODu i migrační skript mají prostor pro vylepšení. Ty byly popsány v kapitole 8.

Důležitým bodem bylo navázání úspěšné spolupráce s komunitou otevřených dat, získání poznatků o fungování otevřených dat v ČR. Laboratoř servisních systémů může dále rozvíjet otevřená data v České republice ve spolupráci s OICT, napomůže směrovat další rozvoj aplikace lokálního katalogu. V neposlední řadě je samozřejmě vhodné zmínit, že je aplikace připravena k nasazení do dalších institucí, v čemž se může Masarykova univerzita angažovat a napomoci tak otevřenosti institucí v České republice.

Bibliografie

1. WALLETZKÝ, L.; ROMANOVSKÁ, F.; TOLLI, A. M.; GE, M. Research Challenges of Open Data as a Service for Smart Cities: In Proceedings of the 10th International Conference on Cloud Computing and Services Science. [N.d.]. ISBN 978-989-758-424-4.
2. KOMÍNEK, J. *Dopravní nehody* [online] [cit. 2021-11-22]. Dostupné z: `https://data.brno.cz/datasets/298c37feb1064873abdccdc2a10b605f_0/about`.
3. ODDĚLENÍ DAT analýz a evaluací, Magistrát města Brna. *Hlasování zastupitelstva: Statutární město Brno - Magistrát města Brna - Otevřená data* [online]. 2021 [cit. 2021-11-22]. Dostupné z: `https://data.brno.cz/documents/f3c663acc9c047cfa898afea94ea3711/about`.
4. Dostupné také z: `http://www.gsoftwis.sk/Downloads/MoneyS3/Pracujeme_s_Money_S3.pdf`.
5. MÁJEK, O.; NGO, O.; CHLOUPKOVÁ, R.; JARKOVSKÝ, J.; PAVLÍK, T.; KOMENDA, M.; DUŠEK, L. Dokumentace k epidemiologickému modelu ÚZIS ČR pro dlouhodobé simulace [online]. 2021 [cit. 2021-11-02]. Dostupné z: `https://share.uzis.cz/s/cmFHjc4jbqPBAER`.
6. Zákon č. 106/1999 Sb. Zákon o svobodném přístupu k informacím [online]. 1999 [cit. 1999-06-08]. Dostupné z: `https://www.zakonyprolidi.cz/cs/1999-106#p2`.
7. HALL, W.; SHADBOLT, N.; TIROPANIS, T.; O'HARA, K.; DAVIES, T. [online] [cit. 2021-11-02]. Dostupné z: `https://socialtechtrust.org/wp-content/uploads/2017/11/Open-Data-and-Charities.pdf`.
8. KLÍMEK, Jakub. *Důležité pojmy v oblasti otevřených dat* [online] [cit. 2021-03-23]. Dostupné z: `https://opendata.gov.cz/informace:d%C5%AFle%C5%BEit%C3%A9-pojmy-v-oblasti-otev%C5%99en%C3%BDch-dat`.

9. KLÍMEK, Jakub. *Důležité pojmy v oblasti otevřených dat* [online] [cit. 2021-03-23]. Dostupné z: https://opendata.gov.cz/cinnost:priprava-katalogizacniho-zaznamu#z%C3%A1znamy_o_datov%C3%A9_sad%C4%9B_doporu%C4%8Den%C3%BDch_datov%C3%BDch_sad.

10. [N.d.]. Dostupné také z: https://data.gov.cz/ofn/pou%C5%BEit%C3%AD-poskytovateli/.

11. DVOŘÁK, M.; SPÁL, R.; MAREK, J.; KLÍMEK, J. [online]. 2020 [cit. 2021-11-16]. Dostupné z: https://ofn.gov.cz/ud%C3%A1losti/2020-07-01/.

12. KLÍMEK, J. Technické standardy pro datové sady na stupni otevřenosti 3 [online]. 2020 [cit. 2021-11-20]. Dostupné z: https://opendata.gov.cz/standardy:technicke-standardy-pro-datove-sady-na-stupni-3.

13. Stupně otevřenosti otevřených dat a česká legislativa [online]. 2020 [cit. 2021-11-20]. Dostupné z: https://opendata.gov.cz/informace:stupn%C4%9B-otev%C5%99enosti-datov%C3%BDch-sad.

14. KUBÁŇ, M. [online]. 2021 [cit. 2021-11-11]. Dostupné z: https://opendata.gov.cz/cinnost:stanoveni-podminek-uziti.

15. MÍŠEK, Jakub. Nové povinnosti pro obce, kraje a orgány státní správy v oblasti otevřených dat [online]. 2021 [cit. 2021-11-18]. Dostupné z: https://data.gov.cz/%C4%8Dl%C3%A1nky/nov%C3%A9-povinnosti-pro-obce-kraje-a-org%C3%A1ny-st%C3%A1tn%C3%AD-spr%C3%A1vy-v-oblasti-otev%C5%99en%C3%BDch-dat.

16. Zákon č. 261/2021 Sb. Zákon, kterým se mění některé zákony v souvislosti s další elektronizací postupů orgánů veřejné moci [online]. 2021 [cit. 2021-09-07]. Dostupné z: https://www.zakonyprolidi.cz/cs/2021-261.

17. KLÍMEK, Jakub. *Vytvoření a správa záznamu o datové sadě v Národním katalogu otevřených dat (NKOD)* [online]. 2021 [cit. 2021-11-12]. Dostupné z: https://opendata.gov.cz/cinnost:sprava-katalogizacniho-zaznamu-v-nkod.

18. Vytvoření publikačního plánu [online]. 2020 [cit. 2021-11-15]. Dostupné z: https://opendata.gov.cz/standardy:vytvoreni-publikacniho-planu.

19. Kurátor dat [online]. 2020 [cit. 2021-11-15]. Dostupné z: https://opendata.gov.cz/role:kurator-dat.

20. ČESKÉ REPUBLIKY, Ministerstvo vnitra. *Standardy publikace a katalogizace otevřených dat VS ČR* [online]. 2015 [cit. 2021-11-16]. Dostupné z: https://opendata.gov.cz/_media/standardy_publikace_a_katalogizace_otevrenych_dat_vs_cr.pdf.

21. [Online]. 2015 [cit. 2021-11-16]. Dostupné z: https://www.mfcr.cz/cs/verejny-sektor/rizeni-a-kontrola-verejnych-financi/otevrena-data-ministerstva-financi/otevrena-data-ministerstva-financi-prakt-20534.

22. [Online] [cit. 2021-11-29]. Dostupné z: https://operatorict.cz/zakladni-info-o-nas/.

23. Dostupné také z: http://docs.ckan.org/en/2.9/user-guide.html.

24. Dostupné také z: http://geoserver.org.

25. Dostupné také z: http://docs.ckan.org/en/2.9/user-guide.html#what-is-ckan.

26. Dostupné také z: https://opendata.praha.eu.

27. NUFFELEN, Bert Van. *DCAT Application Profile for data portals in Europe: Version 2.0.1* [online]. 2020 [cit. 2021-11-08]. Dostupné z: https://joinup.ec.europa.eu/sites/default/files/distribution/access_url/2020-06/e4823478-4458-4546-9a85-3609867ad089/DCAT_AP_2.0.1.pdf.

28. ALBERTONI, R.; BROWNING, D.; COX, S.; BELTRAN, A. G.; PEREGO, A.; WINSTANLEY, P. *Data Catalog Vocabulary (DCAT) - Version 2* [online]. 2020 [cit. 2021-10-25]. Dostupné z: https://www.w3.org/TR/vocab-dcat-2/.

29. MOUAT, Adrian. *Using Docker*. O'Reilly Media, Inc., 2015. ISBN 978-1-491-91576-9.

30. DUCHARME, B. *Learning SPARQL: Querying and Updating with SPARQL 1.1*. O'Reilly Media, Inc., 2013. ISBN 978-1-449-37143-2.

31. LUTZ, Mark. *Programming Python*. 4th ed. O'Reilly Media, Inc., 2011. ISBN 978-0-596-15810-1.

32. GRINBERG, Miguel. *Flask Web Development*. 1st ed. O'Reilly Media, Inc., 2014. ISBN 978-1-449-37262-0.

33. [Online]. 2021 [cit. 2021-01-11]. Dostupné z: https://ofn.gov.cz/rozhran%C3%AD-katalog%C5%AF-otev%C5%99en%C3%BDch-dat/2021-01-11/.

34. *Apache Jena Fuseki* [online]. 2021 [cit. 2021-11-29]. Dostupné z: https://jena.apache.org/documentation/fuseki2/.

35. *The Eclise RDF4J Framework* [online] [cit. 2021-11-29]. Dostupné z: https://rdf4j.org/about/.

36. *Virtuoso OpenSource Edition Introduction* [online]. 2021 [cit. 2021-11-29]. Dostupné z: http://vos.openlinksw.com/owiki/wiki/VOS/VOSIntro.

37. *Oxigraph* [online]. 2021 [cit. 2021-11-29]. Dostupné z: https://github.com/oxigraph/oxigraph.

38. *GraphDB Free Documentation* [online]. 2021 [cit. 2021-11-29]. Dostupné z: https://graphdb.ontotext.com/documentation/free/.

39. *Templates* [online]. 2010 [cit. 2021-01-12]. Dostupné z: https://flask.palletsprojects.com/en/2.0.x/templating/.

40. DATA, ČTU Open. *Cenový barometr: Český telekomunikační úřad* [online]. 2018 [cit. 2021-12-01]. Dostupné z: https://data.gov.cz/datov%C3%A1-sada?iri=https%3A%2F%2Fdata.gov.cz%2Fzdroj%2Fdatov%C3%A9-sady%2Fhttp---data.ctu.cz-api-3-action-package_show-id-55ba0bfd-e2c9-41e0-9acd-a20b526b5399.

41. KLÍMEK, Jakub. Špatné dělení dat do distribucí datové sady [online]. 2020 [cit. 2021-12-01]. Dostupné z: https://opendata.gov.cz/%C5%A1patn%C3%A1-praxe:%C5%A1patn%C3%A9-d%C4%9Blen%C3%AD-distribuc%C3%AD.

A Výstup aplikačního rozhraní CKAN


```
{
"help": "http://sc02.fi.muni.cz:80/api/3/action/help_show?name=
    package_show",
"success": true,
"result": {
    "license_title": null, "maintainer": null, "
        relationships_as_object": [],
    "private": false, "maintainer_email": null, "num_tags": 0, "id":
        "...",
    "metadata_created": "...", "metadata_modified": "...",
    "author": null, "author_email": null, "state": "active",
    "version": null, "creator_user_id": "...",
    "type": "dataset",
    "resources": [{"mimetype": "text/csv", "cache_url": null, "hash":
        "",
        "description": "", "name": "Historie teploty v roce 2009",
        "format": "CSV",
        "url": "http://sc02.fi.muni.cz:80/.../download/teplota_2009.
            csv",
        "datastore_active": true, "cache_last_updated": null,
        "package_id": "...", "created": "...", "state": "active",
        "mimetype_inner": null,
        "last_modified": "...", "position": 0, "revision_id": "...",
        "url_type": "upload", "id": "...", "resource_type": null,
        "size": 9105840}],
    "num_resources": 1, "tags": [], "groups": [], "license_id": null,
    "relationships_as_subject": [],
    "organization": {
        "description": "Data z mestskeho uradu", "created": "...",
        "title": "Mestsky urad", "name": "mestsky-urad",
        "is_organization": true, "state": "active", "image_url": "",
        "revision_id": "...", "type": "organization", "id": "...",
        "approval_status": "approved"
    },
    "name": "historie-teploty-2009", "isopen": false, "url": "upload",
    "notes": null, "owner_org": "...", "extras": [],
    "title": "Historie teploty v roce 2009", "revision_id": "..."
}}
```

B Ukázkový docker-compose.yml pro spuštění aplikace

```
version: "3.3"
services:
  migration:
    container_name: "migration"
    image: skalincz/lkod-python-migration:latest
    networks:
       - dp-network
    ports:
       - 5000
    depends_on:
       - "be"
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.migration.rule=Host(`lkod-migrace.cz`)"
      - "traefik.http.routers.migration.entrypoints=web"
      - "traefik.http.services.migration-svc.loadbalancer.server.port
          =5000"
    environment:
      - FLASK_ENV:production
      - FLASK_DEBUG:False
  be:
    container_name: be
    image: registry.gitlab.com/operator-ict/golemio/code/lkod/lkod-
        backend/development:latest
    networks:
      - dp-network
    ports:
      - 3002:3000
    volumes:
      - /home/lkod-docker/keys:/app/keys
      - /home/lkod-docker/.env.be:/app/.env
    depends_on:
      - database
      - redis
      - fuseki
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.lkod-api.rule=Host(`lkod.fi.muni.cz`) &&
           PathPrefix(`/lkod-api`)"
      - "traefik.http.routers.lkod-api.entrypoints=web"
      - "traefik.http.services.lkod-api-svc.loadbalancer.server.port
          =3000"
  frontend:
    container_name: fe
    image: "lkod-muni-fe:latest"
 #muni-lkod-frontend:latest
    depends_on:
      - be
    ports:
      - 3001:3000
    environment:
```

```
    - REACT_APP_EXTERNAL_FORM_URL:"https://dev.nkod.opendata.cz/
        formul%C3%A1%C5%99/registrace-datov%C3%A9-sady"
    - REACT_APP_BACKEND_URL:"http://lkod.fi.muni.cz/lkod-api"
    - REACT_APP_RETURN_URL:"http://lkod.fi.muni.cz/lkod-api/form-
        data"
    - PUBLIC_URL:"/admin"
  volumes:
    - ./admin-fe-config.js:/app/config.js:ro
    - /home/lkod-docker/.env.fe:/app/.env:ro
  networks:
    - dp-network
  labels:
    - "traefik.enable=true"
    - "traefik.http.routers.frontend.rule=Host(`lkod.fi.muni.cz`) &&
        PathPrefix(`/admin`)"
    - "traefik.http.routers.frontend.middlewares=frontend-
        stripprefix"
    - "traefik.http.middlewares.frontend-stripprefix.stripprefix.
        prefixes=/admin"
    - "traefik.http.routers.frontend.entrypoints=web"
    - "traefik.http.services.frontend-svc.loadbalancer.server.port
        =3000"
database:
  container_name: psql
  image: 'bitnami/postgresql:12.8.0'
  networks:
    - lkod
  ports:
    - "5432:5432"
  env_file: /home/lkod-docker/.env.db
  networks:
    - dp-network
redis:
  container_name: redis
  image: redis:6.0-alpine
  depends_on:
    - database
  networks:
    - dp-network
traefik:
  container_name: traefik
  image: "traefik:v2.4"
  command:
    - "--log.level=DEBUG"
    - "--api.insecure=true"
    - "--providers.docker=true"
    - "--providers.docker.exposedbydefault=false"
    - "--entrypoints.web.address=:80"
  ports:
    - "80:80"
    - "8080:8080"
  networks:
    - dp-network
  volumes:
    - "/var/run/docker.sock:/var/run/docker.sock:ro"
  depends_on:
    - be
```

```
      - frontend
      - migration
  fuseki:
    container_name: sparql
    image: registry.gitlab.com/operator-ict/golemio/extras/fuseki2/
        master:4.2.0
    command: --conf=/etc/fuseki/config.ttl --passwd=/etc/fuseki/users
        --auth=basic
    networks:
      - dp-network
    volumes:
      - ./fuseki-db:/fuseki/databases
      - ./fuseki-db/fuseki-users:/etc/fuseki/users:ro
      - ./fuseki-db/fuseki-config.ttl:/etc/fuseki/config.ttl
    ports:
      - 3030:3030
    environment:
      LOGSPOUT: ignore
      JAVA_OPTIONS: "-Xmx1048m -Xms1048m"
    deploy:
      replicas: 1
      labels:
        traefik.enable: 'true'
        traefik.http.routers.fuseki.rule: Host(`lkod.fi.muni.cz`)
        traefik.http.routers.fuseki.entrypoints: 'websecure'
        traefik.http.services.fuseki-svc.loadbalancer.server.port:
            '3030'
        swarmpit.service.deployment.autoredeploy: 'true'
        traefik.http.routers.fuseki.middlewares: cors-sparql
        traefik.http.middlewares.cors-sparql.headers.
            accesscontrolallowmethods: 'GET,OPTIONS,PUT'
        traefik.http.middlewares.cors-sparql.headers.
            accesscontrolalloworiginlist: '*'
        traefik.http.middlewares.cors-sparql.headers.
            accesscontrolmaxage: 100
        traefik.http.middlewares.cors-sparql.headers.addvaryheader: '
            true'
      placement:
        constraints:
          - node.labels.type-back==true
      resources:
        reservations:
          cpus: '0.05'
          memory: 200M
        limits:
          cpus: '1'
          memory: 1500M
    depends_on:
      - database

networks:
  dp-network:
    driver: bridge

configs:
  traefik:
    file: ./traefik.yaml
```

```
secrets:
  fuseki-users:
     file: ./fuseki-db/fuseki-users
  lkod-key-pub:
    file: ./lkod-key.pub
  lkod-key:
    file: ./lkod-key
```

C Zdrojové kódy

Zdrojové kódy aplikace v jazyce Python jsou přiloženy jako archiv v elektronické podobě. Součástí archivu je také instance aplikace lokálního katalogu a konfigurace docker-compose souboru a jednotlivých configuračních souborů potřebná pro snadné spuštění aplikace.

Struktura archivu je následující:

clm - složka CLM obsahuje zdrojové kódy migrační aplikace, včetně konfiguračních souborů a obrázků

lkod - složka obsahuje konfigurační soubor docker-compose.yml, sloužící ke spuštění jednotlivých komponent aplikace lokálního katalogu. Součástí je základní konfigurace .env souborů.

muni-lkod-frontend - složka obsahuje zdrojové kódy frontendové aplikace lokálního katalogu a referenci na gitlab repozitář frontendové aplikace, upravené pro účely spuštění v prostředí Masarykovy univerzity

datasets - součástí archivu je i adresář obsahující dva soubory zobrazující rozdíly mezi původním výstupem ze CKAN aplikace a zmigrovanou datovou sadou do formátu platného dle DCAT-AP-CZ 2.0.1.