Masarykova univerzita
Fakulta informatiky

Výzkum Open Source Smart City platforem

Diplomová práca

Marek Vaľko

Brno, jeseň 2019

Masarykova univerzita
Fakulta informatiky

Výzkum Open Source Smart City platforem

Diplomová práca

Marek Vaľko

Brno, jeseň 2019

Na tomto mieste sa v tlačenej práci nachádza oficiálne podpísané zadanie práce a vyhlásenie autora školského diela.

Vyhlásenie

Vyhlasujem, že táto diplomová práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Marek Vaľko

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

Poďakovanie

V prvom rade by som sa chcel poďakovať vedúcemu tejto práce pánovi Ing. Leonardovi Walletzkému, Ph.D. za profesionálny prístup a cenné rady, ktoré mi pomohli pri tvorbe tejto práce. Ďakujem za jeho trpezlivosť pri tvorbe textu práce, vysvetlenia nejasností a pojmov, s ktorými som prišiel do kontaktu a za pomoc a usmernenia pri práci a komunikácii s ľudmi. Ďalej by som chcel poďakovať pánovi Ing. Michalovi Bačovskému za cenné rady pri komunikácii s ľuďmi a za možnosť byť súčasťou projektu inteligentného mesta Žďár nad Sázavou.

Zhrnutie

Táto práca zhrňuje poznatky o možnostiach prevádzkovania lokálnych dátových katalógov v mestách za účelom publikovania mestských dát. Tieto dáta splňujú podmienky kladené princípom Otvorených dát a vybrané katalógy zasa podmienky kladené Európskou úniou a Českou republikou na prevádzku lokálnych katalógov, ktoré sú v práci tiež vysvetlené. Sú v nej porovnané viaceré platformi a z nich následne jedna vybratá, ktorej funkcie sú popísané podrobnejšie. Okrajovo sú tu aj spomenuté niektoré vizualizačné nástroje.

Kľúčové slová

Smart City, Open Data, Katalóg dát, Data portál, Open Source, Dspace, CKAN, Dashboard, NKOD, DCAT

Obsah

1 Úvod **1**

2 Úvod do konceptu Smart Cities **3**
2.1 *Otvorené dáta* 4
2.2 *Európsky dátový portál* 5
2.3 *Národný katalóg otvorených dát* 5

3 Správa dát **7**
3.1 *Duraspace* 7
3.1.1 Fedora 7
3.1.2 Dspace 8
3.2 *CKAN* 9
3.3 *Komerčné riešenia* 10
3.3.1 INVIPO 11
3.3.2 ASD Software 12

4 Praktická časť **13**
4.1 *Zber požiadaviek* 13
4.2 *Formát DCAT* 13
4.2.1 DCAT profil Českej republiky 14
4.3 *Porovnanie a výber systému* 17
4.4 *Popis CKANu* 18
4.4.1 Štruktúra CKANu 18
4.4.2 Náhľady dát 19
4.4.3 CKAN API 19
4.4.4 Rozšírenie Datastore 21
4.4.5 Datastore API 22
4.4.6 Rozšírenie Datapusher 23
4.4.7 Rozšírenie DCAT 24
4.5 *Implementácia CKANu* 25
4.5.1 Inštalácia 25
4.5.2 Konfigurácia 26
4.5.3 Spustenie rozšírenia Datastore 28
4.5.4 Vytváranie náhľadov z dát 28
4.5.5 Podpora pre DCAT 29

4.6 *Práca s dátovými sadami* 32

5 Vizualizácia dát **35**
5.1 *Nástroje* 35
5.1.1 Apache superset 36
5.1.2 Redash 37
5.1.3 Metabase 38
5.1.4 RAWGraphs 40
5.1.5 Chart studio 41
5.2 *Knižnice* 41
5.2.1 Príklady knižníc 42

6 Záver **45**

A Elektronické prílohy **55**

B DCAT - Základ pre implementáciu profilu **57**

1 Úvod

Mestá sa snažia produkovať vhodné podmienky pre život ich občanov. V minulosti sa rozhodnutia robili pomocou pocitov a neúplných informácii, ktoré sa niekedy podarilo od občanov získať. Výsledky týchto rozhodnutí neboli vždy najlepšie, no bez chýbajúcich dát nebolo možné presne zhodnotiť, kde nastalo zlyhanie.

S nástupom moderných technológií prišiel aj nový spôsob riešenia problémov. Použitie týchto technológii malo za následok väčšiu produkciu presnejších a kompletných dát, ako aj ich jednoduchšie spracovanie a interpretáciu. Vďaka technológiám bol vytvorený nový pohľad na riešenie problémov a bol nazvaný Smart cities.

V Smart cities sú základom dáta. Tieto dáta je potrebné dostať do jednotnej a zrozumiteľnej podoby. Potom je možné dáta zbierať a uchovávať. Následne je možné s dátami ďalej pracovať, vytvárať štatistiky a iné výsledky, a tie následne jesne a výstižne prezentovať.

Použitie konceptu Smart cities delí mestá na tie, ktoré sú riadené pocitmy a odhadom a na tie, ktoré sú riadené dátami. Ak sú dáta k dispozícii, je možné robiť uvážené a dátami podložené rozhodnutia. V prípade chybného rozhodnutia sú k dispozícií dáta, ktoré vysvetlia, ako sa k takému riešeniu došlo a je možné sa vyhnúť podobným chybám v budúcnosti. Pri rozhodnovaní pomocou pocitov nič také k dispozícii nemáme a nadväzujúce rozhodnutia budú opäť len na základe pocitov.

Dáta je potrebné niekde ukladať. V tejto práci budú spomenuté možné prístupy a platfomy na riešenie tohto problému a porovnané ich funkcie. Následne bude vybratá vhodná platforma, ktorá bude odskúšaná v reálnej prevádzke.

Ďalší prehľad bude obsahovať nástroje, ktoré budú slúžiť na samotné spracovanie a zobrazovanie dát. Dáta by mali byť zobrazované vo forme grafov a tie následne organizované do podoby dashboardov.

V neposledom rade bude potrebné porozumieť, ako fungujú Otvorené dáta a centrálny cloud otvorených dát v Českej republike. Z toho by malo vyplynúť, ako a aké dáta môžu byť na dátovom portáli publikované, či je vôbec možné prevádzkovať takýto portál lokálne v mestách, prípadne aké podmienky sú naň kladené.

2 Úvod do konceptu Smart Cities

Všetky mestá a obce sa pokúšajú vytvárať vhodné a produktívne podmienky pre život ich občanov. Snažia sa načúvať ich problémom a následne ich riešiť. Donedávna to bolo možné iba priamym kontaktom na besedách a stretnutiach, prípadne listovou komunikáciou. To čo sa zistiť nepodarilo, muselo byť doplnené odhadom. S týmito nie úplnými podkladmi sa pracovalo ďalej a niekedy sa dospelo až ku cieľu a riešeniu problému, inokedy to ostalo stáť na mŕtvom bode pre nedostatok informácií a záujmu.

Nástupom moderných technológií sa otvárajú nové možnosti na analýzu a riešenie problémov. Tieto nové nástroje môžu pomáhať pri zbere a analýze chýbajúcich informácií. Nové technológie a služby môžu ďalej pomôcť pri vývoji a organizácií miest, pri riešení problémov s rastom ale aj poklesom populácie, zvyšovaním znečistenia, dokonca aj pri riešení väčších problémov týkajúcich sa celých regiónov a oblastí[1].

Existuje niekoľko definícií pre pojem Smart cities. V podstate záleží od používateľa, v akom kontexte tento termín použije. Ja týmto pojmom budem označovať jednak cestu alebo proces, ktorý používa moderné informačné a komunikačné technológie na zlepšenie životnej a pracovnej úrovne v mestách, ako aj konkrétne mestá, ktoré tento proces využili alebo využívajú. Ďalej sa zhodujem s definíciou, že proces Smart cities je možné vystihnúť nasledujúcimi troma výrazmi: zber, komunikácia a analýza. Smart city zbiera informácie o sebe a o stave, v akom sa nachádza. Pomocou senzorov a ďalších nástrojov tieto dáta posiela a komunikuje na miesto, kde môžu byť ďalej spracované. Potom nasleduje analýza dát, ktoré môže vysvetliť, čo sa v meste deje a amôže načrtnúť, čo sa diať bude [1].

Základom pre budovanie Smart cities sú dáta. Mestá a ich súčasti už aj v dnešnej dobe produkujú množstvo dát. Niektoré z nich sú k dispozícii aj verejnosti, či už dobrovoľne alebo povinne určené zákonom. No väčšinou sú takéto dáta vo forme zhrnutí alebo štatistík, takže na ďalšie podrobné analýzy nie sú príliš vhodné. Ostatné dáta, ktorých je zvyčajne väčšina, sú však len ukladané a archivované. Transformovanie dát do publikovateľnej podoby je pre pracovníka úloha

nadbytočná a pokiaľ to nemá v náplni svojej práce, venovať sa tomu zrejme nebude.

Ak bude mať takýto prístup k publikovaniu dát väčšina, nebude možné s týmito dátami pracovať. Každý si vytvorí svoje štatistiky o tom, aké má výsledky a či prosperoval. Množstvo ďalších užitočných údajov však zostane skrytých. Nebude možné vytvárať kombinácie a korelácie medzi dátami z rôznych oblastí a tým pádom zistiť, či sa nejako neovplyvňujú alebo či by nemohli spolupracovať a nájsť produktívnejšie riešenia.

Ďalšou otázkou je podoba týchto dát. Exituje veľa riešení a možnosti, ako dáta uchovávať a spracovávať. Každý si u seba môže vytvoriť vlastnú logickú reprezentáciu týchto dát. To môže viesť k tomu, že informácia z dvoch rôznych oblastí, ktorá v skutočnosti znamená to isté, môže mať úplne opačný význam.

Odpovede na obe tieto otázky určia, či dáta budú využívané a analyzované na produkovanie výsledkov v oblasti Smart cities. Záleží teda hlavne na ich dostupnosti a forme. Prvý problém s dostupnosťou dát rieši koncept Otvorené dáta. Druhý problém s ich podobou a formou riešia rôzne nariadenia, definície a slovníky, ktoré popisujú, ako majú také dáta vyzerať, v akej podobe majú byť dostupné a zdieľateľné.Tieto pravidlá sú obsiahnuté v katalógoch dát, ktoré takéto dáta zhromažďujú. Oba tieto prístupy a definície budú vysvetlené v nasledujúcich častiach a ich vysvetlenie je základom pre prorozumenie ďalších termínov a procesov, ktoré vyplynuli z výskumu možných platforiem.

2.1 Otvorené dáta

Otvorené dáta sú dáta, ktoré môžu byť voľne používané a zdieľané bez obmedzenia dôvodu, času a priestoru. Sú to dáta publikované pod otvorenou licenciou za cieľom prospechu a zlepšenia. Mali by byť technicky aj legálne dostupné[2].

Technická dostupnosť znamená, že medzi užívateľom a dátami nesmie byť žiadna technická prekážka, ktorá znemožňuje alebo obmedzuje prístup k nim. Príkladom môže byť verejne vytlačený dokument. Prístup k nemu je obmedzený, pretože nie je možné sa k nemu dostať odkiaľkoľvek. Preto by otvorené dáta mali byť v digitálnej podobe a

prístupné cez internet vo formáte, ktorému počítače rozumejú. Ďalším príkladom technického obmedzenia môže byť napríklad žiadosť o registráciu alebo obmedzenie počtu prístupov [3].

Legálna dostupnosť znamená, že prístup a použitie dát nie je nijako trestané ani inak právne obmedzované [2]. Legálnu dostupnosť otvorených dát zabezpečujú otvorené licencie. Otvorené licencie sú zoznamy pravidiel a podmienok, ktoré umožňujú používanie dát, pokiaľ sú podmienky licencie dodržané [4]. Otvorené licencie dovoľujú hocikomu použiť dáta bezplatne a ďalej s dátami pracovať a modifikovať ich s minimálnym alebo žiadným obmedzením. Minimálne obmedzenie môže predstavovať napríklad spomenutie alebo uznanie autora pri zdieľaní alebo prezentovaní dát. Medzi najznámejšie licencie v oblasti otvorených dát patria licencie Creative Commons alebo Open Data Commons [5].

2.2 Európsky dátový portál

Európsky dátový portál zbiera metadáta a informácie z verejného sektora, ktoré sú dostupné z dátových portálov v jednotlivých krajinách Európy [6]. Strategický cieľ tohto portálu je zlepšiť dostupnosť a hodnotu otvorených dát v Európe. Sústredí sa na celý proces práce s dátami od spôsobu ich publikácie až po možnosti ich opätovného použitia.

Dátový portál má za úlohu vyhľadávanie dát a zaraďovanie dát do definovaných kategórii. Určuje spôsoby, ako majú byť dáta publikované. Ďalej prezentuje reálne využitie dát a jeho benefity. V neposlednej rade poskytuje výukové materiály na tému otvorených dát.

Európsky dátový portál popri získavaní dát z národných katalógov zároveň kontroluje hodnoty metadát a upozorňuje vlastníkov na chyby v nich. Tento postup zlepšuje celkovú kvalitu otvorených dát publikovaných v Európe.

2.3 Národný katalóg otvorených dát

Národný katalóg otvorených dát verejnej správy Českej republiky (ďalej používam ako Národný katalóg) je centrálny katalóg otvorených dát v Českej republike. Národný katalóg zhromažďuje metadáta

o dátových sadách publikovaných na dátových portáloch v Českej republike. Obsahuje informácie o datasetoch aj samotných dátach a odkazy na stiahnutie týchto dát z portálov jednotlivých poskytovateľov dát [7]. Poskytovateľom dát môžu byť napríklad centrálne orgány, ministerstvá, kraje, obce a ďalšie subjekty verejnej správy [8].

Registrácia dátovej sady alebo celého lokálneho katalógu prebieha pomocou formulárov [9], kde je potrebné zadať popisné informácie a následne po kontrole bude dátova sada alebo katalóg zobrazovaný v Národnom katalógu.

3 Správa dát

Táto kapitola obsahuje príklady platforiem, ktoré slúžia na správu dát. Takýto softvér by mal dodržovať princíp Otvorených dát, mal by byť jednoduchý na ovládanie pre užívateľov a zároveň by mal byť jednoduchý aj na jeho správu a údržbu pre administrátorov. Výhodou môže byť možnosť rozšírenia softvéru, pridanie chýbajúcej alebo uvoľnenie obmedzujúcej funkcionality bez väčšieho zásahu a modifikácie jeho jadra. Taktiež by mal mať k dispozícií nejaké API, ktoré by umožňovalo automatizáciu práce s dátami. V neposledom rade by mal byť schopný komunikovať s nadradenými katalógmi, s Národným katalógom aj Európskym portálom.

V tejto kapitole budú predstavené niektoré open source nástroje spoločne so stručným vysvetlením ich základných funkcií. Ďalej tu budú spomenuté aj niektoré komerčné riešenia a zdôvodnenie, prečo je vo väčšine prípadov výhodnejšie sa k takýmto riešeniam neviazať.

3.1 Duraspace

Duraspace je nezisková organizácia, ktorá sa zaoberá vývojom otvorených technológií v oblasti uchovávania digitálneho obsahu [10]. V oblasti digitálnych repozitárov vyvíjajú open source softvér Fedora a Dspace, ktoré predstavím v nasledujúcich kapitolách.

3.1.1 Fedora

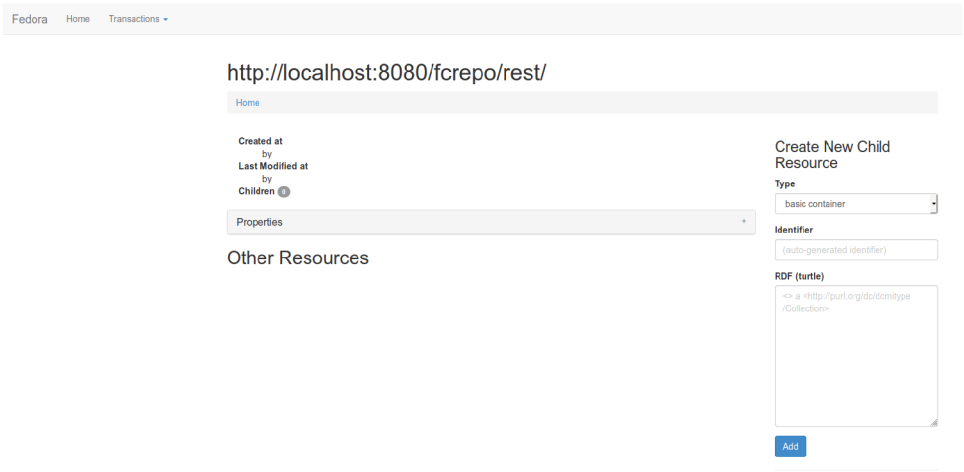
Fedora (Flexibile Extensible Digital Object Repository Architecture) [1] je open source repozitár na správu digitálneho obsahu, vhodná predovšetkým pre digitálne knižnice a archívy [11]. Na prácu s digitálnym obashom sa využíva REST API a dokáže ukladať súbory ľubovoľného typu a veľkosti.

Fedora taktiež podporuje prepojené dáta (Linked Data). Prepojené dáta sú štrukturované dáta publikované na webe, dostupné pod URI (Uniform Resource Identifier) odkazom a prepojením na iné takéto

1. Fedora je aj názov operačného systému od spoločnosti Red Hat, Inc. Obe organizácie sa dohodli, že názov Fedora môžu používať súčasne, každý vo svojej oblasti pôsobenia, viď [11].

dáta môžu byť doplnené a obohatené o ďalšie informácie [12]. Na popis a prepájanie týchto dát sa využíva RDF (Resource Description Framework) [13].

V základnej inštalácií slúži fedora najmä na vytváranie, ukladanie, úpravu a mazanie dát a ich metadát. Jej užívateľské rozhranie tvorí jednoduchý restový klient (Obrázok 3.1), ktorým sa dá s dátami manipulovať. Pre užívateľsky prívetivejšie rozhranie existujú ďalšie nástroje, ktoré využívajú fedoru na správu dát a ponúkajú vlastné webové rozhranie. Alternatívou je možnosť vytvoriť si vlastné užívateľské rozhranie, ktoré by fedoru využívalo a pracovalo s ňou pomocou API.



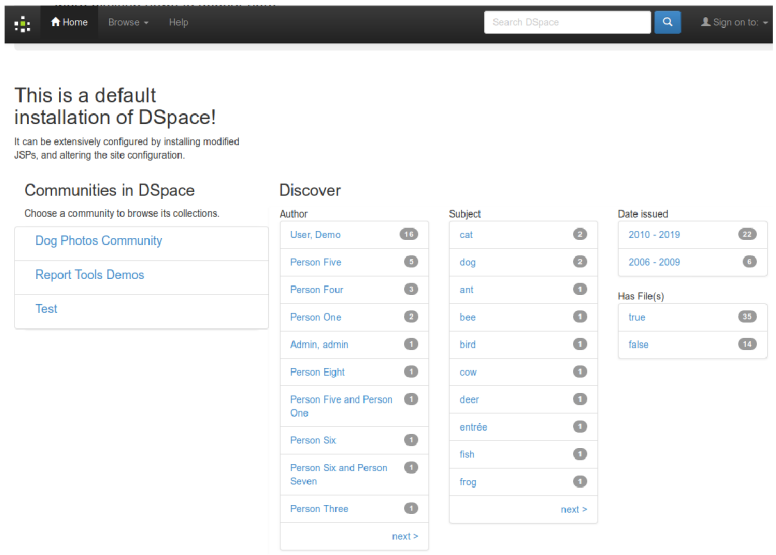
Obr. 3.1: Webové rozhranie restového klienta nástroja Fedora

3.1.2 Dspace

Dspace je open source nástroj na správu digitálneho obsahu ako sú dokumenty, obrázky a datasety [14]. Spravuje samotné dáta aj ich metadáta. Na popis metadát využíva formát Dublin Core [15] a je možné ich exportovať do formátu CSV pomocou príkazového riadku. Metadáta rozdeľuje na popisné (popisujú vlastnosti dát), administratívne (popisujú pôvod a vlastníka) a štruktúrne (obsahujú informácie o type a prezentovaní dát) [16]. Narozdiel od Fedory, Dspace ponúka aj užívateľské rozhranie s viacerými funkciami (Obrázok 3.2), ktoré je môžné modifikovať podľa potreby.

Dspace umožňuje organizáciu dát do hierarchickej štruktúry pomocou komunít a kolekcií [16]. Dáta prispievajú do Dspacu užívatelia, ktorých je možné triediť do skupín. Užívateľom a skupinám sa potom priradzujú práva na prístup do komunít aj kolekcií.

Ďalšími funkciami Dspacu je napríklad fulltextové vyhľadávanie v textových dátach alebo podpora priraďovania licencií. Ďalej je tu možnosť pristupovať k dátam aj pomocou API. Na tvorbu trvalých odkazov sa využíva systém CNRI Handle [17]. Dspace je možné ďalej rozšírovať pomocou pluginov.



Obr. 3.2: Webové rozhranie nástroja Dspace

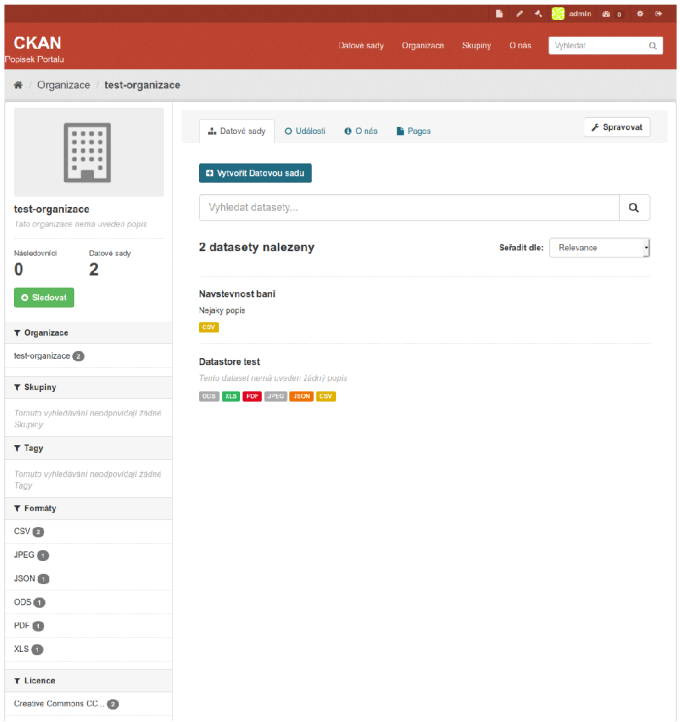
3.2 CKAN

CKAN (Comprehensive Knowledge Archive Network) je nástroj na ukladanie, správu a publikovanie dát [18]. Je to open source softvér využívaný mnohými organizáciami a krajinami na správu otvorených dát.

CKAN obsahuje webové rozhranie (Obrázok 3.3) pre jednoduchú správu digitálneho obsahu pre jeho užívateľov, ktoré je možné prispôsobovať. Pomocou tohto rozhrania sú prístupné aj niektoré nastavenia portálu pre administrátorov, no väčšina nastavení sa uskutočňuje na

príkazovom riadku. Pre automatizáciu práce s katalógom slúži bohaté API. K dátam je možné pridávať metadáta a vytvárať ich náhľady. Dáta je tiež možné priraďovať do rôznych organzácií a skupín. Vyhľadávanie v CKANe umožňuje nielen hľadať dáta, ale prehľadávať aj ich obsah a metadáta.

V prípade chýbajúcej funkcionality CKAN ponúka aj množstvo rozšírení[19]. Pomocou rozšírení CKAN umožňuje napríklad automaticky ukladať štrukturované dáta do databázy prípadne vytvárať komplexnejšie náhľady z rôznych typov súborov.



Obr. 3.3: Webové rozhranie nástroja CKAN

3.3 Komerčné riešenia

Pre ukážku sú v tejto kapitole uvedené aj niektoré komerčné riešenia pre správu a ukladanie dát. Nevýhodou komerčných riešení je zaviazanie sa niektorému uzavretému produktu. Dostať hotové riešenie

a o nič sa nestarať je zaujímavá ponuka pre mestá, no takéto produkty sú finančne nákladnejšie ako open source riešenia a na pridanie chýbajúcich funkcionalít je zvyčajne potrebný ďalší čas a finančné prostriedky. V prípade zrušenia spolupráce s takýmto produktom sú dáta uchované v systéme, z ktorého nemusí byť možné ľahko exportovať kompletné dáta a presunúť ich do nového systému.

Takéto systémy bez bližšej dohody prípadne objednávky sú zvyčajne nedostupné na vyskúšanie, preto informácie o nich boli získané len z verejných zdrojov, ku ktorým bolo možné získať prístup.

3.3.1 INVIPO

Invipo je flexibilná a otvorená platforma zaisťujúca integráciu v smart cities [20]. Invipo dokáže prepojiť dáta z rôznych systémov a zobraziť ich na centrálnom mieste. Dokáže vizualizovať dáta z rôznych zdrojov v jednotnej podobe a prevádzkuje online monitorovanie zariadení. Pomocou nadefinovaných scenárov dokáže upozorniť na zmenu v dátach, prípadne aj sám reagovať a ovplyvniť vzdialené zariadenia [21]. Samotným občanom ponúka rýchlejšiu a aktuálnejšiu dostupnosť informácií, možnosť rozhodovať o veciach týkajúcich sa mesta či prehľad o doprave a parkovaní. Invipo beží na webových technológiach, takže prístup k nemu je možný z akéhokoľvek zariadenia [22].

Invipo ďalej uvádza, že je to otvorená platforma pre partnerov a vývojárov a že niektoré časti platformi sú dostupné celej komunite vývojárov [22]. Mne sa žiaľ nepodarilo nájsť žiadne projekty, softvér ani zdrojové kódy, ktoré by boli dostupné mimo túto komunitu. Domnievam sa, že je potrebné komunikovať cez spoločnosť Incinity, aby sa vývojár mohol stať členom tejto komunity. Incinity je spoločnosť, ktorá stojí za touto platformou.

Invipo je využívané viacerými mestami v Českej republike, ale aj vo svete [23]. Najviac technológií a funkcií prevádzkuje Invipo v tureckom meste Izmir. Majú tam umiestnených viac ako 4000 inteligentných zariadení, ktoré spravujú a monitorujú križovatky, dopravu, vozidlá mestskej hromadnej dopravy, parkovanie, dopravné značenie, meteo stanice a mnohé iné. V Českej republike funguje Invipo napríklad v meste Olomouc, kde prevádzkujú mestskú dopravnú ústredňu.

3.3.2 ASD Software

ASD Software je spoločnosť, ktorá sa zaoberá vývojom softvéru a informačných systémov pre verejnú správu [24]. V ich ponuke sa nachádza aj softvér s podporou pre open data [25], s ktorým už majú skúsenosti a realizovali v tejto oblasti nejaké projekty. Viac informácií však nie je možné z verejných zdrojov získať.

4 Praktická časť

4.1 Zber požiadaviek

Pri tvorbe práce a hlavne výbere platforiem prebiehala spolupráca s mestom Žďár nad Sázavou. Platforma na správu dát musí okrem požiadaviek Európskej únie a Českej republiky spĺňať aj nároky vyplívajúce z tejto spolupráce.

Mesto Žďár nad Sázavou má záujem o prevádzkovanie platformy na centrálnu správu dát. Tieto dáta sú produkované samotným mestom, ale aj množstvom mestkých firiem a organizácií pôsobiacich v ňom.

Prehľad požiadaviek je možné zhrnúť do nasledujúceho zoznamu:

- Správa dát - vytváranie, aktualizovanie, modifikácia a prípadné mazanie dát
- Organizácia dát - logické triedenie dát medzi mestské firmy a organizácie
- Užívateľské rozhranie - jendoduché rozhranie na ovládanie systému pre jeho užívateľov
- Otvorené dáta - podpora pre publikovanie dát na princípe Otvorených dát
- Automatizácia - dostupnosť API alebo iného rozhrania na automatizáciu procesov pridávania a aktualizácie dát
- Vizualizácia dát - pdopora pre zobrazenie dát, prípadne jednoduché prepojenie s ďalšími vizualizačnými nástrojmi
- DCAT - podpora vytvoriť dcat prístupový bod pre katalóg a datasety

4.2 Formát DCAT

Počas skúmania možností kompatibility portálov s Národným katalógom a komunikácií s Ministerstvom vnútra mi bol uvedení do

pozornosti formát DCAT. Pomocou tohto formátu dokáže Národný katalóg zbierať a pochopiť dáta umiestnené v lokálnych katalógoch jednotlivých miest. Tento formát bol vybraný Európskou úniou ako spôsob na prepájanie všetkých katalógov v nej.

DCAT (Data catalog vocabulary) je RDF slovník na umožnenie prepojenia dátových katalógov umiestnených na webe [26]. DCAT v podstate dovoľuje jednotne popisovať informácie o datasetoch a dátových službách v katalógoch a zjednodušuje využívanie a agregovanie metadát z ďalších katalógov. Umožňuje vyhľadávanie v datasetoch umiestnených vo viacerých decentralizovaných katalógoch.

RDF (Resource description framework) je framework na tvorbu informácií o súboroch na internete, aby ich okrem ľudí vedeli pochopiť aj počítače. Skladá sa z troch elementov vo formáte `<objekt> <predikát> <objekt>`, ktoré môžu mať ľubovoľnú podobu. Predikát vysvetľuje vzťah medzi dvoma objektami, napríklad `<Bob> <je> <osoba>` [27].

DCAT je už konkrétnejšia a obmedzenejšia podoba RDF vrámci syntaxe aj sémantiky. Na popis datasetov využíva triedy a atribúty. Pre nasadenie DCATu do nejakej konkrétnej oblasti sa využívajú DCAT profily. Profily sú špecifikácie, ktoré vytvárajú obmedzenia na metadáta dátových katalógov. Profily určujú povinné, doporučené a dobrovoľné polia metadát, ktoré popisujú datasety a dáta v nich. Konkrétnym takýmto profilom je napríklad profil DCAT-AP (DCAT Application Profile) Európskej únie na popis otvorených dát, ktoré sú v rámci nej publikované. Najnovšia verzia DCATu je v2 a je z novembra 2019 [26]. K nemu bol následne vytvorený aj Európsky profil DCAT AP 2.0.0 [28], ktorý mení povinnosť niektorých atribútov.

Štáty si môžu atribúty prispôsobovať podľa svojich potrieb, musia však ostať kompatibilné s profilom Európskej únie. Česká republika v súčasnosti beží ešte na verzii DCAT AP 1.2, preto popíšem triedy, atribúty a ich možné hodnoty, ktoré je nutné používať a sú platné v súčasnosti.

4.2.1 DCAT profil Českej republiky

Profil DCATu používaný v Českej republike je aktuálne definovaný v dokumente Rozhranie katalógov otvorených dát [29]. Tento dokument upresňuje, aké triedy a atribúty musia byť definované v DCAT

prístupovom bode, aby mohol byť dataset alebo katalóg pridaný do Národného katalógu.

Rozhranie definuje dve povinné triedy a to triedu Dátová sada (`dcat:dataset`) a Distribúcia dátovej sady (`dcat:distribution`).

Trieda dátova sada musí obsahovať nasledujúce atribúty:

- `dct:title` - názov dátovej sady
- `dct:description` - popis dátovej sady
- `dct:theme` - tému dátovej sady
- `dct:accrualPeriodicity` - periodicita aktualizácie
- `dcat:keyword` - kľúčové slová
- `dct:spatial` - súvisiace geografické územie
- `dcat:distribution` - distribúcia dátovej sady (samostatná trieda)

Teraz vysvetlím použitie a hodnoty niektorých komplikovanejších atribútov.

Téma dátovej sady predstavuje odkaz na kategóriu podľa slovníka Európskej únie [30]. Tá musí obsahovať celú cestu k hodnote, takže napríklad pre kategóriu hospodárstvo a financie bude táto hodnota vyzerať nasledovne:

`http://publications.europa.eu/resource/authority/data-theme/ECON`

Periodicita aktualizácie predstavuje hodnotu frekvencie, ako často je dátova sada aktualizovaná. Hodnota opäť pochádza z dátového slovníka Európskej únie [31]. Príkladom ročnej aktualizácie je:

`http://publications.europa.eu/resource/dataset/frequency/ANNUAL`

Súvisiace geografické územie odkazuje na geografickú oblasť, ktorej sa dátová sada dotýka. Jej hodnota je zo slovníku RÚIAN (Registr územní identifikace, adres a nemovistostí) a je v tvare:

`https://linked.cuzk.cz/resource/ruian/{SKRATKA}/{KOD}`

Skratka môže naberať hodnôť typu `stat`,`okres`,`obec` a mnoho ďalších. Kód predstavuje identifikačné číslo konkrétneho územia. Vyhľadávanie týchto hodnôt ponúka portál Českého úradu zememeračského

a katastrálneho [32] a všetky dostupné hodnoty sú ďalej popísané v dokumente popisujúcom prepojené dáta v Českej republike [33].

Trieda Distribúcia dátovej sady popisuje samotné dáta patriace dátovej sade. Jej povinné atribúty sú:

- `pú:specifikace` - špecifikácia podmienok používania
- `dcat:downloadUrl` - odkaz na stiahnutie danej distribúcie dátovej sady
- `dct:format` - formát súboru na stiahnutie
- `dcat:mediaType` - media type súboru na stiahnutie

Špecifikácia podmienok použitia obsahuje odkaz na popis podmienok, za ktorých môžu byť dáta použité. Hodnoty sa zvolia podľa návodu na stanovenie podmienok použitia [34] a možu naberať nasledujúce hodnoty:

- `pú:autorské-dílo`
- `pú:databáze-jako-autorské-dílo`
- `pú:databáze-chráněná-zvláštními-právy`
- `pú:osobní-údaje`

Spolu s nimi je potrebné uvádzať aj samotný odkaz, napríklad pre hodnotu `pú:autorské-dílo` je potrebné uviesť aj odkaz

`<https://data.gov.cz/podmínky-užití/neobsahuje-autorská-díla/>`

Formát súboru na stiahnutie musí obsahovať hodnotu z európskeho slovníka pre typy súborov [35]. Príkladom pre súbor typu CSV je:

`http://publications.europa.eu/resource/authority/file-type/CSV`

Nakoniec media type súboru je zvolení zo slovníka Media type IANA [36]. Opäť pre súbor formátu CSV by bola hodnota tohto atribútu:

`http://www.iana.org/assignments/media-types/text/csv`

Atribúte označené ako nepovinné tu popísané niesu. V prípade potreby sa zoznam všetkých atribútov nachádza v dokumente Rozhrania katalógov otvorených dát [29].

4.3 Porovnanie a výber systému

Na správu dát už boli spomenuté systémy Fedora, Dspace a CKAN. Prehľad splnenia požiadaviek sa nachádza v nasledujúcej tabuľke 4.1:

Tabuľka 4.1: Porovnanie požiadaviek a platforiem

Požiadavky	Fedora	Dspace	CKAN
Správa dát	✓	✓	✓
Organizácia dát	✓	✓	✓
Užívateľské rozhranie	✗	✓	✓
Otvorené dáta	✓	✓	✓
Automatizácia (API)	✓	✓	✓
Vizualizácia dát	✗	✗	✓
Dcat	✗	✗	✓

Všetky tieto systémy sú určené na správu dát a podporujú tvorbu užívateľov, čo by mohlo pokryť potrebu logického rozdelenia dát patriacich rôznym mestským firmám. Nástroje Dspace a CKAN však podporujú aj detailnejšie rozďeľovanie dát. V Dspace ide o rozdelnie pomocou komunít, u CKANu sa hlbšie delenie vytvára pomocou organizácii. Všetky tri nástroje ponúkajú automatizáciu pomocou API a fungujú na princípe Otvorených dát.

Fedora slúži ako dobrý základ pre dátový portál, no neponúka jednoduché užívateľké prostredie, iba restového klienta, ktorý môže byť komplikovaný pre užívateľov, ktorý s API ešte nepracovali. Keďže nemá užívateľské rozhranie, neponúka ani náhľady dát. Dáta je možné iba stiahnúť. Taktiež som nenašiel priamu podporu pre dcat, takže by bolo potrebné buď túto funkcionalitu doimplementovať, alebo vytvárať pre každý dataset súbor s dcat informáciami ručne. Ako dátový portál pre smart cities sa preto fedora veľmi nehodí.

Nástroj Dspace je už na tom lepšie s užívateľským prostredím, no stále zaostáva s podporou dcatu. Má síce dobrý editor metadát, no umožňuje iba export do formátu csv. Bolo by potrebné vytvoriť nástroj, ktorý by ich vedel prezentovať vo formáte dcat, aby ho potom bolo možné pripojiť na Národný katalóg. Dspace tiež neumožňuje zobrazovať obsah dát, iba ich stiahnutie.

CKAN na druhej strane pokrýva všetky požiadavky. Dovoľuje vytvárať organizácie a skupiny na granulárne triedenie dát. Ponúka API na prácu s dátami a v prípade tabuľkových dát aj s ich obsahom. Tabuľkové dáta sú ukladané do sql databázy. Na takúto databázu sa môže pripojiť vizualizačný nástroj a pracovať s ňou priamo. Nie je potrebné dáda duplikovať do každej databázy zvlášť a udržiavať ich aktuálne na viacerých miestach. Užívateľ má k dispozícii jednoduché webové rozhranie na prácu s týmto portálom. Pomocou rozšírenia dcat dokáže vytvoriť dcat prístupový bod pre celý katalóg, ktorý je možné pridať do Národného katalógu. Pomocou dcat profilov dokáže ragovať na zmeny v požiadavkách na atribúty dcat.

Ďalšou výhodou CKANu sú jeho referencie. CKAN je reálne využívaný ako dátový portál českých miest Brna [1] alebo Prahy [2], ale aj zahraničných miest ako napríklad Zurich [3].

4.4 Popis CKANu

V tejto budú podrobnejšie písané organizáciu, funkcie a rozšírenia, vďaka ktorým CKAN spĺňa všetky požiadavky.

4.4.1 Štruktúra CKANu

V CKANe sú dáta organizované do datasetov [37]. Dataset je jednotka, ktorá reprezentuje skupinu dát. Obsahuje popisné informácie (metadáta) o dátach, ako aj dáta samotné. Dáta sa v terminológií CKANu nazývajú zdroj a v jednom datasete ich môže byť viacero. Po nahratí dát vytvorí CKAN pre každý takýto zdroj trvalý odkaz, pod ktorým budu prístupné. Po prepísaní zdroja novými dátami sa odkaz na ne nemení. To znamená, že trvalý odkaz na nejaký zdroj pretrváva aj jeho aktualizáciu. Odkazy ostávajú platné a je možné sa odkazovať na aktuálnu verziu dát vždy rovnakým odkazom.

Základným prispievateľom dát do CKANu je uživateľ. Ten je členom organizácií, pod ktorými dáta publikuje. Organizácie sú potom vlastníkmi a reprezentantmi datasetov. Majú za úlohu datasety publi-

1. `https://data.brno.cz/`
2. `http://opendata.praha.eu/`
3. `https://data.stadt-zuerich.ch/`

kovať, spravovať a aktualizovať. Datasety môžu byť označené aj ako súkromné, vtedy k ním majú prístup iba členovia organizácie, pod ktorú dataset patrí. V základnej inštalácii sa organizácie ďalej triediť nedajú. Je ale k dispozícii rozšírenie `hierarchy` [38], ktoré umožňuje rozloženie organizácii do stromovej štruktúry, teda vytvárať nadradené a podradené organizácie.

Ďalšou entitou sú skupiny. Sú to akési kategórie na logické triedenie datasetov. Príkladom by mohli byť napríklad vzdelávanie alebo doprava. Ide o rozdelenie datasetov podľa významu.

K jednotlivým datasetom je možné ešte pridávať značky (tagy) pre lepšiu orientáciu a triedenie.

4.4.2 Náhľady dát

Popri ukladaní dát CKAN dovoľuje aj ich náhľady alebo zobrazenia [39]. Ide najmä o textové formáty typu txt, json a xml a obrázky vo formátoch jpeg a png. Ďalšie typy súborov je možné zobraziť pomocou rozšírení. Tieto náhľady môžu byť vytvárané manuálne užívateľom alebo automaticky pri nahratí nového súboru správneho typu.

4.4.3 CKAN API

CKAN ponúka bohaté API na správu a konfiguráciu služby [40]. Časť API príkazov dovoľuje pristupovať k informáciam aj anonymným užívateľom. Tu zvyčajne ide o funkcie, ktoré sú verejne dostupné aj cez webové rozhranie. Ostatné funkcie potom využívajú na autorizáciu API kľúče, ktoré su vygenerované užívateľom pri registrácií a sú dostupné v prehľade užívateľa na webe. Aké oprávnenia má užívateľ na webe, také isté oprávnenia sa uplatňujú pri použití jeho API kľuča. API kľuč sa vkladá do hlavičky HTTP požiadavky pod parameter authorization.

Telo požiadavky je vo formáte JSON [4]. Pri zasielaní súboru, napríklad pri vytváraní alebo aktualizácii dát, sa využijú formuláre [5].

Odpoveď z API je vo formáte JSON a obsahuje položky:

- success - true alebo false, podľa toho či bola požiadavka úspešná

4. `Content-Type: application/json`
5. `Content-Type: multipart/form-data`

- error - popis chyby v prípade neúspešnej požiadavky
- result - výsledok alebo dáta, o ktoré sme žiadali, v prípade úspešnej požiadavky

Kompletné API sa nachádza v dokumentácií[40], preto len stručne spomeniem jeho funkcie a upozorním na niektoré dôležité časti. Pomocou API sa dajú získavať, vytvárať, modifikovať a mazať datasety, zdroje, skupiny, organizácie alebo užívatelia. Datasety sa v API terminológií označujú ako package. Datasety a zdroje vieme aktualizovať čiastočne alebo úplne. Pri čiastočnej aktualizácií sa pôvodné hodnoty metadát nemenia a doplnia, prípadne sa zmenia len vybrané. Pri úplnej aktualizácii sa všetky hodnoty vymažú a nastavené hodnoty budú len tie, ktoré boli v požiadavke. Podobne existujú aj dve verzie mazania. Jedná len odstráni položku z viditeľnej časti stránky, ktorá je ale stále prístupná na obnovenie, princíp odpadkový kôš na súbory. Druhá potom odstráni objekt natrvalo. Ďalej je možné napríklad overiť alebo získať členstvo užívateľa v organizácii alebo skupine.

Zvláštne fungujúcim API požiadavkom je nahranie a pridanie nového zdroja (súboru) do existujúceho datasetu. Povinným parametrom je id datasetu a url súboru. To platí aj v prípade, že chceme nahrať lokálny súbor a teda url adresu nemáme. Kedže je to povinný parameter, je potrebné zadať ľubovoľnú adresu. Funguje to tak, že pole adresa je ignorované, ak je zároveň zaslaný aj nejaký súbor, ktorý sa posiela cez parameter upload. Aktualizácia súboru už pole url nevyžaduje.

Príkladom volania api na získanie všetkých verejných datasetov je volanie:

`http://CKAN_URL/api/3/action/package_search`

Toto volanie vráti v hodnote result zoznam všetkých verejných datasetov a ich zdrojov aj s metadátami. Pre súkromné datasety je potrebné žiadať podľa konkrétneho užívateľa, skupiny alebo organzácie.

Podľa konrétneho užívateľa by volanie vyzeralo podobne. Do hlavičky volania je ale potrebné pridať pole `Authorization` a za jeho hodnotu vložiť API kľúč užívateľa. Do tela volania je potrebné vložiť JSON objekt a nastaviť `Content-type: application/json`. JSON objekt vyzerá nasledovne:

```
{"include_private": "True"}
```

Toto volanie vráti aj súkromné datasety, ktoré patria pod organizácie, ku ktorým má užívateľ prístup.

Pomocou parametra q je možné v datasetoch aj vyhľadávať. Vyhľadávanie sa uskutočnuje podľa pravidiel vyhľadávacieho nástroja Solr. Ten prehľadáva všetky metadáta datasetov a zdrojov. Napríklad ak pri vyššie uvedenom volaní zmením JSON objekt nasledovne:

```
{
  "include_private": "True",
  "q":"vodomer"
}
```

Bude mi vrátený zoznam datasetov, kde každý z nich obsahuje v niektorom polí metadát reťazec `vodomer`.

Toto API môže byť ešte rozšírené o požiadavky pracujúce priamo nad dátami uloženými v databáze. To má nastarosti rozšírenie Datastore, ktorému sa budem venovať v ďalšej časti.

4.4.4 Rozšírenie Datastore

Rozšírenie Datastore zjednodušuje prácu s tabuľkovými dátami. Takéto dáta ukladá do databázy a ponúka prístup k nim [41]. Dáta nahrané do CKANu su predané Datastoru, ktorý ich uloží do databázy. Keď ich CKAN potom potrebuje, napríklad pri zobrazení tabuľky na webovom rozhraní, pristupuje k ním pomocou tohto rozšírenia. Datastore je ovládaný pomocou jeho API, ktoré ponúka priamy prístup do databázy. Datastore je pripojený na autorizačný prístup CKANu, takže sa použije rovnaký API kľúč, ako pri CKAN API a všetky prístupové práva sú zachované.

Pri nahratí nového súboru do CKANu sú dáta predávané do Datastoru buď automaticky pomocou ďalších rozšírení, alebo manuálne pomocou API.

Datastore ponúka jednoduchý postup na stiahnutie dát uložených v databáze a to aj v rôznych formátoch. Na prístup k dátam slúži adresa `{CKAN-URL}/datastore/dump/{RESOURCE-ID}`, kde CKAN-URL predstavuje webovú adresu CKANu a RESOURCE-ID je id zdroja v CKANe. Defaultný formát je csv, pomocou parametru `format` v url adrese sa dá zvoliť spomedzi formátov tsv, json alebo xml. Ak chceme súbor kompatibilný s programom excel, je potrebné tiež zadať parameter `bom=true`.

4.4.5 Datastore API

Pomocou Datastore API je možné vytvárať, pristupovať a upravovať dáta uložené v Datastore. Spomenuť by som chcel API volanie `datastore_create`, ktorý plní viacero funkcií [42]. Základnou funkciou je vytvorenie tabuľky z dát. Vtedy očakáva parametre buď `resource_id` alebo `resource`. Resource_id určuje id zdroja v CKANe, voči ktorému budú dáta uložené. Ak dáta v CKANe nahraté ešte nie sú, nemajú ešte svoje id, je potrebné zavolať požiadavku s parametrom resource, ktorý je vo formáte slovníku a obsahuje parametre, ktoré sú pradané API volaniu v CKANe na vytvorenie nového zdroja. Ďalej je možné upresniť metadáta o stĺpcoch v tabuľke pomocou parametru `field`, prípadne vložiť nové záznamy pomocou parametru `records`. Oboje tieto parametre sú vo formáte zoznam slovníkov typu JSON.

Ďalej toto API volanie dovoľuje nastavenie aliasov, indexov a primárneho kľuča pomocou parametrov `aliases`, `indexes` a `primary_key`. Tie sú vo formáte reťazec a pri použití s viacerými hodnotami sa tieto hodnoty oddeľujú čiarkou. Pri použití jedného z týchto parametrov je pôvodná hodnota parametru prepísaná novou.

Príkladom volania, keď chcem pridať ďalší záznam do databázy je volanie na adresu:

`http://CKAN_URL/api/3/action/datastore_create`

V hlavičke treba mať nastavené polia `Authorization` a `Content-type`. `Authorization` obsahuje api kľuč užívateľa a `Content-type` zase hodnotu `application/json`.

Telo volania bude vyzerať nasledovne:

```
  {
"resource_id": "1b70d5d0-6dae-4f60-9gaf-53592ebad06a",
"records": [
      {
        "Text": "abc",
        "Cislo": 123,
        "Cas": "10:10:10",
        "Datum": "2019-10-21"
      }
    ]
}
```

`resouce_id` obsahuje id tabuľky, do ktorej budú záznamy vložené. `records` obsahuje pole záznamov, ktoré budú vložené. Záznamy budú vložené na koniec tabuľky. Záznam potom obsahuje polia v tvare `kľuč:hodnota`. Kľúč je názov stĺpca, do ktorého sa bude vkladať a hodnota je hodnota, ktorá sa vloží. Je jedno, v akom poradí sú tieto polia zapísané, dôležitá je presná zhoda kľúča a názvu stĺpca

Datastore API ponúka aj vyhľadávanie v uložených dátach. Na to slúžia príkazy `datastore_search` a `datastore_search_sql`. Prvý príkaz umožňuje vyhľadávanie pomocou parametrov a ich následné filtrovanie, zoradenie a obmedzenie počtu získaných výsledkov. Taktiež dovoľuje vyhľadávanie pomocou SQL jazyka a podporuje aj prístup ku súkromným zdrojom pomocou autorizácie. Druhý príkaz umožňuje vyhľadávanie iba vo verejných dátach pomocou SQL jazyka.

4.4.6 Rozšírenie Datapusher

Popri rozšírení Datastore môže pracovať aj rozšírenie Datapusher. Jeho úlohou je sledovať, aké súbory sú nahrávané do CKANu. Ak sa jedná o tabuľkový súbor, ktorý by mohol byť nahraný do databázy, pokusí sa o to. Odhadne, aké typy dát sa nachádzajú v jednotlivých stĺpcoch súboru. Z hlavičky dát vytvorí schému databázy. Každému záznamu pridá identifikátor a v prípade súboru formátu csv uloží aj plný text každého záznamu. Všetky tieto informácie potom predá Datastoru a ten to do databázy uloží. Pri tomto procese sa originálny súbor nijako nemodifikuje, dáta ostávajú nedotknuté v stave, v akom boli nahrané.

Nedostatkom Datapusheru je podpora formátov. V základnej konfigurácií podporuje iba text, číslo a časovú známku. Časová známka obsahuje informáciu o dátume a čase. Pomocou konfigurácie sa dajú pridať ďalšie formáty [43]. Podľa aktivity v repozitári je Datapusher udržiavaný projekt, no nové funkcie sa doň nepridávajú [44].

Alternatívou k Datapusheru by mohlo byť rozšírenie Xloader [45]. Toto rozšírenie sa snaží odstrániť obmedzenia, ktoré má Datapusher. Je to mladý a aktívny projekt, ktorý vznikol na jeseň 2019 za podpory firmy OpenGov Inc. [45]. Funkcia je rovnaká, ako v prípade Datapusheru. Opať má spracovať tabuľkové súbory a uložiť ich do Datastoru. Xloader beží vrámci CKANu (Datapusher bežal ako samostatná služba) a snaží sa všetky hodnoty uložiť ako text. Ak sa to nepodarí, pokúsi sa detekovať typy v stĺpcoch, podobne ako Data-

pusher. V prípade zle zvoleného typu môže užívateľ zvoliť manuálne správny typ, a to dokonca vo webovom rozhraní. Slúži na to Data Dictionary [46].

Datapusher aj Xloader používajú rovnakú knižnicu na detekciu typov, Messytables [47], a tá obmedzuje typy, aké môžu byť definované v Data Dictionary a použité pri detekcii typov. Messytables je open source projekt a v prípade potreby, dovoľuje doimplementovanie vlastných typov a ich detektorov.

Pri rozhodovaní, ktoré rozširenie z týchto dvoch použiť, by som sa zatiaľ prikláňal k Datapusheru. V prípade problémov je možné kedykoľvek prejsť na Xloader, zmena vyžaduje iba deaktiváciu Datapusheru, aktiváciu XLoaderu a následne je potrebné iba reštartovať CKAN. V súčasnosti je Xloader stále v počiatočnom stave vývoja, aj keď jeho hlavné funkcie sú už plne funkčné a v budúcnosti by mohol Datapusher plne nahradiť.

4.4.7 Rozšírenie DCAT

Aby dátový katalóg mohol byť využívaný oficiálne, musí podporovať formát dcat. Túto podporu je možné do CKANu doplniť pomocou rovnomenného rozšírenia `dcat`. Toto rozšírenie vytvorí dcat prístupový uzol, na ktorý sa môžu napojiť nadradené katalógy. Takýto bod sa vytvorí ako pre celý katalóg, tak aj pre každý dataset v ňom.

K celému katalógu je potom možné pristúpiť pomocou adresy `{CKAN_URL}/catalog.{FORMAT}`. K jednotlivým datasetom stačí pridať `.{FORMAT}` do ich webovej adresy. Za formát je možné dosadiť nasledujúce hodnoty:

- xml - vytvorí dcat vo formáte xml
- ttl - vytvorí dcat súbor vo formáte turtle
- jsonld - vytvorí dcat vo formáte json for linking data

CKAN si ukladá metadáta o datasetoch vo svojom vlastnom formáte. Rozšírenie dcat iba namapuje existujúce metadáta z CKANu na metadáta vo formáte dcat. Kompletné mapovanie je možné vidieť v github repozitári tohto rozšírenia [48]. Takéto mapovanie sa špecifikuje pomocou profilov. K dispozícii je základný profil založený

na DCAT application profile Európskej únie a je plne kompatibilný s verziou DCAT AP v1.1 [49]. Pred nedávnom bola vydaná novšia verzia DCAT AP v2, ktorá upravuje niektoré polia metadát. Riešenie na jeho podporu je vytvorenie vlastného profilu, ktorý bude obsahovať potrebné metadáta.

Príkladom môžu byť profily vytvorené v Nemecku [50] alebo Taliansku [51]. Tieto profily zabezpečujú kompatibilitu CKANu s ich národnými katalógmi a prídávajú ďalšie funkcionality pre zjednodušenie práce s dátami a metadátami využívanými v ich krajinách.

4.5 Implementácia CKANu

V tejto časti popíšem, ako je možné CKAN nainštalovať a spustiť. Ďalej popíšem čo je potrebné a vhodné nakonfigurovať a spomeniem niektoré rozšírenia a ich konfiguráciu, ktoré doplnia potrebnú funkcionalitu do CKANu podľa požiadaviek.

4.5.1 Inštalácia

CKAN ponúka tri postupy na jeho inštaláciu [52]. Pokiaľ bude CKAN jediný softvér na danom stroji, je tu možnosť instalácie z balíčka distribúcie alebo priamo zo zdrojového kódu. V prípade využitia kontajnerových technológií je tu tretia možnosť inštalácie CKANU pomocou kontajnerov docker, ktoré ponúkajú jednoduchú a kompletnú inštaláciu všetkých súčastí potrebných na prevádzkovanie CKANu. Pomocou systému docker compose je možné spustiť všetky časti CKANu jedným príkazom a v priebehu pár minúť mať funkčný a bežiaci systém.

CKAN vyžaduje niekoľko služieb bežiacich zároveň s hlavným programom. Prvou z nich je databáza. CKAN využíva databázu postgres na uchovávanie informácií a prípadné ukladanie datasetov. Druhým programom pre správny beh CKANu je služba Solr. Solr je vyhľadávacia platforma, ktorá indexuje datasety a dáta, aby ich vyhľadávanie bolo rýchlejšie a presnejšie. Ďalšou požadovanou službou CKANu je databáza Redis. Tá slúži na kešovnie práve spracovávaných požiadaviek a dát. Poslednou súčasťou CKANu je webový server, ktorý obsluhuje webové stránky. Na to je možné použiť NGINX alebo Apache.

Po inštalácií je potrebné nakonfigurovať databázu. Je potrebné vytvoriť nového užívateľa a databázu, ku ktorej bude mať ckan prístup. Podľa toho treba následne nastaviť pripojenie v CKANe. V konfiguračnom súbore sa za parameter `sqlalchemy.url` dosadí nasledujúci reťazec `postgres://USERNAME:PASSWORD@HOST/DBNAME` so správnymi údajmi. Pri použití inštalácie pomocou kontajnerov je tento krok vykonaný automaticky. Vytvorí sa databáza aj užívateľ s názvom ckan.

Ďalej je potrebné vytvoriť administrátorsky účet, ktorým je potom možné sa prihlásiť cez webové rozhranie. To sa vykoná nasledujúcim príkazom `/usr/local/bin/ckan-paster -plugin=ckan sysadmin -c /etc/ckan/production.ini add admin`, kde posledné slovo je prihlasovancie meno. Ako funguje príkaz paster je podrobnejšie popísané v časti konfigurácia. V tomto kroku je možné sa prihlásiť a pracovať s CKANom.

4.5.2 Konfigurácia

Po spustení CKANu nasleduje jeho konfigurácia. Niektoré jej časti je možné nastaviť vo webovom rozhraní. K dispozíii je tam jednoduchá správa užívateľov, zmena informácií, loga a štýlu portálu. Ďalej je možné kontrolovať históriu zmien datasetov a zdrojov.

Ďalšia konfigurácia potom prebieha pomocou príkazového riadku. Na ceste `/etc/ckan/production.ini` sa nachádza hlavný konfiguračný súbor a na aplikovnaie jeho zmien je potrebné CKAN reštartovať. V ňom sa nastavujú vlastnosti a parametre inštancie alebo sa aktivujú a konfigurujú rozšírenia.

Po inštalácii je v ňom potrebné nastaviť webovú adresu portálu, adresu vyhľadávacej služby solr, adresu kešovacej databázy redis a prístupové údaje do sql databázy pomocou formátu sqlalchemy. V tomto súbore je možné vypnúť registráciu nových užívateľov na webovom rozhraní a pomocou API. Ďalej je možné nastaviť povolenie užívateľom vytvárať a mazať organizácie a skupiny, nastaviť hlavný a vedľajšie jazyky, v ktorých sa bude webová stránka zobrazovať.

Je možné tu definovať typy súborov, pre ktoré majú byť vytvárané náhľady automaticky. Stačí pridať názov pluginu zodpovedného za daný typ do parametra `ckan.views.default_views`.

Ďalším dôležitým nastavením je nastavenie CORS (Cross-Origin Resource Sharing) [53]. Toto nastavenie umožňuje alebo blokuje prí-

stup k dátam z inej domény, než je primárna doména nastavneá pre CKAN. Predvolené nastavenie je všetky tieto žiadosti blokovať, takže nie je možné si stiahnúť dáta pomocou HTTP požiadavky z cudzej domény. Ďalšie možnosti predstavujú povoliť len niektoré domény alebo povoliť domény všetky.

Poslednou funkciou tohoto konfiguračného súboru je aktivovanie rozšírení a ich čiastočná konfigurácia. Rozširenia, ktoré chceme mať zapnuté je potrebné pridať do do zoznamu parametru `ckan.plugins`.

Na komplexnejšiu konfiguráciu sa používa príkaz paster. Na pochopenie, ako tento príkaz spustiť je potrebné vysvetliť, ako CKAN beží. CKAN je aplikácia vytvorená v jazyku python. Ten na beh jednotlivých programov využíva virutálne priestory, ktoré oddeľujú aplikácie od samých seba aj od hlavného systému. Samotný CKAN beží v jednom takomto virtuálnom priestore. Na prístup do takéhoto priestoru je potrebná jeho aktivácia. Tá sa uskutoční príkazom `source $CKAN_VENV/bin/activate`. V prípade nenastavenia premennej `$CKAN_VENV` je potrebné zadať plnú cestu `/usr/lib/ckan/default`.

Po aktivovaní virtuálneho priestoru je možné spúšťať príkaz paster priamo. Paster potrebuje naznačiť, čoho sa príkaz týka. Zvyčajne ide o konfiguráciu celého CKANu, preto je potrebné použiť prepínač `-plugin=ckan`. Alternatívou je prejsť do priečinku ckan na ceste `/usr/lib/ckan/default/src/ckan`, potom nie je nutné prepínač použiť. Čo je potrebné ešte uviesť je cesta ku hlavnému konfiguračnému súboru, ktorý bude prípadne paster modifikovať. Na to slúži prepínač `-c` nasledovaný cestou ku vyššie spomenutému súboru.

Po všetkých týchto úkonoch je možné príkaz paster použiť. Pomocou neho je možné spravovať datasety, databázu, užívateľov, systémových administártorov, úlohy na pozadí a ďalšie časti CKANu.

Jednou z dôležitých funkcií, ktoré som počas testovania CKANu zistil, je aktualizácia indexu datasetov, ktorú ma príkaz paster nastarosti. Pri zmene konfigurácie a reštarte CKANu sa vymaže zoznam datasetov, ktoré CKAN zobrazuje na webovom portáli., takže sa môže javiť ako prázdny katalóg. Datasety samotné sú však nedotknuté. Vtedy stačí aktualizovať index príkazom `paster search-index` a datasety sa znova zobrazia.

4.5.3 Spustenie rozšírenia Datastore

Datastore je nainštalovaný spolu CKANom. Je potrebné ho aktivovať pridaním do zoznamu aktívnych rozšírení `ckan.plugins` v konfiguračnom súbore. Následne je potrebné vytvoriť databázu s názvom `datastore`. K nej povoliť prístup užívateľovi ckan, a vytvoriť ešte jedného užívateľa, ktorý bude mať prístup k tejto databáze iba na čítanie. To je potrebné následne pridať do konfiguračného súboru CKANu za parametre `ckan.datastore.write_url` pre užívateľa ckan a `ckan.datastore.read_url` pre nové užívateľa s obmedzeným prístupom. Toto pripojenie je vo formáte sqlalchemy.

Datastore je potrebné plniť dátami. Automaticky to umožňujú rozšírenia Datpusher a Xloader. Obe tieto rozšírenia a ich rozdieli som už popísal.

Datapusher je potrebné spustiť ako samostatnú službu. Následne v konfiguračnom súbore CKANu je potrebné nastaviť adresu tejto služby pomocou paremetru `ckan.datapusher.url` a určiť formáty, ktoré budú datapusherom automaticky spracované a predané do Datastoru, parametrom `ckan.datapusher.formats`.

Alternatívne XLoader stačí nainštalovať ako obyčané rozšírenia a spustiť ako službu vrámci CKANu. Kompletný návod sa nachádza v repozitári na githube projektu [45].

4.5.4 Vytváranie náhľadov z dát

Pomocou ďalších rozšírení je možné automaticky vytvárať náľady pre rôzne typy súborov [39]. Pre použitie niektorého z nich je potrebné ho pomocou konfiguračného súboru aktivovať a povoliť pridaním do parametrov `ckan.plugins` a `ckan.views.default_views`.

Na zobrazenie textových súborov sa využíva rozšírenie `text_view`, ktoré vytvára náhľady pre súbory vo formátoch json, xml a txt. Podrobnejšie verzie týchto formátov sa možu nastaviť v konfiguračnom súbore pomocou nastavení `ckan.preview.json_formats`, `ckan.preview.xml_formats` a `ckan.preview.text_formats`. Na zobrazovanie obrázkov sa použije rozšírenie `image_view` a formáty sa špecifikujú parametrom `ckan.preview.image_formats`. Oba tieto rozšírenia sú už nainštalované spolu s CKANom.

Rozšírenie Datastore umožňuje okrem spracovania tabuľkových súborov vytvárať aj ich náhľady. Do parametra `default_views` je potrebné pridať hodnotu `recline_view`. Ďalej pomocou rozšírenia `pdf-view` je možné zobrazovať pdf súbory a pomocou rozšírenia `office-docs` kancelárske súbory produktov Microsoft Office alebo OpenOffice.

V dokumentácii sú spomenuté ďalšie rozšírenia, ktoré vytvárajú zobrazenia pre mnohé ďalšie typy súborov [39]. K dispozícii je tu ešte rozšírenie `dashboard`, ktoré umožní rozloženie viacerých druhov náhľadov do podoby dashboardu.

4.5.5 Podpora pre DCAT

Aktuálny profil podporovaný europskou úniou je DCAT AP v2 [28], ktorý bol vydaný v novembri 2019. Pri implementácii kompatibilného profilu sa však budem držať doporuční Ministerstva vnútra Českej republiky vydané v apríly 2019, aby bol CKAN katalóg kompatibilný s Národným katalógom [29].

V prvom rade je potrebné nainštalovať rozšírenie dcat. Podrobný a zrozumiteľný postup sa nachádza v repozitári na github [48]. Ďalej nainštalujeme ešte jedno pomocné rozšírenie `odzcdataset` [54]. Toto rozšírenie pridáva do webového rozhrania možnosť editovať niektoré nové atribúty, ktoré bude potrebné vložiť do dcatu.

Rozhranie pre lokálne katalógy vyžaduje dve triedy. Prvou triedou je Dátová sada. Tá musí obsahovať nasledovné atribúty:

- title - názov datovej sady
- description - popis dátovej sady
- theme - téma alebo kategória dátovej sady
- accrualPeriodicity - periodicita jej aktualizácie
- keyword - kľúčové slová alebo značky
- spatial - súvisiace geografické územie
- distribution - distribúcia dátovej sady, trieda obsahujúca informácie o dátach

Atribúty title, description, keyword a distribution podporuje a doplní ckan automaticky. Názov a popis datasetu sa doplný prirodzene. Keyword sú v CKANe reprezentované pomocou tagov (značiek). Distribúcia je samotný súbor, ktorý patrí datasetu a tvorí samostatnú triedu. S doplnením ostatných atribútov nám pomôže rozšírenie `odczdataset`. V užívateľskom rozhraní nás budú zaujímať nasledujúce položky:

- "Periodicita aktualizace" - sem je potrebné vložiť odkaz na hodnotu v slovníku, ktorý definovala Európska únia. Odkaz je v tvare `http://publications.europa.eu/resource/authority/frequency/{HODNOTA}` [31]. Táto hodnota sa následne správne namapuje na atribút `accrualPeriodicity`.

- "URI souvisejíciho geografického území" - tu je potrebné opäť vložiť odkaz a to v tvare `https://linked.cuzk.cz/resource/ruian/{TYP}/{KOD}`, kde TYP a KOD je potrebné nájsť na stránkach Čerského úradu zememeračského a katastrálneho [32]. Táto hodnota sa potom namapuje na atribút spatial.

- "URI konceptu klasifikace EUROVOC" - toto pole bolo pôvodne využívané na klasifikáciu EUROVOC, ktorá patrí pod dobrovoľné atribúty v dcate. Keďže je ukladané a mapované na atribút theme, vieme ho použiť aj na tému dátovej sady. Do tohto poľa je potrebné zadať odkaz v tvare `http://publications.europa.eu/resource/authority/data-theme/{HODNOTA}` [30].

Týmto sú vyplnené všetky povinné atribúty pre triedu dataset.

Druhou triedou je Distribúcia dátovej sady, ktorá predstavujú nahratý súbor (zdroj v CKANe). Tu sú povinné nasledovné atribúty:

- "Specifikace podmínek užití"

- downloadURL - odkaz na stiahnutie

- format - formát súboru

- mediaType - media type súboru podľa zoznamu IANA

Atribúty format a mediaType sú prístupné v užívateľskom rozhraní a správne namapované. Pre správne mapovanie zvyšných dvoch atribútov je potrebné buď vytvoriť nový, alebo editovať existujúci dcat profil. Pre stručnejšiu ukážku to znázornim na už existujúcom profile.

Súbory rozšírenia `dcat` sa nachádzajú na ceste `$CKAN_VENV/src/ckanext-dcat` v inštalácií CKANu. $CKAN_VENV je skratka do virtuálneho prostredia, v ktorom CKAN beží. Profily sú následne implementované v súbore `ckanext/dcat/profiles.py`. Ako hlavný profil je nastavený `EuropeanDCATAPProfile`, ktorý je reprezentovaný triedou v jazyku Python. Následne je potrebné nájsť funkciu `graph_from_dataset`, ktorá vytvára RDF graf datasetu.

Najskôr sa budem zaoberať url adresou. CKAN mapuje na atribút downloadURL adresu, pokiaľ je súbor hostovaný niekde inde a pri vytváraní zdroja bol použitý odkaz (nie nahraný súbor). Pre nahrané súbory priamo do CKANu sa použije atribút accessURL, ktorý obsahuje odkaz na súbor v CKANe. Tento odkaz je potrebné namapovať na atribút downloadURL. V časti kódu, kde sa kontroluje obsah downloadURL a vyhodnotí sa, že sa má použiť accessURL pridáme rovnaké volanie pre downloadURL. Toto volanie je v tvare `self._add_triple_from_dict(resource_dict, distribution, DCAT.downloadURL, 'url', _typeURIRef)`. Týmto bude mať každý súbor v CKANe atribút downloadURL so správnou adresou na stiahnutie.

Komplikovanejšie to je s "Specifikace podmínek užití". Tento atribút bol vytvorený špecificky pre Českú republiku a môže naberať hodnôt:

- pú:autorské-dílo <https://data.gov.cz/podmínky-užití/neobsahuje-autorská-díla/>
- pú:databáze-jako-autorské-dílo <https://data.gov.cz/podmínky-užití/není-autorskoprávně-chráněnou-databází/>
- pú:databáze-chráněná-zvláštními-právy <https://data.gov.cz/podmínky-užití/není-chráněna-zvláštním-právem-pořizovatele-databáze/>

- pú:osobní-údaje
  <https://data.gov.cz/podmínky-užití/neobsahuje-osobní-údaje/>

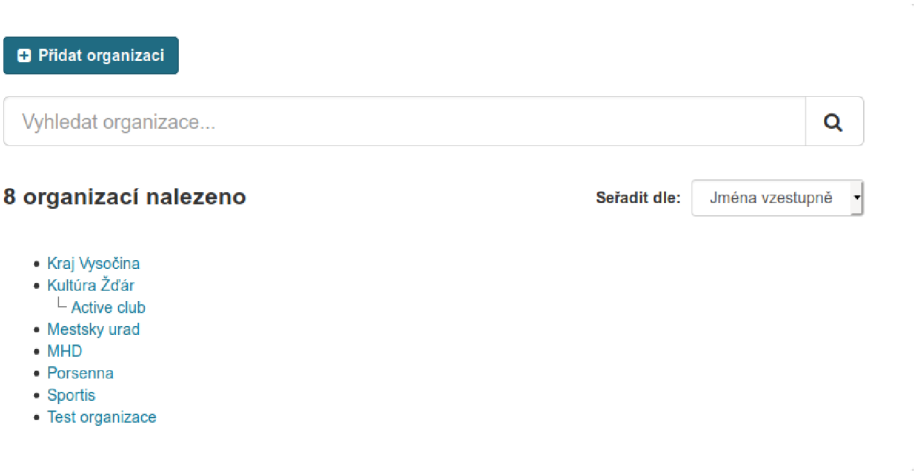
Ich vysvetlenie a spôsob použitia sa nachádza na stránkach otvorených dát Českej republiky [34]. Na uloženie do metadát v CKANe tu vieme využiť pole "Odkaz na podmínky užití", ktoré nám pripravilo rozšírenie `odczdataset`. Do tohto poľa uložíme jeden z odkazov v zozname vyššie. To sa namapuje na atribút `license_link` v CKANe. Ten získame, vytvoríme potrebnú štruktúru, t. j. objekt specifikace, v ktorom sa bude nachádzať samotný odkaz a vložíme ho do RDF grafu. Vzorový zdrojový kód uvádzam v prílohe B.

Na záver by som chcel zhrnúť, že toto riešenie síce správne mapuje dcat atribúty, ale nie je práve najelegantnejšie. Veľkým zjednodušením by bolo vytvoriť rozšírenie (prípadne doplniť a aktualizovať rozšírenie `odczdataset`), ktoré vytvorí zrozumiteľnejšie polia vo webovom rozhraní a dcat profil, ktorý chýbajúce atribúty namapuje. Polia by potom mohli byť prezentované ako výber z možností a nebude potrebné vkladať dlhé a zložité URL adresy, tie by sa mohli generovať až v profile. V prípade zmeny verzie dcatu by potom stačilo zmenu vykonať v jednom rozšírení a to následne aktualizovať. Nemusel by si to každý správca katalógu udržovať sám.

4.6 Práca s dátovými sadami

V spolupráci s mestom Žďár nad Sázavou bol nástroj CKAN tiež testovaný. Od mestských firiem boli dáta získavané, ukladané do CKANu a následne z nich boli vytvárané vizualizácie. Dáta boli následne aktualizované, vytvárané boli aj rôzne hierarchie organizácií a tiež náhľady rôznych typov súborov. Príklad, ako CKAN dokaže deliť organizácie, sa nachádza na obrázku 4.1 a príklad, ako vyzerá taká dátová sada v CKANe na obrázku 4.2.

Počas testovania boli k dispozícii dátové sady z rôznych časti mesta. Mnoho z organizácii bolo na začiatku skeptických a odvolávali sa na legislatívu a ochranu osobných údajov. Po ďalšom vysvetľovaní a presviedčaní to začali niektorí brať ako možnosť prezentovať sa, prípadne v tom videli zjednodušenie ich práce. Dáta bolo za cieľ získavať hlavne

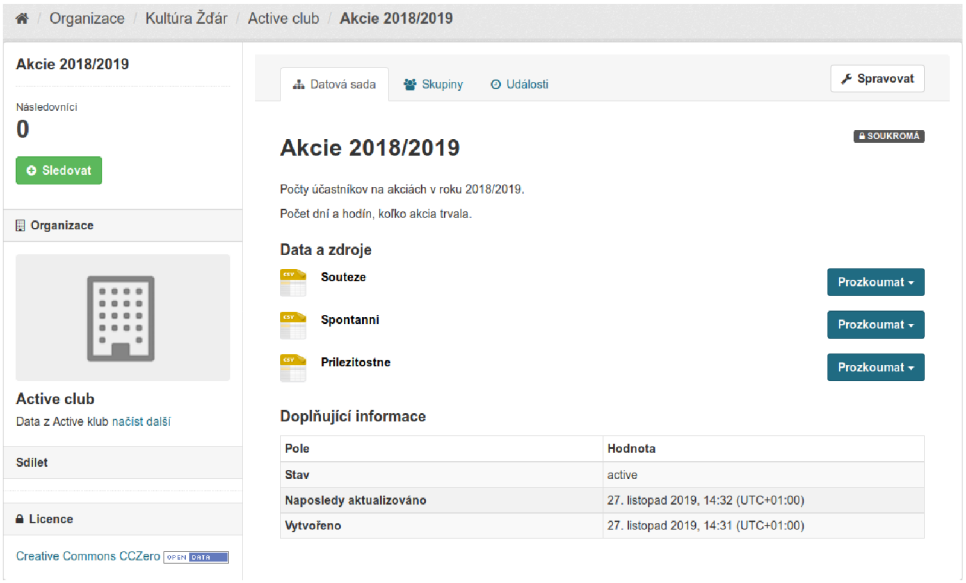


Obr. 4.1: Príklad rozdelenia organizácii v CKANe

anonymizované a v čo najsurovejšej podobe. Zasalené boli dáta v najrôznejších formátoch.

Najlepšie súbory boli v tabuľkových formátoch csv alebo xls. Tie dokáže CKAN spracovať a uložiť do databázy. Vizualizačný nástroj sa potom vie pripojiť priamo na databázu CKANu a je možné dáta ďalej spracovávať a vizualizovať pomocou grafov. K dispozícii boli aj rôzne číselníky a súbory v cudzích formátoch, s ktorými bez ďalšieho vysvetlenie bolo nemožné pracovať. Telorovateľné boli aj prehľadové štatistiky. Ďalej boli získané rôzne tabuľky s estetickou úpravou a rôzmanitým rozložením dát v nich. Po troche spracovania z nich však bolo možné získať strukturovanejšie dáta a exportovať ich do formátu CSV.

Najexotickejším typom dát, aký sme získali, bol asi súbor z RRD databázy (Round robin database). Ide o špeciálny typ real time databázy, ktorý slúži napríklad na monitorovanie prenosu dát na internete. Ten sa nakoniec podarilo konvertovať do xml formátu, v ktorom boli dáta čitateľnejšie a jednoduchšie na spracovanie, a CKAN z nich vedel spraviť textový náhľad. Neskôr ale bolo povedané, že organizácia prechádza na modernejší systém a databázu, takže momentálne rrd súbory spracovávať nie je potrebné.



Obr. 4.2: Príklad dátovej sady uloženej v CKANe

Do kategórie najťažšie spracovateľných súborov určite patrí snímka obrazovky obsahujúca dáta, ktorá bola zaslaná vo formáte pdf.

Ďalšími firmami, ktoré dáta zaslali, bola napríklad firma Zdar a.s., ktorá prevádzkuje v meste mestskú hromadnú dopravu. Zo začiatku boli zasielané iba prehľadové štatistiky, no neskôr prišla aj vzorka surových dát. Podobne sa to vyvíjalo aj s dátami od mestskej firmy Sportis, ktorá má nastarosti športové a rekreačné zariadenia v meste. Štatistiky aj surové dáta prichádzali vo forme excelovských tabuliek, no štrukturované dáta bolo jednoduché a priamočiare previesť do formátu csv.

Existujú však aj mestské firmy, ktoré dodnes nereagovali na žiadosti a prosby o dáta a to ani po konzultáciách s koordinátorom Smart cities pre mesto Žďár nad Sázavou. Ďalej existujú aj organizácie, ktoré zo zákona dáta, o ktoré sme žiadali, poskytovať nemôžu. Ide napríklad o dáta z hygienických staníc.

V archíve záverečnej práce sa náchádajú príklady dát, s ktorými bolo potrebné pracovať. Keďže zatiaľ tieto dáta neboli publikované vo forme Otvorených dát oficálne, boli v nich pozmenené niektoré hodnoty, no ich štruktúra ostala nezmenená.

5 Vizualizácia dát

V tomto kroku máme dáta zhromaždené vo forme otvorených dát a je načase ich začať využívať. Táto úloha je z môjho pohľadu zložitejšia. Na ukladanie dát existujé legislatíva a množstvo iných pravidiel, ktoré obmedzujú výber správneho nástroja. Spracovanie a zobrazovanie dát je v tomto smere voľnejšie. Každý preferuje iný spôsob a tých spôsobov môže byť mnoho. Niekto má rád grafický softvér, iný zase príkazový riadok. Niekto rád dáta analyzuje pomocou programovacieho jazyka, napríklad python, inému zase stačí tabuľkový procesor, napríklad excel. Tu je rozhodujúce, aké dáta mestá vlastne zbierajú a aké funkcie vyžadujú.

Budú predstavené 2 prístupy prezentácie dát. Jedným je použitie hotového nástroja a druhým tvorba vizualizácií pomocou knižníc. Záleží na konkrétnom meste, ktorý a či vôbec niektorý z nich bude použiteľný.

Hlbšie porovnanie týchto prístupov a následný konkrétny výber pre mesto Žďár nad Sázavou bude predmetom nadväzujúcej diplomovej práce.

5.1 Nástroje

Prvým prístupom je použitie softvéru, ktorý ponúka grafické rozhranie na prístup, spracovanie a zobrazovanie dát. Opäť sa bude jednať o open source softvér. Výhodou takéhoto druhu nástroja je, že s ním vie, prípadne sa naučí pracovať aj užívateľ mimo IT sféry. Je mu prezentované webové užívateľské rozhranie. Proces je vždy rovnaký. Užívateľ si vyberie dáta. Tie si vyfiltruje na dáta, s ktorými chce pracovať. Aplikuje funkcie na transformáciu dát, napríklad priemer alebo počet. Následne dáta zobrazí pomocou grafu. Grafy je potom možné organizovať do dashboardov. Oboje, grafy aj dashboardy, sa dajú publikovať na danom portáli alebo vložiť na vlastné prezentačné stránky.

Nevýhodou takéhoto softvéru je, že funkcie sú dané a môžu existovať obmedzenia. Kedže ide o open soruce softvér, tieto obmedzenia sa dajú odstrániť ich doimplementovaním. Ďalšou nevýhodnou môže byť nedostatočná lokalizácia. Ani jeden nástroj, ktorý spomeniem, nie je v čase písania tejto práce dostupný v českom alebo slovenskom

jazyku. Riešenie je podobné ako pri chýbajúcej funkcionalite. Nástroje sú open source a preto je možné prispieť lokalizáciou do ľubovoľného jazyka.

Pri každom nástroji budú spomenuté základné funkcie a potom výrazné odlišnosti, ak nejaké existujú.

5.1.1 Apache superset

Apache superset je open source nástroj na prácu s dátami. Jeho zameraním je tvorba štatistík a grafov pre firmy [55]. Ponúka množstvo rôznych vizualizácií vrátane máp (pomocou deck.gl [56]) a väčšina z nich môže mať aj 3D podobu [57]. Superset podporuje množstvo SQL databáz na prácu s dátami. Ďalej dokáže efektívne pracovať s real time databázou Druid [58].

Superset ponúka jednoduché prostredie na prácu s dátami. Po výbere dát a druhu vizualizácie sa zvolí metrika, podľa ktorej sa bude graf vytvárať. Príklad práce s dátami je na obrázku 5.1. Ak je vyžadovaná komplexnejšia práca s dátami, ponúka superset aj SQL editor. Umožňuje organizáciu grafov do dashboardov, ktoré je možné publikovať.

Superset ponúka viacužívateľské prostredie. Užívatelia sa dajú ďalej organizovať do rôl a je možné im priradiť rôzne politky prístupu k dátam, grafom a dashboardom.

Ohľadom prístupu k dátam, po napojení sa na databázu je potrebné každú tabuľku, s ktorou chceme pracovať, pripojiť zvlášť. Na jednej strane to môže byť pracné, no na druhej strane to ponúka lepšiu kontrolu nad tým, kto má k čomu prístup. Naimportujem len tabuľky, ktoré chcem zobrazovať a každej tabuľke viem podrobne priradiť práva prístupu pre užívateľov.

Ďalšou funkciou supersetu je možnosť nahrať CSV súbor priamo, ktorý sa spracuje a uloží do vybranej databázy a je možné s dátami pracovať.

V čase písania tejto práce je Superset v stave prechodu pod nadáciu Apache[55].

Obr. 5.1: Príklad nástroja Apache Superset

5.1.2 Redash

Redash je nástroj pre firmy na spracovanie a vizualizáciu dát [59]. Je to open source nástroj, ktorý využíva mnoho veľkých spoločností [60].

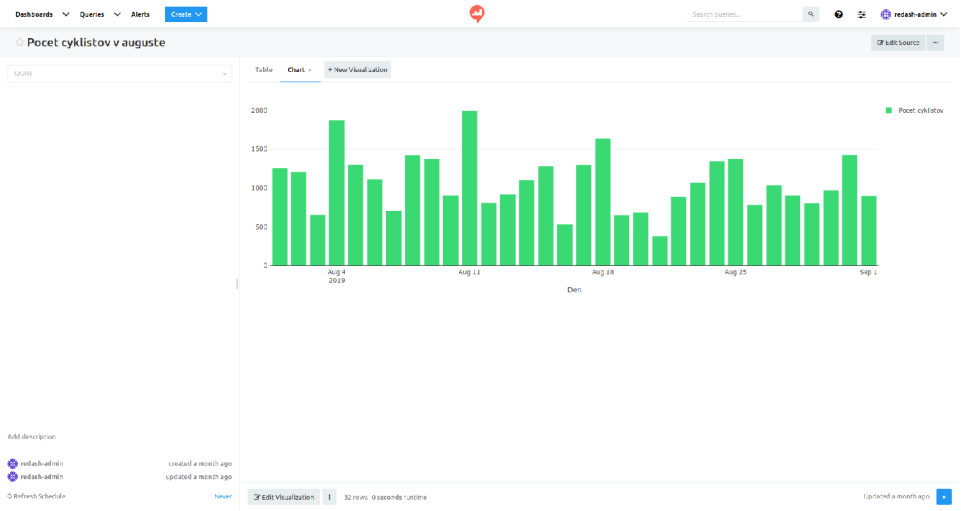
Funkcie sú podobné ako v prípade nástroja apache superset. Podporuje viacej užívateľov organizovaných do skupín. Dokáže sa pripojiť na veľké množstvo databáz, SQL aj real time. Oproti supersetu, tentokrát pripojenie k databáze načíta všetky tabuľky a politiky prístupu sa dajú obmedziť len na celé databázy. Nie je možné granulárne obmedzenie podľa tabuliek.

Samozrejmosťou je možnosť vytvárať grafy a organizovať ich do dashbaordov, ktoré môžu byť potom verejné. Uživateľké rozhranie Redashu a zobrazenie dashboardu je možné vidieť na obrázku 5.2. Na verejnom dashboarde sa v spodnej časti zobrazí logo Redash, ktoré slúži niečo ako reklama projektu. Nenašiel som možnosť, či a ako je to možné odstrániť.

Z môjho pohľadu ponúka redash príjemnejšie a jednoduchšie užívateľské prostredie. Užívateľ má rýchly prístup ku grafom a dashboardom. Ďalšie možnosti sú potom umiestnené v nastaveniach.

Užitočnou funkciou redashu sú upozornenia. Dokáže monitorovať dáta a pri výkyve hodnoty mimo stanovenú mez sa zašle upozornenie,

ktoré môže byť vo forme emailu, notifikácie na četovú aplikáciu alebo aj vlastnú aplikáciu pomocou webhook.



Obr. 5.2: Príklad dashboardu v nástroji Redash

5.1.3 Metabase

Metabase je opäť open source nástroj na prácu s dátami. Príklad tvorby vizualizácie v nástroji Metabase je na obrázku 5.3. Zo začiatku sa užívateľské prostredie môže javiť neprirodzené a zložité. Metabase k dátam pristupuje formou otázok [61]. Otázka môže byť jednoduchá alebo zložitá. Jednoduchá otázka umožňuje výber, filtrovanie a zhrnutie dát a následnú tvorbu vizualizácie. Zložitá otázka podporuje výber viacerých dát, ich spájanie, vlastné stĺpce, filtrovanie a zhrnutie. Spájanie funguje ako spájanie tabuliek v databáze a vlastné stĺpce ponúkajú tvorbu nového stĺpca z hodnôt v iných stĺpcoch a zároveň umožňuje aj aplikovanie funkcie, napríklad súčet stĺpcov. V princípe to pracuje ako sql editor, v ktorom ale nepíšem SQL syntax, ale vyberám graficky čo potrebujem. Z pohľadu užívateľa je to najbohatšia práca s dátami spomedzi týchto troch nástrojov bez použitia jazyka SQL. No a v prípade potreby je možné tento jazyk použiť. SQL editor v tomto prípade tvorí akýsi tretí typ otázky.

Bohatšie funkcie sa nachádzajú aj v oblasti tvorby dashboardov. Opäť viem na dashboardy pridávať grafy. Novým interaktívnym prvkom v dashboardoch sú filtre [62]. Stáva sa, že potrebujeme veľmi podobné grafy líšiace sa iba v niektorých premenných, napríklad dátum. Filtre dokážu obmedziť dáta, ktoré sa na zobrazenie grafu použijú na úrovni dashboardu. Na graf použijem všetky dáta čo mám a uživateľovi dám možnosť vyfiltrovať, čo chce vidieť.

Viem filtrovať podľa:

- času - čas, dátum, mesiac, rok, kvartál, relatívny čas (posledný mesiac), rozsah času
- polohy - mesto, štát, krajina
- identifikátora - id v databáze
- ostatné - kategórie, hodnotenie a iné

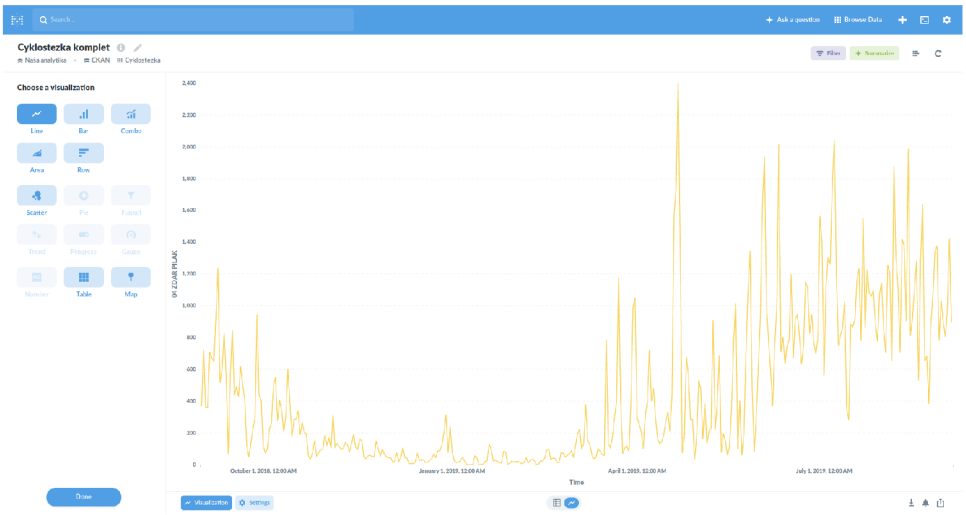
Čas a identifikátor metabase zistí z databázy. Poloha sa da určiť stĺpcu tabuľky, ktorý takéto informácie obsahuje. Možnosť ostatné potom pokrýva zvyšné položky, ktoré môžu byť stĺpcom priradené, napríklad cena, email, webová adresa a iné. Vtedy to funguje ako vyhľadávanie a dáta sa vyfiltrujú pomocou jeho výsledkov. Následne sa podľa filtra graf alebo grafy aktualizujú, ale nezasahuje sa do originálnej otázky. Po refreshy alebo vynulovaní filtra sa zobrazí pôvodný graf.

Grafy a dashboardy je opäť možné označiť za verejné a publikovať ich v metabase, prípadne vložiť na vlastné stránky. Open source verzia tohoto projektu však pod graf alebo dashboard vloží logo a odkaz na ich domovskú stránku. Dá sa to pochopiť aj ako reklama na projekt v bezplatnej verzii. Metabase ponúka aj platenú verziu, v ktorej sa už odkaz nenachádza [63].

Tabuľkám sa dajú priradiť segmenty (pomenované filtre) a metriky (predpočítaný výsledok funkcie), na ktoré sa dá rýchlejšie odkazovať v otázkach [64]. X rays je funkcia, kde sa metabase pokúsi automaticky preskúmať a vytiahnúť dôležité informácie z tabuľky [65].

Užívateľov je možné organizovať do skupín. Metabase potom ponúka granulárne povolenia na úrovni databáz aj tabuliek pre dané skupiny. Taktiež je možné skrývať tabuľky alebo stĺpce v tabuľkách globálne pre všetkých užívateľov.

Metabase pôsobí ako silný nástroj v oblasti práce s dátami, aj keď to môže byť na úkor jednoduchosti prvého použitia. No keď užívateľ pochopí jeho rozloženie a funkcie, dokáže rýchlo a efektívne pracovať s dátami a vytvárať elegantné vizualizácie.



Obr. 5.3: Príklad tvorby vizualizácie v nástroji Metabase

5.1.4 RAWGraphs

RAWGraphs je webová aplikácia, ktorá slúži na tvorbu statických vizualizácií [66]. Na vizualizáciu využíva knižnicu D3.js. Veľkou výhodou nástroja RAWGraphs je jednoduchosť používania. Dovoľuje importovanie dát z rôznych tabuľkových formátov, prípadne aj samotné kopírovanie dát z tabuľkovej aplikácie. Nasleduje možnosť výberu typu grafu, ktorý chceme na vizualizáciu použiť. RAWGraphs umožňuje aj tvorbu vlastných grafov, ktoré sa potom adaptujú na existujúce užívateľské rozhranie [67]. Ďalej nasleduje namapovanie dát na jednotlivé osi a premenné grafov. Čiastočne je možné modifikovať rozloženie, rozostup a farbu grafov. Ak by modifikácia grafov bola nedostatočná, RAWGraphs dokáže exportovať graf do formátu SVG. Ten sa dá potom editovať v rôznych programoch na úpravu vektorových obrázkov, prípadne vložiť na webové stránky alebo ďalej spracovávať v inom softvéri. Ďalším možným formátom je obrázkový formát PNG.

RAWGraphs ponúka jednoduchú cestu na vytvorenie grafu z dát. To môže byť obmedzujúce pri použití komplexnejších dát alebo zložitejších vizualizácií. Druhý prípad sa dá vyriešiť tvorbou vlastného grafu, čo je ale netriviálna úloha. Nie je to platforma všetko v jednom na tvorbu a prezentovanie vizualizácií, no ponúka jednoduchú možnosť vytvárania vizualizácií, oproti použitiu knižnice na tvorbu vizualizácií, ako napríklad D3.js.

5.1.5 Chart studio

Chart studio je produkt firmy Plotly [68]. Je to webová aplikácia, ktorá funguje ako online editor grafov. Je nutná registrácia, po ktorej je možné vytvárať a zdieľať grafy a dashboardy. V bezplatnej verzii je v ponuke iba obmedzený, no pre niekoho dostatočný počet grafov. Chart studio je postavené na open source platforme React chart editor, ktorá umožňuje vytvoriť si vlastnú aplikáciu na tvorbu grafov [69]. Táto platforma je taktiež vyvíjaná firmou plotly. React chart editor buduje webové užívateľské rozhranie, ktoré v základe obsahuje panel na import a spracovanie dát, tvorbu a export vizualizácie. Do tohto panela je možné potom implementovať a pridávať ďalšie funkcie podľa potreby.

5.2 Knižnice

Na začiatku kapitoly bolo spomenuté, že každý pracuje s dátami iným spôsobom. A tým svojim spôsobom sa dostane aj k nejakému výsledku, ktorý je potrebné prezentovať. Jeho nástroj môže, ale aj nemusí obsahovať funkcionalitu na tvorbu vizualizácie. Taktiež, ak by každý vytvoril vyzualizáciu vo svojom nástroji a tie sa zhromaždili, každá by vyzerala inak.

Preto druhou možnosťou vizualizácie dát je tvorba grafov pomocou vizualizačných knižníc. Tie sa dajú vložiť na vlastné prezentačné stránky mesta, vytvárať vlastné dashboardy a publikovať výsledky v jednotnej forme. V tomto prípade je ale potrebný človek, ktorý sa rozumie práci s dátami aj tvorbe stránok. Musí vedieť získať výsledky z dát a prezentovať ich pomocou knižnice.

Jedným z dôvodov na použitie tohto riešenia môže byť aj nedostatočná funkcionalita nástrojov spomenutých v predchádzajúcej časti. Keď sa niečo vytvára od základov, dá sa to jednoduchšie prispôsobiť prvotným požiadavkám.

5.2.1 Príklady knižníc

Základnou knižnicou pre vizualizáciu dát je knižnica D3.js. Je to javascriptová knižnica na manipulovanie html dokumentu na základe dát. Dokáže napojiť dáta na časť html dokumentu a následne ho transformovať do stavu, aby vznikla ich vizualizácia [70]. D3 neponúka žiadnú ďalšiu rozšírenú funkcionalitu, snaží sa iba pracovať s dokumentom rýchlo a efektívne na najnižšej úrovni za použitia webových štandardov html, css, javascript a svg.

Tým, že D3 pracuje na nižšej úrovni, dokáže vytvárať komplexné vizualizácie. Na tvorbu vyzualizácie treba správne nastaviť rôzne operátory, dáta a ďalšie prvky knižnice D3. To spôsobuje, že je potrebný dlhý a komplikovaný kód na úpravu html dokumentu do požadovanej podoby. A to aj na jednoduchý graf.

D3 svojím nízkoúrovňovým prístupom však výborne slúži ako základová knižnica pre rôzne nadstavby. Knižnice postavené na D3 využívajú jej bohaté možnosti na tvorbu vizualizácií a skrývajú komplexnosť do jednoduchších funkcií. Kde je v D3 potrebné správne nadefinovať množstvo vlastností pre každú hodnotu v dátach, nadstavby ich aplikujú za nás. Programátor si len vyberie z ponuky existujúceho grafu. Zvyčajne sa vytvorí javascriptový objekt. Tomu sa pridajú atribúty, napríklad druh grafu alebo farba a zavolajú sa rôzne funkcie na pridanie dát, animácie, umiestnenie grafu v dokumente a podobne. Príkladom takejto dastavby môže byť knižnica C3.js [71].

Príkladom inej knižnice je napríklad knižnica Echarts, ktorú využívajú aj ďalšie nástroje na tvorbu vyzualizácii, napríklad služba Datamatic [72]. Echarts je webový multiplatformový framework na jednoduchú tvorbu vizualizácií [73]. Funguje na princípe nastavovania parametrov a atribútov v objekte pomocou formátu JSON. Dokáže vytvárať 2D aj 3D vizualizácie, zobrazenie dát na mape či zobrazenie real time dát. Pre rôzne druhy vizualizácií sú k dispozícií preddefinované štýly spolu so vzorovým zdrojovým kódom [74].

Ďalším príkladom može byť spoločnosť Plotly [75], ktorá vytvára nielen open source vizualizačné knižnice [76], ale aj framework na tvorbu webových aplikácií, ktoré potom služia na vizualizáciu dát [77].

Ďalej samozrejme sú k dispozícii aj komerčné knižnice, ktoré môžu ponúkať lepšiu podporu pri ich používaní. Príkladom takýchto knižníc sú Anycharts [78] alebo ZingChart [79].

6 Záver

Táto práca sa zaoberala platformami využiteľnými pri implmentácii konceptu Smart cities v mestách. Za úlohu bolo nájsť, porovnať a odporučiť platformu na správu dát a platformu na ich vizualizáciu.

Po vytvorení prehľadu dostupných platforiem na správu dát boli porovnané s požiadavkami mesta Žďár nad Sázavou, s ktorým sa na tejto práci spolupracovalo. Ďalej boli získané požiadavky na lokálne dátové katalógy, ktoré určuje Národný katalóg otvorených dát v Českej republike a tieto požiadavky boli tiež porovnané s funkciami platforiem. Nakoniec z toho vyplynulo, že z vybraných platforiem je vhodná iba platforma CKAN.

Následne bola táto platforma podrobnejšie rozpísaná, ako a prečo vyhovuje, aké ma funkcie a čo všetko je potrebné nastaviť, aby mohol byť CKAN prevádzkovaný ako portál na správu mestských dát.

Ďalej bol CKAN testovaný na reálnych dátach v spolupráci s mestom Žďár nad Sázavou. Od mesta sme získavali dáta a vkladali ich do katalógu. CKAN z tohto testovania vyšiel ako vhodné riešenie.

Úlohou tejto práce bolo aj porovnanie platforiem na vizualizáciu zozbieraných dát. Ukázalo sa však, že práca s výberom a testovaním platformi CKAN bola náročnejšia, než sa na začiatku predpokládalo. Bolo potrebné pochopiť a správne nastaviť množstvo premenných, aby CKAN vyhovoval. Jednou z nich je aj formát DCAT, ktorý sa využíva na prácu s metadátami v Európskej únii, a ktorý vyžadoval mnoho času a úsilia na jeho spracovanie. Niektoré vizualizačné nástroje som v práci spomenul a vymenoval ich základné funkcie, no ich hlbšie porovnanie, nasadenie v meste Žďár nad Sázavou a následné testovanie bude mať za úlohu nadväzujúca diplomová práca.

Ďalšia téma, ktorá sa počas testovania vyskytla a bolo by vhodné sa jej ďalej venovať je spracovanie real time dát. Tie tvorili veľkú časť dát,ktoré sa testovali v Žďári a existuje veľa ďalších možností a postupov na ich spracovávanie.

Používanie formátu DCAT v Českej republike je tiež téma vhodná spracovania. CKAN podporuje DCAT profily, ktoré určujú, aké metadáta musia byť definované. V základnom profile je definovaná už väčšina týchto metadát a na príklade bolo ukázané, ako je možné pridať tie chýbajúce. Bolo by vhodné to zjednotiť do rozšírenia pre CKAN,

ktoré by mohlo byť používané spolu s CKANom a zabezpečovalo by aktuálnosť a kompatibilitu s Národným katalógom.

Na koniec je vhodné dodať, že koncept Smart cities sa stále vyvíja. Už počas tvorby tejto práce dochádzalo k zmenám, na ktoré bolo potrebné reagovať. Európska únia vytvorila nový DCAT profil a boli vytvárané nové rozšírenia pre CKAN, ktoré počas testovania neboli k dispozícii a umožňujú lepšiu funkcionalitu a prácu s CKANom.

V meste Žďár nad Sázavou prebehla aj prezentácia dosiahnutých výsledkov. Táto práca bude s najväčšou pravdepodobnosťou použitá ako základ pre portál otvorených dát v Žďári a je základom rozvíjania ďalších aktivít v meste.

Bibliografia

1. RUSSO, Francesco; RINDONE, Corrado; PANUCCIO, Paola. The process of smart city definition at an EU level. In: 2014, zv. 191, s. 979–989. ISBN 9781845648206. Dostupné z DOI: 10.2495/SC140832.
2. *Defining Open Data* [online]. Open Knowledge Foundation Blog [cit. 2019-12-05]. Dostupné z: https://blog.okfn.org/2013/10/03/defining-open-data/.
3. *Co jsou otevřená data?* [online]. Otevřená data [cit. 2019-12-09]. Dostupné z: https://opendata.gov.cz/informace:start.
4. *What are open licenses* [online]. Year of open [cit. 2019-12-05]. Dostupné z: https://www.yearofopen.org/what-are-open-licenses/.
5. *Conformant Licenses* [online]. Open Definition [cit. 2019-12-05]. Dostupné z: https://opendefinition.org/licenses/.
6. *Our Activities* [online]. European Data Portal [cit. 2019-12-09]. Dostupné z: https://www.europeandataportal.eu/en/what-we-do/our-activities.
7. *Národní katalog otevřených dat (NKOD)* [online]. Otevřená data [cit. 2019-12-09]. Dostupné z: https://opendata.gov.cz/nastroj:narodni-katalog-otevrenych-dat.
8. *Otevřená data v ČR: Portál pro poskytovatele* [online]. Otevřená data [cit. 2019-12-09]. Dostupné z: https://opendata.gov.cz/.
9. *Formulář* [online]. Otevřená data [cit. 2019-12-09]. Dostupné z: https://data.gov.cz/formul%C3%A1%C5%99/.
10. *About Duraspace* [online]. Duraspace [cit. 2019-11-20]. Dostupné z: https://duraspace.org/about/.
11. *About Fedora* [online]. Fedora [cit. 2019-11-20]. Dostupné z: https://duraspace.org/fedora/about/.
12. *Linked data* [online]. W3C, 2016 [cit. 2019-11-20]. Dostupné z: https://www.w3.org/wiki/LinkedData.
13. *RDF* [online]. W3C, 2014 [cit. 2019-11-20]. Dostupné z: https://www.w3.org/RDF/.

14. *About Dspace* [online]. Dspace [cit. 2019-11-20]. Dostupné z: https://duraspace.org/dspace/about/.

15. *Features* [online]. Dspace [cit. 2019-11-20]. Dostupné z: https://duraspace.org/dspace/about/features/.

16. *Functional Overview* [online]. Lyrasis Wiki - Dspace 6.x documentation [cit. 2019-12-09]. Dostupné z: https://wiki.lyrasis.org/display/DSDOC6x/Functional+Overview.

17. *Handle.Net© Registry* [online]. Corporation for National Research Initiatives [cit. 2019-12-09]. Dostupné z: https://www.handle.net/.

18. *CKAN, the world's leading Open Source data portal platform* [online]. CKAN [cit. 2019-11-19]. Dostupné z: https://ckan.org.

19. *CKAN Features* [online]. CKAN [cit. 2019-12-09]. Dostupné z: https://ckan.org/features/.

20. *Smart City Platform* [online]. INVIPO [cit. 2019-12-09]. Dostupné z: https://www.invipo.com/cs/.

21. *Koncept ve Smart City* [online]. INVIPO [cit. 2019-12-09]. Dostupné z: https://www.invipo.com/cs/chytre-mesto/.

22. *Invipo Funkce* [online]. INVIPO [cit. 2019-12-09]. Dostupné z: https://www.invipo.com/cs/funkce/.

23. *Invipo ve světě* [online]. INVIPO [cit. 2019-12-09]. Dostupné z: https://www.invipo.com/cs/reference/.

24. *Úvod* [online]. ASD Software [cit. 2019-12-09]. Dostupné z: https://www.asd-software.cz/.

25. *Open data* [online]. ASD Software [cit. 2019-12-09]. Dostupné z: https://www.asd-software.cz/reseni-a-sluzby/#open-data.

26. *Data Catalog Vocabulary (DCAT) - Version 2* [online]. W3C, 2019 [cit. 2019-11-20].

27. *RDF 1.1 Primer* [online]. W3C, 2014 [cit. 2019-11-20]. Dostupné z: https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/.

28. *DCAT Application Profile for data portals in Europe Version 2.0.0* [online]. European Commission [cit. 2019-12-05]. Dostupné z: https://joinup.ec.europa.eu/sites/default/files/distribution/access_url/2019-12/12f0dc1d-50b6-43e4-90c2-0afe213ac2be/DCAT_AP_2.0.0.pdf.

29. *Rozhraní katalogů otevřených dat* [online]. Otevřená data - Otevřené formálni normy [cit. 2019-12-05]. Dostupné z: https://ofn.gov.cz/rozhran%C3%AD-katalog%C5%AF-otev%C5%99en%C3%BDch-dat/2019-04-04/.

30. *Data theme* [online]. EU Vocabularies [cit. 2019-12-05]. Dostupné z: https://op.europa.eu/cs/web/eu-vocabularies/at-concept-scheme/-/resource/authority/data-theme/?target=Browse&uri=http://publications.europa.eu/resource/authority/data-theme.

31. *Frequency* [online]. EU Vocabularies [cit. 2019-12-05]. Dostupné z: https://op.europa.eu/cs/web/eu-vocabularies/at-concept-scheme/-/resource/authority/frequency/?target=Browse&uri=http://publications.europa.eu/resource/authority/frequency.

32. *Veřejný dálkový přístup k datům registru územní identifikace, adres a nemovitostí* [online]. CUZK [cit. 2019-12-05]. Dostupné z: https://vdp.cuzk.cz/.

33. *Propojená data - Registr územní identifikace, adres a nemovistostí* [online]. Otevřená data - Otevřené formálni normy [cit. 2019-12-09]. Dostupné z: https://ofn.gov.cz/propojen%C3%A1-data/draft/#R%C3%9AIAN.

34. *Stanovení podmínek užití otevřených dat* [online]. Otevřená data v ČR [cit. 2019-12-05]. Dostupné z: https://opendata.gov.cz/cinnost:stanoveni-podminek-uziti.

35. *File type* [online]. EU Vocabularies [cit. 2019-12-09]. Dostupné z: https://op.europa.eu/cs/web/eu-vocabularies/at-concept-scheme/-/resource/authority/file-type/?target=Browse&uri=http://publications.europa.eu/resource/authority/file-type.

36. *Media Types* [online]. IANA [cit. 2019-12-09]. Dostupné z: https://www.iana.org/assignments/media-types/media-types.xhtml.

37. *User Guide* [online]. CKAN [cit. 2019-11-19]. Dostupné z: https://docs.ckan.org/en/latest/user-guide.html.

38. *Organization hierarchy* [online]. CKAN Extensions [cit. 2019-12-01]. Dostupné z: https://extensions.ckan.org/extension/hierarchy/.

39. *Data preview and visualisation* [online]. CKAN Documentation [cit. 2019-12-06]. Dostupné z: https://docs.ckan.org/en/2.8/maintaining/data-viewer.html.

40. *API Guide* [online]. CKAN [cit. 2019-11-19]. Dostupné z: https://docs.ckan.org/en/latest/api/index.html.

41. *DataStore extension* [online]. CKAN Documentation [cit. 2019-12-01]. Dostupné z: https://docs.ckan.org/en/latest/maintaining/datastore.html.

42. *DataStore Extension - datastore_create* [online]. CKAN Documentation [cit. 2019-12-01]. Dostupné z: https://docs.ckan.org/en/latest/maintaining/datastore.html#ckanext.datastore.logic.action.datastore_create.

43. *Using the DataPusher* [online]. DataPusher [cit. 2019-12-01]. Dostupné z: https://docs.ckan.org/projects/datapusher/en/latest/using.html.

44. *DataPusher github repository* [online]. CKAN [cit. 2019-12-01]. Dostupné z: https://github.com/ckan/datapusher.

45. *Express Loader - quickly load data into DataStore. A replacement for DataPusher.* [online]. CKAN [cit. 2019-12-01]. Dostupné z: https://github.com/ckan/ckanext-xloader.

46. *DataStore Extension - Data Dictionary* [online]. CKAN Documentation [cit. 2019-12-01]. Dostupné z: https://docs.ckan.org/en/2.8/maintaining/datastore.html#data-dictionary.

47. *messytables: all your rows are belong to us* [online]. MessyTables [cit. 2019-12-01]. Dostupné z: https://messytables.readthedocs.io/en/latest/.

48. *ckanext dcat* [online]. CKAN [cit. 2019-12-05]. Dostupné z: https://github.com/ckan/ckanext-dcat.

49. *CKAN + DCAT* [online]. CKAN Extensions [cit. 2019-12-05]. Dostupné z: https://extensions.ckan.org/extension/dcat/.

50. *dcatde* [online]. GovDataOfficial [cit. 2019-12-09]. Dostupné z: https://github.com/GovDataOfficial/ckanext-dcatde.

51. *dcatapit* [online]. Geosolutions IT [cit. 2019-12-09]. Dostupné z: https://github.com/geosolutions-it/ckanext-dcatapit.

52. *CKAN Installation* [online]. CKAN [cit. 2019-12-08]. Dostupné z: https://docs.ckan.org/en/2.8/maintaining/installing/index.html.

53. *CKAN configuration file - CORS Settings* [online]. CKAN Documentation [cit. 2019-12-01]. Dostupné z: https://docs.ckan.org/en/latest/maintaining/configuration.html#cors-settings.

54. *odczdataset* [online]. Jakub Klimek [cit. 2019-12-05]. Dostupné z: https://github.com/jakubklimek/ckanext-odczdataset.

55. *Apache Superset (incubating)* [online]. Apache Superset [cit. 2019-12-01]. Dostupné z: https://superset.incubator.apache.org/.

56. *Deck.gl* [online]. Deck.gl [cit. 2019-12-01]. Dostupné z: https://deck.gl/#/.

57. *Visualizations Gallery* [online]. Apache Superset [cit. 2019-12-01]. Dostupné z: https://superset.incubator.apache.org/gallery.html.

58. *Druid* [online]. Apache Superset [cit. 2019-12-01]. Dostupné z: https://superset.incubator.apache.org/druid.html.

59. *Redash helps you make sense of your data* [online]. Redash [cit. 2019-12-01]. Dostupné z: https://redash.io/.

60. *Empowering data-driven decision making in thousands of teams across the globe* [online]. Redash [cit. 2019-12-01]. Dostupné z: https://redash.io/customers/.

61. *Asking questions in Metabase* [online]. Metabase Documentation [cit. 2019-12-01]. Dostupné z: https://www.metabase.com/docs/latest/users-guide/04-asking-questions.html.

62. *Dashboard Filters* [online]. Metabase Documentation [cit. 2019-12-01]. Dostupné z: https://www.metabase.com/docs/latest/users-guide/08-dashboard-filters.

63. *Pricing* [online]. Metabase [cit. 2019-12-01]. Dostupné z: https://www.metabase.com/enterprise/pricing.

64. *Creating Custom Segments and Metrics* [online]. Metabase Documentation [cit. 2019-12-01]. Dostupné z: https://www.metabase.com/docs/latest/administration-guide/07-segments-and-metrics.

65. *X-rays* [online]. Metabase Documentation [cit. 2019-12-01]. Dostupné z: https://www.metabase.com/docs/latest/users-guide/14-x-rays.

66. MAURI, Michele; ELLI, Tommaso; CAVIGLIA, Giorgio; UBOLDI, Giorgio; AZZI, Matteo. RAWGraphs: A Visualisation Platform to Create Open Outputs. In: *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter*. Cagliari, Italy: ACM, 2017, 28:1–28:5. CHItaly '17. ISBN 978-1-4503-5237-6. Dostupné z DOI: 10.1145/3125571.3125585.

67. *Adding New Charts* [online]. RAWGraphs, 2017 [cit. 2019-11-26]. Dostupné z: https://github.com/rawgraphs/raw/wiki/Adding-New-Charts.

68. *Chart studio* [online]. Plotly [cit. 2019-12-01]. Dostupné z: https://plot.ly/online-chart-maker/.

69. *react-chart-editor* [online]. Plotly [cit. 2019-12-01]. Dostupné z: https://github.com/plotly/react-chart-editor.

70. *D3.js - Data-Driven Documents* [online]. Mike Bostok [cit. 2019-11-27]. Dostupné z: https://d3js.org/.

71. *C3.js | D3-based reusable chart library* [online]. C3.js [cit. 2019-12-01]. Dostupné z: https://c3js.org/.

72. *Content Management System for data visualisations* [online]. Datamatic [cit. 2019-12-04]. Dostupné z: https://datamatic.io/index.html.

73. LI, Deqing; MEI, Honghui; SHEN, Yi; SU, Shuang; ZHANG, Wenli; WANG, Junting; ZU, Ming; CHEN, Wei. ECharts: A declarative framework for rapid construction of web-based visualization. *Visual Informatics*. 2018, roč. 2, č. 2, s. 136–146. ISSN 2468-502X. Dostupné z DOI: https://doi.org/10.1016/j.visinf.2018.04.011.

74. *Examples - Apache Echarts* [online]. Echarts [cit. 2019-12-01]. Dostupné z: https://echarts.apache.org/examples/en/.

75. *Bring data science out of the lab and into the business* [online]. Plotly [cit. 2019-12-09]. Dostupné z: Bring%20data%20science%20out%20of%20the%20lab%20and%20into%20the%20business.

76. *Plotly Open Source Graphing libraries* [online]. Plotly [cit. 2019-12-01]. Dostupné z: https://plot.ly/graphing-libraries/.

77. *Introduction to dash* [online]. Plotly [cit. 2019-12-01]. Dostupné z: https://dash.plot.ly/introduction.

78. *AnyChart Features* [online]. AnyChart [cit. 2019-12-04]. Dostupné z: https://www.anychart.com/features/.

79. *See what ZingChart's 35+ built-in chart types & modules can do for your data visualization projects.* [online]. ZingChart [cit. 2019-12-01]. Dostupné z: https://www.zingchart.com/.

A Elektronické prílohy

V archíve sa nachádzajú nasledujúce súbory:

- Priečinok: Príklady získaných dát
  - sportis.xlsx - príklad dát získaných od organizácie Sportis
  - active-club.csv - príklad dát získaných od organizácie Active club

B DCAT - Základ pre implementáciu profilu

V tejto prílohe uvádzam príklad zdrojového kódu, vďaka ktorému je možné správne vytvoriť dcat profil kompatibilný s Národným katalógom. Tento zdrojový kód je určite možné ešte zefektívniť a vložiť do vlastného rozšírenia.

Prvá časť ukazuje, ako doplniť atribút downloadUrl.

```
if url and not access_url:
    if (not download_url) or (download_url and
       ↪ url != download_url):
        self._add_triple_from_dict(resource_dict
           ↪ , distribution, DCAT.accessURL, '
           ↪ url', _type=URIRef)
        self._add_triple_from_dict(resource_dict
           ↪ , distribution, DCAT.downloadURL, '
           ↪ url', _type=URIRef)
```

V druhej časti vytváram atribúty potrebné pre splnenie povinného atribútu `specifikace`.

Na začiatku súboru je potrebné vložiť nový menný priestor patriaci špecifikáciám a pridať jeho skratku do zoznamu menných priestorov.

```
PU = Namespace('https://data.gov.cz/slovník/podmínky-
   užití/')

namespaces = [
  \dots
  'pu': PU,
]
```

Potom je potrebné si vytvoriť novú lokálnu triedu specifikace (`specifikace_ref = BNode()`) a do nej pomocou metódy `self.a.add` pridať správnu špecifikáciu použitia.

```
specifikace_ref = BNode()
self.g.add((specifikace_ref, RDF.type, PU.
   ↪ Specifikace))
self.g.add((distribution, PU.specifikace,
   ↪ specifikace_ref))
```

```
specifikace = resource_dict.get('license_type')
if "neobsahuje-autorská-díla" in license_type:
    self.g.add((specifikace_ref, PU['autorské-
       dílo'], URIRef(resource_dict.get('
       ↪ license_link'))))

elif "není-autorskoprávně-chráněnou-databází" in
   ↪ license_type:
    self.g.add((specifikace_ref, PU['databáze-jako-
       autorské-dílo'], URIRef(resource_dict.get('
       ↪ license_link'))))

elif "není-chráněna-zvláštním-právem-pořizovatele-
   databáze" in license_type:
    self.g.add((specifikace_ref, PU['databáze-
       chráněná-zvláštními-právy'], URIRef(
       ↪ resource_dict.get('license_link'))))

elif "neobsahuje-osobní-údaje" in license_type:
    self.g.add((specifikace_ref, PU['osobní-
       údaje'], URIRef(resource_dict.get('
       ↪ license_link'))))
```

Keďže kľúče v slovníku PU používajú diakritiku, nastáva problém s kódovanim. Python interpreter využíva ASCII a znaky s diakritikou sú mimo tohto rozsahu. Je potrebné na začiatku odkázať interpreterovi, že je potrebné použiť kódovanie zdrojového kódu vo formáte utf8:

```
import sys
reload(sys)
sys.setdefaultencoding('utf8')
```