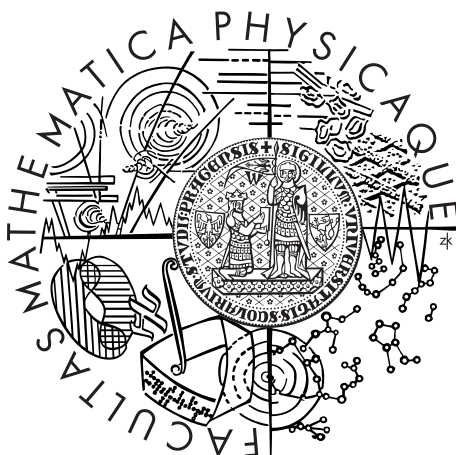


Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## DIPLOMOVÁ PRÁCE



Bc. Pavel Hryzlík

# Využití Linked Data pro sdílení dat o smlouvách veřejných institucí

Katedra softwarového inženýrství

Vedoucí diplomové práce: Doc. Mgr. Martin Nečaský, Ph.D.

Studijní program: Informatika

Studijní obor: I2 Softwarové systémy

Praha 2015

Zde bych rád poděkoval vedoucímu práce Doc. Mgr. Martin Nečaskému, Ph.D. za správné směřování, rady a nápady. Dále bych chtěl poděkovat Ondřeji Profantovi a také PhDr. Ing. Jiřímu Skuhrovcovi. Díky nim jsem pronikl do světa otevřených dat. Díky patří pochopitelně také mým nejbližším.

Prohlašuji, že jsem tuto diplomovou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne .....

Podpis autora

Název práce: Využití Linked Data pro sdílení dat o smlouvách veřejných institucí

Autor: Bc. Pavel Hryzlík

Katedra: Katedra softwarového inženýrství

Vedoucí diplomové práce: Mgr. Martin Nečaský, Ph.D., Katedra softwarového inženýrství

Abstrakt: Cílem diplomové práce je prozkoumat možnosti využití principů Linked Data pro publikaci a sdílení dat o smlouvách veřejných institucí a jejich propojení na související data ve veřejném prostoru (např. obchodní a živnostenský rejstřík, registr veřejných zakázek, apod.). Práce představí kompletní proces otevírání smluv. Definuje datový standard pro otevřené smlouvy a navrhne ontologii pro publikaci dat o smlouvách a jejich propojení. Dále navrhne a implementuje platformu pro publikaci smluv. První částí platformy je konverzní modul umožňující konverzi smluv uložených v relačních databázích do RDF podoby. Využije zde techniky R2RML mapování. Druhou částí je jednotné úložiště stahující údaje o smlouvách v Linked Data podobě. Třetí částí je webová aplikace, která data o smlouvách zpřístupní koncovým uživatelům.

Klíčová slova: Smlouva, Otevřená data, Linked Data, RDF, JSON-LD, R2RML, SPARQL

Title: Exploitation of Linked Data for sharing public agreements data

Author: Bc. Pavel Hryzlík

Department: Department of Software Engineering

Supervisor: Doc. Mgr. Martin Nečaský, Ph.D., Department of Software Engineering

Abstract: The objective of the thesis is to explore the possibilities of using Linked Data principles for publishing and sharing data on contracts of public institutions and their connections to related data in the public domain (eg. Business and trade register, register of contracts, etc.). Thesis presents the entire process of opening up contracts. Defines a data standard for open contracts and proposes an ontology for the publication of data on contracts and their interconnections. Furthermore, designs and implements a platform for publishing contracts. The first part of the platform is a conversion module enabling the conversion of contracts stored in relational databases into RDF form. Employed are R2RML mapping techniques. The second part is a uniform repository that downloads data on contracts in Linked Data format. The third part is a web application that will make the data on contracts available to end users.

Keywords: Contract, Open Data, Linked Data, RDF, JSON-LD, R2RML, SPARQL

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Motivace . . . . .	4
1.2	Cíl práce . . . . .	4
1.3	Struktura práce . . . . .	5
<b>2</b>	<b>Otevřená data a principy Linked Data</b>	<b>6</b>
2.1	Otevřená data (Open Data) . . . . .	6
2.2	Kvalita otevřených dat . . . . .	8
2.3	Stupně otevřenosti[7] . . . . .	8
2.4	Propojitelná data (Linked Data) . . . . .	10
2.5	Otevřená a propojitelná data (Linked Open Data - LOD) . . . . .	11
2.6	Výhody a přínosy otevřených dat a principů Linked data . . . . .	12
2.7	RDF (Resource Description Framework) . . . . .	13
2.8	RDF Ontologie . . . . .	14
2.8.1	Propojování se souvisejícími entitami . . . . .	16
2.9	Publikace . . . . .	17
2.9.1	Příklad dat serializovaných ve formátu N-Triples . . . . .	17
2.9.2	Příklad dat serializovaných ve formátu Turtle . . . . .	18
2.9.3	Příklad dat serializovaných ve formátu JSON-LD . . . . .	19
<b>3</b>	<b>Otevřené smlouvy</b>	<b>22</b>
3.1	Situace ve veřejné správě ČR . . . . .	22
3.2	Standard pro zveřejňování smluv . . . . .	23
3.2.1	Základní struktura . . . . .	23
3.2.2	Reprezentované entity . . . . .	25
3.2.3	Číselníky . . . . .	30
3.3	Publikace . . . . .	32
3.3.1	JSON . . . . .	32
3.3.2	CSV . . . . .	36
3.4	Metodika zveřejňování smluv . . . . .	37
<b>4</b>	<b>Otevřené smlouvy jako Linked Data</b>	<b>38</b>
4.1	Přiřazení identifikátorů jednotlivým entitám otevřených smluv . . . . .	38
4.2	Ontologie pro publikaci dat o smlouvách . . . . .	39
4.2.1	Analýza vhodných, již existujících ontologií . . . . .	40
4.2.2	Tvorba ontologie . . . . .	42
4.2.3	Publikace . . . . .	47
4.3	Možnosti propojení na související data . . . . .	47
4.4	Provázání s datovým formátem JSON . . . . .	48
<b>5</b>	<b>Požadavky na platformu pro otevřené smlouvy</b>	<b>54</b>
5.1	Funkční požadavky . . . . .	54
5.2	Nefunkční požadavky . . . . .	55

<b>6</b>	<b>Návrh platformy pro otevřené smlouvy</b>	<b>56</b>
6.1	Architektura . . . . .	56
6.1.1	Konverzní mechanismus . . . . .	58
6.1.2	Jednotné úložiště . . . . .	59
6.1.3	Propojená datová síť . . . . .	60
6.1.4	Webová aplikace . . . . .	62
<b>7</b>	<b>Implementace platformy</b>	<b>64</b>
7.1	Konverzní mechanismus . . . . .	64
7.1.1	Munis ESML . . . . .	64
7.1.2	R2RML mapování . . . . .	66
7.1.3	Volba R2RML procesoru . . . . .	73
7.1.4	Volba technologií a implementační platformy . . . . .	74
7.1.5	Napojení na datové úložiště . . . . .	74
7.1.6	SPARQL endpoint . . . . .	74
7.1.7	Zpracování RDF výstupu . . . . .	74
7.2	Jednotné úložiště . . . . .	76
7.2.1	Nástroj Unified views . . . . .	76
7.3	Webová aplikace . . . . .	77
7.3.1	Volba technologií a implementační platformy . . . . .	77
7.3.2	Získávání dat . . . . .	77
<b>8</b>	<b>Evaluace</b>	<b>83</b>
<b>9</b>	<b>Linked Data v procesu otevírání smluv</b>	<b>85</b>
	<b>Závěr</b>	<b>87</b>
	<b>Seznam použité literatury</b>	<b>89</b>
	<b>Seznam obrázků</b>	<b>90</b>
	<b>Seznam tabulek</b>	<b>91</b>
	<b>Výpisy kódu</b>	<b>92</b>
	<b>Seznam použitých zkratk</b>	<b>93</b>
	<b>Přílohy</b>	<b>94</b>
	<b>A Příloha</b>	<b>95</b>
	<b>B Příloha</b>	<b>96</b>
	<b>C Příloha</b>	<b>97</b>
	<b>D Příloha</b>	<b>105</b>

# 1. Úvod

V době informační společnosti se využívání internetu stalo naší každodenní rutinou. Skrze různé webové aplikace a služby každodenně pracujeme s obrovským množstvím informací. Běžně komunikujeme přes e-mail, finance spravujeme skrze internetové bankovníctví, část svého osobního života sdílíme na sociálních sítích. Požadavek na on-line vyřizování agendy vůči veřejné správě tedy není překvapujícím.

Problematika elektronizace veřejné správy, jednotně nazývaná jako „e-government“, je aktuálním tématem již po mnoho let. Důsledkem tohoto procesu je generování obrovského množství nesmírně důležitých dat. Tato data ale v naprosté většině případů leží schovaná v databázích jednotlivých veřejných institucí. Mnoho z těchto dat by ale ze zákona mělo být volně dostupných. Často však jediným možným způsobem, jak taková data získat je použití zákona č.106/1999 Sb.<sup>1</sup>, o svobodném přístupu k informacím. Netřeba zmiňovat, že tato snaha se mnohdy může stát značně netriviální.

Řešením je vhodná data, resp. metadata o těchto datech, zpřístupnit on-line. Pro strojově čitelná data zveřejněná na internetu se zažil pojem Otevřená data. Tato data pak může vyhledávat a zpracovávat kdokoli. To přináší řadu dílčích výhod od úspory nákladů, přes boj s korupcí, až po zapojení občanů, nemluvě o podnikatelském potenciálu, převážně možnosti vzniku mnoha užitečných aplikací pracujících nad otevřenými daty. To celé za cenu minimálních nákladů z veřejných rozpočtů.

Otvírání dat můžeme chápat jako další krok v procesu elektronizace veřejné správy. Průkopníky v této oblasti jsou státy s vyspělou formou demokracie, jako USA a Spojené království. Příklad si ale také můžeme vzít od Estonska. Malá země, vědoma si, že nemá nerostné bohatství ani rozvinutý průmysl, se rozhodla prosadit na poli informačních technologií, kde základem jsou otevřené on-line služby veřejné správy. Důležitost otevřených dat si uvědomuje i Evropská unie. Směrnicí 2013/37/EU<sup>2</sup> v podstatě doporučuje členským státům, aby data otvíraly. České republice se také povedlo nastartovat procesy otvírání veřejné správy. Pokrok je cítit hlavně na národní úrovni. Mezi městy a obcemi jsou však otevřená data často stále neznámým pojmem. Problematickou a obecně osvětou otevřených dat se zabývá mimo jiné Ministerstvo vnitra ČR[1], projekt Rekonstrukce státu[2], Fond Otakara Motejla[3], Oživení o.s.[4] či iniciativa OpenData.cz[5].

Otevřená data však nelze chápat jako samospásné řešení problémů veřejné správy. Jsou spíše prostředkem ke zvýšení otevřenosti a transparentnosti. Veřejná služba však může být netransparentní i s otevřenými daty. Řekněme, že pro kvalitní veřejnou službu jsou otevřená data nutnou, nikoli však postačující podmínkou.

Dalším aspektem otevřených dat je jejich kvalita. Kvalitní otevřená data jsou propojena mezi sebou v rámci jednotného sdíleného prostoru, mohou na sebe odkazovat a využívat širokého kontextu, které takový sdílený prostor propojených dat nabízí. Taková data využívají principů Linked Data.

---

<sup>1</sup>Zákon č.106/1999 Sb. - <http://www.zakonyprolidi.cz/cs/1999-106>

<sup>2</sup>Směrnice 2013/37/EU - <http://www.eurlex.cz/dokument.aspx?celex=32013L0037>

## 1.1 Motivace

Základní motivaci pro vznik této práce bych rozdělil do tří pilířů:

### Veřejnoprávní sféra

Na podzim roku 2014 se konal seminář Transparentnost v obcích v Poslanecké sněmovně pořádaný panem Mgr. Janem Farským. V rámci semináře se sešla skupina složená ze zástupců měst a obcí, akademické sféry a neziskového sektoru. Předmětem jednání byla otevřená data. Výsledkem bylo rozhodnutí, že první datovou sadou vhodnou k plošnému otevření, také vzhledem k chystanému zákonu o registru smluv, jsou údaje o smlouvách. Prvním krokem je standardizace datového formátu, resp. určení položek vhodných ke zveřejnění. Motivací bylo, že pokud standard začne využívat netriviální počet měst a obcí, tak je reálná šance k prosazení standardu na národní úroveň. Ustanovila se tedy, pod záštitou Oživení o.s. a Centra aplikované ekonomie o.s.<sup>3</sup>, „akční“ skupina, jejímž cílem byla tvorba datového standardu pro otevřené smlouvy. Bylo mi ctí, se stát členem této skupiny.

### Komerční sféra

Jako externista se podílím na tvorbě software pro veřejnou správu ve společnosti Triada spol. s r.o. Mým úkolem se ke konci roku 2014 stala tvorba modulu ESML pro interní evidování smluv.

### Akademická sféra

V rámci MFF UK ve spolupráci s Fakultou informatiky VŠE vznikla iniciativa OpenData.cz. Jejím cílem je vybudování otevřené datové infrastruktury v České republice. Na MFF UK také probíhá výzkum propojitelných dat, Linked Data. Mým cílem bylo přispět k otevřené datové infrastruktuře, navíc s využitím principů Linked Data. Rozhodnutí věnovat se publikaci dat o smlouvách padlo již v červnu 2014. Konkrétní obrysy však práce získala až s přispěním výše zmíněných pilířů.

Výsledkem je tedy aplikace principů Linked Data pro publikaci a sdílení dat o smlouvách s možností konkrétního využití nad modulem ESML společnosti Triada. To celé s ohledem na vznikající datový standard. Jednou z důležitých motivací bylo, že v případě prosazení datového standardu na národní úroveň mohou města a obce používat modul ESML využitím této práce automaticky zveřejňovat smlouvy v Linked Data podobě, a to s minimálními náklady. Taková data lze pak agregovat do jednotných úložišť, nad kterými mohou vznikat nejrůznější aplikace přinášející konečný přínos pro uživatele.

## 1.2 Cíl práce

Cílem práce je prozkoumat možnosti využití principů Linked Data pro publikaci a sdílení dat o smlouvách veřejných institucí a jejich propojení na související

---

<sup>3</sup>Centrum aplikované ekonomie o.s. - <http://cae.zindex.cz/>



data ve veřejném prostoru. Prvním krokem je definování datového standardu a ontologie pro otevřené smlouvy. Dalším krokem je návrh způsobu konverze dat stávajícími informačními systémy veřejných institucí (v podobě relačních databází) do otevřeného formátu využívající principy Linked Data a implementace konverzního mechanismu pro vybraný konkrétní informační systém (Triada spol. s.r.o). V dalším kroku následuje návrh a implementace jednotného úložiště dat o smlouvách v Linked Data s experimentálním zprovozněním na serveru poskytnutém vedoucím práce. V jednotném úložišti se očekává návrh řešení integračních problémů dané heterogenitou dat publikovaných různými veřejnými institucemi. Následujícím krokem je nad tímto jednotným úložištěm návrh a implementace webové aplikace, která data o smlouvách zpřístupní koncovým uživatelům.

## 1.3 Struktura práce

Obsah práce je rozdělen na 9 kapitol a 3 přílohy. Ve druhé kapitole jsou popsány a vysvětleny základní principy otevřených dat. Třetí kapitola se zabývá pojmem otevřené smlouvy. Kapitola nejdříve rozebere aktuální stav otevřenosti smluv ve veřejné správě a následně nastíní vznikající datový standard. Čtvrtá kapitola zadefinuje otevřené smlouvy jako Linked Data. V páté kapitole se analyzují požadavky na platformu pro otevřené smlouvy. Šestá kapitola zmíněnou platformu navrhne. Sedmá kapitola se zabývá konkrétní implementací platformy. V osmé kapitole jsou znázorněny zátěžové testy některých dílčích částí implementace. Poslední, devátou kapitolou je závěr shrnující práci jako celek. Nedílnou součástí práce je seznam použité literatury a slovníček pojmů. Práce zahrnuje také 3 přílohy. V příloze A je znázorněn harmonogram vývoje standardu otevřených smluv. V příloze B se nachází uživatelská dokumentace. Konečně, příloha C popisuje strukturu přiloženého datového nosiče.TODO

## 2. Otevřená data a principy Linked Data

Předmětem této kapitoly je čtenáře stručně seznámit se základními pojmy a principy otevřených, propojitelných dat a následně s technologiemi sloužícími k jejich zápisu a zpracování.

### 2.1 Otevřená data (Open Data)

*„Open data can help us address the greatest challenges of our time and generate value for everyone“* - Open Data Institute 2012

Začneme definicí, kterou si postupně vysvětlíme. Jako otevřená data můžeme chápat údaje zveřejněná na internetu, která jsou[6]

1. úplná
2. snadno dostupná
3. strojově čitelná
4. používající standardy s volně dostupnou specifikací
5. zpřístupněna za jasně definovaných podmínek užití dat s minimem omezení
6. dostupná uživatelům při vynaložení minima možných nákladů

#### Úplnost

Pokud se rozhodneme zveřejňovat data, tak v případě, že nás neomezuje zákon, či jiná restriktivní opatření, měli bychom dbát na to, aby byla úplná, resp. v maximálním možné rozsahu. Není cílem zveřejňovat útržky ztrácející vypovídající hodnotu.

#### Snadná dostupnost

Základní požadavek na dostupnost otevřených dat spočívá v tom, že by měla být k dispozici kdykoli, ne pouze např. na vyžádání. Otevřená data budou také přínosem pro širokou veřejnost jedině tehdy, pokud budou snadno dohledatelná. Skrytá data za změtí odkazů se hledají špatně.

#### Strojová čitelnost

Klíčovou vlastností otevřených dat je strojová čitelnost. Otevřeným datům by měl porozumět nejen člověk, ale i stroj. Účelem je umožnit data automatizovaně zpracovávat, analyzovat, počítat statistiky apod.

## Otevřené standardy

Software, nástroje či metodiky potřebné k zpracování dat by měly být volně dostupné. Data v uzavřeném formátu, která potřebují ke zpracování konkrétní proprietární software, postrádají smysl otevřenosti.

## Zpřístupněna za jasně definovaných podmínek

Typicky je třeba dbát na to, aby data byla zveřejňována pod otevřenou licencí.<sup>1</sup>

## Dostupná uživatelům s minimem nákladů

Je třeba si uvědomit, že nezveřejňujeme data pro data. Zveřejňujeme pro přídanou hodnotu, např. pro lepší službu nebo vyšší efektivitu. Náklady na zveřejnění by tak neměly přesáhnout případná zlepšení.



Obrázek 2.1: Logo otevřených dat

---

<sup>1</sup>Více k problematice licencování a užití otevřených dat lze dohledat na webu Ministerstva vnitra - <http://www.mvcr.cz/clanek/otevrena-data.aspx?q=Y2hudW09Mg%3d%3d>

## 2.2 Kvalita otevřených dat

Tvůrce WWW a ředitel konsorcia W3C<sup>2</sup> Tim Berners-Lee<sup>3</sup> navrhl pěti hvězdičkový systém, jak kategorizovat otevřená data (viz Obrázek 2.2). Každá hvězdička definuje stupeň otevřenosti, kde 5★ znamená nejvyšší kvalitu dat, 1★ naopak nejmenší. Také platí, že každý stupeň je nadmnožinou (rozšířením) stupně předešlého.

## 2.3 Stupně otevřenosti[7]

### ★ Libovolná zveřejněná data pod otevřenou licencí

- Přínosy pro uživatele - uživatel může data číst, tisknout, ukládat, přenášet, měnit a sdílet podle svého uvážení
- Přínosy/náklady pro vydavatele - velmi nenáročné na publikaci
- Příkladem může být formát PDF

Publikace dat na úrovni 1★ je zdaleka nejjednodušší a nepotřebuje příliš vynaloženého úsilí. Určitě je lepší zveřejňovat data na úrovni 1★, než vůbec. Využitelnost dat však může být velmi obtížná, např. díky nutnosti dolování dat z PDF dokumentů.

### ★★ Strukturovaná data ve strojově čitelném formátu

- Přínosy pro uživatele - uživatel může pokročile zpracovávat data s využitím proprietárních nástrojů k tomu určených
- Přínosy/náklady pro vydavatele - velmi nenáročné na publikaci
- Příkladem může být formát MS Excel (.xls)

V dnešní době už poměrně rozšířený způsob publikace dat. Zpracování dat ale vyžaduje specifické nástroje k tomu určené. Pokud tedy chceme zpracovávat např. excelovskou tabulku (.xls), potřebujeme k tomu komerční produkt MS Excel<sup>4</sup>.

### ★★★ Formát dat je otevřený

- Přínosy pro uživatele - uživatel při zpracování dat není omezen žádným specifickým nástrojem
- Přínosy/náklady pro vydavatele - nenáročné na publikaci, může však vyžadovat transformaci dat, např. z uzavřeného formátu
- Příkladem může být formát CSV

---

<sup>2</sup>W3C - <http://www.w3.org/>

<sup>3</sup>Tim Berners-Lee - životopis - <http://www.w3.org/People/Berners-Lee/>

<sup>4</sup>Toto se netýká formátu .xlsx. Ten již vychází z otevřené specifikace Office Open XML - ata publikovaná v .xlsx formátu tedy můžeme chápat jako 3★.

Teprve v této kategorii se můžeme bavit o “opravdových” otevřených datech. Resp. data musejí mít stupeň otevřenosti minimálně 3★ , aby naplnila základní definici otevřených dat uvedenou výše.

### ★★★★ Jednotlivé objekty jsou identifikovány pomocí URI

- Přínosy pro uživatele - uživatel se může na data odkazovat, odkazy si ukládat, případně data snadno kombinovat s jinými (na stejném, nebo vyšším stupni)
- Přínosy/náklady pro vydavatele - náročnější na publikaci
- Příkladem může být formát RDF

Důležité je dbát na to, aby URI nebylo virtuální, resp. aby se po dotázání uživateli vrátil požadovaný obsah. V prostředí WWW je zajištění obsahu typicky praktikováno skrze protokol HTTP.

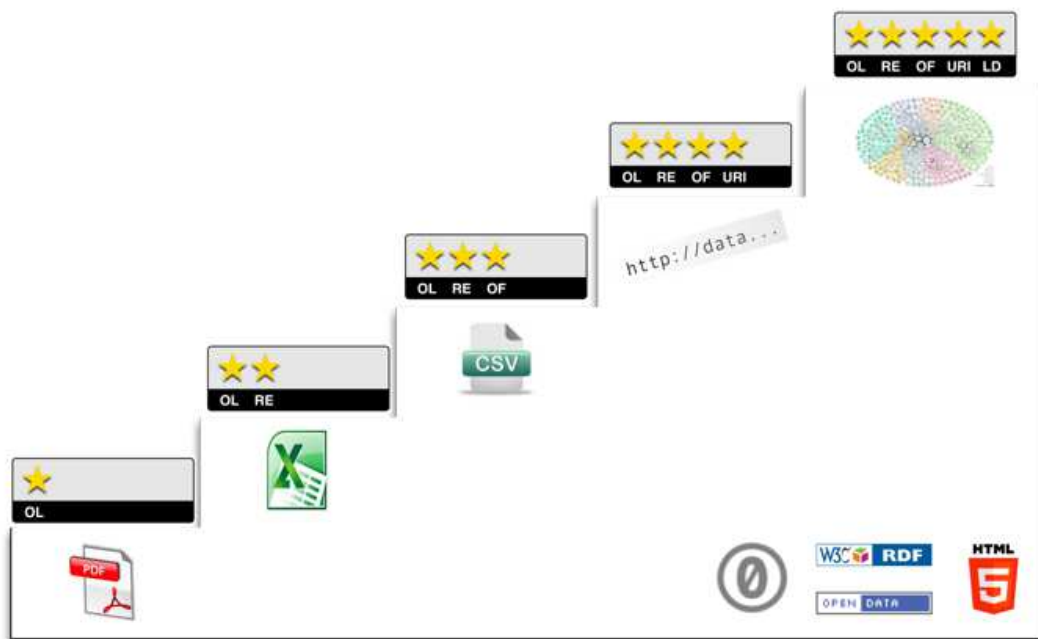
Díky URI identifikaci můžeme data reprezentovat jako orientovaný graf propojených objektů, které na sebe mohou vzájemně odkazovat. K popisu takovýchto dat se používá formát RDF.[[ODKAZ NA KAPITOLU](#)]

V prostředí České republiky považováno jako nadstandard.

### ★★★★★ Data jsou propojena se souvisejícími daty

1. Přínosy pro uživatele - vytvoření efektu datové sítě, větší informační hodnota dat
2. Přínosy/náklady pro vydavatele - náročnější na publikaci
3. Příkladem může být formát RDF

V této nejvyšší kategorii se data mohou stát součástí datové sítě propojených grafů.



Obrázek 2.2: Stupně otevřenosti dat, zdroj: [7]

## 2.4 Propojitelná data (Linked Data)

Linked Data vychází z myšlenky webu aplikované na data. Webu rozumíme jako síti propojených webových stránek. Cílem Linked Data je mít síť propojených, strojově čitelných dat, resp. stavební kámen sémantického webu<sup>5</sup>. Jedná se v podstatě o další krok v evolučním vývoji webu jako takového.

Podle [8] definujeme základní principy Linked Data jako:

1. Každá entita je identifikována pomocí HTTP URI<sup>6</sup>
2. HTTP URI by mělo být vyhledatelné v síti WWW a umožňovat k němu přistupovat a odkazovat se na něj
3. Po přistoupení na HTTP URI entity mají být poskytnuty relevantní informace o dané entitě ve standardizovaném formátu či prostřednictvím API<sup>7</sup>
4. Data k entitám rozšířit o HTTP URI odkazy na další související entity<sup>8</sup>

Jak je vidět, Linked Data naplňují všechny požadavky na 5★ kvalitu dat s jednou výjimkou. Linked Data nemusejí být z podstaty otevřenými daty. Určitě si dovedeme představit mnoho scénářů, kdy je přínosem mít propojená, ale privátní data. Typickým příkladem můžou být korporátní intranetové informační systémy.

<sup>5</sup>Sémantický web - <http://www.cs.umd.edu/~golbeck/LBSC690/SemanticWeb.html>

<sup>6</sup>V prostředí webu to bude typicky hypertextový odkaz ve formě URL[ZDROJ]

<sup>7</sup>Pro popis Linked Data se typicky používá jazyk RDF - <http://www.w3.org/RDF/>, k dotazování k datovému API - SPARQL - <http://www.w3.org/TR/rdf-sparql-query/>

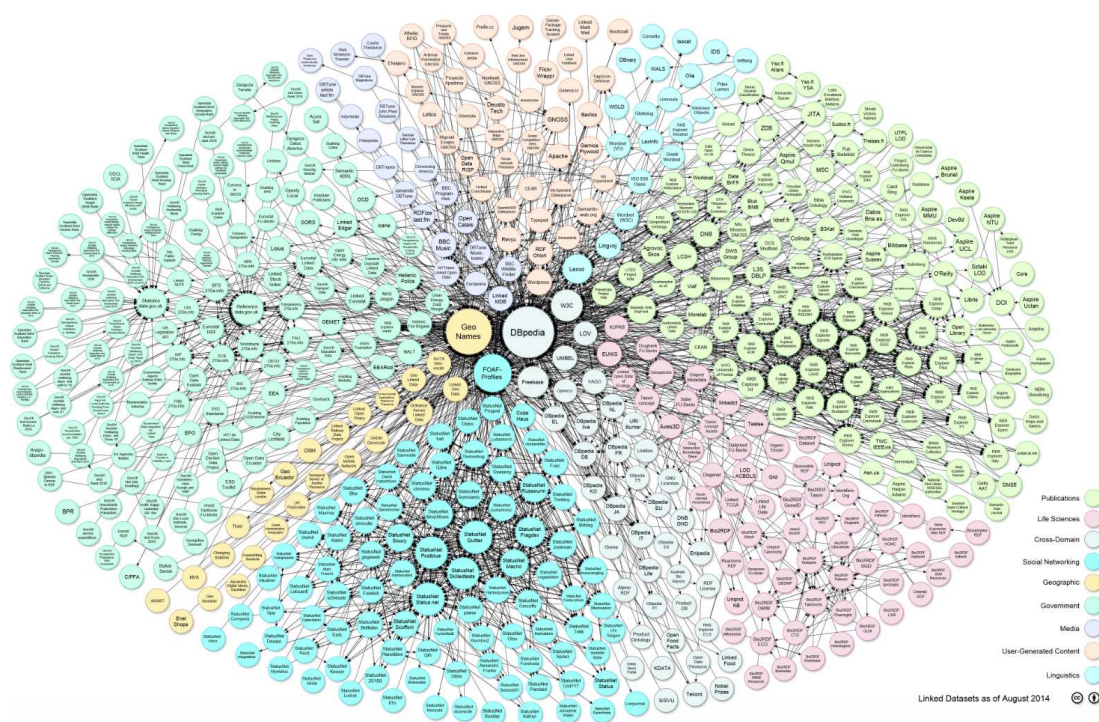
<sup>8</sup>o nám zaručí, že můžeme procházet jednotlivé entity podobným způsobem jako webové stránky v rámci sítě WWW.

## 2.5 Otevřená a propojitelná data (Linked Open Data - LOD)

Otevřená data na úrovni 5★ kvality můžeme tedy chápat jako Linked Open Data. Taková data se mohou stát součástí globálního prostoru sdílených, propojených dat. Připojením datové sady tak můžeme čerpat informační potenciál celého prostoru<sup>9</sup>.

Takový prostor s časem neustále roste. Využití lze nalézt ve většině oblastí lidského konání. Od sdílení a obohacování vědeckých dat, např. biologických, chemických struktur a reakcí s cílem objevů nových postupů v medicíně, přes zpracování dat jednotlivých veřejných správ za účelem kvalitnější veřejné služby až po obohacování kontextu nejružnějšího mediálního obsahu.

Na obr. 2.3 vidíme příklad vizualizace otevřených a propojených (LOD) dat nazývaný Linked Open Data Cloud. Jedná se o datasety obsahující alespoň 1000 trojic (více v kapitole o RDF) a alespoň 50 odkazů na jiná data ve sdíleném prostoru.



Obrázek 2.3: Linked Open Data Cloud, Srpen 2014, zdroj: <http://lod-cloud.net/>

<sup>9</sup>Tvůrci grafu procházejí web a do cloudu přidávají dostupné datasety splňující podmínky Linked Data a podmínky na počet trojic a odkazů. Nezkontrolují ale licence jednotlivých datasetů. Některé datasety proto mohou být chráněny specifickými právy.

## **2.6 Výhody a přínosy otevřených dat a principů Linked data**

### **Obecné výhody otevřených dat**

1. Zapojení uživatelů - kontrola, návrhy ke zlepšení dat
2. Zvýšení transparentnosti vydavatele dat, boj s korupcí
3. Kvalitnější veřejná služba
4. Zvýšení efektivity, úspora nákladů
5. Široké možnosti dalšího využití - analýzy, statistiky, vizualizace

### **Výhody principů Linked Data**

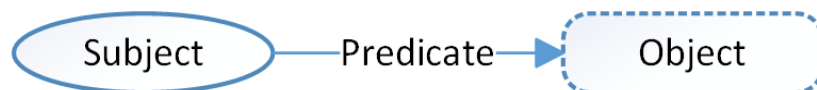
1. Sdílená, rozšiřitelná a snadno znovu použitelná data
2. Data jsou začleněna do kontextu, resp. lze se odkazovat přímo na data
3. Data jsou propojena s dalšími relevantními daty, informační hodnota dat je tedy tím větší, čím více mají vazeb
4. Standardizované formáty pro publikaci



## 2.7 RDF (Resource Description Framework)

Formát RDF byl vyvinut za účelem snadného strojového zpracování a propojování dat. Jedná se o čistě abstraktní formát udávající, jak data popisovat. Nezabývá se tedy konkrétní podobou výsledných dat.

Základním stavebním kamenem RDF je tvrzení, resp. trojice: **Subjekt - Predikát - Objekt** (viz Obrázek 2.4). Subjektem je míněn zdroj, který popisujeme. Predikát je vlastnost, která o objektu něco tvrdí. Objekt je hodnota dané vlastnosti. Jednotlivé trojice mohou na sebe navazovat a vytvořit tak orientovaný graf.



Obrázek 2.4: Základní RDF trojice

### Nyní definujeme několik pravidel a doporučení pro popisování dat v RDF

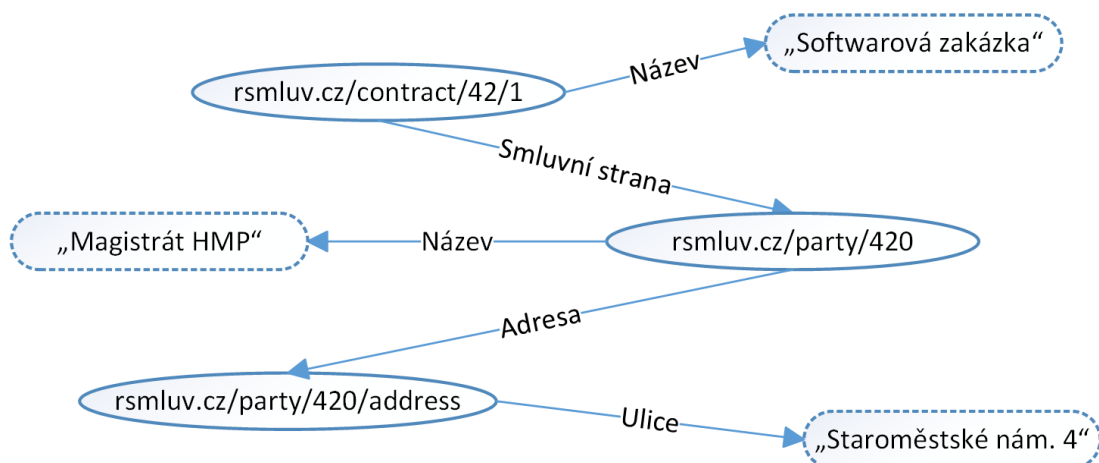
1. Každý subjekt je jednoznačně identifikován pomocí URI, nebo je označen jako anonymní<sup>10</sup>
2. Objektem je buď hodnota (literál), odkaz na subjekt (resource), nebo je označen jako anonymní
3. Pro každý subjekt je specifikován jeho typ (třída) formou URI
4. Každému predikátu je přiřazen také jeho typ formou URI
5. Jednotlivé URI z bodů 3,4 by měly odkazovat na konkrétní slovníky tříd a predikátů, resp. ontologie

Na obr. 2.5 vidíme příklad jednoduchého grafu ve formátu RDF (aplikována pravidla 1 a 2). Popisuje 3 subjekty a přiřazuje jim konkrétní vlastnosti. Vidíme, že každý subjekt je identifikován vlastním URI. Díky tomu mohou subjekty na sebe odkazovat. Jednotlivé trojice by pak vypadaly takto:

1. *http://rsmluv.cz/contract/42/1 - Název - Softwarová zakázka*
2. *http://rsmluv.cz/contract/42/1 - Smluví strana - rsmluv.cz/party/420*
3. *http://rsmluv.cz/party/420 - Název - Magistrát HMP*
4. *http://rsmluv.cz/party/420 - Adresa - rsmluv.cz/party/420/address*
5. *http://rsmluv.cz/party/420/address - Ulice - Staroměstské nám. 4*

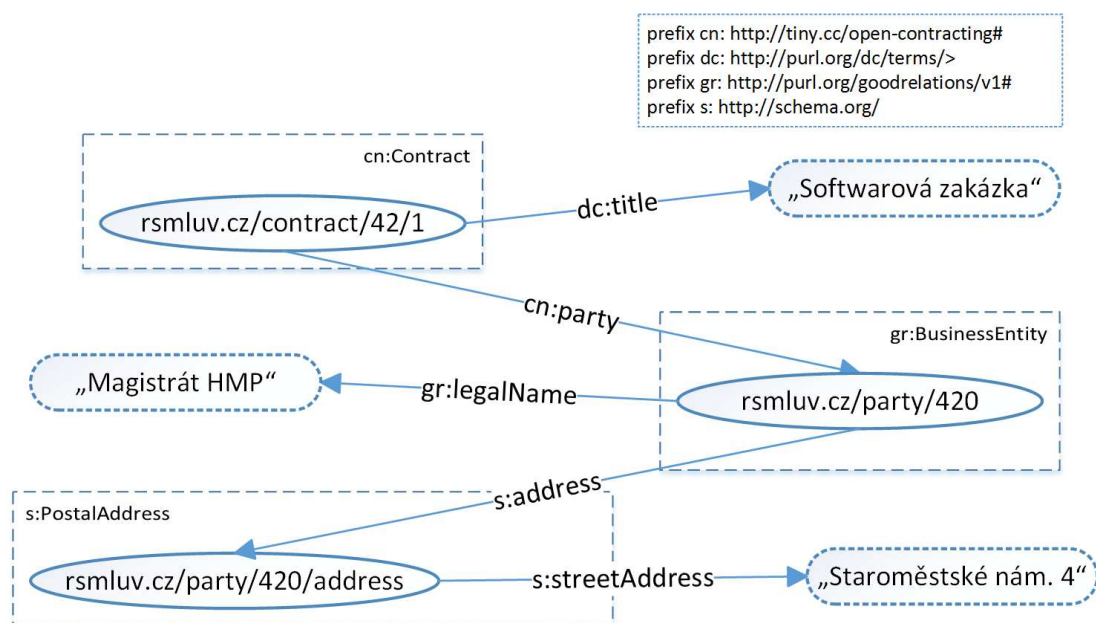
---

<sup>10</sup>Subjekty, příp. objekty lze označit jako anonymní, resp. Blank node. Na anonymní subjekty, resp. objekty ale nelze přímo přistupovat. Používají se typicky k zapouzdření, či jako kontejnery jiných objektů.



Obrázek 2.5: Jednoduchý RDF graf

Ze zmíněného příkladu ale není zřejmý význam, resp. sémantika jednotlivých subjektů a predikátů. Je tedy důležité jim přiřadit konkrétní typy. Každý typ by měl být popsán v konkrétním slovníku tříd a predikátů. Takovéto slovníky nazýváme ontologiemi. Na obr. 2.6 vidíme zmíněný příklad rozšířený o přiřazené typy (aplikována pravidla 3, 4, 5)<sup>11</sup>.



Obrázek 2.6: RDF graf s přiřazenými typy

## 2.8 RDF Ontologie

Pod pojmem ontologie si můžeme představit sadu termínů popisujících určitou věcnou oblast. V případě popisování RDF dat definujeme slovník tříd a vlastností (predikátů), které mohou uživatelé ve svých datech používat.

<sup>11</sup>Pro zapisování typů se kvůli úspornosti používají prefixy definované typicky na začátku dokumentu.

Konkrétní ontologii nelze chápat jako striktně vyžadovaný standard, ale spíše jako sadu doporučení. Buď využijeme k popisu dat nějakou z řady již existujících ontologií, nebo můžeme vytvořit ontologii vlastní. Přesto ale chceme, aby se již existující ontologie používaly co nejvíce. Přínosem je hlavně to, že aplikace a nástroje implementované nad známými ontologiemi budou schopné automaticky rozpoznat naše data<sup>12</sup>.

Základními jazyky pro modelování RDF dat jsou Web Ontology Language (OWL)[9] a RDF Schema (RDFS)[10]. Konkrétní specifikace se provádí opět ve formátu RDF a je publikována pod vlastním URI.

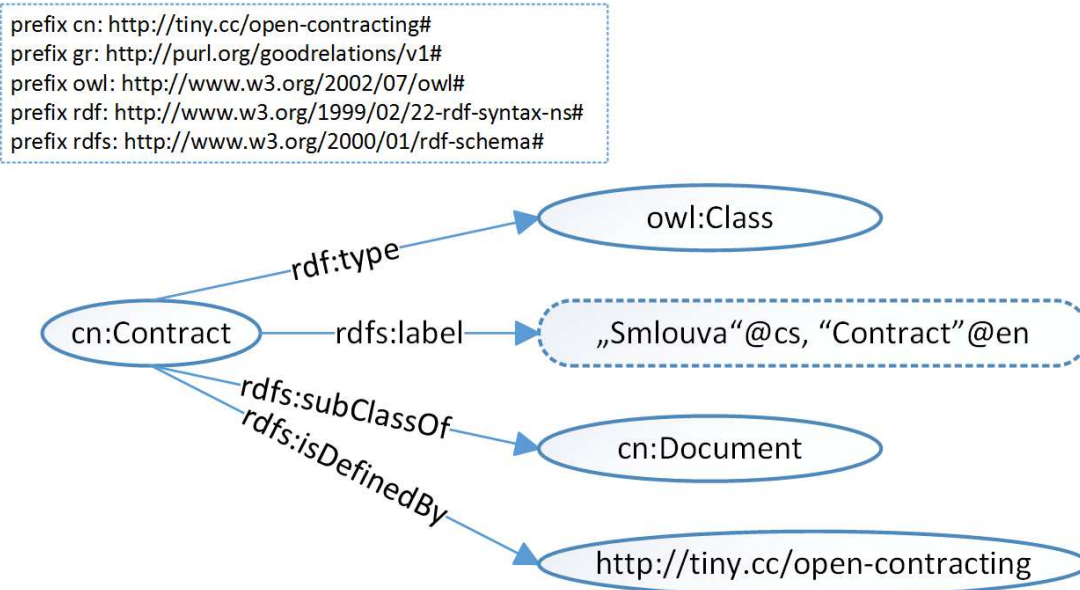
Mezi základní výrazové prostředky jazyka OWL a RDFS patří:

- *owl:Class* - typ entity třída
- *owl:ObjectProperty* - typ entity vlastnost
- *owl:FunctionalProperty* - typ jedinečná vlastnost (nemůže se opakovat)
- *owl:unionOf* - jeden typ třídy z výčtu musí být vyplněn
- *owl:equivalentClass* - definuje, že se jedná o třídu odpovídající jiné třídě
- *owl:equivalentProperty* - definuje, že se jedná o vlastnost odpovídající jiné vlastnosti
- *rdfs:label* - popis třídy/vlastnosti
- *rdfs:comment* - komentář třídy/vlastnosti
- *rdfs:domain* - požadovaný typ domény třídy/vlastnosti
- *rdfs:range* - požadovaný rozsah typů třídy/vlastnosti
- *rdfs:isDefinedBy* - definice zdroje třídy/vlastnosti
- *rdfs:subClassOf* - definice, že se jedná o podtřídu určité třídy
- *rdfs:subPropertyOf* - definice, že se jedná o podvlastnost určité vlastnosti

Na obr. 2.7 vidíme příklad ontologie třídy Contract. Ontologie nám říká, že se jedná o třídu (typ owl:Class) s názvem Smlouva (rdfs:label), která je podtřídou (rdfs:subClassOf) Document a je definovaná (rdfs:isDefinedBy) v ontologii <http://tiny.cc/open-contracting>. Kdokoli pak bude zpracovávat entitu označenou tímto typem, tak díky přiřazené ontologii bude schopen určit, že se jedná o smlouvu.

---

<sup>12</sup>Mezi všeobecně známé ontologie patří např. DublinCore - <http://purl.org/dc/terms/>, Friend-of-a-Friend - <http://xmlns.com/foaf/0.1/> nebo Schema - <http://schema.org/>. Existuje také katalog ontologií - <http://lov.okfn.org/dataset/lov/>



Obrázek 2.7: Ontologie třídy *Contract*

### 2.8.1 Propojování se souvisejícími entitami

Díky RDF můžeme data reprezentovat jako orientovaný graf. Otázka tedy zní, zdali lze propojovat grafy mezi sebou. Ve formátu RDF je to velmi jednoduché. Jako objekt predikátu stačí položit subjekt z jiného grafu. Díky URI identifikaci entit tedy není rozdílem, zdali je cílovým subjektem entita v lokálních datech, nebo entita cizí.

V rámci propojování dat s jinými datasety však není neobvyklé, že stejné entity jsou reprezentované v různých datasetech pod vlastními URI. Je tedy třeba vyjádřit, že se jedná o data reprezentující stejné entity. V jazyku OWL za tímto účelem existuje predikát `sameAs`, kterým můžeme definovat odpovídající si entity (viz Obrázek 2.8).



Obrázek 2.8: Možnost propojení dat

## 2.9 Publikace

V minulých kapitolách bylo řečeno, jak popisovat data pomocí RDF. Jednalo se o sémantický popis. Pokud však data chceme publikovat, je třeba konkrétního datového formátu, který definuje syntaxi, resp. jak RDF data serializovat. Takových formátů existuje celá řada, např.:

- **N-Triple[?]** - nejjednodušší serializace RDF grafu v podobě výčtu trojic
- **N-Quads[12]** - rozšíření pro N-Triples s možností zaznamenat více grafů
- **RDF/XML[13]** - RDF graf serializovaný do XML, využívající prefixového zápisu
- **Turtle[14]** - úsporný textový formát s možností komprese trojic, využívající prefixových zápisů
- **Trig[15]** - rozšíření Turtle pro použití nad více grafy
- **RDFa[16]** - serializace RDF do (X)HTML[17] dokumentů, využívající prefixového zápisu
- **JSON-LD[18]** - specifický zápis RDF grafu, využívající mapování položek JSON dokumentu na RDF ontologie

Pro potřeby této práce si vystačíme s formáty N-Triples, Turtle a JSON-LD. Vysvětlíme si je na příkladech. Jako data k serializaci použijeme příklad z obr. 2.6.

### 2.9.1 Příklad dat serializovaných ve formátu N-Triples

Serializace RDF dat do N-Triples je velmi jednoduchá. Jedná se o seznam trojic oddělených tečkou. Každá trojice je uvedena na vlastním řádku. Tento formát nepoužívá prefixové zkracování URI. Je vhodný pro proudové zpracování velkého množství dat (viz Obrázek 2.1)<sup>13</sup>.

```
1 <http://rsmluv.cz/contract/42/1>
2   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
3   <http://tiny.cc/open-contracting#Contract> .
4 <http://rsmluv.cz/contract/42/1>
5   <http://purl.org/dc/terms/title>
6   "Softwarová zakázka" .
7 <http://rsmluv.cz/contract/42/1>
8   <http://tiny.cc/open-contracting#party>
9   <http://rsmluv.cz/party/420> .
10
11 <http://rsmluv.cz/party/420>
12   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
13   <http://purl.org/goodrelations/v1#BusinessEntity> .
14 <http://rsmluv.cz/party/420>
15   <http://purl.org/goodrelations/v1#legalName>
16   "Magistrát HMP" .
```

<sup>13</sup>Trojice nejsou z důvodu přehlednosti trojice uvedeny na samostatných řádcích

```

17 <http://rsmluv.cz/party/420>
18   <http://schema.org/address>
19   <http://rsmluv.cz/party/420/address> .
20
21 <http://rsmluv.cz/party/420/address>
22   <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
23   <http://schema.org/PostalAddress> .
24 <http://rsmluv.cz/party/420/address>
25   <http://schema.org/streetAddress>
26   "Staroměstské nám. 4" .

```

Výpis kódu 2.1: Příklad RDF dat - N-Triples

## 2.9.2 Příklad dat serializovaných ve formátu Turtle

Formát Turtle umožňuje zkracování URI pomocí prefixů. Umožňuje také zkracovat zápis tím, že nemusíme zapisovat opakující se subjekt. Jednotlivé dvojice predikát-hodnota lze tak přehledně mít u jednoho subjektu. Oddělovačem mezi dvojicemi v rámci subjektu je středník, blok informací o daném subjektu je zakončený tečkou. Pro definování typu subjektu se může použít klíčové slovo „a“, namísto predikátu `rdf:type`. Výhodou je úspornost a velmi dobrá lidská čitelnost (viz Kód 2.2).

```

1 @prefix cn: <http://tiny.cc/open-contracting#>.
2 @prefix dc: <http://purl.org/dc/terms/>.
3 @prefix gr: <http://purl.org/goodrelations/v1#>.
4 @prefix s:  <http://schema.org/>.
5
6 <http://rsmluv/contract/42/1> a cn:Contract;
7   dc:title "Softwarová zakázka";
8   cn:party <http://rsmluv/party/420>.
9
10 <http://rsmluv/party/420> a gr:BusinessEntity;
11   gr:legalName "Magistrát HMP";
12   s:address <http://rsmluv/party/420/address>.
13
14 <http://rsmluv/party/420/address> a s:PostalAddress;
15   s:streetAddress "Staroměstské nám. 4".

```

Výpis kódu 2.2: Příklad RDF dat - Turtle

Díky dobré čitelnosti, se formát Turtle hojně používá pro zapisování ontologií. V kódu 2.3 vidíme znázorněnou jednoduchou ontologii. Popisuje 2 objekty. Prvním je třída `Contract` (typ `owl:Class`). Definuje, že se jedná o smlouvu, je podtřídou (`rdfs:subClassOf`) třídy `Document` a je definována v ontologii (`rdfs:DefinedBy`) `http://tiny.cc/open-contracting`. Je to serializovaný zápis ontologie z obr. 2.7. Druhým objektem je vlastnost `party` (typ `owl:ObjectProperty`). V predikátu `rdfs:domain` je specifikováno, že vlastnost `party` může být použita u třech tříd, a to `Contract`, `Order` nebo `Invoice`. Predikát `rdfs:range` znamená, že očekávaný přiřazený objekt je typu `gr:BusinessEntity`.

```

1 @prefix :    <http://tiny.cc/open-contracting#> .
2 @prefix dc:  <http://purl.org/dc/terms/> .
3 @prefix gr:  <http://purl.org/goodrelations/v1#> .

```

```

4 @prefix owl: <http://www.w3.org/2002/07/owl#> .
5 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
6 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
7
8 :Contract a owl:Class ;
9   rdfs:label "Smlouva"@cs, "Contract"@en ;
10  rdfs:subClassOf :Document ;
11  rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
12
13 :party a owl:ObjectProperty ;
14   rdfs:label "Smluvní strana"@cs, "Party"@en ;
15   rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Contract :Order
16     :Invoice ) ] ;
17   rdfs:range gr:BusinessEntity ;
18   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .

```

Výpis kódu 2.3: Příklad RDF Ontologie - Turtle

### 2.9.3 Příklad dat serializovaných ve formátu JSON-LD

JSON-LD je jedním z poměrně nových formátů pro serializaci RDF. Jednou z motivací k vzniku byla snaha využít hojně využívané JSON dokumenty v dnešních aplikacích a co možná neefektivněji z nich vytvořit RDF data.

Uveďme si modelový příklad. V kódu 2.4 jsou ne-RDF data ve formátu JSON. Jsou validní vůči nějakému JSON Schématu a používají se v konkrétních aplikacích.

V kódu 2.5 máme stejná data v RDF podobě. Jak je vidět, jednotlivým objektům je přiřazen typ a URI. Použije se k tomu klíčových slov `@type`, resp. `@id`. K dokumentu je také přiložen kontext (klíčové slovo `@context`), kde se definuje mapování vlastností původního JSON dokumentu na RDF ontologie. Zachovává se tedy původní struktura JSON dokumentu. Kontext však nemusí být přímo součástí JSON-LD dokumentu, lze se na něj odkazovat.

Výsledkem tedy může být JSON-LD soubor (viz Kód 2.6). Jedná se tedy pouze o lehce rozšířený původní JSON dokument. Z tohoto důvodu bude pravděpodobně takový dokument nadále validní vůči JSON Schématu a použitelný ve stávajících aplikacích. Přináší však tu výhodu, že se zároveň jedná o RDF data.

```

1 {
2   "title": "Softwarová zakázka",
3   "party": {
4
5     "name": "Magistrát HMP",
6     "address": {
7
8       "streetAddress": "Staroměstské nám. 4"
9     }
10  }
11 }

```

Výpis kódu 2.4: Obyčejný JSON dokument

```

1 {
2   "@context": "http://tiny.cc/open-contracting-context",
3
4   "@id": "http://rsmluv.cz/contract/42/1",
5   "@type": "Contract",
6   "title": "Softwarová zakázka",
7   "party": {
8
9     "@id": "http://rsmluv.cz/party/420",
10    "@type": "Party",
11    "name": "Magistrát HMP",
12    "address": {
13
14      "@id": "http://rsmluv.cz/party/420/address",
15      "@type": "Address",
16      "streetAddress": "Staroměstské nám. 4"
17    }
18  }
19 }

```

Výpis kódu 2.5: Příklad RDF dat - JSON-LD s Contextem

```

1 {
2   "@context": {
3
4     "cn": "http://tiny.cc/open-contracting#",
5     "dc": "http://purl.org/dc/terms/",
6     "gr": "http://purl.org/goodrelations/v1#",
7     "s": "http://schema.org/",
8
9     "Contract": "cn:Contract",
10    "Party": "gr:BusinessEntity",
11    "Address": "s:PostalAddress",
12    "title": "dc:title",
13    "party": "cn:party",
14    "name": "gr:legalName",
15    "address": "s:address",
16    "streetAddress": "s:streetAddress"
17  },
18
19   "@id": "http://rsmluv.cz/contract/42/1",
20   "@type": "Contract",
21   "title": "Softwarová zakázka",
22   "party": {
23

```



```

24  " @id" : "http://rsmluv.cz/party/420" ,
25  "@type" : "Party" ,
26  "name" : "Magistrát HMP" ,
27  "address" : {
28
29      " @id" : "http://rsmluv.cz/party/420/address" ,
30      "@type" : "Address" ,
31      "streetAddress" : "Staroměstské nám. 4"
32  }
33 }
34 }

```

Výpis kódu 2.6: Příklad RDF dat - JSON-LD

## 3. Otevřené smlouvy

### 3.1 Situace ve veřejné správě ČR

Pokud se veřejná instituce rozhodne pro publikaci údajů o smlouvách, má dnes (rok 2015) v podstatě dvě možnosti. První možností je vyvinutí vlastní iniciativy a zveřejnění smluv na svých webových stránkách. Druhou variantou je využití již existujícího registru smluv na portálu veřejné správy[19]. Registr je to značně minimalistický, ale řešení je to dostačující.

Vzhledem k chystanému zákonu o registru smluv<sup>1</sup> se ale budoucnost stávajícího registru jeví jako značně nejistá. Lze totiž očekávat, že s velkou pravděpodobností vznikne registr zbrusu nový.

První otázkou je, kolik veřejných institucí již smlouvy zveřejňuje. Na portálu veřejné správy lze dohledat řádově několik desítek subjektů. O těchto institucích můžeme prohlásit, že oficiálně zveřejňují smlouvy. Informace o subjektech, které zveřejňují na svých webových stránkách, není systematicky zdokumentovaná vůbec. Lze ale očekávat, vzhledem k celkovému množství veřejných institucí a počtu subjektů zveřejňujících na portálu veřejné správy, že se jedná o nepatrný zlomek. Klíčem ke zlepšení situace by mohl být již zmíněný zákon o registru smluv, který mimo jiné ukládá povinnost, že pokud smlouva není zveřejněná na internetu, tak je neplatná.

Další otázkou je, jak mají data o zveřejněných smlouvách vypadat, které položky musí, či nemusí obsahovat. Není přeci cílem, aby každá veřejná instituce zveřejňovala smlouvy jinak. Obecně chybí datový standard a metodika pro zveřejňování smluv. Pokrok v tomto směru udělalo Ministerstvo vnitra ČR, které na podzim roku 2015 plánuje vydat sadu standardů pro publikovatelné datové sady veřejných institucí<sup>2</sup>. Bude se mimo jiné jednat o jakési minimální nutné doporučení, co konkrétní datová sada musí obsahovat.

V úvodu již bylo řečeno, že pod záštitou Oživení o.s. a Centra aplikované ekonomie o.s. vzniká datový standard pro otevřené smlouvy. Hlavními postavami koordinujícími vývoj standardu se stali PhDr. Ing. Jiří Skuhrovec a Mgr. Lenka Franková. Na tvorbě standardu participují a mohu konstatovat, že základní verze je již hotová. Velmi pozitivní zprávou je to, že se tento standard s velkou pravděpodobností dostane do oficiálního doporučení Ministerstva vnitra ČR. Zdá se tedy, že celá tato snaha má smysl.

Standardem pro smlouvy to ale nekončí. Myšlenka úzké spolupráce zástupců měst a obcí, akademické a neziskové sféry se osvědčila. Výsledkem je vznik organizace Otevřená města[20], která má za cíl sdružovat veřejné instituce. Pod společnou taktovkou pak financovat společné otevřené projekty. Prvním společným projektem je právě registr smluv.

<sup>1</sup>Návrh zákona o Registru smluv - tisk 42 - <http://www.psp.cz/sqw/historie.sqw?o=7&T=42>

<sup>2</sup>Standardy publikace a katalogizace otevřených dat veřejné správy ČR - <http://opendata.gov.cz/>

## 3.2 Standard pro zveřejňování smluv

V této kapitole se podrobněji seznámíme se standardem pro zveřejňování smluv. Nejdříve je vyložena základní struktura datového standardu, poté jsou popsány konkrétní položky standardu a číselníky. Následně jsou popsány způsoby publikace. Na závěr zmíníme několik informací o vznikající metodice pro zveřejňování smluv.

### 3.2.1 Základní struktura

Základním objektem, který slouží k reprezentaci dat, je dokument. Jedná se o abstraktní entitu, která nabývá tří rozšířeních typu smlouva/příloha/dodatek. Tato rozšíření obsahují všechny položky obsažené v dokumentu a navíc konkrétní položky pro daný typ. Smluvní strany jsou separátní objekty navázané buď na smlouvu, objednávku nebo fakturu pomocí jednoznačného identifikátoru. Objedávka a faktura jsou separátní objekty, které se mohou vázat na konkrétní smlouvu/přílohu/dodatek pomocí jednoznačného identifikátoru. Rozšiřující entity mohou být součástí smlouvy, příp. objednávky. Reprezentují důležité události v životním cyklu dokumentu a jednotlivé transakce.

#### Reprezentované entity

- **Dokument** - základní abstraktní struktura pro evidování údajů o smlouvách/přílohách/dodatcích
  - **Smlouva** - detailní popisné údaje smlouvy
  - **Příloha** - popisné údaje přílohy
  - **Dodatek** - popisné údaje dodatku
  - **Vydavatel** - informace o vydavateli, který zveřejňuje údaje o smlouvách
  - **Verze** - identifikace jednotlivé verze dokumentu
- **Smluvní strana** - popisné údaje smluvní strany
  - **Nadřazená instituce** - informace o řídicí nebo ovládací právní osobě vystupující u smluvní strany
  - **Adresa** - podrobné údaje o adrese u smluvní strany
- **Objedávka** - popisné údaje objednávky, jedná se o doplňující informace k smlouvě/příloze/dodatku
- **Faktura** - popisné údaje faktury, jedná se o doplňující informace k smlouvě/příloze/dodatku
- **Rozšiřující entity** - rozšířené informace ke smlouvě, příp. objednávce
  - **Milník** - reprezentuje důležitou událost v životním cyklu smlouvy
  - **Transakce** - reprezentuje proběhlou platbu na základě smlouvy

Název pole	Popis
Název pole	Jméno reprezentující danou položku
Datový typ	Přípustný datový typ položky
Validita	Stupeň kvality položky.
Popis	Podrobný popis položky

Tabulka 3.1: zdroj:[24]

Datový model je rozdělen do tabulek podle jednotlivých reprezentovaných entit. Každá dílčí položka entity obsahuje tyto informace:

U každé zveřejněné smlouvy rozlišujeme tři stupně validity, resp. správnosti a úplnosti dat: A (kvalitní), B (dobrý), C (základní). Dokumenty musí splňovat alespoň minimální přípustnou kvalitu C. Pokud je nějaký atribut požadován pro stupeň validity C, je níže v textu označen např. takto (C). Položky doplněné systémem jsou označeny (S). Nepovinné položky jsou značeny (N), hvězdička znamená, že položka je kontrolována pokročilejším pravidlem popsáním u konkrétní položky.

Status	Validita	Popis
Nepovinné	N	Nepovinná položka
Základní	C	Povinná položka
Dobrý	B	Rozšiřující položka pro status „Dobrý“
Kvalitní	A	Rozšiřující položka pro status „Kvalitní“
Systémové	S	Položka doplněná systémem

Tabulka 3.2: Validita, zdroj:[24]

### Doplňující validační pravidla

Na entity se vztahují další validační pravidla, která nelze přehledně zachytit v rámci popisu jednotlivých položek. Jejich výčet je zde.

- Dokument je buď v strojově čitelném formátu (viz. Akceptovatelné soubory), nebo je k němu poskytnut plain text. Pro smlouvy účinné od 1.6.2015<sup>3</sup> je přípustná pouze varianta ve strojově čitelném formátu.
- U smlouvy typu darovací nesmí být připojeny faktury, ani jedna smluvní strana nesmí být identifikována jako Payer.
- Entita (Vydavatel/Smluvní strana/Nadřazená instituce) má vyplněno buď ID a nebo NoID = „true“.

### Akceptovatelné soubory

Dokumenty připojené ke smlouvám by měly být strojově čitelné, resp. v těchto formátech:

<sup>3</sup>Předběžně, bude upřesněno

Formát	Validita	Popis
PDF	C	Portable Document Format - ideálně strojově čitelný
DOC	C	Textový dokument Microsoft Word
XLS	C	Tabulka Microsoft Excel
DOCX	B	Textový dokument Microsoft Word
ODT	B	Textový dokument OpenDocument
XLSX	B	Tabulka Microsoft Excel
ODS	B	Tabulka OpenDocument

Tabulka 3.3: Akceptovatelné soubory,zdroj:[24]

### 3.2.2 Reprezentované entity

#### Dokument

Název pole	Datový typ	Validita	Popis
URI	String URI	S	Jednoznačný identifikátor formou URL. Typicky rsmluv.cz/[Typ]/[Id]/[Version], kde Version je vzestupné číslování verzí při změnách dokumentu či metadat
Document	String URI	S	Adresa URL fyzického umístění dokumentu. Typicky rsmluv.cz/[Typ]/[Id]/[Version]/File, viz akceptovatelné soubory
Versions	Object array	S	Údaje o verzi dokumentu. Viz entitia Verze
Type	String/ String enum	C	Typ dokumentu. Nabývá hodnot - Smlouva/Příloha/Dodatek
Publisher	Reference	C	Informace o vydavateli. Viz entitia Vydavatel
Valid	Boolean	B/S	Indikuje, zda dokument je platný, tj. nebyl zneplatněn nebo nahrazen novou verzí
PlainText	String	B/S	Prostý text dokumentu (nestrukturovaný, indexovatelný), alternativa pro scanované dokumenty
ResponsiblePersons	String array	B	Výčet odpovědných osob
Anonymised	Boolean	B	Značí, zda-li byla provedena anonymizace dokumentu

Tabulka 3.4: Vlastnosti dokumentu, zdroj:[24]

#### Vydavatel

Název pole	Datový typ	Validita	Popis
ID	String	N	Identifikační číslo osoby, lze vložit i zahraniční ID
Name	String	C	Název, případně jméno a příjmení (s tituly)
NoID	Boolean	B	Indikuje že subjekt nemá IČO, nebo zahraniční ID
Country	String	B	Země původu, 3-písmenný ISO kód
Authentication	String	C	Za účelem ověření identity

## Verze

Název pole	Datový typ	Validita	Popis
PublisherId	String	N	Libovolný číselný identifikátor verze, spisové číslo apod.
Version	Int	S	Pořadové číslo verze, nejvyšší = aktuální
URI	String URI	S	Identifikátor dané verze
Published	DateTime	S	Datum publikace v systému

Tabulka 3.6: Vlastnosti verze smlouvy, zdroj:[24]

## Smlouva

Název pole	Datový typ	Validita	Popis
AwardID	String	N*	Evidenční číslo veřejné zakázky. Uvádí se volitelně, pokud existuje
AwardProfileID	String	N	Číslo zakázky na profilu zadavatele
Amount	Nullable float	C*	Cena s DPH (u neplátců celková cena). Nejvyšší přípustná hodnota řádného plnění z dané smlouvy, které vynaloží některá smluvní strana. U smluv na dobu určitou se jedná o očekávané celkové finanční plnění strany s nejvyšším plněním, včetně opcí, bez sankcí. U smluv na dobu neurčitou, ve kterých není stanoven strop na celkové plnění, se jedná o nejvyšší očekávané roční plnění. U smluv bez finančního plnění (bartery, darovací smlouvy) je uvedena celková hodnota nefinančního plnění strany s nejvyšším plněním (např. odhadovaná hodnota daru). U smluv s nejasným plněním připustit NULL. Pokud je cena nenulová, tak alespoň jedna Smluvní strana (Party) musí mít příznak Payer = true
AmountNoVat	Nullable float	C*	Cena bez dph, uvádí se povinně pouze v případě, že Amount je s DPH
Title	String	C	Předmět smlouvy
ContractType	String	C	Číselník typů smlouvy, viz Číselníky
Parties	StringURI/ Int array	C	Seznam identifikátorů (URI nebo LocalID) smluvních stran. Viz entitita Smluvní strana
SubjectType	String	B	Číselník typů zboží/služeb, viz. Číselníky
PriceAnnual	Boolean	B	Identifikuje, pokud je v Amount roční částka
Currency	String	B	Měna, 3-písmenný, ISO 4217 formát
DateSigned	Date	B	Datum posledního podpisu
ValidFrom	Date	B	Datum účinnosti smlouvy

Název pole	Datový typ	Validita	Popis
ValidUntil	Date	B	Datum ukončení účinnosti smlouvy (poslední plnění), NULL pro smlouvy na dobu
Funding	String	B	Převažující financování – vlastní, případně název dotačního titulu (bude kontrolován proti číselníku, viz. Číselníky)
Attachments	String URI-array	B	Seznam URI identifikátorů příloh. Viz entita Příloha
Amendments	String URI-array	B	Seznam URI identifikátorů dodatků. Viz entita Dodatek
Competency	String/ String enum	A	Indikuje, zda-li se jedná o soukromoprávní nebo veřejnoprávní smlouvu
CurrentValidContract	String URI	A	Aktuálně platné znění smlouvy (se zapracovanými dodatky)
Description	String	A	Popis předmětu smlouvy
Implementation	Object	A	Objekt reprezentující transakce a milníky, viz entita Implementation

Tabulka 3.7: Vlastnosti smlouvy, zdroj:[24]

## Příloha

Název pole	Datový typ	Validita	Popis
Title	String	C	Název
Contract	String URI	C	Jednoznační identifikátor smlouvy
Number	Int	B	Číslo přílohy

Tabulka 3.8: Vlastnosti přílohy, zdroj:[24]

## Dodatek

Název pole	Datový typ	Validita	Popis
Title	String	C	Název
Contract	String URI	C	Jednoznační identifikátor smlouvy
Number	Int	B	Pořadové číslo dodatku (podle času podpisu)
DateSigned	Date	B	Datum podpisu

Tabulka 3.9: Vlastnosti dodatku, zdroj:[24]

## Smluvní strana

Název pole	Datový typ	Validita	Popis
ID	String	N	Identifikační číslo osoby, lze vložit i zahraniční id
LocalID	String URI/Int	C	Jednoznačný identifikátor v rámci dokumentu
Name	String	C	Název, případně jméno a příjmení (s tituly)
Payer	Boolean	C*	Identifikuje stranu která bude finančně plnit, pokud není zřejmé, nevyplňuje se
NoID	Boolean	B	Indikuje že subjekt nemá IČO, nebo zahraniční ID
Country	String	B	Země původu, 3-písmený ISO kód
Address	String/Referer	A	Adresa subjektu, případně "Anonymizováno". Umožňuje zadat adresu jako prostý řetězec, nebo strukturovaně, viz entitia Adresa
PaysVAT	Boolean	A	Indikuje, zda-li je subjekt plátce DPH
SuperiorInstitution	Reference	N/S	Řídící nebo ovládající právnická osoba, v případě veřejnoprávních smluv nadřízený správní orgán. Viz. Nadřazená instituce

Tabulka 3.10: Vlastnosti smluvní strany, zdroj:[24]

### Nadřazené instituce

Název pole	Datový typ	Validita	Popis
ID	String	N	Identifikační číslo osoby, lze vložit i zahraniční id
LocalID	String URI/Int	C	Jednoznačný identifikátor v rámci dokumentu
Name	String	C	Název, případně jméno a příjmení (s tituly)
NoID	Boolean	B	Indikuje že subjekt nemá IČO, nebo zahraniční ID
Country	String	B	Země původu, 3-písmený ISO kód

Tabulka 3.11: Vlastnosti nadřazené instituce, zdroj:[24]



## Adresa

Název pole	Datový typ	Validita	Popis
StreetAddress	String	A	Ulice, případně "Anonymizováno"
Locality	String	A	Město, případně "Anonymizováno"
PostalCode	Integer	A	PSC, případně "Anonymizováno"
Nuts	String	A	Normalizovaná klasifikace územních celků (např. Praha - CZ010), případně "Anonymizováno"

Tabulka 3.12: Vlastnosti adresy, zdroj:[24]

## Objednávka

Název pole	Datový typ	Validita	Popis
ParrentDocument	String URI	N	Jednoznačný identifikátor dokumentu
SubjectType	String	N	Číselník typů zboží/služeb, viz. Číselníky
Parties	String URI/Int array	N	Seznam identifikátorů (URI nebo LocalID) smluvních stran. Viz entitia Smluvní strana
Title	String	C	Předmět
Amount	Float	C	Cena s DPH
Currency	String	B	Měna, 3-písmenný, ISO 4217 formát
DateSigned	Date	B	Datum posledního podpisu
Implementation	Object	A	Objekt reprezentující transakce a milníky, viz entitia Implementation

Tabulka 3.13: Vlastnosti objednávky, zdroj:[24]

## Faktura

Název pole	Datový typ	Validita	Popis
ParrentDocument	String URI	N	Jednoznačný identifikátor dokumentu
Parties	String URI/Int array	N	Seznam identifikátorů (URI nebo LocalID) smluvních stran. Viz entitia Smluvní strana
Title	String	C	Předmět
Amount	Float	C*	Cena s DPH (u neplátců celková cena).
Currency	String	B	Měna, 3-písmenný, ISO 4217 formát
DateSigned	Date	B	Datum posledního podpisu
DueDate	Date	B	Datum splatnosti

Tabulka 3.14: Vlastnosti faktury, zdroj:[24]

## Rozšiřující entity

### Implementace

Název pole	Datový typ	Validita	Popis
Milestones	Object arra	A	Milníky, pro volnou evidenci událostí (obnova smlouvy, předání apod.). Viz entitia Milník
Transactions	Object array	A	Seznam transakcí, tedy proběhlých plateb na základě smlouvy. Viz entitia Transakce

Tabulka 3.15: Vlastnosti implementace, zdroj:[24]

### Milník

Název pole	Datový typ	Validita	Popis
Title	String	C	Název
DueDate	String	C	Datum

Tabulka 3.16: Vlastnosti milníku, zdroj:  
[24]

### Transakce

Název pole	Datový typ	Validita	Popis
Date	DateTime	C	Datum a čas proběhlé transakce
Ammount	Float	C	Zaplacená cena s DPH, vždy stejná měna jako v Currency
SenderOrganization	Reference	C	Informace o odesílateli. Viz entitia Party
ReceiverOrganization	Reference	C	Informace o příjemci. Viz entitia Party
PublisherId	String	B	Libovolný číselný identifikátor transakce, unikátní v rámci smlouvy

Tabulka 3.17: Vlastnosti transakce, zdroj:[24]

### 3.2.3 Číselníky

V následujících tabulkách jsou znázorněny přípustné hodnoty číselníků Typ dokumentu (vlastnost Type u entity Dokument) a Typ smlouvy (vlastnost ContractType u entity Smlouva). Číselník Typ zboží a služeb (položka SubjectType u entity Smlouva) je zveřejněn na portálu informačního systému o veřejných za-

kázkách<sup>4</sup>.

Hodnota
Smlouva
Příloha
Dodatek

Tabulka 3.18: Číselník typu dokumentu, zdroj:[24]

Hodnota
Nájemní smlouva
Darovací smlouva
Kupní smlouva
Směnná smlouva
Pojistná smlouva
Smlouva o výpůjčce
Licenční smlouva
Mandátní smlouva
Leasingová smlouva
Pachtovní smlouva
Smlouva o zřízení věcného břemene
Smlouva o provedení stavby
Smlouva o provedení práce
Smlouva o provedení uměleckého výkonu
Smlouva o úvěru
Smlouva o uzavření budoucí smlouvy
Veřejnoprávní smlouva
Jiná

Tabulka 3.19: Číselník typu smlouvy, zdroj:[24]

---

<sup>4</sup>Číselníky a klasifikace na portálu informačního systému o veřejných zakázkách - TODO

## 3.3 Publikace

Pro potřeby publikace je třeba zvolit vhodný datový formát v kterém budou otevřené smlouvy přenositelné. Jako kritéria výběru vhodného formátu stanovíme čtyři podmínky:

- otevřený datový formát - tím zaručíme otevřená data na úrovni kvality 3★
- obecná znalost a jednoduchost datového formátu - cílem je, aby valná většina IT specialistů ve veřejných institucích formát znala
- existence volně dostupných nástrojů k čtení a zpracování datového formátu
- možnost tvorby datového schématu - resp. možnost určit soustavu speciﬁkáci a pravidel, jak má datový soubor vypadat, aby byl validní

Není překvapující, že obecně nejznámějšími datovými formáty splňujícími výše zmíněná pravidla jsou formáty XML (Extensible Markup Language) a JSON (JavaScript Object Notation)[21]. Vzhledem k úspornosti a možností rychlejšího zpracování padla volba na formát JSON.

Pokud však chceme, aby datový standard byl součástí plánovaného doporučení Ministerstva vnitra ČR, tak je nutné podporovat také formát CSV (Comma-separated values)[22]. Jedná se o jednoduchý, otevřený datový formát, ale s plochou strukturou. Publikace smluv v CSV si tedy vyžádá řadu omezení.

### 3.3.1 JSON

Základní strukturu datového souboru lze vidět z Tabulky 3.20. Položky Id, Date a Language slouží k popisu datového souboru jako celku. Položky Documents, Parties, Orders a Invoices už obsahují konkrétní výčty entit ze standardu. Položky vyznačené stupněm validity C, jsou povinné.

Ke konkrétní specifikaci jednotlivých položek ve formátu JSON se používá JSON Schema[23]<sup>5</sup>. Lze v něm definovat konkrétní elementy a podelementy, výchozí hodnoty, datové typy, požadovaný obsah apod. Příklad JSON Schématu vycházejícího z datového standardu lze nalézt v příloze A.

Datový soubor, validní vůči JSON schématu, s jednou smlouvou a dvěma smluvními stranami můžeme vidět na příkladu kódu 3.1.

---

<sup>5</sup>popis způsobu zápisu konkrétních položek je nad rámce této práce

```

1 {
2   "id": "89f689cd-e784-4374-bb17-94144679d46f",
3   "published": "2014-03-25T23:20:50+01:00",
4   "language": "cs",
5
6   "documents": [
7     {
8       "uri" : "http://rsmluv.cz/smlouva/12345",
9       "document" : "http://rsmluv.cz/smlouva/12345/Smlouva12345.docx",
10      ,
11      "type" : "Smlouva",
12      "valid" : true,
13      "anonymised" : false,
14
15      "awardID": "486026",
16      "awardProfileID": "OI-010143",
17      "amount": 584520.00,
18      "title": "Brno, Vackova, Šafaříkova – rekonstrukce kanalizace
19      a vodovodu",
20      "contractType": "Kupní smlouva",
21      "subjectType": "Právní, finanční překladatelské, pojišťovnické
22      , poradenské a jiné služby",
23      "priceAnnual": false,
24      "currency": "CZK",
25      "dateSigned": "2011-11-16",
26      "validFrom": "2011-11-02",
27      "validUntil": "2012-06-30",
28      "funding" : "vlastní",
29      "competency" : [ "Soukromoprávní smlouva" ],
30      "currentValidContract": "http://zakazky.brno.cz/?pg=detail&id
31      =18249&list=135",
32      "description": "Projektová dokumentace pro stavební povolení a
33      zadání stavby bude řešit rekonstrukci stávající kanalizační
34      stoky z profilu DN 500 na DN 800/1200 v délce 146 m, rekonstrukci
35      kanalizačních přípojek pod veřejným prostranstvím a přepojení vš
36      ech stávajících dešťových vpustí a také vybourání vozovek a chodn
37      íků nad rýhou a zásyp rýhy recyklátem. Součástí bude inženýrsko-
38      geologický průzkum, geodetické zaměření dotčené oblasti,
39      inventarizace zeleně, výkaz výměr, položkový rozpočet a výkon
40      autorského dozoru až do dokončení stavby. Dokumentace bude
41      projednána s orgány státní správy a s účastníky stavebního řízení
42      a jejich připomínky budou do dokumentace zapracovány.",
43
44      "responsiblePersons" : [ "Ing. Petr Vokřál ", "Mgr. Adriana
45      Krnáčová, MBA" ],
46
47      "publisher" : {
48        "id" : "6003508",
49        "name" : "Statutární město Brno",
50        "noID" : false,
51        "country" : "CZE",
52        "authentication" : "email"
53      },
54
55      "parties" : [ 132456, 987654 ],
56
57      "Implementation" : {
58        "milestones" : [

```

```

44         {
45             "title" : "Výpověď smlouvy",
46             "dueDate" : "2012-06-20T23:20:50+01:00"
47         }
48     ],
49     "transactions" : [
50         {
51             "publisherId" : "1269483",
52             "date" : "2012-01-01T18:35:20+01:00",
53             "amount" : 300000,
54             "senderOrganization" : 987654,
55             "receiverOrganization" : 132456
56         },
57         {
58             "publisherId" : "934584",
59             "date" : "2012-02-01T09:13:40+01:00",
60             "amount" : 284520,
61             "senderOrganization" : 987654,
62             "receiverOrganization" : 132456
63         }
64     ]
65 },
66
67 "versions" : [
68     {
69         "version" : 1,
70         "uri" : "http://rsmluv.cz/smlouva/12345/verze/1",
71         "published" : "2014-09-15T23:20:50+01:00"
72     },
73     {
74         "version" : 2,
75         "uri" : "http://rsmluv.cz/smlouva/12345/verze/2",
76         "published" : "2015-03-15T14:35:28+01:00"
77     }
78 ]
79 }
80 ],
81
82 "parties" : [
83     {
84         "id" : "44992785",
85         "localID" : 132456,
86         "name" : "Statutární město Brno",
87         "payer" : false,
88         "noID" : false,
89         "country" : "CZE",
90         "address" : {
91             "streetAddress" : "Dominikánské náměstí 196/1",
92             "locality" : "Brno-město, Brno",
93             "postalCode" : 60200,
94             "nuts" : "CZ064"
95         },
96         "superiorInstitution" : {
97             "id" : "00064581",
98             "localID" : 56486,
99             "name" : "Magistrát hlavního města Prahy",
100             "noID" : false,
101             "country" : "CZE"

```

```

102     }
103   },
104   {
105     "id" : "46347011",
106     "localID" : 987654,
107     "name" : "Kovoprojekta Brno a.s.",
108     "payer" : true,
109     "noID" : false,
110     "country" : "CZE",
111     "address" : {
112       "streetAddress" : "Šumavská 416/15",
113       "locality" : "Ponava, Brno",
114       "postalCode" : 60200,
115       "nuts" : "CZ064"
116     }
117   }
118 ]
119 }

```

Výpis kódu 3.1: JSON soubor s jednou smlouvou

### 3.3.2 CSV

Formát CSV je jednoduchou plochou strukturou, nelze tedy pomocí tohoto formátu zaznamenat úplnou strukturu datového standardu. Řešením by mohlo být rozdělit údaje o smlouvách do sady CSV souborů. Tím se ale ztrácí výhoda jednoduchosti CSV. Cílem publikace v CSV je maximální jednoduchost pro vydavatele. Proto jsme přistoupili k následujícím omezením<sup>6</sup>

- Vše je smlouvou, tedy nebudeme evidovat dodatky, přílohy, faktury a objednávky
- Vypuštěny/omezeny vlastnosti u smlouvy
  - URI - nahrazeno odkazem na podrobné údaje o smlouvě s validitou A
  - Type
  - Verzování, resp. vypuštěna vazba na verze a vlastnost Valid
  - PlainText
  - Vydavatel, převážně proto, že MV má pro vydavatele speciální strukturu
  - Pouze jedna zodpovědná osoba
  - Vypuštěny rozšiřující entity - milníky a transakce
- Umožněny pouze dvě smluvní strany - Publisher a Partner, a to s vlastnostmi
  - ID, Name, Payer, Country, Address a PaysVAT
- Nové vlastnosti
  - LocalID - Libovolný číselný identifikátor smlouvy, spisové číslo apod. - validita A
  - NumberOfAmendments - Počet dodatků - validita B
  - LastAmendmentDateSigned - Datum podpisu posledního dodatku - validita A
  - FirstInvoiceDueDate - Datum splatnosti první faktury - validita A
  - LastInvoiceDueDate - Datum splatnosti poslední faktury - validita A
  - TotalFillingValue - Celkový objem plnění - validita A

---

<sup>6</sup>Stav k červenci 2015



## 3.4 Metodika zveřejňování smluv

Spolu s datovým standardem vzniká i metodika mající za cíl technicky i věcně datový standard popsat. Tvorbu této metodiky jsem pod taktovkou Jiřího Skuhrovce dostal na starost.

Jedná se o jednoduchou webovou aplikaci na bázi wikipedie[25]. Implementována je pomocí nástroje Dokuwiki[26].

Metodika zveřejňování smluv

HISTORIE: CS

### Dokument

Dokument je základní strukturou pro evidování údajů o [smlouvách/přílohách/dodatcích](#). Každý dokument má přiděleno jednoznačné ID a typ, zda-li se jedná o smlouvu, přílohu nebo dodatek. Pro jednotlivé typy rozlišujeme různé datové struktury.

Název pole	Datový typ	Validita	Popis
URI	String URI	S	Jednoznačný identifikátor formou URL. Typicky <i>rsmluv.cz/[Typ]/[Id]/[Version]</i> , kde Version je vzestupné číslování verzí při změnách dokumentu či metadat
Document	String URI	S	Adresa URL fyzického umístění dokumentu. Typicky <i>rsmluv.cz/[Typ]/[Id]/[Version]/File</i> , akceptovatelné soubory viz. <a href="#">Validita</a>
Versions	Object array	S	Údaje o verzi dokumentu. Viz. <a href="#">Verze</a>
Type	String/String enum	C	Typ dokumentu. Nabývá hodnot - <a href="#">Smlouva/Příloha/Dodatek</a>
Publisher	Reference	C	Informace o vydavateli. Viz. <a href="#">Vydavatel</a>
Valid	Boolean	B/S	Indikuje, zda dokument je platný, tj. nebyl zneplatněn nebo nahrazen novou verzí
PlainText	String	B/S	Prostý text dokumentu (nestrukturovaný, indexovatelný), alternativa pro scanované dokumenty
ResponsiblePersons	String array	B	Výčet odpovědných osob
Anonymised	Boolean	B	Značí, zda-li byla provedena anonymizace dokumentu

Obsah

- Dokument
- Vydavatel
- Verze

Upravit

Obrázek 3.1: Metodika zveřejňování smluv, zdroj: <http://standard.zindex.cz/>

## 4. Otevřené smlouvy jako Linked Data

V minulé kapitole<sup>3</sup> bylo řečeno, jak reprezentovat smlouvy jako otevřená data. Data serializovaná do formátu JSON, či CSV můžeme kategorizovat stupněm 3★. Pokud však chceme dosáhnout otevřenosti dat kategorie 5★, je třeba provést několik dalších kroků:

- Identifikovat reprezentované objekty a vlastnosti pomocí URI
- Vytvořit strojově srozumitelné struktury, resp. napojit data na konkrétní slovníky tříd a predikátů - ontologie
- Propojit smlouvy pomocí odkazů na související data

### 4.1 Přiřazení identifikátorů jednotlivým entitám otevřených smluv

K jednoznačné identifikaci každé entity nám stačí její *typ* a *Id*. Výjimku tvoří dokumenty, které jsou verzované. K nim je nutné přidat informaci o konkrétní verzi. Dále chceme vyjádřit vztah podřízené entity k nadřízené. Řešením je opět přidání informací o typu podřízené entity, příp. jejího *Id*. Resp. základní URI schéma bude:

*http://[domain]/[typ]/[entitaId]/[versionId]/[childEntity]/[childEntityId]*

- domain je doména instituce publikující smlouvy
- údaje ve složených závorkách jsou volitelné

Výsledné identifikátory entit:

- **Document** (Dokument) - *http://[domain]/[type]/[documentId]/[versionId]*  
Type může nabývat hodnot contract/attachment/amendment - resp. jedná se v podstatě o hodnotu položky Uri z datového standardu
- **Publisher TODO** (Vydavatel) -  
*http://[domain]/[type]/[documentId]/[versionId]/publisher* Jedná se o podřízenou položku dokumentu
  - Jedná se o podřízenou položku dokumentu
- **Version** (Verze) - *http://[domain]/[type]/[documentId]/[versionId]/version*  
Jedná se o podřízenou položku dokumentu
  - Jedná se o podřízenou položku dokumentu
- **Order** (Objednávka) - *http://[domain]/order/[orderId]*

- **Invoice** (Faktura) - *http://[domain]/invoice/[invoiceId]*
- **Implementation** (Implementace)
  - *http://[domain]/[type]/[documentId]/[versionId]/implementation*
    - \* Pokud se jedná o podřízenou položku dokumentu
  - *http://[domain]/order/[orderId]/implementation*
    - \* Pokud se jedná o podřízenou položku objednávky
- **Milestone** (Milník)
  - Zde si dovolíme drobné zjednodušení, a to vynechání implementačního z identifikátoru. Adresa bude jednodušší, informační hodnota však zůstane stejná
  - *http://[domain]/[type]/[id]/[versionId]/milestone/[milestoneId]*
    - \* Milník u dokumentu
  - *http://[domain]/order/[orderId]/milestone/[milestoneId]*
    - \* Milník u objednávky
- **Transaction** (Transakce)
  - Zjednodušení, viz Milník
  - *http://[domain]/[type]/[id]/[versionId]/transaction/[transactionId]*
    - \* Transakce u dokumentu
  - *http://[domain]/order/[orderId]/transaction/[transactionId]*
    - \* Transakce u objednávky
- **Party** (Smluvní strana) - *http://[domain]/party/[partyId]*
- **Address** (Adresa) - *http://[domain]/party/[partyId]/address*
  - Jedná se o podřízenou položku smluvní strany
- **SuperiorInstitution** (Nadřazená instituce) - *http://[domain]/superiorInstitution/[superiorInstitutionId]*

## 4.2 Ontologie pro publikaci dat o smlouvách

Než začneme s tvorbou ontologie, je dobré si uvědomit, že vycházíme z již hotového datového standardu. Nemáme tedy v tvorbě úplnou svobodu. Cílem tedy bude tvorba takové ontologie, která bude odpovídat stávajícímu datovému standardu, a přesto se bude snažit využít co nejvíce již existujících ontologií.

Samotnou tvorbu ontologie rozdělíme do dvou kroků:

- Analýza vhodných, již existujících ontologií
- Tvorba samotné ontologie

### 4.2.1 Analýza vhodných, již existujících ontologií

Při tvorbě ontologie se zaměříme na otázku, zdali existuje třída, či predikát v nějaké ontologii sémanticky ekvivalentní třídě, či konkrétní položce datového standardu pro smlouvy. Takových vhodných ontologií ve světě Linked Data může být celá řada. K výběru stačí libovolná z nich.

V následujícím seznamu je výčet vybraných ontologií, které se jeví jako vhodné pro použití při popisování smluv<sup>1</sup>. U každého bodu je zmíněn popis ontologie, důvod, proč byla daná ontologie zvolena, a seznam tříd a predikátů vhodných k použití.

Vybrané ontologie:

- **Commerce** (<https://w3id.org/commerce#>) - ontologie pro popisování obchodních transakcí
  - Důvod použití - užitečná třída transakce
  - Vybrané třídy
    - \* Transaction - třída reprezentující transakci
  - Vybrané predikáty
    - \* contentUrl - adresa obsahu
    - \* source - zdroj transakce (pozor na party)
    - \* destination - cíl transakce (pozor na party)
- **DublinCore** (<http://purl.org/dc/terms/>) - základní ontologie pro popis metadat
  - Důvod použití - základní a všeobecně známá ontologie popisující metadata
  - Vybrané predikáty
    - \* created - datum vytvoření
    - \* creator - tvůrce
    - \* date - obecné datum
    - \* description - popis metadat
    - \* identifier - jednoznačný identifikátor
    - \* issued - datum publikace
    - \* language - jazyk
    - \* modified - datum modifikace
    - \* publisher - vydavatel
    - \* rights - licence
    - \* title - název dokumentu
    - \* type - typ dokumentu

---

<sup>1</sup>Obecně výběr ontologií nemusíme považovat za striktní. Každou třídu, či predikát lze označit jako sémanticky ekvivalentní jiné třídě, či predikátu. Slouží k tomu konstrukce jazyka Owl - *equivalentClass*, resp. *equivalentProperty*.

- **Friend-of-a-Friend** (<http://xmlns.com/foaf/0.1>) - ontologie pro popis vazeb mezi lidmi
  - Důvod použití - vhodná pro označení třídy vydavatele
  - Vybrané třídy
    - \* Person - třída reprezentující osobu
    - \* Organization - třída reprezentující organizaci
  - Vybrané predikáty
    - \* name - jméno osoby
    - \* mbox - email osoby
- **GoodRelations** (<http://purl.org/goodrelations/v1#>) - ontologie pro popis produktů, cen a obchodních dat
  - Důvod použití - známá ontologie, vhodná pro popis smluvních stran a informací o cenách
  - Vybrané třídy
    - \* BusinessEntity - třída popisující hospodářské subjekty
  - Vybrané predikáty
    - \* hasCurrency - měna
    - \* hasCurrencyValue - cena
    - \* legalName - název subjektu
    - \* valueAddedTaxIncluded - plátce DPH
- **PaySwarm** (<https://w3id.org/payswarm#>) - ontologie popisující účtenky, platby, pronájmy a obecně výměnné platby na webu
  - Důvod použití - obsahuje predikáty popisující intervaly platnosti
  - Vybrané predikáty
    - \* validFrom - datum platnosti od
    - \* validUntil - datum platnosti do
- **Schema** (<http://schema.org/>) - obecná ontologie mající za cíl pokrývat co největší možné množství informací
  - Důvod použití - známá ontologie, možnost využití pro popis adresních údajů
  - Vybrané třídy
    - \* PostalAddress - třída reprezentující adresu
  - Vybrané predikáty
    - \* address - adresa
    - \* addressCountry - země
    - \* addressLocality - město
    - \* postalCode - PSČ

\* streetAddress - ulice

- **Vann** (<http://purl.org/vocab/vann/>) - anotační ontologie pro dokumenty
  - Důvod použití - nesouvisí s datovým standardem, tato ontologie je vhodná pro popis ontologií a bude zmíněna níže v publikaci[TODO-odkaz].
  - Vybrané predikáty
    - \* preferredNamespaceUri - preferovaná adresa ontologie
    - \* preferredNamespacePrefix - preferovaná zkratka
    - \* usageNote - poznámka k použití

### 4.2.2 Tvorba ontologie

Každou položku datového standardu namapujeme na třídu, či predikát výsledné ontologie. Pro některé položky využijeme zmíněné predikáty z již existujících ontologií, pro ostatní vytvoříme třídy a predikáty vlastní.

Vlastní ontologii nazveme jako open-contracting a budeme pro ni používat zkratku cn.

Výsledné mapování můžeme vidět v následujících tabulkách 4.1 - 4.13. V prvním sloupečku se nachází entita/vlastnost datového standardu, v druhém namapovaná třída/predikát a v třetím případná poznámka<sup>2</sup>.

#### Dokument

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>uri</b>	cn:uri	
<b>document</b>	com:contentUrl	
<b>versions</b>	cn:version	
<b>type</b>	dcmi:type	
<b>publisher</b>	dc:publisher	
<b>valid</b>	cn:valid	
<b>plainText</b>	cn: plainText	
<b>responsiblePersons</b>	cn:responsiblePerson	
<b>anonymised</b>	cn:anonymised	

Tabulka 4.1: Mapování entity Document

---

<sup>2</sup>Entity jsou uváděny velkým písmem, predikáty malým

## Vydavatel

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Publisher</b>	foaf:Organization	
<b>id</b>	dc:identifier	
<b>name</b>	gr:legalName	
<b>noID</b>	cn:noID	
<b>country</b>	schema:addressCountry	
<b>authentication</b>	cn:authentication	

Tabulka 4.2: Mapování entity Vydavatel

## Verze

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Version</b>	cn:Version	
<b>publisherId</b>	cn:publisherId	
<b>version</b>	cn:versionOrder	
<b>uri</b>	cn:uri	
<b>published</b>	dc:issued	

Tabulka 4.3: Mapování entity Verze smlouvy

## Smlouva

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Contract</b>	cn:Contract	
<b>awardID</b>	cn:awardID	
<b>awardProfileID</b>	cn:awardProfileID	
<b>amount</b>	gr:hasCurrencyValue	
<b>amountNoVat</b>	gr:hasCurrencyValue	
<b>title</b>	dc:title	
<b>contractType</b>	cn:contractType	
<b>parties</b>	cn:party	
<b>subjectType</b>	cn:subjectType	
<b>priceAnnual</b>	cn:priceAnnual	
<b>currency</b>	gr:hasCurrency	
<b>dateSigned</b>	dc:created	
<b>validFrom</b>	ps:validFrom	
<b>validUntil</b>	ps:validUntil	

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>funding</b>	cn:funding	
<b>attachments</b>	cn:attachment	
<b>amendments</b>	cn:amendment	
<b>competency</b>	cn:competency	
<b>currentValidContract</b>	cn:currentValidContract	
<b>description</b>	dc:description	
<b>implementation</b>	cn:implementation	

Tabulka 4.4: Mapování entity Smlouva

## Příloha

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Attachment</b>	cn:Attachment	
<b>title</b>	dc:title	
<b>contract</b>	cn:contract	
<b>number</b>	cn:attachmentOrder	

Tabulka 4.5: Mapování entity Příloha

## Dodatek

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Amendment</b>	cn:Amendment	
<b>title</b>	dc:title	
<b>contract</b>	cn:contract	
<b>number</b>	cn:amendmentOrder	
<b>dateSigned</b>	dc:created	

Tabulka 4.6: Mapování entity Dodatek

## Smluvní strana

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Party</b>	gr:BusinessEntity	
<b>id</b>	dc:identifier	
<b>localID</b>	cn:localID	
<b>name</b>	gr:legalName	
<b>payer</b>	cn:payer	



Entita/Vlastnost	Třída/Predikát	Poznámka
<b>noID</b>	cn:noID	
<b>country</b>	schema:addressCountry	
<b>paysVAT</b>	gr:valueAddedTaxIncluded	
<b>superiorInstitution</b>	cn:superiorInstitution	

Tabulka 4.7: Mapování entity Smluvní strana

### Nadřazená instituce

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>SuperiorInstitution</b>	gr:BusinessEntity	
<b>id</b>	dc:identifier	
<b>localID</b>	cn:localID	
<b>name</b>	gr:legalName	
<b>payer</b>	cn:payer	
<b>noID</b>	cn:noID	
<b>country</b>	schema:addressCountry	

Tabulka 4.8: Mapování entity Nadřazená instituce

### Adresa

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Address</b>	schema:PostalAddress	
<b>streetAddress</b>	schema:streetAddress	
<b>locality</b>	schema:addressLocality	
<b>postalCode</b>	schema:postalCode	
<b>nuts</b>	cn:nuts	

Tabulka 4.9: Mapování entity Nadřazená instituce

### Objednávka

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Order</b>	cn:Order	
<b>parentDocument</b>	cn:parentDocument	
<b>subjectType</b>	cn:subjectType	
<b>parties</b>	cn:party	

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>title</b>	dc:title	
<b>amount</b>	gr:hasCurrencyValue	
<b>currency</b>	gr:hasCurrency	
<b>dateSigned</b>	dc:created	
<b>implementation</b>	cn:implementation	

Tabulka 4.10: Mapování entity Objednávka

## Faktura

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Invoice</b>	cn:Invoice	
<b>parentDocument</b>	cn:parentDocument	
<b>parties</b>	cn:party	
<b>title</b>	dc:title	
<b>amount</b>	gr:hasCurrencyValue	
<b>currency</b>	gr:hasCurrency	
<b>dateSigned</b>	dc:created	
<b>dueDate</b>	cn:dueDate	

Tabulka 4.11: Mapování entity Faktura

## Rozšiřující entity

### Implementace

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Implementation</b>	cn:Implementation	
<b>milestones</b>	cn:milestone	
<b>transactions</b>	cn:transaction	

Tabulka 4.12: Mapování entity Implementace

### Milník

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Milestone</b>	cn:Milestone	
<b>title</b>	dc:title	
<b>dueDate</b>	cn:dueDate	

Tabulka 4.13: Mapování entity Milník

## Transakce

Entita/Vlastnost	Třída/Predikát	Poznámka
<b>Transaction</b>	com:Transaction	
<b>date</b>	dc:date	
<b>amount</b>	gr:hasCurrencyValue	
<b>senderOrganization</b>	com:source	
<b>receiverOrganization</b>	com:destination	
<b>publisherId</b>	cn:publisherId	

Tabulka 4.14: Mapování entity Transakce

### 4.2.3 Publikace

K serializaci výsledné ontologie využijeme formátu Turtle. Soubor se skládá z hlavičky a definicí nově vytvořených tříd a predikátů. V hlavičce definujeme prefixy použitých ontologií a základní informace o ontologii. Použité třídy a predikáty zmíníme v poznámkách k použití (predikát *vann:usageNote*). Příklad ontologie lze nalézt v Příloze B(TODO).

## 4.3 Možnosti propojení na související data

První bezpochyby zajímavou možností je propojení smluv s veřejnými zakázkami, resp. věstníkem veřejných zakázek provozovaným Ministerstvem pro místní rozvoj. Jednotlivé smlouvy mající spojitost s veřejnou zakázkou poznáme podle vlastnosti AwardID, resp. evidenční číslo veřejné zakázky.

Dalšími zajímavými prvky k propojení jsou pokročilé informacemi o smluvních stranách. Každá smluvní strana vystupující ve smlouvě může mít zveřejněno mimo jiné identifikační číslo a adresu. Nabízí se tedy propojení s národními registry ARES a RÚIAN. ARES je registrem informací o ekonomických subjektech provozovaný Ministerstvem financí, RÚIAN je registrem územní identifikace, adres a nemovitostí provozovaný Českým úřadem zeměměřickým a katastrálním. Vhodné informace k propojení tedy jsou:

- Evidenční číslo veřejné zakázky u smlouvy k propojením s věstníkem veřejných zakázek
  - Iniciativa OpenData.cz zpracovává údaje o veřejných zakázkách v RDF podobě, využijeme tedy propojení právě s tímto datasetem
  - Cílové URL -  
<http://linked.opendata.cz/resource/domain/buyer-profiles/contract/cz/EvidencniC>
- Identifikační číslo smluvní strany s možností propojení na ARES

- OpenData.cz aktuálně zpracovávají také údaje z registru ARES
- Cílové URL - <http://linked.opendata.cz/resource/business-entity/CZICO><sup>3</sup>
- Adresa smluvní strany na RÚIAN - TODO

## 4.4 Provázání s datovým formátem JSON

V třetí kapitole jsme si ukázali, jak publikovat smlouvy v datovém formátu JSON. Vzhledem k budoucímu možnému využití v aplikacích pracujících nad smlouvami v JSON dokumentech by bylo dobré nastítnit, jak taková data rozšířit, aby se z nich stala zároveň RDF data. Cílem je ale zachovat původní strukturu JSON souboru, resp. aby data byla validní vůči JSON schématu. Pro tyto účely je ideální formát JSON-LD. Jediné, co nám stačí, je v původních datech každému objektu přiřadit *@id*, *@type* a definovat *@context* (viz první kapitola Příklad serializovaných dat ve formátu JSON-LD). Při tvorbě ontologie jsme si popsali mapování entit a položek z datového standardu na konkrétní třídy a predikáty. V kontextu je tedy přesně takovéto mapování, viz příklad kódu 4.1. Na příkladu kódu 4.2 je již vidět výsledný JSON-LD soubor s jednou smlouvou a dvěma smluvními stranami (pro porovnání, původní JSON soubor viz kód 3.1). Jedná se tedy o RDF data, která popisuje námi definovaná ontologie a zároveň jde o data splňující datový standard, resp. jsou validní vůči JSON Schématu. Hlavním přínosem je to, že RDF data serializovaná v takto definovaném JSON-LD formátu budou použitelná v budoucích aplikacích, příp. registru pracujícím nad datovým standardem.

```

1 {
2   "@context": {
3
4     "cn": "http://tiny.cc/open-contracting#",
5     "com": "https://w3id.org/commerce#",
6     "dc": "http://purl.org/dc/terms/",
7     "dcmi": "http://dublincore.org/documents/dcmi-type-vocabulary/",
8     "foaf": "http://xmlns.com/foaf/0.1/",
9     "gr": "http://purl.org/goodrelations/v1#",
10    "owl": "http://www.w3.org/2002/07/owl#",
11    "ps": "https://w3id.org/payswarm#",
12    "schema": "http://schema.org/",
13    "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
14    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
15    "xsd": "http://www.w3.org/2001/XMLSchema#",
16
17    "id" : "dc:identifier",
18    "language" : "dc:language",
19
20    "Contract" : "cn:Contract",
21    "Attachment" : "cn:Attachment",
22    "Amendment" : "cn:Amendment",
23    "Order" : "cn:Order",
24    "Invoice" : "cn:Invoice",
25    "Publisher" : "foaf:Organization",

```

<sup>3</sup>Informace skrývající se za tímto odkazem jsou sjednocením více datových zdrojů, ne pouze ARESu

```

26     "Version" : "cn:Version",
27     "Party" : "gr:BusinessEntity",
28     "SuperiorInstitution" : "gr:BusinessEntity",
29     "Address" : "schema:PostalAddress",
30     "Implementation" : "cn:Implementation",
31     "Milestone" : "cn:Milestone",
32     "Transaction" : "com:Transaction",
33
34     "documents" : { "@id": "cn:documents", "@container": "@set" },
35     "orders" : { "@id": "cn:orders", "@container": "@set" },
36     "invoices" : { "@id": "cn:invoices", "@container": "@set" },
37
38     "uri" : { "@id": "cn:uri", "@type" : "@id" },
39     "document" : { "@id": "com:contentUrl", "@type" : "@id" },
40     "valid" : { "@id": "cn:valid", "@type" : "xsd:boolean" },
41     "anonymised" : { "@id": "cn:anonymised", "@type" : "xsd:boolean"
42     },
43     "responsiblePersons" : { "@id": "cn:responsiblePerson", "
44     @container": "@set" },
45     "attachments" : { "@id": "cn:attachment", "@container": "@set"
46     },
47     "amendments" : { "@id": "cn:amendment", "@container": "@set" },
48
49     "awardID" : "cn:awardID",
50     "awardProfileID" : "cn:awardProfileID",
51     "amount" : { "@id": "gr:hasCurrencyValue", "@type" : "xsd:float"
52     },
53     "amountNoVat" : { "@id": "gr:hasCurrencyValue", "@type" : "
54     xsd:float" },
55     "title" : "dc:title",
56     "type" : { "@id": "dcmi:type", "@container": "@set" },
57     "contractType" : "cn:contractType",
58     "subjectType" : "cn:subjectType",
59     "priceAnnual" : { "@id": "cn:priceAnnual", "@type" : "
60     xsd:boolean" },
61     "currency" : "gr:hasCurrency",
62     "dateSigned" : { "@id": "dc:created", "@type" : "xsd:date" },
63     "validFrom" : { "@id": "ps:validFrom", "@type" : "xsd:date" },
64     "validUntil" : { "@id": "ps:validUntil", "@type" : "xsd:date" },
65     "funding" : "cn:funding",
66     "currentValidContract" : { "@id": "cn:currentValidContract", "
67     @type" : "@id" },
68     "description" : "dc:description",
69     "competency" : { "@id": "cn:competency", "@container": "@set" },
70     "parties" : { "@id": "cn:party", "@container": "@set" },
71
72     "localID" : { "@id": "cn:localID", "@type" : "xsd:integer" },
73     "name" : "gr:legalName",
74     "payer" : { "@id": "cn:payer", "@type" : "xsd:boolean" },
75     "noID" : { "@id": "cn:noID", "@type" : "xsd:boolean" },
76     "country" : "schema:addressCountry",
77     "paysVAT" : { "@id": "gr:valueAddedTaxIncluded", "@type" : "
78     xsd:boolean" },
79     "superiorInstitution" : "cn:superiorInstitution",
80
81     "publisher" : "dc:publisher",
82     "authentication" : "cn:authentication",

```

```

76     "address" : "schema:address",
77     "streetAddress" : "schema:streetAddress",
78     "locality" : "schema:addressLocality",
79     "postalCode" : { "@id": "schema:postalCode", "@type": "
xsd:integer" },
80     "nuts" : "cn:nuts",
81
82     "versions" : { "@id": "cn:version", "@container": "@set" },
83     "version" : { "@id": "cn:versionOrder", "@type": "xsd:integer"
},
84     "published" : { "@id": "dc:issued", "@type": "xsd:dateTime" },
85
86     "implementation" : "cn:implementation",
87     "milestones" : { "@id": "cn:milestone", "@type": "@id", "@
container": "@set" },
88     "transactions" : { "@id": "cn:transaction", "@type": "@id", "@
container": "@set" },
89     "dueDate" : { "@id": "cn:dueDate", "@type": "xsd:dateTime" },
90     "date" : { "@id": "dc:date", "@type": "xsd:dateTime" },
91     "publisherId" : "cn:publisherId",
92     "senderOrganization" : { "@id": "com:source", "@type": "@id" },
93     "receiverOrganization" : { "@id": "com:destination", "@type": "
@id" },
94
95     "contract" : { "@id": "cn:contract", "@type": "@id" },
96     "parrentDocument" : { "@id": "cn:parrentDocument", "@type": "
@id" }
97 }
98 }

```

Výpis kódu 4.1: JSON-LD Context

```

1 {
2   "@context": "http://tiny.cc/open-contracting-context",
3
4   "@id" : "http://rsmluv.cz/data/89f689cd-e784-4374-bb17-94144679
d46f",
5   "id" : "89f689cd-e784-4374-bb17-94144679d46f",
6   "published": "2014-03-25T23:20:50+01:00",
7   "language": "cs",
8
9   "documents": [
10    {
11      "@id" : "http://rsmluv.cz/contract/12345/2",
12      "@type" : "Contract",
13      "uri" : "http://rsmluv.cz/contract/12345/2",
14      "document" : "http://rsmluv.cz/file/b15a3c45-5595-4a28-b156
-4578edeb2a98/Smlouva12345.docx",
15      "type" : "Smlouva",
16      "valid" : true,
17      "anonymised" : false,
18
19      "awardID": "486026",
20      "awardProfileID": "OI-010143",
21      "amount": 584520.00,
22      "title": "Brno, Vackova, Šafaříkova – rekonstrukce kanalizace
a vodovodu",
23      "contractType": "Kupní smlouva",

```

```

24     "subjectType": "Právní, finanční překladatelské, pojišťovnické
    , poradenské a jiné služby",
25     "priceAnnual": false,
26     "currency": "CZK",
27     "dateSigned": "2011-11-16",
28     "validFrom": "2011-11-02",
29     "validUntil": "2012-06-30",
30     "competency": [ "Soukromoprávní smlouva" ],
31     "currentValidContract": "http://zakazky.bрно.cz/?pg=detail&id
    =18249&list=135",
32     "description": "Projektová dokumentace pro stavební povolení a
    zadání stavby bude řešit rekonstrukci stávající kanalizační
    stoky z profilu DN 500 na DN 800/1200 v délce 146 m, rekonstrukci
    kanalizačních přípojek pod veřejným prostranstvím a přepojení vš
    ech stávajících dešťových vpustí a také vybourání vozovek a chodn
    íků nad rýhou a zásyp rýhy recyklátem. Součástí bude inženýrsko-
    geologický průzkum, geodetické zaměření dotčené oblasti,
    inventarizace zeleně, výkaz výměr, položkový rozpočet a výkon
    autorského dozoru až do dokončení stavby. Dokumentace bude
    projednána s orgány státní správy a s účastníky stavebního řízení
    a jejich připomínky budou do dokumentace zapracovány.",
33
34     "responsiblePersons": [ "Ing. Petr Vokřál ", "Mgr. Adriana
    Krnáčová, MBA" ],
35
36     "publisher": {
37         "@id": "http://rsmluv.cz/contract/12345/2/publisher",
38         "@type": "Publisher",
39         "id": "6003508",
40         "name": "Statutární město Brno",
41         "noID": false,
42         "country": "CZE",
43         "authentication": "email"
44     },
45
46     "parties": [
47         { "@id": "http://rsmluv.cz/party/132456" },
48         { "@id": "http://rsmluv.cz/party/987654" }
49     ],
50
51     "implementation": {
52         "@id": "http://rsmluv.cz/contract/12345/2/implementation",
53         "@type": "Implementation",
54
55         "milestones": [
56             {
57                 "@id": "http://rsmluv.cz/contract/132456/2/milestone
    /5830",
58                 "@type": "Milestone",
59                 "title": "Výpověď smlouvy",
60                 "dueDate": "2012-06-20T23:20:50+01:00"
61             }
62         ],
63         "transactions": [
64             {
65                 "@id": "http://rsmluv.cz/contract/132456/2/transaction
    /132456",
66                 "@type": "Transaction",

```

```

67         "publisherId" : "1269483",
68         "date" : "2012-01-01T18:35:20+01:00",
69         "amount" : 300000,
70         "senderOrganization" : "http://rsmluv.cz/party/987654",
71         "receiverOrganization" : "http://rsmluv.cz/party/987654"
72     },
73     {
74         "@id" : "http://rsmluv.cz/contract/132456/2/transaction
/934584",
75         "@type" : "Transaction",
76         "publisherId" : "934584",
77         "date" : "2012-02-01T09:13:40+01:00",
78         "amount" : 284520,
79         "senderOrganization" : "http://rsmluv.cz/party/987654",
80         "receiverOrganization" : "http://rsmluv.cz/party/987654"
81     }
82 ]
83 },
84
85 "versions" : [
86     {
87         "version" : 1,
88         "@id" : "http://rsmluv.cz/contract/12345/1/version",
89         "@type" : "Version",
90         "uri" : "http://rsmluv.cz/contract/12345/1",
91         "published" : "2014-09-15T23:20:50+01:00"
92     },
93     {
94         "version" : 2,
95         "@id" : "http://rsmluv.cz/contract/12345/2/version",
96         "@type" : "Version",
97         "uri" : "http://rsmluv.cz/contract/12345/2",
98         "published" : "2015-03-15T14:35:28+01:00"
99     }
100 ]
101 }
102 ],
103
104 "parties" : [
105     {
106         "@id" : "http://rsmluv.cz/party/132456",
107         "@type" : "Party",
108         "localID" : 132456,
109         "id" : "44992785",
110         "name" : "Statutární město Brno",
111         "payer" : false,
112         "noID" : false,
113         "country" : "CZE",
114         "address" : {
115             "@id" : "http://rsmluv.cz/party/132456/address",
116             "@type" : "Address",
117             "streetAddress" : "Dominikánské náměstí 196/1",
118             "locality" : "Brno-město, Brno",
119             "postalCode" : 60200,
120             "nuts" : "CZ064"
121         },
122         "superiorInstitution" : {
123             "@id" : "http://rsmluv.cz/superiorInstitution/00064581",

```



```

124     "@type" : "SuperiorInstitution",
125     "id" : "00064581",
126     "localID" : 56486,
127     "name" : "Magistrát hlavního města Prahy",
128     "noID" : false,
129     "country" : "CZE"
130   }
131 },
132 {
133   "@id" : "http://rsmluv.cz/party/987654",
134   "@type" : "Party",
135   "localID" : 987654,
136   "id" : "46347011",
137   "name" : "Kovoprojekta Brno a.s.",
138   "payer" : true,
139   "noID" : false,
140   "country" : "CZE",
141   "address" : {
142     "@id" : "http://rsmluv.cz/party/987654/address",
143     "@type" : "Address",
144     "streetAddress" : "Šumavská 416/15",
145     "locality" : "Ponava, Brno",
146     "postalCode" : 60200,
147     "nuts" : "CZ064"
148   }
149 }
150 ]
151
152 }

```

Výpis kódu 4.2: JSON-LD Soubor s jednou smlouvou

## 5. Požadavky na platformu pro otevřené smlouvy

Jedním z cílů této práce je návrh a implementace platformy pro otevírání smluv veřejných institucí. Platforma bude mít tři základní části. První částí je převod interně uložených smluv v relačních databázích do otevřeného formátu splňujícího principy Linked Data. Druhou je jednotné úložiště smluv v otevřeném formátu a třetí je nad tímto úložištěm zpřístupnit otevřené smlouvy koncovým uživatelům. Shrňme si v následující části základní funkční a nefunkční požadavky na platformu:

### 5.1 Funkční požadavky

- Platforma bude umět převádět údaje o smlouvách z relačních databází veřejných institucí do otevřených dat
  - Jedná se o otevřená data, tedy dostupná on-line ke stažení/prohlížení
  - Otevřená data by měla splňovat datový standard pro otevřené smlouvy (kapitola Otevřené smlouvy)
  - Otevřená data by měla splňovat principy Linked Data (kapitola Otevřená data a principy Linked Data),  
Otevřené smlouvy jako Linked Data))
  - Výstupní formát by měl reflektovat alespoň jeden publikační formát datového standardu pro budoucí kompatibilitu
- Platforma bude otevřené smlouvy ukládat v jednotném úložišti
  - Jednotné úložiště bude reflektovat, že se jedná o Linked Data, resp. bude mít formu databáze trojic (Triplestore)
  - Data budou dostupná skrze API
- Platforma nad jednotným úložištěm zpřístupní otevřené smlouvy koncovým uživatelům formou webové aplikace
  - Aplikace by měla umět zobrazovat seznam smluv
  - Aplikace by měla umět filtrovat smlouvy podle jednotlivých veřejných institucí
  - Aplikace by měla umožňovat prohlédnout si detail veřejné instituce a její smlouvy
  - Aplikace by měla umožňovat zobrazit detail smlouvy

## 5.2 Nefunkční požadavky

- Použitelnost
  - Platforma by měla počítat s použitím v českém prostředí
  - Platforma bude obsahovat uživatelskou dokumentaci
  - Webová aplikace by měla vhodným způsobem demonstrovat výhody Linked Open Data
- Výkon
  - Konverzní mechanismus platformy by měl umět zpracovávat i větší objemy dat (řádově alespoň desetitisíce údajů v relačních databázích)
- Rozšiřitelnost a modifikovatelnost
  - Každá ze tří částí platformy by měla být snadno nahraditelná a modifikovatelná
- Integrovatelnost
  - Nasazení konverzního mechanismu u veřejných institucí by mělo být co nejjednodušší
- Bezpečnost
  - Při použití konverzního mechanismu by nemělo dojít k narušení vnitřních dat veřejných institucí

# 6. Návrh platformy pro otevřené smlouvy

## 6.1 Architektura

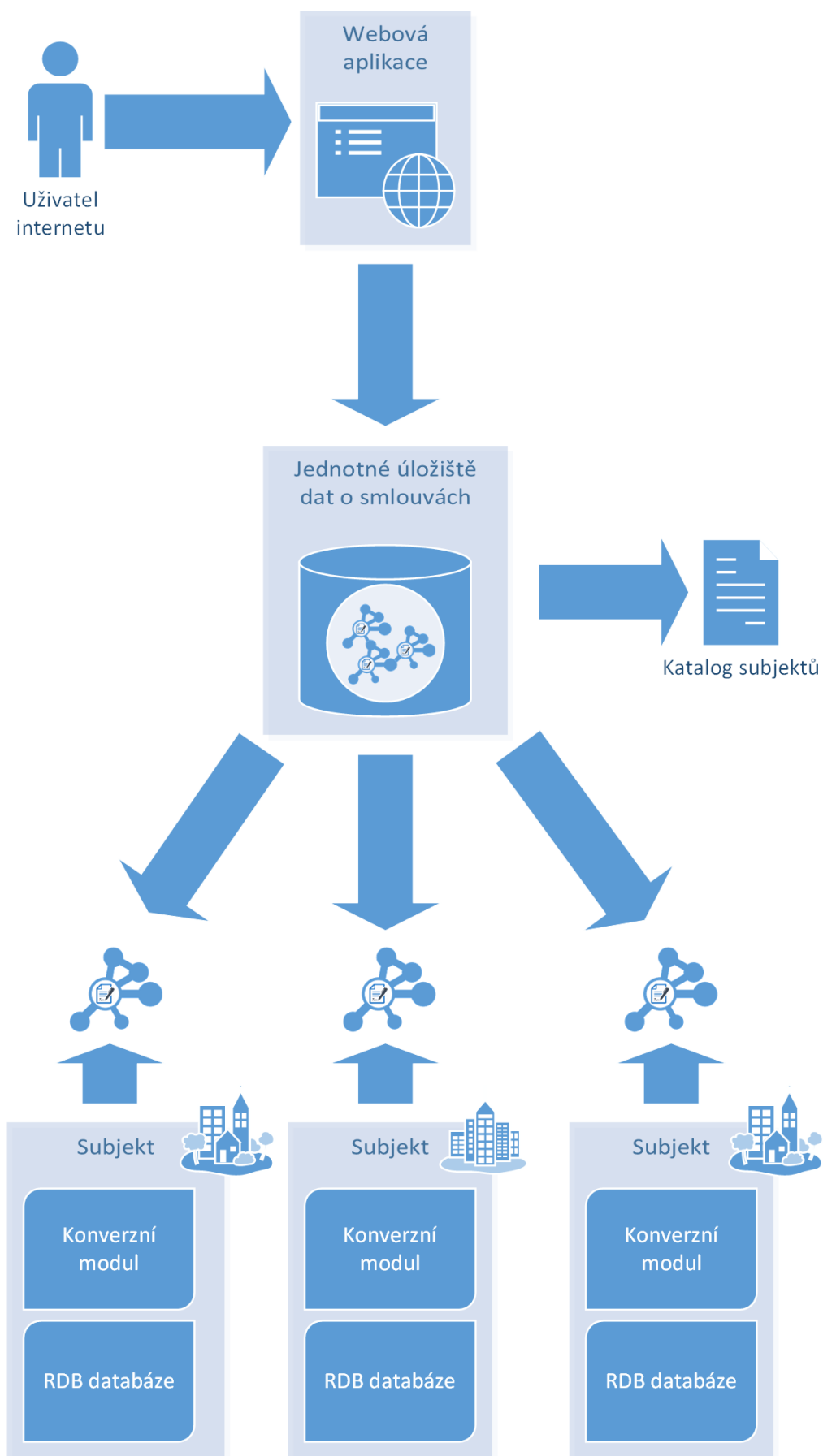
Základem pro platformu pro otevřené smlouvy jsou tři nezávislé moduly. Konverzní modul, jednotné úložiště a webová aplikace. Každá zapojená veřejná instituce bude mít svůj konverzní modul. Jednotlivé konverzní moduly budou pracovat nad konkrétními relačními databázemi jednotlivých veřejných institucí a publikovat smlouvy jako Linked Open Data. Tato data pak budou ukládána v jednotném úložišti.

Základní otázkou, kterou si je třeba položit, je způsob shromažďování dat. První možností je směr centralizovaný, resp. každý konverzní modul publikovaná data pošle do jednotného úložiště. Druhým přístupem je decentralizace, kdy konverzní moduly vystaví data na internetu a registrují přístup k datům v datovém katalogu. Jednotné úložiště si poté taková data na základě katalogu obstarává samo. Výhoda první možnosti spočívá v snadnější udržitelnosti správy nad jednotlivými instancemi veřejných institucí. Subjekty mohou v určitém intervalu posílat převedené smlouvy do úložiště bez dalších nároků na správu. V druhém přístupu jsou naopak zvýšeny nároky na jednotlivé subjekty, které vypublikované smlouvy samy musejí udržovat. Výhodou ale naopak je, že každý subjekt se stane lokálním úložištěm smluv, na které se lze odkazovat, např. na webových stránkách subjektu, vytvářet nad daty aplikace apod. První varianta je vhodná pro myšlenku centralizovaného registru, který pouze definuje podmínky, jak nahrávané smlouvy mají vypadat. Subjekty potom mají svobodu v tom, jak data převedou, pošlou, případně vypublikují. Druhá možnost naopak podporuje myšlenku, že každý subjekt zveřejňuje pouze své smlouvy a po vypublikování je umožní komukoli využít. Není proto překvapením, že myšlenky otevřených a propojených dat vyhovuje více druhý přístup.

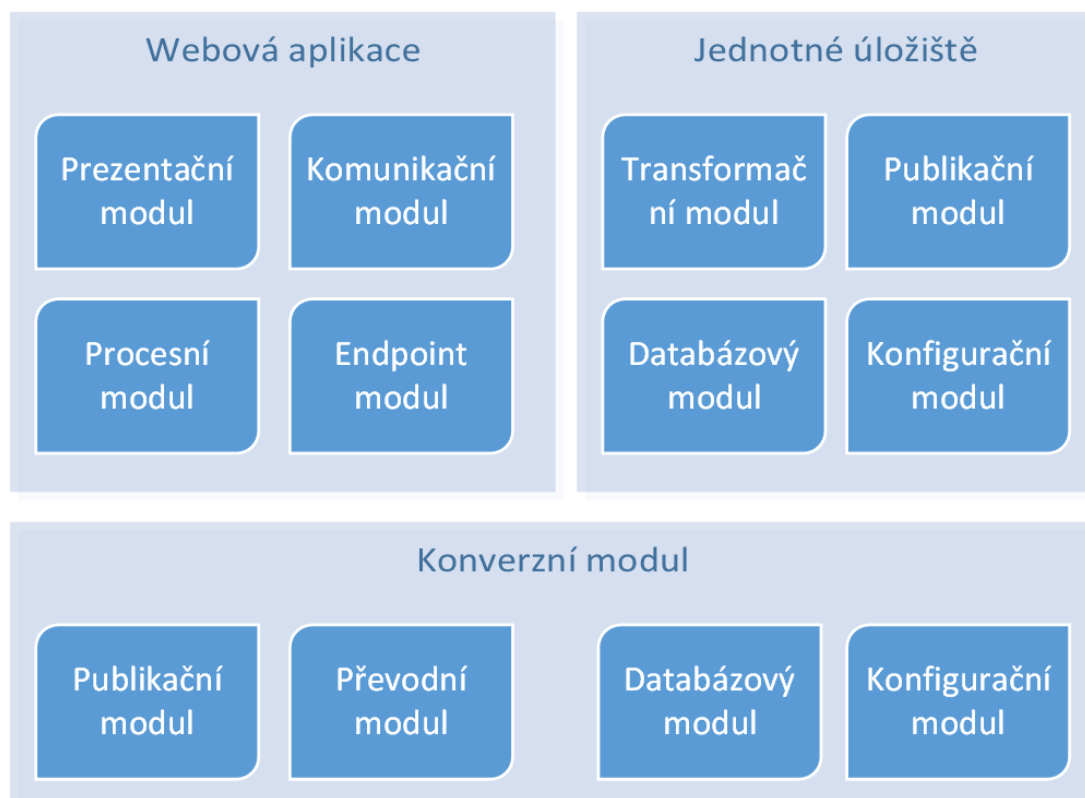
K efektivnímu naplnění druhého přístupu je důležité, že součástí platformy bude konverzní modul, který převod za subjekty řeší a celý proces otevírání smluv pomůže zautomatizovat. Částečně tak vyřešíme nevýhodu druhého přístupu a část agendy otevírání smluv přebere platforma. Na konkrétním subjektu tak zbývá pouze udržovat v chodu databázi a server (typicky webový), kde konverzní modul poběží.

Situaci můžeme dále zjednodušit tím, že subjekt zpřístupní svou databázi online. Konverzní modul tak bude moci pracovat nezávisle na subjektu a data pouze vzdáleně načítat. Narážíme zde ale na bezpečnostní limity. Databáze veřejných institucí bude typicky přístupná pouze v rámci privátní sítě. Řešením by mohlo být, že server, kde poběží konverzní modul, bude připojen do privátní sítě subjektu, druhou možností je např. klonování databáze smluv do veřejně přístupné databáze. Konkrétní řešení přístupu k databázi však necháme na konkrétních potřebách jednotlivých subjektů. Řekněme tedy, že konverzní modul platformy bude vyžadovat pouze připojení ke konkrétní databázi.

Základní pohled na platformu můžeme vidět na Obr. 6.1.



Obrázek 6.1: Základní pohled na platformu otevřených smluv (Logical view)



Obrázek 6.2: Rozdělení platformy do modulů (Decomposition view)

### 6.1.1 Konverzní mechanismus

Návrh konverzního mechanismu rozdělíme do čtyř modulů (viz Obr. 6.2). V rámci databázového modulu bude docházet ke komunikaci mezi připojenou databází a zbytkem konverzního mechanismu. V konfiguračním modulu se očekává definování základních vlastností konverzního mechanismu, převážně potřebné vstupní údaje pro převodní modul.

#### Převodní modul

Účelem tohoto modulu je převod relačních dat do RDF podoby. Nejjednodušším přístupem je manuální konverze, resp. ruční tvorba RDF výstupů. Tento přístup může být vhodný pro úzce specifické situace, avšak pro obecný přístup a možnost znovupoužití konverzního modulu vhodný není. Další možností je tvorba paralelní triplestore databáze. Relační data bychom v určitých intervalech převáděli z jedné do druhé. Tento přístup je výhodný, pokud chceme dosáhnout robustního řešení, budovat vlastní RDF úložiště, obohacovat data, přidávat další datasety apod. Klade však velké softwarové i hardwarové nároky na subjekt, vyžaduje netriviální údržbu a jedná se v podstatě o duplikovaná data z relační databáze. Poslední možností, kterou uvedeme, je tvorba wrapperu nad relační databází. Máme-li požadavek na RDF data, wrapper ho převede na ekvivalentní SQL dotaz do databáze a vrácená data převede zpět do odpovídající RDF podoby. Hlavní výhodou je, že takto uložená data jsou pouze v relační databázi a RDF data jsou tak vždy aktuální. Cílem je řešení s co nejmenší zátěží pro subjekt a s co nejlepším usnadněním znovupoužitelnosti. Ideálním řešením se tedy jeví tvorba

wrapperu.

Požadovanou funkcionalitou wrapperu je namapování datového modelu relační databáze na datový standard pro otevřené smlouvy. Mezi jazyky sloužící k popsání konkrétního mapování patří např. jazyk R2RML, nebo D2RQ. Doporučeným standardem konsorcia W3C je R2RML, zvolíme tedy jazyk R2RML.

## Publikační modul

Převedená data můžeme publikovat třemi základními způsoby:

- Dump - veškerá data jsou zpřístupněna formou stažitelného souboru serializovaného v nějakém z RDF formátů
- Dereferencovaná URI jednotlivých entit - každá entita je dostupná pod svým URI, typicky ve formě HTML stránky
- API - v kontextu RDF se typicky jedná o webovou službu ve formě SPARQL endpointu umožňující libovolné dotazování nad daty.

Naší snahou je, aby vypublikovaná data mohly využívat i jiné aplikace, než pouze jednotné úložiště v rámci platformy. K tomu je ideální API. Zvolíme proto SPARQL endpoint.

Pro naplnění principů Linked Data ale potřebujeme vyřešit dereferencování URI entit. Nad SPARQL endpointem se dereference provede jednoduše tak, že každé HTTP URI odkazující na konkrétní entitu se převede na vhodný SPARQL dotaz vracející požadovaná data.

Data budou publikována jak ve formě HTML stránky, tak v RDF formátech. Nutným základem bývá formát N-Triples a Turtle. Určíme podmínku, že pro dump je nutné umožnit i serializaci ve formátu JSON-LD z důvodu budoucí kompatibility s datovým standardem.

## Anonymizace

Typickým problémem s publikací dat je, že mohou podléhat zákonu o ochraně osobních údajů<sup>1</sup>. Některá data je proto před zveřejněním nutné anonymizovat. Proces a řešení anonymizace není předmětem této práce. Řekněme, že platforma počítá s tím, že subjekt si anonymizaci údajů vyřeší na své straně.

### 6.1.2 Jednotné úložiště

Agendou jednotného úložiště bude sbírat data vypublikovaná jednotlivými subjekty. Zapojené subjekty budeme řešit formou datového katalogu, kde budou odkazy na umístění požadovaných datasetů. Úložiště pak v definovaném intervalu stáhne datasety podle datového katalogu a uloží je do triplestore databáze.

Dekompozici do modulů je možné vidět na Obr. 6.2. Konfigurační a databázový modul má podobný význam jako v konverzním mechanismu.

---

<sup>1</sup><https://www.uoou.cz/anonymizace-osobnich-udaju/d-1764>

## Transformační modul

V prvním kroku je třeba načíst jednotlivé datasety subjektů. Odkazy na konkrétní datasety budou reprezentované ve formě datového katalogu. Samotný katalog budeme zapisovat v RDF a serializovat do formátu Turtle. Využijeme k tomu ontologii Data Catalog Vocabulary. Příklad datového katalogu lze vidět v kódu 6.1.

V dalším kroku je třeba vyřešit otázku heterogenity dat. Platforma by principiálně měla umět přijímat RDF data nejen od subjektů zpracovaná konverzním modulem, ale i jakákoli jiná RDF data splňující datový standard a reflektující definovanou RDF ontologii. Jednotlivé entity by ale měly být identifikované podle vzoru z kapitoly Otevřená smlouva jako Linked Data. To nám zaručí, že žádné dvě entity různých subjektů nebudou mít díky rozdílným doménám stejné URI.

Díky tomu tak můžeme v rámci závěrečného kroku data slít dohromady a uložit do triplestore databáze.

```
1 @prefix dcat: <http://www.w3.org/ns/dcat#> .
2 @prefix dct: <http://purl.org/dc/terms/> .
3 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
4 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
5
6 _:dataset a dcat:Dataset ;
7     dcat:distribution _:distribution1 .
8     _:distribution1 a dcat:Distribution ;
9         dct:identifier '1' ;
10        dcat:accessURL <https://raw.githubusercontent.com/
PavelHryzlik/ContractStandard/master/standard/standard/lod/
contract_sampleA.ttl> ;
11        dct:format [ rdfs:label 'Turtle' ] .
12
13 _:dataset a dcat:Dataset ;
14     dcat:distribution _:distribution2 .
15     _:distribution2 a dcat:Distribution ;
16         dct:identifier '2' ;
17        dcat:accessURL <https://raw.githubusercontent.com/
PavelHryzlik/ContractStandard/master/standard/standard/lod/
contract_sampleB.ttl> ;
18        dct:format [ rdfs:label 'Turtle' ] .
```

Výpis kódu 6.1: Datový katalog pro jednotné úložiště

## Publikační modul

Úložiště by mělo data poskytovat jak k webovému prohlížení, tak skrze API k využití v aplikacích. Tento požadavek vyřešíme vystavením SPARQL endpointu.

### 6.1.3 Propojená datová síť

Nad jednotným úložištěm může vznikat celá řada aplikací využívajících data o smlouvách. Díky propojení smluv se souvisejícími daty, tak můžeme kontext smluv rozšířit o další informace, resp. demonstrovat výhody principů Linked Data jako propojené datové sítě.

V rámci kapitoly 4 jsme definovali odkaz na veřejnou zakázku (pco:Contract) v rámci smlouvy, resp. predikát pco:publicContract ze smlouvy (třída cn:Contract)



a odkaz na ekonomický subjekt (`gr:BusinessEntity`) pomocí predikátů `owl:sameAs` ze Smluvní strany a Vydavatele (třídy `foaf:Organization` a `gr:BusinessEntity`). Nyní si položíme otázku, s jakými dalšími relevantními zdroji můžeme data na základě definovaných odkazů dále propojit.

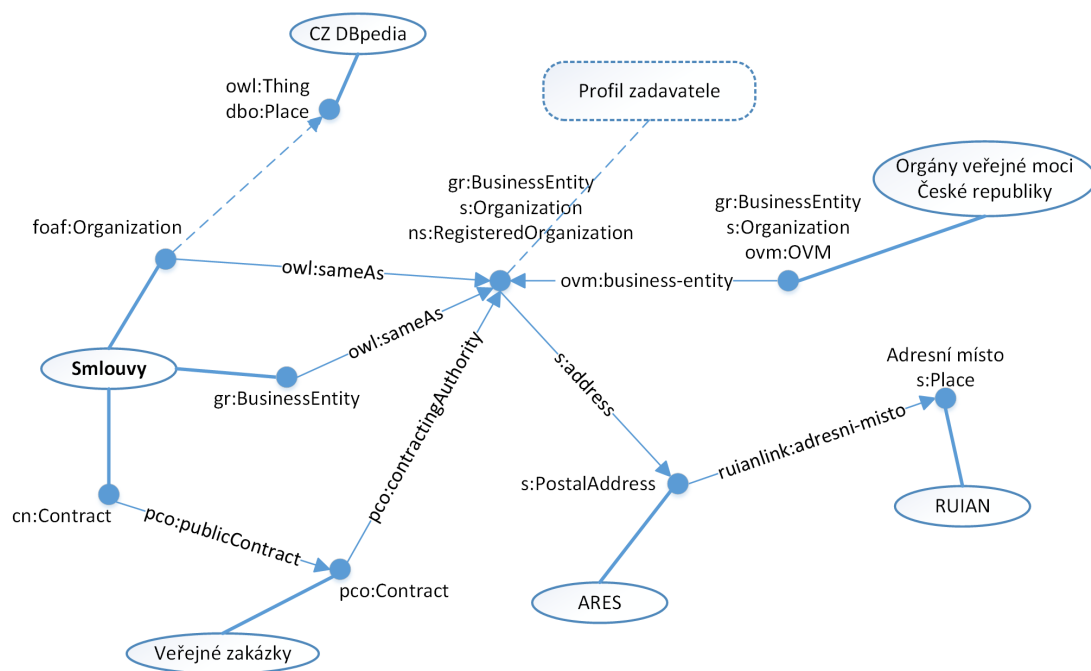
Navrženou datovou síť propojených objektů můžeme vidět na Obr. 6.3. Z každého datasetu (v oválných blocích) vede hrana k vlastnímu objektu. Z nich pak vychází vybrané predikáty odkazující na entitu dalšího datasetu.

Východním bodem je entita ekonomického subjektu (`gr:BusinessEntity`) uprostřed. Konkrétně se jedná o registrované organizace, jejichž data vychází z údajů na Profilu zadavatele, z ARESu a dalších zdrojů. Z této entity využijeme predikáty (`s:address`) k propojení s adresou (`s:PostalAddress`) z ARESu. Z adresy, pak získáme informaci o adresním místu (`s:Place`) z RUIANu pomocí predikátu `ruianlink:adresni-misto`.

Z entity ekonomického subjektu můžeme získat informaci o odpovídajícím objektu reprezentovaném v rámci orgánů veřejné moci díky zpětnému odkazu reprezentovanému predikátem `ovm:business-entity`.

Propojení dosáhneme také mezi ekonomickým subjektem a veřejnou zakázkou (`pco:Contract`). Z veřejné zakázky využijeme odkazu na zadavatele, resp. predikátu (`pco:contractAuthority`). Díky tomu můžeme zjistit údaje o veřejných zakázkách jednotlivých ekonomických subjektů, kde vystupují v roli zadavatele.

Vydavatel smluv (`foaf:Organization`) je často veřejná instituce mající svoji stránku na DBpedii. Nedisponujeme však přímým odkazem na konkrétní reprezentaci vydavatele v DBpedii. Můžeme ale položit SPARQL dotaz, zdali v DBpedii existuje instituce s daným konkrétním jménem (porovnáním predikátů `gr:legalName` vydavatele a `rdfs:label` z DBpedie). Nejedná se tedy o přímé propojení, ale o další možnost, jak kontext smluv obohatit o další informace.



Obrázek 6.3: Propojená datová síť

### 6.1.4 Webová aplikace

Úkolem webové aplikace je nad jednotným úložištěm zpřístupnit údaje o smlouvách k prohlížení koncovým uživatelům. Řešení rozdělíme do čtyř modulů (viz Obr. ??).

#### Endpoint modul

V rámci tohoto modulu definujeme napojení na požadované zdroje dat v podobě SPARQL endpointů.

#### Procesní modul

Úkolem tohoto modulu je načíst požadované údaje o smlouvách a přiřadit k nim údaje z rozšířeného kontextu navrženého v minulé kapitole. Cílem je tedy získat a sjednotit informace z několika zdrojů dat. Existuje několik přístupů, jak se nad požadovanými daty dotazovat<sup>2</sup> - z klientské části aplikace, nebo ze serverové. Z důvodu dalšího zpracování dat se jeví vhodnější serverové dotazování. Samotné dotazy můžeme pokládat také různými způsoby. Jednou z možností je distribuované dotazování, kdy se v rámci jednoho dotazu můžeme odkazovat na více zdrojů. Další možností je se nad každým zdrojem dotazovat zvlášť, resp. sadou dotazů. V souladu s principy Linked Data zvolíme sadu dotazů nad jednotlivými SPARQL endpointy. Výhodou je, že problém jednoho zdroje by neměl ovlivnit výsledky z ostatních zdrojů.

Nad SPARQL endpointy můžeme získávat data buď ve formě RDF (příkaz Construct), nebo v tabulkové formě (příkaz Select). Účelem je zobrazení dat uživateli často právě ve formě tabulek. Volba příkazu Select je tedy lepší volbou.

#### Komunikační modul

Agendou komunikačního modulu je výměna dat mezi procesním a prezentačním modulem. Modul zpracuje požadavky z prezentačního modulu a volá konkrétní funkce procesního modulu. Výsledná data pak pošle prezentačnímu modulu k zobrazení. Rozhraní pro přenos by mělo být ve standardizovaném datovém formátu<sup>3</sup>.

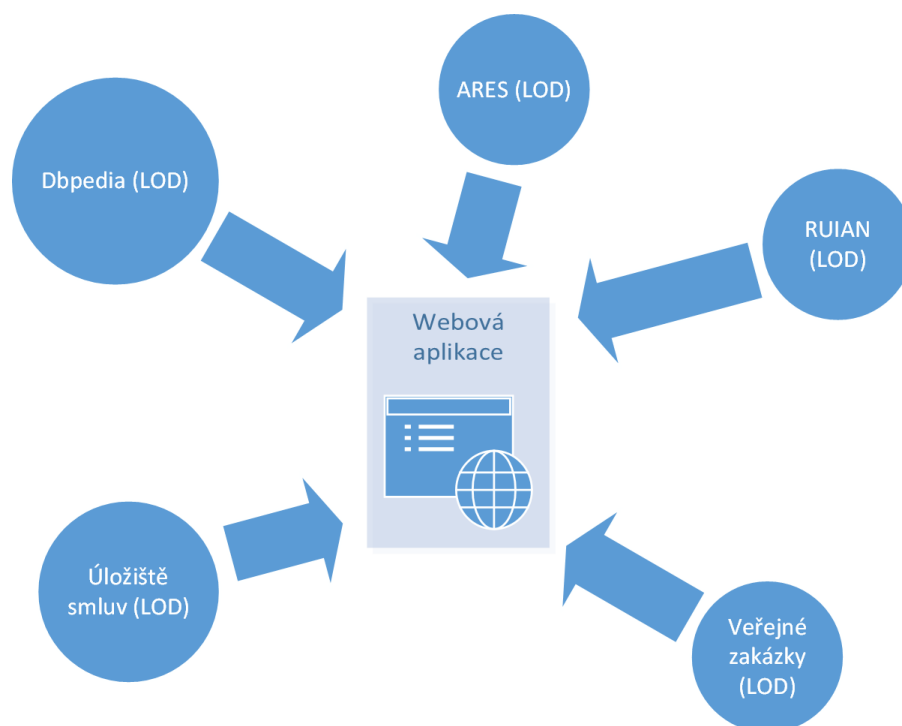
#### Prezentační modul

Účelem tohoto modulu je zobrazování dat uživateli. Výchozím bodem je nabídnout seznam smluv. Úvodní obrazovkou bude tedy pohled s úvodními informacemi a souhrnným seznamem smluv. Z každé položky půjde přejít jak do detailu konkrétní smlouvy, tak do detailu jejího vydavatele. Detail smlouvy zobrazí podrobné informace o zvolené smlouvě, včetně jejích verzí, smluvních stran, příloh, dodatků a milníků. Ze subjektů a smluvních stran majících vyplněný IČ půjde přejít na seznam veřejných zakázek, ve kterých subjekt vystupuje. Detail vydavatele nabídne podrobnější informace o zvoleném publikujícím subjektu. Součástí detailu vydavatele bude také seznam jeho smluv. Z tohoto seznamu půjde také přejít na detaily jednotlivých smluv.

---

<sup>2</sup>Předpokládáme, že aplikace bude obsahovat serverovou část.

<sup>3</sup>Typicky v XML, nebo JSON



Obrázek 6.4: Obohacený kontext smluv díky propojeným datům

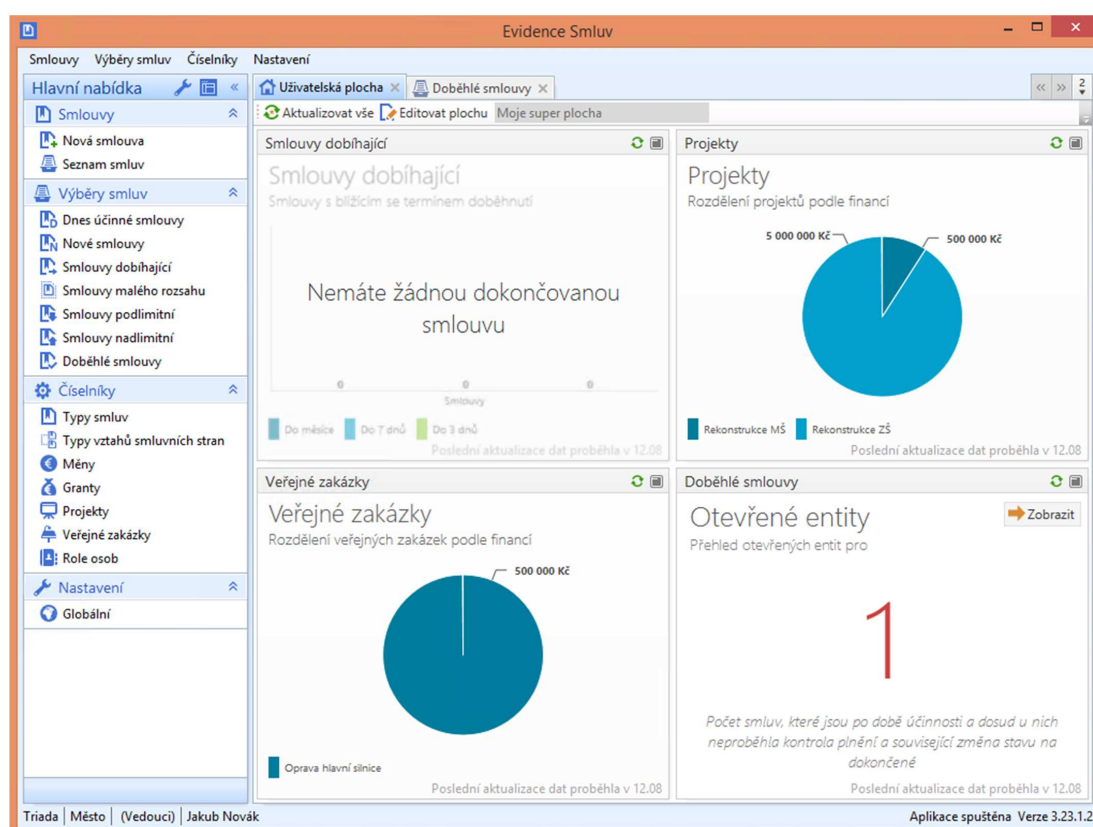
# 7. Implementace platformy

## 7.1 Konverzní mechanismus

Jako zdroj pro implementaci konverzního mechanismu byl zvolen modul Munis ESML. Jedná se o část informačního systému pro města a obce společnosti Triada spol. s.r.o.

### 7.1.1 Munis ESML

Účelem modulu Munis ESML je přehledné evidování odběratelských i dodavatelských smluv. Nabízí přehledné vyhledávání, statistiky, hlídání termínů, nebo možnost přiřadit smlouvy jednotlivým grantům, projektům či veřejným zakázkám (viz. Obr 7.1).



Obrázek 7.1: Modul ESML

## Struktura datového modelu

Základem datového modelu jsou entity *Smlouva* a *Verze smlouvy*. Smlouva je základním stavovým objektem s hierarchickou strukturou. Vycházíme z předpokladu, že dodatek ke smlouvě je také smlouva, proto definujeme:

- Entita *Smlouva* na kořenové úrovni popisuje smlouvu
- Každý syn entity *Smlouva* je jejím dodatkem

Každá smlouva je verzovaná, resp. entita *Smlouva* může mít několik *Verzí smlouvy*. Entita *Verze smlouvy* reprezentuje popisné údaje *smlouvy*. Dále obsahuje vazby na *rozdělovník*, *smluvní strany*, *milníky*, *transakce*, *externí kontakty* a *číselníky*, viz Obr. 7.2.

Každá *Verze smlouvy* může obsahovat hierarchickou strukturu příloh. Každá entita *Příloha smlouvy* reprezentuje fyzický soubor. Přílohy definujeme takto:

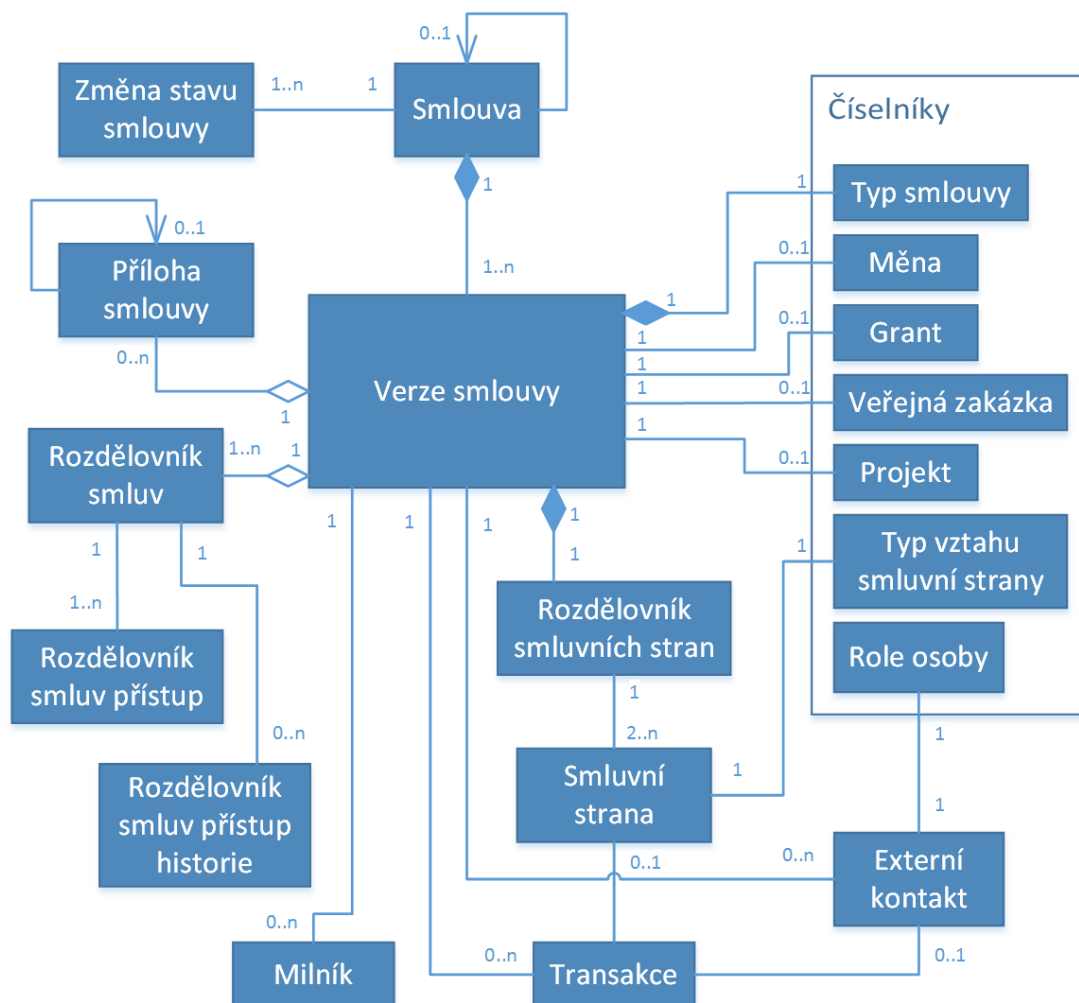
- Každá *Verze smlouvy* může mít pouze jednu kořenovou přílohu
- Kořenová příloha je hlavním dokumentem obsahujícím text smlouvy
- Ostatní jsou dílčími přílohami

Entity *Změna stavu smlouvy*, *Rozdělovník*, *Rozdělovník smluv přístup*, *Rozdělovník smluv přístup historie* nejsou pro naše účely důležité, proto je dále v textu nebudeme zmiňovat.

## Omezení vůči standardu

V porovnání s datovým standardem pro smlouvy disponuje Munis ESML několika omezeními:

- Modul nepodporuje objednávky a faktury
- Transakce nejsou implementovány - s podporou transakcí a obecně smluvního plnění se počítá do dalších verzí



Obrázek 7.2: Zjednodušený datový model (bez atributů) Munis ESML

### 7.1.2 R2RML mapování

Pomocí R2RML skriptu můžeme namapovat konkrétní sloupce z databázových tabulek na RDF predikáty. Pro složitější mapování umožňuje R2RML definovat vlastní SQL pohledy (SQL Views) nad relační databází, čehož využijeme. Pro každou entitu v rámci datového standardu proto definujeme vlastní SQL pohled. Výsledný R2RML skript lze nalézt v příloze C.

V následující části je schématicky naznačeno mapování položek. Každé entitě přiřadíme URI a typ. Následně se namapují jednotlivé položky na predikáty. Vycházíme z informací řečených v kapitole Otevřené smlouvy jako Linked Data. TODO - otázka kolekcí

#### Smlouva

- URI entity - `urlhttp://[domain]/contract/ID/PORADIVERZE`
- Typ - `cn:Contract`

Konstanty:

- Type (`dcmi:type`) - s hodnotou "Smlouva"

- PriceAnnual (cn:priceAnnual) - Nelze určit roční částku, proto vždy “false”

Nenamapované položky:

- AmountNoVat (gr:hasCurrencyValue) - cena bez dph, předpokládaná podpora spolu s podporou podrobného smluvního plnění
- SubjectType (cn:subjectType) - Číselník typů zboží/služeb, předpokládaná podpora u dalších verzí
- PlainText (cn:plainText) - Prostý text dokumentu smlouvy, resp. alternativa k oskenovaným dokumentům. Vyžaduje hlubší analýzu procesu zpracování dokumentů
- Funding (cn:funding) - Vychází zatím z nedefinovaného číselníku datového standardu

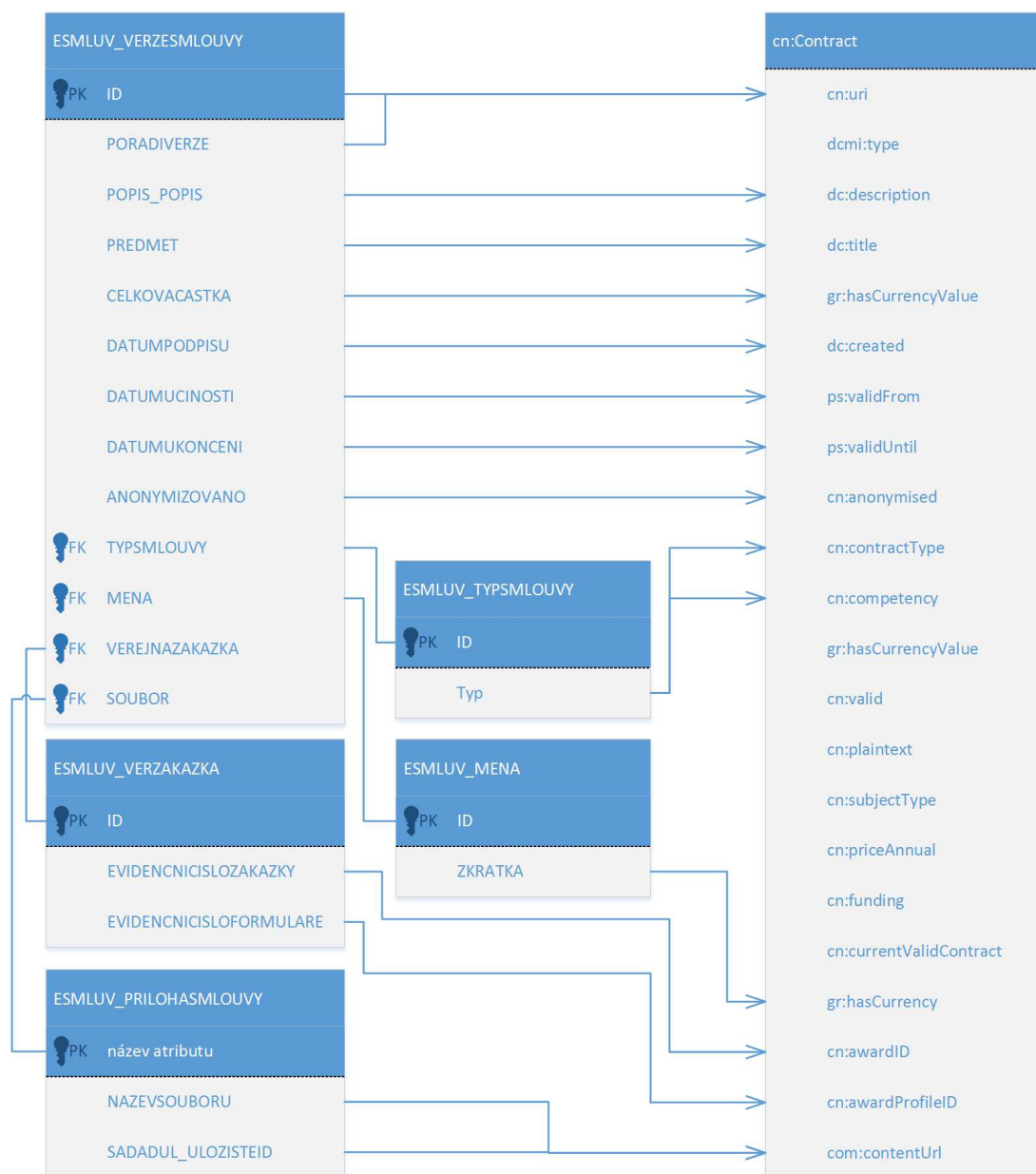
Poznámky:

- URI (cn:uri) - položka je stejná jako u URI entity
- Valid (cn:valid) - položka je “true”, jestliže se jedná o nejnovější verzi smlouvy, jinak je “false”
- Competency (cn:competency) - vyplní se na základě položky Typ u datábázové tabulky Typ smlouvy. Pokud je Typ smlouvy - “Veřejnoprávní smlouva” vyplní se i k položce Competency, jinak se vyplní “Soukromoprávní smlouva”

Mapování kolekcí a odkazů:

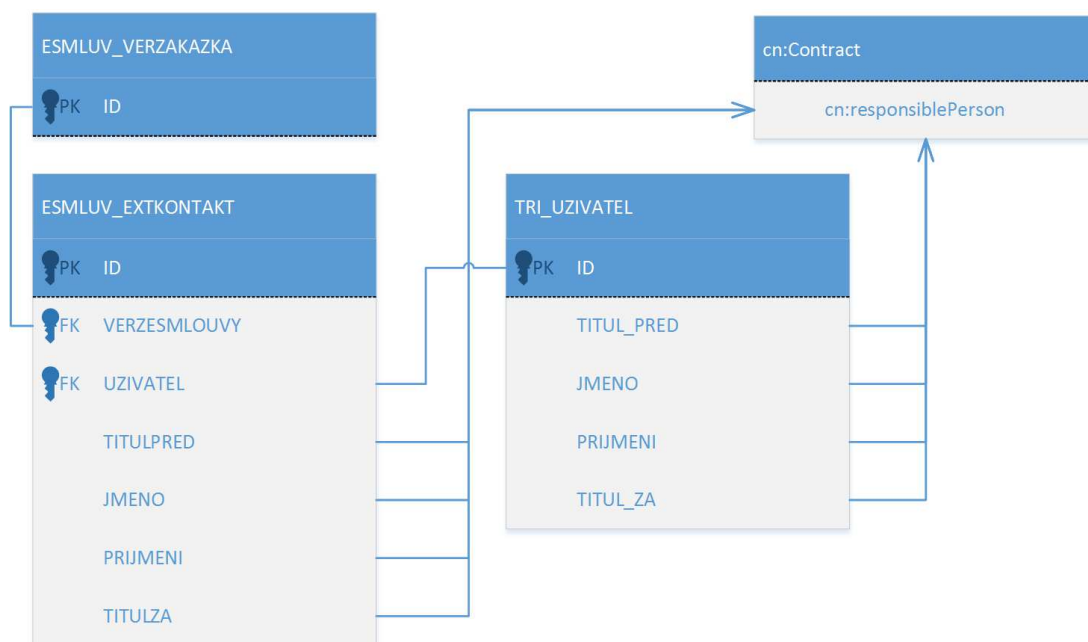
- Document (com:contentUrl) - URI odkazu -  
`http://[domain]/file/{SADADUL_ULOHISTEID}/{NAZEVSOUBORU}`
- Versions (cn:version) - URI odkazu -  
`http://[domain]/contract/{ID}/{PORADIVERZE}/version`
- Publisher (dc:publisher) - URI odkazu -  
`http://[domain]/contract/{ID}/{PORADIVERZE}/publisherTODO`
- Parties (cn:Party) - URI odkazu -  
`http://[domain]/party/{HAD_POUZITA}`
- Amendments (cn:amendment) - URI odkazu -  
`http://[domain]/amendment/{ID}/{PoradiVerzeDodatku}`
- Attachments (cn:attachment) - URI odkazu -  
`http://[domain]/attachment/{ID}/1`
- ResponsiblePersons (cn:responsiblePerson) - Každá veřejná zakázka má vazbu na externí kontakty. Externím kontaktem může být buď uživatel informačního systému (tabulka TRI\_UZIVATEL), nebo jakákoli osoba vyplněná v tabulce Externí kontakt. Pro potřeby mapování se hodnoty spojí do jednoho stringu, viz Obr. 7.3.

- Implementation (cn:implementation) - Uri odkazu - `http://[domain]/contract/{ID}/{PORADIVERZE}/implementation`



Obrázek 7.3: R2RML mapování vlastností Smlouvy





Obrázek 7.4: R2RML mapování vlastností Smlouvy (Externího kontaktu)

## Verze

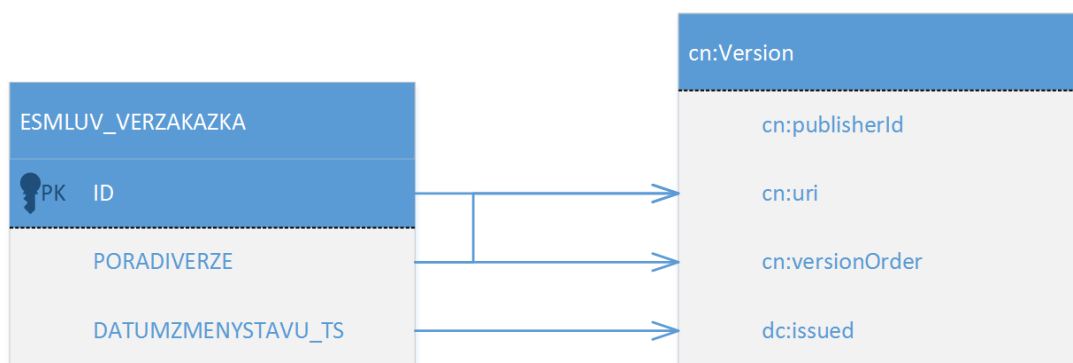
- URI entity - `http://[domain]/[type]/{ID}/{PORADIVERZE}/version`
- Typ - `cn:Version`

Nenamapované položky:

- PublisherId (`cn:publisherId`) - Díky id a verzi smlouvy máme každou entitu jednoznačně identifikovanou, proto není třeba vyplňovat

Poznámky:

- URI (`cn:uri`) - položka je stejná jako u URI entity



Obrázek 7.5: R2RML mapování vlastností Verze

## Smluvní strana

- URI - `http://[domain]/party/{HAD_POUZITA}`
- Typ strany- `gr:BusinessEntity`

Nenamapované položky:

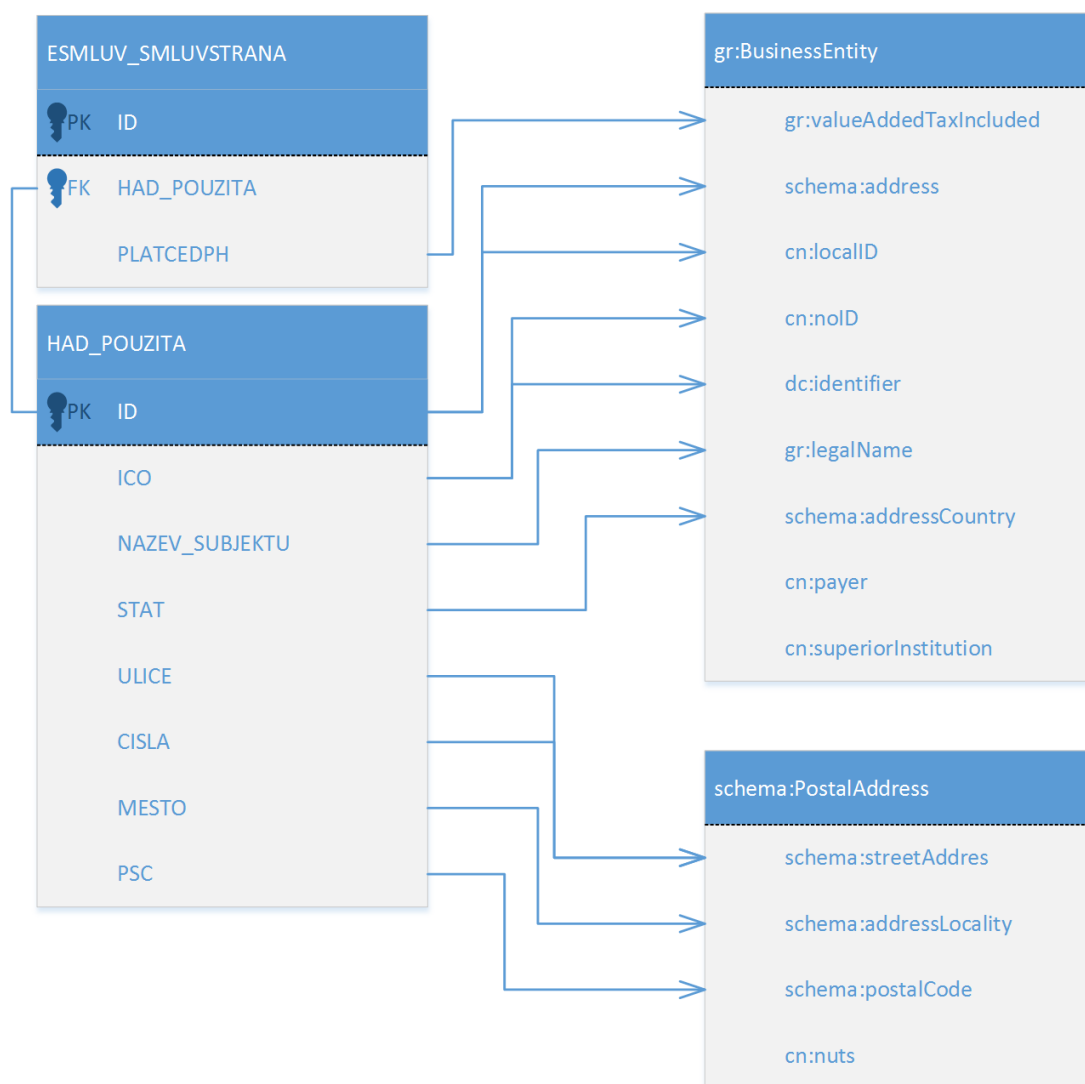
- Payer (cn:payer) - Modul ESML zatím neeviduje smluvní plnění, takže nelze určit
- SuperInstitution (cn:superiorInstitution) - Modul ESML neeviduje nadřazené instituce

## Adresa

- URI - `http://[domain]/party/{HAD_POUZITA}/address`
- Typ - `schema:PostalAddress`

Nenamapované položky:

- Nuts (cn:nuts) - Modul ESML neeviduje hodnoty normalizované klasifikace územních celků



Obrázek 7.6: R2RML mapování vlastností Smluvní strany a Adresy

## Příloha

- URI entity - `http://[domain]/attachment/{ID}/1`
- Typ - `cn:Attachment`

### Konstanty

- Type (`dcmi:type`) - s hodnotou "Příloha"
- Valid (`cn:valid`) - položka je "true", přílohy nejsou verzované

### Nenamapované položky:

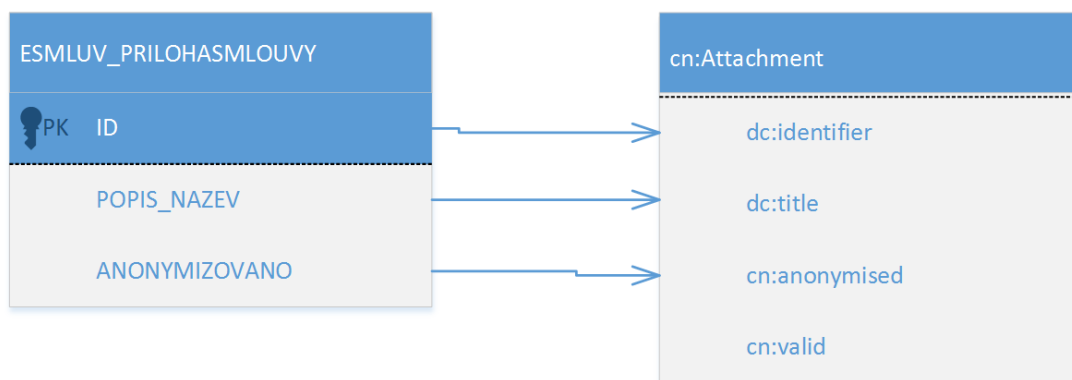
- PlainText (`cn:plainText`) - Prostý text dokumentu přílohy, resp. alternativa k oskenovaným dokumentům. Vyžaduje hlubší analýzu procesu zpracování dokumentů

### Poznámka:

- URI (cn:uri) - položka je stejná jako u URI entity

Mapování kolekcí a odkazů:

- Document (com:contentUrl) - URI odkazu -  
http://[domain]/file/SADADUL\_ULozisteID/NAZEVSouboru
- Versions (cn:version) - URI odkazu -  
http://[domain]/attachment/ID/1/version
- Publisher (dc:publisher) - URI odkazu -  
http://[domain]/attachment/ID/1/publisher
- Contract (cn:contract) - URI nadřazené smlouvy -  
http://[domain]/contract/SmlouvaID/PORADIVERZE



Obrázek 7.7: R2RML mapování vlastností Příloha

## Dodatek

- URI entity -  
http://[domain]/amendment/{ID}/{PoradiVerzeDodatku}
- Typ - cn:Amendment

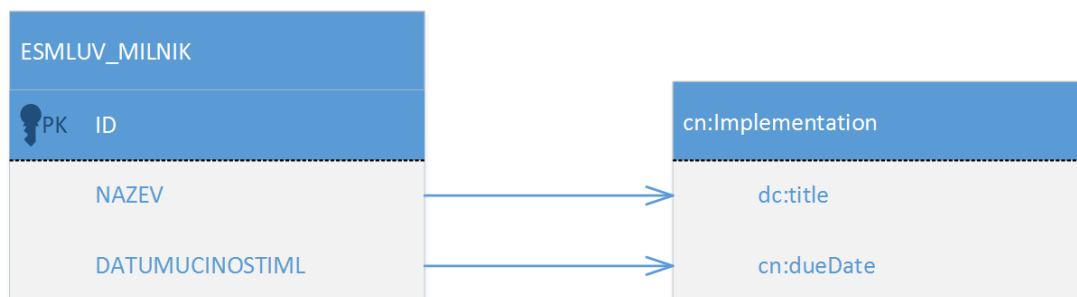
Poznámka: V rámci relační databáze je Dodatek stejná entita jako smlouva. Tudíž použijeme obdobné mapování jako u smlouvy.

## Implementace

- URI entity - http://[domain]/contract/ID/PORADIVERZE/implementation
- Typ - cn:Implementation

## Milník

- URI entity - `http://[domain]/contract/ID/PORADIVERZE/implementation/milestone/MilestoneID`
- Typ - `cn:Milestone`



Obrázek 7.8: R2RML mapování vlastností Milníku

## Vydavatel

- URI entity - `http://[domain]/publisher`
- Typ - `foaf:Organization`

Konstanty:

- Country (`schema:addressCountry`) - Hodnota “CZE”

Nenamapované položky:

- Authentication (`cn:authentication`) - Pro naše účely nemá smysl

Poznámky:

- TODO sameAs

### 7.1.3 Volba R2RML procesoru

K R2RML mapování využijeme projektu `DotNetR2RMLStore`<sup>1</sup> vytvářeném v rámci Katedry softwarového inženýrství na Matematicko-fyzikální fakultě. Jedná se o experimentální R2RML procesor pracující nad relačními databázemi Microsoft SQL<sup>2</sup>.

---

<sup>1</sup>TODO

<sup>2</sup>Většina veřejných institucí využívající produkty firmy Triada. spol, s.r.o. pracuje nad databázemi MS SQL. Proto nebereme MS SQL jako omezení. Munis ESML ale umožňuje práci i nad databází Oracle.

## Omezení R2RML procesoru

Využití zmíněného R2RML procesoru si vyžádalo několik drobných omezení.

Pro náš případ využijeme DotNetR2RMLStore ve verzi 0.0.0.8. Zkoušená vyšší verze mění logiku zpracování dotazů a zatím nepodporuje SQL Views<sup>3</sup>. Je tu i možnost vytvořit SQL views přímo nad databází subjektu a mapovat R2RML skriptem přímo tyto pohledy. Není to problém, ale nelze považovat za samozřejmost, že subjekt zpřístupní databázi k úpravám.

Procesor nepodporuje SPARQL příkazy ASK a DESCRIBE. Pro naše účely ale stačí hlavní příkazy SELECT a CONSTRUCT.

Samotný R2RML skript musí mít v hodnotě template vždy vyplněné absolutní URI. Prefixovaný zápis procesor zpracuje, ale při zpracování dotazů daný template nerozpozná.

Formáty datumových položek v RDF datech by měly splňovat W3C specifikaci, což aktuálně nesplňují. Vracené datumové hodnoty tedy v rámci postprocessingu nahradíme správným formátem<sup>4</sup>.

### 7.1.4 Volba technologií a implementační platformy

Vzhledem k tomu, že projekt DotNetR2RMLStore je implementován v prostředí .Net, tak zvolíme tuto platformu i pro implementaci konverzního mechanismu. Konverzní mechanismus bude mít formu webové aplikace, resp. virtuálního SPARQL endpointu, kterou budeme implementovat v technologii ASP.Net. Využijeme tradičního architektonického vzoru MVC. K práci s RDF daty budeme využívat knihovnu dotnetRdf.

### 7.1.5 Napojení na datové úložiště

Mezi specifika informačních systémů firmy Triada s.r.o. můžeme zmínit, že neukládají fyzické soubory (v našem případě smlouvy) do databáze s ostatními daty, ale do specializovaného datového úložiště. Nutnou podmínkou pro zobrazení těchto dat je proto propojení konverzního modulu s databází datového úložiště. Využijeme k tomu knihovnu TriadaModulZaklad.

V relační databázi jsou uloženy informace o daném souboru. Jedná se mimo jiné o název souboru a jeho jednoznačný identifikátor v datovém úložišti ve formě GUID. Informace tedy namapujeme již na zmíněné URI -

`http://[domain]/file/{SADADUL_ULOZISTEID}/{NAZEVSOUBORU}`. Při přístupu na danou adresu se informace převedou na dotaz do datového úložiště a uživateli se vrátí konkrétní soubor ke stažení.

### 7.1.6 SPARQL endpoint

### 7.1.7 Zpracování RDF výstupu

Příkazy SELECT jsou zpracovávány proudově v tabulkové formě. Seznam definovaných proměnných a výčet hodnot, které jim odpovídají. Definujeme proto handler naslouchající nad R2RML procesorem, kterým výsledky dotazu postupně

---

<sup>3</sup>TODO

<sup>4</sup><http://www.w3.org/TR/NOTE-datetime>

zpracováváme. Pro každý výstupní formát proto implementujeme handler serializující výsledky do zvoleného datového formátu. Aplikace podporuje základní formáty jako HTML, Turtle, N-Triples, RDF/XML, XML, JSON a CSV. Formáty XML, JSON, CSV serializujeme podle doporučení W3C<sup>5</sup>.

Výhodou proudového přístupu je možnost zpracování teoreticky neomezeného množství dat s dobou zpracování lineárně závisající na daném vstupu.

Příkazem CONSTRUCT získáme na výstupu RDF graf ve formě trojic. Dotaz můžeme zpracovávat jak proudově, tak v paměti. Nevýhodou proudového zpracování je, že nám výsledky přicházejí postupně, data proto lze jen obtížně zkracovat pomocí prefixů, sdružovat související informace apod. V druhém případě máme výsledek uložený v interní reprezentaci jako RDF graf. Graf tedy můžeme procházet a formátovat libovolným způsobem. Nevýhodou jsou vysoké paměťové nároky. Aplikace v obou případech podporuje také serializaci do formátů HTML, Turtle, N-Triples, RDF/XML, XML, JSON a CSV. Výstup HTML také slouží k prohlížení dat. Jednotlivé URI jsou ve formě hypertextových odkazů, lze tedy procházet mezi provázanými entitami.

Možnost DUMPU je v podstatě pouze příkaz CONSTRUCT nad všemi daty.

## Zpracování JSON-LD

Zvláštní kapitolou je formát JSON-LD. Tento formát není určen pro dotazování, ale spíše na zpracování výsledného grafu. Zavedeme ho tedy jako další možnost zpracování DUMPU dat. Ke zpracování RDF dat potřebujeme načíst definovaný JSON-LD Context, provést mapování nad RDF daty a následně strukturu upravit tak, aby byla validní vůči JSON schématu datového standardu. K mapování využijeme knihovnu JSON-LD.Net. Knihovna však nereflktuje JSON datové typy, všechny hodnotové typy jsou String. Výsledek by tak nebyl validní vůči JSON schématu. Lehce tedy knihovnu upravíme, aby vracela požadované datové typy (viz kód 7.1).

```
1 // Convert value to corresponding type (reflects JSON Schema types)
2 // e.g. DateTime is string in JSON Schema with dateTime formatting
3 switch (((JValue) value["@type"]).Value.ToString())
4 {
5     // JSON Schema type - boolean
6     case XmlSpecsHelper.XmlSchemaDataTypeBoolean:
7         bool boolValue;
8         if (bool.TryParse(stringValue, out boolValue))
9         {
10             return new JValue(boolValue);
11         }
12         break;
13     // JSON Schema type - integer
14     case XmlSpecsHelper.XmlSchemaDataTypeInteger:
15         int integerValue;
16         if (int.TryParse(stringValue, out integerValue))
17         {
18             return new JValue(integerValue);
19         }
20         break;
21     // JSON Schema type - number
```

<sup>5</sup><http://www.w3.org/TR/sparql11-overview/#sparql11-results>

```

22     case XmlSpecsHelper.XmlSchemaDataTypeFloat:
23         float floatValue;
24         if (float.TryParse(stringValue, out floatValue))
25         {
26             return new JValue(floatValue);
27         }
28         break;
29     }
30     return new JValue(stringValue);

```

Výpis kódu 7.1: Rozšíření knihovny JSON-LD.Net

## 7.2 Jednotné úložiště

K sběru a zpracování dat využijeme nástroje Unified views. Jedná se o nástroj na jehož vývoji spolupracuje katedra softwarového inženýrství na MFF UK v rámci evropského projektu LOD2<sup>6</sup>. .

### 7.2.1 Nástroj Unified views

Nástroj Unified views funguje na bázi zřetězeného zpracování (Pipelining) spojených funkčních jednotek (DPU - Data processing unit), viz Obr. 7.9

V první fázi stáhneme data jednotlivých subjektů z datového katalogu (E-FilesDownload DPU, T-FilesToRdf DPU). V druhé fázi nad daty provedeme tuto operaci:

- Obecně subjekt publikující smlouvy nutně nemusí mít podrobné informace o smluvních stranách, ale např. jen IČ. Za předpokladu, že u smluvních stran je vyplněno jen IČ, tak vytvoříme propojení na odpovídající objekt v Linked Data reprezentaci ekonomického subjektu.

V třetí fázi definujeme Metadata o celé datové sadě reprezentující smlouvy. Popíšeme, k čemu datová sada slouží, jaké má URI, licence apod. V poslední fázi data i metadata publikujeme do triplestore databáze Virtuoso Universal Server<sup>7</sup>. Informace o datové sadě zároveň zveřejníme v rejstříku datových sad. V našem případě nad platformou CKAN<sup>8</sup>.

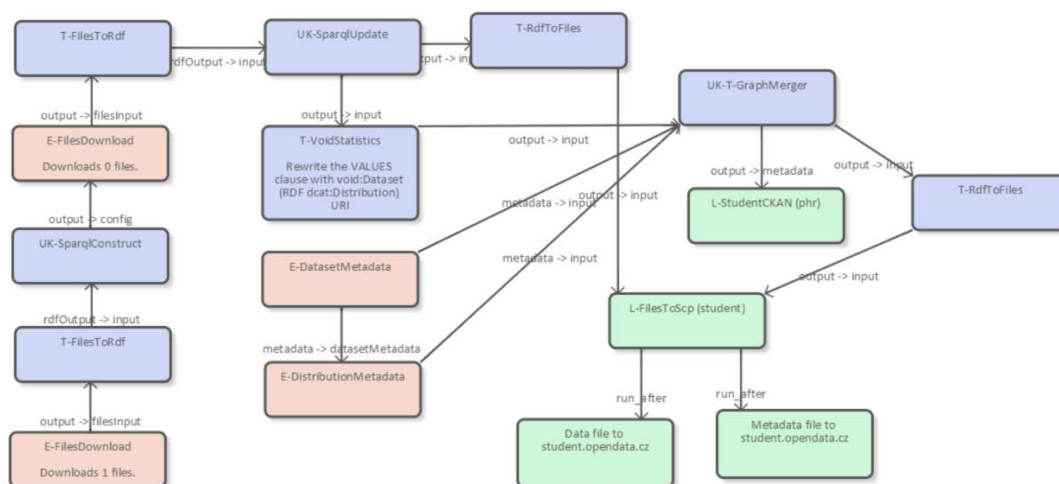
---

<sup>6</sup><http://lod2.eu/Welcome.html>

<sup>7</sup><http://virtuoso.openlinksw.com/>

<sup>8</sup> <http://ckan.org/>





Obrázek 7.9: Pipeline nad jednotným úložištěm pro zpracování dat o smlouvách

## 7.3 Webová aplikace

### 7.3.1 Volba technologií a implementační platformy

Webová aplikace je implementována také v technologii ASP.Net se zvoleným architektonickým vzorem MVC. K práci s RDF daty využijeme také knihovnu `dotnetRdf`. Layout aplikace je tvořen formou responzivního Bootstrap<sup>9</sup> designu.

### 7.3.2 Získávání dat

V rámci aplikace využíváme přístup k datovým sadám z těchto SPARQL endpointů:

- Smlouvy - <http://student.opendata.cz/sparql>
- Organizace, ARES, Orgány veřejné moci - <http://linked.opendata.cz/sparql>
- RUIAN - <http://ruian.linked.opendata.cz/sparql>
- DBpedia - <http://dbpedia.org/sparql>, nebo česká verze <http://cs.dbpedia.org/sparql>

Aplikace se skládá ze čtyř pohledů, Seznam mluv, Detail subjektu, Detail smlouvy, Veřejné zakázky subjektu. Konkrétní data se získají pomocí těchto SPARQL dotazů:

#### Seznam smluv

TODO - oprava publisher!

- `PREFIX cn: <http://tiny.cc/open-contracting#>`
- `PREFIX dc: <http://purl.org/dc/terms/>`
- `PREFIX gr: <http://purl.org/goodrelations/v1#>`
- `PREFIX ps: <https://w3id.org/payswarm#>`

<sup>9</sup>TODO

```

5
6 SELECT ?Uri ?Publisher ?Title ?ContractType ?DateSigned ?ValidFrom ?
   Amount
7 WHERE
8 {
9     ?Uri a <http://tiny.cc/open-contracting#Contract> ;
10         dc:title ?Title ;
11         cn:contractType ?ContractType ;
12         dc:created ?DateSigned ;
13         ps:validFrom ?ValidFrom ;
14         gr:hasCurrencyValue ?Amount .
15
16     BIND(REPLACE(str(?Uri), 'contract.*$', 'publisher') AS ?
publisherStr)
17     BIND(URI(?publisherStr) AS ?pub )
18
19     ?pub gr:legalName ?Publisher .
20 }

```

Výpis kódu 7.2: Získej všechny smlouvy

## Detail subjektu

TODO - oprava publisherů!

```

1 PREFIX dc: <http://purl.org/dc/terms/>
2 PREFIX gr: <http://purl.org/goodrelations/v1#>
3 PREFIX owl: <http://www.w3.org/2002/07/owl#>
4
5 SELECT ?publisher ?ic ?aresLink
6 WHERE
7 {
8     ?publisher dc:identifier ?ic ;
9               gr:legalName @publisher .
10
11     OPTIONAL
12     {
13         ?publisher owl:sameAs ?aresLink .
14     }
15 }

```

Výpis kódu 7.3: Získej vydavatele na základě jména

TODO - oprava publisherů!

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3 PREFIX gr: <http://purl.org/goodrelations/v1#>
4 PREFIX ps: <https://w3id.org/payswarm#>
5
6 SELECT ?Uri ?Title ?ContractType ?DateSigned ?ValidFrom ?Amount
7 WHERE
8 {
9     ?Uri a <http://tiny.cc/open-contracting#Contract> ;
10         dc:title ?Title ;
11         cn:contractType ?ContractType ;
12         dc:created ?DateSigned ;
13         ps:validFrom ?ValidFrom ;
14         gr:hasCurrencyValue ?Amount .

```

```

15
16 BIND(REPLACE(str(?Uri), 'contract.*$', 'publisher') AS ?
publisherStr)
17 BIND(URI(?publisherStr) AS ?pub )
18
19 ?pub gr:legalName @publisher .
20 }

```

Výpis kódu 7.4: Získej všechny smlouvy daného vydavatele úložiště

```

1 PREFIX gr: <http://purl.org/goodrelations/v1#>
2 PREFIX schema: <http://schema.org/>
3
4 SELECT ?localPlace ?streetAddress ?postalCode ?dayOfWeek ?open ?
close
5 WHERE
6 {
7     ?subjekt <http://linked.opendata.cz/ontology/domain/seznam.gov.
cz/ovm/business-entity> @businessEntity ;
8         gr:hasPOS ?localPlace .
9
10    ?localPlace s:address ?address ;
11                gr:openingHoursSpecification ?openingHours .
12
13    ?address s:streetAddress ?streetAddress ;
14            s:postalCode ?postalCode .
15
16    ?openingHours gr:hasOpeningHoursDayOfWeek ?dayOfWeek ;
17                gr:opens ?open ;
18                gr:closes ?close .
19 }

```

Výpis kódu 7.5: Získej otevírací hodiny subjektu

```

1 PREFIX gr: <http://purl.org/goodrelations/v1#>
2 PREFIX schema: <http://schema.org/>
3
4 SELECT *
5 WHERE
6 {
7     @businessEntity s:address ?address .
8
9     ?address ruianlink:adresni-misto ?ruianLink .
10
11    FILTER(CONTAINS(str(?address), 'ares'))
12 }

```

Výpis kódu 7.6: Získej adresní místo z RUIANu

```

1 PREFIX gr: <http://purl.org/goodrelations/v1#>
2 PREFIX schema: <http://schema.org/>
3 PREFIX ruian: <http://ruian.linked.opendata.cz/ontology/>
4
5 SELECT ?longitude ?latitude
6 WHERE
7 {
8     @addressPoint ruian:adresniBod ?addressPoint .
9
10    ?addressPoint s:geo ?geoCoordinates.

```

```

11
12     ?geoCoordinates s:longitude ?longitude ;
13                     s:latitude ?latitude .
14 }

```

Výpis kódu 7.7: Získej polohu subjektu

```

1 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2 PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
3
4 SELECT DISTINCT ?city ?img
5 WHERE {
6     ?city rdfs:label @publisher@cs ;
7         dbpedia-owl:thumbnail ?img .
8 }

```

Výpis kódu 7.8: Získej foto subjektu

## Detail smlouvy

```

1 SELECT * WHERE { @contract ?p ?o }

```

Výpis kódu 7.9: Získej smlouvu

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3 PREFIX gr: <http://purl.org/goodrelations/v1#>
4 PREFIX schema: <http://schema.org/>
5
6 SELECT ?ID ?Uri ?Name ?Country ?PaysVAT ?StreetAddress ?Locality ?
   PostalCode
7 WHERE
8 {
9     @contract cn:party ?Uri .
10    ?Uri a <http://purl.org/goodrelations/v1#BusinessEntity> ;
11         gr:legalName ?Name ;
12         schema:addressCountry ?Country ;
13         schema:address ?Address ;
14         gr:valueAddedTaxIncluded ?PaysVAT .
15
16    OPTIONAL { ?Uri dc:identifier ?ID }
17
18    ?Address a <http://schema.org/PostalAddress> ;
19             schema:streetAddress ?StreetAddress ;
20             schema:addressLocality ?Locality ;
21             schema:postalCode ?PostalCode .
22 }

```

Výpis kódu 7.10: Získej smluvní strany na základě smlouvy

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX com: <https://w3id.org/commerce#>
3 PREFIX dc: <http://purl.org/dc/terms/>
4
5 SELECT ?Uri ?Title ?Document
6 WHERE
7 {
8     ?Uri a <http://tiny.cc/open-contracting#Attachment> ;

```

```

9         dc:title ?Title ;
10        com:contentUrl ?Document ;
11        cn:contract @contract .
12    }

```

Výpis kódu 7.11: Získej přílohy na základě smlouvy

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX com: <https://w3id.org/commerce#>
3 PREFIX dc: <http://purl.org/dc/terms/>
4
5 SELECT ?Uri ?Title ?DateSigned ?Document
6 WHERE
7 {
8     ?Uri a <http://tiny.cc/open-contracting#Amendment> ;
9         dc:title ?Title ;
10        dc:created ?DateSigned ;
11        com:contentUrl ?Document ;
12        cn:contract @contract .
13 }

```

Výpis kódu 7.12: Získej dodatky na základě smlouvy

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3
4 SELECT ?Uri ?Title ?DueDate
5 WHERE
6 {
7     @contract cn:implementation ?Implementation .
8
9     ?Implementation a <http://tiny.cc/open-contracting#
Implementation> ;
10        cn:milestone ?Uri .
11
12     ?Uri a <http://tiny.cc/open-contracting#Milestone> ;
13        dc:title ?Title ;
14        cn:dueDate ?DueDate .
15 }

```

Výpis kódu 7.13: Získej milníky na základě smlouvy

```

1 PREFIX cn: <http://tiny.cc/open-contracting#>
2 PREFIX dc: <http://purl.org/dc/terms/>
3
4 SELECT ?Uri ?Issued ?ContractUri ?VersionOrder
5 WHERE
6 {
7     @contract cn:version ?Uri .
8
9     ?Uri dc:issued ?Issued ;
10        cn:uri ?ContractUri ;
11        cn:versionOrder ?VersionOrder .
12 }

```

Výpis kódu 7.14: Získej verzi smlouvy

**Veřejné zakázky subjektu**

```

1 PREFIX dc: <http://purl.org/dc/terms/>
2 PREFIX gr: <http://purl.org/goodrelations/v1#>
3 PREFIX pc: <http://purl.org/procurement/public-contracts#>
4 PREFIX pccz: <http://purl.org/procurement/public-contracts-czech#>
5 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
6
7 SELECT DISTINCT ?Uri ?KodProfil ?Title ?SupplierUri ?ID ?Amount ?Vat
8 WHERE
9 {
10   ?Uri <http://purl.org/procurement/public-contracts#
contractingAuthority> @businessEntity ;
11     dc:title ?Title .
12
13   OPTIONAL
14   {
15     ?Uri pccz:kodprofil ?KodProfil ;
16         pco:awardedTender ?Tender .
17     ?Tender pco:offeredPrice ?PriceSpec ;
18         pco:supplier ?SupplierUri .
19     ?SupplierUri gr:legalName ?Supplier .
20
21     BIND(CONCAT(str(?SupplierUri), '/identifier') as ?IcStr)
22     BIND(URI(?IcStr) as ?IcUri)
23
24     ?IcUri skos:notation ?ID .
25     ?PriceSpec gr:hasCurrencyValue ?Amount ;
26         gr:valueAddedTaxIncluded 1 .
27   }
28 }

```

Výpis kódu 7.15: Získej veřejné zakázky na základě subjektu

## 8. Evaluace

V rámci této kapitoly se zaměříme na otestování konverzního mechanismu platformy nad daty nastiňující reálnou situaci na úřadech.

V únoru roku 2015 vydalo Ministerstvo vnitra dopadovou studii na odhad nákladů k zavedení zákona o registru smluv<sup>1</sup>. V reakci na tento odhad nedlouho poté vydalo Cetrum aplikované ekonomie o.s. stínový výpočet korigující výsledky Ministerstva vnitra<sup>2</sup>. Na základě těchto studií můžeme získat hrubou představu o tom, kolik jednotlivé subjekty cca uzavírají nedlouho poté smluv. Veřejné instituce tak rozdělíme do čtyř kategorií:

- Malé - Nejmenší instituce, uzavírající jednotky smluv měsíčně s celkovým úhrnem maximálně několika desítek smluv ročně (v rámci měst a obcí jde o nejvyšší zastoupení).
- Střední - Subjekty generující maximálně desítky smluv měsíčně, s jednotkami stovek smluv ročně (v rámci všech subjektů pravděpodobně nevýznamnější zastoupení).
- Středně velké - Instituce, které produkují desítky, až stovky smluv měsíčně s jednotkami tisíců smluv ročně.
- Velké - Velké instituce se stovkami až tisíci smluv měsíčně s roční produkcí tisíců až desetitisíců smluv.

Pro simulaci prostředí jednotlivých kategorií vytvoříme pro každou skupinu testovací relační databázi s desítkami, stovkami, tisíci a desetitisíci smluv. Nad každou databází spustíme konverzní modul a změříme dva pravděpodobně nejčastější požadavky - dump dat, resp. výčet všech smluv a vyhledání jedné konkrétní smlouvy. Dump je základní funkcionalitou k vypublikování otevřených smluv. Potřebujeme ho také v rámci platformy, resp. jednotného úložiště, které dílčí dumpy stahuje. Ukázka vyhledání jedné smlouvy slouží spíše k ukázce, že konverzní modul půjde využít i mimo platformu, např. v rámci webových stránek konkrétní veřejné instituce.

Pro generování dat v SQL databázi byl zvolen nástroj Sql Data Generator<sup>3</sup>. Tento nástroj umožňuje nastavení nejen počtu vygenerovaných dat, ale i např. procentuální zastoupení propojení tabulek nebo šablony pro konkrétní hodnoty v jednotlivých sloupcích. Umožní nám přiblížit se k reálnému obsahu databází veřejných institucí<sup>4</sup>. K samotnému profilingu využijeme klasických prostředků prostředí .Net. Změříme dobu od přijetí požadavku po jeho kompletní zpracování.

Měření probíhalo na sestavě:

---

<sup>1</sup><http://www.janfarsky.cz/wp-content/uploads/2015/05/Dopadov%C3%A1-studie-ke-KPN-k-registru-smluv-PRACOVNI-VERZE-27-02-2015-1.pdf>

<sup>2</sup> <http://www.rekonstrukcestatu.cz/publikace/2015-03-04-stinovy-vypocet-ria-k-registru-smluv.pdf>

<sup>3</sup>TODO - odkaz na zdroj.

<sup>4</sup>Pro představu je příklad XML scriptu přiložen na datovém nosiči. Sql Data Generator je ale komerční nástroj, který neumožňuje zobrazit generovaný sql příkaz.

- Intel Core i5-4200U, CPU @ 1.60GHz - 2.30GHz
- 4GB RAM
- 64bit operační systém
- Databáze - MS SQL 2014

Pro každou kategorii bylo provedeno 15 měření pro dump, resp. vyhledání smlouvy. Z každé sady výsledků se odebrala minimální a maximální hodnota a následně ze zbývajících hodnot byl vypočítán průměr. Pro názornost, u dumpu uvádíme také velikost výstupních dat a počet vygenerovaných trojic. Výsledky lze najít v tabulce viz XX a v následujících grafech.

Z výsledků lze konstatovat, že výkon klesá cca lineárně s množstvím dat. Můžeme říci, že konverzní modul je schopen poskytovat základní funkcionalitu v rozumném čase u menších, středních i středně velkých institucí. U velkých institucí už výsledky nejsou ideální. Institucí publikujících takové množství smluv je ale v prostředí České republiky velmi málo. Příslíbem je také to, že využívaný R2RML procesor podléhá soustavnému vývoji a do budoucna lze očekávat výrazné zrychlení.



## 9. Linked Data v procesu otevírání smluv

Za účelem větší názornosti shrneme dosavadní proces otevírání smluv jedním diagramem (viz Obr. 9.1). Diagram je rozdělen do tří řádků a tří sloupců. První řádek zachycuje proces otevírání dat. Druhý řádek znázorňuje otevírání dat s využitím principů Linked Data. Třetím řádkem je zapojení relačních dat do procesu. V prvním sloupci se nacházíme na úrovni schématu. Zde definujeme standardy, ontologie, schémata. Druhý sloupec znázorňuje produkci otevřených a propojitelných dat. V třetím sloupci jsme na úrovni publikace dat, resp. serializace otevřených a propojitelných dat do přenositelných formátů.

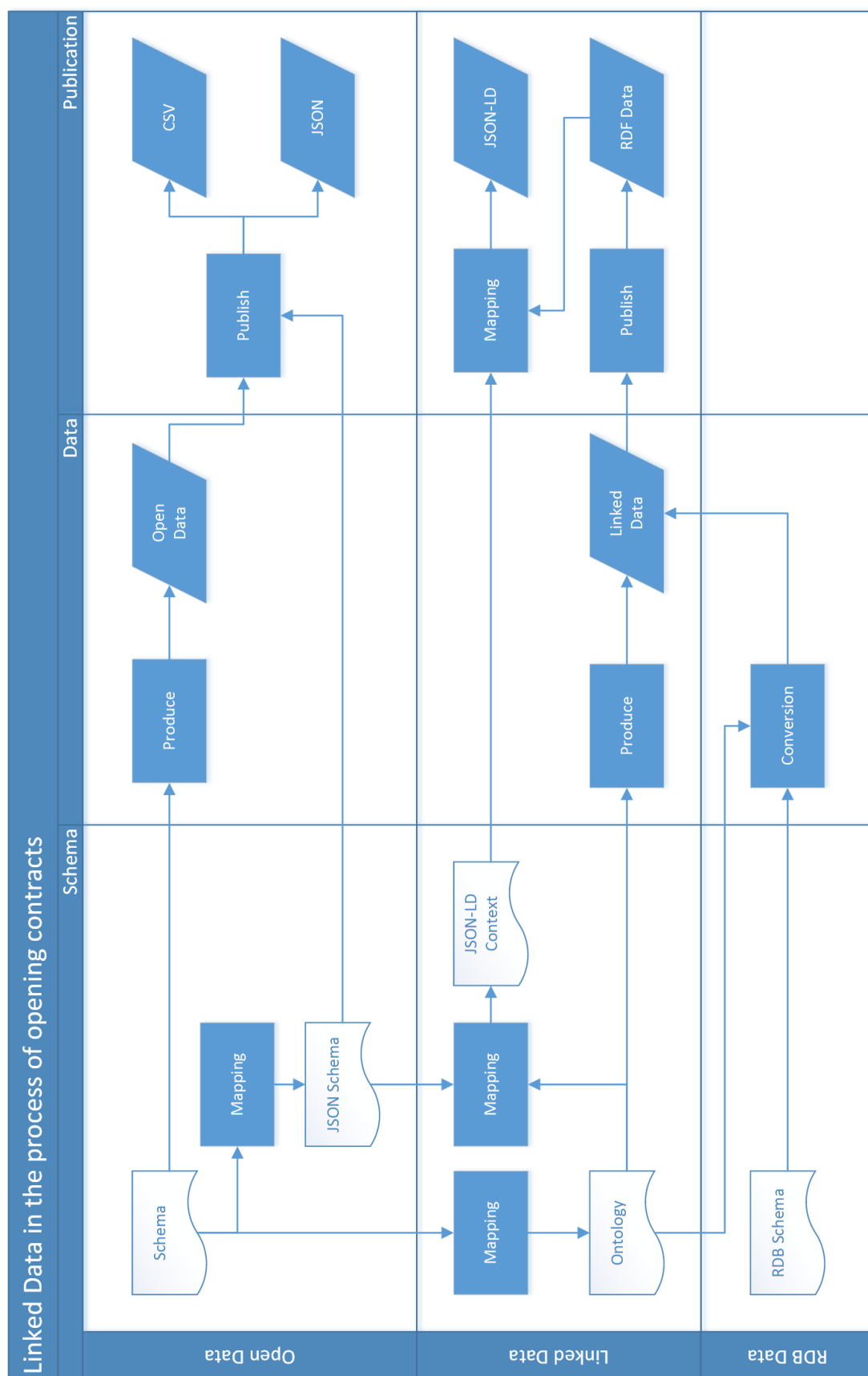
Začátek procesu je v tvorbě schématu. V našem případě se jedná o datový standard pro otevřené smlouvy. Na základě schématu je v diagramu znázorněna možnost produkovat otevřená data. Ta jsou pak serializovatelná do konkrétních datových formátů. Lepšími datovými formáty jsou ale ty, které jsou definované pomocí schématu datového, pro naše účely se jedná o formát JSON. Z diagramu je vidět, že ze schématu vytvoříme datové JSON schéma, které je poté využitelné při publikaci dat.

Na základě schématu můžeme také nadefinovat RDF ontologii, díky které se dostaneme do světa Linked Data. V diagramu je obdobně jako u otevřených dat znázorněna možnost na základě ontologií produkovat Linked Data a ta pak serializovat do RDF datových formátů.

Speciálním případem serializace RDF dat jsou data ve formátu JSON-LD. Na základě schématu, resp. datového JSON schématu a RDF ontologie, můžeme vytvořit takový JSON-LD kontext, že výsledná vypublikovaná JSON-LD data budou naplňovat jak datový standard, tak i RDF ontologii.

Posledním krokem je zapojení relačních dat do procesu otevírání smluv. Tato relační data na základě relačního schématu, resp. jeho datového modelu a RDF ontologie, zkonvertujeme do Linked Data.

Lze si všimnout, že se nebavíme pouze o údajích o smlouvách. Tento postup lze obecně použít nejen pro smlouvy, ale i pro libovolnou jinou doménu.



Obrázek 9.1: Linked Data v procesu otevírání smluv

# Závěr

V rámci této práce jsme si kladli za cíl využít principů Linked Data pro publikaci a sdílení dat o smlouvách.

Začali jsme definováním datového standardu pro otevřené smlouvy. Ten probíhal v rámci akční skupiny pod záštitou Oživení o.s. a Centra aplikované ekonomie o.s. Hlavním přínosem je reálná možnost zařazení standardu do sady doporučení Ministerstva vnitra pro publikovatelné datové sady na podzim roku 2015. Na základě standardu byla navržena podoba datových formátů pro jejich publikaci. Dílčím výsledkem byla tvorba metodiky ve formě webové aplikace mající za cíl technicky i věcně datový standard popsat.

V dalším kroku byla navržena ontologie pro publikace otevřených smluv v RDF podobě. Zaměřili jsme se také na možnost propojení se souvisejícími daty. Ukázali jsme výhody serializace RDF dat v JSON-LD formátu. Klíčovým přínosem JSON-LD formátu je, že vypublikovaná data splňují datový standard pro otevřené smlouvy a zároveň se jedná o RDF data.

V následující části jsme navrhli a implementovali platformu pro otevírání smluv. Platforma je složena ze třech základních součástí: konverzního modulu, jednotného úložiště a prezentační webové aplikace.

V návrhu konverzního modulu jsme se zaměřili na konverzi relačních dat stávajících informačních systémů do RDF podoby splňující principy Linked Data. Jako zdroj pro konkrétní implementaci byl zvolen modul Munis ESML informačního systému Triada spol. s r.o. Řešení přináší zajímavý přístup mapování relačních dat do RDF podoby pomocí R2RML skriptu. Díky tomu lze konverzní mechanismus s drobnými úpravami využít i nad jinými informačními systémy. Druhou součástí platformy bylo navrženo a implementováno jednotné úložiště. Úložiště je na základě definovaného datového katalogu schopno stahovat konkrétní datasety údajů o smlouvách v RDF podobě a ukládat je do triplestore databáze. Díky navržené jednoznačné identifikaci entit odpadly problémy s heterogenitou dat.

Jako poslední součást platformy byla navržena a implementována webová aplikace zpřístupňující údaje o smlouvách z jednotného úložiště koncovým uživatelům. V rámci aplikace jsme se zaměřili na demonstraci přínosů využití principů Linked Data. Navrhli jsme proto síť propojených datasetů s cílem poskytnout uživateli údaje o smlouvách obohacených o informace např. z ARESu, RUIANU, nebo Věstníku veřejných zakázek.

Následně jsme otestovali konverzní mechanismus a webovou aplikaci ve snaze simulovat možnosti reálného využití. Na základě procesu otevírání smluv jsme také uvedli obecný postup otevírání dat využitelný i v jiných doménách.

## Linked Data v procesu otevírání smluv

V rámci této práce jsme ukázali, že využití principů Linked Data je pro doménu smluvních údajů možné. Ukázali jsme také postup, jak toho dosáhnout. Shrňme si tedy základní výhody a nevýhody využití Linked Data v procesu otevírání smluv:

## Výhody

- V našem případě se zároveň jedná o otevřená data. Údaje o smlouvách tedy mohou být dostupné široké veřejnosti na internetu a přinášet veškeré výhody otevřených dat.
- Díky možnosti propojení se smlouvy stanou součástí daleko širšího kontextu otevřených a propojitelných dat. Zvýší se tak informační hodnota každé smlouvy
- Údaje o každé smlouvě, resp. entitě jsou dostupné pod vlastním URI. Smlouva je tak na jednom místě a můžeme se na ni libovolně odkazovat.

## Nevýhody

- Nelze očekávat, že práce nad daty využívajícími principy Linked Data, bude rychlá jako nad relačními databázemi.
- Častou nevýhodou využití principu Linked Data bývá velká náročnost kladená na subjekt, který chce zveřejňovat (v rámci platformy navržený konverzní mechanismus ale nároky na subjekt výrazně redukuje).
- Obecně příprava dat, tvorba standardu, ontologie, definování URI identifikátorů apod. vyžaduje jisté znalosti a netriviální úsilí.

K přípravě dat bych rád doplnil, že před zpracováním podobných domén, jako jsou údaje o smlouvách do Linked Data podoby, je důležité navrhnout datový standard definující, co je vůbec vybrané domény obsahem. Ze zkušenosti v rámci akční skupiny pro tvorbu standardu mohu konstatovat, že tato činnost nemusí být triviální. Každá konkrétní položka může mít různé technické, ale hlavně i právní aspekty, které je třeba podrobit diskuzi s relevantními osobami.

S ohledem na potřebnou přípravu dat se tedy nabízí otázka celkové pracnosti. Náročnost přípravy dat a implementace konverzního modulu bych na základě zkušenosti odhadl zhruba takto:

Datový standard	Linked Data	Konverzní modul + R2ML mapování
2 člověkoměsíce	1,5 člověkoměsíce	2,5 člověkoměsíce
33,3%	25%	41,7%

Tabulka 9.1: Odhad pracnosti přípravy dat a implementace konverzního modulu

Celkovou náročnost otevření jedné domény můžeme tedy odhadnout na zhruba 6 člověkoměsíci.

# Seznam použité literatury

- [1] MINISTERSTVO VNITRA *Ministerstvo vnitra* - <http://www.mvcr.cz/>
- [2] REKONSTRUKCE STÁTU *Rekonstrukce státu* - <http://www.rekonstrukcestatu.cz/cs/>
- [3] FOND OTAKARA MOTEJLA *Fond Otakara Motejla* - <http://www.motejl.cz/>
- [4] OŽIVENÍ O.S. <http://www.oziveni.cz/>
- [5] OPENDATA.CZ <http://opendata.cz/>
- [6] MINISTERSTVO VNITRA - OTEVŘENÁ DATA <http://www.mvcr.cz/clanek/otevrena-data.aspx?q=Y2hudW09Mg%3d%3d>
- [7] 5★ OPEN DATA <http://5stardata.info/>
- [8] LINKED DATA - DESIGN ISSUES *Linked Data - Design Issues* [online], 2006. Dostupné z <http://www.w3.org/DesignIssues/LinkedData.html>
- [9] WEB ONTOLOGY LANGUAGE <http://www.w3.org/2001/sw/wiki/OWL>
- [10] RDF SCHEMA <http://www.w3.org/TR/rdf-schema/>
- [11] N-TRIPLES <http://www.w3.org/TR/n-triples/>
- [12] N-QUADS <http://www.w3.org/TR/n-quads/>
- [13] RDF/XML <http://www.w3.org/TR/REC-rdf-syntax>
- [14] TURTLE <http://www.w3.org/TR/turtle/>
- [15] TRIG <http://www.w3.org/TR/trig>
- [16] RDFa <http://www.w3.org/TR/rdfa-syntax/>
- [17] (X)HTML <http://www.w3.org/TR/xhtml1/>
- [18] JSON-LD <http://www.w3.org/TR/json-ld/>
- [19] PORTÁL VEŘEJNÉ SPRÁVY <http://portal.gov.cz>
- [20] OTEVŘENÁ MĚSTA <http://www.otevrenamesta.cz/>
- [21] DATOVÝ FORMÁT JSON <https://www.ietf.org/rfc/rfc4627.txt>
- [22] DATOVÝ FORMÁT CSV <https://tools.ietf.org/html/rfc4180>
- [23] JSON SCHEMA <http://json-schema.org/documentation.html>
- [24] METODIKA ZVEŘEJŇOVÁNÍ SMLUV <http://standard.zindex.cz/doku.php/cs/introduction>
- [25] WIKIPEDIA <http://wikipedia.org/>
- [26] DOKUWIKI <https://www.dokuwiki.org/>

# Seznam obrázků

2.1	Logo otevřených dat . . . . .	7
2.2	Stupně otevřenosti dat, zdroj: [7] . . . . .	10
2.3	Linked Open Data Cloud, Srpen 2014, zdroj: <a href="http://lod-cloud.net/">http://lod-cloud.net/</a> . . . . .	11
2.4	Základní RDF trojice . . . . .	13
2.5	Jednoduchý RDF graf . . . . .	14
2.6	RDF graf s přiřazenými typy . . . . .	14
2.7	Ontologie třídy <i>Contract</i> . . . . .	16
2.8	Možnost propojení dat . . . . .	16
3.1	Metodika zveřejňování smluv, zdroj: <a href="http://standard.zindex.cz/">http://standard.zindex.cz/</a> . . . . .	37
6.1	Základní pohled na platformu otevřených smluv (Logical view) . . . . .	57
6.2	Rozdělení platformy do modulů (Decomposition view) . . . . .	58
6.3	Propojená datová síť . . . . .	61
6.4	Obohacený kontext smluv díky propojeným datům . . . . .	63
7.1	Modul ESML . . . . .	64
7.2	Zjednodušený datový model (bez atributů) Munis ESML . . . . .	66
7.3	R2RML mapování vlastností Smlouvy . . . . .	68
7.4	R2RML mapování vlastností Smlouvy (Externího kontaktu) . . . . .	69
7.5	R2RML mapování vlastností Verze . . . . .	69
7.6	R2RML mapování vlastností Smluvní strany a Adresy . . . . .	71
7.7	R2RML mapování vlastností Příloha . . . . .	72
7.8	R2RML mapování vlastností Milníku . . . . .	73
7.9	Pipeline nad jednotným úložištěm pro zpracování dat o smlouvách . . . . .	77
9.1	Linked Data v procesu otevírání smluv . . . . .	86

# Seznam tabulek

3.1	zdroj:[24]	24
3.2	Validita, zdroj:[24]	24
3.3	Akceptovatelné soubory,zdroj:[24]	25
3.4	Vlastnosti dokumentu, zdroj:[24]	25
3.5	Vlastnosti vydavatele, zdroj:[24]	25
3.6	Vlastnosti verze smlouvy, zdroj:[24]	26
3.7	Vlastnosti smlouvy, zdroj:[24]	27
3.8	Vlastnosti přílohy, zdroj:[24]	27
3.9	Vlastnosti dodatku, zdroj:[24]	27
3.10	Vlastnosti smluvní strany, zdroj:[24]	28
3.11	Vlastnosti nadřazené instituce, zdroj:[24]	28
3.12	Vlastnosti adresy, zdroj:[24]	29
3.13	Vlastnosti objednávky, zdroj:[24]	29
3.14	Vlastnosti faktury, zdroj:[24]	29
3.15	Vlastnosti implementace, zdroj:[24]	30
3.16	Vlastnosti milníku, zdroj: [24]	30
3.17	Vlastnosti transakce, zdroj:[24]	30
3.18	Číselník typu dokumentu, zdroj:[24]	31
3.19	Číselník typu smlouvy, zdroj:[24]	31
4.1	Mapování entity Document	42
4.2	Mapování entity Vydavatel	43
4.3	Mapování entity Verze smlouvy	43
4.4	Mapování entity Smlouva	44
4.5	Mapování entity Příloha	44
4.6	Mapování entity Dodatek	44
4.7	Mapování entity Smluvní strana	45
4.8	Mapování entity Nadřazená instituce	45
4.9	Mapování entity Nadřazená instituce	45
4.10	Mapování entity Objednávka	46
4.11	Mapování entity Faktura	46
4.12	Mapování entity Implementace	46
4.13	Mapování entity Milník	46
4.14	Mapování entity Transakce	47
9.1	Odhad pracnosti přípravy dat a implementace konverzního modulu	88

# Výpisy kódu

2.1	Příklad RDF dat - N-Triples . . . . .	17
2.2	Příklad RDF dat - Turtle . . . . .	18
2.3	Příklad RDF Ontologie - Turtle . . . . .	18
2.4	Obyčejný JSON dokument . . . . .	20
2.5	Příklad RDF dat - JSON-LD s Contextem . . . . .	20
2.6	Příklad RDF dat - JSON-LD . . . . .	20
3.1	JSON soubor s jednou smlouvou . . . . .	33
4.1	JSON-LD Context . . . . .	48
4.2	JSON-LD Soubor s jednou smlouvou . . . . .	50
6.1	Datový katalog pro jednotné úložiště . . . . .	60
7.1	Rozšíření knihovny JSON-LD.Net . . . . .	75
7.2	Získej všechny smlouvy . . . . .	77
7.3	Získej vydavatele na základě jména . . . . .	78
7.4	Získej všechny smlouvy daného vydavatele úložiště . . . . .	78
7.5	Získej otevírací hodiny subjektu . . . . .	79
7.6	Získej adresní místo z RUIANu . . . . .	79
7.7	Získej polohu subjektu . . . . .	79
7.8	Získej foto subjektu . . . . .	80
7.9	Získej smlouvu . . . . .	80
7.10	Získej smluvní strany na základě smlouvy . . . . .	80
7.11	Získej přílohy na základě smlouvy . . . . .	80
7.12	Získej dodatky na základě smlouvy . . . . .	81
7.13	Získej milníky na základě smlouvy . . . . .	81
7.14	Získej verzi smlouvy . . . . .	81
7.15	Získej veřejné zakázky na základě subjektu . . . . .	82
9.1	RDF Ontologie pro smlouvy . . . . .	97
9.2	R2RML mapovací skript . . . . .	105



# Seznam použitých zkratek

# Přílohy

# A Příloha

## B Příloha

# C Příloha

```
1 @prefix : <http://tiny.cc/open-contracting#> .
2
3 @prefix com: <https://w3id.org/commerce#> .
4 @prefix dcterms: <http://purl.org/dc/terms/> .
5 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
6 @prefix gr: <http://purl.org/goodrelations/v1#> .
7 @prefix owl: <http://www.w3.org/2002/07/owl#> .
8 @prefix ps: <https://w3id.org/payswarm#> .
9 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
10 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
11 @prefix schema: <http://schema.org/> .
12 @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
13 @prefix vann: <http://purl.org/vocab/vann/> .
14 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
15
16 #### — Metadata —
17
18 <http://purl.org/open-contracting/ontology> a owl:Ontology ;
19 owl:versionInfo "0.1" ;
20 dcterms:title ""@cs, ""@en ;
21 dcterms:description ""@cs ;
22 dcterms:description ""@en ;
23 dcterms:modified "2015-05-25"^^xsd:date ;
24 vann:preferredNamespaceUri "http://tiny.cc/open-contracting#" ;
25 vann:preferredNamespacePrefix "ocn" ;
26 dcterms:creator <http://www.ms.mff.cuni.cz/~hryzlikp> ;
27 dcterms:rights <http://creativecommons.org/licenses/by/3.0/cz/> .
28
29 #### — Ontology author —
30
31 <http://www.ms.mff.cuni.cz/~hryzlikp> a foaf:Person ;
32 foaf:name "Pavel Hryzlík" ;
33 foaf:mbox <mailto:hryzlik@gmail.com> .
34
35 #### — Classes —
36
37 :Document a owl:Class ;
38 rdfs:label "Dokument"@cs, "Document"@en ;
39 rdfs:comment ""@cs ;
40 rdfs:comment ""@en ;
41 rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
42
43 :Contract a owl:Class ;
44 rdfs:label "Smlouva"@cs, "Contract"@en ;
45 rdfs:comment ""@cs ;
46 rdfs:comment ""@en ;
47 rdfs:subClassOf :Document ;
48 rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
49
50 :Attachment a owl:Class ;
51 rdfs:label "Příloha"@cs, "Attachment"@en ;
52 rdfs:comment ""@cs ;
53 rdfs:comment ""@en ;
54 rdfs:subClassOf :Document ;
55 rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
```

```

56
57 :Amendment a owl:Class ;
58   rdfs:label "Dodatek"@cs, "Amendment"@en ;
59   rdfs:comment ""@cs ;
60   rdfs:comment ""@en ;
61   rdfs:subClassOf :Document ;
62   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
63
64 :Order a owl:Class ;
65   rdfs:label "Objednávka"@cs, "Order"@en ;
66   rdfs:comment ""@cs ;
67   rdfs:comment ""@en ;
68   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
69
70 :Invoice a owl:Class ;
71   rdfs:label "Faktura"@cs, "Invoice"@en ;
72   rdfs:comment ""@cs ;
73   rdfs:comment ""@en ;
74   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
75
76 :Version a owl:Class ;
77   rdfs:label "Verze"@cs, "Version"@en ;
78   rdfs:comment ""@cs ;
79   rdfs:comment ""@en ;
80   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
81
82 :Implementation a owl:Class ;
83   rdfs:label "Implementace"@cs, "Implementation"@en ;
84   rdfs:comment ""@cs ;
85   rdfs:comment ""@en ;
86   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
87
88 :Milestone a owl:Class ;
89   rdfs:label "Milník"@cs, "Milestone"@en ;
90   rdfs:comment ""@cs ;
91   rdfs:comment ""@en ;
92   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
93
94 com:Transaction vann:usageNote ""@cs, ""@en .
95
96 ### — Properties —
97
98 dcterms:identifier vann:usageNote ""@cs, ""@en .
99 dcterms:issued vann:usageNote ""@cs, ""@en .
100 dcterms:language vann:usageNote ""@cs, ""@en .
101
102 :documents a owl:ObjectProperty, owl:FunctionalProperty ;
103   rdfs:label ""@cs, ""@en ;
104   rdfs:comment ""@cs ;
105   rdfs:comment ""@en ;
106   rdfs:range [ a rdfs:Class ; owl:unionOf ( :Contract :Attachment
107     :Amendment ) ] ;
107   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
108
109 :orders a owl:ObjectProperty, owl:FunctionalProperty ;
110   rdfs:label ""@cs, ""@en ;
111   rdfs:comment ""@cs ;
112   rdfs:comment ""@en ;

```

```

113   rdfs:range :Order ;
114   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
115
116 :invoices a owl:ObjectProperty, owl:FunctionalProperty ;
117   rdfs:label ""@cs, ""@en ;
118   rdfs:comment ""@cs ;
119   rdfs:comment ""@en ;
120   rdfs:range :Invoice ;
121   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
122
123 :party a owl:ObjectProperty ;
124   rdfs:label ""@cs, ""@en ;
125   rdfs:comment ""@cs ;
126   rdfs:comment ""@en ;
127   rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Contract :Order
128     :Invoice ) ] ;
128   rdfs:range gr:BusinessEntity ;
129   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
130
131 :implementation a owl:ObjectProperty, owl:FunctionalProperty ;
132   rdfs:label ""@cs, ""@en ;
133   rdfs:comment ""@cs ;
134   rdfs:comment ""@en ;
135   rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Document :Order ) ] ;
136   rdfs:range :Implementation ;
137   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
138
139 :uri a owl:DatatypeProperty, owl:FunctionalProperty ;
140   rdfs:label ""@cs, ""@en ;
141   rdfs:comment ""@cs ;
142   rdfs:comment ""@en ;
143   rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Contract :Attachment
144     :Amendment :Version ) ] ;
144   rdfs:range xsd:anyURI ;
145   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
146
147 :contract a owl:DatatypeProperty, owl:FunctionalProperty ;
148   rdfs:label ""@cs, ""@en ;
149   rdfs:comment ""@cs ;
150   rdfs:comment ""@en ;
151   rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Attachment :Amendment
152     ) ] ;
152   rdfs:range xsd:anyURI ;
153   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
154
155 :parentDocument a owl:DatatypeProperty, owl:FunctionalProperty ;
156   rdfs:label ""@cs, ""@en ;
157   rdfs:comment ""@cs ;
158   rdfs:comment ""@en ;
159   rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Order :Invoice ) ] ;
160   rdfs:range xsd:anyURI ;
161   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
162
163 :publisherId a owl:DatatypeProperty, owl:FunctionalProperty ;
164   rdfs:label ""@cs, ""@en ;
165   rdfs:comment ""@cs ;
166   rdfs:comment ""@en ;

```

```

167   rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Transaction :Version )
168               ] ;
169   rdfs:range xsd:string ;
170   rdfs:subPropertyOf dcterms:identifier ;
171   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
172
173 :dueDate a owl:DatatypeProperty, owl:FunctionalProperty ;
174   rdfs:label ""@cs, ""@en ;
175   rdfs:comment ""@cs ;
176   rdfs:comment ""@en ;
177   rdfs:domain [ a rdfs:Class ; owl:unionOf ( :Invoice :Milestone ) ]
178               ;
179   rdfs:range xsd:dateTime ;
180   rdfs:subPropertyOf dcterms:date ;
181   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
182
183 ### :Document properties —
184
185 com:contentUrl vann:usageNote ""@cs, ""@en .
186
187 :valid a owl:DatatypeProperty, owl:FunctionalProperty ;
188   rdfs:label ""@cs, ""@en ;
189   rdfs:comment ""@cs ;
190   rdfs:comment ""@en ;
191   rdfs:domain :Document ;
192   rdfs:range xsd:boolean ;
193   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
194
195 :anonymised a owl:DatatypeProperty, owl:FunctionalProperty ;
196   rdfs:label ""@cs, ""@en ;
197   rdfs:comment ""@cs ;
198   rdfs:comment ""@en ;
199   rdfs:domain :Document ;
200   rdfs:range xsd:boolean ;
201   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
202
203 :responsiblePerson a owl:ObjectProperty ;
204   rdfs:label ""@cs, ""@en ;
205   rdfs:comment ""@cs ;
206   rdfs:comment ""@en ;
207   rdfs:domain :Document ;
208   rdfs:range dcterms:Person ;
209   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
210
211 :publisher a owl:ObjectProperty ;
212   rdfs:label ""@cs, ""@en ;
213   rdfs:comment ""@cs ;
214   rdfs:comment ""@en ;
215   rdfs:domain :Document ;
216   rdfs:range foaf:Organization ;
217   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
218
219 :address a owl:ObjectProperty ;
220   rdfs:label ""@cs, ""@en ;
221   rdfs:comment ""@cs ;
222   rdfs:comment ""@en ;
223   rdfs:domain :Document ;
224   rdfs:range schema:PostalAddress ;

```



```

223   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
224
225 :version a owl:ObjectProperty ;
226   rdfs:label ""@cs, ""@en ;
227   rdfs:comment ""@cs ;
228   rdfs:comment ""@en ;
229   rdfs:domain :Document ;
230   rdfs:range :Version ;
231   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
232
233 ### :Contract properties —
234
235 dterms:title vann:usageNote ""@cs, ""@en .
236 dterms:type vann:usageNote ""@cs, ""@en .
237 dterms:created vann:usageNote ""@cs, ""@en .
238 dterms:description vann:usageNote ""@cs, ""@en .
239 gr:hasCurrencyValue vann:usageNote ""@cs, ""@en .
240 gr:hasCurrency vann:usageNote ""@cs, ""@en .
241 ps:validFrom vann:usageNote ""@cs, ""@en .
242 ps:validUntil vann:usageNote ""@cs, ""@en .
243
244 :awardID a owl:DatatypeProperty, owl:FunctionalProperty ;
245   rdfs:label ""@cs, ""@en ;
246   rdfs:comment ""@cs ;
247   rdfs:comment ""@en ;
248   rdfs:domain :Contract ;
249   rdfs:range xsd:string ;
250   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
251
252 :awardProfileID a owl:DatatypeProperty, owl:FunctionalProperty ;
253   rdfs:label ""@cs, ""@en ;
254   rdfs:comment ""@cs ;
255   rdfs:comment ""@en ;
256   rdfs:domain :Contract ;
257   rdfs:range xsd:string ;
258   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
259
260 :contractType a owl:DatatypeProperty, owl:FunctionalProperty ;
261   rdfs:label ""@cs, ""@en ;
262   rdfs:comment ""@cs ;
263   rdfs:comment ""@en ;
264   rdfs:domain :Contract ;
265   rdfs:range xsd:string ;
266   rdfs:subPropertyOf dterms:type ;
267   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
268
269 :subjectType a owl:DatatypeProperty, owl:FunctionalProperty ;
270   rdfs:label ""@cs, ""@en ;
271   rdfs:comment ""@cs ;
272   rdfs:comment ""@en ;
273   rdfs:domain :Contract ;
274   rdfs:range xsd:string ;
275   rdfs:subPropertyOf dterms:type ;
276   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
277
278 :priceAnnual a owl:DatatypeProperty, owl:FunctionalProperty ;
279   rdfs:label ""@cs, ""@en ;
280   rdfs:comment ""@cs ;

```

```

281   rdfs:comment """@en ;
282   rdfs:domain :Contract ;
283   rdfs:range xsd:boolean ;
284   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
285
286 :funding a owl:DatatypeProperty, owl:FunctionalProperty ;
287   rdfs:label """@cs, """@en ;
288   rdfs:comment """@cs ;
289   rdfs:comment """@en ;
290   rdfs:domain :Contract ;
291   rdfs:range xsd:string ;
292   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
293
294 :attachment a owl:ObjectProperty, owl:FunctionalProperty ;
295   rdfs:label """@cs, """@en ;
296   rdfs:comment """@cs ;
297   rdfs:comment """@en ;
298   rdfs:domain :Contract ;
299   rdfs:range :Attachment ;
300   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
301
302 :amendment a owl:ObjectProperty, owl:FunctionalProperty ;
303   rdfs:label """@cs, """@en ;
304   rdfs:comment """@cs ;
305   rdfs:comment """@en ;
306   rdfs:domain :Contract ;
307   rdfs:range :Amendment ;
308   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
309
310 :currentValidContract a owl:DatatypeProperty, owl:FunctionalProperty
311   ;
312   rdfs:label """@cs, """@en ;
313   rdfs:comment """@cs ;
314   rdfs:comment """@en ;
315   rdfs:domain :Contract ;
316   rdfs:range xsd:anyURI ;
317   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
318
319 :competency a owl:DatatypeProperty ;
320   rdfs:label """@cs, """@en ;
321   rdfs:comment """@cs ;
322   rdfs:comment """@en ;
323   rdfs:domain :Contract ;
324   rdfs:range xsd:string ;
325   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
326
327 ### :Version properties —
328
329 dterms:issued vann:usageNote """@cs, """@en .
330
331 :versionOrder a owl:DatatypeProperty, owl:FunctionalProperty ;
332   rdfs:label """@cs, """@en ;
333   rdfs:comment """@cs ;
334   rdfs:comment """@en ;
335   rdfs:domain :Version ;
336   rdfs:range xsd:integer ;
337   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .

```

```

338 ### :Implementation properties —
339
340 :milestone a owl:ObjectProperty, owl:FunctionalProperty ;
341   rdfs:label """@cs, """@en ;
342   rdfs:comment """@cs ;
343   rdfs:comment """@en ;
344   rdfs:domain :Implementation ;
345   rdfs:range :Milestone ;
346   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
347
348 :transaction a owl:ObjectProperty, owl:FunctionalProperty ;
349   rdfs:label """@cs, """@en ;
350   rdfs:comment """@cs ;
351   rdfs:comment """@en ;
352   rdfs:domain :Implementation ;
353   rdfs:range com:Transaction ;
354   rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
355
356 ### :Milestone properties —
357
358 dcterms:identifier vann:usageNote """@cs, """@en .
359
360 ### :Transaction properties —
361
362 dcterms:identifier vann:usageNote """@cs, """@en .
363 dcterms:date vann:usageNote """@cs, """@en .
364 gr:hasCurrencyValue vann:usageNote """@cs, """@en .
365 com:source vann:usageNote """@cs, """@en .
366 com:destination vann:usageNote """@cs, """@en .
367
368 ### :Attachment properties —
369
370 dcterms:identifier vann:usageNote """@cs, """@en .
371 dcterms:title vann:usageNote """@cs, """@en .
372 dcterms:created vann:usageNote """@cs, """@en .
373
374 ### :Amendment properties —
375
376 dcterms:identifier vann:usageNote """@cs, """@en .
377 dcterms:title vann:usageNote """@cs, """@en .
378 dcterms:created vann:usageNote """@cs, """@en .
379
380 ### :Order properties —
381
382 dcterms:title vann:usageNote """@cs, """@en .
383 dcterms:created vann:usageNote """@cs, """@en .
384 gr:hasCurrencyValue vann:usageNote """@cs, """@en .
385 gr:hasCurrency vann:usageNote """@cs, """@en .
386
387 ### :Invoice properties —
388
389 dcterms:title vann:usageNote """@cs, """@en .
390 dcterms:created vann:usageNote """@cs, """@en .
391 gr:hasCurrencyValue vann:usageNote """@cs, """@en .
392 gr:hasCurrency vann:usageNote """@cs, """@en .
393
394 ### gr:BusinessEntity properties —
395

```

```

396 dcterms:identifier vann:usageNote ""@cs, ""@en .
397 gr:legalName vann:usageNote ""@cs, ""@en .
398 gr:valueAddedTaxIncluded vann:usageNote ""@cs, ""@en .
399 schema:addressCountry vann:usageNote ""@cs, ""@en .
400
401 :localID a owl:DatatypeProperty, owl:FunctionalProperty ;
402     rdfs:label ""@cs, ""@en ;
403     rdfs:comment ""@cs ;
404     rdfs:comment ""@en ;
405     rdfs:domain gr:BusinessEntity ;
406     rdfs:range xsd:integer ;
407     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
408
409 :payer a owl:DatatypeProperty, owl:FunctionalProperty ;
410     rdfs:label ""@cs, ""@en ;
411     rdfs:comment ""@cs ;
412     rdfs:comment ""@en ;
413     rdfs:domain gr:BusinessEntity ;
414     rdfs:range xsd:boolean ;
415     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
416
417 :noID a owl:DatatypeProperty, owl:FunctionalProperty ;
418     rdfs:label ""@cs, ""@en ;
419     rdfs:comment ""@cs ;
420     rdfs:comment ""@en ;
421     rdfs:domain gr:BusinessEntity ;
422     rdfs:range xsd:boolean ;
423     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
424
425 :superiorInstitution a owl:ObjectProperty ;
426     rdfs:label ""@cs, ""@en ;
427     rdfs:comment ""@cs ;
428     rdfs:comment ""@en ;
429     rdfs:domain gr:BusinessEntity ;
430     rdfs:range gr:BusinessEntity ;
431     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
432
433 ### foaf:Organization properties —
434
435 :authentication a owl:DatatypeProperty, owl:FunctionalProperty ;
436     rdfs:label ""@cs, ""@en ;
437     rdfs:comment ""@cs ;
438     rdfs:comment ""@en ;
439     rdfs:domain foaf:Organization ;
440     rdfs:range xsd:string ;
441     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .
442
443 ### schema:PostalAddress properties —
444
445 :nuts a owl:DatatypeProperty, owl:FunctionalProperty ;
446     rdfs:label ""@cs, ""@en ;
447     rdfs:comment ""@cs ;
448     rdfs:comment ""@en ;
449     rdfs:domain schema:PostalAddress ;
450     rdfs:range xsd:string ;
451     rdfs:isDefinedBy <http://tiny.cc/open-contracting> .

```

Výpis kódu 9.1: RDF Ontologie pro smlouvy

# D Příloha

```
1 # The R2RML mapping contracts
2
3 @base      <http://tiny.cc/open-contracting#> .
4 @prefix rr: <http://www.w3.org/ns/r2rml#> .
5
6 @prefix cn: <http://tiny.cc/open-contracting#> .
7 @prefix com: <https://w3id.org/commerce#> .
8 @prefix dc: <http://purl.org/dc/terms/> .
9 @prefix dcmi: <http://dublincore.org/documents/dcmi-type-
    vocabulary/> .
10 @prefix foaf: <http://xmlns.com/foaf/0.1/> .
11 @prefix gr: <http://purl.org/goodrelations/v1#> .
12 @prefix owl: <http://www.w3.org/2002/07/owl#> .
13 @prefix pc: <http://purl.org/procurement/public-contracts#> .
14 @prefix ps: <https://w3id.org/payswarm#> .
15 @prefix schema: <http://schema.org/> .
16 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
17 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
18 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
19
20 <#PublisherTableView> rr:sqlQuery """
21     SELECT NAZEVORGANIZACE, ICO,
22     IIF(ICO is NULL, 'true', 'false') As NoID
23     FROM [triada].[TRLORGADR] WHERE HLAVNI = 'T'
24     """ .
25
26 <#ContractsTableView> rr:sqlQuery """
27     SELECT Smlouva.ID,
28     Verze.PORADIVERZE, Verze.PREDMET, Verze.POPIS.POPIS, Verze.
29     TYP_SMLOUVY, Mena.ZKRATKA, Verze.CELKOVACASTKA,
30     Verze.DATUMPODPISU, Verze.DATUMUCINOSTI, Verze.DATUMKONCENI,
31     Verze.SMLUVSTRANROZD, Verze.DATUMZMENYSTAVUTS,
32     VZakazka.EVIDENCNICISLOZAKAZKY, VZakazka.EVIDENCNICISLOFORMULARE,
33     (CASE Verze.ANONYMIZOVANO
34         WHEN 'T' THEN 'true'
35         WHEN 'F' THEN 'false'
36     END) AS Anonymizovano
37     FROM [triada].[ESMLUV_SMLOUVA] AS Smlouva
38     JOIN [triada].[ESMLUV_VERZESMLOUVY] AS Verze ON Smlouva.ID = Verze
39     .SMLOUVA
40     JOIN [triada].[ESMLUV_MENA] AS Mena ON Verze.MENA = Mena.ID
41     LEFT JOIN [triada].[ESMLUV_VERZAKAZKA] AS VZakazka ON Verze.
42     VEREJNAZAKAZKA = VZakazka.ID
43     WHERE Smlouva.RODIC is NULL
44     """ .
45
46 <#ContractTypesTableView> rr:sqlQuery """
47     SELECT Smlouva.ID, Verze.PORADIVERZE,
48     (CASE TypSmlouvy.TYP
49         WHEN '1' THEN 'Nájemní smlouva'
50         WHEN '2' THEN 'Darovací smlouva'
51         WHEN '3' THEN 'Kupní smlouva'
52         WHEN '4' THEN 'Směnná smlouva'
53         WHEN '5' THEN 'Pojistná smlouva'
54         WHEN '6' THEN 'Smlouva o výpůjčce'
55     )
```

```

51     WHEN '7' THEN 'Smlouva o dílo'
52     WHEN '8' THEN 'Licenční smlouva'
53     WHEN '9' THEN 'Mandátní smlouva'
54     WHEN '10' THEN 'Leasingová smlouva'
55     WHEN '11' THEN 'Pachtovní smlouva'
56     WHEN '12' THEN 'Smlouva o zřízení věcného břemene'
57     WHEN '13' THEN 'Smlouva o provedení stavby'
58     WHEN '14' THEN 'Smlouva o provedení práce'
59     WHEN '15' THEN 'Smlouva o provedení uměleckého výkonu'
60     WHEN '16' THEN 'Smlouva o úvěru'
61     WHEN '17' THEN 'Smlouva o uzavření budoucí smlouvy'
62     WHEN '18' THEN 'Veřejnoprávní smlouva'
63     WHEN '19' THEN 'Jiná'
64 END) AS Typ,
65     If(TypSmlouvy.TYP = '18', 'Veřejnoprávní smlouva', 'Soukromoprávní smlouva') As Kompetence
66 FROM [triada].[ESMLUV_SMLOUVA] AS Smlouva
67 JOIN [triada].[ESMLUV_VERZESMLOUVY] As Verze ON Smlouva.ID = Verze.SMLOUVA
68 JOIN [triada].[ESMLUV_TYPSMLOUVY] As TypSmlouvy ON Verze.TYPSMLOUVY = TypSmlouvy.ID
69 WHERE Smlouva.RODIC is NULL
70 " " " .
71
72 <#PartiesTableView> rr:sqlQuery " " "
73     SELECT Smlouva.ID, Verze.PORADIVERZE, SmlStrana.HAD_POUZITA
74     FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
75     JOIN [triada].[ESMLUV_SMLUVSTRANA] AS SmlStrana ON Verze.SMLUVSTRANROZD = SmlStrana.SMLUVSTRANYROZDELOVNIK
76     JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.SMLOUVA
77     WHERE Smlouva.RODIC is NULL
78     " " " .
79
80 <#PartyTableView> rr:sqlQuery " " "
81     SELECT SmlStrana1.HAD_POUZITA,
82     Had.NAZEV_SUBJEKTU, Had.ICO, Had.STAT, Had.ULICE, Had.CISLA, Had.MESTO, Had.PSC,
83     IIF(Had.ICO is NULL, 'true', 'false') As NoID,
84     (CASE SmlStrana1.PLATCEDPH
85         WHEN 'T' THEN 'true'
86         WHEN 'F' THEN 'false'
87     END) AS PlatceDPH
88     FROM [triada].[ESMLUV_SMLUVSTRANA] AS SmlStrana1
89     JOIN
90     (SELECT DISTINCT HAD_POUZITA, MIN(ID) AS MinId
91     FROM [triada].[ESMLUV_SMLUVSTRANA]
92     GROUP BY HAD_POUZITA) AS SmlStrana2
93     ON SmlStrana1.ID = SmlStrana2.MinId
94     JOIN [triada].[HAD_POUZITA] AS Had
95     ON SmlStrana2.HAD_POUZITA = Had.ID_POUZITA
96     " " " .
97
98 <#ContractValidTableView> rr:sqlQuery " " "
99     SELECT Smlouva.ID, Verze.PORADIVERZE,
100     IIF(Smlouva.AKTUALNIVERZE = Verze.ID, 'true', 'false') As Valid
101     FROM [triada].[ESMLUV_SMLOUVA] AS Smlouva JOIN [triada].[ESMLUV_VERZESMLOUVY] As Verze ON Smlouva.ID = Verze.SMLOUVA

```

```

102 WHERE Smlouva.RODIC is NULL
103 " " " .
104
105 <#ContractFilesTableView> rr:sqlQuery " " "
106 SELECT Smlouva.ID, Verze.PORADIVERZE, Soubor.NAZEVSOUBORU, Soubor.
    SADADULULOZISTEID
107 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
108 JOIN [triada].[ESMLUV_PRILOHASMLOUVY] AS Soubor ON Verze.SOUBOR =
    Soubor.ID
109 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.
    SMLOUVA
110 WHERE Smlouva.RODIC is NULL
111 " " " .
112
113 <#ContractResponsiblePersons1TableView> rr:sqlQuery " " "
114 SELECT Smlouva.ID, Verze.PORADIVERZE,
115 LTRIM(CONCAT(
116 ISNULL(Uzivatel.TITUL_PRED, '' ), ' ',
117 ISNULL(Uzivatel.JMENO, '' ), ' ',
118 ISNULL(Uzivatel.PRIJMENI, '' ), ' ',
119 ISNULL(Uzivatel.TITUL_ZA, '' )) AS CeleJmeno
120 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
121 JOIN [triada].[ESMLUV_EXTKONTAKT] AS ExtKontakt ON Verze.ID =
    ExtKontakt.VERZESMLOUVY
122 JOIN [triada].[TRL_UZIVATEL] AS Uzivatel ON Uzivatel.CISLO =
    ExtKontakt.UZIVATEL
123 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.
    SMLOUVA
124 WHERE Smlouva.RODIC is NULL
125 " " " .
126
127 <#ContractResponsiblePersons2TableView> rr:sqlQuery " " "
128 SELECT Smlouva.ID, Verze.PORADIVERZE,
129 LTRIM(CONCAT(
130 ISNULL(ExtKontakt.TITUL_PRED, '' ), ' ',
131 ISNULL(ExtKontakt.JMENO, '' ), ' ',
132 ISNULL(ExtKontakt.PRIJMENI, '' ), ' ',
133 ISNULL(ExtKontakt.TITUL_ZA, '' )) AS CeleJmeno
134 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
135 JOIN [triada].[ESMLUV_EXTKONTAKT] AS ExtKontakt ON Verze.ID =
    ExtKontakt.VERZESMLOUVY
136 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze.
    SMLOUVA
137 WHERE ExtKontakt.UZIVATEL IS NULL AND Smlouva.RODIC is NULL
138 " " " .
139
140 <#SupplementTableView> rr:sqlQuery " " "
141 SELECT Dodatek.ID, Dodatek.RODIC, Verze.PORADIVERZE,
142 VerzeDodatku.PORADIVERZE As PoradiVerzeDodatku, VerzeDodatku.
    PREDMET, VerzeDodatku.POPIS, POPIS,
143 VerzeDodatku.DATUMPODPISU, VerzeDodatku.DATUMZMENYSTAVU_TS,
144 (CASE VerzeDodatku.ANONYMIZOVANO
145     WHEN 'T' THEN 'true'
146     WHEN 'F' THEN 'false'
147     END) AS Anonymizovano
148 FROM [triada].[ESMLUV_SMLOUVA] AS Dodatek
149 JOIN [triada].[ESMLUV_VERZESMLOUVY] As VerzeDodatku ON Dodatek.ID
    = VerzeDodatku.SMLOUVA

```



```

150 JOIN [triada].[ESMLUV_SMLOUVA] As Smlouva ON Dodatek.RODIC =
      Smlouva.ID
151 JOIN [triada].[ESMLUV_VERZESMLOUVY] As Verze ON Smlouva.ID = Verze
      .SMLOUVA
152 WHERE Dodatek.RODIC is not NULL
153 " " " .
154
155 <#SupplementValidTableView> rr:sqlQuery " " "
156 SELECT Smlouva.ID, Verze.PORADIVERZE,
157 IIF(Smlouva.AKTUALNIVERZE = Verze.ID, 'true', 'false') As Valid
158 FROM [triada].[ESMLUV_SMLOUVA] AS Smlouva JOIN [triada].[
      ESMLUV_VERZESMLOUVY] As Verze ON Smlouva.ID = Verze.SMLOUVA
159 WHERE Smlouva.RODIC is not NULL
160 " " " .
161
162 <#SupplementFilesTableView> rr:sqlQuery " " "
163 SELECT Smlouva.ID, Verze.PORADIVERZE, Soubor.NAZEVSOUBORU, Soubor .
      SADADULULOZISTEID
164 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
165 JOIN [triada].[ESMLUV_PRILOHASMLOUVY] AS Soubor ON Verze.SOUBOR =
      Soubor.ID
166 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze .
      SMLOUVA
167 WHERE Smlouva.RODIC is not NULL
168 " " " .
169
170 <#SupplementResponsiblePersons1TableView> rr:sqlQuery " " "
171 SELECT Smlouva.ID, Verze.PORADIVERZE,
172 LTRIM(CONCAT(
173 ISNULL(Uzivatel.TITUL_PRED, '' ), ' ' ,
174 ISNULL(Uzivatel.JMENO, '' ), ' ' ,
175 ISNULL(Uzivatel.PRIJMENI, '' ), ' ' ,
176 ISNULL(Uzivatel.TITUL_ZA, '' ))) AS CeleJmeno
177 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
178 JOIN [triada].[ESMLUV_EXTKONTAKT] AS ExtKontakt ON Verze.ID =
      ExtKontakt.VERZESMLOUVY
179 JOIN [triada].[TRI_UZIVATEL] AS Uzivatel ON Uzivatel.CISLO =
      ExtKontakt.UZIVATEL
180 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze .
      SMLOUVA
181 WHERE Smlouva.RODIC is not NULL
182 " " " .
183
184 <#SupplementResponsiblePersons2TableView> rr:sqlQuery " " "
185 SELECT Smlouva.ID, Verze.PORADIVERZE,
186 LTRIM(CONCAT(
187 ISNULL(ExtKontakt.TITUL_PRED, '' ), ' ' ,
188 ISNULL(ExtKontakt.JMENO, '' ), ' ' ,
189 ISNULL(ExtKontakt.PRIJMENI, '' ), ' ' ,
190 ISNULL(ExtKontakt.TITUL_ZA, '' ))) AS CeleJmeno
191 FROM [triada].[ESMLUV_VERZESMLOUVY] AS Verze
192 JOIN [triada].[ESMLUV_EXTKONTAKT] AS ExtKontakt ON Verze.ID =
      ExtKontakt.VERZESMLOUVY
193 JOIN [triada].[ESMLUV_SMLOUVA] AS Smlouva ON Smlouva.ID = Verze .
      SMLOUVA
194 WHERE ExtKontakt.UZIVATEL IS NULL AND Smlouva.RODIC is not NULL
195 " " " .
196

```



```

197 <#AttachmentTableView> rr:sqlQuery """
198     SELECT DISTINCT Priloha.ID, Priloha.POPIS_NAZEVSOUBORU,
199     Priloha.SADADUL_ULOSISTEID, Priloha.OKAMZIKVYTVORENI,
200     (CASE Verze.ANONYMIZOVANO
201         WHEN 'T' THEN 'true'
202         WHEN 'F' THEN 'false'
203     END) AS Anonymizovano
204     FROM [triada].[ESMLUV_VERZESMLOUVY] As Verze
205     JOIN [triada].[ESMLUV_PRILOHASMLOUVY] As Priloha ON Verze.SOUBOR =
206     Priloha.RODIC
207     """
208 <#AttachmentToContractTableView> rr:sqlQuery """
209     SELECT Smlouva.ID As SmlouvaID, Priloha.ID, Verze.PORADIVERZE
210     FROM [triada].[ESMLUV_SMLOUVA] AS Smlouva
211     JOIN [triada].[ESMLUV_VERZESMLOUVY] As Verze ON Smlouva.ID = Verze
212     .SMLOUVA
213     JOIN [triada].[ESMLUV_PRILOHASMLOUVY] As Priloha ON Verze.SOUBOR =
214     Priloha.RODIC
215     """
216 <#MilestoneTableView> rr:sqlQuery """
217     SELECT Smlouva.ID, Verze.PORADIVERZE,
218     Milnik.ID As MilestoneID, Milnik.NAZEVS, Milnik.DATUMUCINOSTIML
219     FROM [triada].[ESMLUV_SMLOUVA] AS Smlouva
220     JOIN [triada].[ESMLUV_VERZESMLOUVY] As Verze ON Smlouva.ID = Verze
221     .SMLOUVA
222     JOIN [triada].[ESMLUV_MILNIK] As Milnik ON Verze.ID = Milnik.
223     VERZESMLOUVY
224     """
225 <#Publisher> a rr:TriplesMap;
226     rr:logicalTable <#PublisherTableView>;
227     rr:subjectMap [
228         rr:template "http://localhost:7598/publisher";
229         rr:class foaf:Organization;
230     ];
231     rr:predicateObjectMap [
232         rr:predicate gr:legalName;
233         rr:objectMap [ rr:column "[NAZEVSORGANIZACE]" ];
234     ];
235     rr:predicateObjectMap [
236         rr:predicate cn:noID;
237         rr:objectMap [
238             rr:column "[NoID]";
239             rr:datatype xsd:boolean;
240         ];
241     ];
242     rr:predicateObjectMap [
243         rr:predicate dc:identifier;
244         rr:objectMap [ rr:column "[ICO]" ];
245     ];
246     rr:predicateObjectMap [
247         rr:predicate owl:sameAs;
248         rr:objectMap [ rr:template "http://linked.opendata.cz/
249         resource/business-entity/CZ{ICO}" ];
250     ];

```

```

248 rr:predicateObjectMap [
249     rr:predicate schema:addressCountry;
250     rr:objectMap [ rr:constant "CZE" ];
251 ];
252 rr:predicateObjectMap [
253     rr:predicate cn:authentication;
254     rr:objectMap [ rr:constant "email" ];
255 ].
256
257 <#Contract> a rr:TriplesMap;
258 rr:logicalTable <#ContractsTableView>;
259 rr:subjectMap [
260     rr:template "http://localhost:7598/contract/{ID}/{PORADIVERZE}";
261     rr:class cn:Contract;
262 ];
263 rr:predicateObjectMap [
264     rr:predicate dcmi:type;
265     rr:objectMap [ rr:constant "Smlouva" ];
266 ];
267 rr:predicateObjectMap [
268     rr:predicate cn:anonymised;
269     rr:objectMap [
270         rr:column "[Anonymizovano]";
271         rr:datatype xsd:boolean;
272     ];
273 ];
274 rr:predicateObjectMap [
275     rr:predicate dc:title;
276     rr:objectMap [ rr:column "[PREDMET]" ];
277 ];
278 rr:predicateObjectMap [
279     rr:predicate dc:description;
280     rr:objectMap [ rr:column "[POPIS_POPIS]" ];
281 ];
282 rr:predicateObjectMap [
283     rr:predicate gr:hasCurrencyValue;
284     rr:objectMap [
285         rr:column "[CELKOVACASTKA]";
286         rr:datatype xsd:float;
287     ];
288 ];
289 rr:predicateObjectMap [
290     rr:predicate gr:hasCurrency;
291     rr:objectMap [ rr:column "[ZKRATKA]" ];
292 ];
293 rr:predicateObjectMap [
294     rr:predicate dc:created;
295     rr:objectMap [
296         rr:column "[DATUMPODPISU]";
297         rr:datatype xsd:date;
298     ];
299 ];
300 rr:predicateObjectMap [
301     rr:predicate ps:validFrom;
302     rr:objectMap [
303         rr:column "[DATUMUCINOSTI]";
304         rr:datatype xsd:date;

```

```

305     ];
306 ];
307 rr:predicateObjectMap [
308     rr:predicate ps:validUntil;
309     rr:objectMap [
310         rr:column "[DATUMUKONCENI]";
311         rr:datatype xsd:date;
312     ];
313 ];
314 rr:predicateObjectMap [
315     rr:predicate cn:priceAnnual;
316     rr:objectMap [
317         rr:template "false";
318         rr:datatype xsd:boolean;
319     ];
320 ];
321 rr:predicateObjectMap [
322     rr:predicate cn:funding;
323     rr:objectMap [ rr:constant "vlastní" ];
324 ];
325 rr:predicateObjectMap [
326     rr:predicate cn:awardID;
327     rr:objectMap [ rr:column "[EVIDENCNICISLOZAKAZKY]" ];
328 ];
329 rr:predicateObjectMap [
330     rr:predicate pc:publicContract;
331     rr:objectMap [ rr:template "http://linked.opendata.cz/resource
332 /domain/buyer-profiles/contract/cz/{EVIDENCNICISLOZAKAZKY}" ];
333 ];
334 rr:predicateObjectMap [
335     rr:predicate cn:awardProfileID;
336     rr:objectMap [ rr:column "[EVIDENCNICISLOFORMULARE]" ];
337 ];
338 rr:predicateObjectMap [
339     rr:predicate dc:publisher;
340     rr:objectMap [ rr:template "http://localhost:7598/contract/{ID
341 }/{PORADIVERZE}/publisher" ];
342 ];
343 rr:predicateObjectMap [
344     rr:predicate cn:version;
345     rr:objectMap [ rr:template "http://localhost:7598/contract/{ID
346 }/{PORADIVERZE}/version" ];
347 ];
348 ];
349 <#ContractType> a rr:TriplesMap;
350 rr:logicalTable <#ContractTypesTableView>;
351 rr:subjectMap [
352     rr:template "http://localhost:7598/contract/{ID}/{
353 PORADIVERZE}"
354 ];
355 rr:predicateObjectMap [
356     rr:predicate cn:contractType;
357     rr:objectMap [ rr:column "[Typ]" ];
358 ];
359 rr:predicateObjectMap [
360     rr:predicate cn:competency;
361     rr:objectMap [ rr:column "[Kompetence]" ];
362 ];

```

```

359
360 <#ContractValid> a rr:TriplesMap;
361     rr:logicalTable <#ContractValidTableView>;
362     rr:subjectMap [
363         rr:template "http://localhost:7598/contract/{ID}/{
PORADIVERZE}"
364     ];
365     rr:predicateObjectMap [
366         rr:predicate cn:valid;
367         rr:objectMap [
368             rr:column "[Valid]";
369             rr:datatype xsd:boolean;
370         ];
371     ].
372
373 <#ContractFile> a rr:TriplesMap;
374     rr:logicalTable <#ContractFilesTableView>;
375     rr:subjectMap [
376         rr:template "http://localhost:7598/contract/{ID}/{
PORADIVERZE}"
377     ];
378     rr:predicateObjectMap [
379         rr:predicate com:contentUrl;
380         rr:objectMap [ rr:template "http://localhost:7598/file/{
SADADULULOZISTEID}/{NAZEVSOUBORU}" ];
381     ].
382
383 <#ContractResponsiblePerson1> a rr:TriplesMap;
384     rr:logicalTable <#ContractResponsiblePersons1TableView>;
385     rr:subjectMap [
386         rr:template "http://localhost:7598/contract/{ID}/{
PORADIVERZE}"
387     ];
388     rr:predicateObjectMap [
389         rr:predicate cn:responsiblePerson;
390         rr:objectMap [ rr:column "[CeleJmeno]" ];
391     ].
392
393 <#ContractResponsiblePerson2> a rr:TriplesMap;
394     rr:logicalTable <#ContractResponsiblePersons2TableView>;
395     rr:subjectMap [
396         rr:template "http://localhost:7598/contract/{ID}/{
PORADIVERZE}"
397     ];
398     rr:predicateObjectMap [
399         rr:predicate cn:responsiblePerson;
400         rr:objectMap [ rr:column "[CeleJmeno]" ];
401     ].
402
403 <#ContractVersion> a rr:TriplesMap;
404     rr:logicalTable <#ContractsTableView>;
405     rr:subjectMap [
406         rr:template "http://localhost:7598/contract/{ID}/{
PORADIVERZE}/version";
407         rr:class cn:Version;
408     ];
409     rr:predicateObjectMap [
410         rr:predicate cn:uri;

```

```

411     rr:objectMap [ rr:template "http://localhost:7598/contract
412     /{ID}/{PORADIVERZE}" ];
413 rr:predicateObjectMap [
414     rr:predicate cn:versionOrder;
415     rr:objectMap [
416     rr:column "[PORADIVERZE]";
417     rr:datatype xsd:integer;
418     ];
419 ];
420 rr:predicateObjectMap [
421     rr:predicate dc:issued;
422     rr:objectMap [
423     rr:column "[DATUMZMENYSTAVULIS]";
424     rr:datatype xsd:dateTime;
425     ];
426 ].
427
428 <#Parties> a rr:TriplesMap;
429     rr:logicalTable <#PartiesTableView>;
430     rr:subjectMap [
431     rr:template "http://localhost:7598/contract/{ID}/{
432     PORADIVERZE}"
433     ];
434     rr:predicateObjectMap [
435     rr:predicate cn:party;
436     rr:objectMap [ rr:template "http://localhost:7598/party/{
437     HAD_POUZITA}" ];
438     ];
439
440 <#Party> a rr:TriplesMap;
441     rr:logicalTable <#PartyTableView>;
442     rr:subjectMap [
443     rr:template "http://localhost:7598/party/{HAD_POUZITA}";
444     rr:class gr:BusinessEntity;
445     ];
446     rr:predicateObjectMap [
447     rr:predicate gr:legalName;
448     rr:objectMap [ rr:column "[NAZEV_SUBJEKTU]" ];
449     ];
450     rr:predicateObjectMap [
451     rr:predicate dc:identifier;
452     rr:objectMap [ rr:column "[ICO]" ];
453     ];
454     rr:predicateObjectMap [
455     rr:predicate owl:sameAs;
456     rr:objectMap [ rr:template "http://linked.opendata.cz/
457     resource/business-entity/CZ{ICO}" ];
458     ];
459     rr:predicateObjectMap [
460     rr:predicate cn:noID;
461     rr:objectMap [
462     rr:column "[NoID]";
463     rr:datatype xsd:boolean;
464     ];
465     ];
466     rr:predicateObjectMap [
467     rr:predicate cn:localID;

```

```

465     rr:objectMap [
466         rr:column "[HAD_POUZITA]";
467         rr:datatype xsd:integer;
468     ];
469 ];
470 rr:predicateObjectMap [
471     rr:predicate schema:address;
472     rr:objectMap [ rr:template "http://localhost:7598/party/{
HAD_POUZITA}/address" ];
473 ];
474 rr:predicateObjectMap [
475     rr:predicate schema:addressCountry;
476     rr:objectMap [ rr:column "[STAT]" ];
477 ];
478 rr:predicateObjectMap [
479     rr:predicate gr:valueAddedTaxIncluded;
480     rr:objectMap [
481         rr:column "[PlatceDPH]";
482         rr:datatype xsd:boolean;
483     ];
484 ].
485
486 <#Address> a rr:TriplesMap;
487     rr:logicalTable <#PartyTableView>;
488     rr:subjectMap [
489         rr:template "http://localhost:7598/party/{HAD_POUZITA}/
address";
490         rr:class schema:PostalAddress;
491     ];
492     rr:predicateObjectMap [
493         rr:predicate schema:streetAddress;
494         rr:objectMap [
495             rr:template "{ULICE} {CISLA}";
496             rr:termType rr:Literal;
497         ];
498     ];
499     rr:predicateObjectMap [
500         rr:predicate schema:postalCode;
501         rr:objectMap [
502             rr:column "[PSC]";
503             rr:datatype xsd:integer;
504         ];
505     ];
506     rr:predicateObjectMap [
507         rr:predicate schema:addressLocality;
508         rr:objectMap [ rr:column "[MESTO]" ];
509     ].
510
511 <#ContractSupplements> a rr:TriplesMap;
512     rr:logicalTable <#SupplementTableView>;
513     rr:subjectMap [
514         rr:template "http://localhost:7598/contract/{RODIC}/{
PORADIVERZE}"
515     ];
516     rr:predicateObjectMap [
517         rr:predicate cn:amendment;
518         rr:objectMap [ rr:template "http://localhost:7598/amendment/{
ID}/{PoradiVerzeDodatku}" ];

```

```

519     ].
520
521 <#ContractAttachments> a rr:TriplesMap;
522     rr:logicalTable <#AttachmentToContractTableView>;
523     rr:subjectMap [
524         rr:template "http://localhost:7598/contract/{SmlouvaID}/{
PORADIVERZE}"
525     ];
526     rr:predicateObjectMap [
527         rr:predicate cn:attachment;
528         rr:objectMap [ rr:template "http://localhost:7598/attachment/{
ID}/1" ];
529     ].
530
531 <#Supplement> a rr:TriplesMap;
532     rr:logicalTable <#SupplementTableView>;
533     rr:subjectMap [
534         rr:template "http://localhost:7598/amendment/{ID}/{
PoradiVerzeDodatku}";
535         rr:class cn:Amendment;
536     ];
537     rr:predicateObjectMap [
538         rr:predicate dcmi:type;
539         rr:objectMap [ rr:constant "Dodatek" ];
540     ];
541     rr:predicateObjectMap [
542         rr:predicate cn:anonymised;
543         rr:objectMap [
544             rr:column "[Anonymizovano]";
545             rr:datatype xsd:boolean;
546         ];
547     ];
548     rr:predicateObjectMap [
549         rr:predicate dc:title;
550         rr:objectMap [ rr:column "[PREDMET]" ];
551     ];
552     rr:predicateObjectMap [
553         rr:predicate cn:contract;
554         rr:objectMap [ rr:template "http://localhost:7598/contract/{
RODIC}/{PORADIVERZE}" ];
555     ];
556     rr:predicateObjectMap [
557         rr:predicate dc:identifier;
558         rr:objectMap [ rr:column "[ID]" ];
559     ];
560     rr:predicateObjectMap [
561         rr:predicate dc:created;
562         rr:objectMap [
563             rr:column "[DATUMPODPISU]";
564             rr:datatype xsd:date;
565         ];
566     ];
567     rr:predicateObjectMap [
568         rr:predicate dc:publisher;
569         rr:objectMap [ rr:template "http://localhost:7598/amendment/{
ID}/{PoradiVerzeDodatku}/publisher" ];
570     ];
571     rr:predicateObjectMap [

```

```

572     rr:predicate cn:version;
573     rr:objectMap [ rr:template "http://localhost:7598/amendment/{
ID}/{PoradiVerzeDodatku}/version" ];
574 ];
575
576 <#SupplementValid> a rr:TriplesMap;
577     rr:logicalTable <#SupplementValidTableView>;
578     rr:subjectMap [
579         rr:template "http://localhost:7598/amendment/{ID}/{
PORADIVERZE}"
580     ];
581     rr:predicateObjectMap [
582         rr:predicate cn:valid;
583         rr:objectMap [
584             rr:column "[Valid]";
585             rr:datatype xsd:boolean;
586         ];
587     ].
588
589 <#SupplementFile> a rr:TriplesMap;
590     rr:logicalTable <#SupplementFilesTableView>;
591     rr:subjectMap [
592         rr:template "http://localhost:7598/amendment/{ID}/{
PORADIVERZE}"
593     ];
594     rr:predicateObjectMap [
595         rr:predicate com:contentUrl;
596         rr:objectMap [ rr:template "http://localhost:7598/file/{
SADADUL_ULOSTEID}/{NAZEVSOUBORU}" ];
597     ].
598
599 <#SupplementResponsiblePerson1> a rr:TriplesMap;
600     rr:logicalTable <#SupplementResponsiblePersons1TableView>;
601     rr:subjectMap [
602         rr:template "http://localhost:7598/amendment/{ID}/{
PORADIVERZE}"
603     ];
604     rr:predicateObjectMap [
605         rr:predicate cn:responsiblePerson;
606         rr:objectMap [ rr:column "[CeleJmeno]" ];
607     ].
608
609 <#SupplementResponsiblePerson2> a rr:TriplesMap;
610     rr:logicalTable <#SupplementResponsiblePersons2TableView>;
611     rr:subjectMap [
612         rr:template "http://localhost:7598/amendment/{ID}/{
PORADIVERZE}"
613     ];
614     rr:predicateObjectMap [
615         rr:predicate cn:responsiblePerson;
616         rr:objectMap [ rr:column "[CeleJmeno]" ];
617     ].
618
619 <#SupplementVersion> a rr:TriplesMap;
620     rr:logicalTable <#SupplementTableView>;
621     rr:subjectMap [
622         rr:template "http://localhost:7598/amendment/{ID}/{
PoradiVerzeDodatku}/version";

```



```

623     rr:class cn:Version;
624 ];
625 rr:predicateObjectMap [
626     rr:predicate cn:uri;
627     rr:objectMap [ rr:template "http://localhost:7598/amendment
/{ID}/{PoradiVerzeDodatku}" ];
628 ];
629 rr:predicateObjectMap [
630     rr:predicate cn:versionOrder;
631     rr:objectMap [
632         rr:column "[PoradiVerzeDodatku]";
633         rr:datatype xsd:integer;
634     ];
635 ];
636 rr:predicateObjectMap [
637     rr:predicate dc:issued;
638     rr:objectMap [
639         rr:column "[DATUMZMENYSTAVUTS]";
640         rr:datatype xsd:dateTime;
641     ];
642 ].
643
644 <#Attachment> a rr:TriplesMap;
645     rr:logicalTable <#AttachmentTableView>;
646     rr:subjectMap [
647         rr:template "http://localhost:7598/attachment/{ID}/1";
648         rr:class cn:Attachment;
649     ];
650     rr:predicateObjectMap [
651         rr:predicate dcmi:type;
652         rr:objectMap [ rr:constant "Příloha" ];
653     ];
654     rr:predicateObjectMap [
655         rr:predicate cn:anonymised;
656         rr:objectMap [
657             rr:column "[Anonymizovano]";
658             rr:datatype xsd:boolean;
659         ];
660     ];
661     rr:predicateObjectMap [
662         rr:predicate dc:title;
663         rr:objectMap [ rr:column "[POPIS_NAZEV]" ];
664     ];
665     rr:predicateObjectMap [
666         rr:predicate dc:identifier;
667         rr:objectMap [ rr:column "[ID]" ];
668     ];
669     rr:predicateObjectMap [
670         rr:predicate cn:valid;
671         rr:objectMap [
672             rr:template "true";
673             rr:datatype xsd:boolean;
674         ];
675     ];
676     rr:predicateObjectMap [
677         rr:predicate com:contentUrl;
678         rr:objectMap [ rr:template "http://localhost:7598/file/{
SADADUL_ULOVISTEID}/{NAZEVSOUBORU}" ];

```

```

679 ];
680 rr:predicateObjectMap [
681     rr:predicate dc:publisher;
682     rr:objectMap [ rr:template "http://localhost:7598/attachment/{
ID}/1/publisher" ];
683 ];
684 rr:predicateObjectMap [
685     rr:predicate cn:version;
686     rr:objectMap [ rr:template "http://localhost:7598/attachment/{
ID}/1/version" ];
687 ].
688
689 <#AttachmentToContract> a rr:TriplesMap;
690     rr:logicalTable <#AttachmentToContractTableView>;
691     rr:subjectMap [
692         rr:template "http://localhost:7598/attachment/{ID}/1"
693     ];
694     rr:predicateObjectMap [
695         rr:predicate cn:contract;
696         rr:objectMap [ rr:template "http://localhost:7598/contract/{
SmlouvaID}/{PORADIVERZE}" ];
697     ].
698
699 <#AttachmentVersion> a rr:TriplesMap;
700     rr:logicalTable <#AttachmentTableView>;
701     rr:subjectMap [
702         rr:template "http://localhost:7598/attachment/{ID}/1/version
";
703     ];
704     rr:class cn:Version;
705     rr:predicateObjectMap [
706         rr:predicate cn:uri;
707         rr:objectMap [ rr:template "http://localhost:7598/
attachment/{ID}/1" ];
708     ];
709     rr:predicateObjectMap [
710         rr:predicate cn:versionOrder;
711         rr:objectMap [
712             rr:template "1";
713             rr:datatype xsd:integer;
714         ];
715     ];
716     rr:predicateObjectMap [
717         rr:predicate dc:issued;
718         rr:objectMap [
719             rr:column "[OKAMZIKVYTVORENI]";
720             rr:datatype xsd:dateTime;
721         ];
722     ].
723
724 <#ContractToImplementations> a rr:TriplesMap;
725     rr:logicalTable <#MilestoneTableView>;
726     rr:subjectMap [
727         rr:template "http://localhost:7598/contract/{ID}/{
PORADIVERZE}";
728     ];
729     rr:predicateObjectMap [
730         rr:predicate cn:implementation;

```

```

731     rr:objectMap [ rr:template "http://localhost:7598/contract/{ID
732     }/{PORADIVERZE}/implementation" ];
733
734 <#Implementation> a rr:TriplesMap;
735     rr:logicalTable <#MilestoneTableView>;
736     rr:subjectMap [
737         rr:template "http://localhost:7598/contract/{ID}/{
738         PORADIVERZE}/implementation";
739         rr:class cn:Implementation;
740     ];
741     rr:predicateObjectMap [
742         rr:predicate cn:milestone;
743         rr:objectMap [ rr:template "http://localhost:7598/contract/{
744         ID}/{PORADIVERZE}/implementation/milestone/{MilestoneID}" ];
745     ];
746 <#Milestone> a rr:TriplesMap;
747     rr:logicalTable <#MilestoneTableView>;
748     rr:subjectMap [
749         rr:template "http://localhost:7598/contract/{ID}/{
750         PORADIVERZE}/implementation/milestone/{MilestoneID}";
751         rr:class cn:Milestone;
752     ];
753     rr:predicateObjectMap [
754         rr:predicate dc:title;
755         rr:objectMap [ rr:column "[NAZEV]" ];
756     ];
757     rr:predicateObjectMap [
758         rr:predicate cn:dueDate;
759         rr:objectMap [
760             rr:column "[DATUMUCINOSTIML]";
761             rr:datatype xsd:dateTime;
762         ];
763     ];

```

Výpis kódu 9.2: R2RML mapovací skript