Operačný program
Efektívna
verejná správa

Európska únia
Európsky sociálny fond

MINISTERSTVO
INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA
A INFORMATIZÁCIE
SLOVENSKEJ REPUBLIKY

Výstup č. 1.1.6:

Štandardizácia dátovej transformácie

Realizačná zmluva o poskytnutí služieb a o dielo č. 445/2022

Projekt:

Zlepšenie využívania údajov vo verejnej správe

ITMS kód projektu:

314011S979

Document review and approval

Revision history

Version	Author	Date	Revision
1.0	Ceľuchová Bošanská Bárdy Janík	31.05.2023	Pripravená verzia na pripomienkovanie
1.1	Ceľuchová Bošanská Bárdy Janík	30.06.2023	Zapracovanie pripomienok
1.2	Ceľuchová Bošanská Bárdy Janík	18.07.2023	Zapracovanie pripomienok

This document has been reviewed by

	Reviewer	Date reviewed
1		
2		
3		
4		
5		

This document has been approved by

	Subject matter experts		
	Name	Signature	Date reviewed
1			
2			
3			

4
5

ZOZNAM SKRATIEK	
Skratka	**Význam**
API	Aplikačné programovacie rozhranie (Application Programming Interface)
CIP	Centrálna integračná platforma
CMÚ	Centrálny model údajov
CSV	comma-separated values
EDIW	Európska digitálna peňaženka s identitou (European Digital Identity Wallet)
eID	Elektronická identita
eIDAS	Nariadenie Európskej únie č. 910/2014 o elektronickej identifikácii a dôveryhodných službách pre elektronické transakcie na vnútornom európskom trhu.
ELT	Extract, load, transform
ESB	Enterprise Service Bus
ETL	Extract, transform, load
EU	Európska únia
GDPR	Všeobecné nariadenie o ochrane osobných údajov (General Data Protection Regulation)
HTML	HyperText Markup Language
iPaaS	Integration Platform as a Service
IS CSRÚ	Informačný systém centrálnej správy referenčných údajov
IS VS	Informačný systém verejnej správy
JSON	JavaScript Object Notation
JSON-LD	JSON pre linkované údaje (JSON for Linking Data)
JWE	JSON Web Encryption
JWS	JSON Web Signature
JWT	JSON Web Token
mID	Mobilná identita
MIRRI SR	Ministerstvo investícii, regionálneho rozvoja a informatizácie
MOU	Manažment osobných údajov

MV SR	Ministerstvo vnútra SR
OLAP	Online analytické spracovanie (Online analytical processing)
OVM	Orgán verejnej moci
PET	Technológie na zvýšenie súkromia (Privacy-enhancing Technologies)
PIMS	systémy na správu osobných informácií (Personal Information Management System)
RDF	Resource Description Framework
SDG	Jednotná digitálna brána (Single Digital Gateway)
SVG	Scalable Vector Graphics
SvM	Slovensko v mobile
OOTS	Technický systém pre jedenkrát a dosť (Once-Only Technical System)
OVM	Orgán verejnej moci
RDBMS	Systém na manažment relačných databáz (Relational database management system)
TRUSTS	Trusted Secure Data Sharing Space
URI	Jednotný referencovateľný identifikátor
VC	Overiteľné poverenia (Verifiable Credentials)
W3C	World Wide Web Consortium
XDM	XPath Data Model
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XSD	XML Schema Definition
XSL	eXtensible Stylesheet Language
XSLT	XSL Transformations

Obsah

1 Úvod a zhrnutie

Surové údaje sú pre moderné organizácie vzácnym zdrojom - podobne ako zlato ukryté hlboko v bani. Avšak predtým, ako sa dá čokoľvek vyťažiť zo surových údajov, je nevyhnutný proces **dátovej transformácie**. Surové údaje priamo zo zdroja sú často:

- nekonzistentné: Obsahujú relevantné aj irelevantné údaje.
- nepresné: Obsahujú nesprávne zadané informácie alebo chýbajúce hodnoty.
- opakujúce sa: Obsahuje duplicitné údaje.

Dátová transformácia je proces získavania dobrých a spoľahlivých údajov z rôznych zdrojov. Ide o prevod údajov z jednej štruktúry (alebo údajov bez štruktúry) do inej, aby sa dali integrovať s dátovým skladom alebo s rôznymi aplikáciami. Niekedy je potrebné zmeniť nielen štruktúru, ale aj samotný dátový model, či obohatiť údaje o nové informácie. Dátová transformácia umožňuje vystaviť údaje konzumentom údajov s rôznymi požiadavkami na dáta, ako aj pokročilým analytických nástrojom na vytváranie podkladov pre rozhodovanie, hodnotenie výkonnosti a na predpovedanie budúcich trendov.

Aby boli údaje užitočné, musia byť v dostatočnej kvalite, pretože chybné údaje s chýbajúcimi informáciami či nedôveryhodné a nespoľahlivé údaje môžu viesť k nesprávnemu rozhodovaniu a stratám. Podľa spoločnosti Gartner[1] stojí zlá kvalita údajov spoločnosti milióny dolárov ročne na stratených príjmoch – tieto peniaze sú dôkazom dôležitosti dátovej transformácie. Podľa spoločnosti KPMG[2] 71 % generálnych riaditeľov tvrdí, že nebrali do úvahy poznatky z nespoľahlivých údajov. Nie je preto prekvapením, že odborníci na business intelligence strávia 80 % času prípravou (čistením a transformáciou) údajov predtým, ako sa môžu pustiť do skutočnej analýzy[3].

Dátová transformácia je definovaná ako technický proces konverzie údajov z jedného formátu, štandardu, dátového modelu alebo štruktúry do iného - bez zmeny obsahu súborov údajov a s ich prípadným obohatením o ďalšie informácie - zvyčajne s cieľom pripraviť ich na použitie aplikáciou alebo používateľom alebo zlepšiť kvalitu údajov.

1.1 Kontext

Detailný výstup č. 1.1.6: Štandardizácia dátovej transformácie vznikol ako aktualizácia dostupných výstupov v téme transformácie údajov s ohľadom na Centrálny model údajov.

[1] Zdroj: https://www.gartner.com/smarterwithgartner/how-to-stop-data-quality-undermining-your-business, Dátum referencie: 09.05.2023

[2] Zdroj: https://kpmg.com/xx/en/home/campaigns/2019/05/the-evolution-of-the-ceo.html, Dátum referencie: 09.05.2023

[3] Zdroj: https://hbr.org/2018/08/what-data-scientists-really-do-according-to-35-data-scientists, Dátum referencie: 09.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Dokument bol pripravený v rámci projektu „Zlepšenie využívania údajov vo verejnej správe“. Tento projekt má ambíciu transformovať fungovanie inštitúcií verejnej správy tak, aby dokázali maximálne efektívne spravovať a zdieľať údaje, využívať údaje pre lepšie rozhodovanie na základe faktov a dôkazov, pre zlepšenie efektivity a adresnosti služieb na základe lepšieho využívania dát.

Projekt Zlepšenie využívania údajov vo verejnej správe realizuje Dátová kancelária verejnej správy ako špeciálna jednotka Ministerstva investícii, regionálneho rozvoja a informatizácie (ďalej aj MIRRI SR).

Výstupom dokumentu je návrh štandardov a metód pre zabezpečenie transformácie údajov, ktoré spravujú inštitúcie verejnej správy. Základným nástrojom je Talend v rámci Centrálnej integračnej platformy. Transformácia údajov musí prebiehať podľa Centrálneho modelu údajov. Dokument adresuje:

- aktualizáciu štandardu pre transformáciu údajov s využitím centrálneho modelu údajov,
- návrh metód dátovej transformácie,
- výber vhodných metód pre jednotlivé prípady použitia,
- návrh odporúčaní na aplikáciu štandardu pre informačné prostredie verejnej správy a projekty z programu Manažment údajov,
- príklady dobrej praxe aplikácie štandardu.

Výstup vznikol ako realizácia aktivity číslo 1 Manažment kvality údajov a činnosti Návrh štandardného katalógu služieb pre kvalitu údajov. Zámerom je výber a príprava vhodných štandardov a nástrojov, aby bolo jasné, ako technologicky realizovať dátovú transformáciu.

1.2 Metodika realizácie výstupu

Zanalyzovali sme možnosti a aktuálne využívanie ETL funkcionalít v nástroji Talend. Pripravili sme návody, ako efektívnejšie používať tento nástroj pri transformácii dát aj s ohľadom na vytvorený Centrálny dátový model údajov a potrebu integrovať SQL aj non-SQL (napríklad grafové dáta). Identifikovali sme potenciálne slabé stránky tohto nástroja (ako napríklad to, že natívne nepodporuje pri transformácii RDF štandard) a zanalyzovali sme, či je možné a efektívne použiť na transformáciu aj iné nástroje a skripty a algoritmy.

Jedným z hlavných rozšírení nariadenia pre pravidlá Európskej únie pre elektronickú identifikáciu, autentifikáciu, dôveryhodné služby (eIDAS) je okrem rozšírenia minimálneho zoznamu atribútov hlavne definovanie EU digitálnej peňaženky pre identitu v časti Identifikácie (European Digital Identity Wallet – ďalej EDIW).

V rámci národného projektu Manažment osobných údajov (ďalej ako MOU), ktorý definuje službu moje údaje, bolo vybudované osobné úložisko, ktoré môže využívať každá dotknutá osoba (občan alebo právnická osoba). Toto osobné úložisko bude použité ako základ pre budovanie EU elektronickej peňaženky pre EU identitu. Preto je potrebné zosúladiť integráciu údajov IS CSRÚ (IS centrálnej správy referenčných údajov) s požiadavkami EDIW. Venujeme sa minimálnemu povinnému rozsahu osvedčených atribútov, ktoré je potrebné umožniť overovať a používať aj pomocou EDIW.

Tento projekt je podporený z Európskeho sociálneho fondu.

Ďalšou dôležitou témou, ktorú pokrýva štandard pre dátové transformácie, je iniciatíva Jednotnej digitálnej brány - SDG (Single Digital Gateway). Navrhli sme mechanizmus prevodu objektov CMÚ a SDG.

Predmetný materiál obsahuje návrh štandardov a metód pre zabezpečenie transformácie údajov, ktorý sa snaží reflektovať aktuálne platnú a účinnú právnu úpravu. V prípadoch potrebných zmien sú v kapitole 8 Implementácia štandardu:

- identifikované štandardy a metódy, ktoré bude nutné do legislatívy zapracovať prostredníctvom novelizácie súvisiacich štandardizačných právnych predpisov, a to aj v súlade s európskym nariadením o interoperabilite údajov verejnej správy (návrh NARIADENIE EURÓPSKEHO PARLAMENTU A RADY, ktorým sa stanovujú opatrenia na zabezpečenie vysokej úrovne interoperability verejného sektora v celej Únii (akt o interoperabilnej Európe).
- definované organizačné a finančné hľadisko pre centrálnu a lokálnu úroveň.

Tento projekt je podporený z Európskeho sociálneho fondu.

2 Význam dátovej transformácie v rámci dátovej integrácie

Dátová transformácia je proces zmeny formátu, štruktúry, dátového modelu alebo hodnôt údajov. Pre zachovanie integrity údajov však dátová transformácia nesmie meniť informáciu obsiahnutú v údajoch. Pri projektoch analýzy údajov sa údaje môžu transformovať v dvoch fázach dátových „pipelines“. Organizácie, ktoré používajú dátové sklady, zvyčajne používajú proces ETL (extract, **transform**, load), v ktorom je transformácia údajov prostredným krokom. Údaje sa počas fázy extrakcie skopírujú zo zdroja, počas fázy transformácie sa vyčistia a štruktúrujú a potom sa počas fázy načítania presunú do dátového skladu. ETL je lineárny proces, ktorý dobre funguje s relačnými dátovými skladmi, pretože si vyžadujú dátovú transformáciu s cieľom presadiť prísnu schému a kvalitu údajov pred ich načítaním do dátového skladu. Na druhej strane proces ELT (extract, load, **transform**) sa spája s dátovými jazerami („data lakes“), ktoré prijímajú štruktúrované alebo neštruktúrované údaje. ELT predstavuje novšiu metódu na dosiahnutie integrácie údajov z celej organizácie a zabránenie vzniku dátových síl. Údaje sa extrahujú zo svojho zdroja, načítajú sa do dátového úložiska a transformujú sa „at rest“. Transformácia sa zvyčajne uskutočňuje podľa potreby v porovnaní s metódou ETL, pri ktorej sa všetky údaje transformujú pred ich uložením. Dôvodom, prečo ELT môže meniť poradie fáz, je skutočnosť, že údaje sú zvyčajne uložené v dátovom jazere, ktoré prijíma nespracované – surové údaje bez ohľadu na ich štruktúru alebo formát. To umožňuje okamžité načítanie po zachytení údajov a ich neskoršiu transformáciu na účely analýzy.

Dátová transformácia je kľúčová aj pri „enterprise integrácii“ alebo pri aplikačnej integrácii, v rámci ktorej sa prepájajú jednotlivé informačné systémy, aby spolu vedeli komunikovať a vymieňať si údaje. Keďže spravila rôzne informačné systémy vyžadujú údaje v rôznych formátoch alebo štruktúre, je nevyhnutná dátová transformácia. Enterprise integráciu možno implementovať podľa rôznych vzorcov[4], ktorým by mala byť prispôsobená aj dátová transformácia, implementovaná napríklad cez mikroslužby.

Prínosy a výzvy transformácie údajov

Dátová transformácia prináša niekoľko výhod:

- Údaje sa transformujú tak, aby boli lepšie organizované. Transformované údaje sa môžu ľahšie používať ľuďmi aj počítačmi.
- Správne naformátované a overené údaje zlepšujú kvalitu údajov a chránia aplikácie pred potenciálnymi „mínami“, ako sú nulové hodnoty, neočakávané duplicity, nesprávne indexovanie a nekompatibilné formáty či dátové modely.
- Dátová transformácia uľahčuje kompatibilitu medzi aplikáciami, informačnými systémami a rôznymi typmi údajov. Údaje používané na rôzne účely môže byť potrebné transformovať rôznymi spôsobmi.

Účinná dátová transformácia však predstavuje aj určité výzvy:

[4] Zdroj: https://www.enterpriseintegrationpatterns.com/, Dátum referencie: 17.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

- Dátová transformácia môže byť nákladná na vývoj. Náklady závisia od konkrétnej infraštruktúry, softvéru a nástrojov používaných na spracovanie údajov. Výdavky môžu zahŕňať náklady súvisiace s licenciami, výpočtovými zdrojmi a náborom potrebného personálu na vývoj riešenia pre dátovú transformáciu.
- Procesy dátovej transformácie údajov môžu byť náročné na zdroje v prevádzke. Napríklad vykonávanie transformácií v lokálnom dátovom sklade po načítaní alebo dátová transformácia údajov pred ich nahratím do aplikácií môže vytvoriť výpočtovú záťaž, ktorá spomalí ostatné operácie. Ak sa používa dátový sklad alebo dátové jazero umiestnené v cloude, je dobré transformácie vykonávať po načítaní, pretože platforma sa môže škálovať podľa dopytu.
- Nedostatok odborných znalostí a nedbalosť môžu počas dátovej transformácie spôsobiť problémy. Dátoví analytici bez príslušných odborných znalostí v danej oblasti si s menšou pravdepodobnosťou všimnú preklepy alebo nesprávne údaje či chybné dátové modely po transformácii, pretože sú menej oboznámení s rozsahom presných a prípustných hodnôt a nemajú také doménové znalosti o význame údajov.
- Organizácie môžu vykonávať transformácie, ktoré nevyhovujú ich potrebám. Môže nastať, že sa údaje zmenia na špecifický formát pre jednu aplikáciu, len aby sa potom vrátili do predchádzajúceho formátu pre inú aplikáciu.

2.1 Koncepty dátovej integrácie v rámci dátovej architektúry

Nasledujúce koncepty vznikli s nástupom veľkých dát, ktoré zmenili požiadavky na dátovú integráciu, skladovanie a analýzu údajov. Pre využitie veľkých údajov sa stále viac pracuje na budovaní podporných platforiem, ako sú podnikové dátové sklady, dátové jazerá alebo v poslednej dobe „Lakehouses“, ktoré predstavujú kombináciu a výber toho najlepšieho z dátových skladov a dátových jazier („data lakes“). Tejto téme sa viac venujeme v dokumente 4.2.1 Koncept pre zavádzanie analytického spracovania údajov do praxe. Napriek takmer tridsaťročným skúsenostiam a všetkým dostupným vyspelým technológiám mnohé dátovo orientované projekty zlyhávajú. V roku 2016 spoločnosť Gartner odhadla, že 60 % projektov zameraných na veľké údaje zlyhalo[5]. Zanedlho potom Nick Heudecker (analytik spoločnosti Gartner) uviedol, že jeho spoločnosť bola pri svojom 60-percentnom odhade príliš konzervatívna a mieru zlyhania odhadla bližšie k 85 percentám[6]. Bývalý výkonný riaditeľ spoločnosti Microsoft a bývalý riaditeľ spoločnosti Snowflake Bob Muglia tvrdí, že nemôže nájsť spokojného zákazníka, ktorý v praxi využíva Hadoop[7]. Ďalej Muglia hovorí, že počet zákazníkov, ktorí skutočne úspešne „skrotili“ Hadoop, je pravdepodobne menej ako 20 a môže to byť menej ako 10. Z toho môžeme vyvodiť, že hoci sa najväčšie organizácie na svete tešia na to, že sa stanú dátovo orientovanou organizáciou, mnohé z nich považujú výsledky za priemerné.

Existuje viacero prekážok a okolo týchto tém bolo napísaných množstvo článkov: technologický dlh starších informačných systémov, slabá dátová integrácia údajov pochádzajúcich zo separátnych zdrojov, protichodné ciele vo vnútri tej istej organizácie alebo neexistujúce ciele. Mnohé organizácie začínajú s manažmentom údajov ako

[5] Zdroj: https://designingforanalytics.com/resources/failure-rates-for-analytics-bi-iot-and-big-data-projects-85-yikes/, Dátum referencie: 22.05.2023

[6] Zdroj: https://www.datascience-pm.com/project-failures/, Dátum referencie: 22.05.2023

[7] Zdroj: https://hadoop.apache.org, Dátum referencie: 22.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

s nasadzovaním monolitu pre riešenie „všetkého“, pričom konkrétne ciele sa pridajú až neskôr ako vedľajšie iniciatívy. Ďalším veľkým problémom býva nedostatok zručností.

Čo sa môžeme naučiť z minulých neúspechov iniciatív dátových platforiem a aká architektúra by mohla vyriešiť mnohé súčasné problémy s údajmi? Zdá sa, že nová koncepcia je na ceste prevratu a poskytuje novú architektúru podnikových údajov: Nazýva sa „The Data Mesh“ – dátová sieť (kapitola 2.1.2). S touto architektúrou sa vykonal aj experiment („Proof of Concept“) v rámci implementácie Centrálnej integračnej platformy v prostredí slovenského eGovernmentu. Pre úplnosť uvádzame aj ďalšie dve konkurenčné koncepcie a ich rolu v prostredí slovenského eGovernmentu.

2.1.1 „Dátové vlákna“ („Data fabric“)

Spoločnosť Gartner používa pojem „dátové vlákna“ ako komplexný prostriedok integrácie heterogénnych údajov. Z príspevku z roku 2021 o architektúre dátových vlákien vyplynula nasledujúca definícia[8]: „Dátové vlákna“ predstavujú koncepciu, ktorá slúži ako integrovaná vrstva („fabric“) údajov a spájajúcich procesov. „Dátové vlákna“ využívajú nepretržitú analýzu existujúcich, objaviteľných a odvodených metadát na podporu návrhu, nasadenia a používania integrovaných a opakovane použiteľných údajov vo všetkých prostrediach vrátane hybridných a multicloudových platforiem. „Dátové vlákna“ využívajú ľudské aj strojové schopnosti na prístup k údajom na mieste alebo na podporu ich konsolidácie či zdieľania v prípade potreby. Táto koncepcia podporuje neustálu identifikáciu a spájanie údajov z rôznorodých aplikácií s cieľom objaviť jedinečné, pre výkon agendy relevantné vzťahy medzi dostupnými dátovými bodmi.“

Spoločnosť Gartner uverejňuje správy o manažmente údajov s využitím umelej inteligencie a o tom, ako sa mení prostredie manažmentu údajov. „Dátové vlákna“ sa tak stávajú skratkou pre novšie techniky manažmentu údajov, ktoré poskytujú viac automatizovaných možností.

Nástroj Talend, popísaný v kapitole 4.2.1.1, označuje svoju Enterprise verziu produktu ako Talend Data Fabric, čím poukazuje na túto koncepciu a jej implementáciu do praktickej platformy.

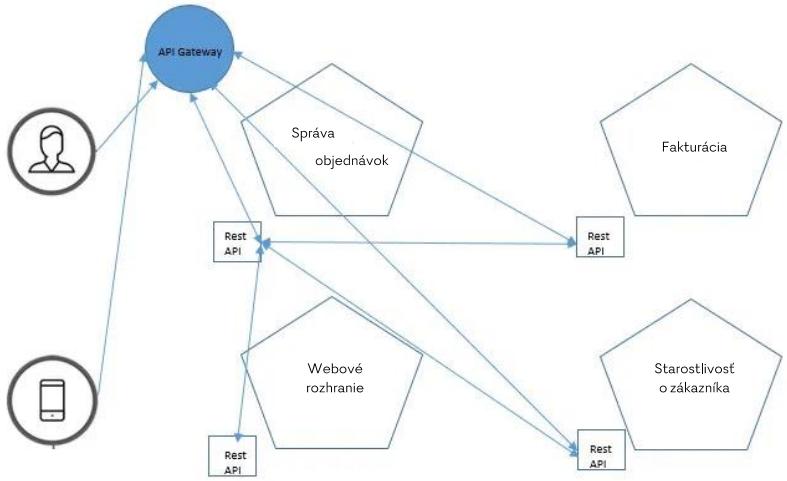
2.1.2 Dátová sieť („Data mesh“)

Riaditeľka pre nové technológie Zhamak Dehghaniová zo spoločnosti Thoughtworks a jej tím prišli s konceptom „dátovej siete“ ako distribuovanej, doménovo orientovanej alternatívy k dátovým skladom a dátovým jazerám. Koncepcia dátovej siete zahŕňa dve dátové roviny - jednu operačnú a druhú analytickú. Tieto dve roviny spája dátová pipeline (väčšinou extrakt > load). Prostriedky na reporting a vizualizáciu vrátane SQL a dashboardov sú prepojené s analytickou rovinou.

[8] Zdroj: https://www.gartner.com/smarterwithgartner/data-fabric-architecture-is-key-to-modernizing-data-management-and-integration, Dátum referencie: 17.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Potreba agilnejšieho vývoja v kontexte komplexných podnikových aplikácií spôsobila, že architektúra SOA („Service Oriented Architectre“[9]) sa vyvinula smerom k **mikroslužbám**. Existuje niekoľko kľúčových výhod, pre ktoré sa mikroslužby rozmohli. Jednou z nich je dekompozícia monolitickej architektúry. Vo svete SaaS (softvér ako služba („Software as a Service“)) sa zmeny často zavádzajú do produkcie mnohokrát denne. To je pri monolitických aplikáciách takmer nemožné. Na aktualizáciu ktorejkoľvek jej časti sa musí nasadiť celá aplikácia. V niektorých prípadoch len opätovné spustenie veľkej aplikácie môže trvať niekoľko minút (5, 10 , 40 alebo aj viac v závislosti od veľkosti a zložitosti). Mikroslužby pomáhajú tento problém vyriešiť pomocou modulárnych, menších služieb s vyššou rýchlosťou vývoja, zmien a opätovného nasadenia. Ďalším problémom, ktorý mikroslužby riešia, je škálovateľnosť. Keď majú rôzne moduly aplikácií protichodné potreby z hľadiska hardvéru, skončí to tak, že sa ohrozia tie menej dôležité. Mikroslužby umožňujú flexibilitu vo veľkej škále. Nasadenia môžu začať v malom a časom sa rozrastať aj v rôznych cloudových prostrediach. Nezávisloslé a voľne prepojené aplikačné moduly umožnia väčšiu flexibilitu pri zavádzaní nových jazykov a technologických rámcov. Pri monolitickej aplikácii by bolo veľmi bolestivé a nákladné prepísať aplikáciu s dvoma miliónmi riadkov kódu do novšieho, efektívnejšieho rámca.



Obrázok 1: Diagram architektúry mikroslužieb[10]

Napríklad vyššie uvedený Obrázok 1 ukazuje možnú implementáciu pomocou mikroslužieb. Každá funkcia aplikácie je implementovaná vlastnou mikroslužbou. Každá služba vystavuje rozhranie REST API, ktoré buď vystavuje službu, alebo konzumuje rozhranie API poskytované predchádzajúcimi službami. Niektoré rozhrania REST API nie sú priamo vystavené konzumentom (mobilným zariadeniam alebo koncovým používateľom), ale namiesto toho používajú bránu API („API Gateway“). Táto brána môže vykonávať vyrovnávanie záťaže, ukladanie do vyrovnávacej pamäte, kontrolu prístupu, monitorovanie atď. Počas behu môže byť každá inštancia kontajnerom dockera, cloudovým virtuálnym počítačom, dokonca môže jedna bežať na jednom poskytovateľovi cloudu a druhá používať iného poskytovateľa cloudu. Technológie ako

[9] Zdroj: https://www.techtarget.com/searchapparchitecture/definition/service-oriented-architecture-SOA, Dátum referencie: 22.05.2023

[10] Zdroj: https://medium.com/@ionut.bruma/the-data-mesh-a-new-paradigm-of-the-data-integration-world-7012ce799d91, Dátum referencie: 23.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Kubernetes prinášajú významné zlepšenia pri nasadzovaní služieb do viacerých cloudov.

Práca s údajmi v takomto distribuovanom a heterogénnom prostredí bola v poslednom čase náročná. Vzťah medzi aplikáciou a databázou už nie je taký priamočiary a jednoduchý. Dogma, ktorá hovorí, že každá služba má svoje vlastné zapuzdrené údaje, viedla k vážnym problémom s konzistenciou údajov, ktoré mali vplyv na niektoré agendové procesy a aplikácie, ktoré potrebovali mať istotu, že stav údajov je pri ich ukladaní konzistentný. **Keď má každá služba svoje vlastné údaje, myšlienka Centrálneho modelu údajov sa vytráca a s tým prichádza na rad duplicita údajov.** Okrem toho predpokladajme, že každá mikroslužba používa vlastný formát údajov (relačný, JSON, XML, graf, priestorový, OLAP). Z toho dôvodu potrebujeme špecifický typ databázy, ktorý je najvhodnejší pre potreby služby.

Práca s údajmi je stále náročný problém, a to najmä preto, že odklonením sa od tradičnej "monolitickej" databázy a ukrytím údajov vo vrstvách API mikroslužieb sa strácajú niektoré výhody, ktoré ponúka unitárna relačná databáza. Predstavme si, že existujú transakcie, ktoré v rámci jednej transakcie aktualizujú viaceré objekty evidencie. To znamená, že namiesto aktualizácie jednej databázy musí aplikácia aktualizovať viacero databáz, ktoré vlastnia rôzne služby. Distribuované transakcie nie sú podporované mnohými súčasnými škálovateľnými databázami NoSQL a sprostredkovateľmi správ („message brokers“). Okrem toho sa zhorší dostupnosť a odolnosť údajov. Čo sa stane, keď niekde uprostred transakcie so zapojenou jednou službou dôjde k zlyhaniu? V akom stave budú údaje uložené? Budú tieto údaje obnovené? Všetky tieto problémy si vyžadujú ošetriť biznis logikou v zdrojovom kóde. Vývojári musia napísať kód na spracovanie čiastočného zlyhania a zabezpečiť, aby údaje zostali konzistentné a mali zachovanú dostupnosť a odolnosť ako v databáze. Niektoré agendové procesy by mohli z tohto dôvodu zlyhať.

Môžeme konštatovať, že mikroslužby sú dobrá vec, ale k údajom a udalostiam („events“) treba pristupovať veľmi obozretne a treba sa vyhnúť dogmám typu „databázy sú zlé, zmenšime dátovú vrstvu“. Namiesto toho je potrebné optimalizovať interakcie medzi službou a údajmi, alebo lepšie povedané vyvinúť interakciu z „point-to-point" na oddelený prístup k údajom, ktorému sa ďalej budeme venovať. Ben Stopford, technický riaditeľ spoločnosti Confluent, vo svojom článku „Build Services on a backbone of events“ (Budovanie služieb na báze udalostí)[11] naznačil tri mechanizmy, prostredníctvom ktorých služby komunikujú s údajmi a jedna s druhou:

1. **Príkazy** („commands“) predstavujú akcie. Ide o žiadosť o vykonanie nejakej operácie v inej službe, čo zmení stav systému. Príkazy očakávajú odpoveď.
2. **Udalosti** („events“) sú skutočnosťou aj spúšťačom („trigger“). Je to niečo, čo sa stalo, vyjadrené ako notifikácia.
3. **Dotazy** („queries“) sú žiadosťou o vyhľadanie niečoho. Dôležité je, že dotazy sú bez vedľajších účinkov; ponechávajú stav systému nezmenený.

Služby by sme chceli oddeliť nielen na úrovni kódu, ale aj na úrovni údajov. V skutočnosti sa prikláňame k tvrdeniu, že čím viac bude aplikácia orientovaná na údaje, tým menej budú používatelia „uzamknutí“ („locked-in“), pretože nebudú musieť zaobchádzať s veľkými čiernymi skrinkami kódu bez toho, aby mali k dispozícii centrálny model údajov. Aby sme teda vytvorili dátovo orientovanú aplikáciu bez potreby užších prepojení medzi

[11] Zdroj: https://www.confluent.io/blog/build-services-backbone-events/, Dátum referencie: 23.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

službami, mali by sme použiť rámec riadený udalosťami. Týmto spôsobom budú služby komunikovať prostredníctvom udalostí a nie volať sa navzájom. Okrem toho sa logika presúva od poskytovateľa ku konzumentovi, pričom konzument vykoná svoju akciu na základe prijatej udalosti. Toto je dôležitý aspekt, pretože otvára akýkoľvek systém vyšším úrovniam pripojiteľnosti („pluggability“).

Ďalším aspektom, ktorý stojí za zmienku, je priamy prístup k databáze, v ktorej sa nachádza aktuálny systém záznamov. Chceli by sme nájsť spôsob, ako poskytnúť službám priamy prístup k systému záznamov a vyhnúť sa využívaniu funkčnosti služby tretej strany, ktorá má prístup k údajom. Napríklad, ak chce služba „Objednávka“ zistiť „Dostupné zásoby“, musí sa opýtať služby „Zásoby“, pretože nemá prístup k základným údajom. Prepojenie dát z viacerých tabuliek alebo databáz („joins“) je takmer nemožné bez ďalšieho prepojovania architektúry. Na vyriešenie tejto dilemy sa pomocou mechanizmu replikácie v reálnom čase posiela kópia údajov v reálnom čase každej službe lokálne, takže potom má každá služba svoju vlastnú databázu. Niekto by si mohol myslieť, že sme sa vrátili tam, odkiaľ sme odišli, aby sme mali centralizovanú databázu pre každú službu. Ale s nástupom „serverless“, databáz založených na API ani tento problém nebude taký náročný ako kedysi. Použitím tohto prístupu s komunikáciou založenou na udalostiach a lokálnymi dotazmi, môžeme dosiahnuť lepšie oddelenie, lepšiu autonómiu a efektívnejšie prepájanie údajov. Avšak s tým súvisí aj jedna nevýhoda - služby sa stanú stavovými[12]. V stavovej službe by bolo napríklad ťažké znížiť stav zásob a propagovať zmenu dostatočne rýchlo, aby ostatní účastníci v systéme mali informáciu o rovnakom množstve zásob, keď príde ďalšia objednávka (medzi objednávkami môžu byť rádovo milisekundy). Princíp jedného „zapisovateľa“ môže do istej miery vyriešiť aj tento problém. To znamená prideliť zodpovednosť za šírenie udalostí z konkrétnej domény jedinečnej službe v jej pridruženej lokálnej databáze. Okrem toho táto služba nebude len jediným zapisovateľom, ale „tímom“, ktorý sa stará o túto doménu, s prevzatím zodpovednosti za kurátorstvo týchto zdieľaných datasetov. Tým sa teda vyriešil aj problém distribuovanej správy („distributed governance“).

Teraz, keď boli opísané vzory interakcie, je zrejmé, že neexistuje žiaden zlatý grál ani dokonalý vzor („pattern“). Pri zložitých aplikáciách je zvyčajne bežné kombinovať vzory. Môžeme potrebovať kombináciu medzi flexibilitou vzdialeného dopytu a autonómiou alebo efektívnosťou lokálne uloženého a synchronizovaného datasetu. Na kombináciu týchto vzorov musíme definovať tzv. ohraničený kontext. Tento ohraničený kontext spája funkcie, ktoré majú rovnaký doménový model, ako ukazuje Obrázok 2. Pri kombinovaní udalostí s dopytmi musíme obmedziť ich rozsah na viazaný kontext. Jednotlivé klastre budú komunikovať len cez udalosti, ale vo vnútri každého klastra môžeme mať jemnejšie toky riadené udalosťami, z ktorých niektoré môžu obsahovať aj vrstvu dotazov.

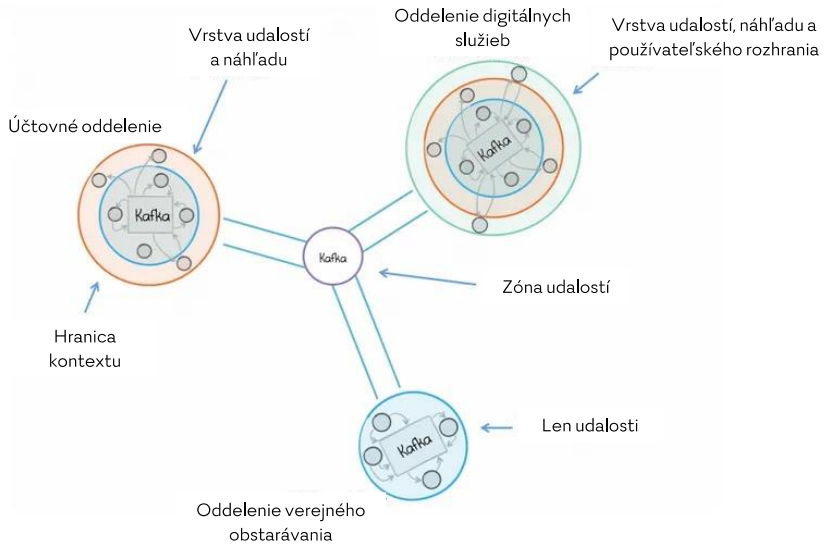
Aby sme lepšie pochopili, čo je doména, musíme porozumieť návrhu riadenému doménou („Domain-Driven Design“)[13]. Každú organizáciu, ktorá rieši nejakú rozsiahlejšiu agendu, si možno predstaviť ako doménu. Každá doména (agenda) sa zasa delí na subdomény. Subdomény zodpovedajú rôznym častiam agendy. Medzi príklady subdomén patrí napríklad: riadenie objednávok, riadenie zásob, katalóg výrobkov, riadenie vodičov. Zatiaľ to vyzerá jednoducho, ale problém nastáva pri pokuse

[12] Zdroj: https://blog.besharp.it/stateful-vs-stateless-the-good-the-bad-and-the-ugly/, Dátum referencie: 22.05.2023

[13] Zdroj: https://www.amazon.com/Domain-Driven-Design-Tackling-Complexity-Software/dp/0321125215, Dátum referencie: 22.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

o identifikáciu subdomén. Na to musíme veľmi dobre poznať kompetencie organizácie, organizačnú štruktúru, oblasti pôsobnosti vo vnútri organizácie.



Obrázok 2: Model klastrov s kontextom[14]

Koncepcia dátovej siete sa teda sústreďuje na doménovo orientovaný, a nie monolitický prístup k architektúre. Využíva výhody tvorby oddelených mikroslužieb založených na báze udalostí. Ďalším faktorom, ktorý umožňuje riešenie podobné dátovej sieti, je prechod od prístupu zameraného na zdrojový kód k prístupu zameranému na údaje, kde dátové udalosti diktujú činnosť služby, a nie volanie služby vo vlastnom kóde. Dátová sieť zachováva výhody mikroslužieb, ako je rýchlosť vývoja a zlepšené DevOps pre kontinuálnu integráciu, ale zároveň organizáciám umožňuje zaviesť vzory manažmentu údajov pre prácu so surovými dátovými udalosťami, kurátorovanie a prípravu údajov a poskytovanie kanonických údajov a master dát komukoľvek v organizácii či mimo nej. Vďaka tejto koncepcii dátovej siete sa dostávame do bodu konvergencie medzi dátovým a produktovým myslením a zavádzame novú paradigmu: „údaje ako produkt“. Vlastníci doménových údajov sa stávajú zodpovednými za poskytovanie svojich údajov ako produktu používateľom. „Údaje ako produkt“ znamenajú, že ponúkané údaje majú potrebnú kvalitu, integritu, dostupnosť atď. na to, aby sa na ne konzumenti mohli spoľahnúť. Dátový produkt je podľa tejto koncepcie „architektonické kvantum“. Takýto produkt je najmenšia jednotka architektúry, ktorá môže sama o sebe fungovať súdržne. Každý produkt z tohto dôvodu obsahuje vlastný kód, údaje, metadáta a infraštruktúru. V neposlednom rade tento prístup k architektúre používa návrh riadený doménou na oddelenie rôznych kontextov a klastrov.

Koncept dátovej siete je zatiaľ v počiatočnom štádiu. Podniky, ktoré skúmali dátové siete, hovoria, že dátová sieť nie je cieľ, ale cesta. Veľkú časť počiatočného úsilia tvorí hľadanie, určovanie vhodných a najlepších prípadov použitia a prideľovanie zdrojov, ktoré si dátová sieť vyžaduje. Celkovo sa táto koncepcia zdá byť sľubná a obsahuje veľa

[14] Zdroj: https://www.confluent.io/blog/build-services-backbone-events/, Dátum referencie: 22.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

odpovedí na súčasné výzvy v oblasti manažmentu údajov. Na to, aby sa koncepcia stala skutočnosťou, sú potrebné správne nástroje na jej realizáciu.

Aby každý doménový tím mohol vytvárať „údaje ako produkt“ bez toho, aby vynakladal úsilie a zručnosti na tú istú vec vo viacerých klastroch, je nevyhnutný spoločný, doménovo-agnostický technologický zásobník („stack“). Možno ho nazvať „dátová infraštruktúra ako platforma“. Vlastníctvo môže prejsť na jeden tím špecialistov pre dátovú infraštruktúru, ktorý môže nastaviť a prevádzkovať potrebné technológie na prijímanie, čistenie, spravovanie, spracovanie a ukladanie „údajov ako produktov“.

Existuje dlhý zoznam komponentov, ktoré riešia dlhý zoznam funkcionalít, ktoré platforma implementujúca architektúru dátovej siete potrebuje, aby sa stala realitou. Medzi nimi sú kľúčové:

- Bezserverové dátové platformy, ako napríklad Oracle Block Storage alebo Amazon S3,
- Kontajnerizované dátové riešenia s Kubernetes,
- Výpočtové cloudové služby,
- Dátové „ledgers“ riadené udalosťami (Kinesis, Event Hub, Kafka, Oracle GoldenGate, atď.)
- Spracovanie a analýza „streamov“, ako napríklad Oracle GoldenGate Stream Analytics alebo Apache Kafka Streaming, Apache Storm)
- Platformy pre upozornenia („alerting“),
- Rozhrania založené na API s bezserverovými databázami, ako je FaunaDB,
- Brány API („API Gateways“) na zabezpečenie, kontrolu používania a metriky,
- Dátový katalóg na riadenie „lineage“ a vyhľadávanie (tomu sa venujeme v dokumente 1.1.2 Štandardizácia pre modelovanie údajov).

2.1.3 Znalostné grafy („Knowledge Graphs“)

Pojem znalostný graf začal získavať na popularite od roku 2012, keď ho zaviedla spoločnosť Google. V priebehu uplynulého desaťročia viac ako 90 % svetových gigantov technologického sektora vytvorilo a používa znalostné grafy. Znalostné grafy využívajú lídri v oblasti farmácie, štátnej správy, finančných služieb, výroby a online maloobchodu.

Klasické a najvýznamnejšie implementácie znalostných grafov využívajú evolúcie „stacku“ sémantického webu, ktorý sa vyvíjal a dozrieval dve desaťročia. Pôvodnou inšpiráciou Tima Bernersa-Leeho pre sémantický web bol „web dát“, pričom obsah, ontológie a relačné údaje boli opísané rovnakým spôsobom ako entity + vzťah alebo ako subjekt-predmet-objekt. Na tejto implementácii je v prostredí slovenského eGovernmentu založený aj Centrálny model údajov a vytváranie a publikovanie datasetov otvorených údajov v najvyššej 5-hviezdičkovej kvalite (viac v dokumente 1.1.2 Štandardizácia pre modelovanie údajov).

Existuje mnoho ďalších konceptualizácií znalostných grafov, ktoré však nie sú pre štandard dátovej transformácie úplne relevantné. Pomerne veľa z týchto alternatív predpokladá úplnú automatizáciu prostredníctvom samostatných štatistických metód spracovania prirodzeného jazyka a súvisiacich metód.

Tento projekt je podporený z Európskeho sociálneho fondu.

2.1.4 Zhrnutie konceptov dátovej integrácie v rámci dátovej architektúry

Žiadny z uvedených konceptov nie je jednoduché zaviesť do praxe. Koncept „Dátové vlákna“ („Data fabric“) zahŕňa najmenej organizačných zmien a ponúka aj najmenej zásadných prínosov. Dátová sieť („Data mesh“) najviac zohľadňuje IT vývojárov a spôsob, akým v súčasnosti pracujú. Znalostné grafy („Knowledge Graphs“) nesú svoje dedičstvo z reprezentácie znalostí a logického programovania, čo nie sú najznámejšie témy pre tradičných IT špecialistov. Hlavný rozdiel medzi tromi prístupmi, ktoré sú popísané v kapitolách 2.1.1 až 2.1.3, súvisí s pôvodom a cieľmi jednotlivých prístupov, ako uvádza Tabuľka 1.

Tabuľka 1: Porovnanie konceptov dátovej integrácie v rámci dátovej architektúry

Prístup	Pôvod a ciele	Aktuálny stav
Dátové vlákna“ („Data fabric“)	Rozšírenie, prijatie a virtualizácia údajov dátových skladov a manažmentu údajov orientovaných na RDBMS	Postupná zmena súčasného stavu s mnohými rovnakými obmedzeniami. Chráni teritórium súčasných prevádzkovateľov údajov.
Dátová sieť („Data mesh“)	Rozdeľ a panuj podľa domén; kľúčové je prijať zásady návrhu a zaviesť kontroly na viacerých úrovniach s cieľom zabezpečiť „údaje ako produkt“ s definíciami kvality. Usilujte sa o samoobsluhu a škálovateľnosť.	Veľká vízia s niektorými užitočnými architektonickými poznatkami a riešeniami pre manažment údajov. Keď sa jedná o otázky, ktoré komunita sémantiky už dávno vyriešila, zbytočne vynalieza koleso. V čase písania tohto dokumentu je to poväčšine ašpirácia, avšak má zmysel experimentovať s týmto konceptom (viď aj príklad dobrej praxe v kapitole 7.3), obzvlášť pre prípady použitia, ktoré zahŕňajú dátovú integráciu medzi informačnými systémami založenými na aktuálne moderných technológiách.
Znalostné grafy („Knowledge Graphs“)	Zabezpečiť konzistentnú sémantiku vo všetkých údajoch a ontológiách špecifických pre danú oblasť alebo dátové modely. Oddelenia pre správu obsahu, znalostí a údajov používajú rovnakú metódu integrácie a spolupráce.	Prívrženci plne podporujú klasické W3C metódy tak, ako sa vyvíjali. Kritikom sa nie úplne pozdávajú a snažia sa ísť vlastnou cestou. Niekoľko významných úspechov, na ktoré sa dá poukázať. „Kmeňová“ lojalita medzi symbolickými (reprezentácia znalostí podľa štandardov W3C) a štatistickými (hlboké učenie / spracovanie prirodzeného jazyka) metódami brzdia širšie prijatie.

Tento projekt je podporený z Európskeho sociálneho fondu.

2.2 Modernizácia dátovej integrácie prostredníctvom nástrojov Camel a Kafka

Vzhľadom na veľký potenciál modernizácie dátovej integrácie cez dátovú sieť, ako bolo popísané v kapitole 2.1.2, ako aj s ohľadom na úspešný experiment („Proof of Concept“) s nástrojom Kafka v rámci Centrálnej integračnej platformy sa v tejto kapitole venujeme práve podpore architektúry založenej na udalostiach, a to s využitím nástrojov Kafka a Camel. Tieto nástroje navyše ponúkajú rozširujúce možnosti dátovej transformácie.

Enterprise integrácia alebo aplikačná integrácia (= Apache Camel) a streamovanie udalostí (= Apache Kafka) zohrávajú kľúčovú úlohu v každej modernej podnikovej architektúre[15]. Open-source je široko rozšírený a zvyčajne sa uprednostňuje v porovnaní s proprietárnymi riešeniami z rôznych dôvodov, vrátane zabránenia takzvaného „vendor lock-inu“. Platí to pre využívanie súkromných aj verejných cloudov.

Preto sa vynára otázka: Kedy sa má na aplikačnú integráciu používať Apache Camel alebo je lepšie riešiť streamovanie udalostí cez Apache Kafka? Alebo sa majú použiť oboje? Alebo jedno rieši aj druhé? Na tieto otázky odpovedá táto kapitola, v závere ktorej sa nachádza rozhodovací strom.

Funkcionality Apache Camel a Apache Kafka

Apache Camel a Kafka majú veľa spoločných pozitívnych aj negatívnych vlastností:

- Otvorený zdrojový kód pod licenciou Apache 2.0,
- Živá komunita a prijatie v odvetví,
- Vyspelý rámec s nasadením v podnikoch po celom svete,
- Náprava „špagetových architektúr“ typu „point-to-point“ pomocou centrálnej integračnej platformy,
- Otvorená architektúra a rozšíriteľnosť o vlastné funkcionality a konektory,
- Možnosť nasadenia v malom aj veľkom rozsahu, ako aj nasadenia s jedným uzlom pre nekritické prípady použitia,
- Prepracované a optimalizované pre nasadenie v cloude (kontajnery, Kubernetes),
- Možnosť pripojenia k akejkoľvek technológii, API, komunikačnej paradigme a SaaS,
- **Transformácia akýchkoľvek typov a formátov údajov**,
- Spracovanie transakčných a analytických výmen údajov,
- Doménovo špecifický jazyk (DSL) na spracovanie správ v čase s podobnou logikou, ako je agregácia, filtrovanie, podmienené spracovanie,
- Relatívne zložité rámce z dôvodu ich robustného súboru funkcionalít, preto nie sú vhodné na riešenie jednoduchého problému,
- Nenahrádzajú databázu, dátový sklad alebo dátové jazero.

[15] Zdroj: https://www.kai-waehner.de/blog/2022/01/28/when-to-use-apache-camel-vs-apache-kafka-for-etl-application-integration-event-streaming/, Dátum referencie: 23.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Okrem podobností majú nástroje Kafka a Camel veľmi odlišné kľúčové vlastnosti vytvorené na riešenie odlišných problémov. Preto je porovnávanie týchto dvoch nástrojov tak trochu porovnávaním jabĺk s hruškami. Niektoré menšie projekty môžu na riešenie problému použiť jeden alebo druhý nástroj, ale pri kritických podnikových projektoch sa rozdiely prejavia rýchlejšie.

2.2.1 Kedy použiť Apache Camel?

Poslanie Camel

Apache Camel je integračný rámec. Rieši konkrétny problém: dátovú integráciu medzi rôznymi aplikáciami, API, protokolmi a komunikačnými paradigmami. Tento koncept sa často nazýva aplikačná integrácia alebo podniková integrácia. Camel implementuje známe vzory podnikovej integrácie („Enterprise integration patterns (EIP)“), ktoré sú založené na princípoch zasielania správ.

Silné stránky Camel

- Platforma („backbone“) založená na udalostiach, ktorá vychádza zo známych a prijatých konceptov EIP,
- Možnosť pripojenia k takmer akémukoľvek API,
- Integrácia, spracovanie a smerovanie informácií pomocou intuitívneho doménovo špecifického jazyka so zameraním na integráciu; poskytovanie možnosti kompozície v programovacom kontexte na jemnejšiu kontrolu v kóde na vykonávanie podmienenej logiky alebo transformácie ,
- Výkonné možnosti smerovania s mnohými zabudovanými EIPs,
- Mnoho možností nasadenia (napríklad samostatne, ako webový kontajner, aplikačný server, Kubernetes prostredníctvom podprojektu Camel K),
- Odľahčená alternatíva k proprietárnym nástrojom ETL a „Enterprise Service Bus (ESB)“.

Slabé stránky Camel

- Len „smerovací stroj“, teda nie je stavaný na dlhodobé ukladanie (potrebná ďalšia cache alebo úložisko), z tohto dôvodu Camel nie je správnou voľbou pre centrálnu platformu ako Kafka,
- Žiadne spracovanie dátových tokov (ako napríklad Kafka Streams alebo Apache Flink), teda nedokáže spracovávať dáta kontinuálne „v pohybe“, preto ho treba porovnávať s nástrojmi pre ETL a ESB,
- Obmedzená škálovateľnosť, nie je stavaný na obrovské objemy údajov,
- Žiadne výkonné vizuálne kódovanie, ako majú k dispozícii proprietárne nástroje ETL/ESB/iPaaS (integračná platforma ako služba).
- Žiadna ponuka bezserverového cloudu,
- Red Hat je jediný dodávateľ, ktorý tento nástroj podporuje,
- Vytvorené na nasadenie v jednom dátovom centre alebo v jednej oblasti cloudu, nie v hybridných alebo multicloudových scenároch.

Tento projekt je podporený z Európskeho sociálneho fondu.

2.2.2 Kedy používať Apache Kafka?

Poslanie Kafky

V dátovo-riadených organizáciách platí pravidlo: „údaje v reálnom čase sú lepšie ako pomalé údaje v akomkoľvek rozsahu“. Platforma na streamovanie udalostí umožňuje spracovávať dáta v pohybe. Kafka je de facto štandardom pre streamovanie udalostí vrátane zasielania správ, integrácie údajov, spracovania dátových tokov a ukladania. Kafka poskytuje všetky funkcionality v jednej infraštruktúre škálovateľným spôsobom. Tento nástroj je spoľahlivý a umožňuje spracovávať analytické a transakčné úlohy.

Silné stránky Kafky

- Škálovateľná platforma na spracovanie miliónov správ v reálnom čase za sekundu,
- Jedinečná kombinácia posielania správ typu Pub/Sub[16], spracovania údajov, dátovej integrácie a ukladania údajov v jedinom rámci,
- Od začiatku vytvorená na obrovské objemy údajov a extrémne škálovanie, pričom jeden rámec možno použiť na transakčné (nízky objem) a analytické (vysoký objem veľkých údajov) prípady použitia,
- Podpora rôznych komunikačných protokolov,
- Skutočné oddelenie medzi producentmi a konzumentmi vďaka komponentu úložiska, čo z tohto nástroja robí de facto štandard pre architektúry mikroslužieb,
- Zaručené poradie udalostí v distribuovanom logu,
- Distribuované ukladanie a spracovanie údajov so zabudovanou odolnosťou voči chybám a obnoviteľnosťou,
- Možnosť opakovaného prehrávania udalostí,
- Rámec na dátovú integráciu pre streamované ETL,
- Rámec na spracovanie údajov pre kontinuálne bezstavové alebo stavové spracovanie tokov,
- De facto štandard pre streamovanie udalostí,
- Vytvorené s ohľadom na hybridnú a multicloudovú replikáciu dát (so zahrnutými nástrojmi, ako je MirrorMaker, a samostatnými, pokročilejšími a jednoduchšími nástrojmi, ako je Confluent Cluster Linking)
- Podpora od mnohých dodávateľov vrátane spoločností Confluent, Cloudera, IBM, Red Hat, Amazon, Microsoft a mnohých ďalších
- Zmena paradigmy: Vytvorené na spracovanie údajov v pohybe od zdroja po jeden alebo viacero cieľov.

Slabé stránky Kafky

- Zmena paradigmy: Organizácie sa musia naučiť a pochopiť pridanú hodnotu streamovania udalostí, novej kategórie softvéru, ktorá umožňuje nové prípady použitia, ale vyžaduje si aj odlišné návrhové vzory a prevádzkové prístupy,

[16] Vysvetlenie Pub/Sub tu: https://cloud.google.com/pubsub/docs/overview, Dátum referencie: 23.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

- Žiadne výkonné vizuálne kódovanie, ako ponúkajú proprietárne nástroje pre ETL/ESB/iPaaS
- Obmedzené možnosti smerovania „out-of-the-box“ (Kafka Connect SMT alebo aplikácia Kafka Streams / ksqlDB robia svoju prácu veľmi dobre, ale nie tak jednoducho ako Camel)
- Zložitá prevádzka (ak ich prevádzkujete sami namiesto použitia nástrojov tretích strán alebo ešte lepšie bezserverovej cloudovej ponuky).

Ak treba „len“ integračný rámec na smerovanie údajov zo zdroja do jedného alebo viacerých cieľov (= ETL / ESB), potom možno použiť aj Camel. Kafka však „zabije dve muchy jednou ranou“ (= dátová integrácia aj ich spracovanie v pohybe, kde je to potrebné).

2.2.3 Rozhodovací strom - Camel alebo Kafka?

V predchádzajúcich častiach sa skúmalo, kedy používať Camel a Kafku. Avšak ako bolo možné vidieť, oba nástroje sa svojimi možnosťami prekrývajú. Ako teda vybrať ten správny?

Najjednoduchší spôsob, ako sa rozhodnúť pre konkrétnu možnosť, je vylúčiť ten nástroj, ktorý nedokáže spĺňať všetky požiadavky.

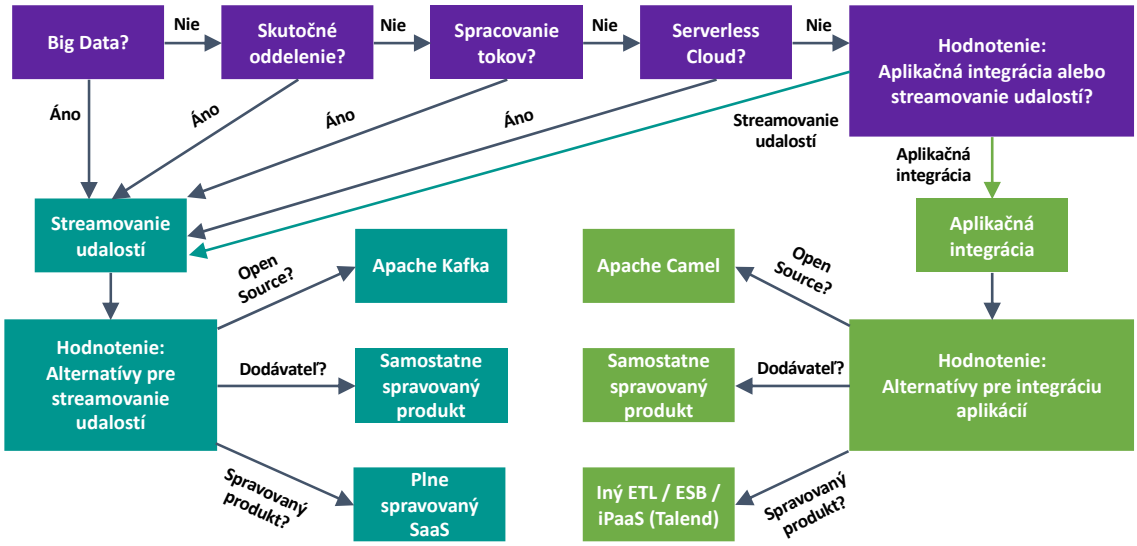
Preto, ak potrebujete:

- Spracovanie veľkých objemov údajov?
- Komponent úložiska na skutočné oddelenie a opakované prehrávanie udalostí?
- Bezstavové alebo stavové spracovanie tokov?
- Bezserverové cloudové riešenie?

Výberom je jednoznačne nástroj Kafka, pretože, ako bolo uvedené v kapitole 2.2.2, ide práve o jej rozlišovacie vlastnosti. Vo všetkých týchto prípadoch možno teda Camel diskvalifikovať, pretože tieto požiadavky nespĺňa. Nejde však nevyhnutne o úplný zoznam. A možno nájdete aj niekoľko aspektov, na základe ktorých budete môcť zas diskvalifikovať Kafku hneď na začiatku. Preto možno začať aj z pohľadu Camel a položiť si otázku: „Kedy by sa nemala používať Kafka? Avšak jednoduchšie je to naopak. Diskvalifikovanie riešení z dôvodu ich obmedzení výrazne uľahčuje rozhodovací strom a proces hodnotenia hneď od začiatku.

Obrázok 3 znázorňuje rozhodovací strom na zistenie, či je Camel alebo Kafka správna voľba a príklady ďalších nástrojov na zváženie:

Tento projekt je podporený z Európskeho sociálneho fondu.



Obrázok 3: Rozhodovací strom pre výber nástroja Camel alebo Kafka[17]

2.2.3.1 Kedy používať Camel a Kafku spoločne?

Je možné používať Camel a Kafku spoločne v jednej integračnej architektúre. Existujú dve možnosti, kedy je to dobrý nápad. Jedna dáva väčší zmysel ako druhá:

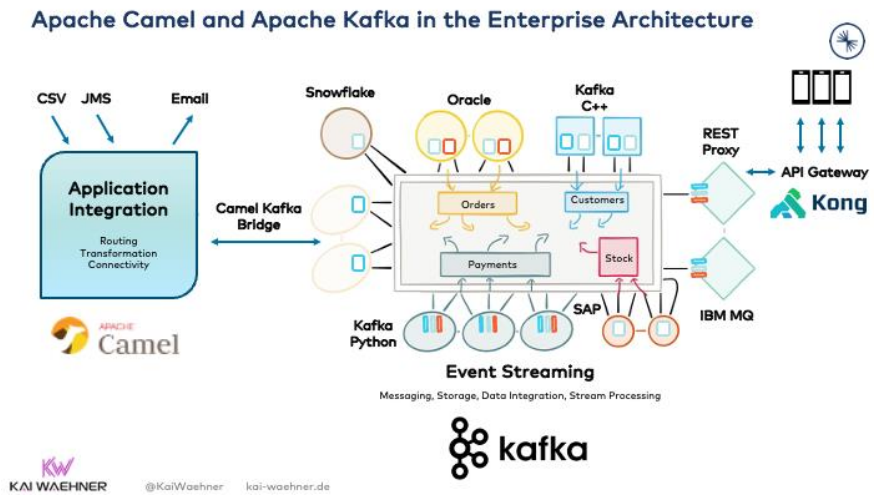
1. **Kafka na streamovanie udalostí a Camel na ETL**

Camel a Kafka sa navzájom dobre integrujú. Natívny komponent Kafky priamo v Camel je najlepším natívnym integračným bodom, ktorý slúži ako most medzi oboma prostrediami (Obrázok 4).

Uvedená architektúra (Obrázok 4) ukazuje, ako vedľa seba žijú nástroje Camel a Kafka. Camel sa používa v podnikovej doméne na aplikačnú integráciu. Kafka je centrálnou platformou medzi aplikačnou integráciou riešenou cez Camel a mnohými ďalšími aplikáciami. Na obrázku je znázornená aj API Gateway, napríklad cez nástroj Kong, aby bolo jasné, že Camel ani Kafka nedokážu vyriešiť každý problém.

[17] Zdroj: https://www.kai-waehner.de/blog/2022/01/28/when-to-use-apache-camel-vs-apache-kafka-for-etl-application-integration-event-streaming/, Dátum referencie: 22.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.



Obrázok 4: Apache Camel a Apache Kafka v podnikovej architektúre[18]

Replikácia údajov v reálnom čase v rámci hybridných a multi-cloudových systémov nie je na uvedenom obrázku (Obrázok 4) zobrazená, ale je tiež súčasťou podnikovej architektúry „out-of-the-box“ s využitím protokolu Kafka.

Vďaka skutočnému oddeleniu („decoupling“) v rámci modernej architektúry mikroslužieb sa každý tím v rámci verejnej správy môže rozhodnúť, či potrebuje aplikačnú integráciu (pomocou Camel) alebo streamovanie udalostí (pomocou Kafka). Často sa môžu použiť obidve možnosti. Na správnu voľbu pre konkrétny prípad použitia je potrebné vyhodnotiť ďalšie otázky týkajúce sa jedného verzus viacerých rámcov a API, podpory dodávateľa, potrieb škálovateľnosti a ďalších vlastností.

2. Konektory Camel vložené do Kafka Connect

Existuje ďalší spôsob, ako skombinovať Kafku a Camel cez „Camel Kafka Connector“. Ide o podprojekt Apache Camel. Je to trochu mätúce. Táto funkcia nie je komponentom Kafka (= konektorom) patriacim do Camel! Namiesto toho ide o relatívne novú iniciatívu na nasadenie komponentov Camel do infraštruktúry Kafka Connect.

Zrejmá výhoda je tá, že týmto spôsobom možno získať stovky nových konektorov „zadarmo“ v rámci ekosystému Kafka. Táto možnosť znie vynikajúco. Treba však zvážiť však celkové náklady na vlastníctvo takéhoto ekosystému a celkové úsilie pri použití tohto prístupu. Aplikačná integrácia je jedným z najnáročnejších problémov v informatike - najmä ak hovoríme o transakčných dátových súboroch, ktoré vyžadujú nulovú stratu dát, presnú sémantiku a žiadne výpadky. Čím viac komponentov sa kombinuje v „end-to-end“ dátovom toku, tým ťažšie je dodržať SLA týkajúce sa výkonu a spoľahlivosti. Preto má používanie komponentov Camel v rámci Kafka Connect značnú nevýhodu: ide o kombináciu dvoch komplexných rámcov s odlišnými koncepciami návrhu.

[18] Zdroj: https://www.kai-waehner.de/blog/2022/01/28/when-to-use-apache-camel-vs-apache-kafka-for-etl-application-integration-event-streaming/, Dátum referencie: 22.05.2023

Pred skombinovaním dvoch integračných nástrojov, ktoré sú samy o sebe výkonné, ale zložité, si to treba dvakrát dobre premyslieť. Spustenie takého ekosystému je len jednou časťou skladačky (tou jednoduchou). Netreba zabudnúť na testovanie „end-to-end“, odolnosť, SLA, podporu naprieč technológiami a API. Ani zakúpenie podpory pre Camel a Kafka od spoločnosti Red Hat (t. j. od jedného dodávateľa) tento prístup nezlepší.

Pravdepodobne by bolo lepšie vyňať biznis logiku a volania API z komponentu Camel a skopírovať ich do šablóny konektora Kafka Connect, aby bolo možné spustiť integráciu natívne len s kódom Kafka. Toto riešenie umožňuje čistú architektúru, „end-to-end“ integráciu s jediným rámcom, za ktorým stojí jediný dodávateľ, a oveľa jednoduchšie testovanie / ladenie / monitorovanie. Odporúča sa používať podprojekt "Camel Kafka Connector" len vtedy, ak nefungujú nasledujúce možnosti:

- Použitie len Apache Camel na aplikačnú integráciu,
- Využitie Apache Kafka na streamovanie udalostí a aplikačnú integráciu,
- Samostatné nasadenie Camel a Kafka a použitie Camel-Kafka-Bridge.

*2.2.3.2 Kedy vôbec **nepoužívať** Camel alebo Kafku?*

Opäť platí, že najjednoduchším spôsobom, ako začať vyhodnocovať, je vyradiť nástroje, ktoré nefungujú na riešenie daného problému. Camel aj Kafka nie sú vytvorené pre nasledujúce scenáre:

- Proxy pre milióny klientov (napríklad mobilné aplikácie) - ale pre niektoré prípady použitia existujú natívne proxy (napríklad REST alebo MQTT Proxy pre Kafku).
- Platforma na správu API - ale takéto nástroje sú zvyčajne doplnkové a používajú sa na vytvorenie správy životného cyklu alebo na speňaženie API nasadených pomocou Camel alebo Kafka.
- Databáza na komplexné dotazy a dávkové analytické úlohy,
- Platforma IoT s funkcionalitami, ako je správa zariadení - ale priama natívna integrácia s (niektorými) protokolmi IoT, ako je MQTT alebo OPC-UA, je možná a prístupná pre niektoré prípady použitia.
- Technológia pre aplikácie s prísnymi požiadavkami na fungovanie v reálnom čase, ako sú bezpečnostné alebo deterministické systémy - ale to platí aj pre akýkoľvek iný IT rámec. „Embedded“ systémy sú iným softvérom ako Camel alebo Kafka.
- Pri nestabilnom sieťovom prepojení medzi klientmi Kafky a brokermi Kafky. Ak je teda sieť nestabilná a klienti sa musia neustále opätovne pripájať k brokerom, potom je prevádzka náročná a SLA sa ťažko dosahujú.

2.2.3.3 Apache Camel vs. Apache Kafka - kto je víťaz?

Jednoduchá odpoveď: Oba! Pretože nejde o vhodné porovnanie, keďže oba nástroje slúžia veľmi odlišným potrebám, aj keď oba dokážu robiť aplikačnú integráciu. V mnohých integračných scenároch možno použiť Camel alebo Kafku.

Camel je ten správny nástroj, ak treba integrovať údaje v rámci kontextu aplikácie alebo oddelenia či sekcie organizácie (bez potreby spracovania tokov, skutočného oddelenia

Tento projekt je podporený z Európskeho sociálneho fondu.

(„decoupling“), prehrateľnosti udalostí, veľkého rozsahu, replikácie medzi dátovými centrami alebo cloudovými regiónmi).

Kafka je centrálny „nervový systém“ založený na udalostiach v rámci sekcií, organizácií, regiónov a hybridných cloudov. Kafka je o streamovaní udalostí. Aplikačná integrácia je len kúskom tejto skladačky. Existuje však množstvo integračných projektov poháňaných Apache Kafka. Často nahrádza iný middleware. Platí to pri modernizácii starších systémov založených na ETL/ESB a v diskusiách o používaní cloudovej natívnej iPaaS.

Tento projekt je podporený z Európskeho sociálneho fondu.

3 Návrh metód dátovej transformácie

Dátová transformácia má obrovský počet metód, ktoré môžu byť naprogramované úplne na mieru konkrétnemu datasetu a veľmi podrobne zadefinovanému prípadu použitia, napríklad na dátovú analýzu veľmi konkrétneho problému na základe presne definovaných premenných. V tejto časti preto uvádzame buď často využívané základné metódy alebo veľmi špecifické metódy dátovej transformácie, ktoré sú hlavne spojené s dátovou transformáciou podľa Centrálneho modelu údajov.

Najbežnejšie typy dátovej transformácie sú:

- **Konštruktívne**: Proces dátovej transformácie pridáva, kopíruje alebo replikuje údaje. Ak sa nemení informácia obsiahnutá v údajoch, možno ich použiť aj pri výmene údajov medzi OVM.
- **Deštruktívne**: Systém odstraňuje polia alebo záznamy, ak nie sú potrebné pre analýzu údajov, alebo ak ich nepotrebuje cieľový systém.
- **Estetické**: Transformácia štandardizuje údaje tak, aby spĺňali požiadavky alebo parametre.
- **Štrukturálne**: Databáza sa reorganizuje premenovaním, presunutím alebo spojením stĺpcov. Údajom sa transformáciou mení ich dátový model.

Pri dátovej transformácii sa berú do úvahy nasledujúce formáty – štandardy údajov (datsetov alebo objektov evidencie):

- Na vstupe dátovej transformácie:
  - XML,
  - JSON,
  - CSV,
- Na výstupe dátovej transformácie:
  - RDF, predovšetkým vo formáte JSON-LD a RDF/XML (viac o tomto formáte a štandardne sa nachádza v dokumente 1.1.2 Štandardizácia pre modelovanie údajov),
  - XML,
  - JSON,
  - CSV,
  - Apache Parquet[19] - predstavuje stĺpcový komprimovaný formát, ktorý sa dá efektívne využiť pre veľké údaje, napríklad v ekosystéme Hadoop[20].

Pre výstup dátovej transformácie v štandarde RDF slúžia ako predpis vzory objektov evidencie vo formáte JSON-LD alebo RDF/XML, ktoré sú v čase písania tohto dokumentu zverejnené na wiki.vicepremier.gov.sk. Pri ich tvorbe sa vychádza z Centrálneho modelu údajov a zo schválených ontológií, ktoré sú publikované na znalosti.gov.sk/metadata.

[19] Zdroj: https://parquet.apache.org, Dátum referencie: 23.05.2023

[20] Zdroj: https://hadoop.apache.org, Dátum referencie: 23.05.2023

3.1 Metódy aplikované pred dátovou transformáciou

Ide o metódy, ktoré zásadne nemenia kvalitu údajov a vôbec nemajú vplyv na informáciu obsiahnutú v jednotlivých dátových prvkoch, avšak slúžia na prípravu dát tak, aby ich bolo možné jednoduchšie transformovať podľa Centrálneho modelu údajov. Môže ísť tiež o zjednodušenie konzumovania údajov v cieľových systémoch. Centrálny model údajov má tú výhodu, že nie je potrebné prispôsobovať dátovú transformáciu každému cieľovému systému a následne ju aj realizovať. Z každého zdrojového systému sa údaje transformujú podľa Centrálneho modelu údajov. Následne sa vykoná dátová transformácia len pre tie cieľové systémy, ktoré nevedia s týmto Centrálnym modelom údajov pracovať. Takáto transformácia je už jednoduchšia, pretože Centrálny model údajov je sémanticky jednoznačne definovaný a aj strojovo zrozumiteľný.

Vo všeobecnosti sa táto metóda nazýva **obohacovanie údajov** o ďalší obsah, ktorý nemení ich informačnú hodnotu. Pri odosielaní správ alebo súborov z jedného systému do druhého je bežné, že cieľový systém vyžaduje viac informácií, ako môže poskytnúť zdrojový systém. Napríklad prichádzajúce správy o adrese môžu obsahovať len poštové smerové číslo, pretože návrhári usúdili, že ukladanie nadbytočného kódu štátu by bolo zbytočné. Je pravdepodobné, že iný systém bude chcieť špecifikovať pole s kódom štátu aj s poštovým smerovým číslom. Ďalší systém nemusí v skutočnosti používať kódy štátov, ale vypíše názov štátu, pretože používa voľný tvar adresy s cieľom podporovať medzinárodné adresy. Podobne nám jeden systém môže poskytnúť rodné číslo občana, ale cieľový systém v skutočnosti vyžaduje meno a adresu občana. Správa o objednávke odoslaná systémom správy objednávok môže obsahovať len číslo objednávky, ale my potrebujeme zistiť identifikačné číslo zamestnanca spojené s touto objednávkou, aby ďalší systém mohol túto objednávku spracovať. Takýchto rôznych prípadov je veľa.

Tieto prípady a potrebné metódy na ich vyriešenie je nutné komplexne zanalyzovať pre všetky objekty evidencie a priebežne aktualizovať podľa toho, ako sa bude ďalej vyvíjať Centrálny model údajov a požiadavky konzumentov. Momentálne boli identifikované prípady a metódy popísané v nasledujúcich podkapitolách.

Ešte je dôležité poznamenať, že dodatočné informácie vložené cez obohacovanie údajov musia byť k dispozícii niekde v systéme. Najbežnejšie zdroje týchto dodatočných informácií (údajov) sú:

- **Výpočet alebo iná biznis logika**, ktorá je schopná vypočítať alebo odvodiť chýbajúce informácie. V takom prípade algoritmus začlení dodatočné informácie do zdrojovej správy alebo súboru. Napríklad, ak cieľový systém vyžaduje dátové pole, ktoré špecifikuje vek fyzickej osoby, možno ju odvodiť z rodného čísla alebo z dátumu narodenia. V tomto prípade nie je potrebný žiadny externý zdroj údajov.
- **Prostredie**: Dodatočné údaje sa môžu dať získať z operačného prostredia. Najbežnejším príkladom je časová pečiatka. Napríklad cieľový systém môže vyžadovať, aby každá správa alebo súbor obsahoval časovú pečiatku. Ak zdrojová správa alebo súbor toto pole neobsahuje, komponent pre obohacovanie môže získať aktuálny čas z operačného systému a pridať ho do správy alebo do súboru.
- **Iný informačný systém:** Táto možnosť je najbežnejšia. Komponent pre obohacovanie musí získať chýbajúce údaje z iného systému. Tento zdroj údajov môže mať rôzne podoby vrátane databázy, súboru, adresára LDAP, systému alebo používateľa, ktorý ručne zadá chýbajúce údaje.

Tento projekt je podporený z Európskeho sociálneho fondu.

Obohacovanie údajov nám pomáha v situáciách, keď konzument správy alebo súboru vyžaduje viac - alebo iné - dátové prvky, než poskytuje tvorca správy. Existuje však prekvapivo veľa situácií, v ktorých je žiaduci opačný efekt: odstránenie dátových prvkov zo správy alebo súboru. Ako si zjednodušiť prácu s rozsiahlou správou alebo súborom, keď nás zaujíma len niekoľko dátových prvkov? Vtedy treba použiť **filtrovanie údajov** na odstránenie nedôležitých dátových prvkov zo správy alebo súboru, pričom sa ponechajú len dôležité prvky.

3.1.1 Stotožnenie číselníkových hodnôt

Číselníkové hodnoty tvoria zatiaľ väčšinu prípadov, kedy treba údaje zo zdroja spravidla vo formáte XML stotožniť so základným číselníkom, prípadne obohatiť. Nastávajú tam však rôzne prípady, ako napríklad:

1. Hodnota zo zdroja príde správne ako číselníkový kód aj s hodnotou (väčšinou textovou) – to je ideálny prípad,
2. Hodnota príde zo zdroja iba ako kód, takže treba k nej doplniť aj textový popis,
3. Hodnota príde zo zdroja ako textový popis, takže treba k nej k nej priradiť kód,
4. Hodnota príde ako lokálna, definovaná len pre daný zdroj, vtedy je potreba k nej priradiť kód aj popis zo základného číselníka - tu je príkladom pohlavie na doklade, ktoré posiela zdroj ako “M”, pričom tam má byť číselníková hodnota “1” s popisom “Muž”.

Tieto prípady sa týkajú aj číselníkov ktoré nie sú základné, pretože snahou pri transformácii je zjednotiť aj takého prípady (hlavne ak v CSRÚ tieto číselníky sú k dispozícii). Pre tieto prípady budú musieť existovať pravidlá pre priradenie URI adries takýchto číselníkov.

3.1.2 Úprava adresy

Druhým častým prípadom je zložený dátový prvok adresy, pri ktorej je potrebné riešiť nasledovné body:

1. Rozpad adresy na jednotlivé dátové prvky (položky v adrese), ak sa v XML súbore nachádza adresa len ako spoločný text na jednom riadku. Poznámka: Keď sa do XML táto adresa upraví do dobrej štruktúry so zrozumiteľne pomenovanými uzlami, tak je k dispozícii všeobecný XSLT template, ktorý bude možné použiť cez všetky transformácie (viac v kapitole 3.2).
2. Dohľadanie kódov územných jednotiek v územnej taxonómii, čo sa dá tiež považovať za prípad číselníka (3), pričom číselníky územných jednotiek sú základné číselníky. Taxonómia je už publikovaná aj na znalosti.gov.sk (teda napríklad Okres Dunajská Streda je SK0211).

Tu netreba zabúdať tiež na všemožné kombinácie toho, ako sa dá adresa napísať do jedného textového poľa, pričom niektoré dátové prvky adresy môžu chýbať, alebo môžu byť nesprávne a podobne. Parsovanie adries je samo o sebe dosť komplexnou záležitosťou, ktorú treba v rámci tejto metódy vyriešiť vo vybranom nástroje alebo v zdrojovom kóde.

Tento projekt je podporený z Európskeho sociálneho fondu.

3.1.3 Externé ukladanie a využívanie textových informácií pre obohacovanie

Niektoré textové informácie (napríklad aj spomínané textové popisy číselníkových kódov) nie je udržateľné vkladať priamo do nástroja alebo zdrojového kódu transformácie či do súborov XSLT (kapitola 3.2), najmä v prípade, keď je potrebné nejaký text opraviť. Takéto prípady by bolo potrebné spravovať externe napríklad v dátovom katalógu, kam by časom ideálne mali pribudnúť preklady do iných jazykov (minimálne anglického). Z tohto externého zdroja by sa potom v rámci metódy obohatenia tieto textové informácie vkladali do súboru zo zdroja, napríklad v XML, a transformácia by ich tam očakávala a použila.

3.1.4 Odvodenie nejakej hodnoty na základe inej hodnoty v zdrojovom XML

Príkladom tohto prípadu je potvrdenie o nedoplatkoch zo Sociálnej poisťovne. V Centrálnom modeli údajov je dátový prvok, respektíve triplet „subjet“ (fyzická alebo právnická osoba) → „predikát “ - “`hasArrears`” (má nedoplatky) → objekt (literál) – „true/false“, ktorý je možné odvodiť na základe elementu `<State>` v zdrojovom XML. Opäť je tu dôležité, aby takáto logika bola čo najmenej súčasťou konkrétnej implementácie transformácie, ako bolo uvedené v kapitole 3.1.3, a tým pádom bola evidovaná ideálne na jednom externom mieste, napríklad v dátovom katalógu, aby ju bolo možné ľahko zmeniť.

3.2 Metódy aplikované na zmenu dátového modelu v prípade zdrojového formátu XML do cieľového formátu XML alebo RDF/XML s využitím XSLT

V tejto metóde sa využíva štandard W3C „XSL Transformations (XSLT)“ vo verzii 3.0[21]. Transformácia v jazyku XSLT je vyjadrená vo forme súboru štýlov („stylesheet“). Súbor štýlov vo všeobecnosti obsahuje prvky, ktoré sú definované v XSLT, ako aj prvky, ktoré nie sú definované v XSLT. Prvky definované XSLT sa rozlišujú použitím menného priestoru („namespace“)[22] `http://www.w3.org/1999/XSL/Transform`. Rok 1999 v URI znamená, že vtedy bola od W3C priradená táto doména. V príkladoch transformácie napísaných v jazyku XSLT sa prefix `xsl:` používa na odkazovanie na prvky v mennom priestore XSLT.

XSLT sa používa na širokú škálu transformačných úloh, nielen na formátovanie a prezentáciu. V rámci tohto štandardu využívame XSLT na zmenu dátového modelu zdrojových údajov, aby bol v súlade s CMÚ. Vtedy sa pri tvorbe XSLT vychádza z definovaných JSON LD alebo RDF-XML na wiki.vicepremier.gov.sk, ako bolo spomínané v úvode tejto kapitoly 3. XSLT sa v prostredí Centrálnej integračnej platformy

[21] Zdroj: https://www.w3.org/TR/xslt-30/, Dátum referencie: 23.05.2023

[22] Zdroj: https://www.w3.org/TR/xslt-30/#xslt-namespace, Dátum referencie: 23.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

avšak dá využiť aj na zmenu štruktúry súboru zo zdroja pre potreby konzumenta alebo pre potreby prezentácie datasetu vo formátoch ako HTML, XHTML a SVG.

Transformácia vyjadrená v jazyku XSLT teda opisuje pravidlá na transformáciu vstupných údajov na výstupné údaje. Všetky vstupy a výstupy budú inštanciami dátového modelu XDM („XQuery and XPath Data Model 3.1“)[23]. V najjednoduchšom a najbežnejšom prípade je vstupom dokument XML označovaný ako zdrojový strom („source tree“) a výstupom je dokument XML označovaný ako výsledný strom („result tree“). Je možné spracovať aj viac zdrojových dokumentov, generovať viac výsledných dokumentov a spracovať aj iné formáty ako XML, napríklad aj JSON. Transformácia sa vykonáva pomocou súboru šablónových pravidiel[24]. Šablónové pravidlo spája vzor („pattern“)[25], ktorý zvyčajne zodpovedá uzlom v zdrojovom dokumente, s konštruktorom sekvencie[26]. V mnohých prípadoch vyhodnotenie konštruktora sekvencie spôsobí vytvorenie nových uzlov, ktoré sa môžu použiť na vytvorenie časti výsledného stromu. Štruktúra výsledných stromov môže byť úplne odlišná od štruktúry zdrojových stromov. Pri konštrukcii výsledného stromu sa môžu uzly zo zdrojových stromov filtrovať a meniť ich poradie a môže sa pridať ľubovoľná štruktúra. Tento mechanizmus umožňuje, aby bol súbor štýlov („stylesheet“) použiteľný na širokú triedu dokumentov, ktoré majú podobné štruktúry zdrojových stromov.

Hlavným cieľom vylepšení v XSLT 3.0 oproti verzii 2.0 je umožniť streamovanie zdrojových dokumentov. To je potrebné, keď sú zdrojové dokumenty príliš veľké na to, aby sa udržali v hlavnej pamäti, a tiež pre aplikácie, v ktorých je dôležité začať poskytovať výsledky skôr, ako je k dispozícii celý zdrojový dokument. Druhým ťažiskom vylepšení v XSLT 3.0 je zavedenie nového mechanizmu pre modularitu súborov štýlov, ktorý sa nazýva balík („package“). Balík definuje rozhranie, ktoré upravuje, ktoré funkcie, premenné, šablóny a iné komponenty sú viditeľné mimo balíka a ktoré možno prepísať. XSLT 3.0 obsahuje aj podporu máp (dátová štruktúra pozostávajúca z dvojíc kľúč/hodnota, v iných programovacích jazykoch niekedy označovaná ako slovníky, haše alebo asociatívne polia). Táto funkcia rozširuje dátový model.

Pri transformácii do RDF treba vytvárať takzvané pomenované grafy („named graphs“), kedy sa RDF grafu priradí meno – v našom prípade ID zdrojového informačného systému, na účely rozumnej správy údajov, riadenia prístupu alebo pripojenia metadát k celkovému grafu, a nie k jednotlivým uzlom. V JSON-LD to znamená, že sa kontextom ide napríklad pre dataset z Registra fyzických osôb:

```
"@id": "isvs:191",

"@graph": [
```

[23] Zdroj: https://www.w3.org/TR/xpath-datamodel-31/, Dátum referencie: 23.05.2023

[24] Zdroj: https://www.w3.org/TR/xslt-30/#rules, Dátum referencie: 23.05.2023

[25] Zdroj: https://www.w3.org/TR/xslt-30/#patterns , Dátum referencie: 23.05.2023

[26] Zdroj: https://www.w3.org/TR/xslt-30/#sequence-constructors, Dátum referencie: 23.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Používanie pomenovaných grafoch umožňuje:

- Porovnávať údaje s tou istou informáciou medzi jednotlivými informačnými systémami (rovnaké triplety môžu byť súčasťou rôznych pomenovaných grafov),
- Prehadzovať údaje medzi jednotlivými pomenovanými grafmi,
- Pomenované grafy umožňujú vytvoriť a naplniť graf a potom ho použiť na vytvorenie jedného alebo viacerých ďalších grafov, ktoré potom môžu spúšťať ďalšie akcie.
- Podmienky v rámci grafov tiež znamenajú, že príkaz DELETE/INSERT zo SPARQL UPDATE[27] môže byť aktivované, len vtedy, ak existujú správne podmienky grafu v príkaze WHERE, čo umožňuje podmienenú logiku.

V rámci transformácie v XSLT sa dá zadefinovať hodnoty aj s jednotkami aj nasledujúcim spôsobom, nielen cez základné dátový typy z `xsd: data types`:

- Ak máme dĺžkovú mieru, namiesto zadávania typu do vlastností sa dá použiť: `"25"^^qudt:Meters`.
- Ak máme informáciu o počte populácie, možno použiť `"8.01E9"^^quantity:People`.

Dajú sa teda použiť vlastné dátové typy, aby sa uviedlo, ako sa parsujú literály, a potom sa pridajú metadáta (napríklad URIs v mennom priestore `quantity` sa budú dať dereferencovať, čím sa bude dať dostať k ďalším metadátam (informáciám – tripletom) o dostupných vlastných dátových typoch v tomto mennom priestore).

Tabuľka 2 ukazuje všeobecný XSLT template pre celý uzol identifikátora, ktorý je možné použiť cez všetky transformácie, pričom vstupné parametre sú nasledovné:

- **id** - samotná hodnota identifikátora (napríklad identifikačné číslo organizácie, počítačové číslo občana, rodné číslo),
- **type** - číselníková hodnota typu identifikátora, z číselníka CL004001,
- **label** - musí obsahovať celý XML „tag“ `skos:prefLabel` s názvom, popisom hodnoty, ideálne aj s označením jazyka, napríklad:
    - `<skos:prefLabel xml:lang="sk">Občiansky preukaz</skos:prefLabel>`

Tabuľka 2: Ukážka univerzálneho XSLT pre identifikátor

```
<xsl:template name="admsIdentifier">
    <xsl:param name="id"/>
    <xsl:param name="type"/>
    <xsl:param name="label"/>
    <adms:identifier>
        <adms:Identifier>
            <skos:notation
            rdf:datatype="http://www.w3.org/2001/XMLSchema#string"><xsl:value-of
            select="$id"/></skos:notation>
```

[27] Zdroj: https://www.w3.org/TR/sparql11-update/#updateLanguage, Dátum referencie: 30.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

```
                    <dct:type>
                        <egov:IdentifierType
                        rdf:about="{concat('https://data.gov.sk/def/identifier-
                        type/',$type)}">
                            <skos:inScheme          rdf:resource="{concat($codelistNs,
                            'CL004001')}"/>
                            <!-- <xsl:apply-templates select="$label"/> -->
                            <xsl:copy-of select="$label" />
                        </egov:IdentifierType>
                    </dct:type>
                    <egov:issuingCountry>
                        <xsl:call-template name="uncountrySlovakRepublic"/>
                    </egov:issuingCountry>
            </adms:Identifier>
        </adms:identifier>
</xsl:template>

<xsl:template name="uncountrySlovakRepublic">
        <loca:UNCountry rdf:about="https://data.gov.sk/def/uncountry/703">
            <skos:prefLabel xml:lang="sk">Slovenská republika</skos:prefLabel>
            <skos:prefLabel xml:lang="en">Slovak Republic</skos:prefLabel>
            <skos:inScheme rdf:resource="{concat($codelistNs, 'CL000086')}"/>
        </loca:UNCountry>
</xsl:template>
```

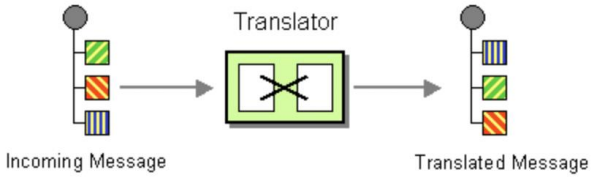
3.3 Metódy transformácie podľa podnikových integračných vzorcov („Enterprise Integration Patterns“)[28]

3.3.1 Prekladač správ na zmenu formátov

Informačné systémy, ktoré musia byť integrované systémom zasielania správ („messaging system“), sa zriedkavo dohodnú na spoločnom formáte údajov. Napríklad účtovný systém bude mať inú predstavu o fyzickej osobe ako daňový či dôchodkový systém. Okrem toho môže jeden systém uchovávať údaje v relačnom modeli, zatiaľ čo druhý systém používa dokumenty XML. Štandard RDF má tiež definované serializácie do rôznych formátov ako RDF/XML, Turtle a JSON LD. Integrácia existujúcich informačných systémov častokrát znamená, že nemáme možnosť upraviť ich tak, aby

[28] Zdroj: https://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageTransformationIntro.html, Dátum referencie: 24.05.2023

ľahšie spolupracovali s inými systémami. Integračné riešenie sa skôr musí prispôsobiť a vyriešiť rozdiely medzi rôznymi informačnými systémami. Vzor „Prekladač správ“ („Message Translator“)[29] ponúka všeobecné riešenie takýchto rozdielov vo formátoch údajov. Pomocou neho sa dá zabezpečiť aj to, že integračné riešenie je schopné komunikovať s externými stranami pomocou štandardizovaných dátových formátov, zatiaľ čo interné systémy sú založené na proprietárnych formátoch.



Obrázok 5: Schéma prekladača správ[30]

Prekladač správ funguje ako adaptér (Obrázok 5), ktorý konvertuje rozhranie komponentu na iné rozhranie, aby ho bolo možné použiť v inom kontexte. Konkrétna biznis logika implementovaná v prekladači správ musí byť prispôsobená zdrojovému a cieľovému formátu.

3.3.2 Zabalenie a rozbalenie údajov „v obálke“

Väčšina systémov integrovaná cez výmenu správ alebo súborov rozdeľuje údaje správy alebo súboru na záhlavie a telo. Záhlavie obsahuje polia, ktoré integračná infraštruktúra používa na riadenie toku správ. Väčšina cieľových systémov, ktoré sa zúčastňujú na integračnom riešení, však vo všeobecnosti o týchto dodatočných dátových prvkoch nevie. V niektorých prípadoch môžu systémy dokonca považovať tieto polia za chybné, pretože nezodpovedajú formátu správy alebo súboru, ktorý používa aplikácia. Na druhej strane, komponenty, ktoré smerujú správy alebo súbory medzi poskytovateľmi a konzumentmi, môžu vyžadovať polia záhlavia a považovali by správu alebo súbor za neplatný, ak by neobsahovali správne polia záhlavia.

Ako sa môžu existujúce systémy zúčastňovať na výmene správ, ktorá kladie špecifické požiadavky na formát správy, napríklad na polia záhlavia správy alebo šifrovanie? Treba použiť vzorec „Envelope Wrapper“ na zabalenie údajov aplikácie do obálky, ktorá je v súlade s integračnou infraštruktúrou. Po doručení správy do cieľa sa údaje z tejto obálky rozbalia.

[29] Zdroj: https://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageTransformationIntro.html, Dátum referencie: 25.05.2023

[30] Zdroj: https://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageTranslator.html, Dátum referencie: 25.05.2023

3.4 Metóda na zlúčenie údajov z viacerých zdrojov a/alebo zmenu schémy cez mapovanie

Cieľom je zlučovanie údajov z rôznych zdrojov do jednotného a konzistentného pohľadu. Na dosiahnutie tohto cieľa je potrebné mapovať a transformovať dátové schémy a formáty zdrojového systému na spoločnú dátovú schému cieľa.

V praxi ide o metódu mapovania, pomocou ktorej možno vstupnú schému mapovať na výstupnú – inými slovami povedať komponentu mapovania, ktorý dátový prvok zo zdrojovej schémy sa má premietnuť do akého dátového prvku cieľovej schémy. V rámci tohto procesu mapovania možno aj vymazať niektoré dátové prvky, ktoré sa na výstupe nemajú objaviť vôbec, zmeniť maximálnu možnú dĺžku dátového prvky, jeho typ – napríklad z reťazca na číslo, spojiť viaceré dátové prvky do jedného (napríklad meno s priezviskom) a podobne. Túto metódu mapovania možno aplikovať aj na viaceré zdrojové systémy, z ktorých údaje alebo súbory sa majú ocitnúť v cieľovom systéme s definovanou schémou.

Metóda filtrovania údajov, ktorá sa spomína v úvode kapitoly 3.1 nemusí nevyhnutne odstraňovať len dátové prvky. Je užitočná aj na zjednodušenie štruktúry správy alebo súboru, čo je obzvlášť potrebné pri zlučovaní viacerých zdrojových súborov do jedného cieľového, určeného na analýzu údajov. Často sa stáva, že súbory sú reprezentované ako stromové štruktúry. Mnohé súbory pochádzajúce z externých systémov alebo softvérových balíkov obsahujú mnoho úrovní vnorených, opakujúcich sa skupín údajov, pretože sú modelované podľa všeobecných, normalizovaných databázových štruktúr. Často sa stáva, že známe obmedzenia a predpoklady spôsobujú, že táto úroveň vnorenia je zbytočná a metóda filtrovania sa môže použiť na „sploštenie“ hierarchie do jednoduchého zoznamu prvkov, ktorý je ľahšie pochopiteľný a spracovateľný inými systémami.

3.5 Metódy aplikované len pred analýzou údajov

Existuje niekoľko metód dátovej transformácie údajov, ktoré môžu pomôcť štruktúrovať a vyčistiť údaje pred analýzou alebo uložením do dátového skladu alebo dátového lakehouse, pričom najbežnejšie sú:

- **Vyhladzovanie**: Ide o proces dátovej transformácie, pri ktorom sa zo súboru údajov odstraňujú skreslené alebo bezvýznamné údaje. Menšie úpravy údajov sa realizujú aj na základe identifikovaných špecifických vzorov alebo trendov v údajoch.
- **Agregácia**: Agregácia údajov zhromažďuje nespracované údaje z viacerých zdrojov a ukladá ich do jedného formátu, dátového modelu a štandardu na presnú analýzu. Táto technika je potrebná, keď sa jedná o datasety s veľkými objemami údajov.
- **Diskretizácia**: Táto technika dátovej transformácie vytvára v spojitých údajoch intervalové značky na zvýšenie efektívnosti a jednoduchšiu analýzu. Tento proces využíva algoritmy rozhodovacích stromov na transformáciu veľkého súboru údajov na kompaktné kategorické údaje.
- **Zovšeobecnenie**: Generalizácia využíva hierarchie pojmov a konvertuje nízkoúrovňové atribúty na vysokoúrovňové, čím vytvára prehľadný obraz údajov.
- **Indexovanie a radenie:** Údaje možno transformovať tak, aby boli logicky usporiadané alebo aby vyhovovali schéme ukladania údajov. Napríklad v relačných

Tento projekt je podporený z Európskeho sociálneho fondu.

systémoch riadenia databáz môže vytvorenie indexov zlepšiť výkon alebo zlepšiť správu vzťahov medzi rôznymi tabuľkami.

- **Anonymizácia a šifrovanie**: Údaje obsahujúce informácie umožňujúce identifikáciu osôb alebo iné informácie, ktoré by mohli ohroziť súkromie alebo bezpečnosť, by sa mali pred šírením anonymizovať. Šifrovanie súkromných údajov je požiadavkou v mnohých odvetviach a systémy môžu vykonávať šifrovanie na viacerých úrovniach, od jednotlivých databázových buniek až po celé záznamy alebo polia.
- **Konštrukcia atribútov**: Táto technika umožňuje organizovať datasety vytvorením nových atribútov z existujúceho datasetu.
- **Normalizácia**: Normalizácia transformuje údaje tak, aby atribúty zostali v určenom rozsahu na efektívnejšie aplikovanie algoritmov dolovania údajov.
- **Manipulácia**: Manipulácia je proces zmeny alebo úpravy údajov, aby boli lepšie čitateľné a organizované. Nástroje na manipuláciu s údajmi pomáhajú identifikovať vzory v údajoch a transformovať ich do použiteľnej podoby na získanie poznatkov.
- **Kompresia údajov** do spomínaného stĺpcového formátu Apache Parquet, veľmi zásadne šetriaceho diskový priestor, s ktorým sa dá veľmi pohodlne pracovať napríklad cez Apache Spark[31]. Transformácia napríklad z formátu CSV do Parquet sa dá jednoducho spraviť napríklad cez spomínaný Apache Spark, a to v rôznych programovacích jazykoch ako Python, SQL, Scala, Java alebo R.

Pokiaľ ide o analýzu údajov, transformácia sa zvyčajne uskutočňuje po extrakcii alebo načítaní údajov (ETL/ELT). Takže v tomto prípade záleží na tom, ako bude vyzerať architektúra a technologický dátový „stack“ pre Konsolidovanú analytickú vrstvu (viac v dokumente 4.2.1 Koncept pre zavádzanie analytického spracovania údajov do praxe).

Existujú dve skupiny nástrojov, v ktorých môže prebehnúť transformácia, a to buď v rámci Centrálnej integračnej platformy, v rámci ktorej je nasadený nástroj Talend (kapitola 4.2.1.1), alebo v rámci dátového „stacku“ Konsolidovanej analytickej vrstvy (napríklad ako je uvedené v kapitole 5.2), kde môžu byť údaje uložené v takzvanej „staging“ databáze, z ktorej sa až po úspešnej transformácii prevedú do cieľovej databázy alebo databáz už na vykonávanie konkrétnych analytických úloh. Uloženie do „staging“ databázy zabezpečí rýchle zvrátenie procesu v prípade, že niečo nepôjde podľa plánu. Počas tejto fázy existuje možnosť vytvárať audítorské správy na účely dodržiavania právnych predpisov alebo diagnostikovať a opravovať prípadné problémy s údajmi.

Tu je ešte dôležité poznamenať, že pred extrakciou údajov do „staging“ databázy alebo do cieľovej databázy na analýzy, ak sa „staging“ databáza nepoužíva, **je nevyhnutné vykonať proces pseudonymizácie a anonymizácie tak, ako je to popísané v dokumente 1.1.5 Štandardizácia anonymizácie údajov.**

[31] Zdroj: https://spark.apache.org, Dátum referencie: 23.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

4 Návrh nástrojov dátovej transformácie

Existujú dva bežné prístupy k dátovej transformácii:

1. Vytvorenie vlastných nástrojov na dátovú transformáciu s využitím skriptovania alebo zdrojového kódu. Tieto nástroje sú de facto štandardom s najväčšou mierou prispôsobenia, flexibility a kontroly nad spôsobom dátovej transformácie.
2. Využitie existujúcich nástrojov, spravidla pre ETL alebo ELT, pričom na prispôsobenie dátovej transformácie daným potrebám sa využíva grafické rozhranie alebo skript – ide teda o takzvané nástroje so žiadnym alebo minimálnym kódom („Low-/No-Code“ nástroje). Tieto riešenia prešli dlhú cestu, najmä v posledných rokoch, a mnohé z nich sú už veľmi prepracované s veľkou škálou rôznych funkcionalít.

V nasledujúcich kapitolách si rozoberieme obe možnosti s ohľadom na ich využiteľnosť v prostredí slovenského eGovernmentu.

4.1 Vlastné nástroje

Mnohé spoločnosti používajú všeobecné programovacie jazyky na písanie vlastných nástrojov ETL. Nemusí sa jednať len o transformáciu v rámci vlastných ETL alebo ELT nástrojov, ale aj o implementáciu transformácie v rámci spomínaných enterprise integračných vzorcov („patternov“) alebo transformačnej mikroslužby, ktorú možno napojiť na zdrojový systém alebo na Centrálnu integračnú platformu. Tento prístup má najväčšiu flexibilitu, ale vyžaduje si aj najväčšie úsilie. Tento prístup tiež vyžaduje, aby používatelia vykonávali vlastnú údržbu, vytvárali vlastnú dokumentáciu, testovali a vykonávali priebežný vývoj. Používatelia vlastných nástrojov dátovej transformácie často ťažko hľadajú pomoc u ľudí mimo vlastného tímu. Ide o tieto programovacie jazyky:

- **SQL**: Ak sú zdroj a cieľ údajov rovnaké, potom je použitie jazyka SQL veľmi efektívnou možnosťou. Jazyk SQL je veľmi efektívny pri čítaní a zápise údajov, ako aj pri vykonávaní základných transformácií. Jazyk SQL nie je veľmi efektívny pri zložitých transformáciách, ktoré zahŕňajú napríklad rozhodovacie stromy alebo volanie externých zdrojov. Jazyk SQL je zabudovaný v databáze, takže nie sú potrebné žiadne ďalšie licenčné poplatky ani technológie. Jazyk SQL je všeobecne zrozumiteľný pre databázových administrátorov a vývojárov.
- **Python** je všeobecný programovací jazyk. Stal sa populárnym nástrojom na vykonávanie úloh ETL vďaka jednoduchému používaniu a rozsiahlym knižniciam na prístup k databázam a technológiám ukladania dát. Python sa môže používať namiesto nástrojov ETL na jednotlivé úlohy ETL. Mnoho dátových inžinierov používa jazyk Python namiesto nástroja ETL, pretože je na tieto úlohy flexibilnejší a výkonnejší.
- **Java** je ďalším univerzálnym programovacím jazykom, ktorý možno použiť na vytvorenie procedúr ETL alebo na implementáciu mikroslužieb či iných integračných vzorcov. Java je jedným z najpopulárnejších programovacích jazykov a má širokú podporu pre rôzne zdroje údajov a transformáciu údajov. Java a Python majú rôzne výhody a nevýhody a treba zvážiť pri nich iné kompromisy pri tvorbe ETL. Pri výbere medzi nimi môžu byť najdôležitejším faktorom existujúce zručnosti v tímoch.

Tento projekt je podporený z Európskeho sociálneho fondu.

- **Spark** a **Hadoop** pracujú s veľkými súbormi údajov na klastroch počítačov. Uľahčujú použitie výkonu mnohých počítačov spolupracujúcich na vykonanie danej úlohy s údajmi. Táto schopnosť je dôležitá najmä vtedy, keď sú údaje príliš veľké na to, aby sa dali uložiť na jeden počítač. V súčasnosti sa Spark a Hadoop nepoužívajú tak ľahko ako Python a existuje oveľa viac ľudí, ktorí poznajú a používajú Python. Spark má však aj svoju verziu v Pythone, ktorá sa nazýva PySpark.

Najčastejšie sa dátové transformácie vykonávajú pomocou jazyka SQL alebo jazyka Python. V najjednoduchšom prípade sa tieto transformácie môžu ukladať do nejakého repozitára a vykonávať pomocou nejakého orchestrátora. Častejšie sa na orchestráciu a volanie transformácií používajú platformy ako dbt[32], ktoré používajú kombináciu jazykov SQL a Python. Tieto nástroje alebo systémy sa často obmedzujú na programové vytváranie tabuliek alebo transformácií pomocou niektorého skriptovacieho jazyka.

4.1.1 Naprogramovanie modulu dátovej transformácie „na mieru“ s využitím knižníc

Centrálny model údajov je definovaný podľa štandardov W3C pre sémantický web (viac v dokumente 1.12 Štandardizácia pre modelovanie údajov). Žiaľ tieto štandardy nie sú bežne podporované v najviac rozšírených nástrojoch pre dátovú transformáciu, ktoré sú popísané v kapitolách. V programovacom jazyku Java je dostupná overená a populárna knižnica RDF4J[33], pomocou ktorej sa dá pracovať s údajmi v štandarde RDF. Okrem iného sa dá využiť aj na zmenu údajov v štandarde RDF na akýkoľvek z podporovaných formátov (teda aj na RDF/XML, JSON LD a Turtle) a validáciu výstupného datasetu cez SHACL.

V rámci štandardizácií W3C sa určujú odporúčania, ako transformovať údaje z formátu XML (prípadne CSV, JSON a pod.) do RDF. Pre XML súbory sa zvolilo odporúčanie využiť štandard XSLT tak, ako je to popísané v metóde v kapitole 3.2. Túto metódu sa dá jednoducho implementovať v programovacom jazyku Java s využitím „Java API for XML Processing (JAXP)“[34] alebo platený produkt Saxon procesor pre XSLT 3.0, XQuery a XML Schema[35], či Saxon open-soruce verziu[36]. Pre zachovanie dôveryhodnosti transformovaných údajov odporúčame:

1. Vhodným spôsobom využiť práve spomínané odporúčania W3C v kombinácii s centrálnym dátovým katalógom a CMÚ.
2. Transformačný modul umiestniť:
   a) Buď priamo na zdrojový informačný systém pod správou OVM – preferovaná, no technicky a časovo náročná alternatíva – dôveryhodnosť údajov vtedy garantuje samotný zdrojový systém rovnako, ako v prípade pôvodných údajov.
   b) Alebo ako modul (mikroslužbu) na IS CSRÚ / CIP, v ktorom sa dajú aj joby rozširovať cez zdrojový kód v Jave – technicky aj časovo ľahšie realizovateľná

[32] Zdroj: https://www.getdbt.com/product/what-is-dbt/ , Dátum referencie: 23.05.2023

[33] Zdroj: https://rdf4j.org/about/, Dátum referencie: 29.05.2023

[34] Zdroj: https://www.oracle.com/java/technologies/jaxp-introduction.html, Dátum referencie: 29.05.2023

[35] Zdroj: https://www.saxonica.com/welcome/welcome.xml, Dátum referencie: 30.05.2023

[36] Zdroj: https://www.saxonica.com/support/opensource.xml, Dátum referencie: 30.05.2023

alternatíva – dôveryhodnosť údajov garantuje zdrojový systém pôvodných údajov v kombinácii s IS CSRÚ / CIP, kde garantom je prevádzkovateľ systému – MIRRI.

4.2 „Low-/No-Code“ nástroje (spravidla ETL nástroje)

Tieto nástroje na dátovú transformáciu sa najľahšie používajú netechnickými používateľmi. Dátová transformácia je najčastejšie súčasťou ETL nástrojov. Umožňujú zhromažďovať údaje z akéhokoľvek zdroja a načítať ich do cieľového riešenia, napríklad dátového skladu pomocou, interaktívneho grafického rozhrania. Tieto nástroje existujú na trhu už viac ako 30 rokov. Za ten čas sa technológia vyvinula a na trh sa dostali rôzne typy riešení. Za posledné desaťročie sa rozšírilo množstvo riešení s minimom zdrojového kódu pre definovanie konkrétnej dátovej transformácie. Existuje niekoľko dodávateľov čisto ETL nástrojov, ako napríklad Informatica. Ďalšie nástroje ponúkajú veľkí dodávatelia softvéru, ako napríklad IBM, Oracle a Microsoft. Nedávno sa objavili nástroje ETL s otvoreným zdrojovým kódom a cloudové služby ETL.

4.2.1 Komerčné ETL nástroje

Komerčné softvérové produkty pre ETL sú na trhu najdlhšie a spravidla sú najrozvinutejšie, pokiaľ ide o prijatie a dostupnú funkcionalitu. Všetky tieto produkty poskytujú grafické rozhrania na navrhovanie a vykonávanie ETL „pipelines“. Všetky sa pripájajú k väčšine relačných databáz. Niektoré z nich podporujú aj iné ako relačné zdroje údajov, ako napríklad JSON a XML. Len niekoľko z nich podporuje zdroje streamovaných udalostí, ako napríklad z nástroja Apache Kafka. V nasledujúcom zozname uvádzame najznámejšie produkty z tejto kategórie:

- **Informatica PowerCenter**: Tento produkt je pravdepodobne najvyspelejším produktom pre ETL na trhu. Používa ho mnoho veľkých spoločností a je dobre hodnotený analytikmi. Je súčasťou rozsiahleho portfólia produktov, ktoré sa spájajú pod názvom Informatica Platform a sú tiež veľmi drahé. Informatica ako ETL pre pološtruktúrované a neštruktúrované zdroje je však pozadu oproti niektorým iným produktom.
- **IBM InfoSphere DataStage**: Ide o veľmi vyspelý produkt pre ETL, ktorý bol odkúpený od spoločnosti Ascential. Je obľúbený najmä v obchodoch IBM. Na rozdiel od mnohých iných nástrojov ETL poskytuje silné možnosti práce s mainframe počítačmi. DataStage je vnímaný ako drahý, zložitý na licencovanie a prekrýva sa s inými produktmi z rovnakej rodiny.
- **Oracle Data Integrator (ODI)**: Ide o produkt pre ETL spoločnosti Oracle, ktorý je k dispozícii už mnoho rokov. Používa zásadne odlišnú architektúru od ostatných produktov na ETL. Namiesto vykonávania transformácií v samotnom nástroji ETL s použitím vyhradeného procesu a hardvérových zdrojov ODI presúva údaje do cieľového miesta a potom vykonáva transformácie s využitím funkcií databázy alebo klastra Hadoop. Takže ide vlastne o nástrol ELT.
- **Microsoft SQL Server Integration Services (SSIS)**: SSIS je medzi používateľmi servera SQL Server veľmi populárny. Je lacnejší ako iné podnikové nástroje na ETL a jednoduchšie sa používa.
- **Ab Initio** je veľmi utajovaná spoločnosť so sídlom v Lexingtone, MA. Produkt začal ako ETL a vyvinul sa tak, aby poskytoval ďalšie funkcionality, ako napríklad katalógy

Tento projekt je podporený z Európskeho sociálneho fondu.

metadát. Ab Initio je proprietárne a veľmi drahé riešenie. Spoločnosť však tvrdí, že je výkonnejšie a jednoduchšie na používanie ako tradičné nástroje na ETL.

- **SAP Data Services**: Nástroj na ETL od spoločnosti SAP je určený predovšetkým na presun údajov medzi aplikáciami SAP. Mimo týchto prostredí nie je široko používaný.
- **SAS Data Manager**: Spoločnosť SAS vyvinula produkt ETL so silnou podporou pre Hadoop, toky údajov a strojové učenie, ale s obmedzenou podporou pre hromadné nahratie údajov do cieľových systémov. Data Manager je vnímaný ako drahý, zatiaľ čo SAS je vnímaný ako dodávateľ s vysokou spokojnosťou zákazníkov.

4.2.1.1 Talend

V Talende sa dajú dátové procedúry vrátane dátovej transformácie písať v programovacom jazyku Java. Talend sa dodáva v dvoch verziách:

3. Talend Open Studio (TOS), ktorý predstavuje súbor open source nástrojov so špecifickými funkcionalitami, ktoré pomáhajú extrahovať, transformovať, pripravovať a načítavať dáta a mnoho ďalšieho,
4. Talend Enterprise, čo je jednotná unifikovaná platforma založená na predplatnom, s viacerými možnosťami a špecifickými vylepšeniami – táto platforma je aktuálne nasadená ako Centrálna integračná platforma vo verzii 7.3 – Talend Data Services Platform.

Za zmienku stojí aj to, že Talend sa ponúka aj vo verejných cloudoch, napríklad na Amazone[37] alebo v Microsoft Azure[38].

V rámci open source verzie Talend Open Studio možno transformáciu robiť v nasledujúcich nástrojoch, ktoré sú ale od seba striktne oddelené a nie sú nijak medzi sebou integrované:

- Talend Open Studio pre dátovú integráciu: Nástroj ETL/ELT s grafickým používateľským rozhraním na vývoj „pipelines“ na integráciu údajov. Má konektory na väčšinu známych databáz, systémov a technológií. Umožňuje správu súborov, ako aj vykonávanie a orchestráciu dátových tokov, v ktorých možno vykonávať transformáciu, agregáciu, obohacovanie údajov a mnoho ďalšieho.
- Talend Open Studio for Big Data: Okrem všetkých vyššie uvedených možností ETL tento nástroj obsahuje špecifické komponenty, ktoré uľahčia interakciu s nástrojmi a ekosystémami pre veľké údaje, ako napríklad interakciu údajov s „data lakes“, konektory na cloudové služby, komponenty Hadoop (HDFS, Hbase, Hive, a ďalšie). A to všetko nad rámec možností nástroja na integráciu údajov.
- Talend Open Studio pre Enterprise Service Bus: Obsahuje všetky možnosti nástroja pre dátovú integráciu, ale pridáva aj niektoré komponenty REST Server, ktoré umožňujú komunikovať s REST API, WSDL, OAuth a ďalšími. Nástroj pracuje aj s protokolmi HTTP, Apache Kafka a mnohými ďalšími protokolmi a okrem vizuálneho

[37] Zdroj: https://aws.amazon.com/marketplace/pp/prodview-5ozbitulzm3ca#, Dátum referencie: 16.06.2023

[38] Zdroj: https://azuremarketplace.microsoft.com/en-us/marketplace/apps/talend.talendclouddi?tab=Overview, Dátum referencie: 16.06.2023

rozhrania typu „drag-and-drop“ obsahuje aj nástroje príkazového riadka a skriptovania.

Okrem nástrojov Open Source poskytuje Talend aj najrozsiahlejšiu verziu Enterprise s názvom Talend Data Fabric. Ide o jednotnú platformu, ktorá v podstate spája všetky komponenty nástrojov Talend Open Studio, a k tomu navyše:

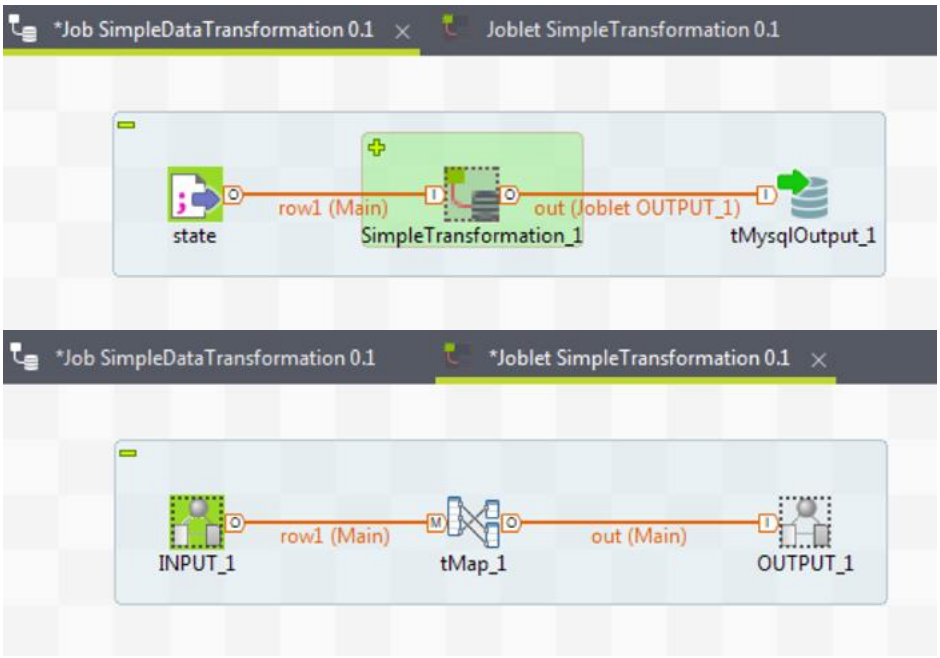
- Jednoduché pripájanie dátových zdrojov s viac ako 900 rôznymi konektormi a komponentmi,
- Spravuje údaje vo všetkých druhoch prostredí, v cloude aj v lokálnom prostredí,
- Podporuje dávkové načítanie, načítanie v reálnom čase, streaming a prípady použitia veľkých dát.
- Obsahuje už zabudované funkcie strojového učenia a kvality údajov.

Aktuálne využívaná verzia 7.3 – Talend Data Services v rámci CIP síce nepodporuje strojové učenie, napojenie na rôzne druhy NoSQL databáz a dátových cloudov, či rozšírenie možnosti ochrany súkromia údajov (podrobný rozpis podporovaných komponentov sa nachádza na linke tu), ale podporuje kľúčové komponenty pre dátovú transformáciu.

Talend Data Fabric a Talend Data Services ponúka nový komponent s názvom Joblet, ktorý pomáha pri pri opätovnom používaní kódu bez jeho duplikovania. Joblet je spôsob zapuzdrenia opakujúcich sa krokov spracovania údajov alebo zložitých transformácií s cieľom opätovne použiť ten istý kód na viacerých miestach alebo urobiť zložitú úlohu prehľadnejšou. Joblet sa dá použiť v rôznych úlohách a/alebo vyžiť niekoľkokrát v tej istej úlohe, pričom kód je napísaný len raz a zdieľaný vo všetkých týchto úlohách, čo je oveľa jednoduchšie na údržbu a úpravu. Keď je úloha spustená, kód Joblet je integrovaný do hlavného kódu úlohy. Nejde o novú triedu v Jave, ktorá by sa spúšťala samostatne, je to rovnaká trieda ako v hlavnej úlohe, zachováva kontext vykonávania a premenné namiesto toho, aby sa vytvorilo nové samostatné vykonávanie, čo by sa stalo pri vytvorení pod-úloh. Nasledujúci Obrázok 6 znázorňuje v grafickom rozhraní Talendu, ako možno využiť Joblet-y na transformáciu. V rámci jednoduchej transformácie je definované premapovanie metadát v dátových prvkoch medzi vstupom a výstupom cez nástroj „tMap“, ktorý je možno doplniť o vlastný Java kód pre definovanie dodatočnej biznis logiky transformácie.

V Enterprise verzii sa dajú vytvárať aj testovacie prípady, ktoré pomôžu pri ladení kódu transformácie a rýchlejšom vývoji. Nemusia sa vykonávať len celé úlohy, pričom netreba manuálne izolovať časti kódu a poskytnúť im vstupy/výstupy, aby sa dali testovať jednotlivo. Z používateľského grafického rozhrania sa dajú jednoducho vybrať komponenty, ktoré treba testovať. Tieto komponenty budú jediné, ktoré sa budú vykonávať v testovacom prípade. Následne sa dajú zadať konkrétne vstupy a výstupy pre tieto komponenty, ktoré sa majú vykonať a otestovať. To umožňuje efektívne izolovať chyby, napríklad aj v jednotlivých Joblet-och. Vďaka tomu netreba ručne upravovať kód, aby sa otestovali len niektoré časti úloh, ako bolo spomenuté, stačí vybrať časti, ktoré treba otestovať, a poskytnúť vstup a/alebo výstup, ktorý potrebujú, čo pomáha ušetriť veľa času počas vývoja.

Tento projekt je podporený z Európskeho sociálneho fondu.



Obrázok 6: Ukážka využitia Joblet-ov na transformáciu

Základným komponentom dátovej transformácie XML súborov je komponent tXSLT[39], ktorý využíva štandard a jazyk XSLT, popísaný v kapitole 3.2. Avšak ak je dokument .xslt napísaný v jazyku XSLT 3.0, nemusí tento komponent Talendu dokázať interpretovať všetky príkazy. Tento komponent tXSLT možno napojiť aj na Talend Data Mapper (ide o komponent tMap popísaný nižšie) a využiť pred transformáciou mapovanie údajov medzi zdrojom a cieľom[40].

Metódu mapovania údajov aj s transformáciou, ako je popísaná v kapitole 3.4, podporuje Talend cez komponent tMap[41]. Komponent tHMap[42] vykonáva transformácie (nazývané mapy) medzi rôznymi zdrojmi a cieľmi využitím možností aplikácie Talend Data Mapper. Komponent tHMap transformuje údaje zo širokej škály zdrojov do širokej škály cieľov vrátane podpory viacerých vstupov a viacerých výstupov. Ak sa musí použiť viacero vstupov, treba použiť vstupno-výstupné funkcie aplikácie Talend Data Mapper. Komponent tHConvertFile[43] konvertuje údaje z jednej reprezentácie do druhej v prostredí Spark.

[39] Zdroj: https://help.talend.com/r/en-US/7.3/xml/txslt, Dátum referencie: 23.05.2023

[40] Zdroj: https://help.talend.com/r/en-US/7.3/data-mapping/using-external-xsl-transformation-in-talend-data-mapper, Dátum referencie: 23.05.2023

[41] Zdroj: https://help.talend.com/r/en-US/7.3/data-integration-job-examples/tmap-job-example, Dátum referencie: 26.05.2023

[42] Zdroj: https://help.talend.com/r/en-US/7.3/data-mapping/thmap, Dátum referencie: 26.05.2023

[43] Zdroj: https://help.talend.com/r/en-US/7.3/data-mapping/thconvertfile, Dátum referencie: 26.05.2023

V dokumentácii Talend[44] je popísaný ilustračný scenár s popisom trojkrokovej úlohy, ktorá generuje náhodné údaje zo vstupnej zložky, transformuje tieto údaje pomocou mapy, ktorá bola predtým vytvorená v perspektíve Mapovania, a potom transformované údaje vypíše do súboru JSON. Pri tejto úlohe sa pracuje s metadátami Talend Data Integration pre vstup a metadátami Talend Data Mapper pre výstup.

4.2.2 Open-source ETL nástroje

Za posledných 10 rokov vývojári softvéru vytvorili niekoľko produktov pre ETL s otvoreným zdrojovým kódom. Tieto produkty sa dajú používať bezplatne. Ich zdrojový kód je tiež voľne dostupný, čo umožňuje rozšíriť alebo vylepšiť ich možnosti. Tieto nástroje sa výrazne líšia kvalitou, integráciami, jednoduchosťou používania, prijatím a dostupnosťou podpory. Podobne ako komerčné nástroje ETL, aj mnohé z týchto open source nástrojov ETL poskytujú grafické rozhranie na navrhovanie a spúšťanie „pipelines“. K týmto nástrojom patria:

- **Talend Open Studio**: **Tento nástroj na ETL od spoločnosti Talend je najpopulárnejším open source produktom ETL**. Open Studio generuje kód v jazyku Java pre potrubia ETL namiesto toho, aby spúšťal konfigurácie potrubia prostredníctvom motora ETL. Tento prístup mu poskytuje určité výkonnostné výhody. Jeho Enterprise verzia však ponúka oveľa viac ako len ETL nástroj – rieši okrem dátovej integrácie aj aplikačnú integráciu. Keďže je nasadená aj v rámci Centrálnej integračnej platformy, venujeme sa tomuto nástroju aj v samostatnej kapitole 4.2.1.1.
- **Pentaho Data Integration (PDI)**: PDI, predtým známy ako Kettle, je open source nástroj na ETL, ktorý je známy vďaka svojmu grafickému rozhraniu s názvom Spoon. PDI vytvára súbory XML na reprezentáciu „pipelines“, ktoré vykonáva prostredníctvom svojho „enginu“ pre ETL. Spoločnosť Pentaho bola v roku 2015 odkúpená spoločnosťou Hitachi Data Systems.
- **Hadoop**. Hadoop je platforma pre distribuované výpočty na rôznorodé účely. Používa sa na ukladanie, manipuláciu a analýzu údajov akejkoľvek štruktúry. Hadoop je komplexný ekosystém open source projektov, ktorý zahŕňa viac ako 20 rôznych technológií. Niektoré z týchto projektov sa používajú na vykonávanie úloh v rámci ETL, napríklad Pig, MapReduce a Apache Spark. Pri použití na ETL sa údaje zvyčajne najprv načítajú do distribuovaného súborového systému Hadoop (HDFS) v takej forme, ako sa nachádzajú v zdrojových systémoch. Sqoop je nástroj používaný na presun údajov z relačných databáz do HDFS. Po uložení údajov do systému Hadoop možno na **transformáciu** a uloženie vyčistených údajov do HDFS použiť ktorýkoľvek z projektov. Hive je populárny projekt na používanie jazyka SQL na definovanie týchto transformácií (dotaz Hive je skompilovaný do MapReduce). Použiť možno aj Pig a Spark. Ďalšie projekty sa používajú na koordináciu viacstupňových transformácií pre ETL. Ekosystém Hadoop sa veľmi rýchlo vyvíja a často sa objavujú nové projekty, ktoré sú určené na vykonávanie transformácie údajov.

[44] Zdroj: https://help.talend.com/r/en-US/7.3/data-mapping/tmap-trowgenerator-tfileoutputraw-tfileoutputraw-transforming-from-data-integration-schema-to-complex-content-schema-standard-component-enterprise-the, Dátum referencie: 26.05.2023

4.2.2.1 Linked Pipes pre podporu znalostných grafov a CMÚ

Tento nástroj pre transformáciu datasetov v rôznych formátoch cez procedúry ETL sa zameriava na podporu webových štandardov W3C. Používaním známych knižníc s otvoreným zdrojovým kódom ponúka najmodernejšiu podporu webových štandardov, ako sú RDF 1.1 a SPARQL 1.1. Iba tam, kde neexistuje opakovane použiteľná podporovaná implementácia, vytvára vlastnú, napríklad pre štandard CSV na zdieľanie na webe.

Implementuje aj výhody prelinkovaných údajov, ako získanie údajov vždy, keď sa dereferencuje URI, alebo pridanú hodnotu pri modelovaní údajov pomocou známych slovníkov. Z tohto nástroja sa dajú nasadiť do vlastného prostredia len tie moduly, ktoré sú potrebné. Takže napríklad na server nie je potrebné nasadiť aj grafické používateľské rozhranie, ktoré je ďalšou výhodou tohto nástroja. Všetky funkcionality sú dostupné cez REST API, takže sa dá vytvoriť aj vlastné grafické používateľské rozhranie. Spracovanie údajov sa dá rozšíriť o mnohé predpripravené komponenty[45].

Najčastejšie sa tento nástroj používa na transformáciu súborov CSV na RDF, pomocou nasledujúcich krokov[46]:

- **Prevedenie tabuľky programu Excel na CSV**: Začneme sťahovaním tabuľky Excel pomocou komponentu HTTP GET systému LinkedPipes-ETL. Prevzatý súbor vložíme do komponentu „Excel to CSV[47]“.
- **Transformácia CSV do RDF**: Aby sme zúžili okruh problémov, ktoré sa v tomto kroku posudzujú, vykonáme len syntaktickú konverziu na dátový model RDF. Ďalšie následné spracovanie údajov oddeľujeme do nasledujúcich krokov. LinkedPipes - ETL poskytuje komponent „Tabular[48]“ na transformáciu CSV na RDF. Tento komponent implementuje štandardné mapovanie špecifikované v odporúčaní W3C na generovanie RDF z tabuľkových údajov na webe[49]. Existuje niekoľko možností, ktoré nám umožňujú prispôsobiť výstup tohto komponentu.
- **Čistenie údajov**: Po predchádzajúcom kroku máme syntakticky platné údaje RDF. Majú však niektoré nedostatky, ktoré môžeme odstrániť. Aby sme údaje vyčistili, vložíme ich do komponentu „SPARQL Update[50]“, ktorý nám umožňuje vykonávať operácie SPARQL 1.1 Update[51] na vstupných údajoch RDF. Šikovným pomocníkom pri vývoji operácií SPARQL Update je SPARQLer Update Validator[52], ktorý sa dá použiť na kontrolu syntaxe operácií na aktualizáciu pred ich zadaním do LinkedPipes

[45] Zdroj: https://etl.linkedpipes.com/components/, Dátum referencie: 30.05.2023

[46] Zdroj: https://etl.linkedpipes.com/tutorials/csv-to-rdf/index, Dátum referencie: 30.05.2023

[47] Zdroj: https://etl.linkedpipes.com/components/t-exceltocsv.html, Dátum referencie: 30.05.2023

[48] Zdroj: https://etl.linkedpipes.com/components/t-tabular.html, Dátum referencie: 30.05.2023

[49] Zdroj: https://www.w3.org/TR/csv2rdf/, Dátum referencie: 30.05.2023

[50] Zdroj: https://etl.linkedpipes.com/components/t-sparqlupdate.html, Dátum referencie: 30.05.2023

[51] Zdroj: https://www.w3.org/TR/sparql11-update/, Dátum referencie: 30.05.2023

[52] Zdroj: http://sparql.org/update-validator.html, Dátum referencie: 30.05.2023

-ETL. Čistenie a mapovanie údajov zvyčajne vyžaduje viacero krokov, ktoré sú implementované viacerými inštanciami komponentu SPARQL Update. Preto je užitočné pridať ku každej inštancii popis, aby sa dali rýchlo rozlíšiť.

- **Explicitné zadanie verzie**: Ak sa v datasete vyskytujú údaje, ktoré nahrádzajú historické verzie, v tomto kroku sa táto zmená verzií dá explicitne zadefinovať, pričom staré verzie údajov sa označia cez vlastnosť: `dcterms:replaces`.
- **Opísanie sémantiky**: V tomto okamihu máme k dispozícii pomerne čisté RDF údaje. Údaje sú však spravidla opísané v zmysle vlastných vlastností, ktoré boli zadefinované len lokálne v zdrojovom súbore Excel. Tieto vlastnosti neprezrádzajú veľa zo svojho významu, čo sťažuje univerzálne pochopenie údajov. Aby sme zlepšili použiteľnosť údajov, mapujeme preto tieto vlastné vlastnosti na termíny zo štandardných slovníkov RDF.
- **Generovanie IRIs**: Komponent „Tabular“ štandardne vytvára prázdne uzly („blank nodes“)[53]. Prázdne uzly nie sú stabilnými identifikátormi, takže sa nezachovávajú naprieč dotazmi alebo aktualizáciami, čo sťažuje manipuláciu s nimi pri transformáciách údajov. Keďže prázdne uzly sú zvyčajne „pohodlím pre poskytovateľa obsahu a záťažou pre konzumenta obsahu“[54], konvertujeme ich na IRI, aby sme uľahčili konzumáciu údajov. IRI vytvárame v rámci domény, ktorú máme pod kontrolou. Tým sa vyhneme kolíziám s inými subjektmi, ktoré vytvárajú IRI vo svojich menných priestoroch, a umožní nám to, aby sa IRI dali dereferencovať.
- **Prepojenie údajov**: V tomto kroku vieme ďalej obohacovať údaje a prepájať ich s externými údajmi pomocou operácii SPARQL CONSTRUCT. LinkedPipes-ETL ponúka komponent „SPARQL CONSTRUCT[55]“ na spúšťanie takýchto dotazov. Podobne ako pri operáciách SPARQL Update sa dajú overiť dopyty SPARQL v nástroji SPARQLer Query Validator[56] predtým, ako sa zadajú do systému LinkedPipes-ETL.
- **Optimalizácia**: Ak spustíme „ pipeline“, môžeme zistiť, že jej úplné vykonanie môže trvať približne aj stovky minút. Čas behu sa môže výrazne líšiť v závislosti od latencie siete alebo špecifikácie počítača, na ktorom sa „pipelineL spúšťa. Táto rýchlosť môže byť pre náš súbor údajov v poriadku, pretože sa mení len raz ročne. Rýchlejšie vykonávanie však prospieva aj vývojovému cyklu „pipeline“, pretože môžeme rýchlejšie iterovať, ak sa výsledky vytvárajú bez veľkého oneskorenia. Existuje niekoľko spôsobov, ako môžeme zlepšiť čas vykonávania „pipeline“. LinkedPipes - ETL vytvára výstup každého kroku spracovania. Aj keď mať k dispozícii medziprodukty je užitočné na ladenie, spomaľuje to vykonávanie „pipeline“, pretože nástroj musí serializovať údaje do súborov. Časti tejto réžie sa môžeme vyhnúť zlúčením operácií SPARQL Update. SPARQL Update umožňuje spojiť viacero operácií do jednej požiadavky, ktorá sa vykoná v jednom kroku. Viacero operácií možno spojiť dohromady, ak sú oddelené bodkočiarkami.

---

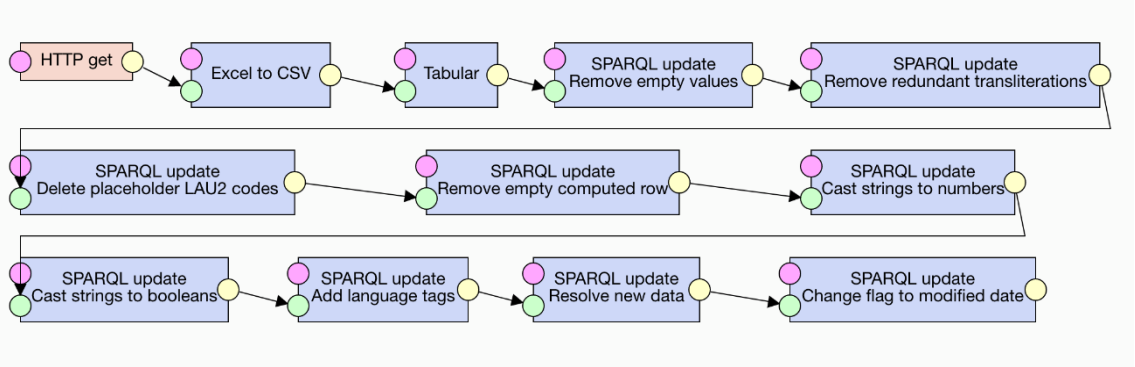
[53] Zdroj: https://www.w3.org/TR/rdf11-concepts/#section-blank-nodes, Dátum referencie: 30.05.2023

[54] Zdroj: https://lists.w3.org/Archives/Public/semantic-web/2011Mar/0068.html, Dátum referencie: 30.05.2023

[55] Zdroj: https://etl.linkedpipes.com/components/t-sparqlconstruct.html, Dátum referencie: 30.05.2023

[56] Zdroj: http://sparql.org/query-validator.html, Dátum referencie: 30.05.2023

- **Načítanie údajov**: LinkedPipes -ETL umožňuje ukladať výsledné údaje mnohými spôsobmi, aby ostatní mohli využívať výsledky vytvorenej „pipeline“. Súbory sa dajú ukladať lokálne alebo sa dajú odoslať na vzdialený server. Údaje sa môžu načítať do úložísk RDF.
- **Pridanie metadát**: V poslednom kroku vytvorené údaje anotujeme metadátami. LinkedPipes -ETL podporuje opis spracovaných datasetov pomocou metadát vyjadrených pomocou štandardného slovníka Data Catalog Vocabulary (DCAT)[57]. Konkrétne používa aplikačný profil DCAT pre dátové portály v Európe (DCAT-AP)[58]. Datasety možno opísať pomocou komponentu „DCAT-AP Dataset[59]“ a ich distribúcie možno pokryť komponentom „DCAT-AP Distribution[60]“. V tomto kontexte distribúcie zodpovedajú špecifickým formám daného datasetu, ako sú súbory na stiahnutie alebo API.



Obrázok 7: Časť "pipeline" pre transformáciu CSV na RDF v grafickom rozhraní LinkedPipes

4.2.3 Cloudové služby pre ETL

Amazon AWS, Google Cloud Platform a Microsoft Azure ponúkajú vlastné možnosti pre ETL ako cloudové služby. Ak sa údaje už nachádzajú na jednej z týchto cloudových platforiem, využívanie ich služieb ETL má niekoľko výhod. Jedným z dôležitých rozdielov je ich integrácia s vlastnými zdrojmi údajov - tradičné nástroje ETL nie je možné použiť na prácu s údajmi v týchto systémoch alebo v nich ponúkajú oveľa menej funkcionalít. Výhodou týchto cloudových ETL služieb je, že poskytujú úzku integráciu s inými cloudovými službami, pružnosť a ceny založené na skutočnom využívaní služby. Tieto riešenia sú však tiež vysoko proprietárne a fungujú len v rámci dodávateľa cloudu – nedajú sa použiť v platforme iného dodávateľa cloudu a nemôžete tieto možnosti presunúť do vlastných dátových centier. Ide o nasledovné cloudové služby:

[57] Zdroj: https://www.w3.org/TR/vocab-dcat/, Dátum referencie: 30.05.2023

[58] Zdroj: https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe, Dátum referencie: 30.05.2023

[59] Zdroj: https://etl.linkedpipes.com/components/e-dcatap11dataset.html,

[60] Zdroj: https://etl.linkedpipes.com/components/e-dcatap11distribution.html,

- **AWS Elastic MapReduce (EMR)**: Ide o produkt od spoločnosti Amazon AWS pre Hadoop. Ak sú údaje prevádzkované v cloude AWS a je potreba používať Hadoop na ETL alebo ELT, dá sa to zrealizovať v službe EMR. Rovnako ako pri každej distribúcii Hadoopu je k dispozícii niekoľko nástrojov na vykonávanie ETL vrátane Hive a Spark. Výhodou tohto riešenia je, že je veľmi výkonné a škálovateľné a dokáže pracovať so štruktúrovanými aj neštruktúrovanými údajmi. Je tiež pružné a používatelia platia len za to, čo používajú. Jeho nevýhodou je, že sa veľmi ťažko používa.
- **AWS Glue**: Ide o relatívne novú plne spravovanú službu pre ETL, ktorú spoločnosť Amazon AWS oznámila koncom roka 2016. Glue je zameraná na vývojárov. Je úzko integrovaná s inými službami AWS cloudu vrátane úložísk údajov, ako sú S3, RDS a Redshift, ako aj s inými službami, ako napríklad Lambda. Glue sa môže pripojiť k lokálnym zdrojom údajov a pomôcť tak používateľom presunúť ich údaje do cloudu. „Pipelines“ pre ETL sú napísané v jazyku Python a vykonávajú sa pomocou Apache Spark a PySpark. Podobne ako väčšina služieb na AWS, aj Glue je určená pre vývojárov, ktorí píšu zdrojový kód na využitie služby, a je vysoko proprietárna – „pipelines“ napísané v Glue budú fungovať len v AWS cloude.
- **AWS Data Pipeline** je cloudový nástroj pre ETL. Možno ho použiť na plánovanie bežných činností spracovania, ako je distribuované kopírovanie údajov, transformácie SQL, aplikácie MapReduce alebo dokonca vlastné skripty, a dokáže ich spúšťať nad viacerými cieľovými databázami, ako napríklad Amazon S3, RDS alebo DynamoDB.
- **Azure Data Factory** je plne spravovaná služba, ktorá sa pripája k širokému spektru cloudových a lokálnych zdrojov údajov. Je schopná kopírovať, transformovať a obohacovať údaje a potom ich zapisovať do dátových služieb Azure ako cieľových miest. Data Factory podporuje aj Hadoop, Spark a strojové učenie ako súčasť transformačných krokov.
- **Microsoft Fabric**[61] predstavuje plne spravovanú službu, ktorá okrem integrácie údajov z veľa zdrojov ponúka aj analytické nástroje, aj v reálnom čase, a strojové učenie. Ide o implementáciu architektúry založenej na dátovej sieti („data mesh“). Platforma Spark umožňuje dátovým inžinierom vykonávať rozsiahle transformácie údajov a demokratizovať údaje prostredníctvom systému „lakehouse“ (ide o kombináciu dátového skladu a dátového jazera). Integrácia Microsoft Fabric Spark s Data Factory umožňuje plánovanie a orchestráciu notebookov a úloh Spark.
- **Google Cloud Dataflow** je tiež plne spravovaná služba určená vývojárom na navrhovanie dávkových a kontinuálnych úloh ETL. Dataflow poskytuje vývojárom rozhrania API pre jazyky Java a Python na pripojenie k zdrojom Google Cloud, aplikovanie transformácií a zápis údajov do iných cieľov Google Cloud. Rozhrania API Dataflow sú založené na Apache Beam. Na rozdiel od iných cloudových služieb sa Dataflow nepripája k zdrojom na vlastnej infraštruktúre („on-premise“).
- **Segment**[62] je technológia poskytovaná ako SaaS, ktorá presúva údaje medzi systémami na základe udalostí. Spoločnosti používajú Segment na presun údajov do svojich dátových skladov a tiež na presun údajov z jednej aplikácie do druhej. Nie

[61] Zdroj: https://www.microsoft.com/en-us/microsoft-fabric, Dátum referencie: 26.05.2023

[62] Zdroj: https://segment.com/, Dátum referencie: 24.05.2023

je to striktne nástroj ETL. Segment sa pripája k mnohým zdrojom, vrátane desiatok populárnych aplikácií ako zdrojov a cieľov, vrátane dátových skladov.

- **Stitch**[63]: V rámci aplikácie Stitch je možné využívať aj iné nástroje, napríklad Stitch Data je poskytovateľ ETL typu SaaS, ktorý je na trhu nový a zameriava sa na vývojárov. Je postavený na open source jadre s názvom Singer.
- **Delta Live Tables (DLT)**[64] ako súčasť dátovej platformy Databricks, ktorú možno využívať vo verejných cloudoch od spoločností Amazon, Microsoft a Google: Pomocou Delta Live Tables sa dajú jednoducho definovať dátové „pipelines" v jazyku SQL alebo Python. Stačí zadať zdroj údajov, transformačnú logiku a cieľový stav údajov - namiesto ručného spájania jednotlivých oddelených úloh spracovania údajov. Automaticky sa dajú udržiavať všetky závislosti údajov v celej „pipeline" a opakovane používať tieto „pipelines" pre ETL so správou údajov nezávislou od prostredia. „Pipelines" sa dajú spúšťať v dávkovom alebo streamovom režime. Delta Live Tables ponúkajú aj možnosti validácie a kontroly integrity údajov a dajú sa preddefinovať politiky pre chyby v údajoch alebo ich nedostatočnú kvalitu (ako zlyhanie, vyradenie, upozornenie alebo karanténa údajov). Okrem toho sa dajú sledovať trendy kvality údajov v čase pre prehľad o tom, ako sa údaje vyvíjajú a kde môžu byť potrebné zmeny. Automatické škálovanie optimalizuje využitie klastra tým, že škáluje zaťaženie len na potrebný počet uzlov pri zachovaní koncových SLA a pri nízkom využití uzly šetrne vypína, aby sa predišlo zbytočným výdavkom.

4.3 Dátová transformácia v rámci rôznych platforiem

Pre prostredie slovenského eGovernmentu sú relevantné aj nasledujúce prístupy k dátovému ekosystému, ktoré nespadajú pod tradičný koncept nástrojov ETL alebo ELT, avšak otvárajú iný svet toho, ako medzi sebou integrovať informačné systémy či pracovať s informáciami a znalosťami z viacerých zdrojov.

4.3.1 Dátová transformácia v ekosystéme znalostných grafov – komerčné riešenia

4.3.1.1 Ontotext

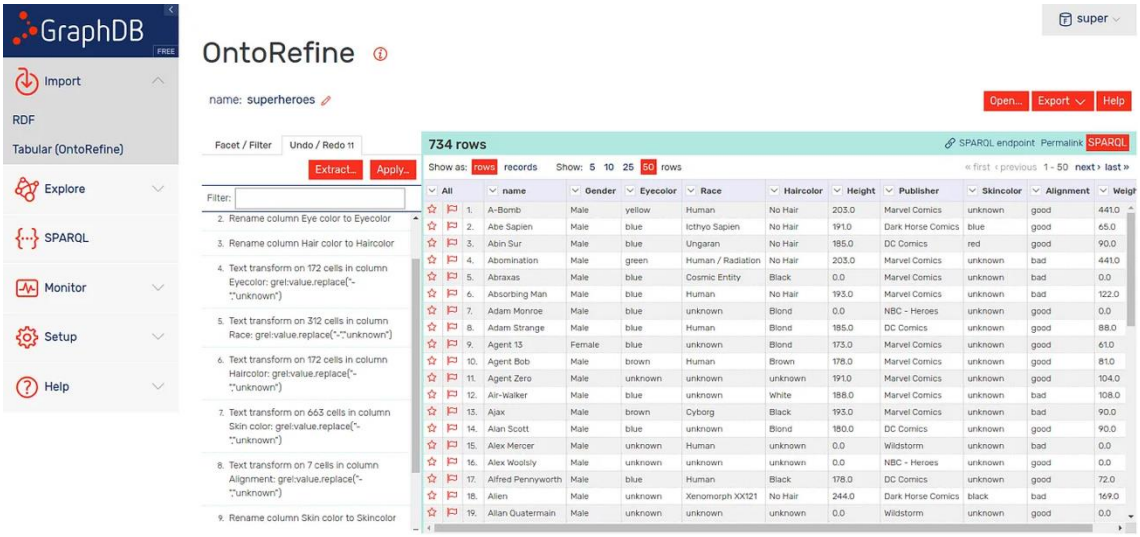
Nástroj OntoRefine podporuje analýzu vstupných surových údajov vo veľa formátoch (TSV, CSV, *SV, XLS, XLSX, JSON, XML), aplikuje rôzne algoritmy čistenia a transformácie údajov, mapuje reťazcové hodnoty na koncepty znalostných grafov a importuje vytvorený model do grafovej databázy GraphDB. Ponúka jednoduchý pracovný postup tvorby RDF údajov založený na deklaratívnych transformáciách a vizuálnom používateľskom rozhraní mapovania. Transformácie sa dajú uložiť a použiť v dávkovom režime pomocou príkazového riadku. Podporuje aj federáciu SPARQL na jednoduchú integráciu transformovaných údajov do existujúceho grafu znalostí.

[63] Zdroj: https://www.stitchdata.com/, Dátum referencie: 24.05.2023

[64] Zdroj: https://www.databricks.com/product/delta-live-tables, Dátum referencie: 24.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Podobne ako v nástroji LinkedPipes, popísanom v kapitole 4.2.2.1, aj nástroj Ontotext poskytuje flexibilnú transformáciu tabuľkových údajov na RDF[65]. OntoRefine je nástroj na transformáciu údajov založený na OpenRefine[66], ktorý je integrovaný do GraphDB[67] pomocou OntoText. GraphDB v Ontotext umožňuje priamo import tabuľkových údajov, ktoré sa dajú prezerať v grafickom rozhraní (Obrázok 8), umožňuje písať operácie SPARQL, ako bolo uvedené aj v príklade LinkedPipes, a výsledné RDF údaje umožňuje vizualizovať ako graf.



Obrázok 8: Importovanie a transformácia tabuľkových údajov do RDF cez OntoRefine

Metadata Studio od spoločnosti Ontotext uľahčuje:

- Vytváranie a kurátorovanie metadáta obsahu ako človekom vytvorený benchmakr pre automatické označovanie;
- Integrovanie rôznych služieb na dolovanie textu („text mining“);
- Získanie prehľad o vlastných referenčných korpusoch a výkonnosti automatického označovania na ich základe, aby sa dali oboje zlepšiť.

Ontotext navyše poskytuje aj celú platformu (Obrázok 9) pre prácu so znalostnými grafmi, vďaka ktorej sa dá:

- Vyvíjať a udržiavať znalostné grafy z rôznych údajov, a to pomocou priebežnej integrácie, normalizácii a prepájaniu údajov z rôznych zdrojov, pričom pri aktualizáciách sa dá udržiavať kvalita údajov.

[65] Zdroj: https://medium.com/wallscope/using-ontorefine-to-transform-tabular-data-into-linked-data-7277ec8c2c0f,

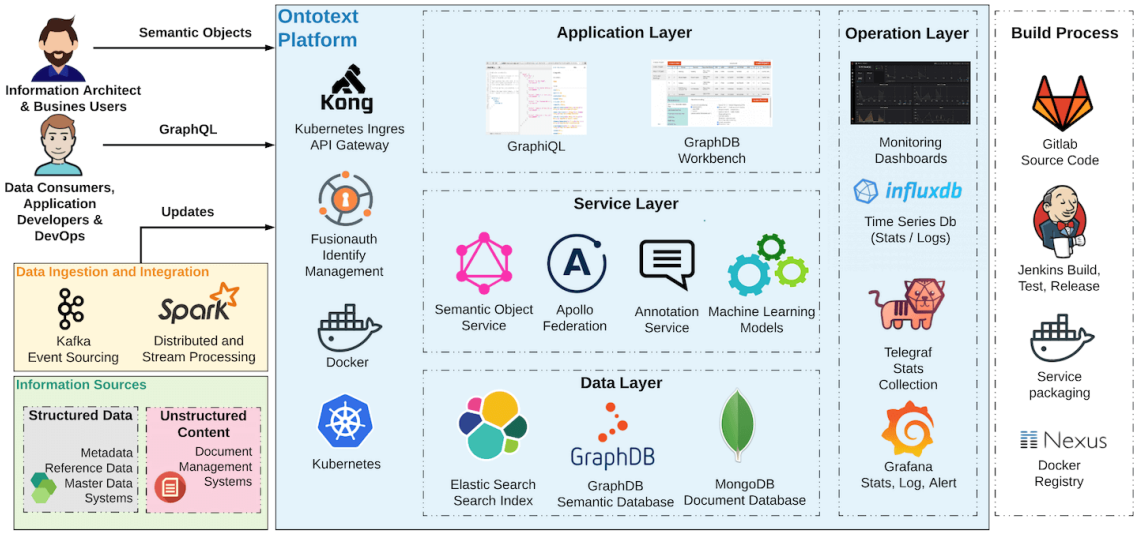
[66] Zdroj: https://openrefine.org,

[67] Zdroj: https://graphdb.ontotext.com,

- Automaticky generovať prístup cez GraphQL (dopytovací jazyk založený na grafoch)[68] z ontológií, čím sa dajú deklarovať zjednodušené informačné pohľady na uľahčenie konzumácie údajov a implementovať riadenie prístupu.
- Generovať sémantické metadáta a získavať znalosti využívaním textovej analýzy na extrahovanie znalostí z neštruktúrovaných dokumentov.
- Efektívne generovanie dotazov SPARQL - nie je potrebné písať a optimalizovať zložité dotazy.
- Jednoduchá integrácia aplikácií vrátane nesémantických zdrojov. Platforma ponúka tiež federáciu, spájanie schém a virtualizáciu údajov.
- Dostupnosť nástrojov priateľských k vývojárom - implementovať používateľské rozhrania sa dá priamo z tvaru údajov.
- Používať autorizáciu a autentifikáciu, a tak uplatňovať všeobecný model na riadenie prístupu k informáciám.
- Využívať vysokú dostupnosť, vyhľadávanie a dopytovanie prostredníctvom GraphDB, ktorá obsahuje zdôvodňovanie („reasoning“), sémantickú podobnosť a triedenie.
- Škálovať údaje, dotazy a transakčné záťaže prostredníctvom integrácie s ElasticSeach a MongoDB.
- Spustiť cloudovo-agnostické nasadenie pomocou Kubernetes, čím sa dá vytvoriť vývojové a produkčné prostredie v priebehu niekoľkých minút.

[68] Zdroj: https://graphql.org, Dátum referencie: 26.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.



Obrázok 9: Architektúra platformy Ontotext[69]

4.3.1.2 PoolParty

PoolParty je komerčný sémantický „middleware“ založený na nástroji LinkedPipes, popísanom v kapitole 4.2.2.1. Efektívne prepája podnikové databázy s aplikáciami, pričom spája všetky štruktúrované a neštruktúrované údaje do takzvaných 360-stupňových pohľadov v rámci organizácie. PoolParty Semantic Suite využíva pokročilé dolovanie textu („text mining“) a algoritmy na spracovanie prirodzeného jazyka (NLP) na automatickú extrakciu relevantných entít a termínov z dokumentov. Okrem extrakcie dôležitých termínov, PoolParty umožňuje tieto entity automaticky označiť na vytvorenie ďalších obohatených metadát. Označený obsah a metadáta sa ukázali ako veľmi cenné pre správu znalostí, vyhľadávače a iné používateľské platformy.

„Pipelines“ pre dátovú transformáciu so zjednotenými pohľadmi sa odohráva v nástroji „Semantic Integrator“, ktorý predstavuje:

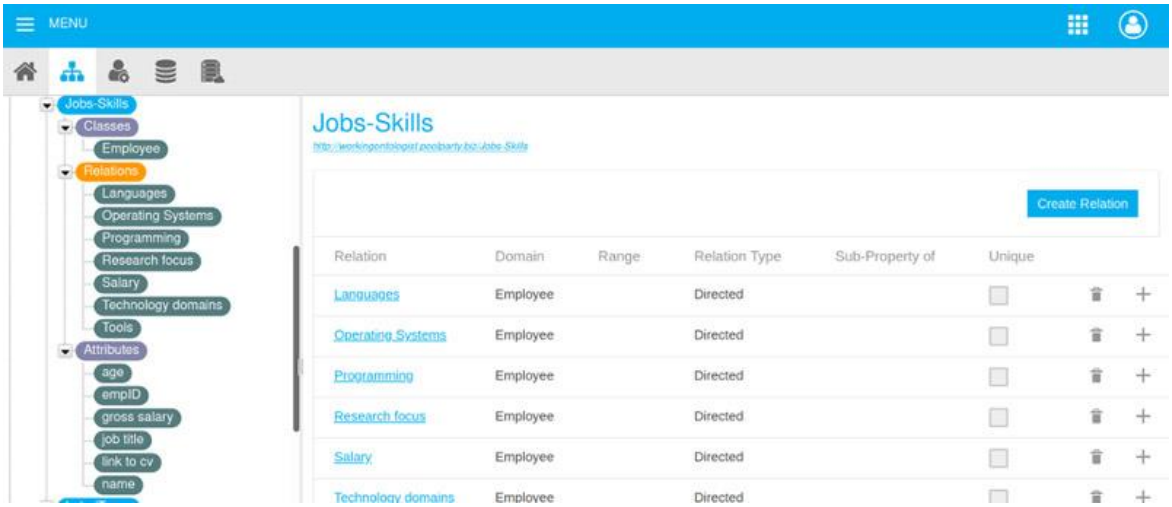
- Platformu na agilnú integráciu údajov,
- Integráciu s grafovými databázami,
- Integráciu s vyhľadávacími „engines“,
- Nástroj na prepojenie a mapovanie entít a transformáciu,
- Server na vyhľadávanie grafov.

Na konverziu tabuľkových údajov, podobne ako v kapitole 4.2.2.1, ponúka PoolParty niekoľko nástrojov. Obrázok 10 zobrazuje vlastnú ontológiu v nástroji PoolParty Ontology Management, ktorá pokrýva doménu Zamestnanie and Zručnosti. Analýza korpusu pomocou PoolParty[70] môže pomôcť doménovým expertom pri vytváraní

[69] Zdroj: https://www.ontotext.com/products/ontotext-platform/,

[70] Zdroj: https://www.poolparty.biz/text-corpus-analysis, Dátum referencie: 26.05.2023

taxonómií od základov tým, že navrhne relevantné termíny a vzťahy, ktoré sú kandidátskymi konceptmi, ako aj nové prepojenia medzi konceptmi.



Obrázok 10: Vytvorenie a správa ontológie v PoolParty

Aby sme mohli prelinkovať údaje, v nástroji vytvárame mapovania, ktoré určujú, ako sa zdrojové štruktúrované údaje transformujú do znalostného grafu na základe mapovania stĺpcov, hlavičiek a atribútov. Keď máme tieto mapovania, môžeme tabuľku programu Excel transformovať na údaje RDF, čo je súbor trojíc (subjekt, predikát, objekt). Paralelne so štruktúrovanými údajmi môžeme transformovať aj neštruktúrované údaje o tých istých subjektoch. Aby boli neštruktúrované údaje prepojiteľné so štruktúrovanými súbormi údajov, extrahujeme znalosti z týchto neštruktúrovaných informácií pomocou PoolParty Extractor[71] - nástroja na dolovanie textu, ktorý kombinuje metódy spracovania prirodzeného jazyka a strojového učenia.

Vzniknutý znalostný graf má byť dynamický v tom zmysle, že sa pravidelne aktualizuje a priebežne obohacuje, čím sa zvyšuje jeho hodnota a úplnosť. Tento krok je veľmi dôležitý - často sa okolo neho vytvára celý model postupov riadenia, ktorý zabezpečuje, aby rôzni ľudia, agenti a odborníci na danú problematiku spolupracovali. Táto myšlienka je ústredným prvkom životného cyklu prepojených údajov, v ktorom rôzne iterácie neustále obohacujú a spresňujú daný znalostný graf.

Ďalším dôležitým komponentom je PoolParty GraphEditor[72], ktorý sa používa na vytváranie, aktualizáciu a odstraňovanie údajov inštancie pomenovaného grafu. Okrem úprav umožňuje GraphEditor vykonávať zložité konjunktívne (AND), ako aj disjunktívne (OR) dotazy, ktoré možno kombinovať s regulárnymi výrazmi. To umožňuje ľuďom, ktorí nepoznajú jazyk SPARQL, písať dopyty jednoduchým výberom tried, vlastností a atribútov v grafickom rozhraní.

[71] Zdroj: https://www.poolparty.biz/poolparty-extractor/, Dátum referencie: 26.05.2023

[72] Zdroj: https://www.poolparty.biz/poolparty-grapheditor, Dátum referencie: 26.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

4.3.2 Dátová transformácia v rámci nástroja Apache Camel

Tomuto nástroju sme sa podrobne venovali v kapitole 2.2. V tejto kapitole sa sústredíme na to, ako v ňom možno realizovať dátovú transformáciu. Camel podporuje prekladač správ podľa podnikových integračných vzorov (kapitola 3.3.1)[73].

Dátová transformácia v nástroji Camel sa zvyčajne vykonáva jedným z nasledujúcich spôsobov:

- Mapovanie pomocou kódu Java a typových konvertorov typov: Pri tomto prístupe musia byť vstupné aj výstupné údaje objektmi jazyka Java a na mapovanie medzi nimi sa používajú príkazy jazyka Java. Keď treba transformovať zdrojový typ na cieľový typ, napíše sa metóda, ktorá vytvorí nový cieľový objekt a naplní ho príslušnými poľami zo zdrojového objektu. Potom sa povie Camel, aby túto metódu zavolal (napr. pomocou „`bean`“ alebo možno aj procesora). Tento prístup využíva plnohodnotnú Javu. Často si vyžaduje napísanie veľkého množstva kódu v Jave. Je však tiež silne typovaný, čo je ale aj dobrá vec. Ak pracujeme s XML, možno použiť JAXB[74] jazyka Java (implementovaný v nástrojoch ako Apache CXF[75]) na vytvorenie tried jazyka Java zo schémy XML a potom previesť tento objekt XML do jazyka Java.
- Pomocou špecializovaného komponentu Camel, napríklad XSLT[76], XSLT Saxon[77] pre lepší výkon a sofistikovanejšie transformácie, Bindy[78] (Bindy je prispôsobiteľný dátový formát, ktorý dokáže spracovať súbory s pevnou šírkou (napr. FIX) a premenlivou šírkou (napr. CSV). Bindy je veľmi dobrá knižnica, ktorá pomáha v núdzi, najmä ak sa pracuje s niektorými dosť nezvyčajnými alebo starými formátmi súborov. Pokiaľ pomocou Bindy dokážete stanoviť pravidlá formátu súboru - napríklad koľko znakov má prvý stĺpec, aké sú oddeľovače atď. – dá sa s daným súborom pracovať v nástroji Camel) alebo Atlasmap[79] (Dokáže mapovať medzi objektmi JSON, XML a Java.)
- „Marshalling/unmarshalling[80]“ pomocou dátových formátov, ako sú CSV a JSON: Dátové formáty v nástroji Camel sú obslužné vrstvy na prácu so súbormi ako ZIP, Avro, súbory kódované v Base64, súbory HL7 (pre zdravotníctvo), JSON, Protobuf

[73] Zdroj: https://camel.apache.org/components/3.20.x/eips/message-translator.html, Dátum referencie: 26.05.2023

[74] Zdroj: https://javaee.github.io/jaxb-v2/doc/user-guide/, Dátum referencie: 26.05.2023

[75] Zdroj: https://cxf.apache.org/docs/jaxb.html, Dátum referencie: 26.05.2023

[76] Zdroj: https://camel.apache.org/components/3.20.x/xslt-component.html, Dátum referencie: 26.05.2023

[77] Zdroj: https://camel.apache.org/components/3.20.x/xslt-saxon-component.html. Dátum referencie: 26.05.2023

[78] Zdroj: https://camel.apache.org/components/3.20.x/dataformats/bindy-dataformat.html, Dátum referencie: 26.05.2023

[79] Zdroj: https://camel.apache.org/components/3.20.x/atlasmap-component.html, Dátum referencie: 26.05.2023

[80] Zdroj: https://www.jguru.com/what-is-the-meaning-of-marshalling-and-unmarshalling/. Dátum referencie: 26.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

a mnohé ďalšie. Po nakonfigurovaní dátového formátu sa dá zapojiť do kroku „marshal“ alebo „unmarshal“.

- Použitie „enginu“ na šablónovanie, napríklad Velocity[81] alebo Mustache[82]: Poslednou možnosťou sú šablóny. Tie sú veľmi užitočné, ak potrebujete vytvoriť akýkoľvek textový výstup, ktorý budú vidieť ľudia, napríklad vo forme dokumentu či webovej stránky. Šablónové „engines“ vedia prijať šablónu a nejaké zdrojové údaje a vytvoriť z nich dokument čitateľný pre človeka.

[81] Zdroj: https://camel.apache.org/components/3.20.x/velocity-component.html, Dátum referencie: 26.05.2023

[82] Zdroj: https://mustache.github.io, Dátum referencie: 26.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

5 Výber vhodných metód a nástrojov pre jednotlivé prípady použitia

5.1 Výmena údajov medzi OVM

Pri rozvíjaní Centrálnej integračnej platformy a pri implementácií vhodných vzorcov pre aplikačnú a dátovú integráciu je hlavnou výzvou súbežná snaha o využitie všetkých troch konceptov naraz, ako bolo popísané v kapitole 2.1:

1. Dátové vlákna („Data Fabric“), podporené nasadenou platformou Talend,
2. Dátová sieť („Data mesh“), ktorá bola úspešne odskúšaná ako pilot v rámci rozvoja CIP a zároveň ide o veľmi sľubný koncept, overený aj dobrou praxou v priemysle (viď príklad dobrej praxe v kapitole 7.3),
3. Znalostné grafy („Knowledge graphs“), na ktorých je postavený Centrálny model údajov, MOU aj Open Data.

V aktuálnej situácii nie je možné rozhodnúť v prospech jedného konceptu. Javí sa ako dobrá stratégia paralelne rozvíjať všetky pre rôzne prípady použitia, a popritom sledovať vývoj na trhu nástrojov, kde vidno istý trend konvergencie. Nástroj Talend podporuje integráciu aj cez vzorec „messaging“ a dátové siete sa snažia inšpirovať sémantickou interoperabilitou v znalostných grafoch. Aj vzhľadom na nemožnosť v krátkej dobe inovovať všetky informačné systémy verejnej správy, ktoré minimálne v každej agende pracujú s vlastnými dátovými modelmi a nie sú prispôsobené CMÚ, bude potrebné aj naďalej fungovať s týmito troma konceptmi naraz. Aj keď jedným z enterprise integračných vzorcov je vytvorenie centrálneho kanonického dátového modelu, problémom ostáva, že ani Talend či Apache Kafka a Apache Camel nepodporujú dátový model založený na znalostných grafoch a RDF. Preto nasledujúca Tabuľka 3 uvažuje s výmenou údajov medzi OVM založenou na konceptoch 1 a 2, keďže aplikačná a dátová integrácia založená na znalostných grafoch nie je rozšírená vo veľkých podnikoch v súkromnej sfére, a tým pádom ani podporená často využívanými nástrojmi.

Tabuľka 3: Odporúčané metódy a nástroje pre výmenu údajov medzi OVM

Typ odporúčania	Názov (s odkazom na príslušnú kapitolu)	Popis
Metódy	Metódy aplikované pred dátovou transformáciou	Pre lepšiu interpretáciu alebo ľahšie spracovanie údajov na zdroji môže byť potrebné napríklad upraviť alebo doplniť číselníkové hodnoty, alebo naopak niektoré dátové prvky odfiltrovať.
	Metóda na zlúčenie údajov z viacerých zdrojov a/alebo zmenu schémy cez mapovanie~~Metóda na zlúčenie údajov z viacerých zdrojov a/alebo zmenu schémy cez mapovanie~~	Táto metóda sa použije vtedy, cieľový systém potrebuje konzumovať údaje naraz s viacerých zdrojových informačných systémov, pričom ešte môže vyžadovať aj zmenu schémy.

Tento projekt je podporený z Európskeho sociálneho fondu.

Typ odporúčania	Názov (s odkazom na príslušnú kapitolu)	Popis
	Metódy transformácie podľa podnikových integračných vzorcov („Enterprise Integration Patterns“)	Táto metóda dátovej transformácie pomáha zavádzať do praxe rôzne vzorce aplikačnej integrácie.
Nástroje	Talend	Tento nástroj ponúka rôzne užitočné komponenty, ktoré sa nachádzajú v skupine pre spracovanie (Processing) a ktoré podporujú mnohé metódy uvedené v kapitole 3.4. Talend v tejto verzii ponúka aj priamo komponenty pre transformáciu, ktoré sú v súlade s metódami v kapitole 3.3. K dispozícii je aj komponent cMap[83], ktorý vykonáva transformácie (nazývané mapy) medzi rôznymi zdrojmi a cieľmi využitím možností aplikácie Talend Data Mapper. Tento komponent transformuje údaje zo širokej škály zdrojov do širokej škály cieľov. Pre transformáciu zdrojového súboru XML na cieľový pre vybraných konzumentov možno použiť aj komponent tXSLT[84]. Talend má navyše tú výhodu, že jednotlivé joby možno rozširovať aj cez vlastný Java kód, teda do istej miery možno vytvoriť aj dátové transformácie na mieru.
	Dátová transformácia v rámci nástroja Apache Camel	Tento nástroj sa odporúča využívať len keď sa v rámci Centrálnej integračnej platformy prejde na koncept dátovej siete (kapitola 2.1.2) a zároveň sa vyhodnotí opodstatnenosť tohto nástroja (kapitola 2.2.3). Aj cez nástroj Apache Kafka sa dajú robiť transformácie, a to či už cez Kafka Connect – Transform[85], kam sa dajú písať aj vlastné rozšírenia v Jave. Ďalšie prístupy k transformácii XML dokumentov v ekosystéme s nástrojom Kafka sú popísané tu[86].

[83] Zdroj: https://help.talend.com/r/en-US/7.3/mediation-map/cmap, Dátum referencie: 29.05.2023

[84] Zdroj: https://help.talend.com/r/en-US/7.3/xml/txslt, Dátum referencie: 29.05.2023

[85] Zdroj: https://datacadamia.com/dit/kafka/connect/transform, https://www.confluent.io/blog/kafka-connect-single-message-transformation-tutorial-with-examples/, https://github.com/OneCricketeer/schema-registry-transfer-smt, Dátum referencie: 29.05.2023

[86] Zdroj: https://www.kai-waehner.de/blog/2020/09/25/kafka-xml-messages-transformation-connector-middleware-comparison-connect-smt-esb-etl-web-services-soap-wsdl-schema/, Dátum referencie: 29.05.2023

5.2 Analytické spracovanie údajov

Tento prípad použitia sa týka predovšetkým analytického spracovania údajov v Konsolidovanej analytickej vrstve (KAV), ktorej sa venujeme v dokumente 4.2.1 Koncept pre zavádzanie analytického spracovania údajov do praxe. V rámci tohto dokumentu bola vybraná architektúra založená na dátovom „lakehouse“, ktorý kombinuje výhody dátového skladu a dátového jazera a snaží sa eliminovať nevýhody oboch. Dátové lakehousy využívajú procedúry ETL aj ELT a môžu uchovávať akýkoľvek typ údajov vrátane štruktúrovaných, pološtruktúrovaných a neštruktúrovaných údajov. Z tohto dôvodu nie je také jednoduché vybrať tú správnu metódu a nástroj z tabuľky (Tabuľka 4) a veľmi záleží od zdrojového systému, povahy údajov a prípadu použitia.

Tabuľka 4: Odporúčané metódy a nástroje pre analytické spracovanie údajov

Typ odporúčania	Názov (s odkazom na príslušnú kapitolu)	Popis
Metódy	Metódy aplikované pred dátovou transformáciou	Pre lepšie štatistické spracovanie údajov môže byť potrebné upraviť alebo doplniť číselníkové hodnoty.
	Metóda na zlúčenie údajov z viacerých zdrojov a/alebo zmenu schémy cez mapovanie~~Metóda na zlúčenie údajov z viacerých zdrojov a/alebo zmenu schémy cez mapovanie~~	Táto metóda sa použije vtedy, ako sa pre koncept transformácie zvolia procedúry ETL.
	Metódy aplikované len pred analýzou údajov	Tieto metódy môžu byť využité či už v koncepte ETL alebo ELT, hoci v koncepte ELT ich je jednoduchšie spraviť naprieč všetkými datasetmi z rôznych zdrojov, keď už údaje zo všetkých zdrojov sa nachádzajú na jednom mieste
Nástroje	Talend	Tento nástroj sa využíva vtedy, ak sa zvolili procedúry ETL. Užitočným komponentom je spomínaný tMap[87] alebo tHMap[88]. Ďalšie užitočné komponenty sa nachádzajú v skupine pre spracovanie (Processing), ktoré podporujú mnohé metódy uvedené v kapitole 3.5 a 3.4. Avšak Talend ponúka vo verzii 7.3 Talend Data Services aj komponenty ELT, ktoré ale sú už prepojené s konkrétnym databázovým systémom, tak nemusia vyhovovať pre každý scenár. Talend má navyše tú výhodu, že

[87] Zdroj: https://help.talend.com/r/en-US/7.3/data-integration-job-examples/tmap-job-example, Dátum referencie: 29.05.2023

[88] Zdroj: https://help.talend.com/r/en-US/7.3/data-integration-job-examples/tmap-job-example, Dátum referencie: 29.05.2023

Typ odporúčania	Názov (s odkazom na príslušnú kapitolu)	Popis
		jednotlivé joby možno rozširovať aj cez vlastný Java kód, teda do istej miery možno vytvoriť aj dátové transformácie na mieru.
	Open-source ETL nástroje	Tu je najrelevantnejší nástroj Hadoop pre špecifické prípady dátových zdrojov a prípadov použitie pre analytické spracovanie. Prípadne možno zvážiť Talend Open Studio, ak zdrojový systém nie je ešte integrovaný s CIP.
	Cloudové služby pre ETL (alebo aj ELT)	Výber konkrétnej služby bude veľmi závisieť na konkrétnom návrhu dátového a analytického „stacku“ pre KAV, ako je naznačene v dokumente 4.2.1 Koncept pre zavádzanie analytického spracovania údajov do praxe. Sem zapadá aj dátová integrácia a následné analytické spracovanie špecifických údajov dostupných cez API, ako napríklad rôzne streamovacie, telemetrické, priestorové či iné špecifické údaje a datasety, dostupné zo zdroja cez dedikované API. Veľmi populárnym cloudovým riešením je aj spomínaný nástroj pre transformáciu už priamo v dátovom lakehouse dbt[89].

Organizácie často používajú ETL, ELT alebo kombináciu týchto dvoch prístupov k dátovej transformácii. Je však niektorý z týchto prístupov lepší?

Tabuľka 5: Porovnanie prístupov ETL a ELT

Parameter	ETL	ELT
Proces	ETL zahŕňa: • extrakciu údajov, • ich načítanie na „staging“ server, • transformáciu údajov na spracovateľskom serveri, • ich načítanie do úložiska.	ELT pozostáva z: • extrakcie údajov, • ich načítania do cieľového servera / úložiska údajov v surovej forme, • transformácie údajov, ktorá sa uskutočňuje podľa potreby a využíva sa pri nej schopnosť spracovania cieľového systému.

[89] Zdroj: https://www.getdbt.com, Dátum referencie: 29.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Parameter	ETL	ELT
Ideálne pre	Postup ETL je ideálny pre lokálne, organizované a relačné dátové modely a funguje lepšie pri menšom objeme údajov.	Postup ELT je vhodný pre štruktúrované a neštruktúrované zdroje údajov v cloude a je vhodný najmä pre veľké objemy údajov.
Efektivita / rýchlosť	Metóda ETL je časovo náročná, pretože najprv načíta údaje na „staging“ server a potom do cieľového systému. Nakoniec sa čaká na transformáciu údajov, takže čas čakania sa zvyšuje s rastúcou veľkosťou údajov.	Metóda ELT je rýchlejšia, pretože načítanie údajov do cieľového systému sa uskutoční len raz. Veľkosť údajov nemá vplyv na rýchlosť procesu ELT. Aj vďaka cloudu je tak škálovateľná.
Kompatibilita	Technológia ETL sa dokáže pripojiť k existujúcim systémom a dokáže filtrovať a integrovať údaje.	ELT je vynikajúcou možnosťou na import údajov do dátového jazera alebo na implementáciu transformácií založených na jazyku SQL. Neponúka však rovnakú úroveň podpory existujúcich zdrojových systémov ako ETL.
Hardvér	Väčšina nástrojov ETL optimálne funguje s lokálnymi servermi s úložiskami, čo celý proces predražuje. Avšak na druhej strane metóda ETL generuje výsledky, ktoré zabezpečujú, že sa zhromažďujú a do dátového lakehouse načítavajú len relevantné údaje, čím sa znižujú náklady na výpočty, úložiská a správu.	Proces ELT v cloude si nevyžaduje žiadne dodatočné výdavky na hardvér.
Bezpečnosť a súlad s reguláciou	ETL eliminuje všetky citlivé alebo životne dôležité údaje, aby boli pred uložením do skladu chránené pred hackermi. Väčšina nástrojov ETL je v súlade s nariadeniami ako GDPR.	Keďže všetky surové údaje sa ukladajú priamo do cieľa, únik údajov a hackerské útoky predstavujú riziko. Navyše, ak sa cloud nachádza v zahraničí, môže dôjsť k porušeniu medzinárodných dohôd na dodržiavanie súladu údajov. Avšak keďže ELT uchováva surové údaje, uľahčuje tým analýzu každej fázy dátovej „pipeline“, a takisto je odhalenie problému rýchlejšie, keď sú k dispozícii všetky medzivýsledky. Takže sa dá ľahšie auditovať.

Prípady použitia prístupu ETL

Tento projekt je podporený z Európskeho sociálneho fondu.

ETL je skvelý prístup, keď dané analytické spracovanie potrebuje špecifický formát údajov, pretože pred načítaním mení údaje. ETL funguje najlepšie, keď:

- Existuje nesúlad v podporovaných typoch údajov medzi zdrojom a cieľom.
- Existuje obmedzenie schopností cieľového servera rýchlo zvýšiť výpočtové kapacity.
- Chcete ukladať všetky štruktúrované a neštruktúrované údaje vo vašej organizácii bez ohľadu na ich veľkosť.
- Existuje potreba prístupu k ľahko dostupnej komunite odborníkov a súborom nástrojov ETL.
- Sú potrebné zdroje na správu dátových jazier.
- Existujú obavy o bezpečnosť, ktoré sťažujú uchovávanie surových údajov.

Prípady použitia prístupu ELT

ELT je skvelý prístup, keď sa zameriava na cloudový dátový lakehouse (umiestnený vo verejnom cloude). ELT najlepšie funguje v situáciách, keď

- Menšie datasety nevyžadujú rozsiahle transformácie,
- Máte zdroje na to, aby ste si mohli ponechať odborníkov na ELT,
- Robustná cieľová centrálna databáza založená na cloude analyzuje prichádzajúce dátové toky,
- Organizácie nemusia dodržiavať GDPR alebo iné regulačné požiadavky, alebo údaje určené na analytické spracovanie neobsahujú citlivé údaje.

5.3 Zverejňovanie otvorených údajov

Otvorené údaje sa zverejňujú na základe publikačného minima pre štátnu[90] a verejnú správu[91], čo predstavuje minimálne požadovanú podmnožinu otvorených údajov na publikovanie. Publikačné minimum je reprezentované v rôznych dátových formátoch. Na základe strategickej priority Otvorené údaje[92] je preferovaný formát RDF ako najvyšší stupeň interoperability a strojovej spracovateľnosti. I keď táto povinnosť patrí jednotlivým OVM, vzhľadom na chýbajúcu softvérovú podporu poskytuje Dátová kancelária maximálnu súčinnosť pri tvorbe tohto formátu. V súčasnosti preto stačí mať CSV formát daného datasetu v predpísanej zverejnenej forme a v spolupráci s Dátovou kanceláriou bude dané RDF vytvorené – tomuto postupu je prispôsobená aj Tabuľka 6. Výhodou formátu RDF je, že používa jednotné referencovateľné identifikátory registrované v MetaIS na identifikáciu entít naprieč všetkým informačnými systémami verejnej správy a na definovanie významu Centrálny dátový model údajov.

[90] Zdroj: https://datalab.digital/publikacne-minimum-statnej-spravy/, Dátum referencie: 29.05.2023

[91] Zdroj: https://metais.vicepremier.gov.sk/publicspace?pageId=67145883, Dátum referencie: 29.05.2023

[92] Zdroj: https://www.mirri.gov.sk/wp-content/uploads/2018/10/SP_Otvorene_udaje_schvalena-2.pdf, Dátum referencie: 29.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Tabuľka 6: Odporúčané metódy a nástroje pre zverejňovanie otvorených údajov

Typ odporúčania	Názov (s odkazom na príslušnú kapitolu)	Popis
Metódy	Metódy aplikované pred dátovou transformáciou	Všetky konkrétne prípady v rámci kapitol 3.1.1 až 3.1.4. boli identifikované pri príprave dátovej transformácie objektov evidencie na moje údaje do platformy MOU. Takže pre to sú relevantné aj pre otvorené údaje, ktoré sú tiež založené na štandarde RDF a CMÚ (ak sa môže rovnaký dataset zverejniť aj ako otvorené údaje, ak neobsahuje osobné údaje alebo ak je anonymizovaný).
	Metódy aplikované na zmenu dátového modelu v prípade zdrojového formátu XML do cieľového formátu XML alebo RDF/XML s využitím XSLT~~Metódy aplikované na zmenu dátového modelu v prípade zdrojového formátu XML do cieľového formátu XML alebo RDF/XML s využitím XSLT~~	Momentálne ide o kľúčovú metódu, ktorou sa transformujú zdrojové XML súbory v ľubovoľnom dátovom modeli na dátový model podľa CMÚ a štandard RDF. Teoreticky je túto metódu možné použiť aj na JSON. Odporúčanie W3C na konverziu CSV do RDF je iné[93] a dostupné vo viacerých nástrojoch.
	Metóda na zlúčenie údajov z viacerých zdrojov a/alebo zmenu schémy cez mapovanie~~Metóda na zlúčenie údajov z viacerých zdrojov a/alebo zmenu schémy cez mapovanie~~	Táto metóda sa použije pred aplikovaním samotnej procedúry dátovej transformácie v rámci procedúr ETL.
Nástroje	Linked Pipes pre podporu znalostných grafov a CMÚ	Momentálne ide o preferovaný a sémantickou komunitou overený nástroj. Veľkou výhodou je aj grafické používateľské rozhranie pre vizualizáciu a nastavenie dátových „pipelines“. Pomocou tohto nástroja sa dá aj jednoducho spraviť konverzia z CSV do RDF[94].
	Dátová transformácia v ekosystéme znalostných grafov – komerčné riešenia	Nástroj PoolParty vychádza z Linked Pipes, avšak poskytuje viaceré praktické rozšírenia, podobne ako nástroj Ontotext. Príkladom je interaktívna vizualizácia a analýza grafových údajov, ako aj prepájanie s externými znalostnými grafmi, ako aj možnosť konverzie textových súborov cez strojové anotácie do

[93] Zdroj: https://www.w3.org/TR/csv2rdf/, Dátum referencie: 29.05.2023

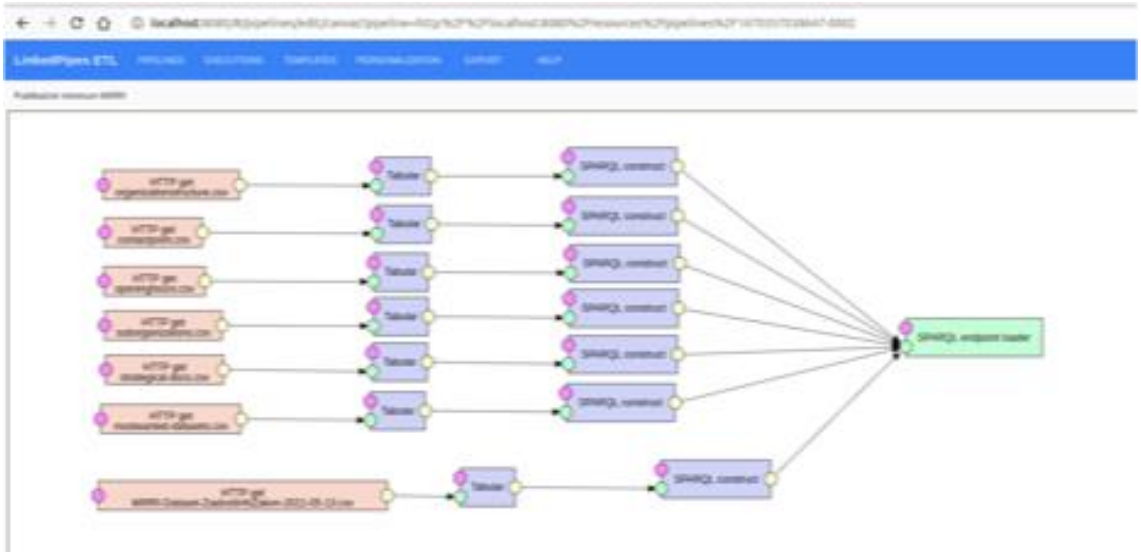
[94] Zdroj: https://etl.linkedpipes.com/tutorials/how-to/convert_csv_to_rdf, Dátum referencie: 29.05.2023

Typ odporúčania	Názov (s odkazom na príslušnú kapitolu)	Popis
		znalostných grafov. Nevýhodou sú však vysoké ročné licencie.

Nasledujúci Obrázok 11 naznačuje, ako možno konvertovať CSV na RDF v nástroji LinkedPipes na splnenie publikačného minima.

Príklad na obrázku (Obrázok 11) pozostáva z:

- Vstupu: tabularizovaný CSV dataset kontaktných miest MIRRI, ktorý sa načítava vo viacerých častiach ako organizačná štruktúra a otváracie hodiny cez protokol HTTP,
- Následne sa pomocou štandardu SPARQL cez jeho povel „Construct“[95] v ďalšej časti „pipeline“ vytvára RDF graf,
- Výstupu: RDF dataset kontaktných miest.



Obrázok 11: Ukážka grafického používateľského rozhrania v LinkedPipes pre transformáciu v rámci publikačného minima

5.4 Poskytovanie mojich údajov cez MOU

Platforma MOU je založená na projekte s otvoreným zdrojovým kódom Solid[96]. Solid je skratkou pre sociálne prelinkované údaje („Social Linked Data“), teda ide o Linked Data, ktoré vychádzajú zo štandardov W3C ako RDF, len môžu obsahovať aj osobné údaje o dotknutej osobe alebo subjekte. Preto údaje, ktoré sa ukladajú v MOU sú plne v súlade

[95] Zdroj: https://www.w3.org/TR/rdf-sparql-query/#construct, Dátum referencie: 29.05.2023

[96] Zdroj: https://solidproject.org/, Dátum referencie: 30.05.2023

s Centrálnym modelom údajov a sú v štandarde RDF ako otvorené údaje. Údaje zo zdrojových informačných systémov verejnej správy preto treba pred ich nahratím do MOU transformovať. Keďže MOU je napojené na IS CSRÚ, zvolil sa nástroj dátovej transformácie v podobe modulu na mieru, implementovaný cez mikroslužby, ktorý využíva štandard XSLT a je rozšírením CIP, ako aj uvádza Tabuľka 7

Tabuľka 7: Odporúčané metódy a nástroje pre poskytovanie mojich údajov cez MOU

Typ odporúčania	Názov (s odkazom na príslušnú kapitolu)	Popis
Metódy	Metódy aplikované pred dátovou transformáciou	Všetky konkrétne prípady v rámci kapitol 3.1.1 až 3.1.4. boli identifikované práve pri príprave dátovej transformácie objektov evidencie na moje údaje do platformy MOU.
	Metódy aplikované na zmenu dátového modelu v prípade zdrojového formátu XML do cieľového formátu XML alebo RDF/XML s využitím XSLT~~Metódy aplikované na zmenu dátového modelu v prípade zdrojového formátu XML do cieľového formátu XML alebo RDF/XML s využitím XSLT~~	Momentálne ide o kľúčovú metódu, ktorou sa transformujú zdrojové XML súbory v ľubovoľnom dátovom modeli na dátový model podľa CMÚ a štandard RDF.
	Metódy transformácie podľa podnikových integračných vzorcov („Enterprise Integration Patterns“)	Táto metóda dátovej transformácie v kapitole 3.3.2 môže pomôcť pri vylepšení integračného vzorca pre MOU a pre vyriešenie zmien v datasetoch u niektorých zdrojov v budúcnosti. Metódu v kapitole 3.3.1 bude možné použiť, pokiaľ zdrojový systém bude využívať dátový model CMÚ bez implementácie štandardu RDF.
Nástroje	Naprogramovanie modulu dátovej transformácie „na mieru“ s využitím knižníc	Momentálne ide o preferovanú a validovanú formu nástroja, ktorý implementuje odporúčania W3C na transformáciu XML do RDF cez štandard XSLT. Nevýhodou je prácnosť písania XSLT transformácií a nedostatok ľudských zdrojov. Pre dátovú transformáciu údajov do MOU (kapitola 5.4.1) a pre potreby EDIW (kapitola 5.4.2) a SDG (kapitola 5.4.3) sa momentálne zvolila možnosť b).
	Talend	Pre transformáciu zdrojového súboru XML na cieľový dátový model v RDF podľa CMÚ možno použiť aj komponent tXSLT[97]. Avšak tento job by bolo potrebné rozšíriť aj cez vlastný Java kód,

[97] Zdroj: https://help.talend.com/r/en-US/7.3/xml/txslt, Dátum referencie: 29.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Typ odporúčania	Názov (s odkazom na príslušnú kapitolu)	Popis
		minimálne o serializáciu do formátu JSON LD a validáciu cez SHACL.

5.4.1 Bežné poskytovanie mojich údajov

Jedným z hlavných cieľov dátovej transformácie je zvýšenie interoperability medzi systémami verejnej správy, priblíženie sa odporúčaniam a štandardom EÚ pre interoperabilitu verejnej správy a medzi krajinami EÚ a zjednodušenie prístupu k údajom pre konzumentov. Za týmto účelom vznikla prvá verzia transformačného modulu v rámci IS CSRÚ, ktorý bude zabezpečovať transformáciu dát z aktuálneho XML formátu do RDF formátu v 5★ kvalite prelinkovaných údajov („Linked Data“)[98].

Konzumentami transformovaných údajov môžu byť, rovnako ako v prípade netransformovaných údajov, informačné systémy v správe OVM, ale najmä platforma MOU. V rámci projektu MOU sa počíta s využitím údajov aj na právne záväzné úkony. Preto je základnou požiadavkou, aby údaje po transformácii mali minimálne rovnakú dôveryhodnosť ako pred transformáciou. Presnejšie, aby procesom transformácie, pri ktorom dochádza k zmene štruktúry, formátu aj objemu pôvodných dát, nedošlo k zníženiu ich dôveryhodnosti.

V procese transformácie sa predpokladá aj obohatenie a doplnenie pôvodných údajov tak, aby spĺňali požiadavku na kvalitu výstupu na úrovni 5★ RDF a aby boli v súlade s CMÚ (Podrobne sa CMÚ venuje dokument 1.1.2 Štandardizácia pre modelovanie údajov).

Pre zachovanie dôveryhodnosti údajov v procese transformácie je potrebné zabezpečiť:

1. Do výsledného transformovaného datasetu sa musia dostať všetky dátové prvky objektu evidencie z pôvodného datasetu. Nedôjde tak k strate zásadných informácií oproti pôvodným údajom.
2. Transformácia zohľadňuje a aplikuje štandardy pre modelovanie údajov, popísaných v dokumente 1.1.2 Štandardizácia pre modelovanie údajov UP, najmä kapitola 2.3 Modelovanie údajov vo verejnej správe na Slovensku, z toho predovšetkým (ale nie len) využívanie CMÚ a dodržiavanie pravidiel pre jednotné referencovateľné identifikátory (URI).
3. Proces transformácie sa ďalej vykonáva pomocou štandardov, metód a nástrojov definovaných v kapitole 5.4, pričom na výsledný RDF dataset pridá kvalifikovaná pečať MIRRI.
4. Pri obohacovaní sa vychádza výlučne z informácií obsiahnutých v pôvodných údajoch a informácií CMÚ. Je možné aplikovať biznis pravidlá, ktoré zabezpečia zlepšenie kvality údajov, ich výsledok však musí byť jednoznačný. Pri

[98] Zdroj: https://5stardata.info/en/, Dátum referencie: 08.03.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

nejednoznačnom výsledku transformácie údaju sa údaj nesmie transformovať. Príkladom jednoznačnej transformácie s obohatením je zmena pôvodného údaju pohlavia fyzickej osoby v podobe „M“ (ako muž) na číselníkovú hodnotu CMÚ základného číselníka Pohlavie aj s definovanou URI https://data.gov.sk/def/sex/1 a prípadne s doplňujúcim údajom preferovaného pomenovania „Muž“ a referenciou na daný číselník https://data.gov.sk/set/codelist/CL003003.

Príkladom nejednoznačnej transformácie môže byť stotožnenie adresného bodu z Centrálneho registra adries (jednoznačného identifikátora adresného bodu) na základe pôvodného neúplného údaju adresy v neštruktúrovanej forme “Čakanovce 79”, kedy nie je možné rozhodnúť, o ktorú obec sa presne jedná, nakoľko sa ich na Slovensku nachádza viacero. Vtedy sa do transformovaných údajov má dostať iba pôvodný údaj adresy, nie však identifikátor adresného bodu.

5. Dôveryhodnosť samotného prostredia vykonávajúceho transformáciu - z tohoto pohľadu za dôveryhodné považujeme samotné zdrojové IS VS v správe OVM a IS CSRÚ / CIP.

Pre zachovanie dôveryhodnosti údajov vo vzťahu k CMÚ je potrebné navyše zabezpečiť správnosť a aktuálnosť použitých údajov CMÚ v procese samotnej transformácie. Toto je ideálne zabezpečiť priamou aplikačnou integráciou, napríklad s portálom znalosti.gov.sk alebo metais2.vicepremier.gov.sk na úrovni API, prípadne pravidelnou aktualizáciou týchto údajov v lokálnych úložiskách modulov transformácie, a zamedziť tak použitiu neaktuálnych alebo neplatných údajov v procese transformácie. Pri aktualizáciách CMÚ je tiež nevyhnutné myslieť na spätnú kompatibilitu a interoperabilitu transformovaných údajov.

5.4.2 Poskytovanie mojich údajov cez EDIW

Projekt EDIW[99] nemá definované vlastné dátové modely pre elektronické doklady. Vodičský preukaz vychádza zo štandardu ISO/IEC 18013-5, v ktorom má označenie mDL („mobile driving license“)[100]. Atribúty - nazývané mDL dátové prvky a definované v tomto štandarde - sú len tie, ktoré môžu byť potrebné pre vodičský preukaz v súlade s normou ISO (táto norma definuje povinné a nepovinné dátové prvky)[101]. Tieto dátové prvky musia byť zakódované v stručnom binárnom objekte (CBOR) alebo JavaScript Object Notation (JSON), v závislosti od toho, či čítačka mDL získava údaje z mDL alebo od orgánu vydávajúceho mDL. Všetky dátové prvky majú identifikátor. Hodnota dátového prvku môže byť akákoľvek platný CBOR alebo JSON dátový prvok, vrátane mapy alebo poľa. Dátový model v tejto norme je však nastavený tak, aby sa okrem mDL mohli vytvárať aj iné mobilné poverenia („credentials“), označované súhrnne ako mdoc, jednoducho definovaním iného menného priestoru a definovaním nových dátových prvkov v rámci tohto menného priestoru pri dodržaní všetkých ostatných ustanovení normy. Všetky dátové prvky mDL sú definované v rámci menného priestoru s hodnotou

[99] Zdroj: https://utimaco.com/news/blog-posts/eidas-20-roadmap-toolbox-and-european-digital-identity-wallet-architecture, Dátum referencie: 30.05.2023

[100] Zdroj: https://collateral-library-production.s3.amazonaws.com/uploads/asset_file/attachment/36416/CS676613_-_Digital_Credentials_promotion_campaign-White_Paper_R3.pdf, Dátum referencie: 30.05.2023

[101] Zdroj: https://unece.org/fileadmin/DAM/trans/doc/2013/wp1/WP1-Presentation-2013-5e.pdf, Dátum referencie: 30.05.2023

„org.iso.18013.5.1.“ Aby sa predišlo kolíziám názvov, norma navrhuje používať pre hodnoty menného priestoru prístup opačného rozšírenia domény, ako je uvedené v hodnote mDL menného priestoru vyššie.

Okrem menných priestorov sa v štandarde ISO/IEC 18013-5 používa aj koncept typov dokumentov, ktoré používajú podobnú konvenciu. Hodnota typu dokumentu pre mDL je „org.iso.18013.5.1.mDL“. Rovnako ako v prípade menných priestorov, hocikto môže špecifikovať ďalšie typy dokumentov (mdoc). Typ dokladu sa uvádza v každej správe o žiadosti alebo odpovedi. Dokument daného typu môže obsahovať dátové prvky z niekoľkých rôznych menných priestorov. To umožňuje vydavateľovi, napríklad mobilných vodičských preukazov, zahrnúť niektoré dátové prvky vo svojich mDL, ktoré sú definované a používané len na vnútroštátnej úrovni, ale nie na medzinárodnej.

Keďže EDIW pridal do svojej špecifikácie ako nepovinný formát aj JSON-LD a dátový model nie je definovaný, plánuje sa dátová transformácia realizovať v tomto prípade rovnakým spôsobom, ako pre ostatné údaje, ktoré sa ukladajú do MOU (pričom platia aj postupy uvedené v 5.4.1). Dátové modely elektronických dokladov ako občiansky preukaz[102], cestovný pas[103] a vodičský preukaz[104] vychádzajú z ontológie CMÚ pre fyzickú osobu, ktorá je založená na ontológiách od SEMIC pre interoperabilitu na úrovni EÚ[105], ako aj na ďalších kontrolovaných slovníkoch[106]. Ak bude overovateľ potrebovať overiť JSON súbor, dokáže na to využiť aj JSON LD, v ktorom sa v prípade potreby môže upraviť kontext, aby jeho využitie bolo jednoduchšie. Pre splnenie označovania menných priestorov podľa štandardu ISO/IEC 18013 sa JSON LD obohatí o túto informáciu, pričom bude na znalosti.gov.sk vedená ekvivalencia medzi URI daného dátového prvku v CMÚ a označením dátového prvku podľa ISO/IEC 18013 cez predikát štandardu OWL „owl:SameAs“.

Minimálny povinný rozsah osvedčených atribútov, ktoré bude potrebné umožniť overovať a používať aj pomocou EDIW, bude teda prístupných vyššie uvedeným postupom cez nasledujúce doklady a datasety, transformované do RDF štandardu v súlade s CMÚ:

- Občiansky preukaz[107]:
  - Address (adresa),
  - Age (vek),

---

[102] Zdroj: https://wiki.vicepremier.gov.sk/pages/viewpage.action?pageId=97518583, Dátum referencie: 30.05.2023

[103] Zdroj: https://wiki.vicepremier.gov.sk/pages/viewpage.action?pageId=97518587, Dátum referencie: 30.05.2023

[104] Zdroj: https://wiki.vicepremier.gov.sk/pages/viewpage.action?pageId=97518585, Dátum referencie: 30.05.2023

[105] Zdroj: https://joinup.ec.europa.eu/collection/semic-support-centre/solution/core-person-vocabulary/release/100, Dátum referencie: 30.05.2023

[106] Zdroj: https://op.europa.eu/en/web/eu-vocabularies/authority-tables, Dátum referencie: 30.05.2023

[107] Zdroj: https://wiki.vicepremier.gov.sk/pages/viewpage.action?pageId=97518583, Dátum referencie: 30.05.2023

  - Gender (pohlavie),
  - Číslo občianskeho preukazu,
- Cestovný pas[108]:
  - Číslo cestovného pasu,
- Dataset z Registra fyzických osôb (dataset o fyzickej osobe)[109]:
  - Rodné číslo, medzisystémový identifikátor,
  - Civil status (rodinný stav),
  - Family composition (zloženie rodiny),
- Pripravené pre OOTS (kapitola 5.4.3):
  - Educational qualifications, titles and licenses (vzdelanie, tituly a licencie),
  - Professional qualifications, titles and licenses (profesionálne kvalifikácie, tituly a licencie),
  - Public permits and licenses (verejné povolenia a licencie) – okrem vodičského preukazu[110], pripraveného podľa EDIW a ISO/IEC 18013,
  - Financial and company data (finančné a firemné údaje).

[108] Zdroj: https://wiki.vicepremier.gov.sk/pages/viewpage.action?pageId=97518587, Dátum referencie: 30.05.2023

[109] Zdroj: https://wiki.vicepremier.gov.sk/pages/viewpage.action?pageId=97517681, Dátum referencie: 30.05.2023

[110] Zdroj: https://wiki.vicepremier.gov.sk/pages/viewpage.action?pageId=97518585, Dátum referencie: 30.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

5.4.3 Poskytovanie mojich údajov cez OOTS

Ďalšou dôležitou témou, ktorú pokrýva štandard pre dátové transformácie, je iniciatíva Jednotnej digitálnej brány - SDG (Single Digital Gateway), ktorou sa majú vymieňať údaje v rámci princípu „jedenkrát a dosť" cez systém „Once and Only" (OOTS). Pôvodným zámerom na úrovni EÚ bolo vytvárať spoločné dátové modely pre OOTS, ktoré už aj boli urobené pre niektoré dôkazy, avšak žiaľ bola táto iniciatíva zastavená[111]. Súčasný prístup k SDG OOTS sa zameriava na opätovné použitie existujúcich dátových modelov, ak je to možné, čo teda zahŕňa aj možnosť využívania lokálnych dátových modelov, ktoré existujú na národnej úrovni. V rámci štandardu dátovej transformácie sa zvolila stratégia, že sa potrebné dôkazy pre OOTS definujú už v CMÚ, pričom základom je európska ontológia od SEMIC pre dôkaz („Evidence")[112]. Tým pádom sa pre dátovú transformáciu zdrojového datasetu opäť použijú metódy a nástroje uvedené v úvode kapitoly 5.4, pričom platia aj postupy uvedené v 5.4.1. Pre OOTS platí preferovaný formát na zdieľanie údajov medzi členskými štátmi XML (XSD), založený na definícii schémy XML (XSD) alebo ekvivalentnom formáte, doplnený o iné široko používané formáty serializácie, ak je to možné (viď. článok 7 impl. Nariadenia[113]) Ak bude tretia strana vyžadovať JSON alebo XML súbor, dá sa použiť JSON LD (fungujú naň štandardné knižnice pre parsovanie dokumentov JSON) a RDF/XML (funguje naň parsovanie cez XPath, alebo sa dá napríklad aj deserializovať do objektov v programovacom jazyku Java). Ak bude potrebné zdieľať súbor PDF alebo iný formát čitateľný človekom, využije sa na to priamo funkcionalita v klientovi MOU, ktorá umožňuje JSON LD údaje vizualizovať ako PDF a zdieľať tretím stranám. Alternatívou je vytvoriť aj pre vizualizáciu ďalší súbor pre transformáciu v XSLT, ako sa uvádza v kapitole 3.2.

[111] Zdroj: https://github.com/SEMICeu/SDG-sandbox, Dátum referencie: 30.05.2023

[112] Zdroj: https://wiki.vicepremier.gov.sk/display/IN/CCCEV-AP-SK+1.0, Dátum referencie: 30.05.2023

[113] Zdroj: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32022R1463&qid=1663658868787, Dátum referencie: 30.05.2023

Tabuľka 8: Mapovanie objektov evidencie vo vzťahu k Jednotnej digitálnej bráne - OOTS

Životné udalosti	Postupy	Očakávaný výstup na základe prípadného posúdenia žiadosti príslušným orgánom v súlade s vnútroštátnym právom	Možné objekty evidencie s odkazom na ontológiu (predpis v JSON LD)
Narodenie	Žiadosť o doklad o zápise narodenia	Doklad o zápise narodenia alebo rodný list	• Rodný list • Výpis z IS RFO – údaje o narodení
Pobyt	žiadosť o doklad o pobyte	Potvrdenie o registrácii na novej adrese	• Potvrdenie o pobyte (aktuálny stav) (možno vytvoriť z predpisu RFO) • Výpis z IS RFO – údaje o pobyte občana SR • Výpis z IS RFO – údaje o pobyte cudzinca
Štúdium	Žiadosť o financovanie terciárneho vzdelávania, ako sú študentské štipendiá a pôžičky od verejného subjektu alebo inštitúcie	Rozhodnutie o žiadosti o financovanie alebo potvrdenie o prijatí žiadosti	• Momentálne nemáme takto definovaný ani objekt evidencie ani predpis JSON-LD
	Podanie pôvodnej žiadosti o prijatie na verejnú terciárnu vzdelávaciu inštitúciu	Potvrdenie o prijatí žiadosti	• Potvrdenie o podaní prihlášky na VŠ
	Žiadosť o akademické uznanie diplomov, vysvedčení alebo iných dokladov o absolvovaní štúdia alebo kurzov	Rozhodnutie o žiadosti o uznanie	**Vysokoškolské vzdelávanie**: • Potvrdenie o kvalifikačnom stupni terciárneho vzdelania • Potvrdenie o výpise známok vysokoškolského vzdelávania • Potvrdenie o vysokoškolskom vzdelaní

Tento projekt je podporený z Európskeho sociálneho fondu.

Životné udalosti	Postupy	Očakávaný výstup na základe prípadného posúdenia žiadosti príslušným orgánom v súlade s vnútroštátnym právom	Možné objekty evidencie s odkazom na ontológiu (predpis v JSON LD)
			• Potvrdenie o zápise na vysokoškolské vzdelávanie • Rozhodnutie o žiadosti o uznanie akademického titulu **Stredoškolské vzdelávanie** • Potvrdenie o odbornom vzdelávaní, ktorý oprávňuje na prijatie na vysokoškolské vzdelanie • Potvrdenie o stredoškolskom vzdelaní • Potvrdenie o výpise známok stredoškolského vzdelávania • Potvrdenie o zápise na strednú školu • Potvrdenie o znalostiach cudzích jazykov stredoškolského vzdelávania **Ostatné potvrdenia**: • Potvrdenie o akreditácii vysokovýkonného športovca
Práca	Žiadosť o určenie uplatniteľných právnych predpisov v súlade s hlavou II nariadenia (EÚ) č. 883/ 2004 (1)	Rozhodnutie o uplatniteľných právnych predpisoch	• Momentálne nemáme takto definovaný ani objekt evidencie ani predpis JSON-LD

Tento projekt je podporený z Európskeho sociálneho fondu.

Životné udalosti	Postupy	Očakávaný výstup na základe prípadného posúdenia žiadosti príslušným orgánom v súlade s vnútroštátnym právom	Možné objekty evidencie s odkazom na ontológiu (predpis v JSON LD)
	Oznámenie o zmenách osobnej alebo pracovnej situácie osoby poberajúcej dávky sociálneho zabezpečenia, ktoré sú pre takéto dávky relevantné	Potvrdenie o prijatí oznámenia o takýchto zmenách	**Sociálne zabezpečenie:** • Existencia nedoplatku na poistnom na sociálne poistenie • Údaje o nemocenských dávkach Sociálnej poisťovne • Údaje o dôchodkových dávkach Sociálnej poisťovne • Údaje o dôchodkových dávkach zo Starobného dôchodkového sporenia (II.pilier) • Registrácie fyzickej osoby v Sociálnej poisťovni • Existencia zamestnanca Sociálnej poisťovni • Poberanie invalidného dôchodku
	Žiadosť o európsky preukaz zdravotného poistenia (EPZP)	Vydanie európskeho preukazu zdravotného poistenia (EPZP)	• Momentálne nemáme takto definovaný ani objekt evidencie ani predpis JSON-LD
	Podanie priznania k dani z príjmu	Potvrdenie o prijatí priznania k dani z príjmu	• Daňové priznanie FO - typ B • Daňové priznanie PO
Presťahovanie	Registrácia zmeny adresy	Potvrdenie o odhlásení z predchádzajúcej adresy a o registrácii novej adresy	• Potvrdenie o pobyte (aktuálny stav – neevidovaný na žiadnej adrese v SVK) • Výpis z IS RFO– údaje o pobyte občana SR

Tento projekt je podporený z Európskeho sociálneho fondu.

Životné udalosti	Postupy	Očakávaný výstup na základe prípadného posúdenia žiadosti príslušným orgánom v súlade s vnútroštátnym právom	Možné objekty evidencie s odkazom na ontológiu (predpis v JSON LD)
			• Výpis z IS RFO– údaje o pobyte cudzinca
	Registrácia motorového vozidla, ktoré pochádza z členského štátu alebo je už zaradené do evidencie v členskom štáte, prostredníctvom štandardných postupov (2)	Doklad o prihlásení motorového vozidla do evidencie	• Technický preukaz
	Získanie známok na používanie vnútroštátnej cestnej infraštruktúry: časových poplatkov (diaľničných známok), poplatkov za prejazdenú vzdialenosť (mýto), ktoré vydal verejný subjekt alebo inštitúcia	Príjmový doklad o zaplatení mýta alebo diaľničnej známky alebo iný platobný doklad	• Výpis z NBS registra pre diaľničné známky • Výpis z NBS registra pre mýto
	Získanie emisnej známky, ktorú vydal verejný subjekt alebo inštitúcia	Príjmový doklad o zaplatení emisnej známky alebo iný platobný doklad	• Preukaz o emisnej kontrole
Odchod do dôchodku	Uplatnenie nároku na dôchodkové a preddôchodkové dávky z povinných systémov	Potvrdenie o prijatí uplatnenia nároku alebo rozhodnutie o nároku na dôchodkové alebo preddôchodkové dávky	**Sociálne zabezpečenie:** • Údaje o dôchodkových dávkach Sociálnej poisťovne • Údaje o dôchodkových dávkach zo Starobného dôchodkového sporenia (II.pilier) • Poberanie invalidného dôchodku

Tento projekt je podporený z Európskeho sociálneho fondu.

Životné udalosti	Postupy	Očakávaný výstup na základe prípadného posúdenia žiadosti príslušným orgánom v súlade s vnútroštátnym právom	Možné objekty evidencie s odkazom na ontológiu (predpis v JSON LD)
	Žiadosti o informácie o údajoch týkajúcich sa dôchodku z povinných systémov	Vyhlásenie o údajoch o osobnom dôchodku	**Sociálne zabezpečenie:** • Údaje o dôchodkových dávkach Sociálnej poisťovne • Údaje o dôchodkových dávkach zo Starobného dôchodkového sporenia (II.pilier) • Poberanie invalidného dôchodku
Začatie podnikateľskej činnosti, priebeh podnikateľskej činnosti a ukončenie podnikateľskej činnosti	Oznámenie o podnikateľskej činnosti, povolenia na výkon podnikateľskej činnosti, zmeny podnikateľskej činnosti a ukončenie podnikateľskej činnosti, ktoré sa netýka postupov v prípade platobnej neschopnosti alebo likvidácie, s výnimkou pôvodnej registrácie podnikateľskej činnosti v obchodnom registri a s výnimkou postupov týkajúcich sa založenia alebo akéhokoľvek následného podania zo strany spoločností alebo firiem v zmysle článku 54, druhého pododseku ZFEÚ	Potvrdenie o prijatí oznámenia alebo zmeny alebo žiadosti o povolenie podnikateľskej činnosti	• RPO (odpis)
	Registrácia zamestnávateľa (fyzickej osoby) v povinnom dôchodkovom systéme alebo systémoch poistenia	Potvrdenie o registrácii alebo identifikačné číslo sociálneho zabezpečenia	**Sociálne zabezpečenie:** • Registrácie fyzickej osoby v Sociálnej poisťovni • Existencia zamestnanca Sociálnej poisťovni

Tento projekt je podporený z Európskeho sociálneho fondu.

Životné udalosti	Postupy	Očakávaný výstup na základe prípadného posúdenia žiadosti príslušným orgánom v súlade s vnútroštátnym právom	Možné objekty evidencie s odkazom na ontológiu (predpis v JSON LD)
	Registrácia zamestnancov v povinnom dôchodkovom a poistnom systéme	Potvrdenie o registrácii alebo identifikačné číslo sociálneho zabezpečenia	**Sociálne zabezpečenie:** • Registrácie fyzickej osoby v Sociálnej poisťovni • Existencia zamestnanca Sociálnej poisťovni
	Podanie priznania k dani z príjmu právnických osôb	Potvrdenie o podaní priznania k dani z príjmu právnických osôb	• Daňové priznania PO
	Oznámenie sociálnej poisťovni o ukončení zmluvy so zamestnancom s výnimkou postupov hromadného ukončenia pracovných zmlúv	Potvrdenie o doručení oznámenia	• Existencia zamestnanca SP
	Úhrada príspevkov na sociálne zabezpečenie zamestnancov	Príjmový doklad alebo iná forma potvrdenia o úhrade príspevkov na sociálne zabezpečenie zamestnancov	**Sociálne zabezpečenie:** • Existencia nedoplatku na poistnom na sociálne poistenie

Tento projekt je podporený z Európskeho sociálneho fondu.

6 Návrh odporúčaní na aplikáciu štandardu

V tejto kapitole sú zosumarizované navrhované odporúčania na aplikáciu štandardu pre informačné prostredie verejnej správy, pričom sa zameriame aj na špecifické projekty z programu Manažment údajov.

Pri dátovej integrácii a transformácii odporúčame intenzívnejšie využívať nasadené riešenie Talend (kapitola 4.2.1.1). Pri modernizácií informačných systémov poskytovateľov a konzumentov, ako aj pri potrebe aplikačnej integrácie či vysoko flexibilného a rýchleho streamovania udalostí odporúčame využiť Apache Camel a/alebo Apache Kafka (kapitola 2.2).

Samotnú dátovú transformáciu odporúčame implementovať v nasledujúcich piatich fázach, pre ktoré budú ďalej rozpracované konkrétne odporúčania:

1. **Preskúmanie údajov**: Identifikácia a interpretácia pôvodného formátu údajov je prvým krokom. Táto fáza pomáha určiť, čo je potrebné urobiť s údajmi, aby sa transformovali do požadovaného stavu.
2. **Mapovanie údajov**: Počas tejto fázy sa plánuje samotný proces dátovej transformácie – ide o metódu, ktorá porovnáva alebo prepája dátové prvky objektov evidencie alebo datasetov z jedného zdroja do druhého.
3. **Extrakcia údajov**: Počas tejto fázy sa údaje presúvajú zo zdrojového systému do cieľového systému. Extrakcia môže zahŕňať štruktúrované (databázy) alebo neštruktúrované (toky udalostí, logy) zdroje.
4. **Nastavenie nástrojov na transformáciu a/alebo generovanie kódu**: Aby bol proces dátovej transformácie úspešný, je nevyhnutné mať k dispozícii kód na vykonanie transformačnej úlohy.
5. **Používanie nástrojov na transformáciu a/alebo vykonávanie kódu**: Vykonanie kódu alebo spustenie skriptov v nástroji uvedie do pohybu vopred naplánovaný a nakódovaný proces dátovej transformácie a transformuje vstupné údaje na požadovaný výstup.
6. **Validácia dátovej transformácie**: V tejto fáze sa preskúmajú transformované údaje, či spĺňajú všetky očakávania a kritériá kvality.

6.1 Preskúmanie údajov

V tejto fáze si potrebujeme zadefinovať, čo je potrebné urobiť s údajmi, aby sa transformovali do požadovaného stavu pre daný prípad použitia, definovaný v kapitole 5. Zvyčajne sa na to používa nástroj na profilovanie údajov. Profilovanie údajov sa vzťahuje na proces skúmania, analýzy, validácie a sumarizácie datasetov s cieľom získať prehľad o kvalite údajov. V rámci profilovania zisťujú analytické algoritmy charakteristiky datasetov, ako je priemer, minimum, maximum, percentil a frekvencia, aby bolo možné preskúmať údaje do najmenších detailov. Následne algoritmy vykonávajú analýzy na odhalenie metadát vrátane rozdelenia frekvencie výskytu, kľúčových vzťahov, kandidátov na cudzie kľúče a funkčných závislostí. Nakoniec sa všetky tieto informácie použijú na odhalenie toho, ako sú tieto faktory v súlade so štandardmi a cieľmi dátového programu a dátovej kvality. Profilovanie údajov tak dokáže eliminovať nákladné chyby, ktoré sú bežné v databázach. Medzi tieto chyby patria nulové

Tento projekt je podporený z Európskeho sociálneho fondu.

hodnoty (neznáme alebo chýbajúce hodnoty), hodnoty, ktoré by sa nemali ukladať, hodnoty s neobvykle vysokou alebo nízkou frekvenciou výskytu, hodnoty, ktoré sa neriadia očakávanými vzormi, a hodnoty mimo bežného rozsahu. Takéto automatizované profilovanie poskytuje aj nasadený nástroj Talend a v čase písania tohto dokumentu sa implementuje v rámci projektu Dátovej integrácie – časť Dátová kvalita. V rámci tohto projektu sa robí aj ďalšia častá aktivita, ktorá je kľúčová pre profilovanie a dátovú kvalitu, a tou je definovanie takzvaných regulárnych výrazov („regular expressions (REGEX)“) pre vybrané dátové prvky[114] - tieto informácie sú aj zaevidované priamo v dátovom slovníku nástroja Talend, ktorý sa využíva aj na spúšťanie kampaní pre zvyšovanie kvality údajov[115]. Následne možno v spomínanom nástroji Talend získať profil datasetu, ktorý analyzuje percento údajov, ktoré spĺňajú daný vzor (v tomto prípade REGEX pre e-mailovú adresu), ako aj ďalšie informácie, ako napríklad koľko položiek v datasete má nevyplnený údaj o e-mailovej adrese a koľko má duplicitný údaj (Obrázok 12). Toto profilovanie údajov sa dá realizovať aj v otvorenej verzii („open source“) nástroja Talend – „Talend Open Studio for Data Quality“, ktorý si možno stiahnuť na tejto linke[116]. Pre pokročilé profilovanie možno do toho nástroja inštalovať aj rôzne pluginy[117]. Ďalším odporúčaným „open-source“nástrojom pre profilovanie je Apache Griffin[118].

[114] Zdroj: https://wiki.vicepremier.gov.sk/pages/viewpage.action?pageId=101833534, Dátum referencie: 09.05.2023

[115] Zdroj: https://help.talend.com/r/en-US/Cloud/data-stewardship-user-guide/managing-campaigns, Dátum referencie: 09.05.2023

[116] Zdroj: https://help.talend.com/r/en-US/8.0/studio-getting-started-guide-open-studio-for-data-quality/introduction, Dátum referencie: 09.05.2023

[117] Napríklad: https://sourceforge.net/projects/talendprofiler/, Dátum referencie: 09.05.2023

[118] Zdroj: https://griffin.apache.org/docs/profiling.html, Dátum referencie: 09.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Obrázok 12: Dobrá prax profilovania datasetov v nástroji Talend[119]

Konkrétne výsledky porovnávania vzorov ukazujú (Obrázok 12), že približne 10 % e-mailových záznamov nezodpovedá štandardnému REGEX pre e-mailovú adresu. Výsledky jednoduchej štatistiky ukazujú, že približne 8 % e-mailových záznamov je prázdnych a že približne 5 % je duplicitných. Výsledky frekvencie vzorov poskytujú počet najčastejších záznamov pre každý odlišný vzor. Z toho vyplýva, že údaje nie sú konzistentné a pred spustením dátovej transformácie alebo v rámci nej je potrebné opraviť a vyčistiť údaje o e-mailových adresách.

[119] Zdroj: https://help.talend.com/r/en-US/8.0/data-quality-job-examples/executing-analysis-and-displaying-profiling-results, Dátum referencie: 09.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Dátoví špecialisti používajú techniky profilovania údajov alebo skripty na zachytenie štruktúry a vlastností údajov a určenie, ako by sa mali upraviť. Teda, čo z nasledujúceho zoznamu je potrebné pre daný objekt evidencie alebo dataset spraviť:

1. Zmeniť formát,
2. Zmeniť štruktúru,
3. Zmeniť dátový model,
4. Aplikovať iný štandard,
5. Zvýšiť kvalitu údajov bez zmeny obsahu údajov, napríklad odstránením chýbajúcich hodnôt, určením („casting“) a konverziou dátových typov pre kompatibilitu, úpravou dátumov a časov pomocou posunov a lokalizácie formátu, premenovaním schém, tabuliek a stĺpcov pre prehľadnosť, či vylepšením formátu danej hodnoty dátového prvku alebo obohatením údajov.

Tu je dôležité poznamenať, že v rámci dátovej transformácie by sa mali robiť len tie úkony na zvýšenie dátovej kvality, ktoré nie je možné spraviť v rámci služieb implementovaných v projekte Dátovej integrácie, časti Dátová kvalita pre nasadený nástroj Talend.

6.2 Mapovanie údajov

Mapovanie údajov („data mapping“) zahŕňa extrakciu dátových prvkov z viacerých zdrojov a ich následné priradenie k cieľovým dátovým prvkom u konzumenta – v cieľovom informačnom systéme.

Mapovanie údajov pomáha konsolidovať údaje ich extrahovaním, transformáciou a načítaním do nového dátového súboru, tabuľky, schémy atď. Vďaka nemu môže zdieľanie a migrácia údajov fungovať plynulejšie, pretože umožňuje kombinovať údaje z rôznych zdrojov. Tieto mapované údaje sa potom dajú využiť na získanie relevantných poznatkov, ktoré môžu zvýšiť efektivitu procesov a rozhodovania. Údaje zozbierané z rôznych zdrojov sa však musia transformovať do formátu, štruktúry a dátového modelu, prípadne štandardu, ktoré sú vhodné pre prevádzkové a analytické potreby organizácií. To sa vykonáva prostredníctvom **modelovania údajov** (tomu sa venuje predovšetkým dokument 1.1.2 Štandardizácia pre modelovanie údajov a čiastočne dokument 4.2.1 Koncept pre zavádzanie analytického spracovania údajov do praxe), ktoré je neoddeliteľným krokom v rôznych procesoch manažmentu údajov. Modelovanie údajov a následná dátová transformácia údajov do podoby použiteľnej na analytické účely je pre každú organizáciu kľúčová, pretože inak prichádza o cenné poznatky obsiahnuté v dátach.

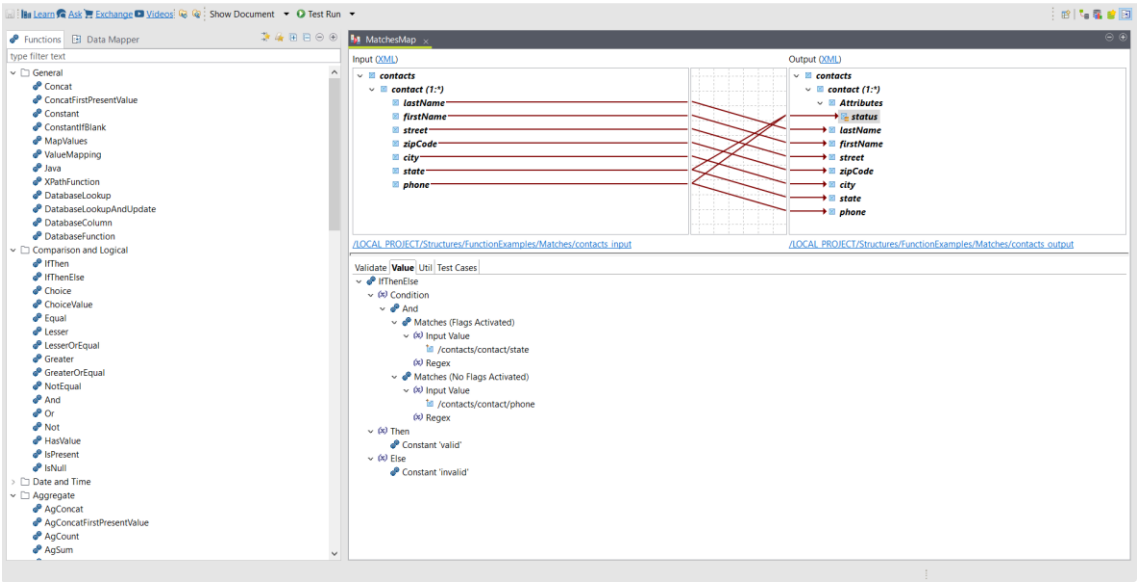
Pre úspešnú dátovú integráciu musia mať zdrojové a cieľové údaje podobné dátové modely. Je však zriedkavosťou, aby dve dátové úložiská mali rovnakú schému. Tu prichádza na rad mapovanie údajov, ktoré ukáže, kde sú nezrovnalosti medzi dvoma datasetmi. Pri tomto procese môžu pomôcť rôzne nástroje na mapovanie údajov. Konkrétne môžu pomôcť preklenúť rozdiely v schémach zdrojov údajov a cieľov, čo organizáciám umožňuje rýchlo konsolidovať informácie z rôznych zdrojov. Výber správneho nástroja na mapovanie údajov pre organizáciu má zásadný význam pre každý projekt integrácie údajov, transformácie údajov a dátových skladov. Existuje niekoľko kľúčových vlastností, ktoré musí mať riešenie na mapovanie údajov:

Tento projekt je podporený z Európskeho sociálneho fondu.

1. Užitočný nástroj mapovania údajov dokáže spracovať širokú škálu zdrojových vstupov a pripojiť sa k rôznym štruktúrovaným zdrojom údajov vrátane databáz a rozhraní REST API a formátov „plochých“ súborov, ako sú XML, CSV a textové súbory.
2. Vhodný nástroj ponúka spôsoby vytvárania dátových máp bez potreby vytvorenia zdrojového kódu alebo s nutnosťou vytvoriť len minimalistický zdrojový kód („code-free“ a „no-code“ nástroje) a mal by spracovávať údaje pomocou zabudovaných transformácií. Jednoduchosť používania je dôležitou vlastnosťou v závislosti od toho, kto bude nástroj používať. Biznis používatelia a analytici sú pravdepodobne viac zvyknutí na grafické používateľské rozhranie (GUI) bez použitia zdrojového kódu, zatiaľ čo dátoví vedci a inžinieri sú pravdepodobne viac zvyknutí na skriptovanie alebo rozšíriteľnosť pomocou zdrojového kódu.
3. Nástroj na mapovanie údajov musí byť schopný plánovať mapovanie údajov a automatizovať tento proces. Takéto nástroje obsahujú šablóny na mapovanie údajov, ktoré umožňujú extrahovať požadované údaje z neštruktúrovaných reportov. Tieto nástroje dokážu automatizovať procesy modelovania údajov a dátové transformácie, a tým rýchlejšie poskytovať údaje pripravené na zdieľanie a analýzu a zároveň šetriť čas a zdroje dátových tímov.
4. Možnosť okamžitého zobrazenia údajov kdekoľvek počas procesu na overenie, či sa všetky procedúry dejú podľa návrhu.
5. Riešenie nezrovnalostí v názvoch polí pomocou synoným a funkcie „data lineage“ na riešenie problémov súvisiacich s konfliktami názvov. Používatelia môžu vytvoriť synonymá pre rôzne polia, čo môže ďalej automatizovať a urýchliť proces mapovania a dátovej transformácie.

Aj nasadený nástroj Talend poskytuje takéto mapovanie údajov, ako ukazuje Obrázok 13.

Tento projekt je podporený z Európskeho sociálneho fondu.



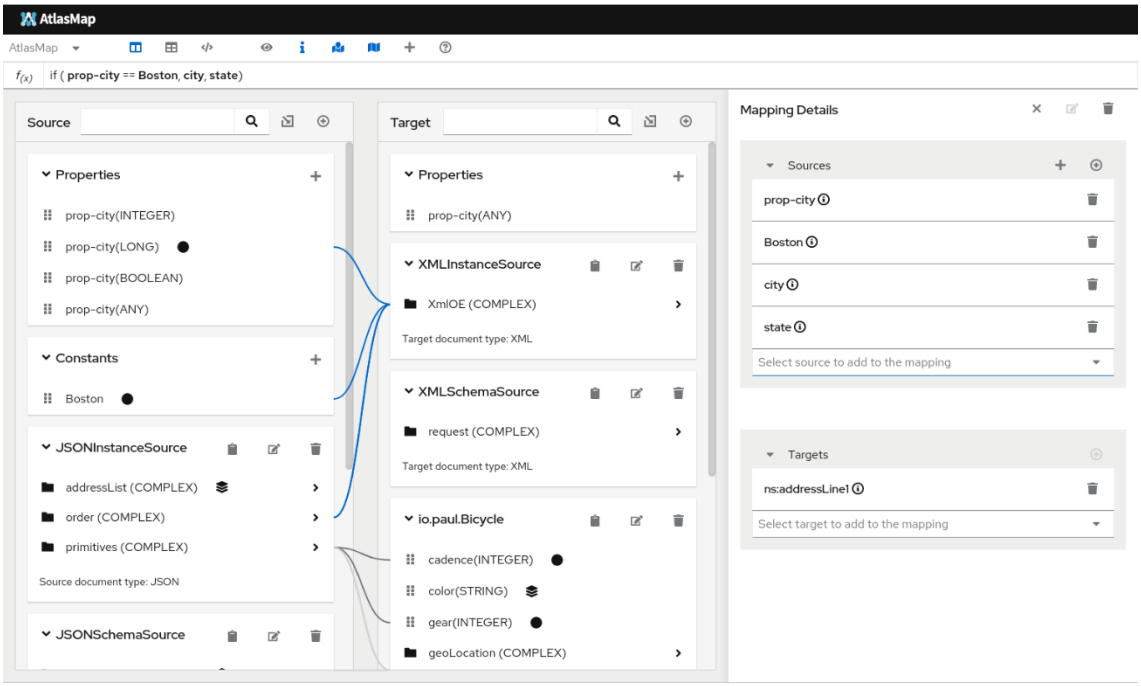
Obrázok 13: Nástroj na mapovanie údajov v platforme Talend[120]

Alternatívne možno použiť vizuálny dizajnér pre plánovanie transformácie v rámci Apache Camel[121], v rámci ktorého je k dispozícii AtlasMap (Obrázok 14). AtlasMap je riešenie na mapovanie údajov s interaktívnym webovým používateľským rozhraním, ktoré zjednodušuje konfiguráciu integrácie medzi zdrojmi údajov Java, XML, CSV a JSON. Mapovanie údajov možno navrhnúť pomocou takzvaného „canvas“ používateľského rozhrania „AtlasMap Data Mapper“ a potom toto mapovanie údajov spustiť prostredníctvom runtime enginu.

[120] Zdroj: https://help.talend.com/r/en-US/7.3/data-mapper-user-guide/what-is-talend-data-mapper, Dátum refencie: 09.05.2023

[121] Zdroj: https://github.com/designer-for-camel/camel-designer, Dátum refencie: 11.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.



Obrázok 14: Nástroj AtlasMap[122] na mapovanie údajov v rámci Apache Camel Visual Designer

Keďže podnikové údaje sa nachádzajú na rôznych miestach a v rôznych formátoch, dátová transformácia je nevyhnutná na prelomenie dátových síl a získanie poznatkov. Po vykonaní mapovania údajov dokážeme pochopiť, čo potrebujeme na výstupe dátovej transformácie, aby údaje správne zodpovedali mape. V ideálnom prípade nemusí byť dátovú transformáciu ani potrebné vykonať. V tejto fáze je teda potrebné naplánovať samotný proces dátovej transformácie. V prvom kroku si zosumarizujeme a zvalidujeme jeho ambíciu na základe požiadaviek konzumenta (cieľového informačného systému) a výstupov mapovania údajov (napríklad cez nástroj v platforme Talend[123]) – či ideme len zmeniť formát alebo aj štruktúru. To nie je vždy jednoduchý proces, ale existujú transformačné nástroje, ktoré môžu pomôcť prečítať mapu údajov a upraviť údaje tak, aby zodpovedali tejto mape bez straty integrity údajov.

Takže teda na základe vyhodnotenia mapovania údajov môže ísť o **jednoduchú transformáciu** (položky v zozname 1 a 2 v kapitole 6.1), alebo o **komplexnú transformáciu** (položky v zozname 3, 4 a 5 v kapitole 6.1). Pri zmene dátového modelu ide spravidla o zosúladenie údajov zo zdroja s Centrálnym modelom údajov, ktorý je podrobne popísaný v dokumente 1.1.2 Štandardizácia pre modelovanie údajov. S tým súvisí aj aplikovanie iného štandardu, ak sú zdrojové údaje napríklad vo formáte XML alebo JSON, je potrebné ich previesť do štandardu RDF, obohatiť o ďalšie informácie z ontológií Centrálneho dátového modelu, a následne serializovať napríklad do formátu JSON-LD. Tento postup má však význam len vtedy, ak konzument dokáže pracovať so

[122] Zdroj: https://www.atlasmap.io/, Dátum referencie: 11.05.2023

[123] Zdroj: https://help.talend.com/r/en-US/8.0/data-catalog-user-guide/working-with-data-mappings, Dátum referencie: 09.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

štandardom RDF pre „Linked Data“. Podrobnejšie sa tomuto postupu venujeme v prípade použitia pre Manažment osobných údajov (MOU) v kapitole 5.4.

6.3 Extrakcia údajov

V tejto fáze ide o získanie údajov priamo zo zdroja, a to jeho dátovou integráciou cez Centrálnu integračnú platformu, alebo napojením sa na dedikované API. Alternatívou je napojenie nástroja na transformáciu priamo na zdroj.

V prvom kroku je potrebné získať pseudonymizované testovacie údaje (ak sú originálne údaje osobné alebo iným spôsobom citlivé), aby bolo možné nastaviť ciele transformácie a vytvoriť očakávaný výstup, ako aj neskôr otestovať implementované procedúry transformácie. Následne po overení implementácie sa napoja zdrojové údaje.

6.4 Nastavenie nástrojov a/alebo generovanie kódu

Na tvorbu kódov pre transformačné úlohy v rámci dátovej transformácie sa zvyčajne používa nástroj alebo platforma na transformáciu údajov (odporúčané nástroje sú definované v kapitole 4). V tejto fáze procesu sa počítačový kód potrebný na dátovú transformáciu vytvára prostredníctvom technológií na dátovú transformáciu alebo dátovými špecialistami, ktorí vyvíjajú skripty. V princípe pre jednoduchú transformáciu, definovanú v kapitole 6.2, nám majú postačovať vybrané a nasadené nástroje pre dátovú transformáciu. Pre komplexnú transformáciu, tiež definovanú v kapitole 6.2, je spravidla nevyhnutný zdrojový kód, ktorý môže vybraný nástroj vygenerovať, a následne ho musí programátor upraviť, alebo ho programátor vytvára od začiatku s využitím existujúcich knižníc.

6.5 Používanie nástrojov a/alebo vykonávanie kódu

V tejto fáze sa údaje získavajú zo zdroja(-ov), ktorými môžu byť štruktúrované, streamovacie, a telemetrické datasety alebo logy. Údaje sa potom transformujú metódami, popísanými v kapitole 3, ako je filtrovanie, konverzia formátu alebo zlučovanie, ako sa plánovalo vo fáze mapovania (kapitola 6.2). V tomto bode sa údaje skutočne menia.

6.6 Validácia dátovej transformácie

V tejto fáze sa preskúma formát a štruktúra transformovaných údajov, aby sa zabezpečila ich presnosť. Počas tejto fázy dátoví špecialisti alebo koncoví používatelia - konzumenti kontrolujú, či tok údajov spĺňa vopred stanovené kritériá transformácie, a ak nie, riešia a opravujú prípadné anomálie alebo chyby. Takáto kontrola býva aj súčasťou platforiem („Test Bed“) na testovanie IT systémov, či sú správne implementované podľa špecifikácie na interoperabilitu – ide o proces testovania zhody („conformance testing“). Výmena údajov medzi rôznymi IT systémami je možná len vďaka zabezpečeniu interoperability IT systémov, čo znamená, že systémy komunikujú spoločným spôsobom a majú spoločnú predstavu o vymieňaných správach a údajoch v nich a ich spracovaní. Dátová transformácia je teda zásadná pre zabezpečenie interoperability IT systémov na úrovni údajov. Takýto Test Bed pre testovanie zhody na

Tento projekt je podporený z Európskeho sociálneho fondu.

podporu interoperability IT systémov vytvorila aj Európska komisia[124] a možno ho využiť a prispôsobiť jednotlivým prípadom použitia.

Pre údaje transformované do štandardu RDF sa na validáciu používa štandard SHACL, popísaný v dokumente 1.1.2 Štandardizácia pre modelovanie údajov. Testovanie zhody transformovaných údajov so špecifikáciou na interoperabilitu je súčasťou vývoja a nasadenia modulu dátovej transformácie. Ak začne cez dátovú transformáciu prúdiť veľké množstvo údajov, je potrebné zabezpečiť, aby ich kvalita bola prvoradá a aby nedegradovala ani v prípade zmeny zdrojových kódov alebo nastavení nástrojov v celom procese výmeny údajov. To sa dá dosiahnuť zavedením takzvanej „pozorovateľnosti údajov“ („data observability“). Príkladom nástrojov, ktoré zabezpečujú okrem iného pozorovateľnosť údajov, sú Datafold alebo integrate.io[125], ktoré ale možno nasadiť len do moderného dátového stacku, aký sa odporúča pre ďalší rozvoj Konsolidovanej analytickej vrstvy v rámci dokumentu 4.2.1 Koncept pre zavádzanie analytického spracovania údajov do praxe.

[124] Zdroj: https://joinup.ec.europa.eu/collection/interoperability-test-bed-repository/solution/interoperability-test-bed/about, Dátum referencie: 15.05.2023

[125] Zdroj: https://www.integrate.io/blog/top-data-observability-tools/ , Dátum referencie: 15.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

7 Príklady dobrej praxe aplikácie štandardu

Štandard dátovej transformácie nemožno aplikovať v izolácií – vždy je súčasťou komplexnejšieho programu alebo štandardu pre zabezpečenie interoperability IT systémov alebo zdieľaných údajov, a o tom sú aj nasledovné príklady dobrej praxe. Cieľom takéhoto širšieho programu alebo skupiny štandardov je zabezpečiť, aby údaje do systémov vstupovali len z jedného zdroja („once and only principle“) a/alebo aby údaje boli vhodné na opakované použitie (termín používaný pre Open Data) či na sekundárne účely (termín používaný pre analytické spracovanie údajov).

7.1 Rámec riadenia pre zdieľanie údajov vo Veľkej Británii

Prečo je potrebný takýto rámec pre zdieľanie údajov

Údaje sú nevyhnutné pre poskytovanie kvalitných verejných služieb, tvorbu a hodnotenie politík a širokú škálu plnenia agendy verejnej správy. Údaje, ktoré sú na to potrebné, sa často nachádzajú v rôznych častiach verejnej správy, a je potrebné ich zdieľať, aby sa mohli používať. Britská vláda sa zaviazala riešiť problémy, ktoré znižujú schopnosť čo najlepšie využívať a zhodnocovať svoje údaje.

Rovnako ako sú jasné a spoločné technické štandardy základom toho, ako sú údaje reprezentované, zaznamenávané, opisované, uchovávané, transformované, zdieľané a sprístupňované, sú potrebné aj jasné a spoločné štandardy riadenia, aby sa údaje dostali na správne miesto v správnom čase a v správnej forme. Práca v silách, rôzne úrovne vyspelosti údajov a riešenie problémov izolovane prispeli k nejednotnosti a nesprávnemu nastaveniu riadenia zdieľania údajov. Tento rámec riadenia je pre Britániu príležitosťou na zhodnotenie a zmenu smerovania. Tento rámec poskytuje zásady a opatrenia na zníženie alebo odstránenie bežných netechnických prekážok zdieľania údajov. Zdôrazňuje vzťah medzi technickými dátovými štandardmi (vrátane štandardov formátov údajov, metadát a transformácie) a širším riadením údajov.

Uplatňovanie zásad a opatrení uvedených v tomto rámci má nasmerovať ministerstvá a verejné orgány k väčšiemu zosúladeniu systémov a procesov zdieľania údajov a zakotviť zdieľanie údajov ako strategickú prioritu v celej verejnej správe.

Päť princípov pre efektívne zdieľanie údajov

Existuje päť princípov, ktoré sa odporúča dodržiavať, aby sa v danej organizácii mohlo zefektívniť zdieľanie údajov. Každý princíp sa zaoberá príčinou konkrétnej prekážky alebo trecieho bodu v procese zdieľania údajov. Každý princíp je podložený opatreniami, ktoré podporujú väčšie zosúladenie medzi jednotlivými orgánmi verejnej správy v oblasti riadenia zdieľania údajov. Ide o nasledovné princípy:

1. Zaviazať sa k vedeniu a zodpovednosti za zdieľanie údajov,
2. Uľahčiť začatie zdieľania údajov,
3. Maximalizovať hodnotu údajov, ktoré sú k dispozícii,
4. Podporovať zodpovedné zdieľanie údajov,

Tento projekt je podporený z Európskeho sociálneho fondu.

5. Zabezpečiť, aby zdieľané údaje boli nájditeľné, prístupné, interoperabilné a opakovane použiteľné (ide o takzvané „FAIR“ údaje[126])

Stratégia orgánu pre štandardizáciu údajov na roky 2020 až 2023[127]

Na štandardoch pre údaje záleží. Štandardizácia spôsobu, akým štátna správa používa údaje, vedie k:

- lepšie verejné služby, rozhodovanie a politiky založené na dôkazoch,
- zníženie ťažkostí pri vyhľadávaní, chápaní, zdieľaní a používaní údajov, napríklad pri riadení prepojení medzi datasetmi a spájaní datasetov s cieľom vytvoriť nové datasety,
- zlepšenie interoperability, aby boli údaje porovnateľnejšie a ľahšie sa vymieňali
- lepšia kvalita, prístup a konzistentnosť údajov
- rýchla a lacná spolupráca medzi zamestnancami vo verejnej správe, ako aj medzi verejnou správou a občanmi a podnikateľmi,
- zvýšenie miery prijatia a implementácie nových riešení, aby sa zamestnanci mohli sústrediť na úlohy s vysokou hodnotou namiesto administratívnej práce, ako je formátovanie údajov, čistenie údajov alebo zosúlaďovanie údajov,
- dodržiavanie zákonov o ochrane údajov,
- zníženie technických prekážok s cieľom zlepšiť opätovné používanie administratívnych údajov a znížiť potrebu nákladného zhromažďovania údajov,
- zlepšenie rýchlosti a účinnosti riadenia zmien v existujúcich dátových štandardoch a usmerneniach, ktoré ich podporujú.

Na realizáciu výhod dátových štandardov musí vláda:

- zlepšiť koordináciu práce v oblasti vládnych dátových štandardov spojením existujúcich príkladov osvedčených postupov s cieľom zdieľať iniciatívy a poznatky, ako aj znížiť duplicitu práce,
- centralizovať stanovenie dátových štandardov a zvýšiť ich prijatie s cieľom prelomiť rezortné silá a pomôcť rezortom objaviť správne páky na zabezpečenie širšieho prijatia štandardov,
- vypracovať a realizovať koherentnú stratégiu na riešenie potrieb používateľov dátových štandardov naprieč celým verejným sektorom, pričom by sa rezorty povzbudili k tomu, aby skôr plánovali ako reagovali a koordinovali priority.

V záujme zlepšenia interoperability údajov sa stratégia zameriava na stanovenie štandardov, ktoré sa týkajú:

- metadát,

---

[126] Zdroj: https://www.go-fair.org/fair-principles/, Dátum referencie: 15.05.2023

[127] Zdroj: https://www.gov.uk/guidance/data-standards-authority-strategy-2020-to-2023, Dátum referencie: 15.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

- osobných a podnikových identifikátorov,
- referenčných údajov,
- geopriestorových údajov.

Na dohľad nad kvalitou údajov sa zriadil Data Quality Hub[128].

7.2 Interoperabilita údajov a informačných systémov na úrovni Európskej únie

Interoperabilná Európa[129] je iniciatíva Európskej komisie na posilnenie politiky interoperability verejného sektora. Navrhuje mechanizmus strategickej spolupráce v oblasti interoperability v celej Európskej únii. Občania a podniky v celej EÚ očakávajú od verejnej správy vysokú kvalitu verejných služieb. Poskytovanie verejných služieb zriedka funguje izolovane. Pri poskytovaní služieb si údaje často musia vymieňať rôzne orgány verejnej moci. Táto výmena sa uskutočňuje predovšetkým na miestnej, ale aj na regionálnej, vnútroštátnej a európskej úrovni a vyžaduje si interoperabilitu údajov. Keďže nemožno očakávať, že všetky systémy budú pracovať s rovnakým dátovým modelom, formátom a štandardom údajov, je na dosiahnutie interoperability nevyhnutá dátová transformácia na základe vopred dohodnutých pravidiel a kritérií.

Napriek úsiliu v posledných desaťročiach pretrvávajú v oblasti interoperability a digitalizácie verejných služieb v Európe určité prekážky, ktoré sú právneho, organizačného, sémantického a technického charakteru.

Podporné centrum SEMIC je prvým krokom na ceste k interoperabilnej európskej správe. V súlade s Aktom o interoperabilnej Európe sa komunita SEMIC zasadzuje za spoluvytváranie a výmenu osvedčených postupov, nástrojov a štandardov na podporu sémantickej interoperability. SEMIC a jeho podporné centrum ponúka niekoľko služieb, ktorých cieľom je pomôcť orgánom verejnej správy navzájom si vymieňať skúsenosti a pomáhať si pri realizácii jednotného trhu s údajmi pre zdieľanie a opätovné používanie údajov:

- **Špecifikácie** sú k dispozícii v otvorených a opakovane použiteľných formátoch a boli postupne vyvíjané s ohľadom na potreby a spätnú väzbu celej komunity sémantickej interoperability. Základné slovníky sú základným prvkom sémantickej interoperability medzi orgánmi verejnej správy, pretože poskytujú štandardizovaný prístup na opis kľúčových pojmov, ako sú miesta, podniky, organizácie a fyzické osoby[130]. Vychádza z nich aj Centrálny model údajov.
- **Piloty:** SEMIC podporuje zavádzanie technológií v počiatočnej fáze riešenia súčasných výziev, ako je napríklad vytváranie verzií zložitých a rýchlo sa meniacich súborov údajov, implementácia sprostredkovateľov údajov na správu osobných údajov a znalostných grafov s otvoreným zdrojovým kódom. Odborné poznatky a

[128] Zdroj: https://www.gov.uk/government/organisations/government-data-quality-hub, Dátum referencie: 30.05.2023

[129] Zdroj: https://joinup.ec.europa.eu/interoperable-europe, Dátum referencie: 30.05.2023

[130] Zdroj: https://joinup.ec.europa.eu/collection/semic-support-centre/specifications, Dátum referencie: 30.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

znalosti SEMIC sú k dispozícii orgánom verejnej správy a odborníkom na údaje s cieľom podporiť prechod na rozšírené zdieľanie údajov v prospech občanov a podnikov.

- **Toolkit** umožňuje orgánom verejnej správy začať svoju cestu k čoraz väčšej interoperabilite v EÚ. Každý nástroj v súbore SEMIC reaguje na požiadavky na zvýšenie úrovne interoperability. Nástroje sú predstavené pojmami súvisiacimi so sémantickou interoperabilitou, a teda aj tým, ako z nich urobiť každodennú prax v organizácii.
- **Centrum znalostí** poskytuje zbierku zdrojov na riešenie výziev v oblasti interoperability, od správ, štúdií a usmernení až po zasadnutia na výmenu poznatkov a záznamy z webových seminárov.

Spomínaný Test Bed (kapitola 6.6) implementovaný a poskytovaný Európskou komisiou slúži na podporu a testovanie širšej interoperability informačných systémov, nielen sémantickej. Pre podporu sémantickej interoperability poskytuje aj RDF validátor[131] - webovú aplikáciu na overovanie obsahu RDF podľa tvarov SHACL. Validácia je možná prostredníctvom niekoľkých kanálov:

- Webové používateľské rozhranie na manuálne použitie používateľmi,
- REST API na integráciu medzi strojmi,
- SOAP API na integráciu medzi strojmi a použitie v testovacích prípadoch zhody GITB Test Description Language[132].

7.3 Infraštruktúra pre dátovú platformu v Tesle založená na Kafke[133]

Čo je na príbehu spoločnosti Tesla zaujímavé z pohľadu dátovej transformácie

Spracovanie obrovských objemov údajov v reálnom čase je jednou zo zásadných schopností Kafky. Tesla nie je len výrobca automobilov. Tesla je technologickou spoločnosťou, ktorá tvorí množstvo inovatívneho a špičkového softvéru. Poskytuje energetickú infraštruktúru pre autá so svojimi nabíjačkami Tesla Superchargers, výrobu solárnej energie vo svojich továrňach Gigafactories a mnoho ďalšieho. Spracovanie a analýza údajov z ich vozidiel, inteligentných sietí a tovární a integrácia s ostatnými backendovými službami IT v reálnom čase je kľúčovou súčasťou ich úspechu. **Pritom údaje neprúdia s rovnakými alebo stabilnými dátovými modelmi či formátmi, takže ich je potreba neustále transformovať a integrovať medzi sebou.**

Spoločnosť Tesla vybudovala infraštruktúru dátovej platformy založenú na Kafke, ktorá podporuje milióny zariadení a prijíma, spracováva, ukladá a poskytuje bilióny dátových bodov denne. Spracovaním týchto biliónov dátových bodov denne z každej časti podniku získavajú v Tesle prehľad o ich vozovom parku, pričom na to stačí len niekoľko ľudí. To

[131] Zdroj: https://joinup.ec.europa.eu/collection/interoperability-test-bed-repository/solution/rdf-validator/about, Dátum referencie: 15.05.2023

[132] Zdroj: https://www.itb.ec.europa.eu/docs/tdl/latest/, Dátum referencie: 15.05.2023

[133] Zdroj: https://www.confluent.io/kafka-summit-san-francisco-2019/0-60-teslas-streaming-data-platform/, Dátum referencie: 23.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

pomohlo spoločnosti Tesla stať sa lídrom v odvetví tým, že sa jej otvorili nové možnosti, ako napríklad prichádzať s komplexnými predpoveďami na základe analýzy údajov z celého vozového parku s cieľom optimalizovať efektívnosť výroby a prevádzky a navrhovať novú generáciu výroby.

Akým výzvam čelí spoločnosť Tesla pri dátovej integrácii

Dátové toky internetu vecí sa na prvý pohľad veľmi podobajú na bežné udalosti v logoch webového servera. Generujú sa udalosti, niekedy vo veľkých objemoch, ktoré je potrebné spracovať a sprístupniť následným konzumentom alebo uložiť do databáz. Internet vecí by teda mal znamenať len viac udalostí oproti iným typom informačných systémov.

Po podrobnejšej analýze sa však ukáže, že hoci sa vyskytujú všetky bežné problémy serverových logov, k tomu pribúda aj množstvo nových, s ktorými sa treba popasovať. Namiesto webových serverov, ktoré sú pravdepodobne vo vašej sieti a pod vašou kontrolou, je súčasťou ekosystému veľké množstvo zariadení s premenlivou konektivitou, čo vedie k roztrieštеným údajom a dlhému zoznamu verzií firmvéru (nedá sa len tak prestať podporovať niektoré verzie) **so starými formátmi údajov**. Niektoré z týchto zariadení sa zároveň môžu „zblázniť“ a začať na infraštruktúru posielať hromady údajov, čo môže pripomínať útok typu „Denial-of-Service (DoS)“. Pri takomto dlhom zozname zariadení sa udalosti podobné DoS môžu stať súčasťou procesu a je potrebné ich vopred navrhnúť.

Bohužiaľ, to ešte nie je všetko. Niektoré dátové toky - najmä ak máte zariadenia, ktoré súvisia so zdravím populácie a bezpečnosťou - môžu mať vysokú prioritu a vyžadujú si veľmi nízku latenciu. Tieto dátové toky sa môžu miešať s tokmi, ktoré sú len vysoko-objemovými tokmi bežnej prevádzky, ktoré používajú analytici na vyhodnotenie a pochopenie stavu všetkých zariadení. Tým vznikajú zmiešané úrovne služieb v spoločnom prostredí, o ktoré sa treba starať.

Aby sa problémy ešte viac skomplikovali, mnohokrát môžu mať tieto dátové toky veľmi odlišné formáty údajov, ktoré sa nezávisle od seba vyvíjajú. Ako tím zameraný na spracovanie dátových tokov pravdepodobne tiež nemáte kontrolu nad tým, kde a kedy k týmto zmenám dochádza. A formáty údajov v zariadeniach pravdepodobne nebudú „príjemné“ formáty ako Apache Avro™[134] alebo Protobuf[135], a to z dôvodu požiadaviek na procesor a/alebo snahy o zhustené ukladanie a prenos údajov. Treba pritom dúfať, že dátové toky budú aspoň podporovať verzovanie, ak nie záruky kompatibility, aby sa uľahčilo spracovanie.

Okrem toho treba pridať aj niektoré základné funkcie správy a prehľadu vozového parku. Je nevyhnutné pochopiť, či sú zariadenia zdravé, či odosielajú dobré údaje, či negenerujú nejaké skreslenie atď. Vzhľadom na to, že sa jedná o hardvér, budú sa nevyhnutne vyskytovať podivné chyby, ktoré sa spustia len raz z milióna prípadov; v takej veľkej škále sú takéto odľahlé prípady každodenným javom, pred ktorým sa treba nielen chrániť, ale ho aj aktívne monitorovať.

---

[134] Zdroj: https://avro.apache.org, Dátum referencie: 23.05.2023

[135] Zdroj: https://protobuf.dev, Dátum referencie: 23.05.2023

Tento projekt je podporený z Európskeho sociálneho fondu.

Prečo nástroj Apache Kafka bol riešením pre spoločnosť Tesla

Ešte pred rozhodnutím o technológiách si v Tesle položili prvú otázku: Aké možnosti od systému potrebujeme? Pre internet vecí ide o tento súbor základných požiadaviek:

- Odolné úložisko,
- Jednoduchá horizontálna škálovateľnosť,
- Vysoká priepustnosť,
- Nízka latencia.

Treba mať na pamäti, že veľkým problémom tu nie je, ako získať údaje zo zariadení. Výzvou je skôr získané údaje rýchlo spracovať a sprístupniť ich následným používateľom, aby mohli vytvárať poznatky a zlepšenia, ktoré skutočne posúvajú spoločnosť vpred.

Základnou technológiou, ktorá umožňuje splniť všetky tieto ciele, je Apache Kafka®. Poskytuje odolné úložisko, ktorému možno dôverovať, že nestratí dôležité správy. Dokáže sa horizontálne škálovať aj o dva rády bez toho, aby sa prevádzka stala náročnejšou alebo aby sa významne zvýšila prevádzková réžia. Kafka poskytuje aj natívne možnosti spracovania tokov a bez ohľadu na to, či sa použijú, alebo sa využijú len API producenta a konzumenta, Kafka vie byť bleskovo rýchla (s nízkou latenciou) pri zachovaní neuveriteľne vysokej priepustnosti.

Okrem toho, že Kafka spĺňa všetky požiadavky uvedené pre internet vecí, je aj veľmi stabilná, dobre podporovaná a má silnú komunitu. Pokiaľ budeme pri návrhu systému okolo Kafky opatrní, nemal by byť problém so škálovaním až na desiatky biliónov udalostí denne, pričom prevádzková záťaž škálovania našich systémov by mala rásť sublineárne s objemom údajov. Najdôležitejšie je, aby sme vytvorili systém, ktorý bude možné úspešne prevádzkovať v neustále rastúcom rozsahu.

Pozrime sa ešte na to, čo musíme okolo tohto jadra založeného na Kafke vybudovať. Potrebujeme preniesť údaje zo zariadení do Apache Kafka, implementovať tokové spracovanie, aby boli tieto surové údaje použiteľné, a sprístupniť ich ostatným. Ako vždy, diabol sa skrýva v detailoch, ako toto spracovanie štruktúrovať tak, aby bolo flexibilné, škálovateľné a udržiavateľné. Netreba zabúdať, že tieto kroky spracovania údajov sa budú používať pre mnoho rôznych dátových tokov v rôznych formátoch, a to všetko pri zachovaní schopnosti spracovať bilióny udalostí každý deň. Našou úlohou je teda definovať primitívy, ktoré sú dostatočne flexibilné na to, aby sa dali opakovane použiť pre všetky rôzne prípady použitia a zároveň sa dali škálovať. Potrebujeme súbor opakovateľných vzorov, o ktorých vieme, že sa dajú škálovať na akýkoľvek objem. Tieto vzory sa dajú ľahko aplikovať na nové dátové toky a zariadenia, takže budeme pripravení škálovať nielen podľa objemu, ale aj podľa typu dát a dátového toku - a zároveň obmedziť prevádzkové zaťaženie tímov.

Ako sa pracuje s rôznorodými formátmi a schémami dátových tokov

Keď máme surové údaje v Kafke, musíme uľahčiť ich používanie konzumentskými aplikáciami. Údaje by sme mohli umiestniť priamo do databázy alebo objektového úložiska (teda do dátového jazera), ale aj iné tímy by mohli chcieť pristupovať k údajom ako k tokom udalostí s nízkou latenciou. Je tiež užitočné oddeliť logické fázy spracovania, aby sme mohli lepšie izolovať operácie a nezávisle škálovať fázy. Na tento

Tento projekt je podporený z Európskeho sociálneho fondu.

účel môžeme vytvoriť prechodný kanonický formát údajov, ktorá je agnostická pre všetky konkrétne procesy. Táto kanonická reprezentácia údajov je potom jediným rozhraním pre všetky nadväzujúce operácie. Funkciu, ktorá transformuje zdrojové údaje do tejto podoby, nazývame „parser“. Logiku parsovania môžeme vložiť do nástroja na spracovanie tokov (Kafka Streams[136] alebo alpakka-kafka[137] sú dve skvelé voľby), aby sme kanonické údaje sprístupnili s nízkou latenciou následným konzumentom.

Mohli by sme skombinovať logiku parsovania s logikou ukladania a logikou následných tém („topics“, ktoré sa definujú v Kafke), ale tým by sa nahromadilo veľa zložitosti do jednej fázy. Tým by sa tiež zbytočne prepojila priepustnosť spracovania s priepustnosťou, ktorú dokáže podporovať systém ukladania (t. j. databáza alebo dátové jazero).

Spomínaný prechodný kanonický formát je veľmi užitočný aj pri sprístupňovaní dátových tokov udalostí iným tímom v organizácii. Pomáha obmedziť zložitosť, s ktorou sa potýka zvyšok organizácie; nie každý musí byť schopný spustiť vlastný parser surových údajov alebo vedieť o veľkom úložisku správ, aby získal údaje. Môžeme tiež zabezpečiť, aby schéma v tejto parsovanej kanonickej téme zostala spätne kompatibilná.

Pri správe kanonických formátov môže byť veľmi nápomocný register schém Confluent („Confluent Schema Registry“)[138]. Je navrhnutý tak, aby bol centrálnym miestom na sledovanie schémy pre každú tému („topic“ definovanú v Kafke), čo uľahčuje pochopenie a interpretáciu toho, aké údaje sa v týchto kanonických témach nachádzajú. Je tiež vybavený konfigurovateľnou kompatibilitou pre jednotlivé témy - doprednou, spätnou alebo úplnou - takže sa garantuje, že údaje zostanú čitateľné.

V závislosti od dátových formátov, v ktorých prichádzajú údaje zo zariadení, budeme pravdepodobne potrebovať vytvoriť niektoré parsery pre štandardné formáty, ako sú JSON a CSV. Mali by sa dať pomerne rýchlo spustiť a vyriešia ihneď problémy mnohých tímov, ale určite by ste sa mali pozrieť na svoje bežné prípady - a porozprávať sa s používateľmi predtým, ako sa pustíte do implementácie týchto štandardov. Zároveň sa na platforme pravdepodobne začnú objavovať aj tímy, ktoré vytvárajú udalosti na strane servera (t. j. tie, ktoré pochádzajú z interného ekosystému, nie zo zariadení IoT). Hoci spoločným formátom je často len JSON, tímy na strane servera majú len zriedkakedy ospravedlnenie, aby nevytvárali správy s deklarovanou schémou (napríklad pomocou Avro); majú plnú kontrolu nad všetkými producentmi správ a pre spoločné formáty existujú knižnice v takmer každom jazyku. Na riešenie tohto problému stačí vystaviť rozhranie parsera pre spracovanie tokov, ktorý prijme správu a vytvorí z nej patričný počet udalostí. Rámec spracovania sa postará o zvyšok „kúzla“, ktoré spočíva v premene týchto udalostí na kanonické správy, ich odosielaní a zdieľaní pokroku. Tímy si potom musia len vybrať existujúci parser - alebo si môžu vytvoriť vlastný, ktorý bude podporovať ich vlastné formáty.

Keď máte veľký vozový park, ktorý musíte spravovať a analyzovať, je dôležité pochopiť, aká časť z tohto vozového parku je v prevádzke za daný deň. V závislosti od správania

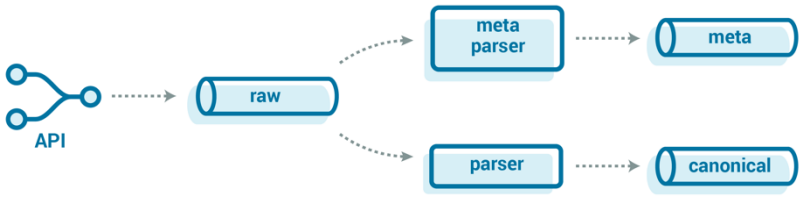
[136] Zdroj: https://docs.confluent.io/platform/current/streams/overview.html, Dátum referencie: 23.05.2023

[137] Zdroj: https://doc.akka.io/docs/alpakka-kafka/current/home.html, Dátum referencie: 23.05.2023

[138] Zdroj: https://github.com/confluentinc/schema-registry, Dátum referencie: 22.05.2023

sa používateľov v rôznych časoch dňa môžeme denne vidieť 90 % flotily alebo 30 % flotily. Ak toto percento náhle klesne, budeme chcieť vedieť aj to.

V tomto prípade je veľmi účinný tok metadát. Je postavený na tom istom toku surových údajov, ktorý sme použili na štandardné parsovanie surových údajov do kanonického formátu, ale sleduje spoločné vlastnosti udalostí bez úplného rozboru každej správy. Patrí sem typ zariadenia, UUID zariadenia, verzia firmvéru, čas prijatia, zdrojová téma („topic“ v Kafke), veľkosť správy a prípadne čas začiatku/konca správy.



Obrázok 15: Tok surových údajov s parsovaním na metadáta a do kanonického formátu[139]

Keď máme tieto údaje, môžeme vykonávať veľmi zaujímavé dotazy, ktoré nám pomôžu pochopiť stav vozového parku, ako napríklad:

- Pre akú časť flotily sme získali údaje za posledný deň? A čo za posledné dva dni alebo posledný týždeň?
- Ktoré zariadenia nikdy neodoslali údaje alebo chronicky meškajú?
- Aké veľké sú správy?

Existuje viacero spôsobov, ako tento tok metadát zhmotniť a odpovedať na otázky. Môže to byť ďalšia tabuľka v databáze časových radov. Metadáta o časovom rade sú často samy o sebe len časovým radom. Ak máte veľa metadát o udalostiach, ktoré treba zhromaždiť a analyzovať, mohol by fungovať aj Elasticsearch[140] alebo zhlukovanie udalostí v dátovom jazere alebo v technológii, ako je Apache Druid[141].

[139] Zdroj: https://www.confluent.io/blog/stream-processing-iot-data-best-practices-and-techniques/, Dátum referencie: 23.05.2023

[140] Zdroj: https://www.knowi.com/blog/what-is-elastic-search/, Dátum referencie: 23.05.2023

[141] Zdroj: https://druid.apache.org, Dátum referencie: 23.05.2023

8 Implementácia štandardu

Tabuľka 9: Prehľad úloh potrebných na implementáciu štandardu

Hľadisko	Téma	Úloha
Legislatívne	Rozšírenie Vyhlášky o štandardoch pre informačné technológie verejnej správy (č. 78/2020 Z. z.)	Zadefinovanie dátovej transformácie. Určenie cieľového formátu (JSON-LD).
		Doplnenie metód dátovej transformácie: • Obohacovanie údajov • Stotožňovanie údajov • Odvodzovanie hodnôt • Zmeny dátového modelu (XSLT transformácie) • Prekladače správ • Zabalenie a rozbalenie údajov • Zlúčenie údajov z viacerých zdrojov
		Doplnenie metód dátovej transformácie pre účely analýzy údajov: • Vyhladzovanie • Agregácia • Diskretizácia • Zovšeobecnenie • Indexovanie a radenie • Anonymizácia a šifrovanie • Konštrukcia atribútov • Normalizácia • Manipulácia • Kompresia údajov
Organizačné	Centrálna realizácia projektov pre interoperabilitu na úrovni EÚ	Zabezpečenie poskytovania mojich údajov cez EDIW
		Zabezpečenie poskytovania mojich údajov cez OOTS
	Vybudovanie centrálnych nástrojov pre transformáciu údajov	Implementácia centrálneho transformačného modulu (ako súčasť CIP)
		Rozšírenie KAV o nástroje pre metódy dátovej transformácie pre potreby analýzy údajov
	Centrálna realizácia transformácie	Rozvoj Centrálneho modelu údajov
		Vytvoriť transformačné schémy pre objekty evidencie dostupné v rámci CIP a používané v MOU

Tento projekt je podporený z Európskeho sociálneho fondu.

Hľadisko	Téma	Úloha
	Lokálna konzumácia transformácie	Úprava integračných väzieb na úrovni ISVS pri konzumácii údajov pre prepojené údaje
Finančné	Zabezpečenie kapacít	Dátová kancelária verejnej správy bude mať dostatočné personálne kapacity (aspoň 3 zamestnancov), ktorý budú spravovať Centrálny model údajov a spravovať transformácie v rámci transformačného modelu CIP
	Rozvoj informačných systémov verejnej správy, aby pracovali s prepojenými údajmi	Rozvojové zmluvy na ISVS by mali obsahovať časť na využívanie prepojených údajov

Tento projekt je podporený z Európskeho sociálneho fondu.

Contact us

Some or all of the services described herein may not be permissible for KPMG audit clients and their affiliates or related entities.

www.kpmg.com