Operačný program
Efektívna
verejná správa

Európska únia
Európsky sociálny fond

MINISTERSTVO
INVESTÍCIÍ, REGIONÁLNEHO ROZVOJA
A INFORMATIZÁCIE
SLOVENSKEJ REPUBLIKY

Výstup č. 1.1.4

Štandardizácia pre strojovú spracovateľnosť a open API

Zmluva o dielo č. 445/2022

Projekt:

Zlepšenie využívania údajov vo verejnej správe

ITMS kód projektu:

314011S979

KPMG

Tento projekt je podporený z Európskeho sociálneho fondu.

Preskúmanie a schválenie dokumentu

História revízií

Verzia	Autor	Dátum	Poznámka

Tento dokument schválil

	Meno	Dátum schválenia
1		
2		
3		
4		
5		

Tento projekt je podporený z Európskeho sociálneho fondu.

Slovník pojmov

Skratka / Pojem	Vysvetlenie
API	Aplikačné programové rozhranie
API Gateway	Centrálny bod pre správu, riadenie a zabezpečenie prístupu k viacerým API prostredníctvom jednotného rozhrania.
EÚ	Európska únia
JSON-LD	JavaScript Object Notation for Linked Data
HVD	Súbory údajov s vysokou hodnotou (z angl. High Value Datasets)
MIRRI	Ministerstvo investícií, regionálneho rozvoja a informatizácie Slovenskej republiky
Mock API	Nástroj, ktorý imituje správanie skutočného API pre testovanie a simuláciu bez potreby pripojenia k reálnemu systému alebo dátam.
NKIVS	Národná koncepcia informatizácie verejnej správy
NASES	Národná agentúra pre sieťové a ekonomické služby
Open API	Aplikačné programové rozhranie (API), ktoré je voľne dostupné resp. zverejnené, t.j. umožňuje vývojárom použitie API na naprogramovanie vlastného softvéru alebo aplikácie. Dostupnosť open API môže byť zároveň obmedzená na základe určitých podmienok.
OP II	Operačný program Integrovaná infraštruktúra
OVM	Orgán verejnej moci
RDF	RDF (Resource Description Framework) - štandard výmeny dát na webe
sandbox	testovacie prostredie/testovacia služba pre sw vývojárov
SLA	SLA (Service Level Agreement) je dohoda medzi poskytovateľom služby a zákazníkom, ktorá stanovuje úroveň služieb, kvalitu, dostupnosť a očakávané výkonnostné parametre služby.
SR	Slovenská republika
URI	Jednotný identifikátor (kompaktný reťazec znakov používaný na identifikáciu alebo pomenovanie zdroja)
VS	Verejná správa
XML	Rozšíriteľný značkovací jazyk (z angl. eXtensible Markup Language), ktorý bol vyvinutý a štandardizovaný konzorciom W3C (World Wide Web Consortium) ako pokračovanie jazyka SGML a zovšeobecnenie jazyka HTML. Umožňuje jednoduché vytváranie konkrétnych značkovacích jazykov na rôzne účely a široké spektrum rôznych typov údajov.

Obsah

Tento projekt je podporený z Európskeho sociálneho fondu.

1 Úvod a zhrnutie

Výstup č. 1.1.4: Štandardizácia pre strojovú spracovateľnosť a open API 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, zlepšiť efektivitu a adresnosť služieb na základe lepšieho využívania dát.

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

Tento výstup vznikol ako realizácia aktivity číslo 1 Manažment kvality údajov, pričom zámerom tejto aktivity je zaviesť manažment kvality údajov do procesov vybraných inštitúcií verejnej správy.

1.1 Metodika realizácie výstupu

Tento dokument sa venuje sprístupňovaniu údajov prostredníctvom open API vzhľadom na významný nárast globálneho open API trhu ako aj smerovanie v rámci EÚ v zmysle aktuálnej digitálnej dekády, digitálnej transformácie vlád a európskej digitálnej stratégie.

V prvej kapitole je nastolený rámec pre posúdenie súčasnej aplikácie a trendov v štandardizácii open API, pozornosť je venovaná predovšetkým:

- aktuálnym trendom EÚ a v zahraničí v podobe
    - rámca pre API vo verejnom sektore s uvedením vybraných príkladov zo zahraničia, ktoré sú zároveň podkladom pre vytvorenie tohto rámca
    - zvýšeného dôrazu kladeného na strojovú spracovateľnosť a dostupnosť údajov prostredníctvom API v spojitosti so súbormi údajov s vysokou hodnotou
    - princípov posilnenia práv jednotlivca pri zdieľaní svojich údajov s tretími stranami, najmä v podobe vplyvu GDPR a riešení uplatňujúcich tieto princípy
- posúdeniu súčasnej situácie na Slovensku s cieľom uvedenia základného rámca pre štandardizáciu, už realizovaných alebo prebiehajúcich aktivít a identifikácie priestoru na zlepšenie v závere kapitoly.

V ďalšej kapitole sú uvedené vybrané príklady dobrej praxe, ktoré predstavujú prierez osvedčených postupov publikovaných medzinárodnými organizáciami alebo inými štátmi vo vzťahu k definovaniu a zavádzaniu open API či už v súkromnom alebo verejnom sektore.

Tento projekt je podporený z Európskeho sociálneho fondu.

Na základe zistení z prvých dvoch kapitol dokument načrtáva odporúčania, ktoré v podmienkach Slovenska môžu smerovať k lepšej štandardizácii pri aktivitách MIRRI a orgánov verejnej moci.

Záverečná kapitola je venovaná aktualizácii štandardu v podobe:

- návrhu na doplnenie štandardu open API pre Moje dáta (služba Moje dáta je postavená na Open API infraštruktúre, pričom posledná verzia štandardu pre open API pre Moje dáta[1] už nezohľadňuje aktuálny vývoj v rámci riešenia manažmentu osobných údajov),
- návrhu na vytvorenie best practices ako zdroja pri usmernení v oblasti zavádzania open API.

[1] Moje-Data-02-Standard-OPEN-API_v2-1.pdf (datalab.digital)

Tento projekt je podporený z Európskeho sociálneho fondu.

2 Posúdenie súčasnej aplikácie a trendov v štandardizácii open API pre prístup k údajom

Globálny open API trh mal v roku 2021 hodnotu 2,39 miliónov dolárov (USD) a predpokladá sa, že táto hodnota narastie do roku 2030 na 13,21 miliónov dolárov. Open API trh v sebe z pohľadu segmentácie zahŕňa aj vládne open API, pričom z pohľadu regiónov sa očakáva, že najvyššiu mieru rastu na tomto trhu bude vykazovať Európa (trhu v súčasnosti dominuje Severná Amerika)[2].

Digitálna transformácia prebiehajúca práve v Európe má za cieľ posilniť postavenie podnikov a ľudí a smerovať k udržateľnej, prosperujúcej a na človeka orientovanej digitálnej budúcnosti[3]. Jedným z cieľov súčasnej digitálnej dekády (do roku 2030) je maximalizovať digitalizáciu služieb poskytovaných verejným sektorom[4].

2.1 Rámec pre API vo verejnom sektore

Významným predpokladom digitálnej transformácie sú API (aplikačné programové rozhrania), čoho dôkazom je aj snaha o definovanie rámca pre API vo verejnom sektore[5] na úrovni EÚ. Tento rámec smeruje k jednotnému (štandardizovanému) postupu pri tvorbe API vo vládnom prostredí stanovením súboru dvanástich navrhovaných opatrení, ktoré sú popísané nižšie.

Navrhované opatrenia sú priradené trom úrovniam aktivít/riadenia:

- strategickej (podpora na úrovni politík),
- taktickej (úroveň riadenia, na ktorej dochádza k rozhodnutiu o alokovaní zdrojov),
- operatívnej (úroveň samotnej implementácie API).

Pre každú úroveň sú brané do úvahy štyri piliere:

- politiky,
- platforma a ekosystémy,
- ľudia,
- procesy.

Tabuľka 1 - Prehľad navrhovaných opatrení (Rámec pre API vo verejnom sektore)

Úroveň aktivity/riadenia	Politiky	Platforma a ekosystémy	Ľudia	Procesy

[2] Open API Market Analysis, Share, Report to 2030 (straitsresearch.com)
[3] Europe's Digital Decade: digital targets for 2030 (europa.eu)
[4] Europe's Digital Decade | Shaping Europe's digital future (europa.eu)
[5] An Application Programming Interfaces (APIs) framework for digital government - Publications Office of the EU (europa.eu)

Tento projekt je podporený z Európskeho sociálneho fondu.

Strategická úroveň	1. Zosúladenie API s cieľmi definovanými v politikách	2. Definovanie vízie digitálnej platformy na vládnej úrovni	3. Vytvorenie riadiacich štruktúr	4. Usmernenie v podobe princípov pre API procesy
Taktická úroveň	5. Navrhnutie metrík a prioritizácie API	6. Harmonizovanie platformy a ekosystému	7. Vytvorenie multikompetenčných tímov	8. Uplatňovanie produktového prístupu k API
Operatívna úroveň	9. Meranie vplyvu API	10. Vytvorenie API komponentov (komponentov API platformy)	11. Vymenovanie produktových manažérov a vytvorenie API tímov	12. Osvojenie prístupu orientovaného na životný cyklus API

1. Zosúladenie API s cieľmi definovanými v politikách

Toto opatrenie vyžaduje zosúladenie API s kľúčovými vládnymi politikami, stratégiami a celkovými plánmi zvážením skutočnosti, či API pomôžu dosiahnuť stanovené ciele.

V súkromných spoločnostiach sa odporúča, aby vyvíjali API v súlade so svojou obchodnou stratégiou - podobný prístup by sa dal použiť aj v kontexte verejného sektora. Vlády majú stanovené svoje politiky a rozhodovacie procesy, pričom ich ciele v oblasti digitálneho prostredia môžu byť dosiahnuté efektívnejšie využitím API.

Tento rámec navrhuje pre vlády prístup „API-first“, pri ktorom organizácia už vo fáze navrhovania riešenia vyhodnocuje, či API môžu pomôcť naplneniu definovaných cieľov. Implementácia takéhoto prístupu zároveň vyžaduje revíziu všetkých politík z pohľadu zodpovedania otázky, či ich ciele môžu byť dosiahnuté prostredníctvom API (nasleduje identifikácia zodpovedných osôb za aktivity vyplývajúce z danej politiky, pochopenie čo presne má byť dosiahnuté a pod.).

Príklad: V štáte Viktória (Austrália) zaviedli na vládnej úrovni API-first prístup, podľa ktorého všetky nové digitálne služby a poskytnuté údaje musia byť vytvárané s použitím API.

2. Definovanie vízie digitálnej platformy na vládnej úrovni

Potvrdenie celoštátnej vízie digitálnej transformácie na vládnej úrovni napomáha pri budovaní vzťahov zo zainteresovanými stranami a pri nastavovaní priorít v rámci zavádzania API.

Jasne definovaná platforma napomáha lepšej koordinácii pri rozhodovaní o prideľovaní zdrojov a voľbe implementačných postupov. Zavádzanie API vládami predstavuje platformovo založený model, ktorého súčasťou sú služby a údaje zdieľané interne (v rámci verejného sektora napríklad medzi jednotlivými OVM alebo medzi oddeleniami jednej OVM) alebo externe s tretími stranami. V takomto modeli vláda môže vystupovať súčasne v roli poskytovateľa (služby/údajov), konzumenta (služby/údajov) ako aj v roli regulátora. Bez jasnej definície by mohlo vzniknúť riziko, že dôjde iba k replikovaniu existujúcich papierových procesov do podoby digitálnych služieb.

Príklad: Národná agentúra pre digitálnu transformáciu v Austrálii prepracovala digitálne služby na automatizované a vyvolané kľúčovými životnými stavmi (ako akcie založené na životných siuáciách). Podobný model bol zavedený v Singapure ako súčasť ich API a digitálnej stratégie s názvom „Iniciatíva životné okamihy“, ktorá vyžadovala zároveň prepojenie medzi jednotlivými orgánmi verejnej moci za účelom poskytnutia služby občanovi v prípade výskytu príslušného životného okamihu (životnej situácie).

Tento projekt je podporený z Európskeho sociálneho fondu.

3. Vytvorenie riadiacich štruktúr

Vytvorené riadiace štruktúry majú zabezpečiť súlad API so stanovenými politikami a prioritnými prípadmi použitia, zohľadniť bezpečnostné hrozby a riziká, vyhodnotiť dopady zavádzaných API a zabezpečiť použitie relevantných a dohodnutých štandardov resp. pokynov (návodov). Vlády majú skúsenosti s budovaním efektívnej správy údajov použitím rámcov na riadenie informácií, pričom niektoré vlády v súčasnosti používajú obdobné princípy aj na riadenie API.

Riadiace štruktúry uskutočňujú integráciu verejných služieb, zabezpečujú holistické riadenie činností interoperability naprieč administratívnymi úrovňami a sektormi, identifikujú a vyberajú vhodné štandardy a špecifikácie, participujú na štandardizácii práce nevyhnutnej pre uspokojenie potrieb organizácie. Zároveň riadia riziká, stanovujú spoločné pravidlá na riešenie neočakávaných problémov a zabezpečujú dodržiavanie širších princípov a zásad vládnej politiky.

Podľa európskeho rámca pre interoperabilitu si integrované riadenie verejných služieb za účelom zabezpečenia interoperability vyžaduje organizačné štruktúry a roly zodpovedné za poskytovanie a prevádzku verejných služieb, dohody o úrovni služieb, vytváranie a riadenie dohôd o interoperabilite, postupy riadenia zmien a plány zabezpečenia kontinuity a kvality údajov. Štruktúry riadenia si vyžadujú viacero úrovní dohľadu.

Na úrovni EÚ je potrebné usmernenie týkajúce sa návrhu API a štandardov riadenia ich životného cyklu, ktoré by malo biznis aj technické zameranie a pôsobilo by ako centrum excelentnosti.

Na úrovni inštitúcií EÚ, členských štátov, regiónov a miest sú potrebné celonárodné výbory, ktoré by dohliadali na interoperabilitu, rozhodnutia o infraštruktúre s podporou API, princípy návrhu API a stratégie riadenia produktov. Verejná správa by mohla na každej úrovni tiež mapovať zdieľanie schopností (kapacít) a spoločné súbory údajov a prideľovať vedúcich oddelení pre API a registre a spoločné úložiská údajov, ktoré by mali fungovať v rámci celej verejnej správy. Každé oddelenie by potom potrebovalo svoj vlastný riadiaci výbor, ktorý by zabezpečil, že API stratégie na úrovni oddelenia využívajú vládou stanovenú architektúru, že sa používajú zdieľané informácie a že sa dodržiavajú najlepšie praktiky (tzv. „best practices“).

Ďalšou výzvou bude vytvorenie relevantných riadiacich štruktúr zameraných na podporu zdieľania znalostí, predchádzanie duplicity a zabezpečenie používania interoperabilných a štandardizovaných metodológií bez vytvárania príliš veľkého počtu dodatočných výborov alebo prerozdeľovania projektových zdrojov na vytváranie nových riadiacich štruktúr.

Príklad: V štáte Viktória (Austrália) bolo vytvorené usmernenie pre riadenie informácií, ktoré môže byť užitočným pre vládu snažiacu sa vytvoriť a spravovať riadiace štruktúry a zároveň bol vytvorený celovládny API-gateway tím, ktorý je zodpovedný za:

- vytvorenie technológie na podporu vývoja a vystavenia API v rámci vládnych organizácií,

Tento projekt je podporený z Európskeho sociálneho fondu.

- vytváranie a spoluprácu pri tvorbe API štandardov,
- konzultácie, školenia a podporu zlepšenia celkovej gramotnosti v oblasti API a integrácií,
- zapojenie a rozšírenie API a integrácií či už interne alebo aj voči externým partnerom a celej komunite mimo vládneho sektora.

4. Usmernenie v podobe princípov pre API procesy

Zhromaždenie existujúcich princípov vzťahujúcich sa k poskytovaniu digitálnych služieb, výberu technológií, zabezpečeniu ochrany údajov a kyberbezpečnosti je užitočné pri alokovaní zdrojov a navrhovaní aktivít smerujúcich k implementácii API.

Silná a zdokumentovaná základňa v podobe definovaných a dostupných princípov pomáha zabezpečiť, že všetky zainteresované strany sú schopné tieto princípy reflektovať a navrhovať činnosti, ktoré sú v súlade so stanovenými zásadami.

Vedúci predstavitelia vlády môžu preskúmať základné princípy organizácie a zabezpečiť, aby im porozumeli tímy zodpovedné za dohľad nad digitalizáciou, interoperabilitou a aktivitami zameranými na API na úrovni celej vlády a ministerstiev. Tieto princípy by zdieľali členovia jednotlivých štruktúr riadenia, aby sa zabezpečilo ich začlenenie do procesov dohľadu.

5. Navrhnutie metrík a prioritizácie API

Vlády majú skúsenosti s prioritizáciou práce a jednotlivých aktivít v súvislosti s obmedzenými zdrojmi a rozpočtovými obmedzeniami. Podobne aj pre oblasť API je vhodné prioritizovať poradie vládnych činností súvisiacich s API a definovať, ako sa bude merať úspech pre každú oblasť činnosti.

V kontexte digitálnej transformácie vlád často dochádza k realizácii mnohých činností súvisiacich s API, ktoré sú však na sebe nezávislé. Aby sa zabezpečilo, že API dosahujú ciele stanovenej politiky, bude potrebné zaviesť metriky na meranie vplyvu API.

Príklad: Regionálna vláda v Lombardii (Taliansko) vytvorila platformu E015 ako súčasť celoštátnej aktivity pre svetovú výstavu Expo s cieľom demonštrovať digitálne partnerstvá. Tento príklad si vyžadoval, aby celovládny koordinačný orgán spolupracoval s jednotlivými rezortmi a dohodol si prioritné činnosti. Súčasťou platformy boli spoločné prvky naprieč všetkými oddeleniami (ako infraštruktúra platformy a portál pre vývojárov) a zároveň aj individuálne aktivity oddelení za účelom vytvorenia API vo vzťahu k ich doménovej oblasti.

6. Harmonizovanie platformy a ekosystému

API platformy vyžadujú, aby sa množstvo zainteresovaných strán dohodlo na minimálnom súbore spoločných oblastí pre zlepšenie interoperability (spoločné API štandardy, spoločne definované modely údajov, spoločné architektonické možnosti) ako aj na identifikovaní komponentov služieb, ktoré môžu byť opätovne použité.

Tento projekt je podporený z Európskeho sociálneho fondu.

Implementácia vládnych platforiem je stále v ranom štádiu, pričom niektoré vlády sa zameriavajú na interné ekosystémy vzájomnej spolupráce (v podobe ministerstiev a verejných orgánov spolupracujúcich v danej doménovej oblasti) a niektoré zahŕňajú aj externých partnerov a ďalšie zainteresované strany. Prístup spoločnej podnikovej IT architektúry navrhuje európska komisia a zároveň je presadzovaný niekoľkými štátmi (napríklad Fínsko, Dánsko), pričom takýto prístup môže vytvoriť základy pre vytvorenie API gateway a celoštátnych vývojárskych portálov. Nevýhodou však môže byť otázka flexibility takýchto systémov, pretože príliš rigidná centralizovaná architektúra, v ktorej všetky služby musia smerovať cez jeden „lievik“, môže byť prekážkou pre neskoršiu flexibilitu potrebnú pre budúce prípady použitia.

V súlade s európskym štandardom pre interoperabilitu by sa definícia spoločnej architektúry pre vládne platformy mala interpretovať na základe použitia štandardizovaných prístupov na rôznych vrstvách, použitia infraštruktúry s podporou API a zdieľaného porozumenia navrhnutých vzorov.

Potrebné je zosúladenie všetkých zainteresovaných strán v nasledovných oblastiach, resp. v rámci nasledovných komponentov:

- prioritné ekosystémy (zainteresované strany, ktoré sa podieľajú na odbornej expertíze v danej doméne – doprava, poľnohospodárstvo a pod., ktoré sú schopné identifikovať relevantné prípady použitia a potreby pre API),
- registre údajov (Mali by sa vyberať a analyzovať zdieľané údaje s cieľom určenia minimálnych požiadaviek na formát údajov, sémantiku, vlastnosti, vzťahy a pod., aby sa predišlo ad-hoc iniciatívam využitím opätovného použitia existujúcich modelov a modelov všeobecne uznávaných ako napríklad Schema.org. Jednotné slovníky môžu byť použité ako jediný zdroj pravdy, ktorý eliminuje duplicity a umožní opakované použitie.),
- zdieľané služby (služby v podobe jednej časti reťazca, ktoré je možné opätovne použiť ako napríklad overenie identity, ktoré môže byť vyvinuté raz a použité v rámci viacerých služieb rôznych oddelení),
- jeden inventárny bod (API katalóg alebo API portál, ktorý umožňuje používateľom prístup k službám a súborom údajov cez API),
- spoločné technologické štandardy (zabezpečia, že API sú ľahko zrozumiteľné a replikovateľné, pretože zdieľajú jednotnú nomenklatúru a ďalšie prvky dizajnu),
- spoločný právny rámec dohôd (zabezpečenie spolupráce medzi rôznymi organizáciami v súlade s platným legislatívnym rámcom, stratégiami a politikami).

Príklad: Podľa talianskej digitálnej stratégie sú doménovo orientované ekosystémy považované za kľúčové pri pomoci vláde prioritizovať a realizovať aktivity k implementácii API, najmä:

- podporovať víziu orientovanú na občana a podniky, ktorá má viesť k vytváraniu služieb umožňujúcich jednoduchšiu interakciu s verejnou správou,
- štandardizovať prístup k rozvoju služieb verejnej správy,
- stimulovať interoperabilitu,

Tento projekt je podporený z Európskeho sociálneho fondu.

- zúročiť skúsenosti, ktoré jednotlivé orgány verejnej správy získali v rámci skvalitňovania osvedčených postupov.

V Írsku boli identifikované problémy dátovej architektúry, ktoré viedli k nedostatočnému zdieľaniu údajov medzi orgánmi verejnej správy ako aj k duplicitám pri ukladaní a zbere údajov. Írska vízia efektívnejšieho dátového ekosystému je založená na využití API – zavádzajú registre (ako jediné autoritatívne zdroje údajov, ktoré orgány verejnej správy musia používať a ku ktorým majú prístup cez API) a povzbudzujú k zverejňovaniu všetkých API do jedného katalógu.

7. Vytvorenie multikompetenčných tímov

Tímy zodpovedné za riadenie a správu API vyžadujú celý rad zručností vrátane pochopenia relevantných politík, produktového manažmentu ako aj zručností potrebných pre oblasť technickej implementácie. Vládne API tímy potrebujú zapojenie technicky orientovaných osôb ako aj osôb zameraných na príslušné politiky/programy.

Typické zloženie tímu by malo zahŕňať vedúceho tímu, ktorý môže vykonávať aktivity prislúchajúce produktovému manažérovi za účelom zabezpečenia použiteľnosti API ako aj jeho súladu s potrebami používateľov. Vedúci API tímu tiež musí komunikovať s tvorcami politík, aby zabezpečil, že API zároveň slúži aj cieľom stanoveným v príslušných politikách. Ďalší členovia tímu pri tvorbe API musia vykonávať rozhodnutia vyplývajúce z možných uskutočniteľných riešení a založené na osvedčených postupoch, ktorých následkom môže byť napríklad zmena zamerania API. V takýchto prípadoch sa vedúci tímu musí uistiť, že zmeny nebudú mať za následok nesúlad s pôvodným zámerom a príslušnými politikami, resp. cieľmi, ktorých naplnenie malo byť dosiahnuté práve prostredníctvom vytvorenia API. V neposlednom rade súčasťou API tímu musia byť IT architekti a vývojári, ktorí technicky zabezpečia nasadenie API do prevádzky a tým umožnia integráciu interných alebo aj externých používateľov na vládne API.

Tie osoby, ktoré sú zapojené do procesov digitálneho vládnutia, digitálnych inovácií alebo projektov interoperability by mali byť vyškolení v oblasti API a procesoch dizajnérskeho myslenia.

Príklad: Odporúčania v oblasti open API pre mestá od 6Aika[6] (Fínsko) obsahujú osvedčené postupy a odporúčania k multidisciplinárnym tímom za účelom riadenia a implementácie API stratégií. V USA boli prijaté zákony, ktoré majú zabezpečiť, aby vláda mala prístup k online úložisku nástrojov, osvedčených postupov („best practices“) a štandardov, ktoré majú uľahčiť zavádzanie API.

8. Uplatňovanie produktového prístupu k API

Vlády musia vyčleniť zdroje na správu rozhraní ako „pokračujúce“ aktíva aplikovaním programového rozpočtovania. Čím viac sa vlády posúvajú smerom k platformovému

[6] 6Aika spojenie šiestich najväčších fínskych miest, ktoré sa spoločne vytvorili tzv. Six City Strategy zahŕňajúcu množstvo projektov zameraných na udržateľný rozvoj miest Smart Cities Work Together - 6Aika

Tento projekt je podporený z Európskeho sociálneho fondu.

modelu, tým viac strán bude pri vytváraní a poskytovaní produktov čoraz viac závislých od vládnych API. Pre takéto produkty a služby je nevyhnutné, aby vládne API, ktoré využívajú, boli dostupné a fungovali podľa očakávaní.

Produktový prístup k API má zabezpečiť, že vládne rezorty prideľujú zdroje a systémy potrebné na vybudovanie dôveryhodného ekosystému (zároveň, že API vytvárajú hodnotu).

Vlády často dlhodobo poskytujú služby vo forme programov, pričom na splnenie konkrétnych cieľov alebo riešenie krátkodobých potrieb sú využívané časovo obmedzené projekty. Pri zavádzaní API do vlád musia ministerstvá uvažovať o API ako o programoch, resp. v terminológii súkromného sektora možno použiť pojem produkt. Znamená to, že sa s nimi zaobchádza ako so strednodobými alebo dlhodobými aktívami, napr.:

- budú vyžadovať udržateľnú a aktualizovanú dokumentáciu, ktorá bude relevantná pre rôzne používateľské skupiny,
- bude potrebné ich pravidelne kontrolovať, zlepšovať a aktualizovať,
- bude potrebné monitorovať ich využívanie, aby bolo možné vyhodnotiť, či stále vytvárajú hodnotu a či napĺňajú ciele organizácie.

Súčasťou API produktového prístupu sú aj jasne definované práva a oprávnenia súvisiace s používaním API. Ak sú raz API sprístupnené a zdokumentované ako opätovne použiteľné komponenty produktov a služieb (či už interne vo vládnom prostredí alebo externe pre tretie strany), používatelia si musia byť istí, že API je dostupné, výkonné a prístupné. Dostupnosť znamená, že API sa dá nájsť a pochopiť napríklad prostredníctvom katalógu vládnych API, a zároveň, že neprestane neočakávane fungovať. Výkonnosť znamená, že API poskytuje dáta alebo služby včasným a konzistentným spôsobom. Prípustnosť znamená, že koncoví používatelia chápu svoje zodpovednosti a majú primeranú úroveň zabezpečenia a autorizácie potrebnú pre použitie API pre ich konkrétny prípad. Napríklad externí používatelia budú musieť vedieť, že majú povolené používať vládne API pre produkt súkromného sektora a zároveň musí byť zabezpečená ochrana osobných a citlivých údajov, ktoré využívajú interné API, avšak externí používatelia k nim nemôžu mať prístup.

API by sa mali najskôr používať interne, t. j. pri identifikácii prípadov použitia na tvorbu API pre poskytovanie údajov alebo služieb, by mali byť interné prípady použitia prioritné. API by sa malo používať interne na riadenie toku informácií alebo na umožnenie funkčnosti v rámci oddelenia/ministerstva alebo medzi oddeleniami/ministerstvami. Každé API by malo mať definované ciele na úrovni služby alebo očakávané hodnoty výkonu pre interné zainteresované strany (prípadne ak sa sprístupní širšiemu okruhu používateľov, malo by mať definované očakávania, ako bude fungovať a ako sa bude používať, keď bude vystavené tretím stranám).

Príklad: Požitie API produktového prístupu odporúčajú usmernenia kanadskej vlády, ktoré sú postavené na štyroch základných aktivitách:

- používajte interne čo vytvoríte (interné API piloty),
- podporujte API počas celého životného cyklu,

Tento projekt je podporený z Európskeho sociálneho fondu.

- merajte a zverejňujte API benchmarky,
- publikujte a dokumentujte API.

Zdroje 6Aika Toolkit (Fínsko) pomáhajú pracovníkom verejných služieb pochopiť úlohu API a ich potenciál, pričom zároveň poskytujú odporúčania na podporu štandardizovaných osvedčených postupov pri poskytovaní API vo verejnej správe.

Úrad austrálskej vlády pre digitálnu transformáciu podporuje neustále zlepšovanie API a snahu o dosahovanie určitej úrovne výkonnostných štandardov. Produktový manažment pre vysoko kvalitné API musí zahŕňať nasledovné:

- API, ktorého použitie je dlhodobo hodnotené ako ťažké, by malo byť opravené, Je potrebné zabezpečiť, aby menej ako 20% spätnej väzby spočívalo v negatívnom hodnotení v podobe ťažkej použiteľnosti API;
- chybové hlásenia by mali obsahovať ľudským okom čitateľné chybové hlásenie, ktoré je určené na čítanie a pochopenie zo strany používateľa;
- chybové hlásenia by mali obsahovať diagnostickú správu obsahujúcu technické podrobnosti, ktoré môže použiť vývojár/správca aplikácie, ktorá API využíva;
- všetky API by mali mať zverejnenú SLA zmluvu a podľa nej aj skutočne fungovať,
- zverejnenie dokumentácie a poskytnutie odkazu na dokumentáciu z koncového bodu API.

9. Meranie vplyvu API

Hodnota API sa musí merať priebežne a transparentne, rovnako ako sa meria a monitoruje akýkoľvek vládny program, aby sa zabezpečilo, že funguje podľa očakávaní a vytvára hodnotu pre občanov, podniky a životné prostredie. Hodnota a vplyv API by sa mali merať aby sa zabezpečilo, že sú výkonné a poskytujú hodnotu pre vládu a ostatné zainteresované strany v rámci ekosystému, a že neúmyselne nespôsobujú škodu alebo prehĺbenie nerovnosti.

Kým na meranie API z technickej stránky existuje množstvo príkladov, na meranie vplyvu API z pohľadu ich vplyvu na politiky (vplyvu hodnoty, ktorú vytvárajú na politiky štátu) je nedostatok. Obdobne je to aj v prípade súkromného sektora, ktorý sa prevažne zameriava na metriky generovania príjmov a metriky technickej výkonnosti API.

Každodenné využívanie API so sebou prinieslo tri základné typy metrík:

- Výkon

  Pôvodne boli metriky pre API zavedené s cieľom zabezpečiť, aby boli API robustné a výkonné. Napríklad metriky ako dostupnosť, bezpečnosť a rýchlosť odozvy pomohli vývojárom zabezpečiť, že budú dosiahnuté ciele služby. Tieto metriky predstavujú najbežnejšiu formu merania, ktorá je využívaná súkromným sektorom ako aj vládami.

- Strategická hodnota

Vzhľadom na to, že API sa čoraz viac považujú za reálny spôsob, ako podniky môžu dosiahnuť svoje strategické ciele, vytvorili sa kľúčové ukazovatele výkonnosti na lepšie meranie vplyvu API na podnikanie, ako napríklad schopnosť vytvárať príjmy alebo zvyšovať angažovanosť na cieľových trhoch. Takýto prístup je uplatňovaný v súkromnom sektore, zatiaľ nie je aplikovaný ako zaužívaný prístup vo vládnom prostredí.

- Vplyv na ekosystém

  V súlade s produktovým prístupom spolu s meraním výhod API na podnikanie, boli vytvorené nové opatrenia (resp. metriky) na zabezpečenie prijatia treťou stranou, napríklad:
  - meranie času potrebného na to, aby nový vývojár začal používať API (tzv. Time To First Hello World, alebo TTFHW),
  - spokojnosť vývojárov a pravdepodobnosť odporučenia API svojím kolegom (tzv. Net Promoter Score).

  Tieto metriky sú využívané súkromným sektorom aj vládami, najmä s cieľom zmerať prijatie API alebo zdieľaním príkladov toho, ako sú API používané externými zainteresovanými stranami.

Vlády môžu využívať tieto tri typy metrík, avšak zároveň sa musia zamyslieť aj nad možnými negatívnymi dopadmi API, napríklad ak sú API poskytované bezplatne všetkým používateľom, API tím sa musí ubezpečiť, že neúmyselne neposkytuje výhodu veľkým technologickým gigantom na úkor malých a stredných podnikov. Monitorovanie potenciálneho negatívneho vplyvu API je podstatnou súčasťou vládnej politiky v oblasti merania vplyvov.

Vlády vykonávajú meranie API po výkonnostnej a technickej stránke, čo by malo byť zároveň súčasťou životného cyklu API. Za účelom merania vplyvu API na politiky, by vlády mohli prehodnotiť procesy hodnotenia vplyvu, ako súčasť počiatočnej analýzy zameranej na ciele, ktoré môžu byť dosiahnuté pomocou API (opatrenie č. 1 vyššie). Tieto hodnotenia vplyvu možno použiť na identifikáciu potenciálnych ukazovateľov, ktoré by sa mali merať pravidelne, aby bolo možné pochopiť, aký vplyv bude mať zavedenie API. Vlády by tiež mali identifikovať priority a prístupy k meraniu (často je vhodné začať s jednou/dvoma metrikami a postupne vylepšovať spôsob, akým sú dáta zbierané, zaznamenávané a analyzované v priebehu času).

10. Vytvorenie API komponentov

Zdokumentovanie API infraštruktúry a vydanie spoločných usmernení pre navrhovanie API pomáha vytvárať odolnú a robustnú API infraštruktúru (takýto prístup je bežný aj v súkromnom sektore aj vo verejnom sektore).

Tento projekt je podporený z Európskeho sociálneho fondu.

Nedostatok asertívnych vyhlásení nabádajúcich k zavedeniu API a API architektonických štýlov (ako je napr. REST) všade tam, kde je to možné, spomaľuje úsilie o prechod na poskytovanie digitálnych vládnych služieb. Preto už napríklad nová smernica o otvorených dátach a opakovanom použití informácií verejného sektora poukazuje na strategický cieľ spočívajúci v poskytovaní dynamických údajov a súborov údajov s vysokou hodnotou prostredníctvom API.

Podľa osvedčených postupov („best practices“) by sa pri tvorbe API malo zvážiť ich vytvorenie pomocou webových štandardov ako je REST (najviac využívaný). Vlády vytvárajú usmernenia, ktoré dokumentujú interné postupy a štandardy pre tvorbu API – tieto štandardy často navrhujú architektonický štýl REST.

Príklad: Estónska vláda preorientovala svoju infraštruktúru smerom k REST API s cieľom zabezpečiť väčšiu hodnotu z ich „API-first“ prístupu. Od roku 2017 vládna stratégia v Taliansku zahŕňa napríklad:

- REST API zapísané prostredníctvom open API špecifikácie,
- trhom riadené štandardy,
- iteratívne upgrady a verziovanie vládnych API,
- verejný API katalóg.

Model v Taliansku zostáva bimodálny, čím umožňuje existujúcim SOAP službám zostať v prevádzke.

11. Vymenovanie produktových manažérov a vytvorenie API tímov

Produktoví manažéri, resp. vlastníci API sú potrební na to, aby sa zabezpečilo priebežné spravovanie API v rámci zdrojov vlády. Títo vlastníci sú zodpovední za to, že API sú použiteľné, prístupné a že sú v súlade s cieľmi príslušných politík. Produktoví manažéri sú zvyčajne zodpovední aj za identifikáciu potenciálnych zlepšení API a napomáhajú uľahčeniu diskusie medzi používateľmi a vývojármi zodpovednými za vytváranie a riadenie technických aspektov API. V súkromnom aj verejnom sektore sa kladie dôraz na určenie vlastníka každého API, aby skutočne bol zabezpečený súlad API s očakávaným prínosom.

V súčasnosti nie je definovaná optimálna organizačná štruktúra pre vládne organizácie, ktorá by reflektovala tento prístup. Niektoré vlády uplatňujú využitie jedného vlastníka pre viacero API, ktorý je produktovým manažérom na úrovni príslušnej služby. Aktuálne nie je dostupný príklad jednoznačného všeobecného uplatniteľného rozhodnutia o tom, či prístup vymenovania produktového manažéra API má striktne vyžadovať, aby pre každé API bol vymenovaný individuálny vlastník, alebo či sa má uprednostniť model produktového manažéra služby, ktorý je vlastníkom viacerých API (súvisiacich s konkrétnou digitálnou službou). V každom prípade sa však vyžaduje, aby oddelenie poskytujúce API identifikovalo pre toto API zamestnanca, ktorý bude vystupovať v roli

Tento projekt je podporený z Európskeho sociálneho fondu.

produktového manažéra, resp. vlastníka (na to musí nadväzovať aj popis jeho činností a zodpovedností, úloh a ukazovateľov výkonnosti).

Po definovaní a prijatí programu k zavedeniu API na úrovni oddelenia musia byť alokované zdroje potrebné pre všetky činnosti súvisiace s vytvorením, zavedením a správou API – kapacity zamestnancov oddelenia musia byť pridelené k roli technického vedúceho a členov technických tímov.

Zodpovednosť za efektívnosť obmedzených zdrojov alokovaných za účelom vytvorenia funkčných API, ktoré sú v súlade s požiadavkami na bezpečnosť a ochranu osobných údajov má produktový manažér API. Produktový manažér komunikuje s technickým vedúcim ako aj s tvorcami politík, aby mohol identifikovať prípady použitia a potrebu nových funkcionalít a zároveň monitoruje, či hodnota vytváraná prostredníctvom API je v súlade s cieľmi politík a či nedochádza k nežiadúcim negatívnym vplyvom na spoločnosť.

Príklad: Usmernenie Spojeného kráľovstva k dokumentovaniu API rozhraní poskytuje kontrolný zoznam osvedčených postupov, ktoré môže produktový manažér použiť pri dokumentovaní API a zabezpečovaní jednoduchosti používania API pre jeho používateľov. Už skôr spomínaný fínsky projekt 6Aika sa snaží definovať odporúčania zamerané na API dokumentáciu ako na kľúčový aspekt produktového riadenia API, podľa ktorých dobrá API dokumentácia by mala poskytovať odpovede na nasledujúce otázky:

- Čo môžem (a čo nemôžem) robiť s vaším API?
- Zodpovedá vaše API potrebám mojej spoločnosti?
- Ako vaše API vníma môj svet?
- Ako je vaše API zabezpečené?
- Ako dlho trvá, kým začnem API používať?
- Poskytujete SDK[7]?
- Aké koncové body a integrácie ponúka vaše API?
- Prečo sa mi zobrazuje tento chybový kód alebo neočakávanú odpoveď?

12. Osvojenie prístupu orientovaného na životný cyklus API

Prístup orientovaný na životný cyklus API zabezpečí, že API sú dobre navrhnuté, sú v súlade s cieľmi politík a potrebami organizácie, ich funkčnosť bola otestovaná, fungujú podľa očakávaní, sú bezpečné a efektívne. Takýto prístup predstavuje udržateľný a efektívny spôsob navrhovania, vytvárania a správy API či už v súkromnom alebo verejnom sektore.

Riadenie životného cyklu API by malo obsahovať tieto aspekty:

[7] Tzv. Software Development Kit – súbor nástrojov pre vývoj poskytovaný vlastníkom API (ide o podpornú technickú dokumentáciu pre vývojárov, ktorá napríklad môže obsahovať ukážkový kód)

Tento projekt je podporený z Európskeho sociálneho fondu.

- stratégia,
- dizajn,
- dokumentácia,
- vývoj,
- testovanie,
- nasadenie,
- bezpečnosť,
- monitorovanie,
- riadenie zmien.

Podľa osvedčených postupov viacerých vlád by mali byť realizované nasledovné aktivity

- definovanie vhodných politík pre autentifikácie, autorizácie a sprístupňovanie API,
- vytvorenie metadátových špecifikácií pre všetky API (kde je to možné, mali by sa použiť uznávané štandardy dokumentácie ako OAS vrátane popísania účelu a prípadov použitia ešte pred technickým návrhom),
- vytvorenie návodu pre navrhovanie API (tzv. API design guidelines),
- zavedenie kontrol kybernetickej bezpečnosti a dodržiavania ochrany osobných údajov,
- zabezpečenie nástrojov na návrh životného cyklu API vrátane stanovenia štandardov organizácie tak, aby akýkoľvek návrh, ktorý nie je v súlade s týmito štandardmi bol signalizovaný v čase vývoja,
- prijatie agilných metód vývoja uplatňovaných v rámci API tímov (ak je to vhodné),
- definovanie testovacích procesov a DevOps procesov, uplatňovanie neustáleho vývoja a neustáleho procesu zlepšovania,
- zabezpečenie monitorovania používania a výkonu API.

Príklad: Vláda v štáte Viktória vytvorila vládne štandardy pre navrhovanie API, ktoré obsahujú osvedčené postupy pre ministerstvá pri tvorbe a správe API. Holandské usmernenia k API stratégii obsahujú odporúčania týkajúce sa navrhovania API a prístupu orientovaného na životný cyklus API.

Rámec pre API vo verejnom sektore bol vytvorený na základe analýzy rozsiahlej dostupnej literatúry týkajúcej sa zavádzania API vo vládnom prostredí, vďaka čomu zároveň predstavuje prehľad vývoja a trendov v oblasti štandardizácie API.

Najväčší prínos rámca spočíva v tom, že vzhľadom na rastúci význam API (konkrétne API poskytovaných verejnou správou) poukazuje na nevyhnutnosť toho, aby takéto API boli poskytované jednotným (štandardizovaným) spôsobom, ktorý je v súlade s politikami stanovenými na celoštátnej úrovni a zároveň ponúka príklady z viacerých krajín.

2.2 Potreba štandardizácie strojovej spracovateľnosti a API posilnená zavedením konceptu súborov údajov s vysokou hodnotou

Trend strojovej spracovateľnosti a poskytovania a štandardizovaného dokumentovania API rozhraní je ešte viac posilnený zavedením konceptu súborov údajov s vysokou hodnotou (tzv. High Value Datasets, alebo v skratke HVD) na úrovni Európskej únie. Oblasť súborov údajov s vysokou hodnotou je upravená smernicou[8] a vykonávacím nariadením[9], podľa ktorých súbory údajov s vysokou hodnotou sú:

- dostupné bezplatne,
- strojovo čitateľné,
- poskytované prostredníctvom API,
- v príslušných prípadoch poskytované na hromadné stiahnutie.

Súbory údajov s vysokou hodnotou sú vymedzené ako dokumenty, ktorých opakované použitie sa spája s významnými prínosmi pre spoločnosť, životné prostredie a hospodárstvo, najmä preto, že sú vhodné na tvorbu služieb s pridanou hodnotou, aplikácií a nových, vysoko kvalitných a dôstojných pracovných miest, ako aj vzhľadom na počet osôb, ktoré môžu využívať prínosy týchto služieb a aplikácií s pridanou hodnotou založených na týchto súboroch údajov. Európska legislatíva prostredníctvom vykonávacieho nariadenia vytvorila zoznam súborov údajov s vysokou hodnotou, ktoré sú rozdelené do šiestich tematických kategórií:

- geopriestorové údaje,
- pozorovanie Zeme a životné prostredie,
- meteorológia,
- štatistika,
- spoločnosti a vlastníctvo spoločností,
- mobilita.

Vytvorenie zoznamu súborov údajov s vysokou hodnotou neznamená pre členské štáty potrebu tvorby nových súborov údajov, pretože ide o už existujúce súbory údajov, ktoré tým, že boli označené za súbory údajov s vysokou hodnotou, musia spĺňať vyššie uvedené požiadavky na dostupnosť, strojovú spracovateľnosť a prístup prostredníctvom API (alebo aj hromadného stiahnutia).

Koncept súborov údajov s vysokou hodnotou má byť v členských štátoch implementovaný do júna 2024, pričom následne sú členské štáty povinné poskytnúť Komisii správu o vykonaných opatreniach za účelom implementácie.

[8] Smernica EÚ 2019/1024 o otvorených dátach a opakovanom použití informácií verejného sektora EUR-Lex - 32019L1024 - EN - EUR-Lex (europa.eu)
[9] Vykonávacie nariadenie Komisie 2023/138, ktorým sa stanovuje zoznam konkrétnych súborov údajov s vysokou hodnotou a podmienky ich uverejňovania a opakovaného použitia EUR-Lex - 32023R0138 - EN - EUR-Lex (europa.eu)

Tento projekt je podporený z Európskeho sociálneho fondu.

Okrem toho, že samotné súbory údajov s vysokou hodnotou majú byť strojovo čitateľné a poskytované prostredníctvom API, podobné požiadavky sú kladené aj na API dokumentáciu. Subjekty, ktoré majú v držbe súbory údajov s vysokou hodnotou budú povinné aj stanoviť a zverejňovať podmienky používania rozhrania API a kritériá kvality služby týkajúce sa výkonu, kapacity a dostupnosti. Podmienky používania API majú byť v zmysle nariadenia dostupné v ľudským okom čitateľnom a strojovo čitateľnom formáte, pričom priložená má byť dokumentácia API rozhrania (v otvorenom, ľudským okom čitateľnom a strojovo čitateľnom formáte uznávanom v EÚ alebo medzinárodne).

Ako príklad možno uviesť Francúzsko, ktoré v rámci svojej centrálnej platformy pre otvorené dáta (data.gouv.fr) definovalo a zverejnilo podmienky používania API, vďaka čomu sú dostupné vo formátoch čitateľných ľudským okom aj strojovo. Zároveň stanovili kritériá kvality služieb na zabezpečenie výkonu, kapacity a dostupnosti API[10].

Okrem vyššie uvedeného sa v zmysle smernice a nariadenia vyžaduje aj určenie kontaktného bodu pre otázky a záležitosti týkajúce sa API rozhrania a súbory údajov s vysokou hodnotou majú prostredníctvom opisu v metaúdajoch obsahovať špecifické označenie, podľa ktorého je možné identifikovať, že ide o súbor údajov s vysokou hodnotou.

Podľa smernice tam, kde je to možné, by sa mali používať open API. Zároveň zriadenie a využívanie API musí byť založené na niekoľkých zásadách:

- dostupnosť,
- stabilita,
- udržiavanie počas životného cyklu,
- jednotnosť používania a štandardov,
- ľahká použiteľnosť,
- bezpečnosť.

Vyžadovať sa bude podávanie správy o vykonaných opatreniach za účelom implementácie konceptu súborov údajov s vysokou hodnotou. Správa, ktorú budú podávať členské štáty Komisii má okrem iného obsahovať aj trvalý odkaz na rozhranie API zabezpečujúce prístup k súborom údajov s vysokou hodnotou, pričom Komisia bude posudzovať aj dostupnosť a využívanie API.

Členské štáty vrátane Slovenska sú tak v súčasnosti (do júna 2024) povinné ešte intenzívnejšie sa venovať strojovej spracovateľnosti a prístupu k údajom, ktoré majú v držbe aj z hľadiska implementácie konceptu súborov údajov s vysokou hodnotou.

2.3 GDPR a súvisiace trendy

V neustále sa rozvíjajúcom digitálnom prostredí je pokračujúcim trendom kladenie dôrazu na zabezpečenie ochrany osobných údajov. V podmienkach Európskej únie v roku 2018 nadobudlo účinnosť nariadenie o ochrane fyzických osôb pri spracúvaní

[10] https://doc.data.gouv.fr/api/intro/

Tento projekt je podporený z Európskeho sociálneho fondu.

osobných údajov (tzv. GDPR)[11], podľa ktorého rýchly technologický rozvoj a globalizácia so sebou priniesli nové výzvy v oblasti ochrany osobných údajov. Rozsah získavania a zdieľania osobných údajov sa výrazne zväčšil. Technológia umožňuje súkromným spoločnostiam a orgánom verejnej moci pri výkone ich činností využívať osobné údaje v doteraz bezprecedentnom rozsahu. Fyzické osoby stále viac zverejňujú svoje osobné údaje, a to aj v globálnom meradle. Technológia transformovala hospodársky aj spoločenský život, pričom tento vývoj si vyžaduje silný a súdržnejší rámec ochrany údajov v EÚ, ktorý sa bude intenzívne presadzovať vzhľadom na význam vybudovania dôvery, ktorá umožní rozvoj digitálnej ekonomiky v rámci vnútorného trhu. Fyzické osoby by mali mať kontrolu nad svojimi vlastnými osobnými údajmi.

GDPR posilňuje postavenie jednotlivca vo vzťahu k spracúvaniu jeho osobných údajov prostredníctvom:

- práva na prístup k údajom,
- práva na opravu alebo vymazanie (právo “na zabudnutie”),
- práva na obmedzenie spracúvania,
- práva na prenosnosť údajov,
- práva namietať a pod.

Cieľom tejto kapitoly nie je detailný popis a analýza GDPR, ale skôr načrtnutie toho, ako aj tento trend ovplyvnil poskytovanie digitálnych služieb verejným sektorom, pretože GDPR poskytuje väčšiu kontrolu aj nad údajmi vedenými v databázach informačných systémov a zároveň za osobný údaj považuje aj online identifikátory ako napr. cookies (ak existuje možnosť identifikácie alebo vyčlenenia jednotlivca).

V súlade s GDPR a s európskou stratégiou pre dáta[12], ako aj cieľmi uvedenými na rok 2030 v rámci digitálnej dekády[13] dochádza k snahe o vytváranie takých riešení, ktoré budú jednotlivcom poskytovať informácie o tom, kto a za akým účelom spracúva ich osobné údaje, pričom im umožnia aktívne participovať na zdieľaní svojich údajov s tretími stranami (poskytnutím súhlasu). Jedným z prínosov je zároveň väčšia kontrola jednotlivcov nad ich osobnými údajmi.

Konceptom smerujúcim k posilneniu práv jednotlivca v tejto oblasti je napríklad koncept MyData[14]. Príkladom riešenia založenom na zásadách MyData môže byť riešenie „The Blue Button“[15] v Holandsku:

The Blue Button rieši problémy spojené pri procese oddlženia, kedy dlžník potrebuje zhromaždiť svoje údaje o dlhoch spolu s ďalšími finančnými a osobnými údajmi. Častokrát ľudia nemali priamy prístup k takýmto údajom. Navyše veľa dlžníkov ani nevedelo, kde všade majú dlhy (v ktorých všetkých inštitúciách). Výsledkom bolo, že

[11] Nariadenie EP a Rady 2016/679 o ochrane fyzických osôb pri spracúvaní osobných údajov a o voľnom pohybe takýchto údajov EUR-Lex - 32016R0679 - EN - EUR-Lex (europa.eu)
[12] A European Strategy for data | Shaping Europe’s digital future (europa.eu)
[13] Europe’s Digital Decade: digital targets for 2030 (europa.eu)
[14] About (mydata.org)
[15] Use Case - The Blue Button (Blauwe Knop) by The Kloosterhoeveberaad

Tento projekt je podporený z Európskeho sociálneho fondu.

z dôvodu problémov pri snahe o získanie všetkých týchto údajov, dlžníci nedostávali relevantnú pomoc, ktorú v ich vážnej situácii potrebovali.

The Blue Button je tlačidlo na webovej stránke príslušnej organizácie, ktoré umožní jednotlivcovi stiahnuť jeho vlastný prehľad o osobných dátach, ktoré daná organizácia o ňom spracováva (napr. dlh).

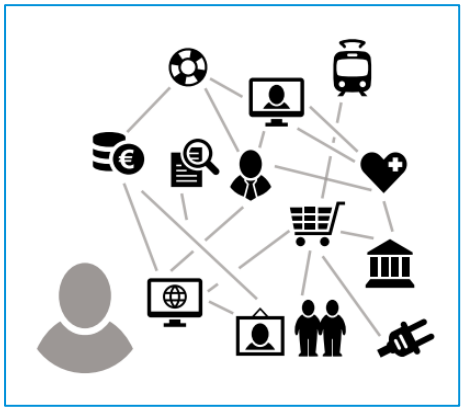
Princípy MyData, na ktorých je postavené riešenie The Blue Button:

- kontrola jednotlivca nad osobnými údajmi,
- jednotlivec ako bod integrácie,
- posilnenie jednotlivca,
- prenosnosť (prístup a opätovné použitie),
- transparentnosť a zodpovednosť,
- interoperabilita.

MyData koncept zavádza tzv. model operátora, ktorý je definovaný nižšie na základe porovnania s modelom API ekosystému a platformovým mo.

Model API ekosystému poskytuje určitú mieru kontroly ale nižšiu mieru integrácie, pričom ak sa zvýši počet služieb, počet pripojení sa zvýši ešte rýchlejšie.

Obrázok 1 - Model API ekosystému



Platformový model poskytuje vyššiu mieru integrácie ale nižšiu mieru kontroly, pričom tento model uľahčuje vývoj aplikácií, avšak neexistuje žiadna motivácia pre vlastníkov platforiem smerovať svoju snahu k interoperabilite medzi platformami.

Tento projekt je podporený z Európskeho sociálneho fondu.

Obrázok 2 - Platformový model

Model operátora je robustný a škálovateľný, pretože nie je závislý od poskytovateľov infraštruktúry.

Obrázok 3 - Model operátora

V modeli operátora vystupuje človek ako bod integrácie, pričom osoba kontroluje použitie jej osobných údajov službami prostredníctvom udelenia alebo odmietnutia prístupu.

2.4 Data Governance Act

Relatívne novým a významným trendom je v súčasnosti koncept zdieľania údajov upravený v už schválenom Data Governance Act[16] (DGA) ako súčasť Európskej dátovej stratégie[17] a spoločného európskeho dátového priestoru ako jednotného trhu s údajmi, kde možno údaje využívať bez ohľadu na miesto ich uloženia v rámci Európskej únie. Inovácie založené na údajoch prinesú občanom obrovské výhody, napríklad v podobe lepšej personalizovanej medicíny, novej mobility a pod.

[16] EUR-Lex - 52020PC0767 - EN - EUR-Lex (europa.eu)
[17] European data strategy (europa.eu)

Tento projekt je podporený z Európskeho sociálneho fondu.

Režim opakovaného používania upravený v DGA by sa mal vzťahovať na údaje, ktorých poskytovanie je súčasťou verejných úloh subjektov verejného sektora v zmysle zákona alebo iných záväzných pravidiel členských štátov.

DGA upravuje mechanizmus opakovaného použitia určitých kategórií chránených údajov verejného sektora, ktorý podlieha dodržiavaniu práv iných strán (najmä z dôvodu ochrany osobných údajov, ale aj ochrany práv duševného vlastníctva a obchodného tajomstva). Mechanizmus sa týka údajov, ktoré sú v držbe subjektov verejného sektora, ktoré sú chránené z dôvodu:

- dôvernosti obchodných údajov,
- štatistickej dôvernosti,
- ochrany práv duševného vlastníctva tretích strán,
- ochrany osobných údajov.

Opakované použitie takýchto údajov nepatrí do rozsahu pôsobnosti Smernice 2019/1024 o otvorených dátach a opakovanom použití informácií verejného sektora (pozri podkapitola 2.2). Tento mechanizmus nezakladá právo na použitie takýchto údajov, alestanovuje súbor harmonizovaných základných podmienok, za ktorých možno opakované použitie takýchto údajov povoliť. Subjekty verejného sektora umožňujúce takéto opakované použitie by museli mať technické vybavenie na zaistenie úplného zachovania ochrany údajov, súkromia a dôvernosti. Členské štáty budú musieť zriadiť jednotné kontaktné miesto na podporu výskumníkov a inovačných podnikov pri identifikácii vhodných údajov, ako aj štruktúry na podporu subjektov verejného sektora technickými prostriedkami a právnou pomocou.

Ďalej má byť posilnená dôvera v zdieľanie osobných a iných údajov a má sa dosiahnuť zníženie transakčných nákladov na zdieľanie údajov medzi podnikmi (B2B), ako aj na zdieľanie údajov spotrebiteľov s podnikmi (C2B) vytvorením notifikačného režimu pre poskytovateľov služieb zdieľania údajov. Títo poskytovatelia budú musieť splniť niekoľko požiadaviek, najmä zachovať si neutralitu, pokiaľ ide o vymieňané údaje. Nesmú takéto údaje použiť na iné účely.

Prostredníctvom DGA bude posilnený dátový altruizmus v podobe dobrovoľne sprístupňovaných údajov jednotlivcami alebo firmami pre spoločné dobro – za týmto účelom bude vytvorený európsky formulár súhlasu so spracovaním údajov na altruistické účely. S dobrovoľným sprístupňovaním údajov v zmysle DGA súvisí aj vytvorenie registra uznaných organizácií dátového altruizmu.

Na základe DGA bude vytvorený Európsky dátový inovačný výbor vo forme expertnej skupiny, ktorý má uľahčiť šírenie osvedčených postupov orgánov členských štátov, najmä v oblasti:

- spracovania žiadostí o opakované použitie údajov, na ktoré sa vzťahujú práva iných subjektov,
- zabezpečenie konzistentného používania notifikačného rámca pre poskytovateľov služieb zdieľania údajov,
- dátového altruizmu.

Tento projekt je podporený z Európskeho sociálneho fondu.

2.5 Situácia na Slovensku

Podľa národnej koncepcie informatizácie verejnej správy z roku 2021 (ďalej aj „NKIVS“) v rámci strategickej priority multikanálový prístup budú služby verejnej správy poskytované[18]:

- elektronicky (ústredný portál verejnej správy, špecializovaný portál),
- telefonicky (ústredné kontaktné centrum, iné call centrá subjektov verejnej správy),
- osobne (pracoviská subjektov verejnej správy, klientské centrá, integrované obslužné miesta),
- listinnou formou (podateľne subjektov verejnej správy),
- cez čet (nasadenie modulu okamžitých správ poskytujúceho interaktívnu komunikáciu a navigáciu občana aj mimo pracovných hodín),
- mobilným zariadením (Slovensko v mobile),
- tretími stranami (centrálna API manažment platforma pre publikovanie služieb verejnej správy cez open API, ktorá sprístupní aplikačné programové rozhranie tretím stranám).

Východiskami pre zaradenie takto definovaného multikanálového prístupu do NKIVS je okrem iného aj skutočnosť, že elektronické služby verejnej správy zatiaľ nemajú systémovo prístupné open API pre tretie strany a nie sú široko dostupné z mobilných zariadení[19].

NKIVS v rámci svojich princípov uvádza aj prednostné využívanie digitálnych služieb a dát pre rozhodovanie vo verejnej správe, prednostné používanie aplikačných rozhraní (API-first) a jednoduchú prístupnosť služieb vrátane budovania služieb s prepojením na životné situácie.

Cieľom do roku 2026 je napríklad zvýšiť počet komplexných životných situácií (z atomických na bezproblémové zreťazené) a vybavených plne elektronicky na 80%. Na rovnakú hodnotu (80%) by sa mal dostať aj podiel sledovaných elektronických služieb, ktoré publikujú open API cez API gateway z celkového počtu sledovaných elektronických služieb pre strojovú komunikáciu navonok a dovnútra verejnej správy.

2.5.1 Publikovanie elektronických služieb do multikanálového prostredia

Pre všetky agendové systémy, poskytujúce elektronické služby verejnej správy, a ich správcov, existuje povinnosť publikovať (formou web služieb) služby pre spracovanie elektronických podaní, a zároveň všetky súvisiace (pomocné) služby pre úspešné vyplnenie a prípravu podania (napr. doťahovanie údajov, validácia vstupov) do multikanálového prostredia (publikácia na API Gateway)[20].

[18] NIKVS | Vicepremier (gov.sk)
[19] Tamže, s. 20
[20] Pravidla_Publikovania_Sluzieb_v1_0-1.pdf (gov.sk)

Tento projekt je podporený z Európskeho sociálneho fondu.

Pravidlá publikovania elektronických služieb do multikanálového prostredia verejnej správy vyžadujú:

- aby všetky elektronické služby verejnej správy aplikovali šifrovanie už na úrovni transportnej vrstvy, čo v prípade základného protokolu znamená HTTPS optimálne s nadstavbou HSTS (HTTP Strict Transport Security);

  Dôvodom je, že elektronické služby zvyčajne zbierajú citlivé údaje o používateľoch, pričom je potrebné zabezpečiť, aby si tieto údaje neprečítal niekto nepovolaný počas ich prenosu z internetového prehliadača používateľa smerom na server, ktorý elektronickú službu publikuje.

- pre publikovanie služieb do multikanálového prostredia bude podporovaný prístup na báze REST služieb;

  Starším prístupom, ktorý bol využívaný je SOAP (používajúci XML formát pri výmene správ), avšak jeho použiteľnosť je oproti REST prístupu nižšia. REST prístup, využívajúci na výmenu správ formát JSON poskytuje širšiu podporu zariadení (aplikácie v smartfónoch, tabletoch) a jednoduchšie aplikácie autorizačných a autentifikačných protokolov.

- používanie štandardu Unicode Transformation Format (UTF-8) pre kódovanie všetkých dotazov a odpovedí v rámci služieb v celom prostredí eGovernmentu;

- využitie štandardu JSON ako formátu pre definovanie obsahu dotazov a odpovedí pre všetky nové publikované REST služby. (Pre publikáciu existujúcich služieb, ktoré sa nebudú v dohľadnej dobe meniť, je možné použiť štandard XML.);

- všetky nové publikované REST služby (do multikanálového prostredia eGov) budú popísané pomocou štandardu OpenAPI 3.0 (a vyšším). Tieto definičné súbory budú publikované na MetaIS, odkiaľ ich bude možné (automatizovane) nasadzovať na API gateway;

- pre autorizáciu a autentifikáciu (ak to kontext služby vyžaduje) je potrebné podporovať štandardy Oauth2 a OpenIDConnect, pričom centrálnu implementáciu autorizačného servera bude poskytovať centrálny komponent API gateway;

- centralizáciu publikovania API v jednom bode v podobe API gateway (ako vstupnej bráne k službám eGovernmentu), ktorá zároveň podporuje manažment celého životného cyklu API (verzionovanie, testovanie, podpora pre proces sprístupňovania API vlastníkom pre konzumentov, podpora autorizácie prístupu na služby v mene občana a podnikateľa a pod.).

Tento projekt je podporený z Európskeho sociálneho fondu.

2.5.2 Súbory údajov s vysokou hodnotou

Smernica o otvorených dátach a opakovanom použití verejného sektora bola do slovenského právneho poriadku transponovaná prostredníctvom infozákona[21], podľa ktorého povinná osoba sprístupňuje informácie na účely ich opakovaného použitia v podobe a spôsobom, ktoré umožňujú jej technické podmienky, prednostne v elektronickej podobe ako otvorené údaje umožňujúce automatizované spracovanie spolu s ich metaúdajmi. Formáty a metaúdaje musia v čo najväčšom rozsahu spĺňať formálne otvorené štandardy. Zároveň je povinná sprístupňovať dynamické údaje na účel ich opakovaného použitia bezodkladne po ich vzniku prostredníctvom aplikačného programového rozhrania a ak je to vhodné, prostredníctvom ich hromadného stiahnutia.

Ďalej sa podľa tohto zákona súbor údajov s vysokou hodnotou sprístupňuje v súlade s vykonávacím nariadením Európskej únie ustanovujúcim zoznam konkrétnych súborov s vysokou hodnotou na účel opakovaného použitia v strojovo spracovateľnom formáte prostredníctvom aplikačného programovacieho rozhrania a ak je to vhodné, prostredníctvom hromadného stiahnutia.

Aktuálne sú realizované aktivity smerujúce k implementácii požiadaviek stanovených európskou legislatívou s cieľom zavedenia konceptu súborov údajov s vysokou hodnotou:

- prebieha identifikácia existujúcich súborov údajov vo vzťahu k zoznamu súborov údajov s vysokou hodnotou uvedených vo vykonávacom nariadení[22],
- na pracovných skupinách sú OVM oboznamované s pojmom súborov údajov s vysokou hodnotou a s doteraz vykonanými aktivitami alebo realizovanými rozhodnutiami (napríklad o harvestovaní súborov údajov prostredníctvom jedného centrálneho bodu – portálu otvorených dát data.gov.sk)[23],
- prebieha definovanie postupu (roadmap) pre implementáciu konceptu súborov údajov s vysokou hodnotou (Výstup č. 5.1.1 Štandardy pre zverejňovanie údajov verejnej správy vo formáte otvorených údajov).

2.5.3 Štandardizácia pre informačné technológie

V podmienkach Slovenska sú štandardy pre informačné technológie verejnej správy vymedzené prostredníctvom vyhlášky[24], ktorá definuje:

- štandardy vzťahujúce sa na technické prostriedky, sieťovú infraštruktúru a programové prostriedky, a to štandardy
  prepojenia,
  prístupu k elektronickým službám,

[21] Zákon č. 211/200 o slobodnom prístupe k informáciám 211/2000 Z.z. - Zákon o slobodnom prístupe k inform... - SLOV-LEX
[22] Zoznam HVD datasetov - Metodika pre otvorené údaje (opendata.gov.sk) - Confluence
[23] PS 15. 03. 2023 - YouTube
[24] Vyhláška č. 78/2020 Z.z. o štandardoch pre informačné technológie verejnej správy

Tento projekt je podporený z Európskeho sociálneho fondu.

webových služieb,
integrácie dát,

- štandardy prístupnosti a funkčnosti webových sídiel a mobilných aplikácií vzťahujúce sa na aplikačné programové vybavenie podľa zákona,
- štandardy použitia súborov vzťahujúce sa na formáty výmeny údajov,
- štandardy názvoslovia elektronických služieb vzťahujúce sa na sieťovú infraštruktúru,
- dátové štandardy vzťahujúce sa na údaje, registre a číselníky,
- štandardy elektronických služieb verejnej správy vzťahujúce sa na údaje, registre, číselníky a aplikačné programové vybavenie podľa zákona,
- štandardy poskytovania údajov v elektronickom prostredí vzťahujúce sa na databázové prostredie, spoločné moduly, aplikačné programové vybavenie, údaje, registre, číselníky a formáty výmeny údajov,
- štandardy poskytovania cloud computingu a využívania cloudových služieb vzťahujúce sa na technické prostriedky a programové prostriedky,
- štandardy formátov elektronických dokumentov podpísateľných elektronickým podpisom,
- štandardy základných číselníkov.

Vyhláška upravuje požiadavky na strojovú spracovateľnosť, pričom základom je vymedzenie pojmov ako centrálny model údajov, ontológia, prepojené údaje a pod. Zároveň z pohľadu otvorenosti dát vyhláška pracuje s tzv. päťhviezdičkovou schémou[25], ktorá je vo vyhláške vymedzená stanovením úrovne kvality poskytovaného datasetu nasledovne:

- úroveň 0, pri ktorej nie je dataset poskytovaný v elektronickej podobe,
- úroveň 1, pri ktorej je dataset dostupný vo webovom prostredí,
- úroveň 2, pri ktorej je splnená požiadavka uvedená v predchádzajúcej odrážke a obsah datasetu je štruktúrovaný tak, že umožňuje automatizované spracovanie,
- úroveň 3, pri ktorej sú splnené požiadavky uvedené v predchádzajúcej odrážke a dataset je poskytovaný v otvorenom formáte, nezávislom na konkrétnom proprietárnom softvéri,
- úroveň 4, pri ktorej sú splnené požiadavky uvedené v predchádzajúcej odrážke a na identifikáciu údajov datasetu a ich vzťahov sa používajú referencovateľné identifikátory,
- úroveň 5, pri ktorej sú splnené požiadavky uvedené v predchádzajúcej odrážke a dataset a jeho interné a externé vzťahy sú prepájané prostredníctvom referencovateľných identifikátorov a sú opísané prostredníctvom Centrálneho modelu údajov.

Pre automatizované spracovanie je podľa vyhlášky požadovaná kvalita najmenej úrovne 3. Významným predpokladom strojovej spracovateľnosti a automatizovaného spracovania údajov je definovanie dátových prvkov verejnej správy v štruktúrovanej, prepojenej a strojovo spracovateľnej podobe. Toto je v podmienkach Slovenskej republiky okrem iného podporené centrálnym modelom údajov, ktorý predstavuje množinu ontológií zverejnenú v centrálnom metainformačnom systéme, ktorá sa používa

[25] 5-star Open Data (5stardata.info)

opise dátových prvkov verejnej správy a ktorá je vyjadrením sémantických vzťahov medzi dátovými prvkami, vyjadrenými prostredníctvom jednotných referencovateľných identifikátorov. Aktuálne centrálny model údajov verejnej správy obsahuje tieto ontológie[26]:

- ontológia fyzickej osoby,
- ontológia právneho subjektu,
- ontológia finančných entít,
- ontológia egovernment entít,
- ontológia lokácie,
- medzinárodné ontológie.

Vyhláška pracuje s pojmom „verejne dostupné aplikačné rozhrania“, ktorými sú aplikačné rozhrania dostupné komukoľvek po splnení ustanovených podmienok, ktoré umožňujú používať elektronickú službu pomocou vlastných softvérových aplikácií alebo aplikácií tretích strán.

Štandardom verejne dostupného aplikačného rozhrania je

- poskytovanie verejne dostupného aplikačného rozhrania elektronických služieb potrebných na výkon verejnej moci elektronicky,
- poskytovanie kvality elektronickej služby a podpory pre elektronickú službu na rovnakej úrovni ako pre orgány riadenia,
- zverejnenie úplných podmienok prevádzky a používania verejného aplikačného rozhrania v centrálnom metainformačnom systéme,
- poskytovanie úplnej štruktúrovanej dokumentácie rozhrania v centrálnom metainformačnom systéme, a to v štruktúre údajov zverejnenej v centrálnom metainformačnom systéme,
- umožnenie udelenia a odobratia oprávnenia na delegovaný prístup k elektronickým službám pre aplikácie poskytované tretími stranami; oprávnenia udeľuje autentifikovaný používateľ,
- umožnenie časového ohraničenia delegovaného prístupu a k údajom používateľa, ak sa možnosť časového ohraničenia poskytuje,
- zabezpečenie oprávnení pre prístup k elektronickej službe (podľa piatej odrážky vyššie) prostredníctvom centrálneho modulu alebo iným spôsobom po dohode s orgánom vedenia.
- registrácia elektronickej služby v centrálnom metainformačnom systéme,
- poskytovanie testovacieho verejného aplikačného rozhrania.

Okrem iného sa v rámci štandardov uvedených vo vyhláške požaduje napríklad:

- pri prenose dát používanie protokolu FTP (File Transfer Protocol) alebo http (Hypertext Transfer Protocol);
- v rámci sieťovej komunikácie používanie protokolu Simple Object Access Protocol (SOAP) najmenej vo verzii 1.2 alebo protokolu Representational State

[26] Centrálny model údajov verejnej správy – DataLab

Transfer (REST) pri komunikácii medzi servermi v rámci jednej správy a komunikácii medzi klientom a serverom;
- špecifikácie OpenAPI Specification najmenej vo verzii 3.0 na definíciu webovej služby pri použití protokolu Representational State Transfer (REST);
- špecifikácie OpenID Connect podľa OpenID Foundation s OAuth2 podľa osobitnej špecifikácie RFC 6749, ak sa pri použití protokolu Representational State Transfer (REST) vyžaduje autentifikácia a určenie rozsahu oprávnení;
- pri výmene dátových prvkov používanie formátu XML, pričom ak je cieľom automatizované spracovanie rozlišujúce význam obsahu dátových prvkov, je možné namiesto XML použiť dátový model RDF opísaný formátmi RDF/XML alebo JSON-LD (pri otvorených údajoch je možné použiť aj formát CSV alebo JSON).

V súvislosti s vyššie uvedenou vyhláškou je potrebné spomenúť aj Komisiu pre štandardizáciu informačných systémov verejnej správy ako poradný a konzultačný orgán MIRRI – platné štandardy sú verejne dostupné a rozdelené do troch kategórií:

- sémantická interoperabilita,
- technická interoperabilita,
- právna interoperabilita.

Sémantická interoperabilita - Publikačné minimum

Príkladom premietnutia vyhlášky do praxe je napríklad publikačné minimum orgánu štátnej správy a samosprávy (ako súčasť existujúcich štandardov v rámci sémantickej interoperability), ktoré je definované prostredníctvom zoznamu zverejňovaných datasetov.

Jedným z týchto datasetov je dataset Faktúry, ktorý je dostupný na stiahnutie vo formátoch CSV, XML, JSON (resp. aj prostredníctvom služby API) v Centrálnom ekonomickom systéme (CES) pre dátových kurátorov (resp. iné oprávnené osoby) v rámci OVM napojených na CES[27]. Zároveň je k tomuto datasetu vytvorený dátový model, ktorý je v súlade s centrálnym modelom údajov.

Technická interoperabilita - Proces pre open API popísaný na znalosti.gov

V rámci technickej interoperability je na stránke wiki.vicepremier.gov.sk venovaná jedna karta téme open API, konkrétne sú tam popísané napríklad nasledovné informácie:

- definícia a základné požiadavky:
    - open API - reprezentácia elektronickej služby verejnej správy prístupná verejnosti aj prostredníctvom tretích strán,
    - podmienenosť použitia prihlásením do centrálnej autority identít (NASES)
    - prístup k službe prostredníctvom API gateway,

[27] Štandardy ISVS (gov.sk)

Tento projekt je podporený z Európskeho sociálneho fondu.

- preferencia OpenID Connect pre bezpečné poskytovanie identity používateľov a správy prístupov k službe,

- proces registrácie open API služby,
- proces povolenia prístupu k open API službe pre tretie strany,
- proces použitia open API služby v aplikácii tretej strany.

Karta venovaná open API bola naposledy aktualizovaná v roku 2017.

2.5.4 Prebiehajúce projekty

Cieľom tejto kapitoly je uviesť vybrané prebiehajúce projekty, ktoré sú na Slovensku realizované a ktoré priamo nadväzujú na skôr uvedené trendy vo vývoji strojovej spracovateľnosti a open API.

2.5.4.1 CAMP

MIRRI plánuje do roku 2023 (do ukončenia OP II) prostredníctvom projektu Centrálna API Manažment Platforma (CAMP) nasadiť centrálnu API manažment platformu, ktorá sprístupní aplikačné programové rozhrania tretím stranám pre služby verejnej správy s grafickým používateľským rozhraním[28].

Snaha o vytvorenie takejto platformy bola odpoveďou na nežiaduci stav ústredného portálu verejnej správy poskytujúceho elektronické služby, pričom ako problematické boli definované predovšetkým nasledovné oblasti[29]:

- prezentácia služieb IS VS pre občana/podnikateľa nie je dostatočná a ani atraktívna,
- používateľské prostredie existujúcich služieb nie je komfortné a intuitívne,
- služby sú neprehľadné,
- neexistujú mobilné služby verejnej správy (nie sú podporené štandardy pre komunikáciu v mobilnom svete ako OpenAPI3+),
- aktuálny multikanálový prístup k službám VS nie je dostatočne flexibilný a používateľsky príjemný,
- neexistuje jednotná platforma na publikovanie a správu otvorených rozhraní (open API) agendových IS VS pre budovanie lepších a atraktívnejších služieb vo forme aplikácií komerčným sektorom pre občanov a podnikateľov,
- nie je vynucovaný princíp „API First“ aj z dôvodu komplikovaného publikovania v heterogénnom prostredí IS VS,
- nie sú splnené predpoklady pre dokončenie digitálnej transformácie napriek tomu, že existuje dopyt po prístupe k dátam a službám agendových systémov VS prostredníctvom iných informačných systémov mimo prostredia VS,
- chýbajúci nástroj pre jednotné riadenie životného cyklu API pre potreby poskytovateľov aj konzumentov služieb,

[28] Národná koncepcia informatizácie verejnej správy NIKVS | Vicepremier (gov.sk)
[29] MIRRI_SKIT_CAMP_PID_V1.pdf

Tento projekt je podporený z Európskeho sociálneho fondu.

- nedostatočne aplikované bezpečnostné štandardy a postupy pre sprístupnenie API rozhraní IS VS komerčnému sektoru mimo VS,
- neexistujúca správa a manažment poskytovateľov služieb a rolí konzumentov služieb IS.

Projekt uplatňuje API-first prístup, pričom dôležitou skutočnosťou je, že centrálna API manažment platforma nebude slúžiť iba na publikovanie služieb ale aj na sprístupnenie služieb, ktoré budú využiteľné pre koncového používateľa.

Súčasťou centrálnej API manažment platformy budú nasledovné moduly:

- bezpečnosť (DDoS, OWASPOAuth, SAML, TLS Autentifikácia, prístupové roly obrana voči útokom botmi),
- manažment tretích strán (registrácia a certifikácia, poskytovanie prístupu, vývojársky portál),
- API manažment (registrácia API, orchestrácia API, verzionovanie API),
- monetizácia vo väzbe na Ministerstvo financií a Štátnu pokladnicu(predaj, monitorovanie používania API, zúčtovanie, komunikácia s účtovným systémom MF SR a so Štátnou pokladnicou),
- administrácia API (analýza, audit, zostavy, logovanie, monitorovanie, sledovanie využitia),
- testovanie (publikovanie testovaných služieb, publikovanie autorizácie, príprava a údržba testovacích dát, prevádzka testovacieho prostredia).

Projekt využije aj už existujúce moduly ako napríklad IS CSRÚ (informačný systém centrálnej správy referenčných údajov) alebo MOU.

2.5.4.2 CIP a MOU

V súčasnosti prebiehajú aktivity na národnom projekte Rozvoj platformy integrácie údajov (centrálna integračná platforma – CIP) a manažment osobných údajov (MOU)[30], ktorý je súčasťou reformného zámeru Koncepčné budovanie digitálnej a inovatívnej verejnej správy. Projekt má dve časti[31] – CIP a IS MOU.

CIP

CIP rozširuje a zabezpečuje novú funkčnosť pre potreby IS CSRÚ, IS MOU a konzumentov dátových služieb verejnej správy (napríklad používateľov portálu OverSi.gov.sk) pomocou piatich modulov:

- PaaS pre manažment údajov poskytujúci podporné platformové služby pre inštitúcie verejnej správy riešiace manažment údajov
- distribúcia údajov (priamy push model)

[30] Rozvoj platformy integrácie údajov (CIP) a Manažment osobných údajov | Ministerstvo investícií, regionálneho rozvoja a informatizácie SR (gov.sk)
[31] Projekt: Rozvoj platformy integrácie údajov (centrálna integračná platforma) a Manažment osobných údajov (gov.sk)

Tento projekt je podporený z Európskeho sociálneho fondu.

- obslužná zóna (modul pre riadenie dátovej integrácie, ktorý umožní publikovanie a evidenciu oprávnení pre získavanie údajov, žiadostí pre získavanie údajov pre zníženie administratívnej náročnosti procesu integrácie, monitoring integračných väzieb, zobrazenie metaúdajov a ďalšie služby)
- podporné služby pre poskytovateľov údajov (napríklad vytvorenie zápisovej služby pre komunikáciu zdrojového registra a referenčného registra, vytvorenie služby anonymizácie dát)
- WEB prístup (rozšírenie a vylepšenie GUI pre zobrazenie a prístup k údajom a referenčným údajom)

IS CSRÚ zabezpečuje centrálnu správu referenčných údajov verejnej správy, s napojením rôznych OVM a objektov evidencie. Kvantitatívny rozvoj IS CSRÚ (čo do počtu integrovaných OVM a objektov evidencie je riešený paralelne prebiehajúcim projektom Dátová integrácia[32]).

Oversi.gov.sk je existujúci portál pre poskytovanie výpisov/odpisov vybraných registrov verejnej správy koncovým používateľom, ktorí nemajú vlastný agendový systém integrovaný na IS CSRÚ a potrebujú k týmto údajom pristupovať (v rámci projektu bude portál rozšírený o funkčnosť poskytovania konsolidovaných dát).

MOU

MOU umožní jednotlivcom svoje údaje zdieľať a rozhodovať o ich ďalšom využití a skladá sa z nasledovných častí:

- GDPR,
- logovanie prístupov,
- modul správy osobných údajov,
- modul správy súhlasov.

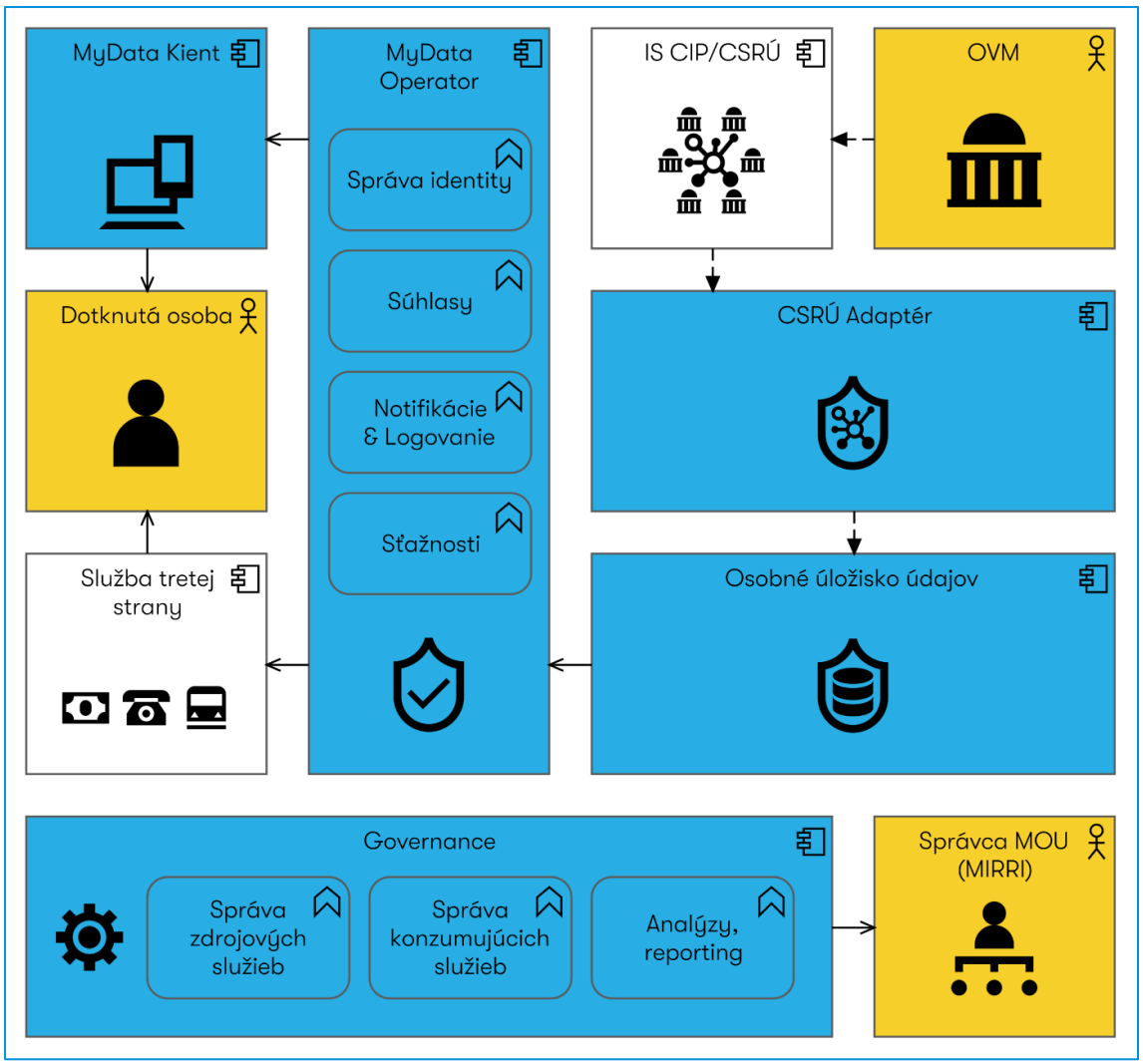
Manažment osobných údajov (MOU) predstavuje nový prístup k digitálnym verejným službám, ktoré sú bezpečnejšie, užitočnejšie a rešpektujú právo občana na súkromie. V rámci projektu MOU ide o vybudovanie národnej infraštruktúry pre podporu dátového hospodárstva založeného na MyData (Moje dáta) princípoch do verejnej správy. Jednotlivci (občania aj podnikatelia) získajú prístup k dátam, ktoré štát o nich eviduje, v strojovo-spracovateľnej podobe.

Výmenu údajov v digitálnom priestore bude riadiť jednotlivec, čím vznikne priestor na nové a inovatívne služby pre súkromný sektor (na základe súhlasu môžu k dátam jednotlivca pristupovať napríklad banky a výrazne zjednodušiť a skvalitniť svoje služby).

[32] Projekt: Dátová integrácia: sprístupnenie údajovej základne VS vrátane otvorených údajov prostredníctvom platformy dátovej integrácie (gov.sk)

Tento projekt je podporený z Európskeho sociálneho fondu.

Obrázok 4 - Koncepčná architektúra riešenia MOU



Na vizualizácii vyššie sú žltou farbou znázornení základní aktéri, modrou farbou komponenty dodávané v rámci projektu MOU a bielou farbou sú znázornené externé komponenty, s ktorými MOU interaguje.

Dotknutá osoba (autentifikovaný používateľ MOU, vlastník účtu MOU) využíva mobilnú alebo webovú aplikáciu MyData Klienta pomocou ktorej pristupuje k službám MOU. V aplikácii vidí svoje údaje, spravuje súhlasy, sleduje využívanie svojich údajov. Dotknutá osoba využíva aj Služby tretích strán – aplikácie tretích strán registrovaných v MOU, ktoré na základe jej súhlasu spracovávajú jej údaje. (napríklad sú to služby banky, ktorá pomocou MOU získava údaje z daňového priznania pri poskytovaní hypotéky).

OVM spravuje a poskytuje osobné údaje. Vystupuje v roli poskytovateľa údajov, je napojený na IS CIP/CSRÚ, ktorý centrálne zabezpečuje komunikáciu medzi MOU a OVM. MOU pomocou CSRÚ Adaptéru komunikuje s IS CIP/CSRÚ, získava osobné

údaje, spravuje notifikácie a podobne (v CSRÚ dochádza aj k transformácii údajov tak, aby boli poskytované v súlade s centrálnym modelom údajov).

Osobné údaje sú ukladané do Osobného úložiska. Osobné úložisko je súčasťou vládneho cloudu, každá dotknutá osoba prihlásená do MOU v ňom má vytvorený svoj vlastný priestor, kde sa ukladajú jej dáta v zašifrovanej podobe tak, že má k nim prístup iba vlastník účtu. Tieto údaje sú, so súhlasom dotknutej osoby, potom poskytované tretím stranám so zaregistrovanými službami.

MyData Operator zabezpečuje správu identít/účtov, registráciu služieb tretích strán a pripojenie služieb tretích strán k používateľovi, správu súhlasov, notifikácie a logovanie, podávanie a sledovanie stavu sťažností a návrhov na opravu osobných údajov zastrešuje správu účtov, identít, súhlasy občana, manažment notifikácií a sťažností.

V rámci MOU riešenia sú využívané nasledovné základné štandardy:

Oauth2.0
Slúži na delegáciu prihlasovania. Tento systém umožňuje poskytnúť obmedzený prístup k používateľskému účtu nachádzajúcemu sa v inej službe. Toto môže byť využité na overenie používateľa. Server inej služby autorizuje používateľa a poskytne túto informáciu službe, ktorá potrebuje overiť používateľa. Používateľ tak môže mať u služby používateľské konto, ale služba nemusí ošetrovať overovanie používateľa v rámci svojej funkcionality.

Oauth 2.0 je v Štandarde Open API pre Moje dáta využívaný na autentifikáciu používateľov.

Open ID Connect
Otvorený štandard implementácie autorizácie prostredníctvom Oauth 2.0. Oauth 2.0 totiž poskytuje veľa možností, akým spôsobom je možné ju implementovať. OpenID je overený, otestovaný a funkčný spôsob jej nasadenia. Zároveň sa jedná o otvorený štandard a môže tak byť použitý bez obmedzení. Autorizácia v OpenID je uskutočňovaná prostredníctvom tokenov, ktoré sú vo formáte JSON Web Token.

Slovenská služba Moje dáta bude tiež využívať štandard OpenID Connect ako implementáciu Oauth 2.0 na autorizáciu používateľov.

JSON Web Token[33]
JavaScript Object Notation (JSON) Web Token (JWT) je otvorený štandard (RFC7519[34]), ktorý predstavuje kompaktný spôsob bezpečného prenosu informácií medzi rôznymi stranami vo forme JSON objektu. Digitálny podpis zabezpečuje overiteľnosť a dôveryhodnosť prenášaných informácií. Podpis tokenu párom verejného a súkromného kľúča zároveň dokladá, že podpis bol skutočne uskutočnený osobou držiacou osobný kľúč. Dva hlavné scenáre využitia JWT sú autorizácia a výmena informácií. Autorizácia je realizovaná tak, že po prihlásení používateľa do systému jeho

[33] https://jwt.io/introduction/
[34] https://tools.ietf.org/html/rfc7519

interakcie obsahujú JWT, ktoré ho jednoznačne identifikujú. Pri výmene informácií je využívaná vlastnosť, že vďaka podpisu je jasne určený skutočný pôvodca a zároveň JSON zaručuje, že obsah správy nebol modifikovaný.

JSON Web Tokeny budú v slovenskej službe Moje dáta ako prostriedok implementácie OpenID Connect určeného na autentifikáciu používateľov a tiež na zabezpečenie dôveryhodnosti prenášaných informácií.

UMA – User Managed Access[35]
Autorizačný protokol vybudovaný nad OAuth 2.0. Umožňuje používateľovi zdieľať svoje dáta s inými používateľmi alebo službami. Môže určiť na ako dlho chce údaje poskytnúť a tiež aj za akých podmienok. Autorizácia je uskutočňovaná prostredníctvom tokenov.

WebID
WebID je spôsob identifikácie identít a internetových služieb na internete. Protokoly založené na WebID (Solid OIDC, WebID-TLS, WebID-TLS + Delegation) ponúkajú nový spôsob prihlasovania sa do internetových služieb. WebID je reprezentovaný identifikátorom URI so schémou HTTP alebo HTTPS, ktorý označuje identitu (osobu, organizáciu, skupinu, zariadenie atď.).

WebID je W3C štandard [https://www.w3.org/2005/Incubator/webid/spec/identity/]

Vo vzťahu k MOU a použitým štandardom je potrebné spomenúť, že v roku 2019 bol vytvorený štandard pre open API pre Moje dáta, ktorý zároveň zohľadňoval štandard MyData Architecture Framework – táto architektúra umožňuje fungovanie tzv. operátorov aj bez osobného úložiska. Jedným z komponentov MOU riešenia je však osobné úložisko (popísané vyššie), pričom sa postupuje podľa princípov riešenia SoLiD[36]. Z tohto dôvodu je nevyhnutné štandard pre open API doplniť práve v súvislosti so zavedením osobného úložiska.

2.5.5 Zákon o údajoch

V súčasnosti prebieha aktualizácia pripravovaného zákona o údajoch na základe podnetov vyplývajúcich z prebiehajúcich projektov a výsledkov predchádzajúceho medzirezortného pripomienkového konania[37].

Zákon bude klásť dôraz okrem iného aj na:

- transformáciu údajov do strojovo spracovateľného formátu (digitálna transformácia údajov),
- umožnenie dispozície s vybranými kategóriami údajov fyzických osôb a právnických osôb osobám, ktorých sa tieto údaje týkajú (personálna elektronická disponibilita údajov).

[35] https://kantarainitiative.org/confluence/display/uma/Home
[36] Home · Solid (solidproject.org)
[37] Legislatívny proces - SLOV-LEX

Tento projekt je podporený z Európskeho sociálneho fondu.

Významným prínosom bude úprava konceptu mojich údajov a zároveň jeho uplatňovanie prostredníctvom API.

2.6 Záver

Súčasný vývoj spoločnosti smeruje k posilneniu digitálnej ekonomiky a s tým súvisiacej digitálnej transformácie vlád. Globálny trh s open API sa zväčšuje výrazným tempom, pričom tento trend sa nevyhnutne musí odzrkadliť aj v prístupe jednotlivých štátov vo vzťahu k interakcii s jednotlivcami (či už fyzickými osobami alebo podnikateľmi). Významná časť týchto interakcií prebieha a bude prebiehať v online prostredí.

Ďalším nemenej významným trendom je snaha o vytvorenie dostatočne kvalitnej dátovej základne umožňujúcej maximálne využitie údajov v držbe orgánov verejnej moci či už na prijímanie rozhodnutí alebo na využitie tretími stranami. Pre využitie potenciálu takýchto údajov sú na vlády kladené požiadavky na prijatie relevantných štandardov podporujúcich strojovú spracovateľnosť a prístup k údajom a službám prostredníctvom open API. Zároveň sa dôraz kladie na dodržiavanie ochrany súkromných údajov a do popredia sa dostávajú práva jednotlivca, ktoré sa premietajú do vyššej angažovanosti o rozhodovaní o tom, ktoré údaje bude zdieľať s vybranými inštitúciami.

Slovensko v porovnaní s vývojom v rámci EÚ a iných krajín nezaostáva, pričom v rámci národnej koncepcie informatizácie[38] sú definované relevantné prioritné osi v podobe:

- lepších služieb (je zameraná na zvyšovanie spokojnosti so službami, zvyšovanie podielu elektronickej komunikácie na jej iných spôsoboch, zjednodušovanie služieb a zavádzanie komplexných životných situácií),
- digitálnej a dátovej transformácie (prinesie postupy, ktoré budú nezávislé na procesoch papierového sveta a budú v plnom rozsahu využívať možnosti digitálnych technológií a zdieľanie údajov),
- efektívnych IT (má skracovať čas na realizáciu projektov, zvyšovať hodnotu nasadených systémov a optimalizovať náklady na prevádzku systémov),
- kybernetickej a informačnej bezpečnosti (má posilňovať ľudské kapacity, minimalizovať bezpečnostné incidenty a škody; a zvyšovať úroveň ekosystému kybernetickej a informačnej bezpečnosti).

Z úrovne Európskej únie badať snahu o zvýšenie využitia potenciálu dostupných súborov údajov zavedením konceptu súborov údajov s vysokou hodnotou, pričom s tým súvisiace aktivity už prebiehajú aj na Slovensku. Zároveň na európskej úrovni môže byť významným pomocníkom Rámec pre API vo verejnom sektore, navrhujúci 12 opatrení pre zavádzanie API, ktoré už sú v menšej či väčšej miere odzrkadlené aj v aktivitách realizovaných v podmienkach Slovenska.

Ako nedostatok v tejto oblasti sa javí absencia usmernení v podobe tzv. best practices pre orgány verejnej moci pri tvorbe, zavádzaní a správe API. V nasledovnej kapitole sú uvedené príklady best practices publikovaných rôznymi entitami (z verejného aj súkromného sektora). Zároveň na Slovensku nie je dostupný

[38] Národná koncepcia informatizácie verejnej správy NIKVS | Vicepremier (gov.sk)

Tento projekt je podporený z Európskeho sociálneho fondu.

nástroj, ktorý by poskytoval jednoznačný prehľad o API poskytovaných verejným sektorom.

Štandardy pre informačné technológie verejnej správy vrátane pravidiel publikovania služieb do multikanálového prostredia odzrkadľujú súčasný vývoj vo vzťahu k definovaniu API prostredníctvom REST štýlu a open API dokumentácie a zároveň podporujú strojovú spracovateľnosť údajov prostredníctvom formátov ako JSON alebo JSON-LD (pre RDF dátový model).

Premietnutie strategických cieľov do reality je vo veľkej miere závislé od úspešnosti prebiehajúcich aktivít ako napríklad projekty CAMP, MOU alebo prijatie zákona o údajoch.

V súčasnosti je potrebné doplniť existujúci štandard open API pre moje dáta, ktorý nezohľadňuje zavedenie osobného úložiska v rámci riešenia MOU.

3 Príklady dobrej praxe aplikácie štandardu

3.1 Open API Iniciatíva – best practices

Open API špecifikácia (Open Api Specification[39] – OAS) je open-source formát na popis a dokumentáciu API rozhraní. Špecifikácia bola pôvodne vyvinutá v roku 2010 za účelom zachovania súladu medzi API dizajnom a dokumentáciou. Táto špecifikácia sa postupne stala štandardom pre navrhovanie a popis REST API, pričom ju používa veľké množstvo vývojárov a organizácií pri vývoj ich API (či už interných alebo externých).

OAS umožňuje vytvorenie open API dokumentu popisujúceho API v strojovo čitateľnom formáte. Na popis sa využíva JSON alebo YAML.

Ukážka JSON syntaxe:

```
{
  "anObject": {
    "aNumber": 42,
    "aString": "Ukazka syntaxe",
    "aBoolean": true,
    "nothing": null,
    "arrayOfNumbers": [
      1,
      2,
      3
    ]
  }
}
```

Ukážka YAML syntaxe:

```
# Anything after a hash sign is a comment
anObject:
  aNumber: 42
  aString: Ukazka syntaxe
  aBoolean: true
  nothing: null
  arrayOfNumbers:
    - 1
    - 2
    - 3
```

Pokiaľ ide o porovnanie JSON a YAML a výber medzi nimi - JSON v podstate nepodporuje vkladanie komentárov (komentáre v YAML sa vkladajú prostredníctvom mriežky) a vyžaduje:

- čiarky oddeľujúce polia,
- zložené zátvorky okolo objektov,
- dvojité úvodzovky okolo reťazcov,

[39] Home - OpenAPI Initiative (openapis.org)

- hranaté zátvorky okolo polí.

Na druhej strane YAML vyžaduje pomlčky pred položkami poľa a vo veľkej miere sa spolieha na odsadenie, ktoré môže byť pri veľkých súboroch ťažkopádne (odsadenie je v JSON úplne voliteľné). YAML sa zvyčajne uprednostňuje kvôli mierne zmenšenej veľkosti súboru.

Open API dokument je JSON objektom, ktorý obsahuje polia zodpovedajúce štruktúre vyžadovanej v open API špecifikácii (OAS). Koreňovým objektom ("root object") pre open API dokument je open API objekt, ktorý má iba dve povinné polia **openapi** a **info**. Okrem toho sa vyžaduje aspoň jedno z polí: **paths**, **components** a **webhooks**.

- **openapi**: Označuje verziu OAS, ktorú tento dokument používa, napr. „3.1.0".
- **info**: Poskytuje všeobecné informácie o API (ako je jeho popis, autor a kontaktné informácie), ale jediné povinné polia sú **title** a **version**.
    - **title**: Ľudsky čitateľný názov rozhrania API, napríklad „Testovacie REST API".
    - **verzia**: Označuje verziu dokumentu API (nezamieňať s verziou OAS vyššie). Relevantné nástroje môžu použiť toto pole na vygenerovanie kódu, ktorý zaistí, že klienti a servery budú interagovať napríklad prostredníctvom rovnakej verzie API.
- **paths**: Popisuje všetky koncové body rozhrania API vrátane ich parametrov a všetkých možných odoziev servera. Kód servera a klienta možno vygenerovať z tohto popisu spolu s jeho dokumentáciou.

Open API dokument môže pozostávať z jedného dokumentu alebo môže byť rozdelený do viacerých spojených častí podľa uváženia autora. Odporúča sa, aby bol hlavný open API dokument pomenovaný: openapi.json alebo openapi.yaml.

Ukážka minimálnych polí v open API dokumente vo formáte YAML (bez definovaných koncových bodov):

```yaml
openapi: 3.1.0
info:
  title: Testovacie REST API
  version: 0.0.1
paths: {} # No endpoints defined
```

OAS popisuje ďalšie používané objekty na popis API, ktoré je možné nájsť tu.

Open API iniciatíva poskytuje aj osvedčené postupy (best practices), ktoré sa nemusia vyslovene týkať open API špecifikácie, avšak zjednodušujú vytváranie a udržiavanie open API dokumentov:

Použite prístup design-first
Tradične existujú dva hlavné prístupy pri vytváraní open API dokumentov: Code-first a Design-first.

V prístupe Code-first sa API najskôr implementuje do kódu a potom sa z neho vytvorí jeho popis pomocou komentárov kódu, anotácií kódu alebo sa popis jednoducho napíše od začiatku (tzv. "from scratch"). Tento prístup nevyžaduje, aby sa vývojári učili ďalší

jazyk, takže sa zvyčajne považuje za najjednoduchší. Naopak, v Design-first prístupe sa najprv napíše popis API a potom nasleduje kód. Prvými zjavnými výhodami je, že kód už má kostru, na ktorej sa dá stavať, a že niektoré nástroje dokážu automaticky poskytnúť štandardný kód. Prebehlo množstvo búrlivých debát o relatívnych výhodách týchto dvoch prístupov, ale podľa názoru open API iniciatívy už nie je možné ešte viac zdôrazniť dôležitosť používania prístupu Design-first.

Dôvod je jednoduchý: Počet rozhraní API, ktoré je možné vytvoriť v kóde, je oveľa vyšší ako to, čo je možné opísať v open API. Open API nie je schopné opísať každé možné HTTP API, pretože má obmedzenia. Preto pokiaľ tieto obmedzenia nie sú dokonale známe a nezohľadňujú sa pri kódovaní API, budú neskôr pri pokuse o vytvorenie popisu predstavovať vážny problém a v takomto prípade bude správnou opravou zmena kódu tak, aby používal API, ktoré možno skutočne opísať pomocou open API (alebo úplne prejsť na Design-first prístup). Niekedy však, keďže sme už príliš ďaleko v procese, sa môže vyskytnúť tendencia preferovať prispôsobenie popisu API tak, aby sa viac-menej zhodoval so skutočným API. Takéto riešenie vedie k neintuitívnym a neúplným popisom, ktorých negatívne dôsledky sa budú postupom času iba zintenzívňovať.

K dispozícii je však množstvo validačných nástrojov, ktoré dokážu overiť, či implementovaný kód zodpovedá popisu open API. Spustenie týchto nástrojov ako súčasť kontinuálneho procesu integrácie umožní zmenu open API dokumentu s istotou, že odchýlky v správaní kódu budú okamžite zistené.

Udržujte jediný zdroj pravdy
Bez ohľadu na to, či využívate prístup Design-first alebo Code-first, vždy udržujte iba jeden zdroj pravdy. To znamená, že informácie nebudú udržiavané duplicitne na viacerých miestach. Ide o podobný princíp ako v programovaní, keď sa opakujúca časť kódu presunie pod jednu spoločnú funkciu.

V opačnom prípade sa môže stať, že vznikne nekonzistencia z dôvodu, že jeden zdroj bol aktualizovaný a druhý nie. Napríklad je bežnou praxou používať anotácie kódu na vygenerovanie popisu open API a následne takýto popis upraviť na základe kontroly, zatiaľ čo ale zdroj zostane nezmenený. V takýchto prípadoch vzniká riziko, že napríklad nový člen tímu nesprávne usúdi, ktorý popis je aktuálny a bude postupovať podľa nesprávneho.

Alternatívou je vykonávanie kontinuálneho testovania za účelom zabezpečenia súladu obidvoch zdrojov.

Pridajte open API dokumenty do source control
Popisy OpenAPI nie sú len artefaktom dokumentácie, ale sú to prvotriedne zdrojové súbory, ktoré môžu riadiť veľké množstvo automatizovaných procesov, napríklad generovanie štandardných modelov, testovanie a pod.

Open API dokument by mal byť postúpený do source control medzi prvými súbormi a odtiaľ by sa mal podieľať na procesoch kontinuálnej integrácie.

Sprístupnite open API dokumenty používateľom
Rozsiahlo napísaná dokumentácia môže byť pre používateľov API veľmi užitočná, avšak niekedy budú skôr potrebovať zdroj open API popisu (napríklad aby si pre seba vygenerovali klientsky kód alebo pre vytvorenie automatických väzieb v určitom jazyku).

Sprístupnenie open API dokumentov je pre používateľov bonusom navyše. Dokument môže byť dokonca sprístupnený prostredníctvom rovnakého API rozhrania, aby bolo jeho objavovanie umožnené rovno za behu.

Zriedkakedy je potrebné písať open API dokumenty ručne
Keďže open API dokumenty sú obyčajné textové súbory v ľahko čitateľnom formáte (či už je to JSON alebo YAML), dizajnéri API sú zvyčajne v pokušení písať ich ručne. Aj keď vám v tom nič nebráni a v skutočnosti sú ručne písané popisy API zvyčajne najstručnejšie a najúčinnejšie, pristupovať k akémukoľvek veľkému projektu takýmto spôsobom je veľmi nepraktické.

Namiesto toho by ste mali vyskúšať iné existujúce možnosti tvorby dokumentácie a vybrať si tú, ktorá lepšie vyhovuje vám a vášmu tímu (nie sú potrebné žiadne znalosti YAML alebo JSON):

- Open API editory: Či už ide o textové editory alebo GUI editory, zvyčajne dokážu spravovať opakujúce sa časti, umožňujú vám uchovávať knižnicu opakovane použiteľných komponentov a poskytujú náhľad vygenerovanej dokumentácie v reálnom čase.
- DSL (Domain-Specific Languages): DSL sú jazyky na popis API prispôsobené špecifickým oblastiam vývoja. Potom sa použije nástroj na vytvorenie open API dokumentu. Nevýhodou je potreba naučiť sa nový jazyk, avšak na oplátku vďaka DSL možno dosiahnuť mimoriadne výstižné opisy.
- Anotácie kódu: Väčšina programovacích jazykov vám umožňuje anotovať kód, či už špecifickou syntaxou alebo všeobecnými komentármi kódu. Tieto anotácie možno napríklad použiť na rozšírenie popisu metódy o informácie týkajúce sa koncového bodu API a metódy HTTP, ktoré k nemu vedú. Použitím príslušného nástroja potom môžete analyzovať anotácie kódu a automaticky generovať open API dokument. Táto metóda je vhodná hlavne v prípade použitia prístupu Code-first, ktorý však nie je odporúčaný.
- Kombinácia všetkého vyššie uvedeného: Je možné vytvoriť veľkú časť open API dokumentu pomocou editora alebo DSL a potom ručne vyladiť výsledný súbor. V takýchto prípadoch je potrebné venovať zvýšenú pozornosť udržaniu jedného zdroja pravdy.

Práca s veľkými dokumentmi
Toto je sumár odporúčaní pri práci s rozsiahlou dokumentáciou popisujúcou API:

- Neopakujte sa. Ak sa rovnaký kus YAML alebo JSON objaví v dokumente viackrát, je potrebné ho presunúť do sekcie komponentov a odkazovať naň z iných miest pomocou $ref (Výsledný dokument bude nielen menší, ale tiež oveľa jednoduchší na údržbu).
- Na komponenty je možné odkazovať z iných súborov, takže ich môžete dokonca znova použiť v rôznych API dokumentoch.

Tento projekt je podporený z Európskeho sociálneho fondu.

- Rozdeľte dokument do niekoľkých súborov: V menších súboroch je jednoduchšia navigácia, ale príliš veľa súborov je tiež nežiaducich. Potrebné je nájsť vyhovujúci kompromis.
- Dobrým pravidlom je použiť prirodzenú hierarchiu prítomnú v adresách URL na vytvorenie štruktúry súborov. Napríklad vložte všetky cesty začínajúce na /users (ako /users a /users/{id}) do rovnakého súboru (predstavte si to ako „sub-API“).
- Majte na pamäti, že niektoré nástroje môžu mať problémy s veľkými súbormi, zatiaľ čo niektoré iné nástroje nemusia správne spracovať príliš veľa súborov.
- Použite tagy na usporiadanie obsahu: Tagy vám môžu pomôcť usporiadať vaše operácie a rýchlejšie ich nájsť. Tag je treba vnímať ako časť metadát (jedinečný názov a voliteľný popis), ktoré môžete pripojiť k operáciám.

3.2 Swagger – best practices

V rámci projektu Swagger došlo k snahe o elimináciu často sa opakujúcej dokumentácie API rozhraní a opakovanému generovaniu klientskeho SDK zavedením zobrazenia API vo formáte JSON. Swagger je predchodcom Open API 3.0. Zároveň boli v rámci Swagger publikované osvedčené postupy (best practices)[40] pre dizajnovanie API uvedené nižšie.

Efektívny API design by mal mať vo všeobecnosti tieto vlastnosti:

- Jednoduchý na čítanie a prácu s ním (S dobre navrhnutým API sa bude ľahko pracovať a vývojári, ktorí s ním neustále pracujú si ľahko zapamätajú jeho zdroje a s nimi súvisiace operácie).
- Jednoduchá integrácia (Implementácia a integrácia dobre dizajnovaného API je jednoduchý proces, pričom napísanie nesprávneho kódu je potom málo pravdepodobné.
- Úplnosť a stručnosť (Úplné API umožní vývojárom vytvárať plnohodnotné aplikácie vo vzťahu k vystavovaným údajom. Vďaka tomu vývojári môžu postupne stavať na existujúcich API rozhraniach, čo je ideálom pre spoločnosti.).

Na účely ilustrácie nižšie uvedených pojmov bude použitý príklad aplikácie na zdieľanie fotografií. Táto aplikácia umožňuje používateľom nahrávať fotky, pričom ich charakterizuje miestom, kde boli nasnímané a hashtagmi (#popiskami), ktoré popisujú emócie s nimi spojené.

Kolekcie, zdroje a ich URL

Pochopenie zdrojov a kolekcií: Zdroje sú základom conceptu REST API, ide o objekty, ktoré sú dostatočne dôležité na to, aby sa na ne referencovalo. Súčasťou zdroja sú dáta, vzťahy k iným zdrojom a metódy, prostredníctvom ktorých sa k nemu pristupuje. Skupina zdrojov sa nazýva kolekcia. Obsah kolekcií a zdrojov závisí od organizačných a spotrebiteľských požiadaviek.

[40] Best Practices in API Design (swagger.io)

Ak existuje napríklad požiadavka na vystavovanie základných informácií o používateľskej základni produktu, je možné ju vystaviť ako kolekciu alebo zdroj. URL identifikuje online umiestnenie zdroja, pričom URL odkazuje na miesto, kde sú zverejnené zdroje API rozhrania (základná cesta URL je konzistentnou súčasťou tohto umiestnenia).

Napríklad pri aplikácii na zdieľanie fotografií, by sme mohli sprístupniť údaje o používateľoch, ktorí aplikáciu používajú, viď nižšie:

/users: a collection of users
/users/username1: a resource with information about a specific user

Používanie podstatných mien pre URL: Základná URL adresa by mala byť prehľadná a jednoduchá, aby ju vývojári používajúci váš produkt mohli ľahko použiť vo svojich aplikáciách. Dlhá a ťažko čitateľná základná URL adresa je nielen zlá na pohľad ale môže byť aj náchyná na chyby pri pokuse o jej prekódovanie.

Použitiu podstatných mien treba venovať zvýšenú pozornosť – neexistuje žiadne pravidlo na ponechanie podstatných mien v jednotnom alebo množnom čísle, aj keď v prípade kolekcií je vhodné ponechať množné číslo. Dobrou praxou je uplatňovanie množného čísla konzistentne vo vzťahu ku všetkým zdrojom a kolekciám. Udržiavanie podstatných mien ako samovysvetľujúcich napomáha vývojárom porozumieť druhu zdroja opísaného v URL adrese.

Príklad: Povedzme, že aplikácia na zdieľanie fotografií má verejné API s kolekciami /user a /photos. Tieto podstatné mená sú uvedené v množnom čísle a sú tiež samovysvetľujúce, pretože z toho možno vyvodiť, že:

/users poskytuje informácie o používateľskej základni,
/photos poskytuje informácie o zdieľaných fotografiách.

Popíšte operácie súvisiace so zdrojmi pomocou HTTP metód

Ku zdrojom sa vzťahuje množina metód používaných pre prácu s údajmi, ktoré API rozhranie sprístupňuje. REST API pozostávajú hlavne z HTTP metód, ktoré jednoznačne popisujú a definujú akcie uplatňované vo vzťahu ku zdrojom.

Zoznam bežne používaných HTTP metód, ktoré definujú CRUD operácie pre akýkoľvek zdroj alebo kolekciu v REST API:

GET – získanie reprezentácie zdroja
POST – vytváranie nových zdrojov
PUT – aktualizácia existujúcich zdrojov
PATCH – aktualizácia existujúcich zdrojov
DELETE – odstránenie existujúcich zdrojov

Dobrým pravidlom je udržiavanie týchto slovies (GET, POST, PUT, PATCH, DELETE) mimo vašich URL. Tieto slovesá sú už použité na operácie s vaším zdrojom popísaným v URL, preto ich zavedenie do URL namiestio podstatných mien (pozri vyššie Používanie podstatných mien pre URL) by mohlo pôsobiť zmätočne.

V aplikácii na zdieľanie fotografií s koncovými bodmi /users a /photos s nimi môže koncový spotrebiteľ vášho API jednoducho intuitívne pracovať pomocou operácií popísaných vyššie.

Poskytnite spätnú väzbu vývojárom
Poskytovanie hodnotnej spätnej väzby pre vývojárov o tom, ako dobre používajú váš produkt, výrazne zlepšuje jeho prijatie a udržateľnosť. Každá požiadavka klienta a odpoveď na strane servera je správou, ktorá je v ideálnom REST API ekosystéme samopopisnou.

Dobrá spätná väzba zahŕňa pozitívnu informáciu o správnej implementácii a informáciu o chybe pri nesprávnej implementácii, ktorá môže používateľom pomôcť pri vylaďovaní a náprave spôsobu použitia vášho produktu. V prípade API rozhrania sú chyby skvelým spôsobom ako poskytnúť kontext k jeho používaniu. Chyby zosúlaďte s HTTP chybovými kódmi, pričom existuje veľké množstvo kódov pre odpovede a vo všeobecnosti pripadajú do úvahy tri možné výsledky pri používaní vášho API:

- chyba je na strane klientskej aplikácie, kód odpovede 4xx (napríklad chybné volanie na strane klienta),
- chyba je na strane servera, kód odpovede 5xx (t. j. API rozhranie sa správalo chybne),
- fungujú obidve strany (klient aj API), t. j. kód odpovede 2xx.

K chybovým kódom poskytnite používateľovi dostatočné informácie, aby mohol začať pracovať na odstránení – ak je to vhodné, poskytnite odkazy na ďalšiu dokumentáciu.

Poskytnite príklady ku všetkým odpovediam na operáciu GET
Dobre navrhnuté API obsahuje aj príklady odpovede, ktorú môže používateľ očakávať v prípade úspešného volania. Takýto príklad by mal byť jednoduchý, jasný a ľahko pochopiteľný. Dobrým pravidlom je pomôcť vývojárom presne pochopiť, ako by vyzerala úspešná odpoveď získaná za menej ako päť sekúnd.

Príklad: Aplikácia na zdieľanie fotiek mala definované URL pre /users a /photos. Kolekcia /users by poskytla používateľské meno a dátum pripojenia všetkých používateľov. Na definovanie API v open API špecifikácii možno použiť nástroj na návrh API takto:

```
responses:
200:
description: Successfully returned information about users
schema:
type: array
items:
```

```
type: object
properties:
username:
type: "string"
example: "kesh92"
created_time:
type: "dateTime"
example: "2010-01-12T05:23:19+0000"
```

Úspešná odpoveď, ktorú by používateľ získal v JSON formáte by vyzerala nasledovne:

```
{
“data”:[
{
“Username”:“example_user1”,
“created_time":“2013-12-23T05:51:14+0000 ”
},
{
“username”:“example_user2”,
“created_time":“2015-3-19T17:51:15+0000 ”
}
….
]
}
```

Ak používateľ úspešne zavolá koncový bod pomocou GET, mal by získať vyššie uvedené údaje spolu s kódom odpovede 200 (overenie správneho použitia).

3.3 Národný API dizajn štandard pre Austráliu

V Austrálii je vytvorený celoštátny portál pre API štandardy[41]. S modernizáciou služieb poskytovaných verejným sektorom, predovšetkým so zvyšovaním sa podielu digitálnych služieb sa vynárajú nové výzvy v súvislosti s konektivitou, interoperabilitou služieb a aspektmi bezpečnosti zdieľania údajov.

Tieto výzvy sú ešte významnejšie, keď jednotlivé organizácie navrhujú služby využívané inými vládnymi subjektmi v rámci celej Austrálie – vyžaduje sa strategický prístup, ktorý zohľadňuje súčasných známych používateľov a ich požiadavky, ako aj ešte neznámych používateľov, ktorí budú služby v budúcnosti potrebovať. Na vyriešenie tohto problému sa od orgánov verejnej moci očakáva, že vytvoria holistickú integračnú stratégiu ako kritickú súčasť svojho technického arzenálu, ktorá zefektívni opätovné použitie služieb, zjednoduší existujúce prostredia a odstráni riziká pri implementácii. Kľúčovou súčasťou

[41] api.gov.au

takejto integračnej stratégie je štandardizovaný dizajn rozhraní medzi rôznymi systémami.

V roku 2019 Austrálska rada pre údaje a digitálne transformácie schválila národný API dizajn štandard podľa príkladu štátu Viktória a iných jurisdikcií, ktoré individuálne rozvíjajú štandardy pre API dizajn určené pre osoby navrhujúce design API vo svojich vlastných jurisdikciách.

Všeobecnými cieľmi tohto národného štandardu sú:

- vytvorenie štandardu API, ktorý sa bude používať pri zdieľaní údajov medzi jednotlivými jurisdikciami,
- zlepšenie celkovej kvality API rozhraní, ako aj gramotnosti týkajúcej sa API a integrácií,
- vybudovanie pracovnej skupiny odborníkov na API rozhrania zastupujúcich všetky jurisdikcie, ktorí môžu pokračovať v presadzovaní štandardu na území celej Austrálie.

Štandard je referenčným dokumentom pre fázu dizajnu v procese vývoja nového API rozhrania, pričom predstavuje spoločné vzory pre všetky API a integračné scenáre, ktoré poskytujú dohodnutý pohľad na to, ako by sa mali tieto scenáre riešiť. Jeho súčasťou sú vzorové šablóny OpenAPI vo verzii 2.0 a 3.0 vo formáte JSON a YAML, ktoré môžu byť použité ako základ v procese definovania API[42].

Návrh je orientovaný na REST API vzhľadom na jeho súčasné rozšírenie medzi vládnymi inštitúciami v Austrálii, pričom sa predpokladá zároveň budúci vývoj tohto štandardu vo vzťahu k GraphQL a gRPC/JSON-RPC.

V kontexte štandardu je API definované ako REST API - spôsob komunikácie medzi systémami, kde sú zdroje definované pomocou URI a operácie sú definované použitím HTTP metód.

Zdroje: Aby bolo možné navrhnúť použiteľné API, vaše systémy musia byť rozdelené do logických skupín (často nazývaných modely alebo zdroje). Vo väčšine prípadov sú zdroje „podstatné mená“ vášho systému (napríklad v HR systéme môžu byť zdroje zamestnanci, pozície a žiadosti o dovolenku).

Identifikátor zdroja: Každý dostupný zdroj vášho systému musí byť v rámci systému unikátne identifikovateľný. Identifikátor zdroja môže mať formát čísla, reťazca, GUID, dátumu.

Reprezentácie: Kľúčovou súčasťou REST API je reprezentácia zdroja v konkrétnom čase. Keď od systému budeme požadovať informáciu o zamestnancovi, dostaneme ako odpoveď reprezentáciu toho zamestnanca, napríklad:

[42] vzor JSON, vzor YAML

Tento projekt je podporený z Európskeho sociálneho fondu.

```
HTTP 1.1 GET /employees/0d047d80-eb69-4665-9395-6df5a5e569a4
Accept: application/json

200 OK
Content-Type: application/json

{
  "name" : "John Smith",
  "employee_id" : "0d047d80-eb69-4665-9395-6df5a5e569a4",
  "position" : "Manager",
  "onLeave" : false
}
```

Menný priestor (Namespace): Definuje zoskupenie súvisiacich funkcií a môže byť uvedený na vyššej alebo nižšej úrovni (napríklad názov organizácie/oddelenia alebo projekt/tím/služba). Menný priestor je relevantný pri navrhovaní URL štruktúry.

Operácie: Aby vývojári mohli použiť ktorýkoľvek z menných priestorov, zdrojov a zdrojových identifikátorov, musia použiť operácie. Operácia je definovaná použitím HTTP metódy a cesty ku zdroju.

Požiadavky na API

Dokumentácia API: Dobre zdokumentované API je kritickou súčasťou implementácie API. Tam, kde je to možné, bude štandardizovaná aj štruktúra, metódy, konvencie názvov a reakcie, aby bol poskytnutý spoločný základ pre vývojárov. Všetky API rozhrania vytvorené pre austrálske orgány verejnej moci musia byť popísané prostredníctvom OpenAPI dokumentu vo verzii 2.0, pretože má najširšiu podporu. Do budúcnosti môže byť poskytnutý aj OpenAPI dokument vo verzii 3.0. Ďalej sú v štandarde uvedené odporúčané časti OpenAPI dokumentu.

Vývoj API: Štandard uvádza pokyny, ktoré majú byť dodržané pri vývoji API, ako napríklad:

- dokumenty popisujúce API by mali obsahovať API dokumentáciu (high-level informácie a popisy) a informáciu o verzii.
- Mock API by mali byť vytvorené pomocou popisu API, aby bola pre vývojárov umožnená skorá integrácia pri tvorbe kódu,
- správanie API a jeho zámer by mali byť popísané čo najširšie uvedením čo najväčšieho množstva informácií,
- ak je to možné, dokumentácia by mala byť zverejnená a ľahko prístupná pre tých, ktorí ju potrebujú,
- očakávané telá požiadaviek a odpovedí by mali byť poskytnuté v plnom znení,
- mali by byť používané korektné kódy odpovedí (HTTP Response Codes),
- očakávaný výkon, doba prevádzky, SLA/OLA by mali byť jasne zdokumentované,
- celá API dokumentácia by mala byť vytlačiteľná alebo exportovateľná.

Dostupnosť popisu API – Súbory s popisom API musia byť dostupné:

- online: ako súčasť produktu API pod /<namespace>/<project-name>/v<x>/api-docs
- offline: ako súčasť produktového balíka

Formát súboru – Všetky OpenAPI dokumenty by mali byť poskytované vo formáte JSON.

Verziovanie – Pre každú hlavnú verziu musí existovať popis open API. To znamená, že ak produkt API udržiava tri verzie, musia byť poskytnuté tri popisy open API (jeden pre každú verziu: v1, v2, v3).

Vývojárska skúsenosť a jednoduchosť použitia – API rozhranie, ktoré sa ťažko používa, znižuje pravdepodobnosť, že ho budú používatelia ďalej používať a zároveň jeho používanie ani nebudú odporúčať ďalším potenciálnym používateľom. Preto sa navrhované API odporúča testovať so skutočnými spotrebiteľmi, pričom každá spätná väzba by mala byť zohľadnená a relevantným spôsobom zapracovaná do API, aby sa dosiahol čo najlepší výsledok. Tím pracovnej skupiny odborníkov na API zastupujúci všetky jurisdikcie poskytuje proces kontroly, ktorý má zabezpečiť, že API spĺňa základnú úroveň použiteľnosti ešte predtým, ako bude poskytnuté potenciálnym spotrebiteľom na získanie spätnej väzby.

Stabilita API – API musia byť navrhnuté s ohľadom na spätnú kompatibilitu, pretože keď sa budú zavádzať zmeny, je málo pravdepodobné, že ich používatelia okamžite implementujú do svojich aplikácií. Vlastníci produktu API by mali zdokumentovať životnosť podpory pre služby API (napríklad ako dlho budú podporované). Nové funkcie musia byť zavedené spôsobom, ktorý neovplyvní existujúcich používateľov.

Zrelosť API dizajnu – Pri navrhovaní API je jedným z kľúčových hľadísk vývojárska skúsenosť s používaním tohto API, pričom zrejme najznámejším konceptom pre vývojárov API je štýl REST API. Ako pomoc pre osoby, ktoré API navrhujú, je použitý Richardsonov model zrelosti, ktorý rozdeľuje REST API do štyroch rôznych úrovní na základe používania URI, HTTP metód a HATEOAS:

- úroveň 0 (základný stav pre akékoľvek nové API),
- úroveň 1 (API implementuje rôzne URI ale iba jednu HTTP metódu, napríklad POST),
- úroveň 2 (API implementuje rôzne URI a viaceré HTTP metódy, napríklad CRUD cez GET/POST/PUT/DELETE),
- úroveň 3 (API implementuje rôzne URI, viaceré HTTP metódy a HATEOAS na reprezentáciu vzťahov medzi objektami).

Všetky API navrhované v súlade s austrálskym štandardom musia byť navrhnuté podľa úrovne 2 Richardsonovho modelu zrelosti (môže byť zvolená aj úroveň 3, avšak nie je to povinné).

3.4 Veľká Británia – technický a dátový štandard

Vo Veľkej Británii je verejne dostupný technický a dátový štandard[43], pre ľudí tvoriacich API vo vládnom prostredí. Podľa popisu štandardu ide o návod, ktorý poskytuje osvedčený postup pre návrh, tvorbu a správu API na používanie vo vládnom prostredí a službách. Nižšie je ako príklad uvedený (zjednodušený) obsah tohto štandardu:

- Návrh API

Pred vytvorením API by ste si mali dôkladne naplánovať, čo bude robiť a ako bude fungovať. Tím, ktorý API navrhuje by mal definovať potreby používateľa vrátane identifikácie toho, kto bude API používať a čo s ním bude potrebovať urobiť. Začnite:

- zhromaždením používateľských a biznis požiadaviek, ktoré umožnia definovať, čo má API robiť
- identifikáciou kľúčových entít (inštitúcií), s ktorými musia používatelia interagovať prostredníctvom API,
- vytvorením špecifikácie pred samotnou tvorbou API (t. j. predtým ako začnete „kódovať“),
- testovaním vašich predpokladov s používateľmi,
- iterovaním návrhu na základe získanej spätnej väzby.

Toto vám umožní zamerať sa na zjednodušenie rozhrania a elimináciu všetkých funkcií, ktoré nie sú pre používateľov potrebné. Je potrebné, aby bolo vaše API čo najjednoduchšie predvídateľné a intuitívne, pretože nie všetci používatelia čítajú vždy celú dokumentáciu.

Mali by ste sa vyhnúť zavádzaniu zmien – po integrácii API so službou môžu byť ľudia menej ochotní alebo schopní aktualizovať svoj kód. Po uvedení do prevádzky budú chcieť vykonať čo najmenej zmien. Mali by ste sa zamyslieť nad službami, ktoré závisia od vášho API a ako by ich zmeny mohli ovplyvniť. Ak je to možné, nevykonávajte zmeny, ktoré zmenia spôsob interakcie medzi službou a API. Využitie používateľsky orientovaného dizajnovania zvýši pravdepodobnosť, že API splní potreby používateľov a počas svojej životnosti nebude vyžadovať príliš veľa zmien. Ak však bude potrebné zaviesť nejaké zmeny, je potrebné postupovať podľa pokynov na vytváranie verzií a vyraďovanie API z prevádzky.

Skontrolujte existujúce API – preverte, či neexistujú už nejaké API, ktoré by bolo možné použiť namiesto tvorby nového API (môže ísť o interné/externé API alebo komerčne dostupné API). Mali by ste si pozrieť katalóg dostupných API[44]. Opätovné použitie už existujúceho API je rýchlejšie a jednoduchšie ako vytváranie nového API od začiatku. Začlenenie preverenia použiteľnosti už existujúceho API do procesu dizajnovania

[43] API technical and data standards - GOV.UK (www.gov.uk)
[44] Vo Veľkej Británii je dostupný tzv. cross-government UK API catalogue API Catalogue (www.api.gov.uk)

Tento projekt je podporený z Európskeho sociálneho fondu.

zvyšuje pravdepodobnosť, že nové API sa vytvárajú iba v prípade skutočnej potreby (a uprednostňuje sa opätovné použitie už existujúcich API).

Dodržiavajte Technologický kódex[45], dátové štandardy vydané vládou a všetky zákonné požiadavky - API by ste mali navrhnúť tak, aby bolo v súlade so všetkými príslušnými štandardmi vydanými vládou (Technologický kódex - Technology Code of Practice, je súbor kritérií, ktoré pomáhajú vládnym inštitúciám vo Veľkej Británii navrhovať, vytvárať a kupovať technológie; zoznam schválených štandardov sa dá nájsť v katalógu dátových štandardov[46]).

Dizajnujte cez prístup API-first, t. j. vytvorte API ako prvé rozhranie k vaším údajom a zvyšok služby môže byť vystavaný okolo tohto API. Takýto prístup znižuje riziko opätovnej práce, ak by bolo neskôr potrebné externé API. Zároveň API-first má ďalšie výhody:

- iné služby môžu využiť vaše API,
- testovanie API vašimi vlastnými internými službami, čo umožní kontinuálne zlepšovanie vrátane vylepšenia API aj pre externých používateľov (tým sa zlepšuje ich skúsenosť napríklad tak, že API bude mať dokumentáciu, ktorá skutočne spĺňa svoj účel),
- modularita a opätovné využitie kódu, pretože API nebude prispôsobené konkrétnej už existujúcej službe,
- základné dátové štruktúry API budú zodpovedať svojmu účelu, t. j. navrhnutie dátovej štruktúry pre API je hneď na začiatku oddelené od biznis logiky služby.

Na vytvorenie API použite REST – to znamená, že API rozhranie sa riadi architektonickým štýlom REST a pracuje s RESTful webovými službami. V závislosti od konkrétneho prípadu je možné zvoliť aj iný štýl, napríklad GraphQL je zas užitočný pre prototypové služby, pri ktorých si nie ste istí, aké zobrazenie údajov budú potrebovať ďalší vývojári. Voľba architektonického štýlu API by vždy mala vyplývať z potrieb konkrétneho prípadu.

Vytvorte open API dokument – špecifikácia open API je štandardizovaný spôsob popisu API, pričom dobrým zvykom je vytvorenie open API dokumentu už v procese dizajnovania, t. j. dokumentácia sa vyvíja spolu s návrhom API. Takýto prístup umožní:

- preukázať, že API bolo vyvinuté konzistentne v spolupráci s ostatnými z vašej organizácie,
- testovanie API vo vzťahu k požadovaným pravidlám a bezpečnostným otázkam,
- automatické generovanie referenčnej dokumentácie.

[45] The Technology Code of Practice - GOV.UK (www.gov.uk)
[46] Standards - Data Standards Authority (alphagov.github.io)

Tento projekt je podporený z Európskeho sociálneho fondu.

Naplánujte si svoje API bezpečne – bezpečnosť API by ste mali zvažovať už od samotného začiatku procesu navrhovania API. Bezpečnosť API nie je triviálna záležitosť a zahŕňa:

- zabezpečenie na úrovni údajov (používatelia majú prístup iba k tým z poskytovaných údajov, ku ktorým majú oprávnenie),
- zabezpečenie na úrovni aplikácie (prístup k API majú iba oprávnení používatelia),
- auditovanie (používanie API je monitorované).

Nadácia OWASP vytvorila zoznam desiatich hlavných bezpečnostných rizík pre API[47] – pri navrhovaní API sa uistite, že je zabezpečené voči všetkým z nich.

Bezpečne hostujte svoje API – keď navrhujete API, je dôležité myslieť na to, kde ho budete hostovať a ako bude fungovať počas svojho životného cyklu. Keď vytvárate názov a rozhodujete sa o hostovaní svojho API, mali by ste postupovať podľa návodu na výber API doménového názvu[48].

- Vytvorenie API

Na kódovanie API rozhrania použite štandard UTF-8 – unicode je svetový štandard pre konzistentné kódovanie, reprezentáciu a spracovanie textu vo väčšine globálnych systémov písania. Pri kódovaní znakovej sady Unicode by ste mali používať štandard Unicode Transformation Format (UTF-8)[49].

Pre formáty odpovede použite JSON – ak je to možné, pri štruktúrovaní formátov odpovedí REST API rozhrania používajte štandard JSON. Mali by ste použiť oficiálnu špecifikáciu pre JSON[50]. V súčasnosti sa používa viacero formátov JSON, pričom (ak vaša organizácia ešte žiadny nešpecifikovala) odporúčame použiť JSON: API[51], ktorý je špeciálne navrhnutý pre API odpovede.

V niektorých prípadoch budete potrebovať alternatívu k REST na priradenie k dátovej štruktúre. Ak napríklad budete aktualizovať API, ktoré momentálne poskytuje odpovede vo formáte XML, môže byť pre používateľov lepšie, ak zachováte rovnaký formát. Mali by ste sa však uistiť, že je to dobre zdokumentované, pretože XML je stále menej používaným formátom.

Voľba všeobecne prijatého štandardu vždy, keď je to možné, poskytuje výhody ako napríklad:

- úspora času elimináciou diskusií o tom, aký formát použiť,
- umožnenie využiť stanovené „best practices“,
- uľahčenie integrácie pre externých vývojárov.

[47] OWASP API Security Project | OWASP Foundation
[48] Get an API domain on GOV.UK - GOV.UK (www.gov.uk)
[49] Encoding characters - GOV.UK (www.gov.uk)
[50] JSON
[51] JSON:API — A specification for building APIs in JSON (jsonapi.org)

Tento projekt je podporený z Európskeho sociálneho fondu.

Používajte konzistentné názvy pre zdroje – vývojári by mali byť schopní prevziať názvy zdrojov z vášho API na základe kontextu. Pre podobné zdroje použite podobné výrazy, napríklad user_id a address_id. Používatelia by nemali nutne potrebovať znovu čítať dokumentáciu aby vedeli, aký je názov konkrétneho zdroja.

Majte perzistentné zdroje – zdroje, ktoré poskytuje vaše API (odpovede na požiadavky) by sa nemali meniť. Mali by ste sa uistiť, že nezáleží na tom, či stĺpce vo vašej databáze napríklad zmenia názov, pretože API rozhranie by malo poskytnúť mapu z názvu zdroja na základné údaje.

Použite štandardné HTTP odpovede – uistite sa, že kódy chýb sú zhodné s HTTP odpoveďami. Konzistentné a ľahko čitateľné chybové kódy sú kľúčové pre pochopenie toho, kde došlo k chybe. Mali by ste zdokumentovať všetky chybové kódy a zabezpečiť, aby sa dali ľahko nájsť. Zároveň vlastné chybové kódy by mali obsahovať iba informácie potrebné na diagnostiku problému a nemali by obsahovať žiadne nepodstatné informácie, ktoré by mohli pomôcť útočníkovi zacieliť na službu (napríklad technické podrobnosti o systéme, na ktorom API beží).

Nastavte úrovne autorizácie používateľov – keď vytvárate svoje API, ako čo najvhodnejšie povoliť prístup k jeho údajom (prečítajte si pokyn na správu rozhrania API[52]). Autentifikácia na úrovni používateľa je vhodná na audit a riadenie prístupu (použite ju, ak chcete ovládať, kto môže pristupovať k vášmu API – je to nevyhnutné pri práci s osobnými údajmi). Autorizáciu na úrovni aplikácie použite, ak chcete ovládať, ktoré aplikácie môžu pristupovať k vášmu API bez obmedzenia toho, kto k nim môže pristupovať.

Zvážte výkon a škálovateľnosť – výkon vášho API môžete merať podľa toho, ako rýchlo sa dokáže vysporiadať s jednou požiadavkou a urobiť odpoveď. Jeho škálovateľnosť je množstvo požiadaviek, ktoré dokáže riešiť súčasne pri zachovaní prijateľnej úrovne výkonu. Výkon a škálovateľnosť odozvy API môžete zlepšiť tak, že ju nastavíte do vyrovnávacej pamäte. To znamená, že odpoveď API môže ukladať kópie často prístupných údajov pozdĺž cesty požiadavky a odpovede.

Poskytnite testovaciu službu API – mali by ste sa pokúsiť poskytnúť používateľom testovaciu službu (známu aj ako „sandbox“), ktorá im uľahčí integráciu. Generovanie sandboxov uľahčuje vytvorená open API dokumentácia. Svoje testovacie služby by ste mali sprístupniť bez overenia, pričom je potrebné zabezpečiť, aby testovacia služba nikdy neobsahovala skutočné údaje. To, že máte k dispozícii testovaciu službu znamená, že vývojári sa môžu s API oboznamovať prostredníctvom sandboxu veľmi skoro. Môžu napríklad začať v testovacej službe s testovacími údajmi, zatiaľ čo sa pripravuje dohoda o zdieľaní údajov na prístup k skutočným údajom.

Otestujte súlad vášho API – musíte zabezpečiť, aby vaše API spĺňalo štandardy, ktoré sú požadované aj z právneho hľadiska (pre vašu organizáciu aj tím). Zároveň musíte dodržiavať stanovené GDPR pravidlá.

[52] Defining an API management strategy - GOV.UK (www.gov.uk)

Umiestnite svoje API do API katalógu – po otestovaní súladu vášho API s právnymi požiadavkami by ste mali API pridať do API katalógu. Zverejnenie (zviditeľnenie) vášho API zvyšuje pravdepodobnosť jeho použitia.

Zdokumentujte svoje API – pri tvorbe dokumentácie by ste mali postupovať podľa nasledovných pokynov:

- dokumentovanie API[53]
- písanie referenčnej dokumentácie API[54].

- Správa API

Podporujte staršie verzie svojho API – pri tvorbe nových verzií API by ste sa mali snažiť o to, že nevykonávate zmeny, ktoré by zabránili správnemu fungovaniu starších verzií vášho API. Ak je to nevyhnutné a nie je možné zachovať fungovanie starších verzií, mali by ste používateľov informovať, že tieto staršie verzie už nie sú podporované.

Použite systém (alebo „gateway“) na správu API – platforma na správu API rozhrania poskytuje služby, ktoré nemusíte nutne vytvárať sami pri vývoji (napríklad kontrola a autorizácia prístupu, audit a protokolovanie, správa siete a pod.). Ide o dôležité služby, ktoré môžu byť spoľahlivejšie a jednoduchšie poskytované nástrojom na správu API (alebo „gateway“).

Kedy autentifikovať vaše API – autentifikácia umožní sledovať kto a na aký účel používa vaše API, pričom ju môžete použiť na:

- obmedzenia prostredníctvom limitov, aby používatelia nepoužívali nadmerne vaše zdroje,
- auditovanie (kontrola k akým údajom ktorí používatelia pristupovali),
- fakturáciu (ak bude API spoplatnené),
- autorizáciu (umožnenie rôznych úrovní prístupu na základe používateľov alebo rolí).

Na autentifikáciu odporúčame použiť nástroj tretej strany, buď ako súčasť alebo v spojení s API gateway.

Zaznamenávajte používanie API – ak vaše API poskytuje osobné alebo citlivé údaje, musíte zaznamenať (log), keď sú nejaké dáta poskytnuté a komu. Toto pomôže zostať v súlade s GDPR pravidlami, reagovať na žiadosti dotknutých osôb o prístup k údajom a odhaľovať podvody alebo zneužitie údajov.

Monitorujte API rozhranie v súvislosti s nezvyčajnou aktivitou – bezpečnosť všetkých technológií vo vašej organizácií musí byť zabezpečená, API nevynímajúc. Oboznámte sa s pokynmi technického kódexu[55]. Logovanie a používanie auditovacích nástrojov v rámci API gateway vám pomôže identifikovať, kedy sa používanie API rozhrania v priebehu času mení.

[53] Documenting APIs - GOV.UK (www.gov.uk)
[54] Writing API reference documentation - GOV.UK (www.gov.uk)
[55] Make things secure - GOV.UK (www.gov.uk)

Tento projekt je podporený z Európskeho sociálneho fondu.

3.5 Odporúčania pre open API pre mestá od skupiny 6Aika

Skupina šiestich najväčších fínskych miest známa ako 6Aika vytvorila odporúčania ohľadne open API adresované mestám[56]. Tento dokument popisuje význam API a s nimi súvisiace ciele z perspektívy miest. Zároveň je predmetom úpravy aj spoločný pohľad na vývoj API na základe vzájomnej spolupráce medzi mestami. Vízia 6Aika je postavená na spoločných cieľoch pre všetkých 6 miest a na opatreniach na dosiahnutie týchto cieľov na úrovni jednotlivých miest.

Spoločné ciele:

- Otvorenosť pri používaní a vývoji
- Jednoduchý prístup a implementácia
- Technické usmernenia a usmernenia ku kvalite
- Zabezpečenie kompetencie
- Open API sú súčasťou strategických politík
- Individuálny a potenciálne aj spoločný model pre manažment open API
- Spolupráca s domácimi a zahraničnými subjektami

Odporúčania pre dosiahnutie spoločných cieľov:

Pracujte na otvorenosti vo vzťahu k vývoju a používaniu API
Mestá by mali publikovať dáta zo svojich systémov ako otvorené dáta, pokiaľ to nie je vylúčené zo špecifických dôvodov. Rovnako mestské API by mali byť otvorené vždy, keď je to možné. Mestá môžu ťažiť z informácií, ktoré im dokážu poskytnúť tretie strany, vývojárske komunity a aj jednotliví vývojári – preto ak je to možné, cieľoví používatelia open API by mali byť už od začiatku zapojení do procesu tvorby API v rámci fázy navrhovania a vývoja. Čím viac relevantných pohľadov sa získa, tým vyššia je pravdepodobnosť, že API bude napĺňať potreby používateľov.

Uľahčenie objavovania API a jeho používania
Mestá by sa mali snažiť, aby sa nimi poskytované open API dali ľahko nájsť a aby boli ľahko prístupné na začatie používania.

Budujte API v súlade so štandardmi
Open API by mali byť v súlade s národnými a medzinárodnými štandardmi pre technológie a dátové modely, aby bola zabezpečená požadovaná úroveň kvality. Mali by ste nadviazať vzťahy so štandardizačnými organizáciami aby ste sa uistili, že vaše harmonizované API rozhrania a štandardy podporujú interoperabilitu aj napriek rozšíreniam alebo úpravám špecifickým pre jednotlivé mestá.

Veľmi dôležitá je zároveň príprava kvalitnej a aktuálnej dokumentácie, pričom dokumentácia spolu s pokynmi by mali byť potom zároveň zverejňované, aby ich mohol ktokoľvek použiť.

[56] Open API recommendations for cities - 6Aika

Tento projekt je podporený z Európskeho sociálneho fondu.

Zabezpečte relevantné kompetencie
Aby sa zachovala požadovaná úroveň kvality open API rozhraní aj pri implementácii, mestá by sa mali uistiť, že osoba zadávajúca zákazku na obstaranie, je dostatočne kompetentná na špecifikovanie a uvedenie všetkých potrebných informácií do výzvy na predloženie ponuky a do zmluvy.

Mestá by mali posilniť svoju kompetentnosť v oblasti open API organizáciou relevantných interných školení zamestnancov ako aj otvoreným šírením a zdieľaním týchto znalostí. Významnou pomocou môže byť aj vytvorenie podrobných pokynov/návodov.

Zahrňte open API do strategickej politiky mesta
Mestá by mali začleniť open API do svojich politík a zároveň by ich mali zohľadňovať v rámci svojich architektúr. Všeobecným pravidlom by malo byť, že open API by mali byť vždy súčasťou obstarávania pri obstarávaní nových IT systémov. Životný cyklus open API by mal byť navrhnutý rovnakým spôsobom ako životný cyklus iných IT riešení (treba však mať na pamäti, že životný cyklus open API môže byť odlišný od životného cyklu jeho back-end systému).

Vytvorte model riadenia pre open API štandardy a produkty
V rámci každej mestskej organizácie by pre každé open API mal byť vytvorený model riadenia. Konečným cieľom by mal byť spoločný model riadenia open API pre všetky mestá.

V rámci modelu riadenia sa už v počiatočných fázach plánovania berie do úvahy, či open API bude používať iba jedno mesto alebo viacero miest. Návrh by mal uprednostňovať agilný vývoj, avšak pri zohľadnení zásad stanovených na úrovni mesta, napríklad obmedzenie z pohľadu podporovaných platforiem. Akékoľvek postrehy ohľadne API musia byť zaznamenané, aby ich mesto mohlo zvážiť, pokiaľ bude aktualizovať existujúce zásady.

Posilnite spoluprácu realizovanú medzi miestnou, národnou a medzinárodnou úrovňou
Jednotlivé mestá ťažia zo vzájomnej spolupráce pri rozvoji ich digitálnej interoperability (mali by spolupracovať pri vytváraní harmonizovaných definícií, spoločných zásad a pod.). Mestá tvoriace 6Aika napríklad spolupracujú so štátom, Asociáciou fínskych lokálnych a regionálnych autorít, lokálnymi a globálnymi developerskými komunitami a s ďalšími subjektmi, ktoré rozvíjajú a využívajú open API a s tým súvisiace štandardy.

Zároveň je dôležité, aby sa implementovali štandardy a riešenia, ktoré sú bežne používané na medzinárodnej úrovni a zároveň by sa mestá mali snažiť o distribúciu riešení už existujúcich v zahraničí a zvyšovať tak potenciál ich využitia.

Tento projekt je podporený z Európskeho sociálneho fondu.

Opatrenia na úrovni jednotlivých miest:

Podporte definíciu open API

Spoločné vymedzenie open API umožní jednoduchšiu diskusiu ako aj lepšie šírenie povedomia o výhodách open API.

Open API je v stručnosti aplikačné programové rozhranie, ktoré je verejné a môže byť používané bez akýchkoľvek obmedzení a podmienok. To však neznamená, že sa hocikto dostane ku všetkým dátam alebo systému na pozadí API. Open API umožňuje jednoduchý prístup k tým dátam, ktoré sú na sprístupnenie určené. Zároveň za open API je možné považovať aj také API, pri ktorom sú stanovené podmienky pre používanie údajov alebo ktoré vyžaduje autentifikáciu používateľa alebo verifikáciu oprávnení používateľa.

6Aika podporuje definíciu open API vytvorenú COSS (Centrum pre otvorené systémy a riešenia, fínska nezisková organizácia[57]), Open Knowledge Finland a API:Suomi community. Ak je to potrebné, mestá môžu rozvíjať aj iné spoločne dohodnuté definície.

Napríklad ak je potrebné definovať "čiastočne" otvorené API, ktoré možno poskytnúť partnerom na špecifický účel (v praxi môže ísť o situáciu, keď sa umožní konkrétnym používateľom prideliť rozsiahlejšie prístupové práva oproti bežným používateľom).

Vytvorte model riadenia API
Implementácia open API vždy ovplyvní aktivity realizované v rámci interných procesov mesta, preto jeho riadenie musí byť dobre naplánované, aby bol plne využitý jeho potenciál (aby z neho mesto mohlo čo najviac profitovať a zároveň aby ho používatelia pokladali za stabilné a zaujímavé).

Každé mesto by si malo naplánovať interný model riadenia API už v ranom štádiu. Musia byť jasne stanovené roly a zodpovednosti týkajúce sa správy API, pretože poskytovanie open API je súčasťou služieb mesta. Musí byť dodržané, že vlastníctvo API nemôže byť outsourcované. Interný model riadenia na úrovni mesta musí brať do úvahy minimálne nasledovné otázky:

- Kto je vlastník služby?
- Kto je zodpovedný za údržbu obsahu?
- Kto je zodpovedný za vývoj obsahu?
- Kto je zodpovedný za technickú údržbu?
- Kto je zodpovedný za technický rozvoj?
- Kto je zodpovedný za komunikáciu?

Keď mestá harmonizujú svoje API medzi sebou, tiež sa vyžaduje model riadenia API medzi mestami. Cieľom je vytvorenie jedného spoločného (medzimestského) modelu riadenia API. Aj v prípade, že si každé z miest udržiava svoje vlastné harmonizované API, aj tak sa musia spoločne dohodnúť na úprave nasledovných oblastí:

[57] About COSS - COSS.fi

- Kto vlastní API definície?
- Ako bude financovaný vývoj a údržba?
- Ako bude koordinovaný vývoj?
- Ako sa bude API v praxi vyvíjať?
- Kto bude zodpovedný za správu verzií?

Pri replikovaní API medzi mestami by sa malo vziať do úvahy, že by mal existovať spoločný model riadenia, ktorý by umožnil, aby všetky harmonizované API boli dostupné na jednom mieste s potrebou jediného prihlásenia.

Vždy, keď je to možné, používajte otvorené licencie

Pre open data a open API by sa mali používať otvorené licencie, pokiaľ je to možné. Mali by byť vybrané už zavedené licencie. Podmienky používania údajov, podmienky používania API ako aj kód API sú licencované oddelene.

Podmienky používania údajov by mali byť licencované na základe licencií Creative Commons. Podmienky používania open API by mali podporovať jednoduchú implementáciu z pohľadu používateľa. Tiež definujú podmienky používania a implementácie API v rámci individuálnych systémov. Zdrojový kód API by mal byť otvorene licencovaný, pokiaľ tomu nič nebráni.

Treba poznamenať, že sú možné rôzne kombinácie licencií (napríklad je možné, aby API rozhranie bolo otvorené, ale na údaje, ktoré sú prostredníctvom neho dostupné, sa vzťahujú obmedzenia). Cieľom je však zabezpečiť čo najvyššiu možnú mieru otvorenosti, pričom ak je otvorenosť z nejakého dôvodu zakázaná, toto musí byť zaznamenané spolu s odôvodnením.

Harmonizujte dátové formáty, dátový model alebo API
Škálovateľnosť služieb a ich rozšírenie na trh je jednoduchšie, ak sú open API a dátové formáty harmonizované medzi mestami. Odporúča sa publikovať harmonizované open API aj v prípade, ak ešte nie je možné vytvoriť jednotný dátový model. Ak niektoré údaje nemôžu byť získané zo zdrojového systému, tieto polia zostanú prázdne, pričom harmonizácia dátového modelu môže pokračovať aj po publikovaní open API. Cieľom je zabezpečiť zverejnenie údajov jednotným spôsobom do tej miery, ako je to maximálne možné.

Všetky spoločne implementované harmonizácie by mali byť komplexne zdokumentované a dostupné.

Zamerajte sa na efektívne SLA
SLA pre open API vychádza z charakteru API a jeho špecifických potrieb, pričom cieľom je poskytovanie takého API, na ktorého prevádzkyschopnosť sa môžu používatelia (interní aj externí) spoľahnúť. V prípade chyby potrebujú mať používatelia prístup k čo najaktuálnejšej verzii údajov.

Životný cyklus API by mal byť navrhovaný rovnakým spôsobom ako životný cyklus systému, pričom do úvahy by sa mal brať životný cyklus celej služby. V prípade, že životnosť zdrojového systému pre API sa blíži ku koncu, musí sa naplánovať aj to, akým

Tento projekt je podporený z Európskeho sociálneho fondu.

spôsobom sa API nahradí alebo stiahne tak, aby to čo najmenej poškodilo iné systémy alebo externé strany.

Uprednostňujte otvorené nástroje
Vždy, keď je to možné, na technickú špecifikáciu API a verziovanie zdrojového kódu, by mali byť použité otvorené nástroje. Na technickú špecifikáciu API sa obyčajne používajú všeobecne uznávané otvorené jazyky. Zároveň s cieľom vyhnúť sa tzv. vendor lock-inom je potrebné pre open API manažment použiť open source riešenia.

Vytvorte pokyny a distribuujte ich otvorene
Tvorba kvalitných aktuálnych pokynov a dokumentácií umožní zvýšenie kompetentnosti súvisiacej s API a distribúciu open API. Toto si zároveň vyžaduje určenie zodpovedností a dostatočných zdrojov. Keď sa API obstaráva, výzva na predloženie ponuky musí obsahovať aj zmienku o tom, že súčasťou objednávky je aj dokumentácia vrátane informácie o tom, ako má byť podrobná. Je potrebné uviesť, že vytvárané pokyny sú určené pre rôznych adresátov:

- technická dokumentácia pre vývojárov softvéru (vrátane príkladov kódu a iných materiálov, ktoré uľahčia používanie),
- popisy procesov a ďalšie informácie potrebné pre stranu implementujúcu systém,
- pokyny pre obstarávanie a prípadný vzor dokumentu pre stranu, ktorá pripravuje objednávku,
- pokyny pre ostatný personál, ktorý musí byť informovaný o tom, ako open API ovplyvní ich dennodenné pracovné aktivity.

Usporiadajte školenia k open API
Malo by sa usporiadať vyškolenie personálu aby sa zabezpečilo, že API bude možné efektívne využiť a zároveň, že doplní existujúci prevádzkový model. Personál by mal byť zároveň schopný poskytnúť podporu vývojárom, ktorí o API prejavia záujem.

Napríklad ľudia, ktorí budú mať na starosti spracovanie spätnej väzby, musia byť vyškolení na riešenie spätnej väzby, resp. musia byť schopní odpovedať na spätnú väzbu odoslanú prostredníctvom nového API. Zároveň ľudia zodpovední za objednávku systému musia byť vyškolení na to, ako monitorovať kvalitu objednaného API.

Tento projekt je podporený z Európskeho sociálneho fondu.

4 Návrh odporúčaní

Vzhľadom na rastúci vplyv strojovej spracovateľnosti a API v súvislosti s digitálnou transformáciou vlád, ktorá je významným spôsobom podporovaná a zdôrazňovaná aj na úrovni EÚ (pozri kapitola 2), je nevyhnutné aby aj v podmienkach Slovenska bolo vytvorené prostredie, ktoré maximalizuje ich potenciál či už interne (vo vzťahu k orgánom verejnej moci) alebo externe (vo vzťahu k tretím stranám v podobe občanov, firiem a iných zainteresovaných subjektov). Snahy Slovenska v tejto oblasti ako aj aktívny prístup MIRRI je nezanedbateľný (pozri podkapitola 2.5), pričom naplnenie niektorých odporúčaní uvedených nižšie, môže byť v praxi významne ovplyvnené už prebiehajúcimi aktivitami (napr. projekt CAMP).

Zvýšenie podpory a angažovanosti OVM
Súčasťou štandardizácie by mali byť usmernenia vytvorené na centrálnej úrovni (MIRRI), ktoré by poskytovali podporu pri zavádzaní API. Takéto usmernenia by mali zároveň obsahovať osvedčené postupy (best practices) týkajúce sa navrhovania API na základe všeobecne dostupných príkladov dobrej praxe.

Využiteľnosť API poskytovaných verejným sektorom je významne ovplyvnená spôsobom, ako sú API navrhnuté a popísané, preto by sa mal klásť dôraz na to, aby OVM postupovali v rámci týchto aktivít jednotne a aby využívali metódy, ktoré sú široko akceptované.

Zviditeľnenie poskytovaných API
Podiel elektronických služieb, ktoré publikujú open API cez API gateway by sa mal v najbližších rokoch významne zvýšiť (podľa NKIVS je cieľom do roku 2026 dosiahnuť u sledovaných elektronických služieb podiel 80%). Dôležitým aspektom na dosiahnutie tohto cieľa bude úspešnosť realizácie bežiacich projektov ako napríklad Centrálna API manažment platforma (CAMP) alebo Rozvoj platformy integrácie údajov (CIP) a manažment osobných údajov (MOU).

Základným predpokladom používania open API poskytovaných verejnou správou je vedomosť o tom, aké open API sú na Slovensku dostupné - preto by bolo vhodné už v súčasnosti vytvoriť jeden spoločný priestor, ktorý by tieto informácie poskytoval.

Príkladom podobne zameraného priestoru môže byť portál otvorených údajov, ktorý poskytuje informácie o tom, aké otvorené dáta sú dostupné v danom štáte. Obdobne aj pre API by mohol existovať priestor, kde by boli katalogizované API poskytované orgánmi verejnej moci spolu s uvedením cesty k týmto API (súčasný portál otvorených údajov poskytuje možnosť katalogizácie open API, pričom s touto možnosťou sa počíta aj v rámci pripravovanej novej verzie portálu).

Katalóg existujúcich open API poskytovaných verejnou správou by zároveň mohol slúžiť ako nástroj na eliminovanie neefektívnej práce a potenciálnej duplicity – preverenie už existujúcich API pri zamýšľaní vytvoriť nové API by umožnil vykonať rozhodnutie, že či pre ich potreby a ciele je možné využiť niektoré z už existujúcich API.

Tento projekt je podporený z Európskeho sociálneho fondu.

Vytvorenie centrálneho open API tímu
Pre zabezpečenie koordinácie všetkých aktivít súvisiacich s implementáciou a rozširovaním open API by bolo vhodné na centrálnej úrovni (MIRRI) vytvoriť kapacity (rozšíriť a/alebo špecializovať existujúce kapacity) pre aktivity prislúchajúce tzv. open API tímu v podobe:

- definovania a udržiavania/aktualizovania API stratégie,
- poskytovania podpory vytváraním, zverejňovaním a aktualizovaním usmernení pre navrhovanie, tvorbu a zavádzanie open API, vrátane poskytovania konzultácií v relevantných prípadoch,
- zbierania spätnej väzby s cieľom zapracovania relevantných podnetov do príslušných usmernení.

Napríklad, v súčasnosti je rôznymi pracovnými skupinami za účelom zverejňovania štandardov, usmernení a iných súvisiacich informácií používaný portál wiki.viceremier.gov.sk. Podobne by mal byť vytvorený spoločný priestor výlučne zameraný na oblasť API, kde by tzv. open API tím zverejňoval relevantné informácie.

Výsledkom tohto odporúčania nemusí byť nevyhnutne vytvorenie novej „pracovnej skupiny“ alebo nového portálu. Využité môžu byť už existujúce dostupné kapacity spomínaných pracovných skupín alebo aj pracovníkov MIRRI a rozšírenie existujúceho portálu wiki.vicepremier.gov.sk. Dôležité však je, aby v rámci týchto existujúcich zdrojov boli jasne definované a určené zodpovednosti vo vzťahu k aktivitám prislúchajúcim open API tímu, aby bola API stratégia jasne komunikovaná a podpora prehľadne realizovaná.

Pri dedikovaní kapacít za účelom zabezpečenia centrálneho open API tímu by bolo vhodné vziať do úvahy všetky už realizované alebo prebiehajúce aktivity (CAMP, CIP, MOU a pod.), aby príslušné roly a zodpovednosti boli realizované komplementárne, nie konkurenčne a duplicitne. Dôležitým cieľom je synergia a účelné využitie zdrojov miesto neefektívneho trieštenia úsilia.

Doplnenie štandardu open API pre moje dáta
Potrebné je doplniť štandard open API pre moje dáta na základe v súčasnosti známeho riešenia zvoleného v rámci MOU (Príloha A).

Poznámka: Súčasná verzia detailného návrhu riešenia MOU zohľadňuje aktuálne požiadavky v zmysle vyhlášky č. 78/2020 o štandardoch pre informačné technológie verejnej správy[58]. Zároveň by však bolo vhodné minimálne zvážiť štandardizáciu služby moje dáta v nadväznosti na aktualizovaný štandard (pozri podkapitola 5.1 vrátane príloh tohto dokumentu, napríklad vo vzťahu k použitiu Solid princípov a štandardov ako napríklad W3C Overiteľné poverenia http API) v rámci relevantnej úpravy vo vyhláške o štandardoch pre informačné technológie verejnej správy, prípadne prostredníctvom inej vhodnej normy.

Vyčlenenie finančných zdrojov
Aktivity smerujúce k zvyšovaniu strojovej spracovateľnosti údajov ako aj k zavádzaniu riešení podporujúcich súčasný trend posilnenia publikovania open API a s tým súvisiacu

[58] 78/2020 Z.z. - Vyhláška Úradu podpredsedu vlády Sl... - SLOV-LEX

Tento projekt je podporený z Európskeho sociálneho fondu.

digitálnu transformáciu, je potrebné chápať nie ako jednorazové projekty ale ako kontinuálne aktivity, ktoré budú pokračovať aj po ukončení individuálnych projektov s tým spojených.

Životné situácie, ktoré majú byť riešené a uspokojené vďaka digitálnej transformácii služieb verejnej správy, sa budú vyskytovať nepretržite, a preto je nevyhnutné dlhodobo presadzovať snahu o vyčlenenie dostatočných finančných zdrojov nielen ako jednorazových aktív (určených napríklad na implementáciu konkrétneho riešenia) ale aj ako dlhodobo dostupných prostriedkov (na zabezpečenie fungovania riešenia po jeho odovzdaní do prevádzky).

Dôležitú rolu v posilnení open API poskytovaných verejnou správou a rozširovaní ich používania (ako aj pri dosahovaní cieľov stanovených napríklad v NKIVS) zohrávajú okrem MIRRI aj jednotlivé OVM. Za týmto účelom by bolo vhodné, aby na úrovni OVM:

- boli zabezpečené kapacity potrebné pre identifikáciu, popísanie a zavádzanie open API. Aj na úrovni OVM sa odporúča vytvorenie open API tímu, ktorý by v rámci OVM mal v zodpovednosti túto oblasť. Tento tím by postupoval v súlade s open API stratégiou a usmerneniami publikovanými zo strany MIRRI. Zároveň by realizoval komunikáciu s používateľmi open API a získaval ich spätnú väzbu.

- aktívne pristúpili k implementácii konceptu súborov údajov s vysokou hodnotou (strojovo spracovateľné súbory údajov poskytované cez API s dokumentáciou API tiež v strojovo spracovateľnej podobe) v súlade s navrhnutým postupom (roadmap) v rámci výstupu *č.5.1.1 Štandardy pre zverejňovanie údajov verejnej správy vo formáte otvorených údajov*.

- bol dôraz kladený na strojovú spracovateľnosť predovšetkým zosúlaďovaním dátových modelov s centrálnym modelom údajov. Je prirodzené, že už existujúce dátové modely na úrovni OVM nie sú „zo dňa na deň“ skonsolidované s centrálnym modelom údajov, avšak minimálne v prípade zavádzania novej služby (systému) je nevyhnutné, aby OVM pri tejto príležitosti zároveň dbali na definovanie dátového modelu, ktorý využíva práve koncept centrálneho modelu údajov. Za týmto účelom by mali aktívne pristupovať k snahe o získanie zdrojov v rámci potenciálne dostupných zdrojov na to vyčlenených.

Tento projekt je podporený z Európskeho sociálneho fondu.

5 Aktualizácia štandardu pre strojovú spracovateľnosť a definovanie open API

5.1 Štandard open API pre moje dáta

Štandard pre open API pre Moje dáta[59] v súčasnosti nezohľadňuje existenciu osobného úložiska, preto je potrebné ho aktualizovať. Navrhovaná aktualizácia vychádza z detailného návrhu riešenia MOU.

Osobné úložisko
Osobné úložisko je súčasťou vládneho cloudu, každá dotknutá osoba prihlásená do MOU v ňom má vytvorený svoj vlastný priestor, kde sa ukladajú jej dáta v zašifrovanej podobe tak, že má k nim prístup iba vlastník účtu. Vlastník účtu môže zadaním súhlasu povoliť prístup k svojmu osobnému úložisku aj službe tretej strany. Služba tretej strany môže pristupovať ku konkrétnym osobným údajom vlastníka účtu v osobnom úložisku, ak sú splnené nasledovné podmienky:

- služba tretej strany je zaregistrovaná v MOU,
- vlastník účtu si priradil túto službu,
- vlastník účtu zadal súhlas tretej strany na prístup ku konkrétnym osobným údajom.

Komunikácia s externými systémami
Pre komunikáciu medzi internými a externými systémami sú podstatné nižšie uvedené scenáre:

- pripojenie služby prostredníctvom API rohzrania,
- práca s údajmi (čítanie dát z osobného úložiska na základe súhlasu, zápis dát do osobného úložiska na základe súhlasu),
- integrácia s autentifikačnou službou a službou pre súhlasy.

Štandardom komunikácie je:

- komunikácia prebieha pomocou REST volaní, formát údajov je JSON,
- prenos dát cez HTTPS protokol, certifikát bude vydaný dôveryhodnou certifikačnou autoritou,
- služby vystavené navonok sú popísané prostredníctvom open API.

Tretie strany sa integrujú so službami MOU pomocou SDK[60] (MyData Client SDK – komponent MOU), ktoré poskytuje rozhranie na pripojenie služby, autentifikáciu a prácu s osobnými údajmi. Ide o knižnicu na podporu implementácie MyData klientov, ktorá zabezpečuje kompatibilitu klienta s MOU riešením.

[59] Moje-Data-02-Standard-OPEN-API_v2-1.pdf (datalab.digital)
[60] Software Development Kit

Tento projekt je podporený z Európskeho sociálneho fondu.

Zodpovednosti SDK pre mobilné aplikácie:

- umožňuje integráciu do mobilných aplikácií tretích strán,
- poskytuje UI v rámci narábania s osobným úložiskom,
- umožnuje bezpečné overenie identity užívateľa voči službám MOU,
- umožnuje pripojenie služby tretej strany a zadanie súhlasu na prístup k osobným údajom,
- umožňuje zdieľanie požadovaného datasetu tretej strane.

Zodpovednosti SDK pre služby tretích strán:

- umožní integráciu do backendových aplikácií tretích strán (napríklad internet-bankingové aplikácie - načítanie listu vlastníctva, výpis časti daňového priznania a podobne),
- poskytne integráciu so serverovým riešením tretej strany,
- umožní autentifikáciu tretej strany pre prístup k službám MOU,
- zabezpečí načítanie zdieľaného datasetu, prípadne umožní zapísať dataset tretej strany do užívateľovho úložiska na základe súhlasu,
- umožní získať zoznam udelených súhlasov jednotlivými osobami a ich stav,
- sprostredkuje notifikácie v prípade zmien v datasetoch,
- zabezpečí šifrovanie prenášaných dát.

Súhlas
Súhlas je vyjadrením súhlasu so spracovávaním osobných údajov (právny základ). Zaznamenanie udeleného súhlasu (záznam súhlasu – Access Grant) sa realizuje prostredníctvom modulu správy súhlasov a je uchovávané v osobnom úložisku. Pre používateľa (vlastníka účtu) je dôležitá informácia, akých údajov sa súhlas týka:

- služba konzumenta údajov získava súhlas na čítanie vybraných údajov a špecifikácia účelu použitia údajov,
- služba poskytovateľa údajov získava súhlas na zapisovanie údajov do osobného úložiska.

Záznam stavu súhlasu definuje, v akom stave je záznam súhlasu a či teda možno na základe daného súhlasu spracovávať údaje. Táto informácia je následne zaslaná službám, ktoré majú povinnosť ju uchovávať.

Štandard súhlasu

Súhlasy sú dynamické, jednoducho pochopiteľné pre používateľov, strojovo čitateľné, štandardizované a riadené koordinovaným spôsobom. Štandardy pre manažment súhlasov sú postavené na štandardoch W3C. Pre riadenie prístupov k osobnému úložisku využíva Solid protokoly Access Control Policies (ACP) a Web Access Control (WAC)[61]. Záznam súhlasu je implementovaný ako „verifiable credential“. Nad rámec základných prístupových protokolov sú definované vlastnosti, ako: identifikácia subjektu súhlasu (cez WebID), časová platnosť a dôvod súhlasu. Mechanizmus API pre manažment súhlasov je postavený na štandarde UMA.

[61] https://solid.github.io/web-access-control-spec/

Tento projekt je podporený z Európskeho sociálneho fondu.

Bližší popis relevantných štandardov je súčasťou prílohy A.1 v podobe relevantných častí vybratých z Detailného návrhu riešenia MOU (DNR MOU) s cieľom ich lepšieho zviditeľnenia.

Hlavné transakcie viažuce sa k správe súhlasov sú:

- zadanie súhlasu,
- zrušenie súhlasu,
- obnovenie súhlasu,
- vyradenie služby tretej strany z registra služieb tretích strán,
- overenie záznamu súhlasu.

Transakcie sú podporované cez API, ktoré môžu využívať služby tretích strán. API vychádza zo štandardu W3C pre „verifiable credentials“[62] a nazývame ho Consent API. API obsahuje základné metódy pre ukladanie súhlasu (VC), ukladanie statusu o revokácii (záznamu o stave súhlasov), volanie cez query, podpisovanie a overovanie súhlasov.

Tretia strana sa môže na Consent API integrovať priamou integráciou, alebo využiť špeciálne pripravené SDK. Popis API pre interakcie servera a pre prístup k mojim dátam sú súčasťou príloh A.2 a A.3 v podobe relevantných častí vybratých z Detailného návrhu riešenia MOU (DNR MOU) s cieľom ich lepšieho zviditeľnenia.

5.2 Osvedčená prax pre open API

Táto kapitola navrhuje osvedčené postupy pre zavádzanie API, ktoré môžu byť použité na účely štandardizácie postupu (pozri kapitola 4) napríklad v podobe základu pre usmernenie od centrálneho API tímu.

Vytvorenie API tímu
Vytvorte tím, ktorý bude za API zodpovedný. Nevyhnutnou súčasťou pri tvorbe API tímu je určenie vlastníka API, ktorý je zodpovedný za:

- definovanie API (v súlade s potrebami používateľov API),
- zabezpečenie súladu API s cieľmi stanovenými v relevantných politikách,
- zabezpečenie synchronizácie medzi členmi tímu, ktorí API navrhujú a členmi tímu, ktorí API vyvíjajú,
- zabezpečenie riadenia API počas jeho životného cyklu.

Vlastník API ako vedúci API tímu zodpovedá za vytvorenie API tímu.

Design-first prístup
Uprednostnite design-first prístup pred code-first prístupom, to znamená, že samotnému vývoju (kódovaniu) predchádza návrh (dizajn).

Pri navrhovaní API je potrebné vždy realizovať nasledovné aktivity:

[62] https://www.w3.org/community/credentials/

Tento projekt je podporený z Európskeho sociálneho fondu.

- identifikujte, aké ciele majú byť prostredníctvom API dosiahnuté (ciele v rámci politík organizácie a ciele stanovené na základe národných politík a stratégií),
- identifikujte, kto bude API používať,
- definujte, čo má API robiť (za týmto účelom zozbierajte používateľské a biznis požiadavky).

Ak máte k dispozícii výstup z predchádzajúcich aktivít, je potrebné, aby ste si preverili, či je vytvorenie nového API relevantné. Cieľom je zistiť, či už neexistuje API, ktoré napĺňa identifikované potreby a ciele – opätovné použitie už existujúceho API je rýchlejšie a jednoduchšie ako vytváranie nového API. Zároveň tak zvýšite pravdepodobnosť, že nové API sa vytvárajú iba vtedy, keď je to naozaj potrebné.

Poznámka: Na preverenie už existujúcich API by mal slúžiť API katalóg (pozri kapitola 4, zviditeľnenie poskytovaných open API).

Open API dokumentácia
Vytvorte API špecifikáciu v strojovo spracovateľnej podobe prostredníctvom open API dokumentu v súlade s OAS (open API specification[63]) – minimálne vo verzii 3.0 a vyššej.

V súčasnosti je strojová spracovateľnosť údajov základným predpokladom pre fungovanie organizácií v digitálnom prostredí. Požiadavka strojovej spracovateľnosti sa neviaže iba k údajom, ku ktorým sa prostredníctvom API pristupuje, ale zároveň sa viaže aj k samotnému API vo forme strojovo spracovateľnej dokumentácie – preto je žiaduce API špecifikovať prostredníctvom OAS vo formáte JSON alebo YAML.

Vo vzťahu k dokumentácii je nevyhnutné, aby bola vytvorená jediná platná dokumentácia, inými slovami – udržiavajte jediný zdroj pravdy. Používatelia musia vedieť presne identifikovať, ktorá dokumentácia sa vzťahuje k vášmu API, pričom nie je vhodné poskytovať viacero zdrojov dokumentácie a zvyšovať tak riziko neaktuálnosti niektorého z nich. Vždy, keď je to možné dokumentáciu zverejnite.

V rámci dokumentácie používajte konzistentné názvy pre zdroje, vývojári by mali byť schopní prevziať názvy zdrojov z vášho API na základe kontextu. Operácie súvisiace so zdrojmi popisujte pomocou štandardných http metód, pričom vždy uvádzajte aj príklady odpovedí. Využívajte HTPP kódy pre odpovede, pričom k týmto kódom uvádzajte dostatočné informácie pre používateľa, aby ste mu v prípade chyby uľahčili prácu na jej odstránení – ak je to vhodné, poskytnite aj odkazy na súvisiacu relevantnú dokumentáciu.

Súlad s existujúcou legislatívou a štandardmi
Pri tvorbe API je nevyhnutné, aby boli dodržané všetky relevantné právne predpisy a platné štandardy. Vždy identifikujte, ktoré právne predpisy, technické štandardy a politiky súvisia s vaším API a ubezpečte sa, že vaše aktivity a výstupy sú s nimi v súlade.

[63] OpenAPI Specification v3.1.0 | Introduction, Definitions, & More

Tento projekt je podporený z Európskeho sociálneho fondu.

Poznámka: Pre lepšiu orientáciu v tejto oblasti by mal slúžiť spoločný priestor zameraný výlučne na API (pozri kapitola 4, vytvorenie centrálneho open API tímu), ktorý by mohol obsahovať prelinkovanie na relevantné právne predpisy a štandardy.

Spätná väzba
Spolupracujte s používateľmi API a získavajte od nich spätnú väzbu. Získavanie spätnej väzby je kľúčové pre zabezpečenie toho, aby API napĺňalo potreby používateľov. Preto je nevyhnutné spolupracovať s používateľmi API či už pri návrhu, testovaní ale aj po implementácii API s cieľom vyhodnotenia jeho relevantnosti.

Používatelia predstavujú cenný zdroj informácií aj pre identifikáciu potreby rozšírenia alebo úpravy API, preto by komunikácia s používateľmi mala prebiehať počas celého životného cyklu API. Okrem toho by ste mali zároveň proaktívne zisťovať aké sú potreby vašich používateľov napríklad prostredníctvom prieskumu najžiadanejších API a výsledky zohľadňovať pri tvorbe nových API.

Tento projekt je podporený z Európskeho sociálneho fondu.

A Prílohy

A.1 Popis štandardov súhlasu (MOU)

1. W3C Overiteľné poverenia http API (The W3C Verifiable Credentials API)

https://w3c-ccg.github.io/vc-api/

Pôvodná špecifikácia bola uverejnená komunitnou skupinou Credentials. Nie je to štandard W3C, ani nie je súčasťou W3C Standards Track.

Overiteľné poverenia poskytujú mechanizmus na vyjadrenie poverení na webe spôsobom, ktorý je kryptograficky bezpečný, rešpektujúci súkromie a strojovo overiteľný. Táto špecifikácia poskytuje dátový model a protokoly HTTP na vydávanie, overovanie, prezentáciu a správu údajov používaných v takomto ekosystéme. The VC Dátový Model definuje tri základné úlohy: vydavateľa (Issuer), overovateľa (Verifier) a držiteľa (Holder). Aktéri, ktorí plnia každú z týchto rolí, môžu používať niekoľko softvérových alebo servisných komponentov na realizáciu rozhrania VC API na výmenu VC. Každá rola sa spája s aplikáciou, službou a správcom špecifickým pre danú rolu, ako aj s vlastnou vyhradenou službou ukladacieho priestoru. Okrem toho môže vydavateľ spravovať aj službu stavu pre odvolateľné poverenia vydané vydavateľom.

Každá daná implementácia VC API sa môže rozhodnúť kombinovať ktorúkoľvek alebo všetky tieto komponenty do jednej funkčnej aplikácie. Hranice a rozhrania medzi týmito komponentmi sú definované v tejto špecifikácii, aby sa zabezpečila interoperabilita a nahraditeľnosť v celom ekosystéme v súlade s VC. Okrem toho sa implementátori môžu rozhodnúť sfunkčniť akúkoľvek danú úlohu v akomkoľvek počte aktívnych inštancií nasadeného softvéru.

Tieto komponenty definujeme nasledovne:

- Issuer / Verifier / Holder Aplikácia – Aplikácia Vydavateľa vykonáva pravidlá o tom, kto získa aké poverenia, vrátane toho, ako sú strany, ktoré vytvárajú alebo prijímajú tieto poverenia, overené a autorizované. Aplikácia Overovateľa komunikuje so službou overovania, aby najprv skontrolovala pravosť a včasnosť daného VC alebo VP, potom použije obchodné pravidlá overovateľa predtým, ako nakoniec prijme alebo zamietne túto VC alebo VP. Aplikácia držiteľa vykonáva obchodné pravidlá schvaľovania toku poverení pod kontrolou držiteľa, od vydavateľov po overovateľov.
- Issuer / Verifier / Holder Služba – Služba Vydavateľa prijíma žiadosti o vydanie VC z autorizovaných aplikácií vydavateľa a vráti dobre vytvorené, podpísané overiteľné poverenia. Služba Overovateľa prijíma žiadosti o overenie VC a VP a vráti výsledok kontroly ich dôkazov a stavu (ak je prítomný). Služba Držiteľa prijíma žiadosti o vytvorenie VP z voliteľnej množiny VC a vráti dobre vytvorenú, podpísanú VP obsahujúcu tieto VC.
- Služba Stavu – Služba stavu poskytuje prostriedky na zachovanie súkromia na publikovanie a kontrolu stavu akéhokoľvek VC vydaného vydavateľom. Služby

Tento projekt je podporený z Európskeho sociálneho fondu.

overovateľa používajú koncový bod stavu vydavateľa pre kontrolu včasnosti daného VC ako súčasť overovania.

- Issuer / Verifier / Holder Služba Ukladania – Od každého aktéra v systéme sa očakáva, že podľa potreby uloží svoje vlastné VC. Riešenia ukladania vydavateľa a overovateľa môžu alebo nemusia používať zabezpečené ukladanie údajov.

Issuer / Verifier / Holder Admin – Súčasť Správca je potvrdenie, že každá z ostatných súčastí potrebuje spôsob konfigurácie a správy bez toho, aby predpísala rozhrania alebo prostriedky tejto konfigurácie.

2. W3C Model vytvárania zoznamu zrušených poverení (The W3C Revocation List Model)

https://w3c-ccg.github.io/vc-status-rl-2020/

Pôvodná špecifikácia bola uverejnená komunitnou skupinou Credentials. Nie je to štandard W3C, ani nie je súčasťou W3C Standards Track.

Táto špecifikácia opisuje mechanizmus na zachovanie súkromia, priestorovo efektívny a vysokovýkonný mechanizmus na publikovanie stavu zrušenia VC. Pre vydavateľa VC je často užitočné prepojiť sa s miestom, kde môže overovateľ skontrolovať, či bolo poverenie zrušené. Existujú rôzne aspekty ochrany osobných údajov a výkonnosti, ktoré sa robia pri navrhovaní, publikovaní a spracovaní zoznamov zrušených poverení.

Táto špecifikácia navrhuje mechanizmus zoznamu zrušených poverení na základe bitstringu so silnými charakteristikami zachovania súkromia, ktorý je kompatibilný s architektúrou webu, je vysoko priestorovo efektívny a dobre sa požičiava sieťam distribúcie obsahu. Ako príklad použitia tejto špecifikácie je možné vytvoriť zoznam zrušených poverení, ktorý môže byť vytvorený pre 100 000 overiteľných poverení, čo je v najhoršom prípade približne 12 500 bajtov. V prípade, že bolo zrušených niekoľko stoviek poverení, veľkosť zoznamu je menšia ako niekoľko stoviek bajtov a zároveň poskytuje súkromie pre 100 000 jednotlivcov.

Na najzákladnejšej úrovni sú informácie o zrušení pre všetky VC vydané emitentom vyjadrené ako jednoduché binárne hodnoty. Emitent vedie zoznam všetkých VC, ktoré vydal. Každý VC je priradený k pozícii v zozname. Ak je binárna hodnota pozície v zozname 1, VC sa zruší, ak je 0, nie je zrušený. Jednou z výhod používania bitstringu je, že ide o vysoko komprimovateľný formát údajov, pretože v priemernom prípade zostane veľké množstvo poverení nedotknutých. Tým sa zabezpečia dlhé časti bitov, ktoré majú rovnakú hodnotu, a teda vysoko stlačiteľné pomocou kompresných techník, ako je ZLIB [RFC1950].

3. W3C Prepojené dátové doklady (The W3C Linked Data Proofs)

Ako štandard VC naďalej dozrieva a adopcia sa zvyšuje, to, čo sa začína objavovať, je sieť overiteľných údajov. Údaje na tomto "webe" pochádzajú z mnohých rôznych zdrojov a mnohých rôznych kontextov, takže je dôležité, aby sme mali niektoré spoločné normy na udržanie hygieny a kvality údajov, ktoré používame. Rozšírené používanie formátu, ako je JWT, vedie k hlbokým problémom s kvalitou údajov, keď chcete vybudovať

Tento projekt je podporený z Európskeho sociálneho fondu.

ekosystém na konzistentných a vysoko kvalitných prepojených údajoch. Keďže JWT zle reprezentuje kontext údajov, jeho užitočnosť je pomerne obmedzená.

Našťastie máme alternatívny model, ktorý prekonáva toto významné obmedzenie. Prepojené dátové doklady (*Linked Data Proofs,* tiež ako *LD-Proofs*) ponúkať niekoľko vylepšení okrem JSON. Hlavnou výhodou formátu JSON-LD, ktorý používa LD-Proofs, je to, že vychádza zo spoločného súboru sémantiky, ktoré umožňujú širšiu interoperabilitu ekosystému. Poskytuje štandardnú slovnú zásobu, vďaka čomu sú údaje prenosnejšie, ako aj ľahko konzumované a zrozumiteľné v rôznych kontextoch. Aby sme vytvorili prehľadávateľnú sieť overiteľných údajov, je dôležité, aby sme uprednostnili silné opätovné použitie schém údajov. Bez neho riskujeme vytvorenie systému, v ktorom sa na reprezentáciu rovnakých presných informácií používa mnoho rôznych dátových schém, čím sa vytvárajú druhy dátových síl, ktoré dnes vidíme na internete. JSON-LD robí sémantiku prvotriednym princípom, a preto je pevným základom pre konštrukciu implementácií VC.

JSON-LD je dnes široko prijatý na webe, pričom W3C hlási, že ho používa 30% webu a Google z neho robí de facto technológiu optimalizácie pre vyhľadávače. Pokiaľ ide o overiteľné poverenia, bolo by výhodné rozšíriť a integrovať prácu okolo VC s existujúcim rastúcim ekosystémom prepojených údajov.

4. The W3C Ed25519 Signature Suite

https://w3c-ccg.github.io/lds-ed25519-2020/

Pôvodná špecifikácia bola uverejnená komunitnou skupinou Credentials. Nie je to štandard W3C, ani nie je súčasťou W3C Standards Track.

Táto špecifikácia popisuje Ed25519 Signature Suite vytvorený v roku 2020 pre špecifikáciu LD-Proof. Signature Suite využíva podpisy Ed25519 EdDSA a multibase. Vo všeobecnosti používajú RDFDataset Normalization Algorithm na transformáciu vstupného dokumentu do jeho kanonickej podoby. Kanonická podoba sa potom hashuje a podpisuje pomocou oddeleného podpis algoritmu.

Signature Suite - Špecifikovaná sada kryptografických primitív zvyčajne pozostáva z kanonizačného algoritmu, algoritmu digestu správ a algoritmu podpisu, ktoré sú spojené kryptografmi pre vývojárov z bezpečnostných dôvodov.

Ed25519 Signature Suite 2020 sa MUSÍ používať v spojení s podpisovacími a overovacími algoritmami v špecifikácii LD-Proofs. Súprava sa skladá z nasledujúcich algoritmov:

Parameter	Hodnota	Špecifikácia
Algoritmus kanonizácie	https://w3id.org/security#URDNA2015	[RDF-DATASET-NORMALIZATION]
Algoritmus digestu správ	SHA-256	[RFC6234]

Tento projekt je podporený z Európskeho sociálneho fondu.

Algoritmus podpisu	Edwards-Curve Digital Signature Algorithm (EdDSA)	[RFC8032]

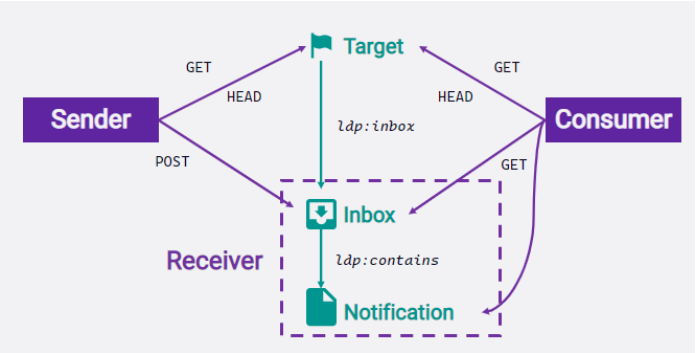
5. W3C Oznámenia o prepojených údajoch (The W3C Linked Data notifications)

https://www.w3.org/TR/ldn/

Oznámenia o prepojených údajoch je protokol, ktorý popisuje, ako servery tzv. Prijímače (*Receivers*) môžu mať správy, ktoré im tlačia aplikácie tzv. Vysielače (Senders), ako aj ako aj iné aplikácie tzv. Spotrebitelia (*Consumers*) môžú tieto správy načítať. Každý zdroj môže inzerovať prijímajúci koncový bod (*Inbox*) pre správy. Správy sú vyjadrené v RDF a môžu obsahovať akékoľvek údaje.

Oznámenia o prepojených údajoch (LDN) podporujú zdieľanie a opakované použitie oznámení v aplikáciách bez ohľadu na to, ako boli generované. To umožňuje modulárnejšie systémy, ktoré oddelia ukladanie údajov od aplikácií, ktoré zobrazujú alebo inak využívajú údaje. Protokol je určený na to, aby odosielateľom, príjemcom a spotrebiteľom oznámení, ktoré sú nezávisle implementované a prevádzkované na rôznych technologických riešeniach, umožnili bezproblémovú spoluprácu, čo prispieva k decentralizácii našich interakcií na webe.

Namiesto toho, aby sa s oznámeniami zaobchádzalo ako s pominuteľnými alebo pretrvávajúcimi entitami, táto špecifikácia umožňuje pojem oznámenia ako jednotlivej entity s vlastným identifikátorom URI. Oznámenia je možné získať a opätovne použiť. Odporúča sa autentifikácia a overovanie oznámení, ale mechanizmus je na uvážení príjemcov a spotrebiteľov, pretože potreby sa líšia v závislosti od typov oznámení a rôznych oblastí aplikácií.



Odosielatelia a spotrebitelia objavia adresu URL *inboxu* zdroja prostredníctvom vzťahu v hlavičke alebo tele zdroja HTTP Link.

Odosielateľ:

- Vytvorí telo oznámenia podľa potrieb aplikácie.
- Odošle upozornenie na adresu URL inboxu vykonaním žiadosti POST obsahujúcej telo v JSON-LD alebo v inej serializácii prijateľnej serverom.

Prijímač:

- Odpovedá na žiadosti o získanie inbox na adresu URL inboxu so zoznamom adries URL oznámení, ktoré boli predtým prijaté.
- Odpovedá na žiadosti GET vykonané na jednotlivé adresy URL oznámení pomocou JSON-LD (alebo voliteľne iných serializácií).
- Prijíma požiadavky POST na adrese URL inbox na vytvorenie oznámení.
- Voliteľne vynucuje obmedzenia upozornení odoslaných do inbox.

Spotrebiteľ:

- Načíta obsah adresy URL inbox so žiadosťou GET a používa sa podľa potrieb aplikácie.

6. Grant na spravovaný prístup používateľa 2.0 (User Managed Access Grant 2.0)

https://docs.kantarainitiative.org/uma/wg/rec-oauth-uma-grant-2.0.html

Táto špecifikácia definuje prostriedok pre klienta, ktorý zastupuje žiadajúcu stranu, použiť ticket na povolenie na vyžiadanie prístupového tokenu OAuth 2.0 na získanie prístupu k chránenému zdroju asynchrónne od času, keď vlastník zdroja povolí prístup. Táto špecifikácia definuje grant rozšírenia OAuth 2.0 [RFC6749]. Grant zvyšuje schopnosti OAuth nasledujúcimi spôsobmi:

- Vlastník zdroja povolí chránený prístup k zdrojom klientom používaným entitami, ktoré sú v úlohe žiadajúcej strany, čo umožňuje party-to-party autorizáciu.
- Autorizačný server a server prostriedkov interagujú s klientom a žiadajúcou stranou spôsobom, ktorý je asynchrónny, pokiaľ ide o interakcie vlastníkov zdrojov. To umožňuje vlastníkovi zdrojov nakonfigurovať autorizačný server s pravidlami udelenia autorizácie (podmienky politiky) podľa potreby, namiesto toho, aby povolil synchrónne vydanie prístupového tokenu hneď po overení.

Napríklad bankový zákazník (Vlastník zdrojov) Alice so službou bankového účtu (server zdrojov) môže použiť službu správy zdieľania (autorizačný server) hosťovaný bankou na správu prístupu k jej rôznym chráneným zdrojom manželom Bobom, účtovným profesionálom Charline a agregačnou spoločnosťou finančných informácií, a to všetko pomocou rôznych klientskych aplikácií. Každý z jej bankových účtov je chráneným zdrojom a dva rôzne rozsahy prístupu, ktoré na ne môže ovládať, sú prezeranie údajov o účte a prístup k platobným funkciám, kde:

- Vlastník zdrojov – Subjekt schopný poskytnúť prístup k chránenému zdroju
- Žiadajúca strana – Fyzická alebo právnická osoba, ktorá používa klienta na hľadanie prístupu k chránenému zdroju. Žiadajúca strana môže alebo nemusí byť tou istou stranou ako vlastník zdrojov
- Klient – Aplikácia, ktorá je schopná podať žiadosti o chránené zdroje so súhlasom vlastníka zdroja a v mene dožadujúcej strany
- Server prostriedkov – Server, ktorý hosťuje zdroje v mene vlastníka zdrojov a je schopný prijímať a reagovať na žiadosti o chránené zdroje
- Autorizačný server – Server, ktorý v mene vlastníka prostriedku chráni prostriedky hosťované na serveri prostriedkov.

Tento projekt je podporený z Európskeho sociálneho fondu.

Chyba! Nenašiel sa

Druhá voliteľná špecifikácia, [UMAFedAuthz], definuje prostriedok pre UMA-povolený autorizačný server a zdrojový server, ktorý sa má voľne spojiť, v kontexte vlastníka zdroja. Táto špecifikácia spolu s [UMAFedAuthz], predstavuje UMA 2.0.

A.2 Interakcie servera súhlasu

1. Zisťovanie koncových bodov

Ako príklad:
Koncový bod: https://consent.pod.inrupt.com/.well-known/vc-configuration

Aplikácia bude potrebovať mechanizmus na zistenie dvoch informácií pred začatím toku žiadosti:

- Umiestnenie alebo štruktúra zdroja údajov, ku ktorým sa požaduje prístup (cesta k údajom, ku ktorým požaduje aplikácia prístup). Aplikácia sa bude spoliehať buď na už existujúce znalosti (napríklad všetky zdroje označené ako ex:Photo typ alebo URL zo súkromného, neobjaviteľného zdroja) alebo aplikácia bude prechádzať cez Pod hierarchiu kontajnmentu.
- Miesto vydavateľa súhlasu na používanie. Solid používatelia by mali mať možnosť vybrať si, ktorému emitentovi dôverovať. Rovnako ako pri Solid-OIDC, Používatelia pridajú jednu alebo viac Triples do ich WebID profilov s formulárom: <WebID> solid:vcIssuer <issuer>. Ak sú uvedení viacerí emitenti, aplikácia pravdepodobne požiada používateľa, aby si vybral.

Akonáhle je emitent identifikovaný, koncové body pre tohto emitenta možno nájsť vykonaním kroku vyjednávania podobného vyjednávaniu OIDC: *append "/.well-known/vc-configuration"* na koniec URL adresy emitenta.

```
{
"@context": [ "https://www.w3.org/2018/credentials/v1", "https://consent.pod.inrupt.com/credentials/v1" ],
"issuer": "https://consent.pod.inrupt.com/issue", "verifier": "https://consent.pod.inrupt.com/verify",
"query": "https://consent.pod.inrupt.com/query",
...
}
```

2. Žiadosť o súhlas

Koncový bod:

```
POST /issue
```

Táto súčasť je definovaná W3C CCG.

Ak by aplikácia chcela prístup k definovanej množine zdrojov, a to buď ich explicitným uvedením, alebo vyžiadaním tvaru údajov aplikácia najprv vytvorí dokument JSON-LD, ktorý sa má prezentovať koncovému bodu */issue*. Základnú štruktúru tejto žiadosti definuje W3CCCG.

V kontexte Solid bude tento koncový bod chránený tak, ako je popísané v časti Autorizácia.

Okrem toho tvar údajov tvoriacich túto požiadavku bude obmedzený serverom súhlasu, ako je popísané v časti Obmedzenie tvaru. Tieto údaje budú ako také zahrnuté do žiadosti:

- WebID Žiadateľa o Súhlas,
- Úroveň prístupu požadovaná z hľadiska acl:mode (napríklad READ alebo WRITE),
- Idp:inbox kde je možné kontaktovať tohto žiadateľa s platným súhlasom,
- Adresy URL zdrojov, ku ktorým sa žiada prístup,
- Účel žiadosti,
- Časový rámec, počas ktorého by bol súhlas platný.

Príklad žiadosti:

```
POST /issue
Content-Type: application/json Authorization: DPoP <access-token> DPoP: <proof>
{
 "credential": {
  "@context": [
   "https://www.w3.org/2018/credentials/v1",
   "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "https://vc.mou.dev.cloud/credentials/3732",
  "type": [
   "VerifiableCredential",
   "SolidCredential",
   "SolidConsentRequest"
  ],
  "expirationDate": "2022-10-25T03:21:58.708Z",
  "credentialSubject": {
   "id": "https://solid.mou.dev.cloud/requestingParty/profile/card#me",
   "inbox": "https://solid.mou.dev.cloud/requestingParty/inbox/",
   "hasConsent": {
    "mode": [
     "http://www.w3.org/ns/auth/acl#Read",
     "http://www.w3.org/ns/auth/acl#Write"
    ],
    "hasStatus": "https://w3id.org/GConsent#ConsentStatusRequested",
    "forPersonalData": [
     "https://solid.mou.dev.cloud/john-doe/dataset/mydata"
    ],
    "forPurpose": "https://example.com/SomeSpecificPurpose"
   }
  }
 }
}
```

Výsledkom tejto HTTP operácie bude podpísané VC:

```
HTTP/2 201
Content-Type: application/json
Location: https://consent.pod.inrupt.com/vc/53973727-f4d0-9e8dbdc041fd
{
 "@context": [
  "https://www.w3.org/2018/credentials/v1",
  "https://www.w3.org/2018/credentials/examples/v1"
 ],
 "id": "https://vc.mou.dev.cloud/credential/599fbe62-ab9b-4eb3-ae10-240474e0eef0",
```

```
"type": [
 "VerifiableCredential",
 "SolidCredential",
 "SolidConsentRequest"
],
"expirationDate": "2022-10-25T03:21:58.708Z",
"credentialStatus": {
 "id": "https://vc.mou.dev.cloud/status/xCZZ#0",
 "revocationListCredential": "https://vc.mou.dev.cloud/status/xCZZ",
 "revocationListIndex": "0",
 "type": "RevocationList2020Status"
},
"credentialSubject": {
 "id": "https://solid.mou.dev.cloud/requestingParty/profile/card#me",
 "inbox": "https://solid.mou.dev.cloud/requestingParty/inbox/",
 "hasConsent": {
  "mode": [
   "http://www.w3.org/ns/auth/acl#Read",
   "http://www.w3.org/ns/auth/acl#Write"
  ],
  "hasStatus": "https://w3id.org/GConsent#ConsentStatusRequested",
  "forPersonalData": [
   "https://solid.mou.dev.cloud/john-doe/dataset/mydata"
  ],
  "forPurpose": "https://example.com/SomeSpecificPurpose"
 }
},
"issuer": {
 "id": "https://vc.mou.dev.cloud"
},
"issuanceDate": "2022-02-08T06:10:41.296Z",
"proof": {
 "created": "2022-02-08T06:10:41.297Z",
 "type": "JsonWebSignature2020",
 "proofPurpose": "assertionMethod",
 "verificationMethod": "https://vc.mou.dev.cloud/key/ovsDKYBjFemIy8DVhc-w2LSi8CvXMw2AYDzHj04yxkc",
 "jws": "eyJhbGciOiJFZERTQSIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..sDChclhU0jOqYoLjSwiNzlxerroUv4g-
T1xTwrXxA9EktzPkrLT_awIWl623aVE7NUNA3IMwbZxvYdUQzQZ6DQ"
 }
}
```


3. Oznámenie o súhlase

Koncový bod: (rôznorodý, definovaný aplikáciou)
Táto súčasť je definovaná v špecifikácii oznámenia o prepojených údajoch.

Klientska žiadosť, ktorá požiadala o VC (Žiadosť o Súhlas) od emitenta, môže byť predložená vlastníkovi zdrojov, v tomto prípade:https://pod.inrupt.com/alice/profile/card#me. Tento klient je zodpovedný za odoslanie tejto žiadosti vlastníkovi zdrojov, ktorý je zabalený ako VP. Doručenie tejto žiadosti sa môže uskutočniť prostredníctvom *LDN inbox* alebo prostredníctvom iného mechanizmu dohodnutého na základe aplikácie. Napríklad:

```
POST /alice/inbox/
Content-Type: application/ld+json
{
"@context": [ … ], "type": [
"VerifiablePresentation", "SolidConsentRequestPresentation" ], "verifiableCredential": [
… Consent Request content … ], "proof": {
… Bob's app signs this request with an keypair managed by Bob's app … }
```

```
}
```

Ak klientska aplikácia nie je schopná doručiť VP, aplikácia sa môže rozhodnúť odvolať žiadosť o súhlas.

Rovnaký proces oznamovania sa vzťahuje na udelení súhlasu s jedným ďalším krokom. Keďže toky súhlasu sa zvyčajne iniciujú so Žiadosťou o Súhlas, je zodpovednosťou Žiadateľa o Súhlas zahrnúť *ldp:inbox* miesto, kde je možné odoslať odpoveď. Je to táto *inbox* hodnota, ktorá sa používa na určenie miesta pre POSTovanie Udelenia Súhlasu.

4. Udelenie súhlasu

Koncový bod:

```
POST /issue
```

Táto súčasť je definovaná W3C CCG.

Keď Vlastník Zdroja dostane Žiadosť o Súhlas, aplikácia príjemcu najprv overí žiadosť. To možno vykonať na overovacom koncovom bode servera súhlasu, na inom serveri alebo úplne v kontexte samotnej aplikácie. Mechanika validácie je určená prepojeným dátovým dôkazom, ktorý sa používa, napríklad Ed25519Signature2020.

Po overení aplikácia zobrazí používateľovi podrobnosti Žiadosti o Súhlas, v ktorých sa uvádzajú adresy URL prostriedkov a/alebo akékoľvek metaúdaje o zdrojoch, ktoré zodpovedajú tejto požiadavke.
V tomto bode môže vlastník zdrojov prostredníctvom klientskej aplikácie vykonať ktorýkoľvek z nasledujúcich krokov:

- Úplne prijať žiadosť, ako bola predložená Žiadateľom o Súhlas,
- Čiastočne prijať žiadosť predloženú Žiadateľom o Súhlas,
- Odmietnuť žiadosť,
- Ignorovať žiadosť.

O štvrtej položke (ignorovanie žiadosti) nie je čo povedať. Všetky ostatné akcie sa riadia rovnakým základným modelom:

- extrahovať údaje zo Žiadosti o Súhlas,
- požiadať o vydanie nového overiteľného poverenia na základe údajov uvedených v Žiadosti o Súhlas.

Odmietnutie jednoducho nastaví *hasStatus* vlastnosť na "*ConsentStatusRefused*". Prijatie Žiadosti o Súhlas bude používať túto štruktúru:

```
POST /issue Content-Type: application/json Authorization: DPoP DPoP:
{
"credential": { "@context": [
"https://www.w3.org/2018/credentials/v1", "https://consent.pod.inrupt.com/credentials/v1" ],
"type": [
"VerifiableCredential", "SolidCredential", "SolidConsentGrant" ], "issuanceDate": "2021-05-26T16:40:03",
"expirationDate": "2021-06-23T16:40:03", "credentialSubject": {
"id": "https://pod.inrupt.com/alice/profile/card#me", "providedConsent" : {
"mode": ["Read","Write"],
"hasStatus": "ConsentStatusExplicitlyGiven",
```


Tento projekt je podporený z Európskeho sociálneho fondu.

"forPersonalData": "https://pod.inrupt.com/alice/private/data", "forPurpose" : "https://example.com/SomeSpecificPurpose",
"isConsentForDataSubject":
"https://pod.inrupt.com/bob/profile/card#me" } }
}

Vlastník Zdrojov nie je povinný prijať Žiadosť o Súhlas bez zmeny. Vlastník zdrojov môže napríklad udeliť súhlas so zdrojom, ale v obmedzenejšej lehote, v tomto prípade: Dva týždne namiesto štyroch a iba READ prístup namiesto READ/WRITE.

Content-Type: application/json Authorization: DPoP <access-token> DPoP: <proof>
{
"credential": { "@context": [
"https://www.w3.org/2018/credentials/v1", "https://consent.pod.inrupt.com/credentials/v1"],
"type": [
"VerifiableCredential", "SolidCredential", "SolidConsentGrant"], "issuanceDate": "2021-05-26T16:40:03",
"expirationDate": "2021-06-09T16:40:03", "credentialSubject": {
"id": "https://pod.inrupt.com/alice/profile/card#me", "providedConsent" : {
"mode": ["Read"],
"hasStatus": "ConsentStatusExplicitlyGiven",
"forPersonalData": "https://pod.inrupt.com/alice/private/data", "forPurpose" : "https://example.com/SomeSpecificPurpose",
"isConsentForDataSubject":
"https://pod.inrupt.com/bob/profile/card#me" } }
}

Prípadne môže Vlastník Zdrojov jednoducho zamietnuť štvortýždňovú žiadosť o prístup (namiesto vrátenia upraveného súhlasu), ale to je detail ponechaný na implementáciu klientskej aplikácie.

Rovnako ako pri Žiadostiach o Súhlas, koncový bod */issue* servera súhlasu vytvorí nové VC pre klientsku aplikáciu. A rovnako ako pri Žiadostiach o Súhlas, tento koncový bod vyžaduje autentifikáciu.

Ako je popísané v časti Autorizácia vyššie, Udelenia Súhlasu môžu byť vydané len v mene Vlastníka Zdrojov. Ak sa agent pokúsi udeliť súhlas pre zdroje, ktorým agent nie je Vlastníkom Zdrojov, žiadosť zlyhá.

Akonáhle má klientska aplikácia k dispozícii tento VC, aplikácia odošle oznámenie Žiadateľovi o Súhlas prostredníctvom *inbox* vlastnosti z pôvodnej žiadosti. Oznámenie funguje presne tak, ako je popísané vyššie v sekcii Oznámenie o Súhlase.

5. Overenie súhlasu

Koncový bod:

```
POST /verify
```

Tento koncový bod je definovaný W3C CCG.

Zatiaľ čo Služba Súhlasu ponúkne overovací koncový bod, aplikácia môže použiť akýkoľvek externý systém na overenie VC alebo VP. V skutočnosti môže aplikácia vykonať toto overenie úplne na strane klienta. Validácia je dôležitým krokom v rôznych fázach Solid Toku Súhlasov, a klientska aplikácia by mala mať možnosť vybrať si, ako a kde sa toto overenie vykonáva.

Tento projekt je podporený z Európskeho sociálneho fondu.

Koncový bod */verify* sprístupnený službou súhlasu sa riadi CCG API definíciou a môže byť použitý na kontrolu VC alebo VP. Okrem overovacích kontrol vymedzených v používanom podpisovej súprave ako aj kontrola možného zrušenia, tento koncový bod pridá tri Obmedzenia špecifické pre Solid pre VC overenie.
Pri overovaní Solid VC:

- Skontrolujte, či pole typu obsahuje "*SolidCredential*"
- Skontrolujte, či pole *credentialSubject.id* je *WebID*
- Pri dereferencovaniu identifikátora WebID je zdrojom dokument RDF obsahujúci triple: <WebID> solid:vcIssuer <issuer> tak, aby <issuer> sa rovnal hodnote poľa emitenta vo VC.

Vyššie uvedené kroky sa vzťahujú na všetky vložené Solid VC: Tie VC, ktoré tiež majú pole *proof.domain* rovné reťazcu "*Solid*". Všetky ostatné VC sa považujú za bežné VC. Príklad výmeny je uvedený nižšie:

```
POST /verify
Content-Type: application/json
{
 "verifiableCredential": {
  "@context": [
   "https://www.w3.org/2018/credentials/v1",
   "https://www.w3.org/2018/credentials/examples/v1"
  ],
  "id": "https://vc.mou.dev.cloud/credential/a79e2b46-c7d4-4613-b7e3-d93f637fbb29",
  "type": [
   "VerifiableCredential",
   "SolidCredential",
   "SolidConsentRequest"
  ],
  "expirationDate": "2022-10-25T03:21:58.708Z",
  "credentialStatus": {
   "id": "https://vc.mou.dev.cloud/status/xCZZ#0",
   "revocationListCredential": "https://vc.mou.dev.cloud/status/xCZZ",
   "revocationListIndex": "0",
   "type": "RevocationList2020Status"
  },
  "credentialSubject": {
   "id": "https://solid.mou.dev.cloud/requestingParty/profile/card#me",
   "inbox": "https://solid.mou.dev.cloud/requestingParty/inbox/",
   "hasConsent": {
    "mode": [
     "http://www.w3.org/ns/auth/acl#Read",
     "http://www.w3.org/ns/auth/acl#Write"
    ],
    "hasStatus": "https://w3id.org/GConsent#ConsentStatusRequested",
    "forPersonalData": [
     "https://solid.mou.dev.cloud/john-doe/dataset/mydata"
    ],
    "forPurpose": "https://example.com/SomeSpecificPurpose"
   }
  },
  "issuer": {
   "id": "https://vc.mou.dev.cloud"
  },
  "issuanceDate": "2022-02-07T16:05:14.666Z",
  "proof": {
   "created": "2022-02-07T16:05:14.666Z",
   "type": "JsonWebSignature2020",
```


Tento projekt je podporený z Európskeho sociálneho fondu.

```
  "proofPurpose": "assertionMethod",
  "verificationMethod": "https://vc.mou.dev.cloud/key/ovsDKYBjFemIy8DVhc-w2LSi8CvXMw2AYDzHj04yxkc",
  "jws":
"eyJhbGciOiJFZERTQSIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..Fpr1thmowrDeXhLFLkdJVdAlXcqTCn6aQ_T6vhXeJHejM8zK
1eWBVaOcnXHhgQ-BbhzfQL5n-oWd2bd-IxxdBw"
 }
 }
}
```

S úspešnou overovanou odpoveďou:

```
HTTP/2 200
Content-Type: application/json
{
 "checks": [
  "proof",
  "expirationDate",
  "credentialStatus"
 ],
 "warnings": [],
 "errors": []
}
```

6. Odvolanie súhlasu

Koncový bod:

```
POST /status
```

Tento koncový bod je definovaný W3C CCG

Pozrite si prosím W3C Revocation List 2020 Report viac informácií o podrobnostiach odvolania súhlasu. Toto je chránený koncový bod, ktorý vyžaduje prístupový token. Klient môže zrušiť poverenia pre VC prostredníctvom POST:

```
POST /status
Content-Type: application/json Authorization: DPoP <access-token> DPoP: <proof>
{
 "credentialId": "https://vc.mou.dev.cloud/credentials/3732",
 "credentialStatus": [
  {
   "type": "RevocationList2020Status",
   "status": "0"
  }
 ]
}
```

S nasledovnými možnými odpoveďami:

Code	Popis
200	Stav VC bol úspešne zmenený
400	Chybná požiadavka
404	Záznam VC nebol nájdený
500	Interná chyba systému

Tento projekt je podporený z Európskeho sociálneho fondu.

7. Autorizácia založená na súhlase

UMA 2.0 špecifikácia definuje množinu interakcií, pomocou ktorých môžu klienti získať prístupové tokeny pre konkrétne zdroje. Keď klient pôvodne požiada o zdroj na Pod, 401 odpoveď bude zahŕňať *WWW- Authenticate header* s nasledujúcimi informáciami:

```
HTTP/2 401 Unauthorized
WWW-Authenticate: UMA realm="Solid Pod" as_uri="https://uma.inrupt.com" ticket="eyJraWQiOiI0..."
```

as_uri poučí klienta, kde sa nachádza autorizačný server UMA, a *ticket* sa použije na priradenie pôvodnej žiadosti k následnému prístupovému tokenu.

Ticket bude podpísaný JWT obsahujúce informácie o požadovanom zdroji spolu s rozumnou hodnotou *exp* (expirácie). Autorizačný server bude vykonávať štandardné JWT overenie pomocou verejného kľúča emitenta, ktorý bol nakonfigurovaný na používanie.

Počas fázy žiadosti o Token žiadajúcej strany (tiež ako Requesting Party Token, RPT), klient poskytne niekoľko informácií. Po prvé, štandardný DPoP-viazaný Solid prístupový token bude zahrnutý do hlavičiek požiadavky.

Vlastnosť *grant_type* bude URL-kódovaná hodnota *urn:ietf:params:oauth:grant-type:uma-ticket*. *ticket claim* bude hodnota z ticketu uvedeného v odpovedi vyššie. Vlastnosť *claim_token* bude obsahovať BASE64-kódovaný VP, a vlastnosť *claim_token_format* bude obsahovať URL-kódovanú hodnotu https://www.w3.org/TR/vc-data-model/#json-ld.

```
POST /token
Authorization: DPoP <access-token> DPoP: <proof>
grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Auma-ticket &ticket=eyJraWQiOiI0...
&claim_token=eyJAY29udGV4dCI... &claim_token_type=https%3A%2F%2Fwww.w3.org%2FTR%2Fvc-data-model%2F%23json-ld
WebID prístupového tokenu musí byť v súlade s WebID ktorý bol poskytnutý v poskytnutom VP. Podobne zdroj pripojený k parametru ticketu musí byť v súlade s udelením súhlasu: ak je súhlas určený pre konkrétny zdroj, táto adresa URL zdroja sa musí zhodovať.
Úspešná výmena vytvorí prístupový token, ktorý môže klient použiť s Pod serverom:
HTTP/2 200 OK
Content-Type: application/json
{
"access_token": "            ",
"token_type": "DPoP"
}
```

Výsledný prístupový token bude obsahovať nároky identifikujúce typy VC, ktoré boli zahrnuté do výmeny VP: (typ), URL adresa vydavateľa autorizačného servera (iss) a cieľ zdroja (aud). Tento prístupový token nebude obsahovať WebID nárok.

Token vydaný na základe súhlasu:

- neobsahuje **webId** claim
- obsahuje claim **aud**, ktorý definuje dáta, ku ktorým sa môže pristúpiť (napr. "aud": "https://pod.inrupt.com/user/private/data")

- obsahuje claim **mode** ktorý definuje typ prístupu (napr. "mode": ["Read", "Append"])
- claim **iss** odkazuje na dôveryhodný UMA server, ktorý vydal tento token
- claim **kid** odkazuje na id kľúča použitého na podpísanie tohoto tokenu

A.3 API pre prístup k mojím dátam

Údaje dotknutej osoby (datasety) sa nachádzajú v osobnom úložisku a MOU aplikácia k nim pristupuje prostredníctvom nasledujúceho API (MOU-FP 55).

Poznámka: všetky volania na prístup k mojim dátam sú zabezpečené a môže k nim pristupovať iba autentifikovaný užívateľ.

1. Načítanie datasetu

Koncový bod:

```
GET /{podName}/dataset/{fileName}
```

Služba vráti požadovaný dataset.

Parametre služby:

- podName: Názov PODu, napr. 'PCO123456'
- fileName: Názov súboru pod ktorým je dataset uložený, začiatok napr. datasetId

Príklad volania služby:

```
GET https://solid.mou.dev.cloud/2100214426/dataset/informacie-o-fo-
z-rfo_identifikatory
Headers:
Authorization: DPoP | Bearer <access-token>
DPoP: <dpop-token>
```

Odpoveď volania služby:

```
{
    "@context": "https://contexts.mou.dev.cloud/contexts/mou-
common",
    "metadata": {
        "@type": "DatasetMetadata",
        "datasetId": "informacie-o-fo-z-rfo_identifikatory",
        "name": "Identifikátory",
        "dataProvider": "Ministerstvo vnútra SR",
        "category": [
            {
                "@type": "DatasetCategory",
                "label": "Identifikačné údaje",
                "description": "",
                "icon": ""
            },
            {
                "@type": "DatasetCategory",
                "label": "Štátne registre",
                "description": "",
                "icon": ""
            }
        ],
        "synchronizationType": "oneTime",
```


Tento projekt je podporený z Európskeho sociálneho fondu.

```
        "synchronizationFrequency": null,
        "state": null,
        "lastSynchronizationDate": "2023-02-16T01:02:38.011+00:00",
        "protectedDataset": {
            "protected": "eyJlbmMiOiJBMjU2R0NNIn0",
            "unprotected": {
                "alg": "ECDH-ES+A256KW"
            },
            "recipients": [
                {
                    "header": {
                        "kid": "2100214426",
                        "apu": "g3oz1pk0GJvx5_xuP577AsONwQxAc-
9s7nV2LzD-v1eRMeAfirN3rODBUJlP7QpLaeH3uVOm3nusZFXPEDeccg",
                        "epk": {
                            "kty": "EC",
                            "crv": "P-256",
                            "x":
"6hLKnaGo3d7PlZ5I8FAlS_mPHwdr86aey1wghV4oC-g",
                            "y": "M01M1VLrNj3wLlNb5d-
Skvj3LeUnmoH1wc8-6qzvklg"
                        }
                    },
                    "encrypted key": "nYr744UDxOGjTzjiVb9S90SVeEFc-
wpxrKRRk1rM5HBE9bnxvUMHPw"
                }
            ],
            "iv": "nj1SjesnqFWPuoyD",
            "ciphertext": "RQP7i5RRzgCbu9YTE7gZ_mD6vmccYUTtd-
qty8g7r8YdYwmQAkf8BXOBBV8c21tr74e-GzUvvlOBA-...",
            "tag": "dY2s6uKJTWmmElpl9OxRUA"
        }
    }
}
```


2. Nahratie datasetu

Koncový bod:

```
PUT /{podName}/dataset/{fileName}
```

Služba tretej strany môže nahrať dataset do určeného kontajnera s nastavenými právami. Kontajner dataset je vytvorený pri vytváraní podu. POST request uloží telo requestu do súboru z URL requestu.

Parametre služby:

- podName: Názov PODu, napr. 'PCO123456'
- fileName: Názov súboru pod ktorým je dataset uložený, začiatok napr. datasetId

Obsah správy:

- obsah datasetu

Príklad volania služby:

```
PUT https://solid.mou.dev.cloud/2100214426/dataset/informacie-o-fo-
z-rfo identifikatory
Headers:
Authorization: DPoP | Bearer <access-token>
DPoP: <dpop-token> (ak je zapnute DPoP)
Content-Type: application/ld+json
Body:
{
  "@context": "https://contexts.mou.dev.cloud/contexts/mou-common",
  "metadata": {
      "@type": "DatasetMetadata",
      "datasetId": "informacie-o-fo-z-rfo identifikatory",
      "name": "Identifikátory",
      "dataProvider": "Ministerstvo vnútra SR",
      "category": [
          {
              "@type": "DatasetCategory",
              "label": "Identifikačné údaje",
              "description": "",
              "icon": ""
          },
          {
              "@type": "DatasetCategory",
              "label": "Štátne registre",
              "description": "",
              "icon": ""
          }
      ],
      "synchronizationType": "oneTime",
      "synchronizationFrequency": null,
      "state": null,
      "lastSynchronizationDate": "2023-02-16T01:02:38.011+00:00",
      "protectedDataset": {
          "protected": "eyJlbmMiOiJBMjU2R0NNIn0",
          "unprotected": {
              "alg": "ECDH-ES+A256KW"
          },
          "recipients": [
              {
                  "header": {
                      "kid": "2100214426",
                      "apu": "g3oz1pk0GJvx5 xuP577AsONwQxAc-
9s7nV2LzD-v1eRMeAfirN3rODBUJlP7QpLaeH3uVOm3nusZFXPEDeccg",
                      "epk": {
                          "kty": "EC",
                          "crv": "P-256",
                          "x":
"6hLKnaGo3d7PlZ5I8FAlS_mPHwdr86aey1wghV4oC-g",
                          "y": "M01M1VLrNj3wLlNb5d-Skvj3LeUnmoH1wc8-
6qzvklg"
                      }
                  },
                  "encrypted_key": "nYr744UDxOGjTzjiVb9S90SVeEFc-
wpxrKRRk1rM5HBE9bnxvUMHPw"
              }
```


Tento projekt je podporený z Európskeho sociálneho fondu.

```
          ],
          "iv": "nj1SjesnqFWPuoyD",
          "ciphertext": "RQP7i5RRzgCbu9YTE7gZ_mD6vmccYUTtd-
qty8g7r8YdYwmQAkf8BXOBBV8c21tr74e-GzUvvlOBA...",
          "tag": "dY2s6uKJTWmmElpl9OxRUA"
      }
  }
}
```

3. Filtrovanie a vyhľadávanie datasetov

Koncový bod:

```
GET /{podName}/dataset/filter?{query}
```

Služba vráti zoznam datasetov vyhovujúcich zadanému filtru.

Parametre služby:

- podName: Názov PODu, napr. 'PCO123456'
- query: SPARQL dopyt na získanie vyfiltrovaného zoznamu datasetov

Príklad volania služby:

```
GET
https://solid.mou.dev.cloud/2100214426/dataset/filter?query=PREFIX
moucmn: <https://mojedata.gov.sk/vocab/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?resourceUri WHERE {
  {
    {
      SELECT ?resourceUri ?metadataNode ?reccnt ?id {
          {
              SELECT (count(*) as ?reccnt) ?id {
                  ?id moucmn:datasetMetadata ?metadataNode.

              } GROUP BY ?id
          }
          OPTIONAL { ?id moucmn:resourceUri ?resourceUri. }
          ?id moucmn:datasetMetadata ?metadataNode.
      }
    }
    ?metadataNode moucmn:datasetProvider ?value.
    FILTER (regex(str(?value), "meno", "i"))
  }
  UNION {
    {
      SELECT ?resourceUri ?metadataNode ?reccnt ?id {
          {
              SELECT (count(*) as ?reccnt) ?id {
                  ?id moucmn:datasetMetadata ?metadataNode.

              } GROUP BY ?id
```


Tento projekt je podporený z Európskeho sociálneho fondu.

```
          }
          OPTIONAL { ?id moucmn:resourceUri ?resourceUri. }
          ?id moucmn:datasetMetadata ?metadataNode.
      }
    }
    ?metadataNode dcterms:title ?value.
    FILTER (regex(str(?value), "meno", "i"))
  }
  UNION {
    {
      SELECT ?resourceUri ?metadataNode ?reccnt ?id {
          {
              SELECT (count(*) as ?reccnt) ?id {
                  ?id moucmn:datasetMetadata ?metadataNode.

              } GROUP BY ?id
          }
          OPTIONAL { ?id moucmn:resourceUri ?resourceUri. }
          ?id moucmn:datasetMetadata ?metadataNode.
      }
    }
    ?metadataNode moucmn:datasetCategory/rdfs:comment ?value.
    FILTER (regex(str(?value), "meno", "i"))
  }
  UNION {
    {
      SELECT ?resourceUri ?metadataNode ?reccnt ?id {
          {
              SELECT (count(*) as ?reccnt) ?id {
                  ?id moucmn:datasetMetadata ?metadataNode.

              } GROUP BY ?id
          }
          OPTIONAL { ?id moucmn:resourceUri ?resourceUri. }
          ?id moucmn:datasetMetadata ?metadataNode.
      }
    }
    ?metadataNode moucmn:datasetId ?datasetIdMetadata.
    ?userConfig a moucmn:datasetUserConfig .
    ?userConfig moucmn:datasetId ?datasetIdMetadata.
    ?userConfig moucmn:datasetTags ?value.
    FILTER (regex(str(?value), "meno", "i"))
  }
  UNION {
    {
      SELECT ?resourceUri ?metadataNode ?reccnt ?id {
          {
              SELECT (count(*) as ?reccnt) ?id {
                  ?id moucmn:datasetMetadata ?metadataNode.

              } GROUP BY ?id
          }
          OPTIONAL { ?id moucmn:resourceUri ?resourceUri. }
          ?id moucmn:datasetMetadata ?metadataNode.
      }
```


Tento projekt je podporený z Európskeho sociálneho fondu.

```
    }
    ?metadataNode moucmn:datasetId ?datasetIdMetadata.
    ?userConfig a moucmn:datasetUserConfig .
    ?userConfig moucmn:datasetId ?datasetIdMetadata.
    ?userConfig moucmn:datasetNote ?value.
    FILTER (regex(str(?value), "meno", "i"))
  }
}
```

Odpoveď volania služby:

```
{
  "@context":"http://www.w3.org/ns/ldp",
  "count":3,
  "contains":[
     "https://solid.mou.dev.cloud/2100214426/dataset/informacie-o-
fo-z-rfo meno",
     "https://solid.mou.dev.cloud/2100214426/dataset/new2-
informacie-o-fo-z-rfo_meno-a-priezvisko_meta",
     "https://solid.mou.dev.cloud/2100214426/dataset/informacie-o-
fo-z-rfo_meno_meta"
  ]
}
```

Tak isto ako datasety, tak aj notifikácie sa nachádzajú v osobnom úložisku. Na prístup k notifikáciám sa využíva nasledovné API.

4. Načítanie všetkých notifikácií

Koncový bod:

```
GET /{podName}/inbox
```

Služba vráti zoznam všetkých notifikácií v inboxe používateľa.

Parametre služby:

- podName: Názov PODu, napr. 'PCO123456'

Príklad volania služby:

```
GET https://solid.mojedata-test.gov.sk/2100214426/inbox/
Authentication: Bearer|DPoP <access-token>
```

Odpoveď volania služby:

```
{
  "@context":"http://www.w3.org/ns/ldp",
  "count":13,
  "contains":[
     "https://solid.mojedata-test.gov.sk/2100214426/inbox/6dea63c2-
605c-4e0c-9dea-663e915d3b6e",
     "https://solid.mojedata-test.gov.sk/2100214426/inbox/fcc4a606-
42bb-4ffc-9faf-df314501ea4d",
```

```
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/8420bb54-
6a7b-46fa-9e06-7a8499005955",
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/aa682a2c-
8da7-4765-a349-d622976ccbe6",
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/034978cb-
1bfe-4b12-be84-1615133fea98",
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/b711f3b7-
604b-449a-97e7-a7167584596e",
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/9e93e5fa-
7695-4b99-a80c-5920dc4b1cfd",
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/a705b37e-
9a70-4f28-b402-d230841b55e7",
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/a61657bf-
2977-422f-b8e9-e98d0d0a39b5",
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/be6925e0-
4d18-4462-aae2-994e4c08f350",
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/ada7a54a-
d72a-475e-8a0f-97f3f7583676",
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/e3f92549-
c97a-4f1a-96b0-5bab2e8cf98c",
    "https://solid.mojedata-test.gov.sk/2100214426/inbox/4ebdd356-
0441-491a-a543-a5e2357021c6"
  ]
}
```

5. Načítanie notifikácie

Koncový bod:

```
GET /{podName}/inbox/{id}
```

Služba vráti zoznam všetkých notifikácií v inboxe používateľa.

Parametre služby:

- podName: Názov PODu, napr. 'PCO123456'
- id: Identifikátor notifikácie

Príklad volania služby:

```
GET https://solid.mojedata-test.gov.sk/2100214426/inbox/6dea63c2-
605c-4e0c-9dea-663e915d3b6e
Authentication: Bearer|DPoP <access-token>
```

Odpoveď volania služby:

```
{
  "@context":"https://contexts.mojedata-test.gov.sk/contexts/mou-
common",
  "@type":"Message",
  "observedDate":"2023-02-03T07:06:19.008698738Z",
  "source":"https://solid.mojedata-test.gov.sk/d1f44fb3-70d3-4b52-
9ea7-bef21276e043/profile/card#me",
  "target":"2100214426",
  "messageType":"moucmn:DataTransferPublicSectorInformation",
  "payload":{
```


Tento projekt je podporený z Európskeho sociálneho fondu.

```
      "@type":"MessagePayload",
      "createdDate":"2022-10-23T11:56:05.838494Z",
      "fromIS":"https://data.gov.sk/id/egov/isvs/191",
      "toIS":"https://data.gov.sk/id/egov/isvs/546",
      "oe":"Register fyzických osôb (RFO)",
      "subject":"Osobné dáta prenesené medzi orgánmi verejnej moci",
      "message":"Vaše osobné údaje boli zdieľané medzi orgánmi
verejnej správy: • Zdieľané z IS: Ministerstvo vnútra SR, • Zdieľané
do IS: Sociálna poisťovňa, • Objekt evidencie: Register fyzických
osôb (RFO)"
   }
}
```

6. Nahratie (vytvorenie) notifikácie

Koncový bod:

```
POST /{podName}/inbox
```

Služba vytvorí notifikáciu. Obsah request body by mal byť JSON-LD.

Parametre služby:

- podName: Názov PODu, napr. 'PCO123456'

Obsah správy:

- obsah notifikácie

Príklad volania služby:

```
POST https://solid.mojedata-test.gov.sk/2100214426/inbox/
Authentication: Bearer <operator-access-token>
Content-Type: application/ld+json
X-API-Key:
Body:
{
  "@context":"https://contexts.mojedata-test.gov.sk/contexts/mou-
common",
  "@type":"Message",
  "observedDate":"2023-02-03T07:06:19.008698738Z",
  "source":"https://solid.mojedata-test.gov.sk/d1f44fb3-70d3-4b52-
9ea7-bef21276e043/profile/card#me",
  "target":"2100214426",
  "messageType":"moucmn:DataTransferPublicSectorInformation",
  "payload":{
     "@type":"MessagePayload",
     "createdDate":"2022-10-23T11:56:05.838494Z",
     "fromIS":"https://data.gov.sk/id/egov/isvs/191",
     "toIS":"https://data.gov.sk/id/egov/isvs/546",
     "oe":"Register fyzických osôb (RFO)",
     "subject":"Osobné dáta prenesené medzi orgánmi verejnej moci -
TEST",
     "message":"Vaše osobné údaje boli zdieľané medzi orgánmi
verejnej správy: • Zdieľané z IS: Ministerstvo vnútra SR, • Zdieľané
```

```
do IS: Sociálna poisťovňa, • Objekt evidencie: Register fyzických
osôb (RFO)"
  }
}
```

7. Filtrovanie a vyhľadávanie notifikácií

Koncový bod:

```
GET /{podName}/inbox/filter?{query}
```

Služba vráti zoznam notifikácií vyhovujúcich zadanému filtru.

Parametre služby:

- podName: Názov PODu, napr. 'PCO123456'
- query: SPARQL dopyt na získanie vyfiltrovaného zoznamu notifikácii

Príklad volania služby:

```
GET https://solid.mojedata-
test.gov.sk/holdertest2/inbox/filter?query=PREFIX moucmn:
<https://mojedata.gov.sk/vocab/>
SELECT ?id ?cnt ?resourceUri {
  {
    SELECT ?id ?created ?resourceUri WHERE {

      ?id moucmn:messagePayload ?payload.
      ?id moucmn:resourceUri ?resourceUri.
      ?id <http://purl.org/dc/terms/created> ?created.
      ?id moucmn:messageType ?messageType.

      FILTER (?messageType = moucmn:NotificationAccessRequest)

    } ORDER BY desc(?created)
      LIMIT 20
      OFFSET 0
  }
  {
    SELECT (count(*) as ?cnt) WHERE {

      ?notification a moucmn:Message.
      ?notification moucmn:messageType ?messageType.

      FILTER (?messageType = moucmn:NotificationAccessRequest)

    }
  }
}
```

Odpoveď volania služby:

```
{
  "@context":"http://www.w3.org/ns/ldp",
  "count":4,
```

```
  "contains":[
     "https://solid.mojedata-test.gov.sk/holdertest2/inbox/109e7d95-
247c-4b52-a04c-e8d6b1920e46",
     "https:// solid.mojedata-test.gov.sk
/holdertest2/inbox/5975b520-9983-486f-aa26-634825d977c1",
     "https:// solid.mojedata-test.gov.sk
/holdertest2/inbox/95fc2316-fd29-456c-bc44-cdc4f3cc090e",
     "https:// solid.mojedata-test.gov.sk
/holdertest2/inbox/28b529dd-7c27-42e7-ac6e-164a8d315aa7"
  ]
}
```

A.4 Doplnenie vysvetľujúcich častí pre interakcie serveru súhlasu a API pre prístup k mojím dátam

Autorizácia

Autorizačné služby súhlasu

Interakcia so Službou Súhlasu bude vyžadovať prístupové tokeny DPoP-viazané Solid-OIDC. Tieto prístupové tokeny budú uplatňovať webový prístup agenta, ktorý bude vždy tvoriť hodnotu výsledného VC *credentialSubject.id* pole.

Služba Súhlasu bude vydávať VC rôznych typov s rôznymi štruktúrami. Napríklad Žiadosť o Súhlas bude štruktúrovaná inak ako Potvrdenie o Súhlase. V prípade Potvrdení o Súhlase musí mať agent, ktorý žiadosť vyhovel konkrétnemu zdroju, *acl:Control* prístup k príslušnému zdroju alebo zdrojom.

Dátové cesty prepojené súhlasom (Consent Linked Data Paths)

Pre integráciu s Autorizáciou je možné definovať rôzne štruktúry pre potvrdenia o súhlase. V predvolenom nastavení, GConsent slovníku sa používa s *providedConsent / forPersonalData* cestou. Napríklad:

```
{
"@context": [ "https://www.w3.org/2018/credentials/v1", "https://consent.pod.inrupt.com/credentials/v1" ],
"credentialSubject": {
"id": "https://pod.inrupt.com/bob/profile/card#me", "providedConsent" : {
"hasStatus": "ExplicitlyGiven", "mode": ["Read"],
"isConsentForDataSubject": "https://pod.example/alice/profile/card#me", "forPersonalData":
"https://pod.inrupt.com/bob/private/data", "forPurpose" : "https://example.com/SomeSpecificPurpose" } }
}
```

V tomto príklade agent (https://pod.inrupt.com/bob/profile/card#me) bude potrebovať *acl:Control* prístup k zdroju https://pod.inrupt.com/bob/private/data aby mohol požiadať o vydanie tohto súhlasu.

Obmedzenia tvaru

Okrem vymedzenia konkrétnej prepojenej cesty údajov, ktorá sa používa na autorizáciu pri udelení súhlasu, je možné definovať požadované tvary prepojených údajov spolu s mapovaním týchto tvarov ku konkrétnym typom RDF.

Napríklad Žiadosť o Súhlas by vyžadovala nasledujúci tvar, ktorý by v prípade súladu s príslušným typom automaticky pridala do výsledného VC vydávajúcou službou.

```
PREFIX gc: <https://w3id.org/GConsent#> PREFIX acl: <http://www.w3.org/ns/auth/acl#> PREFIX ldp:
<http://www.w3.org/ns/ldp#>
PREFIX consent: https://consent.pod.inrupt.com/shapes#

consent:ConsentRequestShape { gc:forPurpose IRI+ ; gc:forPersonalData IRI+ ; gc:hasStatus gc:Requested ; acl:mode IRI+
}
consent:RequestShape {
gc:hasConsent @consent:ConsentRequestShape ; ldp:inbox IRI
}
```

Podobne by udelenie súhlasu vyžadovalo nasledujúci tvar:

```
PREFIX gc: <https://w3id.org/GConsent#> PREFIX acl: <http://www.w3.org/ns/auth/acl#>
PREFIX consent: https://consent.pod.inrupt.com/shapes#

consent:ConsentGrantShape { gc:forPurpose IRI+ ; gc:forPersonalData IRI+ ; gc:isConsentForDataSubject IRI ; gc:hasStatus
gc:ExplicitlyGiven ; acl:mode IRI+
}
consent:GrantShape {
gc:providedConsent @consent:ConsentGrantShape
}
```

Autorizácia Pod služby pomocou udelení súhlasu

Špecifikácia UMA 2.0 definuje množinu interakcií, pomocou ktorých môžu klienti získať prístupové tokeny pre konkrétne zdroje.

Existuje len veľmi málo, čo Solid pridáva k základnému toku založenému na UMA. V Solid kontexte UMA */token* koncový bod bude vyžadovať DPoP-viazaný Solid prístupový token a nasledujúce (inak nepovinné) vlastnosti:

- claim_token: BASE64-šifrovaný VP
- claim_token_format: hodnota kódovaná adresou URL https://www.w3.org/TR/vc-data-model/#json-ld

Úspešná interakcia vytvorí nový DPoP-viazaný (JWT) prístupový token, ktorý možno použiť na interakciu s príslušným zdrojom.

Zdrojový server po prijatí tohto prístupového tokenu s ním bude zaobchádzať inak ako zo Solid-OIDC prístupovými tokenmi. Najmä tieto tokeny nebudú obsahovať *WebID* nárok. Tieto tokeny identifikujú zdroj (napríklad "*aud*": "https://pod.inrupt.com/user/private/data") a špecifické povolené režimy prístupu (napríklad "*mode*": ["*READ*", "*APPEND*"]). Nárok emitenta (prostredníctvom

konfigurácie) ukáže na dôveryhodný server UMA. *kid* vlastnosť bude odkazovať na príslušný identifikátor kľúča z prostriedku JWKS autorizačného servera.

Politiky kontroly prístupu (ACP)

Pri používaní politík kontroly prístupu, prístup založený na udelení súhlasu môže byť povolený pre jednotlivé zdroje alebo hierarchie zdrojov. Pre ACP, existuje *VerifiableCredentialMatcher* trieda s *acp:requiresType* predikátom. Tento predikát sa môže použiť na zhodu s jedným alebo viacerými požadovanými VC podľa ich *rdf:type* vlastnosti. Napríklad:

```
@prefix acp: <http://www.w3.org/ns/solid/acp#> . @prefix consent: <http://www.w3.org/ns/solid/consent#> .
<#ConsentMatcher> a acp:Matcher ; acp:requiresType consent:SolidConsentGrant .
```

Viaceré typy VC môžu byť zložené takým spôsobom, že napríklad možno požadovať súhlas spolu s lekárskym poverením od konkrétneho emitenta.

JSON-LD kontext

Koncový bod: https://consent.pod.inrupt.com/credentials/v1
Služba súhlasu ponúka konfigurovateľný JSON-LD Kontext, napríklad na https://consent.pod.inrupt.com/credentials/v1. V tomto kontexte sa vymedzujú relevantné vlastnosti používané v kontexte Solid. Tento kontext sa MUSÍ použiť pri interakcii so službou. Napríklad:

```
{
"@context" : { "@version": "1.1", "@protected": true,
"gc": "https://w3id.org/GConsent#", "ldp": "http://www.w3.org/ns/ldp#", "acl": "http://www.w3.org/ns/auth/acl#", "inbox": {
"@id": "ldp:inbox",
"@type": "@id" }, "mode": {
"@id": "acl:mode",
"@type": "@id" }, "Read": {
"@id": "acl:Read",
"@type": "@id" }, "Write": {
"@id": "acl:Write",
"@type": "@id" }, "Append": {
"@id": "acl:Append",
"@type": "@id" }, "providedConsent": {
"@id": "gc:providedConsent",
"@type": "@id" }, "hasStatus": {
"@id": "gc:hasStatus",
"@type": "@id" }, "forPersonalData": {
"@id": "gc:forPersonalData", "@type": "@id" },
"forPurpose": {
"@id": "gc:forPurpose",
"@type": "@id" }, "hasConsent": {
"@id": "gc:hasConsent",
"@type": "@id" },
... }
}
```


Tento projekt je podporený z Európskeho sociálneho fondu.

Chyba! Nenašiel sa

Kontaktujte nás

Rudolf Sedmina
partner
Management Consulting
E rsedmina@kpmg.sk

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

Tento projekt je podporený z Európskeho sociálneho fondu.