ÚRAD PODPREDSEDU VLÁDY SR
PRE INVESTÍCIE
A INFORMATIZÁCIU

Pravidlá publikovania elektronických služieb do multikanálového prostredia verejnej správy

Finálna verzia

(Verzia 1-1)

Informácia o dokumente:

Názov:	Pravidlá publikovania elektronických služieb do multikanálového prostredia verejnej správy
Stav:	Finálna verzia
Pripravil:	Pracovná skupina K9.3 Architektúra
Verzia:	1.1
Dátum:	10.4.2018
Dátum revízie:	31.5.2018

Členovia pracovnej skupiny K9.3 Strategická architektúra:

Meno a priezvisko	Organizácia
Tomáš Kysela	ÚPPVII
Ivan Šajban	MV SR
Ľubomír Ivanov	NASES
Štefan Lompart	MF SR
Peter Miazdra, Ivan Krištek	DEUS
Rastislav Machel	MK SR
Karol Janeček	MS SR
Marián Šimegh	NCZI
Ján Tóbik	MŽP SR
Jozef Chren	ÚJD SR
Ján Suchal	Slovensko.Digital
Rastislav Cesnek	PPP
Vojtech Bálint	ITAS

História verzií

Verzia	Dátum verzie	Pripravil/ Zmenil	Pripomienkoval	Kľúčové zmeny
0.1	28.03.2018	Tomáš Kysela		Príprava dokumentu, osnova tém, príprava dokumentu na pripomienkovanie jednotlivých kapitol pracovnou skupinou

Verzia	Dátum verzie	Pripravil/ Zmenil	Pripomienkoval	Kľúčové zmeny
0.5	27.04.2018	Tomáš Kysela		Návrh textácie jednotlivých kapitol, príprava a prezentácia dokumentu na pracovnej skupine
1.0	14.05.2018	Tomáš Kysela	Všetci členovia skupiny	Zapracovanie pripomienok a finalizácia formátu dokumentu. Príprava na pripomienkovanie odbornou verejnosťou
1.1	31.05.2018	Tomáš Kysela	Zozbierané vstupy od odbornej verejnosti.	Zapracovanie pripomienok a finalizácia dokumentu.

Obsah

1. Úvod

Dva kľúčové strategické dokumenty informatizácie (*Strategická priorita: Integrácia a orchestrácia* a *Referenčná architektúra integrovaného informačného systému verejnej správy)*, zadefinovali pre všetky agendové systémy (poskytujúce elektronické služby verejnej správy) **povinnosť publikovať** (formou web služieb) **služby pre spracovanie elektronických podaní, a zároveň všetky pomocné služby** (doťahovanie údajov, validácia vstupov) pre prípravu samotného podania **do multikanálového prostredia**. Typicky ide o služby navrhnuté pre *synchrónnu* komunikáciu. Hlavnou výhodou povinnej publikácie služieb z AIS-ov je samozrejme ich širšia dostupnosť a využiteľnosť v prostredí eGovernmentu (dostupnosť pre všetky prístupové, resp. „kontaktné" body - aj novovznikajúce), no vzniká tak aj základný predpoklad pre otváranie týchto služieb (a ich rozhraní) pre využitie systémami tretích strán – tzv. koncept „OpenAPI".

So samotnou publikáciou prichádzajú dve dôležité otázky. Prvá z nich sa sústreďuje na oblasť štandardizácie „formy" publikovania služieb. Je namieste predpokladať, že pokiaľ budú služby publikované s využitím najnovších štandardov a metodík, a zároveň budú tieto štandardy a metodiky aplikované jednotne – zvyšuje sa dramaticky využiteľnosť daných služieb a zjednodušuje sa situácia pisateľom systémov, ktoré tieto služby konzumujú. Toto uvažovanie potvrdzujú aj štandardizačné aktivity z iných krajín EÚ (napríklad v UK publikovaná špecifikácia pre štandardizáciu API služieb https://www.gov.uk/guidance/gds-api-technical-and-data-standards). Druhou zaujímavou témou je centralizácia publikovania *fasády* uvedených služieb (ich rozhraní) do jedného logického bodu a možné výhody z toho vyplývajúce. Pod pojmom *fasáda* služby rozumieme rozhranie, ktoré je publikované v samostatnom (zväčša centralizovanom) bode a ktoré ďalej „smeruje" komunikáciu na cieľovú službu. Cez *fasádu* tak komunikácia len prechádza - od konzumenta smerom na službu a späť. Predsunutie fasády a jej centralizácia prináša niekoľko výhod – napríklad je možné centrálne merať a vyhodnocovať využívanie a výkonové parametre služieb. Zároveň je možné jednotne implementovať bezpečnostné politiky a protokoly – od autentifikácie, cez autorizáciu a prideľovanie oprávnení na vyvolávanie služieb v mene občana a podnikateľa – až po dynamickú ochranu pred kyber-útokmi (napr. DDoS), atď. Centrálny bod, do ktorého sú fasády rozhraní publikované, sa z pohľadu architektúry stáva dôležitou **vstupnou bránou** (preto ďalej technický pojem „API Gateway", resp. skrátene „API GW") pre využívanie služieb eGovernmentu a do značnej miery sa podieľa na vnímaní celkovej kvality poskytovaných elektronických služieb verejnej správy občanmi, podnikateľmi, resp. súkromným, či neziskovým sektorom.

Načrtnuté základné oblasti (štandardizácia formy rozhraní poskytovania služieb a špecifikácia úloh a zodpovednosť centrálnej brány publikujúcej fasády rozhraní služieb) sú premietnuté aj do hlavnej štruktúry dokumentu (dve základné kapitoly). Záverečná kapitola sa venuje vysvetleniu postupu implementácie z pohľadu OVM, keďže príprava služieb do jednotného formátu môže v čase výrazne predbiehať publikáciu fasády tohto rozhrania do centrálneho prvku (API GW).

Potrebu definovania metodiky a štandardov pre publikáciu služieb do multikanálového prostredia verejnej správy definuje jeden zo základných dokumentov NKIVS: *Detailný akčný plán informatizácie verejnej správy (2017-2020).* Dokument zaväzuje OVM postupovať v jeho zmysle pri rozvíjaní svojich agendových systémov, ako aj publikácií elektronických služieb pre občanov a podnikateľov. Dokument ukladá povinnosť štandardizačným pracovným skupinám ÚPPVII premietnuť obsah a závery z neho pri najbližšej aktualizácii výnosu Ministerstva financií Slovenskej republiky zo 4. marca 2014 č. 55/2014 Z. z. o štandardoch pre informačné systémy verejnej správy v znení neskorších predpisov.

2. Štandardizácia publikovania rozhraní pre elektronické služby VS do multikanálového prostredia

V súčasnosti dominantným spôsobom implementácie platformovo nezávislých elektronických služieb sú tzv. **web služby**, fungujúce na báze protokolu HTTP. Úlohou štandardizácie publikovania rozhraní pre elektronické služby verejnej správy sú tak rozhodnutia o technológiách, protokoloch, či štandardoch nad vrstvou základného protokolu HTTP. Pred tým, ako budú predstavené dva, v súčasnosti dominujúce prístupy v implementácii web služieb, sa prvé štandardizačné rozhodnutie dotýka priamo vrstvy základného protokolu (HTTP).:

2.1. HTTPS ako základný protokol pre publikáciu elektronických služieb

Elektronické služby verejnej správy zvyčajne pre svoje fungovanie zbierajú od svojich používateľov citlivé údaje (či už osobné, alebo napríklad o podnikaní), preto je dôležité, aby si tieto údaje neprečítal niekto nepovolaný počas ich prenosu z internetového prehliadača používateľa smerom na server, ktorý el. službu publikuje.

Rozhodnutie.: *Je nevyhnutné, 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).*

Zároveň je nutné v súvislosti s protokolom HTTPS aplikovať nasledovné.:

- TLS (Transport Layer Security) musí byť použité minimálne vo verzii 1.2
- HTTP dotazy by nemali byť transparentne presmerované na HTTPS.

Nasledujú usmernenia pre „základnú“ vrstvu implementácie web služieb.

2.2. REST/Restful ako jediný podporovaný prístup implementácie web služieb

V súčasnosti sú web služby implementované pomocou dvoch základných prístupov. Starším z nich je SOAP, ktorý je postavený na báze výmeny správ v štandarde XML, novším je REST/Restful v prevedení s výmenou správ v štandarde JSON. Výhodou novšieho prístupu, je širšia podpora zariadení (napríklad aplikácie v smartfónoch a tabletoch dokážu volať len REST služby) a aj jednoduchšie špecifikácie

autorizačných, či autentifikačných (nadstavbových) protokolov. Nevýhodou bol ešte donedávna pretrvávajúci nedostatočný formalizmus napríklad v popise rozhraní REST služieb, no dnes už existuje otvorený štandard (OpenAPI 3.0), podporený voľne dostupnými nástrojmi (napr. Swagger).

Otázkou teda zostáva, či v rámci usmernenia pre verejnú správu podporovať len jeden (a keď tak ktorý) spôsob publikácie web služieb, prípadne oba. Simultánna podpora oboch spôsobov predražuje implementácie v eGovernmente (najmä u tzv. centrálnych prvkov) a hovorí proti nej aj fakt, že prístup REST je dnes už v podstate plnohodnotnou, ale modernejšou alternatívou k prístupu SOAP.

Rozhodnutie.: *Vzhľadom na jeho vyššiu penetráciu a širšiu podporu zariadení (smartfóny a tablety), bude pre publikovanie služieb do multikanálového prostredia eGov podporovaný výhrade prístup na báze REST služieb.*

Zároveň je nutné aplikovať nasledovné.:

- Na modelovanie obsahu, dostupného cez REST API, sú využívané tzv. webové zdroje (dokumenty, entity, kolekcie).
- Každý webový zdroj, je identifikovaný pomocou svojho identifikátora – URI. Odporúčame URI definovať malými písmenami (a názvy spájať pomlčkami), tzv. *kebab-case*. Je zakázané v URI používať podčiarknutia a medzery.
    - Na modelovanie variabilných častí URI sú využívané tzv. URI šablóny (RFC 6570), napr.: **/osoby/{id}**
    - REST URI je už z definície bezvýznamový (môže to byť hocijaký textový reťazec), no pre lepšie komunikovanie účelu daného API je odporúčané využiť tzv. hierarchický dizajn identifikátorov, napr.: **/osoby/1453/podania/12**
- Asynchrónne služby sú povolené, aj keď sa nepredpokladá ich intenzívne využívanie pri službách publikovaných do multikanálového prostredia. Prípadné bližšie usmernenia z pohľadu štandardov a technológií tak budú doplnené neskôr, keď budú s ich používaním väčšie skúsenosti.
- V samotnom dotaze využívať nepovinnú hlavičku identifikujúcu danú top-level transakciu na spôsob „correlation-id“. Ak bude prítomná, systémy jej hodnotu budú pridávať do logov, čo uľahčí orientáciu aj analýzu prípadnej chyby v architektúre mikroslužieb.
- V rámci REST služieb sa používajú nasledovné typy webových zdrojov.:
    - *Dokument* reprezentuje jednu vec, jednu entitu (alebo logický agregát).
        - Názvoslovie: Podstatné meno v jednotnom čísle (môže mať čitateľný, alebo bezvýznamový – technický identifikátor), ale pokiaľ je naň referované cez *kolekciu,* odporúčame názov v množnom čísle.
        - Príklady URI: **/osoby/1453** alebo **/návody/ako-vybavit-obciansky**
    - *Kolekcia*, ktorá predstavuje množinu (viacerých) zdrojov.
        - Názvoslovie: Odporúčame podstatné meno v množnom čísle.
        - Príklady URI: **/osoby** alebo **/osoby/1453/podania**

- *Kontrolér*, predstavuje generický zdroj reprezentujúci „vykonateľnú akciu“ (čo samo o sebe prekračuje základný slovník protokolu HTTP).
  - Názvoslovie: Sloveso reprezentujúce danú akciu.
  - Príklady URI: **/osoby/1453/vytvor-spravu** alebo **/osoby/vyhladaj**

- Pre dopyty je potrebné využívať HTTP dopytové metódy pričom ak tieto nestačia, využíva sa *Kontrolér*.:
  - **GET** metóda musí byť využívaná na získanie reprezentácie webového zdroja, teda získanie *Dokumentu* alebo získanie *Kolekcie*.
    - **GET** metóda dotazu nemá telo. Odpoveď musí v tele obsahovať reprezentáciu zdroja. **GET** dotaz nesmie mať „vedľajšie“ účinky na daný zdroj - iné ako súvisiace so získaním zdroja. Ani pri viacnásobnom volaní a jej výsledok je potencionálne kešovateľný.
    - **GET** metóda dotazu môže obsahovať parametre, ktoré sa využívajú na filtrovanie, alebo stránkovanie v prípade *kolekcií*. Odporúčame využívať jednotný štýl cez celé API. Parametre dotazu je môžu slúžiť aj na filtrovanie, ktoré atribúty budú (alebo nebudú) poskytované v odpovedi (reprezentácia zdroja).
  - **POST** metóda je najuniverzálnejšou metódou protokolu HTTP. Využíva sa na posielanie informácií z klienta smerom na server.
    - **POST** dotaz musí obsahovať telo. Odpoveď naň môže obsahovať telo. **POST** request môže mať „vedľajšie účinky“, aj pri viacnásobnom volaní, môže byť potencionálne kešovateľný.
    - **POST** dotaz je typicky využívaný na vloženie nového webového zdroja do *kolekcie*.
    - **POST** je metódou, ktorá sa používa aj pri práci so zdrojom typu *kontrolér.*
    - **POST** dotaz môže byť využitý aj na filtrovanie a stránkovanie *kolekcií*, pokiaľ nie je možné, alebo je nepraktické využiť parametre dotazu (využíva sa v takom prípade podriadený *kontrolér*, pretože priame využitie *kolekcie* by viedlo ku kolízii s významom metódy pre pridávanie nového zdroja do kolekcie).
    - **POST** metóda sa nesmie využívať na aktualizáciu, resp. vymazanie webového zdroja, na to je potrebné využívať metódy **PUT** a **DELETE**.
  - **PUT** metóda je využívaná na vloženie, alebo aktualizáciu jedného *dokumentu*, alebo nahradenie celej *kolekcie*.
    - Vkladanie nového dokumentu pomocou **PUT** metódy implikuje, že klient je zároveň schopný navrhnúť URI pre nový zdroj (resp. nejakú identifikáciu). Ak to nedokáže, mal by vkladať nový webový zdroj cez **POST** metódu.
    - Pri vytváraní, server odpovedá pomocou **201 Created** s hlavičkou umiestnenia, kde je finálne URI (ak je iné, ako to čo bolo navrhnuté klientom). Odpoveď v tele môže niesť nejaký obsah.
    - Pri aktualizácii, server odpovedá buď s kódom **200 OK** (pričom vracia aktuálny stav entity), alebo **204 No Content** (v tele odpovede sa nevracia žiadny obsah, klient musí v prípade potreby znovu načítať webový zdroj), alebo **303 See Other** s presmerovaním na zdroj s aktuálnym obsahom (čo je rovnaké URI ako bolo využité pri PUT metóde, ale je potrebné znovu načítať cez GET).
      - To, akým spôsobom očakáva klient komunikáciu úspešnej operácie, môže byť uvedené pomocou atribútu PUT dotazu.

- **DELETE** metódu dotazu je potrebné využívať na odstránenie zdroja. Výsledkom je, že dané URI už viac nie je validné a dotazy naň by mali vracať **410 Gone** (**404 Not Found** je prípadne tiež akceptovateľné).
    - **DELETE** metóda nemá byť využívaná na akékoľvek „predbežné“ vymazanie (napríklad len zmena stavu na neplatný), teda operáciu, ktorej výsledok je možné vrátiť, resp. odvolať. Ak bol výsledok operácie **DELETE** úspešný, žiadny webový zdroj by sa nemal pod daným URI ďalej využívať (to samozrejme neznamená, že daná služby si striktne musí záznam vymazať zo svojho úložiska, stav týchto dát môže produkovať odpovede s kódom **410**, alebo **404**, len sa na to už nedá pozerať ako na ten istý zdroj pod tým istým URI.
- **HEAD** metóda dotazu sa využíva na získanie hlavičiek pre **GET** dotazy (pričom v odpovedi sa telo neuvádza).
- **OPTIONS** metóda dotazu sa využíva na získanie podporovaných metód dotazu pre aktuálny zdroj.
- **PATCH** metóda môže byť využívaná v prípade čiastočných aktualizácií. V takomto prípade musí byť používaný špecifický *content-type (application/merge-patch+json*) s „čiastočným“ telom z PUT dotazu.
    - V kontexte typických služieb, môže byť mechanizmus **PATCH** metódy pre implementujúceho príliš generický (musí ošetrovať príliš veľa možných prípadov volania), preto sa veľmi často pri čiastočnej aktualizácii využíva **POST** dotaz na sub-*kontrolér* so špecificky navrhnutým telom dotazu. Napríklad.: **/osoby/{id}/premenuj**

- Výsledky HTTP dopytov sú odpovede, ktoré pomocou špeciálnych kódov komunikujú štandardným spôsobom výsledok danej operácie. Je potrebné dodržať nasledovné pravidlá.:
    - V prípade úspechu danej operácie (a presmerovania).:
        - Vo všeobecnosti sa na komunikáciu úspešnej operácie využívajú kódy v rozsahu **2xx**. V prípade z nejakého dôvodu neúspechu pri operácii nie je povolené používať kódy v rozsahu **2xx**.
        - **200 OK** je najuniverzálnejším kódom komunikujúcim úspešný výsledok operácie v prípade, že výsledok obsahuje nejaké „telo“. Ak tomu tak nie je, potrebné použiť **204 No Content**
        - **201 Created** je najvhodnejším kódom odpovede v prípade vytvárania nového zdroja, pričom obsahuje hlavičku **Location** s URI adresou novovzniknutého webového zdroja. V prípade, že hlavičku s umiestnením neobsahuje, implicitne sa predpokladá, že URI dotazu je umiestnením novovzniknutého zdroja.
        - **202 Accepted** je najvhodnejším kódom odpovede v prípade asynchrónnych operácií a výhradne pri operáciách nad *kontrolérmi*. Nesmie obsahovať **Location** hlavičku.
        - **301 Moved Permanently** je indikáciou, že zdroj sa presunul na iné URI.
        - **303 See Other** môže byť použitý po spustení asynchrónnej operácie na presmerovanie na URI, kde je možné vyzdvihnúť výsledok (alebo stav) danej operácie. Klient sa môže rozhodnúť, kedy bude presmerovanie nasledovať (napr. pri snahe nezahlcovať komunikačné pásmo).
        - **304 Not Modified** kód je indikáciou, že zdroj nebol zmenený a môže byť využívaná nakešovaná reprezentácia.
    - V prípade neúspechu vyvolanej operácie, sú výsledkom tzv. chybové odpovede. Tie by okrem nižšie uvádzaných štandardných kódov, mali obsahovať aj správy vysvetľujúce príčinu chyby, resp. čo to znamená pre klienta.

- Kódy v rozsahu **4xx** indikujú chybu na strane klienta (nesprávne volanie) a komunikuje sa očakávanie, že klient svoj dotaz upraví tak, aby operáciu bolo možné vykonať (pričom poslanie znovu toho istého dotazu nedáva význam).
    - **400 Bad Request** je najuniverzálnejším kódom komunikujúcim neúspešnú operáciu z dôvodu zlého dotazu zo strany klienta. Nižšie uvádzané špecifickejšie kódy by mali byť využívané len v prípadoch, kde vyplývajú z použitých špecifikácií, alebo štandardov.
    - **401 Unauthorized** by sa mal využívať v prípade, že v dotaze chýbajú autorizačné informácie.
    - **403 Forbidden** je potrebné použiť v prípade, ak zdroj nie je pre daného klienta (dotaz autorizačné informácie obsahuje) dostupný z bezpečnostných dôvodov (nemá naň z nejakého dôvodu prístup).
    - **404 Not Found** by sa mal využívať v prípade, že na danom URI neexistuje žiadny webový zdroj.
    - **406 Not Acceptable** indikuje, že klient vyžaduje v odpovedi „content-type“, ktorý server nemôže (alebo nevie) produkovať.
    - **409 Conflict** indikuje nekonzistentnú aktualizáciu (napr. kvôli paralelnému vyvolaniu vloženia, alebo aktualizácie zdroja).
    - **410 Gone** indikuje, že zdroj na danom URI už nie je dostupný (a ani v budúcnosti dostupný nebude – príklad vymazaného zdroja pomocou DELETE metódy).
    - **415 Unsupported Media Type** by sa mal využívať v prípade, že klient posla dotaz v nepodporovanej reprezentácii (content-type).
    - **422 Unprocessable Entity** môže byť využitý v prípade ak formálne je dotaz v poriadku, ale validácia našla chybu na inom mieste (Napríklad využíva WebDAV, RFC 4918).
    - **429 Too Many Requests** je nutné využívať v prípade obmedzovania počtu volaní na webový zdroj.
- Kódy v rozsahu **5xx** naopak indikujú chybu na strane servera (kde znovu poslanie toho istého dotazu o malú chvíľu môže byť úspešné).
    - V zásade je odporúčané využívať univerzálny kód **500** na komunikáciu všetkých chýb, okrem tých, ktorým z použitých protokolov, alebo štandardov vyplývajú iné kódy z rozsahu **5xx** .

2.3. Pre všetky dotazy a odpovede z web služieb využívať štandard Unicode

Unicode je medzinárodný štandard, ktorého cieľom je definovať kódovaciu schému schopnú reprezentovať väčšinu znakov používaných v písaných jazykoch spolu s inými symbolmi. Unicode je dominantná kódovacia schéma používaná pri internacionalizácii softvéru a viacjazyčných prostredí..

Rozhodnutie.: *Povinnosť používať štandard Unicode Transformation Format (UTF-8) pre kódovanie všetkých dotazov a odpovedí v rámci služieb v celom prostredí eGov.*

2.4. JSON ako preferovaný formát pre telá dotazov a odpovedí v rámci REST web služieb

JSON (JavaScript Object Notation) je novší formát pre výmenu údajov v rámci dotazov a odpovedí web služieb, ktorý sa pre svoju jednoduchosť presadil najmä v spojení s REST web službami. Výhodou je najmä jeho jednoduchšia implementácia oproti predchádzajúcemu štandardu - XML (parsovanie údajov vo formáte JSON je jednoduchšie ako parsovanie údajov vo formáte XML). Je preto logické vyžadovať, aby nové služby boli vytvárané aplikovaním novšieho a jednoduchšieho štandardu. Zároveň však vzhľadom na to, že už existuje časť služieb, ktoré je potrebné publikovať do multikanálového prostredia eGovernmentu, a tieto služby boli písané s využitím štandardu XML pre výmenu údajov – je v takomto prípade povolené podporovať aj tento.

Rozhodnutie.: *Všetky nové publikované REST služby (do multikanálového prostredia eGov) budú využívať štandard JSON ako formát pre definovanie obsahu dotazov a odpovedí. Pre publikáciu existujúcich služieb (ktoré sa nebudú v dohľadnej dobe meniť) je možné použiť štandard XML.*

Zároveň je nutné v súvislosti s JSON formátom v dotazoch a odpovediach aplikovať nasledovné.:

- JSON formát implikuje identifikáciu obsahu (*content type*) ako **application/json**.
- Všade, kde to je možné, vytvárať dotazy a odpovede (web služieb) vo formáte JSON objektov a nie vo formáte JSON polí – polia môžu limitovať schopnosť rozhraní komunikovať metadáta o výsledkoch (operácie), resp. v budúcnosti spôsobovať zmeny na najvyššej úrovni API (ďalšie atribúty).
- Atribúty s „prázdnou“ hodnotou (**null**) môžu byť pri dotazoch, alebo odpovediach z web služby vypustené, toto chovanie však musí byť zdokumentované v špecifikácii služby (viď ďalšiu podkapitolu).
- Nezáleží na poradí atribútov v JSON objekte (dané samotnou JSON špecifikáciou).
- Zadefinovať (v dokumentácii služby) a implementovať konzistentné zoraďovanie záznamov v JSON poliach v snahe predchádzať chybám, ktoré vzniknú vďaka nepresným predpokladom zo strany klientov web služieb.

- Využívanie „obálok“ (Envelopes) nie je v rámci odpovedí žiadúce, najmä nie pre *Dokumenty*.
    - Je možné zvážiť využívanie „obálky“ pre pridanie informácie o stránkovaní pre *Kolekcie* (GET/POST metódy s významom vyhľadávania, resp. vypísania zoznamu).
        - V prípade využitia obálky, je nutné dáta dávať konzistentne pod definovaný atribút (typicky **data**).
    - Špeciálne atribúty v prípade jedného zdroja (typ *Dokument)* je možné definovať pomocou prefixu (napr. **_linky**)
- Všade kde je to možné, mal by byť podporovaný ľahko čitateľný formát a logické zoradenie JSON atribútov.

2.5. Špecifikácia a dokumentácia služieb pomocou štandardu OpenAPI 3.0

Aplikačné programové rozhrania (z angl. Application Programming Interface, skrátene API) tvoria základnú prepájaciu vrstvu pre všetky moderné informačné systémy. Bolo teda nevyhnutné, aby vznikol otvorený (bez závislosti od konkrétneho výrobcu), platformovo nezávislý a technologicky prenositeľný štandard pre definície (schému) REST API služieb. V zásade sa jedná o alternatívu WSDL súboru pre REST web služby. Štandard, v súčasnosti vo verzii OpenAPI 3.0, je spravovaný organizáciou The OpenAPI Initiative, ktorá združuje viaceré veľké technologické spoločnosti ako Google, Microsoft, IBM, Oracle a SAP.

Rozhodnutie.: *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 centrálnu bránu - WebAPI GW.*

Zároveň je nutné v súvislosti s definíciou OpenAPI dodržať nasledovné.:

- Každé publikované API musí byť zdokumentované podľa štandardu OpenAPI (verzia, podporované metódy a ich výsledky, atď. – viď vyššiu sekciu ohľadom REST služieb).
    - Súčasťou dokumentácie sú aj príklady volaní, ktoré plasticky dokresľujú, ako môžu vývojári dané API využívať.
- Každé publikované API je možné testovať v rámci tzv. „sandboxu“ (testovacie prostredia, vylúčená interakcia s produkčnými systémami). Pre lepšie testovanie, môžu byť súčasťou testovacieho prostredia aj sada predkonfigurovaných testovacích údajov, alebo objektov.
    - Účelom testovania je, aby používateľ API mohol skontrolovať, či vyvoláva dané API správne.
- V prípade rozvoja API je potrebné zabezpečiť, aby zmeny mali čo najmenší dopad na systémy, ktoré už dané API využívajú. S tým súvisí aj dostatočne predstihové plánovanie a označovanie častí, ktorých sa zmeny budú týkať.
    - V ideálnom prípade robiť v API zmeny, ktoré sú vždy spätne kompatibilné.
- Verzionovanie API zabezpečiť pomocou zakomponovania označenia verzie do URI v rámci elementu API najvyššej úrovne. Napríklad **v1** v prípade: **https://financnasprava.apigov.sk/v1/priznania**.
    - Pre zachovanie jednoduchosti pri používaní, je vyžadované aby API bolo verzionované ako celok, nie po jeho jednotlivých častiach.

- V prípade špeciálnej potreby, je možné verzionovanie aj pomocou mechanizmu negociácie obsahu v rámci protokolu HTTP. Napríklad namiesto jednoduchého označenia typu obsahu **application/json**, sa použije **application/vnd.financnasprava.apigov.sk.v1+json**. Vo všeobecnosti však negociáciu obsahu pre verzionovanie neodporúčame, kvôli vyššej zložitosti a nižšej prehľadnosti kontraktu API. Jedná sa tak o výnimku, ktorá je dostupná, ale musí byť pri návrhu API zdôvodnená.

- V prípade spätne nekompatibilných zmien je potrebné tieto zmeny urobiť v rámci novej verzie API, pričom nová verzia nesie typicky v označení inkrementované celé číslo (z **v1** sa stane **v2** a pod.).
  - Zároveň je potrebné pre nevyhnutnú dobu podporovať (a udržiavať funkčné) obe verzie API (starú aj novú, ktorá vznikla po spätne nekompatibilnej zmene).
  - Jasne komunikovať, dokedy bude stará verzia udržiavaná a dokedy je potrebné, aby používatelia API migrovali na novšiu verziu.

2.6. Pre autentifikáciu a autorizáciu využiť štandardy OAUTH2 a OpenIDConnect.

Autentifikácia a autorizácia (v kontexte Open API je to udelenie oprávnenia tretej strane v mene občana, alebo podnikateľa zavolať nejakú službu) je dôležitou súčasťou API, samozrejme tam, kde je to potrebné. V tejto oblasti sa vo svete REST web služieb presadili dva štandardy, OAUTH2 (autorizácia) a jeho nadstavba pre autentifikáciu – OpenIDConnect. Nedáva zmysel, aby si autorizačný server (v zmysle OAUTH2 špecifikácie) implementoval každý poskytovateľ API, preto bude túto časť poskytovať WebAPI GW ako štandardnú funkcionalitu.

Rozhodnutie.: *V prípade, že to kontext služby vyžaduje – je potrebné pre autentifikáciu a autorizáciu podporovať štandardy OAUTH2 (Bearer Token) a OpenIDConnect. Centrálnu implementáciu autorizačného servera bude poskytovať centrálny komponent WebAPI GW. Ako token bude použitý JWT token s asymetickým podpisovaním.*

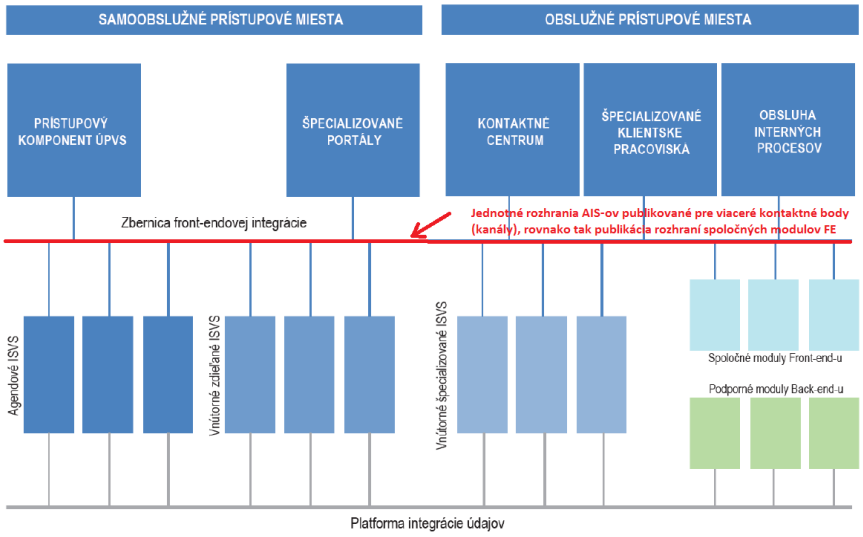
Zároveň je nutné v súvislosti so štandardmi OAUTH2 a OpenIDConnect dodržať nasledovné.:

- Vydávanie tokenov bude realizovať centrálna služba (API GW), avšak je žiadúce, aby vydané tokeny boli overiteľné decentralizovane.
- Prirodzeným kandidátom na túto formu je JWT token s asymetrickým podpisovaním. Takto bude možné centrálou podpísané tokeny považovať za dôveryhodné bez nutnosti ďalšej komunikácie s ňou.

3. Vstupná brána k službám eGovernmentu (API GW)

Samostatnou témou je centralizácia publikovania *fasády* štandardizovane definovaných služieb (ich REST rozhraní, viď predošlú kapitolu) do jedného logického bodu a možné výhody z toho vyplývajúce. Centrálny bod, do ktorého sú fasády rozhraní publikované, sa z pohľadu architektúry stáva dôležitou **vstupnou bránou** (preto ďalej technický pojem „API Gateway“, resp. skrátene „API GW“) pre využívanie služieb eGovernmentu. Možné prínosy takéhoto prístupu je možné zhrnúť do nasledovných bodov.:

a) **Jednotné sprístupnenie služieb** (AISov aj spoločných modulov front-endu) **pre viaceré kanály** (Kontaktné centrum, ÚPVS, atď..). To znamená nechceme, aby pre každý nový kanál (v budúcnosti budú vznikať ďalšie, napríklad mobilná aplikácia) budovali AIS-y nové integračné služby, čo z pohľadu investičných aj prevádzkových nákladov je neefektívna integrácia každého AIS-u s každým prístupovým bodom. Jednotné prepojenie AIS-ov s multikanálovými „prístupovými miestami“ (inými slovami kanálmi), sekundárne aj prepojenie oboch so spoločnými modulmi front-endu, predpokladá aj platná Referenčná architektúra IISVS (viď obrázok dole, červenou sú zvýraznené a doplnené informácie týkajúce sa tejto diskutovanej oblasti):



Obrázok 4: Schéma referenčnej architektúry IIS VS.

b) **Zjednodušenie procesu pripájania systémov prístupových bodov** (aj systémov tretích strán) **a podpora celého životného cyklu poskytovania rozhraní** na agendové systémy a spoločné moduly FE. V súčasnosti je pripájanie napríklad systémov tretích strán na spoločné moduly FE zdĺhavé, prácne a finančne náročné. Dobre navrhnutá API brána znamená, že nebude potrebné

robiť zdĺhavé dohody o integračných zámeroch, nebude nutné budovať špecializované integračné infraštruktúry (VPN tunely). Namiesto toho bude možné využívať jednoduchý a flexibilný nástroj na manažment celého životného cyklu API (verzionovanie API, testovací „sandbox“, podpora pre proces sprístupňovania API vlastníkom pre konzumentov, podpora autorizácie prístupu na služby v mene občana a podnikateľa, atď.)

c) **eGovernment je riadeným spôsobom otváraný pre inovácie „z vonku“.** V súčasnosti sa eGovernment buduje na všetkých úrovniach. Od tých základných (core biznis logika AIS-ov, základné prepojenie systémov ako platforma integrácie údajov a zbernica front-end integrácie) až po IT podporu prístupových miest a s nimi spojených komunikačných kanálov. Z praxe a trendu v zahraničí je však vidieť, že ak štát robí dodávky uzavreté (všetko a všetkým), tak je to na jednej strane veľmi drahé a na strane druhej nikdy nevie vyhovieť mnohým špecifickým skupinám. Preto by sa mal štát (a samozrejme je to aj trend v súkromnom sektore) sústrediť na to, aby publikoval a poskytoval základnú biznis logiku (služby) a dobre integrované prostredie pod nimi (v ktorom funguje 1x a dosť). Ostatné môže prenechať na komunitu, resp. komerčné firmy, aby vytvorili také aplikácie a systémy, ktoré budú presne vyhovovať konkrétnym skupinám používateľov.

d) **Podpora decentralizovaného budovania komplexných služieb eGovernmentu**. Dobre navrhnutá vstupná brána do eGovernemtu, predstavuje príležitosť vybudovať na všetkých úrovniach podmienky pre decentralizované poskytovanie nadstavbovej pridanej hodnoty orchestráciou viacerých služieb. Pri budovaní 1x a dosť na to štát myslí a komunikuje jednotlivým OVM, že pomocou Platformy integrácie údajov budú môcť reagovať na zmeny v (s ich agendou) súvisiacich registroch a proaktívne spúšťať pre občana služby, pričom výhodou je, že inovácia vzniká decentralizovane, u vlastníkov jednotlivých biznis služieb. Publikovanie služieb do centrálnej brány bude slúžiť ako spúšťač snáh o inovatívnu orchestráciu služieb u vlastníkov prístupových miest (a informačných systémov tretích strán, podporujúcich určité skupiny používateľov v jednotlivých kanáloch).

e) **Vstupná brána k službám predstavuje sama o sebe autoritatívny zdroj početnosti volaní a monitoringu SLA eGovernment služieb.** Akýkoľvek zmysluplný rozvoj informatizácie štátu musí byť podporovaný rastom úžitkovej hodnoty. Pokiaľ nie je jasné, ktoré služby sú občanmi alebo používateľmi využívané, pokiaľ nevieme koľko z nich sa elektronizáciou začalo využívať cez

elektronické kanály a pod. – tak štát robí investície bez spätnej väzby (a rastie riziko, že neefektívne). Samostatným problémom je, že súčasné scenáre v eGov sú (najmä vďaka centralizačným filozofiám) výrazne prepojené. Dostupnosť jednotlivých služieb tak veľakrát závisí od dostupnosti (a výkonnosti) služieb, ktoré daný OVM nevlastní. Vznikajú tak situácie, kedy sa nedostupnosť centrálnej služby „stiahne“ so sebou aj služby, ktoré sa na ňu spoliehajú. Toto je potrebné z úrovne centrálnej IT koordinácie strážiť, preto je proaktívny monitoring SLA parametrov služieb nutným predpokladom efektívnej prevádzky eGovernmentu.

f) **Zabezpečenie najvyššej možnej ochrany a dostupnosti elektronických služieb.** Centrálna brána vytvára príležitosť vybudovať a udržiavať najvyššiu možnú úroveň ochrany pred útokmi na publikované elektronické služby „z vonku“. Či už sa jedná o ochranu pred nechceným preťažením služby (pomocou techniky limitovania počtu volaní), čoraz modernejšími DDoS útokmi, alebo pred neautorizovanými operácie v mene občana (pomocou SQL injection) – je užitočnejšie (vyšší stupeň ochrany) a nákladovo efektívnejšie robiť to centrálne, ako keď sa touto témou musí zaoberať každý OVM samostatne.

4. Postup implementácie v prechodnom období, z pohľadu OVM

Z pohľadu OVM je potrebné oddeliť implementáciu služieb podľa definovaných štandardov a odporúčaní, a ďalej samotné publikovanie služieb (resp. fasády) cez vstupnú bránu do multikanálového prostredia. Je to dôležité z dôvodu, že tieto dve aktivity sú časovo nezávislé. OVM nemusí čakať na fungujúcu vstupnú bránu, pretože pripraviť si služby s rozhraniami podľa štandardov a odporúčaní môže okamžite. Zároveň nepredpokladáme, že dnes hotové a fungujúce služby sa začnú masívne prerábať len preto, aby splnili nejaké formálne podmienky. Naopak, ak sa služba bude prerábať už z dôvodu dodania nejakej novej úžitkovej hodnoty (alebo ako súčasť širšej zmeny agendového systému, či systému spoločného modulu FE) – napríklad v rámci dopytovej výzvy „drobného zlepšenia služieb eGov“ – bude vyžadované, aby bola pripravená na publikovanie v rámci tu definovaných štandardov a odporúčaní. Identifikáciou kandidátov na takéto quick-win služby sa zaoberá PS Lepšie služby.

Samotné publikovanie štandardizovanej služby do multikanálového prostredia, je následne už skôr administratívna, nie technologická záležitosť. Očakáva sa tak, že v krátkom čase od okamihu prvého spustenia API GW, bude pomocou nej publikované takmer celé spektrum štandardizovaných služieb.