ÚRAD PODPREDSEDU VLÁDY SR
PRE INVESTÍCIE
A INFORMATIZÁCIU

Referenčná architektúra Informačného systému verejnej správy v cloude

Bratislava, október 2017

Informácia o dokumente:

Názov:	Referenčná architektúra Informačného Systému Verejnej Správy v Cloude
Stav:	Pracovná verzia
Pripravil:	Pracovná skupina K9.3 Architektúra
Verzia:	1.0
Dátum:	18.7.2017
Dátum poslednej revízie:	18.10.2017

Členovia pracovnej skupiny K9.3 Strategická architektúra:

Meno a priezvisko	Organizácia
Tomáš Kysela, Václav Bárta	ÚPPVII
Ivan Šajban	MV SR
Ľubomír Ivanov	NASES
Štefan Lompart	MF SR
Peter Harvaník, Ivan Krištek	DEUS
Rastislav Machel	MK SR
Karol Janeček	MS SR
Pavol Mihalkovič	ŠÚ SR
Marián Šimegh	NCZI
Marek Orlický	MPaRV SR
Jozef Chren	ÚJD SR
Ján Suchal	Slovensko.Digital
Pavel Hrabě	PPP
Vojtech Bálint	ITAS

História verzií

Verzia	Dátum verzie	Pripravil/ Zmenil	Pripomienkoval	Kľúčové zmeny
0.1	18.07.2017	Tomáš Kysela		Príprava dokumentu, osnova tém, príprava dokumentu na pripomienkovanie jednotlivých kapitol pracovnou skupinou
0.2	28.09.2017	Tomáš Kysela		Návrh textácie jednotlivých kapitol, príprava a prezentácia dokumentu na pracovnej skupine
0.3	02.10.2017	Tomáš Kysela		Dopracovanie poslednej kapitoly podľa vstupov z dokumentu Akčný plán, príprava

Verzia	Dátum verzie	Pripravil/ Zmenil	Pripomienkoval	Kľúčové zmeny
				finalizovaného dokumentu (z pohľadu textácií) na pripomienkovanie členmi PS
1.0	12.10.2017	Tomáš Kysela	Všetci členovia skupiny	Zapracovanie pripomienok a finalizácia dokumentu.
1.1	17.10.2017	Tomáš Kysela	ÚPPVII	Zapracovanie pripomienok z IPK a PV

Obsah

1 Úvod

Účelom tohto dokumentu je, v zmysle úlohy B.5. uznesenia vlády Slovenskej republiky č. 437/2016 z 28. septembra 2016 k Národnej koncepcii informatizácie verejnej správy Slovenskej republiky (NKIVS), podrobne rozpracovať jeden z výstupov definovaných v kapitole 9 NKIVS - Referenčná architektúra. Špecificky sa však zameriava na popis referenčnej architektúry z pohľadu jednej *časti* - informačného systému verejnej správy (ISVS) postavenej natívne v prostredí cloudu. Dokument je logickým nasledovníkom schváleného dokumentu popisujúceho referenčnú architektúru *celku* – Referenčná architektúra Integrovaného Informačného Systému Verejnej Správy (IISVS).

Dokument ďalej nadväzuje na schválené dokumenty strategických priorít[1] (SP Multikanálový prístup, SP Integrácia a orchestrácia, SP Manažment údajov, SP Vládny cloud). Jeho cieľom je zachytiť architektonické rozhodnutia, resp. kontrakt medzi **centrálnou úrovňou,** ktorá vykonáva dohľad nad budovaním a rozvojom cloudového prostredia, či už v rovine vládneho alebo hybridného cloudu (reprezentuje ju Cloudová kancelária VS - CKVS, resp. pre oblasť prevádzky Kancelária riadenia prevádzky IISVS) a **jednotlivými organizáciami verejnej správy** v úlohe správcov ISVS, ktoré sú zodpovedné za rozvoj legislatívou definovaných agendových, resp. vnútorných informačných systémov (reprezentujú ich segmentoví architekti a architekti jednotlivých riešení). V prípade nesúladu obsahu tohto dokumentu s dokumentom Detailný akčný plán informatizácie 2017-2020 má prednosť ustanovenie z Detailného akčného plánu.

Úplnosť a jednoznačný výklad architektonických rozhodnutí je základným predpokladom pozitívneho efektu využívania centrálnych cloudových komponentov a platforiem budovaných v rámci OPIS, resp. plánovaných v rámci OPII projektov a programov, na rast nákladovej efektívnosti eGovernmentu v SR. S prechodom do cloudu sa nemení len okolie aplikácií/systémov, ale aj aplikácie a systémy samotné - ich záber (témy, s ktorými sa musia vyporiadať), vnútorná architektonická skladba, spôsob vývoja a publikovania nových verzií či prevádzkové postupy. Do popredia sa tak dostávajú systémy, ktoré sú od základu (natívne) postavené v kontexte flexibilného cloudového prostredia. Tento dokument predstavuje jednak cieľový popis referenčnej architektúry ISVS v kontexte natívnej cloudovej architektúry

[1] Dokumenty sú k dispozícii na stiahnutie na http://www.informatizacia.sk/strategicke-priority-erf/24190c.

a postupnosť krokov, pomocou ktorých je možné existujúce systémy do tejto architektúry posunúť.

Dôvody potreby zadefinovania základných architektonických rozhodnutí v kontexte natívnej cloudovej architektúry popisuje kapitola 2. Kapitola 3 rekapituluje jednotlivé cloudové režimy a ich význam pre ISVS vo vývoji (alebo rozvoji), mapuje jednotlivé režimy na informácie uvádzané v NKIVS a dokumentoch strategických priorít, pričom práve vo vrstve PaaS identifikuje špecializovanú časť, pomocou ktorej je možné dosiahnuť zjednodušenie a zrýchlenie tvorby ISVS pomocou cloudu a optimalizovať správu prostredí a využívanie IT zdrojov samotnými aplikáciami. Dôslednejšiemu popisu tejto „vyššej vrstvy abstrakcie tvorby aplikácií“ sa venujú ďalej kapitoly 4 (definované princípy natívnej cloudovej architektúry) a 5 (popis a príklady jej použitia pri zjednodušení, optimalizovaní vývoja a prevádzky ISVS a dosahovaní cieľov či napĺňaní potrieb uvedených v kapitole 2). Kapitola 6 popisuje pripojenie na Integrovaný informačný systém verejnej správy z pohľadu jedného (vyvíjaného alebo rozvíjaného) ISVS a na záver kapitola 7 uvedené informácie rekapituluje vo forme odporúčaní ako by mal správca konkrétneho ISVS postupovať pri vývoji nového, resp. rozvoji existujúceho ISVS v súlade s odporúčaniami cieľovej referenčnej architektúry v cloude.

2 Natívna cloudová architektúra a jej dopad na vývoj/prevádzku IS

Presun ISVS do cloudového prostredia je témou, ktorej primárny význam je komunikovaný v rovine ekonomickej racionalizácie.:

- Investičných nákladov. Ide napríklad o využitie zdieľanej infraštruktúry, platformových služieb (úložisko) až po spoločné softvérové moduly (SaaS) poskytovaných formou centralizovanej, alebo decentralizovanej služby.
- Prevádzkových nákladov – optimalizácia starostlivosti a flexibilnej škálovateľnosti infraštruktúry, platformových a softvérových služieb outsourcovaná do dátového centra.

Racionalizačný efekt je pritom vnímaný ako niečo, čo v plnej miere vzniká automaticky samotným presunom ISVS do cloudového prostredia. V skutočnosti je však miera racionalizácie nákladov dvojstupňová, v závislosti od toho, či je samotný ISVS vytvorený (alebo postupne pretvorený) do tzv. *natívnej cloudovej architektúry*. Pod pojmom *natívna cloudová architektúra* chápeme sadu pravidiel a princípov, ktoré vedú k vyššej schopnosti absorbovať (pomocou jasného oddelenia unikátnej biznis logiky aplikácie od IT prostredia/zdrojov/infraštruktúry/zdieľaných modulov či aplikácií prostredníctvom rozhraní) a lepšie využívať (napr. biznis logika je písaná tak, aby vedela v každom kroku spracovania využívať možnosť paralelizácie a bolo ju tak možné flexibilne škálovať) dynamicky sa meniace prostredie, v ktorom je daný ISVS prevádzkovaný. Z pohľadu samotnej prevádzky je natívna cloudová architektúra postavená na filozofii štandardizácie (štandardizované prostredia, zjednotené prevádzkové postupy), vďaka ktorému zrýchľuje celý proces, znižuje riziko (výpadkov) nasadzovania nových verzií (tzv. kontinuálny vývoj), a v neposlednom rade optimalizuje (ľudské) zdroje potrebné na prevádzku ISVS (automatizáciou manažmentu kompletných prostredí).

Okrem primárneho (racionalizačného) významu, je však dôležité mať na pamäti aj prínos sekundárny – prispôsobenie IT prostredia (cloudu) na aplikáciu princípov natívnej cloudovej architektúry sa výrazne zvyšujú šance zapojenia menších firiem pri tvorbe unikátnych softvérových riešení pre verejnú správu. Pričom v segmente vývoja celých (alebo častí) unikátnych softvérových riešení sa každoročne preinvestuje najväčšia časť IT výdavkov

verejnej správy. V rámci IT prostredia (cloudu), ktoré má filozofiu natívnej cloudovej architektúry priamo vo svojej „DNA“, nie je nutné riešiť tak široké spektrum problémov súvisiacich s budovaním a správou prevádzkových prostredí (pretože sú postavené a nakonfigurované automaticky) nie je nutné mať také hlboké znalosti napr. o bezpečnosti či škálovaní (pritom sú vyvíjané systémy špičkovo zabezpečené aj škálované). Riešiť plnú šírku problémov si častokrát môžu dovoliť spoločnosti až od určitej veľkosti, čím sa síce prirodzene – ale vytvára bariéra pre tie menšie. Odstránenie tejto bariéry bude mať pozitívny ekonomický efekt na IT sektor, prinesie lepšie konkurenčné prostredie a na konci zefektívni čerpanie prostriedkov na informatizáciu vo verejnej správe.

3 Cloudové režimy a ich dopad na tvorbu ISVS

Pri vývoji ISVS je potrebné, aby jeho správca rozumel základným pojmom, ktoré sa v súvislosti s cloudom používajú, možnostiam, ktoré jednotlivé cloudové režimy voči jeho aplikácií prinášajú a mapovaniu, kde je možné k nim (režimom) dohľadať detailnejšie informácie.

1. Základom všetkého je zdieľaná infraštruktúra (hardvér). Prakticky dnes neexistuje dôvod, aby (bežný) moderný ISVS potreboval k svojmu behu špeciálny hardvér (so špeciálnym hardvérom výrazne rastie cena na vývoj aj prevádzku). Veľkej väčšine ISVS tak na svoj beh postačuje **zdieľaný** komoditný hardvér, ktorý je možné nakupovať a prevádzkovať centrálne (do dátových centier štátu) a využívať tak „úspory z rozsahu“ pri hospodárení s verejnými financiami. Zdieľanie infraštruktúrnych prvkov ako CPU, pamäť RAM, resp. sieťovej prenosovej kapacity či prvkov smerovania (teda akýsi vyšší stupeň virtualizácie hardvéru) nesie označuje **IaaS** (z anglického *Infrastructure as a Service*) a podrobnejšie je rozobraná v rámci dokumentu *Strategická priorita Vládny cloud.* Okrem komoditného HW môže byť nevyhnutné v odôvodnených prípadoch využiť špecializovaný HW (HSM, WAF a pod.).

2. Bežná aplikácia (a teda aj typický ISVS) pozostáva z niekoľkých vrstiev: vrstvy prístupovej (sprístupňovanie rozhraní aplikácie do otvorenej siete), vrstvy obrazoviek (zobrazovania), vrstvy biznis logiky (aplikačných služieb), vrstvy orchestračno-integračnej (publikované procesy biznis logiky, resp. integračné služby) a vrstvy perzistenčnej – kde je uchovávaný stav (databáza). V jednotlivých vrstvách všetky aplikácie využívajú relatívne štandardné technologické komponenty (ako napríklad rovnomerné rozhadzovanie záťaže na prístupovej vrstve, technologický komponent zbernice a procesnej orchestrácie na vrstve integračnej, či SQL/NoSQL databáza na vrstve perzistenčnej). Namiesto toho, aby si každá aplikácia nosila takéto riešenie v rámci svojej inštalácie, môže tieto štandardizované „funkčnosti“ využívať ako štandardné služby, ktoré sú všetkým aplikáciám poskytované v rámci cloudovej „platformy“ – čo nesie označenie **PaaS** (z anglického *Platform as a Service*) a v takomto význame sa mu venuje opäť dokument *Strategická priorita Vládny cloud*.

Pri vývoji unikátnych softvérových riešení pre verejnú správu je však možné identifikovať aj špecializovanú PaaS vrstvu, ktorá vývoj takýchto aplikácií (a ISVS) uľahčuje, zjednodušuje a zefektívňuje. Takto vyvíjané aplikácie sú postavené v duchu tzv. „natívnej cloudovej architektúry“, ktorej základné princípy sú uvedené v nasledujúcej kapitole tohto dokumentu a podrobnejšie rozpracované (konkrétne príklady pre vývoj a prevádzku) v kapitole 5. Špeciálna PaaS vrstva pre vývoj natívnych cloudových aplikácií popísaná v jadre tohto dokumentu je tak komplementárnym popisom možností, ktoré PaaS vrstva pre ISVS prináša.

3. Nie všetky oblasti biznis domény, ktoré nejaká aplikácia pokrýva, resp. nie všetky kroky biznis procesu, ktorý daná aplikácia podporuje – sú unikátne. Naopak v príbuzných doménach verejnej správy je možné nájsť subdomény (alebo podprocesy), ktoré sa (len v možno iných nastaveniach) opakujú v rôznych oblastiach, či riešeniach. Je preto logické, že takéto opakujúce sa oblasti sú vyvíjané ako špecializované aplikácie (či už ako jedno – centrálne, alebo ako sada certifikovaných) riešení – čím vytvárajú vrstvu s označením **SaaS** (z anglického *Software as a Service*). Softvérové riešenia ako služba môžu byť poskytované buď „inštalovaním“ (a konfigurovaním) vlastnej inštancie per prípad jej použitia (do prostredia aplikácie či systému, ktorý sa chystá takýto zdieľaný komponent využiť) – alebo iba nakonfigurovaním centrálne prevádzkovanej aplikácie/systému (na ktorú sa následne pripojí rozvíjaná aplikácia pomocou integračných rozhraní ako na akúkoľvek zdieľanú službu). Úvodné informácie k SaaS vrstve obsahuje dokument *Strategická priorita Vládny cloud* a detailnejší pohľad poskytujú výstupy pracovnej skupiny Digitalizácia agend verejnej správy.

4. Niekedy sú časti domén, alebo podprocesy, v ktorých je možné etablovať spoločné (zdieľané) riešenie – vhodnými kandidátmi na užšiu špecializáciu v rámci inštitúcií verejnej správy a ich kompetencií. Následkom čoho je, že ich nejaká (špecializovaná) inštitúcia celé zastreší (ich vykonávanie aj IT podporu jednotným nástrojom) pričom ostatné OVM k nej tieto podprocesy „outsourcujú“. Táto vrstva nesie označenie BPaaS (z anglického *Business Process as a Service*) a jej detailnejšiemu popisu sa venujú výstupy pracovnej skupiny „Digitalizácia agend verejnej správy“.

Z uvedených režimov vyplýva, že ideálna aplikácia by mala zastrešovať len unikátne časti domény a procesov (pre ostatné využívať zdieľané komponety či už vo vrstve BPaaS alebo

SaaS) a tieto unikátne časti poskytovať pomocou PaaS vrstvy, špeciálne formou „vyššej úrovne abstrakcie“ pre vývoj a prevádzku aplikácií, ktorá je bližšie predstavená v nasledujúcich dvoch kapitolách.

Služby Vládneho cloudu sú popísané v katalógu služieb. V prípade špecifických potrieb ISVS a splnenia predpokladu efektívneho zdieľania, je možné doplniť do katalógu nové služby. Postup pre doplnenie novej služby bude transparentne definovaný pre všetky úrovne služieb XaaS. Na úrovni IaaS to môže rošírenie o ďalší operačný systém alebo špecializovaný virtuálny "appliance", pre PaaS to môže byť doplnenie rozšírenie o ďalší špecializovaný stavebný prvok (napr. „distribuovaný proces vykonávajúci úlohy podľa rozvrhu“), programovací jazyk, databázovú platformu, apod.

V súčasnosti vládny cloud poskytuje služby v 2 lokalitách, takže má základné predpoklady pre zabezpečenie "Disaster Recovery", ako to legislatíva pre ISVS vyžaduje. Vládny cloud umožní dynamicky a maximálnej možnej miere automatizovane poskytovanie služieb ISVS z náhradnej lokality, v prípade, že dôjde k nepredvídaným udalostiam v "primárnej lokalite", v ktorej bude ISVS prevádzkovaný. V prípade využívania služieb PaaS, resp. SaaS to bude pre konzumenta služieb transparentné, v prípade využívania služieb IaaS vládny cloud poskytne prostriedky pre automatizáciu prevádzkovateľovi ISVS. V bežnej situácii budú napr. údaje v DB synchronizované do sekundárnej lokality a využitie výpočtového výkonu bude len na úrovni potrebnej na tento účel. V prípade potreby bude automaticky výpočtový výkon využívaných XaaS služieb navýšený tak, aby služby ISVS mohli byť posktytované zo sekundárnej lokality. Obdobný princíp bude použitý pre potreby "capacity managementu". ISVS v bežných podmienkach často potrebuje využívať len zlomok výpočtového výkonu, pričom potreby rastú len v niektorých obdobiach skokovo. Tento problém rieši vládny cloud možnosťou dynamického prideľovania dodatočnej výpočtovej kapacity ISVS na určitú dobu.

4 Princípy referenčnej architektúry ISVS v cloude

Budovanie ISVS v duchu natívnej cloudovej architektúry, si vyžaduje aplikáciu, resp. dodržiavanie nasledujúcich princípov, ktoré pre daný systém znamenajú, okrem iného, aj:

- používanie deklaratívnych formátov pre automatizovaný manažment prostredí, čoho výsledkom je kratší čas a nižšie náklady pre vývojára pridať sa k projektu vývoja daného systému,
- transparentný a jasný kontrakt s operačným systémom v rámci ktorého systém beží, čo umožňuje maximálnu prenositeľnosť medzi exekučnými prostrediami,
- ľahkú prenositeľnosť nasadenia medzi zavedenými komerčnými cloudovými platformami (čím sa priamo podporuje hybridný cloudový model),
- minimalizáciu rozdielov medzi prostrediami (napr. vývojové, integračné, testovacie, produkčné), čo je základný predpoklad nasadzovania a rozvoja systému bez rizika výpadkov (a prerušenia poskytovania služby) a agilného vývoja,
- flexibilnú škálovateľnosť bez dramatických zmien do nástrojov, architektúry, alebo spôsobu vývoja.

Doleuvedené princípy sú nezávislé od programovacieho jazyka/platformy, nástroja či technologickej komponenty.

1. **Jeden repozitár zdrojového kódu pre jednu „aplikáciu“** (napriek tomu že aplikácia môže mať viacero nasadení). Ak je systém (ISVS) zložený z viacerých častí, tak sa jedná o distribuovaný systém, pričom každá jeho časť je pre účely týchto pravidiel „aplikácia“ (a má svoj vlastný repozitár zdrojového kódu). Zdrojový kód aplikácie je tak len jeden, aj keď nasadení (bežiacich inštancií v rôznych prostrediach) danej aplikácie môže byť viac a tieto inštancie môžu byť založené na rôznych verziách zdrojového kódu, ale stále pochádzajúceho z jedného jediného repozitára. Súčasťou repozitára môže byť aj dokumentácia a príručky (aktualizovaná spolu s aplikáciou v zmysle „dokumentácia ako zdrojový kód“).

2. **Explicitná deklarácia a izolácia závislostí aplikácie**. Aplikácia sa nikdy nespolieha na implicitne dostupné (napríklad v danom technologickom prostredí) knižnice alebo

moduly. Naopak, všetky svoje závislosti transparentne deklaruje a zároveň zamedzuje prienikom „implicitných“ závislostí – na toto všetko využíva mechanizmy/nástroje správy závislostí dostupných v rámci zvolenej technologickej platformy.

3. **Konfigurácia (aplikácie) súčasťou prostredia, nie aplikácie.** Za porušenie tohto princípu sa považuje stav, ak konfigurácia aplikácie je súčasťou jej zdrojového kódu (priamo alebo ako konfiguračný súbor) a na to, aby sa zmenila konfiguračná hodnota - je potrebné artefakty nasadenia znovu vytvoriť/skompilovať. Naopak uloženie konfigurácie aplikácie v systémových premenných prostredia (tzv. „*environmental variables*“) odstraňuje jednak závislosť konfigurácie od použitej technologickej platformy (programovacieho jazyka) a zvyšuje prenositeľnosť aplikácie medzi prostrediami.

4. **Nezávislosť aplikácie od konkrétneho poskytovateľa podpornej služby back-endu.** Aplikácia musí byť písaná takým spôsobom, aby boli podporné služby back-endu (napríklad databázy, systémy výmeny správ a udalostí a pod.) dodávané zväčša v rámci vývoja alebo prevádzky - kedykoľvek nahraditeľné za také, ktoré prevádzkujú tretie strany. Podporné služby sú tak využívané aplikáciou ako „zdroje“, pričom jednotlivé nasadenia a prostredia linkujú konkrétne inštancie (prevádzkované napríklad tretími stranmi) pomocou URL identifikátorov ako súčasť konfigurácie (viď bod 3).

5. **Jasné oddeľovanie jednotlivých štádií transformácie zdrojového kódu na bežiacu aplikáciu.** Zdrojový kód počas transformácie prechádza troma štádiami. *„Skonštruovanie“ (tzv. „build“) artefaktov nasadenia* je štádium, kedy sa zdrojový kód transformuje na spustiteľné artefakty (napríklad súbory). V štádiu *„uvoľnenia verzie“(tzv. „release“)* sú artefakty nasadenia skombinované s konfiguráciou aplikácie pre jednotlivé exekučné prostredia (výsledkom čoho je aplikácia, ktorú je možné v daných prostrediach nasadiť a spustiť). Uvoľnená verzia (release) má unikátny identifikátor a jej obsah nie je možné meniť (akákoľvek zmena znamená novú verziu). Tretím štádiom je *spustenie aplikácie (tzv. „run“)* v konkrétnom exekučnom prostredí. Vo všeobecnosti platí, že v štádiu spustenia (a spúšťania) nie je možné robiť žiadne zmeny do aplikácie (zdrojového kódu ani konfigurácie), čo sleduje napr. cieľ rýchleho nábehu aplikácie v prípade výpadku.

6. **Spustená aplikácia beží ako jeden alebo viac bezstavových procesov.** Komunikácia medzi procesmi nie je nikdy závislá od zdieľania spoločného pamäťového bloku, súboru - v rámci jedného procesu, alebo transakcie - uvedené prostriedky však môžu slúžiť ako dočasná cache. Procesy svoj stav synchronizujú (aj medzi sebou) výhradne pomocou podpornej služby back-endu (napríklad perzistenčnej – databázy).

7. **Aplikácia je sama zodpovedná za publikáciu svojich komunikačných koncových bodov (portov).** Aplikácia sa tak nespolieha na to, že na svoj beh musí byť vložená do špeciálneho kontajnéra (HTTP Server a pod.), ale má celé svoje „spustenie“ pod kontrolou, aktívneho ho vie riadiť a ovplyvniť. V rámci toho je zodpovedná za publikovanie svojich komunikačných koncových bodov (portov).

8. **Jednoduché škálovanie výkonu pomocou spúšťania a zastavovania (paralelných) bezstavových procesov.** Písať aplikáciu v súlade s bodom 6 znamená okrem iného aj to, že spúšťanie nových procesov pre spracovanie (rôznych typov úloh) je jednoduchou a spoľahlivou operáciou. Aplikácia pri svojom behu musí akceptovať, že manažment procesov priamo neriadi (pretože by mohol nastávať problém s prenositeľnosťou medzi prostrediami), ale reaguje na riadiace udalosti (napr. vypnutie, reštart) manažmentu procesov daného operačného systému prostredia, v ktorom aplikácia beží. Toto všetko samozrejme nevylučuje možnosť paralelného spracovania *vo vnútri* procesov (pomocou vlákien), ktoré si samozrejme riadi daná aplikácia.

9. **Okamžité reakcie procesov na požiadavky spustenia a zastavenia** smerujú k flexibilnejším (z pohľadu škálovania výkonu) a robustnejším aplikáciám. Procesy musia byť písané tak, aby minimalizovali čas potrebný od okamihu svojho spustenia do ich plného behu (napríklad obsluhy prichádzajúcich požiadaviek). Rovnako tak musia byť procesy pripravené v prípade prijatia signálu na zastavenie (napr. pri vypnutí) – aplikovať postup riadeného zastavenia procesu (napríklad okamžite prestať prijímať nové požiadavky, dokončiť tie, čo sa práve vybavujú a ukončiť beh procesu).

10. **Minimalizovať rozdiely medzi prostrediami (najmä vývojovým a produkčným).** Aplikácia v natívnej cloudovej architektúre by mala byť vyvíjaná a rozvíjaná pomocou techniky kontinuálneho rozvoja – zmeny sa do produkcie dostávajú rýchlo (hodiny, max. dni namiesto týždňov), vývojári sa priamo zúčastňujú na aktivitách súvisiacich

s nasadzovaním (majú tak kontakt so správaním aplikácie na všetkých prostrediach, zároveň prevádzkový team nie je nutné budovať v takom rozsahu, ako keď bol budovaný separátne od vývojového) a jednotlivé prostredia sú si z pohľadu využívaných nástrojov (napr. pri nasadení), alebo podporných služieb back-endu – čo najviac podobné. Už na vývojom prostredí je tak možné odchytiť väčšinu problémov, vďaka čomu sa v maximálnej možnej miere môže predchádzať problémom vznikajúcim na poslednú chvíľu pri pokuse o nasadenie do ostrej prevádzky (produkcie).

11. **Aplikácia nikdy neriadi (a nespolieha sa na proprietárny) spôsob spracovania logov.** Aplikácia nikdy nezapisuje logy priamo do súborov ani sa nesnaží implementovať logiku ich manažmentu. Logy zapisuje do tzv. výstupného *stream*-u (*outpustream*) pričom v štádiu vývoja je tento stream presmerovaný do vývojárskej konzoly (aby vývojár okamžite videl chyby a vedel reagovať), v produkčnom prostredí je (stream) automaticky presmerovaný do centrálneho úložiska, kde sú logy indexované a pripravené na dohľadávanie pracovníkmi prevádzky pri plnení svojich úloh.

12. **Admin/manažment úlohy sú vyvíjané a vykonávané ako jednorazové procesy.** Procesy, ktoré vykonávajú admin, alebo rôzne manažment úlohy (skript na migráciu databázy a pod.) sú vyvíjané ako súčasť zdrojového kódu aplikácie, pod rovnakým manažmentom verzií (nie separátne) a spúšťané na rovnakých prostrediach (a ich konfiguráciách), ako ostatné procesy aplikácie.

13. **Pre maximalizáciu robustnosti a minimalizáciu výpadkov aplikácie,** je potrebné (tam, kde je to možné a efektívne) využívať tzv. „modro-zelený“ systém nasadzovania. Jeho podstata spočíva v paralelnom behu (v okamihu nasadzovania novej verzie do produkcie) dvoch identických produkčných prostredí, pričom používateľov (alebo prichádzajúce požiadavky) obsluhuje vždy len jedno z nich. Postup pri nasadzovaní je taký, že na jednom sa vykoná finálna príprava a odladenie releasu nad konfiguráciou produkčného prostredia a následne sa prepne presmerovanie požiadaviek z doteraz obsluhujúceho (stará verzia aplikácie) na prostredie obsahujúce odladenú novú verziu (pričom staré prostredie je stále pripravené byť zapojené v prípade, že sa vyskytnú neočakávané chyby).

14. **Vývojové/integračné/produkčné prostredia musia byť optimalizované na minimálny čas spustenia** (rovnako ako aplikácie – bod 9)**.** To znamená, že pridanie nového prvku (nový databázový uzol, alebo nový výpočtový uzol) musí byť do čo najkratšieho času (do niekoľkých sekúnd). Na to je potrebné využívať techniku kontajnerizácie a štartovania predpripravených „obrazov“ jednotlivých uzlov, je porušením tohto princípu, ak sa škálovanie robí pomocou automatizovanej inštalácie do „čistého“ virtuálneho servera.

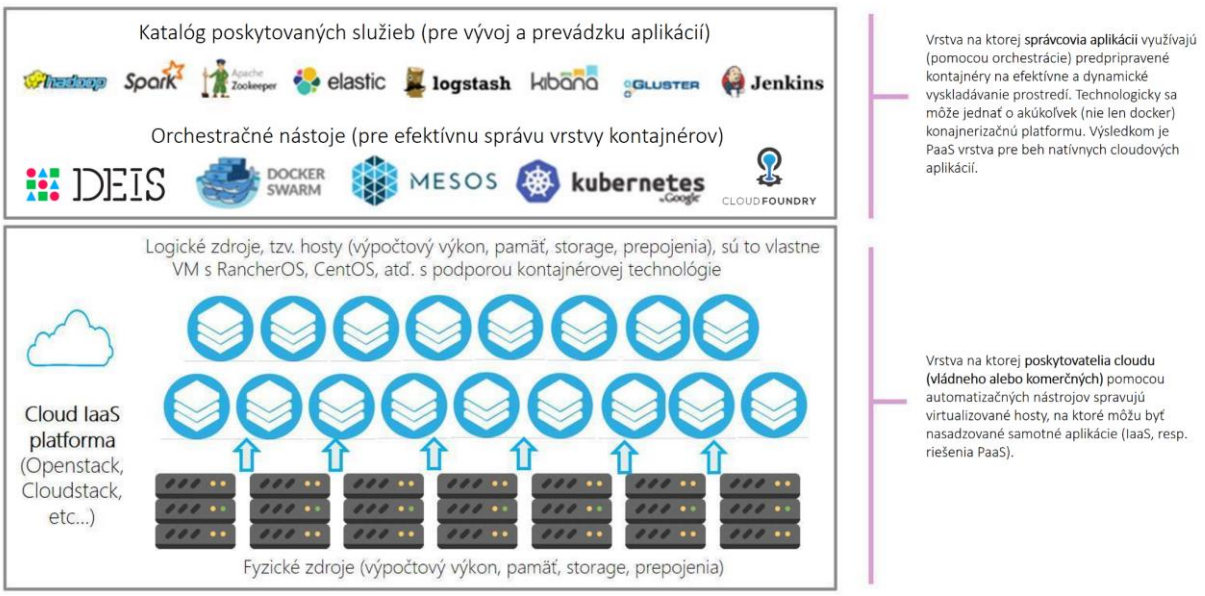
5 „DevOps“ v cloude - prevádzka natívnej cloudovej aplikácie

Prevládajúcim prístupom, ktorý sa úspešne presadil v oblasti optimalizácie a zvyšovania efektivity prevádzky, je tzv. prístup „DevOps“, ktorý vnáša prvky agilných metodík do postupov a nastavenia prevádzkových činností. V rámci prístupu „DevOps“ sa do popredia dostáva tlak na čo najužšie prepojenie procesov vývoja a prevádzky aplikácie/systému. Procesy vývoja a prevádzky tak neprebiehajú oddelene, ale sú medzi sebou poprepájané, počas celého životného cyklu aplikácie (napríklad aspekty prevádzky vstupujú už do samotného návrhu aplikácie, alebo z opačnej strany - prevádzkové postupy a prostredia sú nastavené tak, aby vývoj a zmeny aplikácie mohli byť robené priebežne aj počas ostrej prevádzky aplikácie). Mnohé princípy prezentované v kapitole 3 sú práve odrazom aplikovania filozofie „DevOps“.

Ako už bolo uvedené v kapitole 2, sledujeme aj vyšší princíp - než je len optimalizácia a efektivita existujúcich postupov, a to - zapojenie širšieho spektra, najmä menších spoločností do tvorby unikátnych softvérových riešení pre verejnú správu. Základným predpokladom je zjednodušenie tvorby a manažmentu prevádzkových prostredí (vývojové, testovacie, integračné, ostrá prevádzka). V praxi by malo vyzerať tak, že osoba zodpovedná za vývoj/prevádzku (vývojár, devops inžinier) je schopná nadefinovať dané prostredie veľmi jednoducho, bez toho aby musela detailne riešiť (a zodpovedať za) nastavenia rôznych technologických vrstiev/komponentov. Definícia tak prebieha na *vyššej úrovni abstrakcie,* napríklad pomocou definície koľko „web“ procesov/spracovávateľov (pričom proces môže mať rôzne priradené výpočtové prostriedky ako CPU alebo pamäť) spracováva prichádzajúce requesty, v akom rozsahu a dostupnosti je pod nimi nastavená služba perzistencie (databáza - PaaS), koľko beží v prostredí back-end procesov/spracovávateľov (ktoré robia spracovávanie na pozadí), resp. kedy a ako sa spúšťajú procesy jednorázové (administračné). Vývojár/devops inžinier tak nemusí riešiť detaily, že HTTP requesty sú smerované len do web procesov (a automaticky rozdeľované tak, aby ich zaťažovali rovnomerne), že tieto web procesy sú automaticky chránené voči rôznym typom útokov (napr. DoS, DDoS), nemusí riešiť a konfigurovať škálovanie a konzistenciu (aj v prípade výpadkov) perzistenčnej služby, atď. Uvedené nastavenia sa musia v prostrediach prejavovať okamžite (definícia môže prebiehať formou spúšťania príkazov v administratívnej konzole, alebo pomocou grafického administrátorského UI/obrazoviek). Vývojár sa tak sústreďuje na to podstatné (biznis logiku svojej aplikácie), pričom pomocou clodu je zabezpečené to ostatné (nevyhnutné).

Schopnosť efektívnej správy prostredí pomocou „vyššej úrovne abstrakcie“ zjednodušuje a podporuje efektívnu implementáciu mnohých „Dev/Ops“ praktík – napríklad ako zabezpečiť kontinuálny rozvoj aplikácie/systému, s minimalizovaním odstávok a rizík spojených s nasadzovaním nových verzií (napríklad pomocou tzv systému „modro-zeleného“ produkčného prostredia - princíp č. 13 v kapitole 3).

Vyššia úroveň abstrakcie a jej efektívny manažment **otvára úplne nový priestor** (pre vlastníka aplikácie/služby) **pre dynamické a efektívne** (= automatizované) **škálovanie prostredí**. Efektívne (a dynamické) škálovanie prostredí je základným predpokladom hospodárneho využívania IT zdrojov a zabezpečenia dostatočného výkonu v tej chvíli, pre také aplikácie a v takých ich častiach – kde to práve používatelia (občania, podnikatelia, pracovníci verejnej správy), alebo prevádzkové procesy systémov – potrebujú. Pomocou vyššej úrovne abstrakcie tak môže vývojár zakomponovať napr. automatizované stratégie škálovania (kedy a na základe čoho sa spustia nové procesy spracovania web dotazov, kedy a na základe čoho sa zvýši výkon perzistenčnej služby alebo procesov spracovávania na pozadí). Toto všetko samozrejme závisí od toho, či vyššiu vrstvu abstrakcie (kontajnerizáciu) bude cloud v rámci PaaS podporovať, alebo nie. Nasledujúca schéma vizualizuje túto špecializovanú PaaS nadstavbu (pre natívne cloudové aplikácie).:



Z uvedenej schémy vyplýva aj pozicionovanie špecializovanej PaaS vrstvy pre tvorbu a beh natívnych cloudových aplikácií voči klasickej cloudovej infraštruktúre – IaaS (ako jej

nadstavba). Poskytovanie uvedenej vyššej úrovne abstrakcie (založenej na technologickej vrstve tzv. kontajnérov) pre efektívnu správu a manažment prostredí v cloude je dnes už štandardnou ponukou komerčných cloudových riešení (Amazaon, Azure, Heroku, atď) a pre svoje nesporné benefity (optimalizácia prevádzky, efektívny manažment vyťažovania výpočtových zdrojov, vytvorenie priestoru pre vstup malých firiem - hlboko špecializovaných na vývoj samotných aplikácií - do prostredia vývoja unikátnych softvérových riešení pre verejnú správu) musí byť súčasťou cloudovej stratégie a nesmie chýbať vo funkčných vlastnostiach aj Vládneho cloudu.

6 Napojenie sa na Integrovaný informačný systém verejnej správy (IISVS)

Z referenčnej architektúry IISVS vyplýva, že akýkoľvek ISVS je možné s ostatnými systémami v rámci eGovernmentu prepojiť pomocou:

- zbernice FE integrácie
- platformy integrácie údajov (IS CSRÚ).

Bez ohľadu na to, či sa jedná o agendový alebo vnútorný (zdieľaný či špecializovaný) ISVS, je potrebné vyriešiť nasledovné.:

- Prihlasovanie (autentifikáciu) pripojením sa na konkrétny spoločný modul IAM (prihlasovanie občanov a prihlasovanie úradníkov môžu riešiť rozdielne IAM moduly).
- Poskytovanie informácií o prípadnom využívaní osobných údajov (kvôli GDPR) do platformy integrácie údajov, odkiaľ ich ďalej spracováva a komunikuje s občanom modul „Moje údaje“.
- Poskytovanie, alebo využívanie údajov (referenčných a/alebo otvorených, analytické spracovanie údajov) pripojením sa na platformu integrácie údajov. Sem patrí aj reagovanie na vzniknuté a propagované udalosti v iných ISVS (napríklad poskytovaním proaktívnych služieb v prípade agendových ISVS) a propagácia vlastných udalostí (na ktoré môžu a budú reagovať iné ISVS).

Pre agendové systémy je ďalej potrebné (a pre ostatné ISVS voliteľné) riešiť:

- Autorizáciu akcií a dokumentov (podpisovanie).
- Napojenie na zdieľaný modul elektronickej registratúry v prípade jej využívania.
- Publikovanie (a zdokumentovanie) všetkých dôležitých služieb (pre agendové systémy najmä služieb biznis logiky) cez Web API GW, odkiaľ môžu byť využívané akýmkoľvek „kontaktným bodom“ (v zmysle bodu obsluhy v multikanálovom prostredí). Následne je potrebné vyviesť relevantnú podmnožinu služieb do OpenAPI prostredia.

V prípade agendových systémov je navyše potrebné vyriešiť aj:

- Pripojenie na elektronické doručovanie a modul elektronických formulárov. V prípade, že agendový systém využíva vlastné používateľské rozhranie pre zadávanie údajov

počas konfigurácie poskytovania služby – je potrebné zabezpečiť prepojenie UI s kontaktnými bodmi (najmä ÚPVS).

- Vyvedenie vybraných služieb z Web API GW do modulu procesnej orchestrácie, kde sú tieto zapojené do komplexných orchestrovaných služieb.
- Pripojenie na ostatné spoločné moduly FE (štátny messenger/chat box, ...)

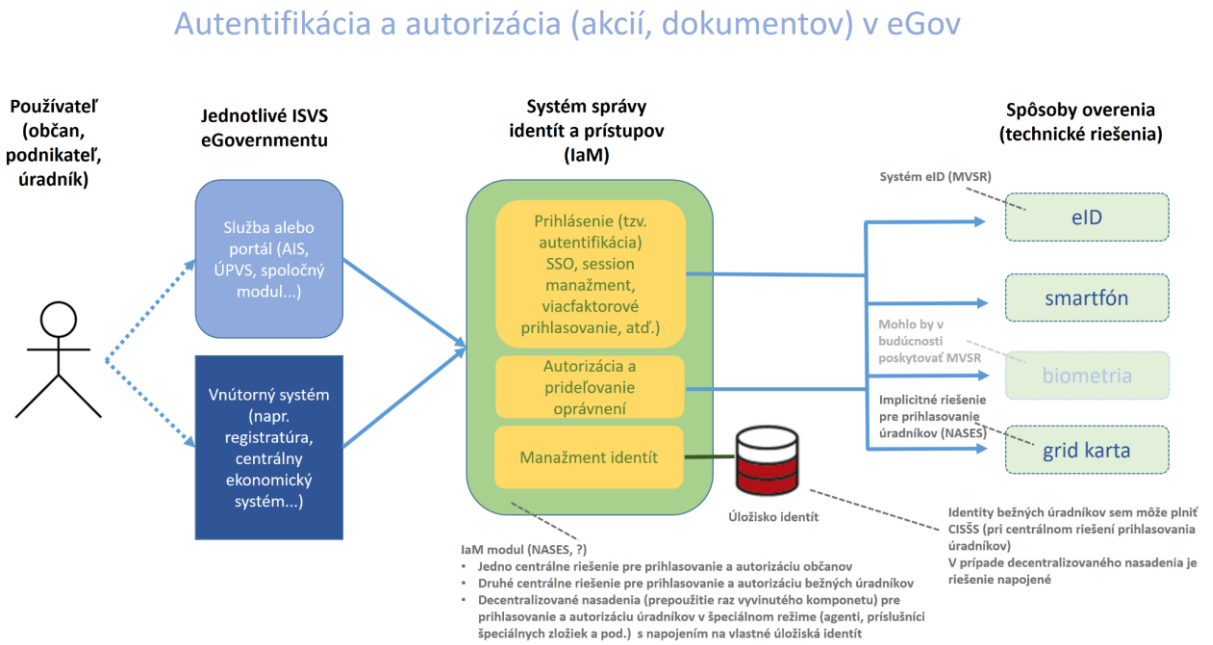
Podrobnejšie budú ďalej popísané témy, ktoré sa dotýkajú všetkých ISVS (bez ohľadu, či sa jedná o agendové, alebo vnútorné).:

6.1 Autentifikácia a autorizácia (akcií, dokumentov) v eGovernmente

Pri riešení prihlasovania (autentifikácie) a autorizácie (akcií, dokumentov) je potrebné identifikovať dva rozdielne prvky:

- Systém správy identít a prístupov (IaM modul – súčasť množiny spoločných modulov FE). V rámci neho sú typicky riešené témy nenaviazané na konkrétnu metódu overenia (napríklad pomocou eID karty) – správa a životný cyklus autentifikačnej „session“ a jej využívanie všetkými systémami (SSO), pripojenie na úložisko identít, pripojenie na rôzny spôsoby overenia a pod. Špeciálne v tejto časti je dôležité uviesť, že typicky IaM systémy v rámci „autorizácie“ zastrešujú aj tému centrálnej správy a prideľovania „aplikačných oprávnení“. V praxi sa však táto technika presadzuje skôr pozvoľne, pričom sa skúmajú rôzne modely manažmentu a prideľovania oprávnení (napr. ABAC, RBAC,...) s rôznou ambíciou generalizácie a s kolísavou mierou úspechu v rámci rôznych prípadov použitia/nasadenia. Vzhľadom na uvedené, nepovažujeme vyriešenie otázky **centrálneho manažmentu aplikačných oprávnení** za architektonickú prioritu (na rozdiel od centrálneho riešenia pre autorizáciu úkonov v rámci systémov, alebo autorizovanie dokumentov). Jednotlivé ISVS v rámci oblastí (alebo segmentov) samozrejme môžu takéto spoločné riešenie využívať a zdieľať.
- Množina spôsobov overenia (pomocou eID karty, pomocou smartfónu, pomocou biometrie (napr. tváre), a pod.). Overenie v prípade „autorizácie“ akcie, alebo dokumentu sa na pozadí realizuje pomocou dostatočne dôveryhodného spojenia udelenia oprávnenia (autorizácie) s dokumentom, alebo akciou/transakciou. Pričom rôzne služby môžu vyžadovať buď nižšiu (elektronický podpis), alebo vyššiu (kvalifikovaný elektronický podpis) úroveň zabezpečenia.

Prihlasovanie a udeľovanie oprávnení *občanov* (pre transakcie, akcie, dokumenty) je v prostredí IISVS riešené centrálne (viď nasledujúci obrázok), v rámci spoločných modulov FE. Centralizácia riešenia v tomto prípade prirodzene vyplýva z povahy problému. Nie je dôvod, aby bolo budované špeciálne riešenie pre nejakú konkrétnu podskupinu občanov – naopak je žiadúce, aby všetci občania mali k dispozícii jednotné riešenie (samozrejme sprístupňujúce viaceré metódy overenia).



Mierne zložitejšia je situácia v prípade pracovníkov verejnej správy. Postupný nárast využívania centrálnych riešení pre rôzne časti agend či procesov výkonu verejnej moci stavajú do popredia otázku (jednotného) prihlasovania a autorizácie úkonov pracovníkov VS do týchto systémov. Je zrejmé, že na to nie je možné využiť existujúci IaM modul (aspoň nie bez úprav) – a napojiť sa na fungujúce overenie pomocou eID karty, keďže vlastnenie eID nie je povinné. Vytvára sa tak priestor pre iné riešenie IaM modulu, resp. úpravu pôvodného – pričom logický model zostáva rovnaký (viď predchádzajúci obrázok), so špecifickou požiadavkou na schopnosť jeho decentralizovaného nasadenia. Existujú skupiny pracovníkov verejnej správy, pri ktorých z rôznych dôvodov (zväčša bezpečnostných) nie je možné využívať centralizované riešenie. Metódy overovania autentifikácie alebo autorizácie pre pracovníkov verejnej správy môžu byť podobné (alebo dokonca zdieľané) ako v prípade prihlasovania občanov a podnikateľov, prípadne doplnené o špecifické (elektronické „grid“ karty).

6.2 *Poskytovanie alebo využívanie údajov pripojením sa na platformu integrácie údajov, poskytovanie informácií o využívaní osobných údajov*

Každý správca ISVS v rámci svojho systému vedie evidenciu tzv. **primárnych** údajov, pričom sekundárne je povinný (podľa zákona 305/2013) využívať/pripojiť sa na referenčné údaje z iných systémov (registrov). Nad svojimi primárnymi údajmi, je ďalej každý správca ISVS povinný (podľa NKIVS) zabezpečiť poskytovanie:

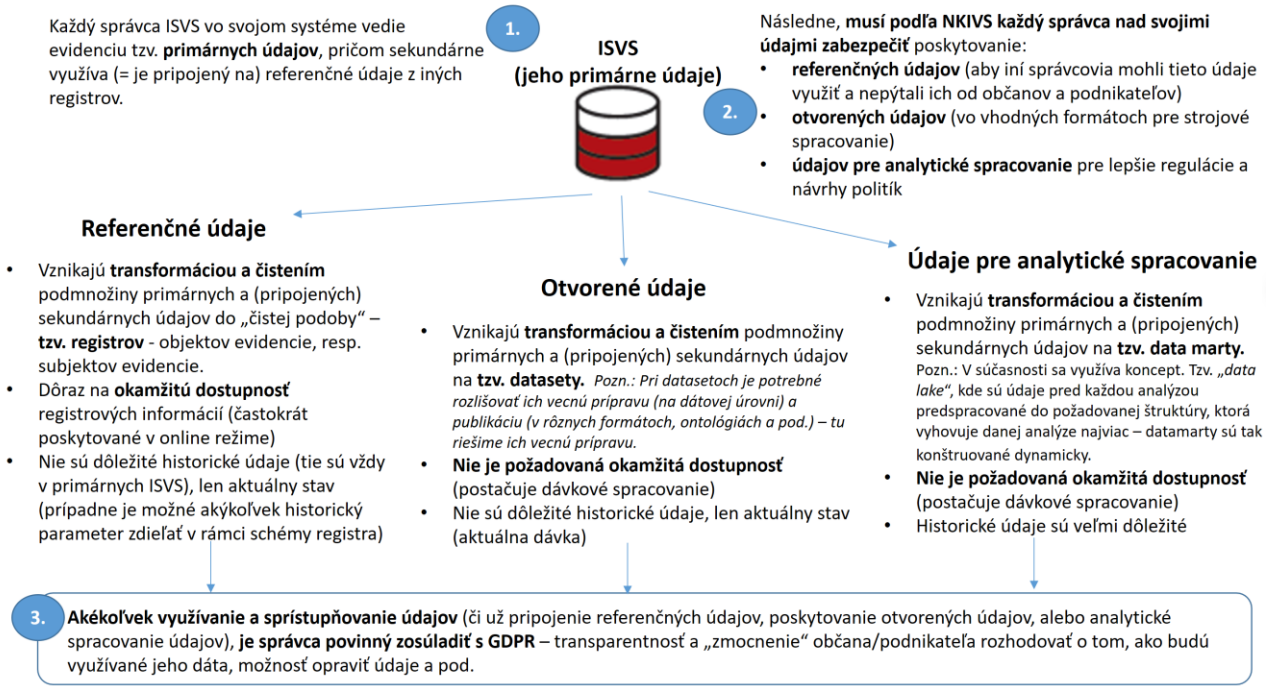
a) *referenčných údajov* (aby sa na nich iné systémy mohli pripojiť a nemuseli ich požadovať pri obsluhe občana, alebo podnikateľa; sem patrí aj publikovanie a konzumácia udalostí o zmenách údajov, čo je základný mechanizmus pre implementáciu proaktívnych služieb)
b) *otvorených údajov* (vo vhodných formátoch pre strojové spracovanie)
c) *údajov* pre analytické spracovanie pre lepšie regulácie a návrhy politík.

V neposlednom rade bude v roku 2018 každý správa ISVS povinný (podľa smernice GDPR) sprístupňovať informácie o využívaní osobných údajov, pričom pod využívaním rozumieme:

- zdieľanie údajov s inými správcami ISVS, resp. v budúcnosti s definovanými partnermi mimo VS (referenčné údaje)
- využívanie údajov ako podklad pri tvorbe datasetov (otvorené údaje) a to aj v prípade, ak výsledné datasety identifikáciu konkrétnej osoby alebo podnikateľa neobsahujú (anonymizácia)
- využívanie údajov na analytické spracovanie a to aj v prípade, že použité dat-marty identifikáciu konkrétnej osoby alebo podnikateľa neobsahujú (anonymizácia)
- evidencia a spracovanie údajov k danému občanovi alebo podnikateľovi v rámci vlastnej agendy

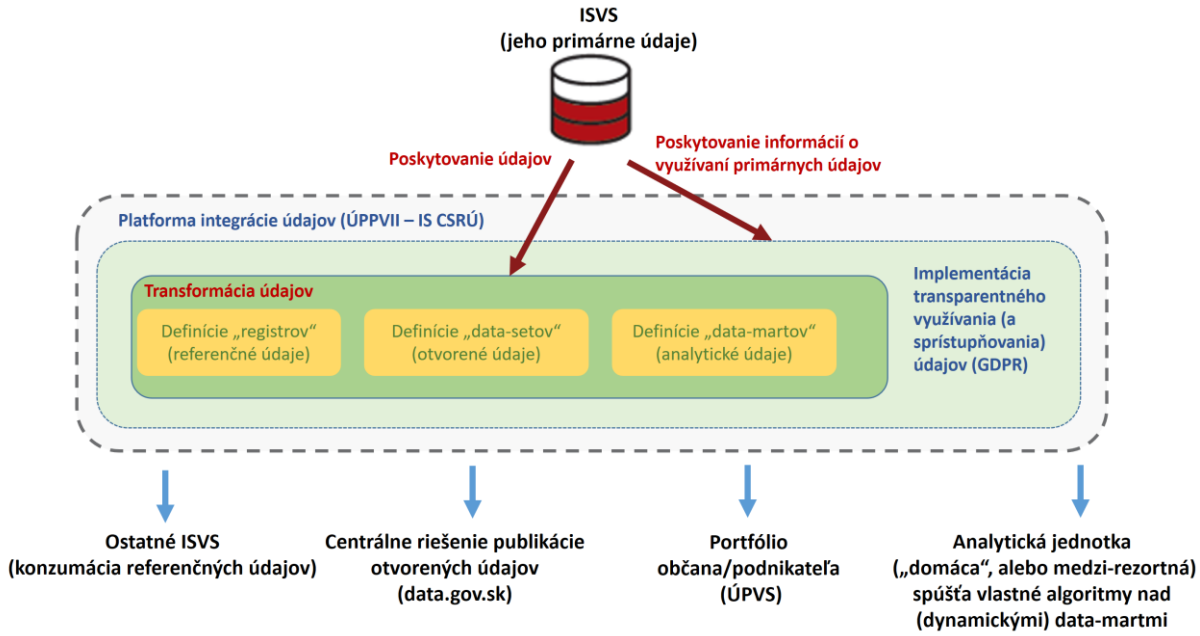
Poskytovanie uvedených typov údajov vždy predpokladá určitú transformáciu údajov z podoby ako sú vedené v zdrojovom systéme na podobu charakteristickú pre daný typ a ďalšie využitie. Pri referenčných údajoch hovoríme o transformácii do „registrov“, pri otvorených údajoch do „datasetov“ a pri analytickom spracovaní údajov do tzv. (dynamických) „data-martov“. Na pozadí sú vždy aplikované techniky dátovej transformácie z čoho vyplýva, že je možné pri každej transformácii využiť tie isté nástroje, tú istú platformu. V prípade centrálneho modelu poskytovania je možné navyše využiť synergický efekt z optimálneho využívania a zdieľania expertných kapacít dátových inžinierov - špecialistov. Pozitívny efekt centrálneho riešenia sa ešte znásobí, ak táto platforma zároveň zastreší povinnosť sprístupňovania

informácií o využívaní osobných údajov (tri zo štyroch spôsobov využívania o ktorých je potrebné informovať sa dejú priamo cez ňu, pričom pre štvrtý postačuje vytvoriť unifikované rozhranie pre všetky ISVS a ich správcov aby mohli tieto informácie dodať). Uvedené rekapituluje nasledovná schéma.:



Pohľad na platformu integrácie údajov (ktorý vychádza z usmernenia a architektonických schém uvedených v dokumente Referenčná architektúra IISVS) **z pozície jedného ISVS** tak vizualizuje nasledovná schéma, kde je možné vidieť jednotlivé vrstvy:

- vrstva integrácie údajov,
- vrstva transformácií pre referenčné, otvorené a analytické údaje,
- vrstva sprístupňovania informácií o využívaní osobných údajov (podľa smernice GDPR) – tzv. „Moje údaje“.:


ISVS
(jeho primárne údaje)
Poskytovanie údajov
Poskytovanie informácií o využívaní primárnych údajov
Platforma integrácie údajov (ÚPPVII – IS CSRÚ)
Transformácia údajov
Definície „registrov“ (referenčné údaje)
Definície „data-setov“ (otvorené údaje)
Definície „data-martov“ (analytické údaje)
Implementácia transparentného využívania (a sprístupňovania) údajov (GDPR)
Ostatné ISVS (konzumácia referenčných údajov)
Centrálne riešenie publikácie otvorených údajov (data.gov.sk)
Portfólio občana/podnikateľa (ÚPVS)
Analytická jednotka („domáca“, alebo medzi-rezortná) spúšťa vlastné algoritmy nad (dynamickými) data-martmi

7 Postup centrálnej úrovne a správcu v transformácii ISVS do natívnej cloudovej architektúry

Pri definovaní postupnosti krokov, je dôležité uvažovať vo fázach, ktoré definuje dokument *Akčný plán informatizácie* (kaptitola: *Postup OVM pri implementácii akčného plánu),* a ktoré vychádzajú z postupnosti implementácie definovanej už v NKIVS.:

1. Počas **prvej** fázy (s názvom „*1x a dosť a odstraňovanie bariér*“) sa OVM sústreďujú na zmeny na rozhraniach/údajoch/okrajoch (teda nemodifikujú „črevá“ biznis logiky) svojich ISVS. Natívnu cloudovú architektúru v tejto fáze primárne rozvíja hlavne centrálna úroveň, pričom OVM sa sústreďujú skôr na migráciu existujúcich systémov do cloudu (IaaS režim). Centrálna úroveň (ktorú reprezentuje architektonická kancelária VS), pripravuje natívnu cloudovú architektúru v nasledovnej postupnosti.:
    a. Vysokoúrovňová definícia (špecifikácia) služieb špecializovanej PaaS platformy, na základe vstupov z pripravovaných projektov *PaaS* (MVSR) a *Vytvorenie dôveryhodného prostredia (certifikácie) a Sprostredkovateľ modelu Hybridného vládneho cloudu* (ÚPPVII). Vysokoúrovňová definícia/špecifikácia sa stane súčasťou aktualizácie tohto dokumentu (predbežne do 12/2017).
    b. Na základe špecifikácie cieľového stavu, budú OVM - ktoré sa chystajú budovať nové ISVS – môcť uskutočniť tento vývoj v komerčných cloudových prostrediach tak, aby vo finálne boli tieto riešenia s minimálnou prácnosťou prenositeľné do finálnej podoby PaaS (hybridného) vládneho cloudu.
    c. V rámci projektu *Vytvorenie dôveryhodného prostredia (certifikácie) a Sprostredkovateľ modelu Hybridného vládneho cloudu* (ÚPPVII) bude v tejto fáze vykonaný pilot špecializovanej PaaS vrstvy (a overenia jej funkcionalít) a na základe jeho výsledkov budú priebežne aktualizované smernice pre migráciu do cloudu aj usmernenia k štandardizácii prevádzky v natívnej cloudovej architektúre.

    Paralelne sa v rámci tejto fázy rozvíjajú SaaS/BPaaS zdieľané moduly (bližšie informácie - výstupy z PS Digitalizácia agend VS, resp. aktualizovaný dokument Akčného plánu), ktoré sú tiež vstupom do finálnej podoby natívnej cloudovej architektúry.

2. Počas **druhej** fázy (s názvom „*Služby*“) sa OVM sústreďujú na zmeny v biznis logike
    a. automatizácia spracovania,
    b. „zbavenie sa“ častí biznis logiky, ktoré je možné zabezpečiť zdieľaním spoločných modulov/blokov,
    c. zreťazenie a orchestrácia služieb do komplexných životných situácií,
    d. **prechod systémov do natívnej cloudovej architektúry**.