

# Implementace integračních služeb

Metodický dokument

# 1 Historie dokumentu

## 2 Obsah

<b>1</b>	<b>Historie dokumentu .....</b>	<b>2</b>
<b>2</b>	<b>Obsah .....</b>	<b>3</b>
2.1	Seznam tabulek .....	5
2.2	Seznam obrázků .....	6
2.3	Seznam zkratek a pojmů .....	7
<b>3</b>	<b>Úvod.....</b>	<b>8</b>
3.1	Rozsah a cíle metodiky .....	8
3.2	Odchytky od standardu .....	8
<b>4</b>	<b>Integrace v IS VZP ČR.....</b>	<b>9</b>
4.1	Integrační platforma .....	9
4.2	Integrační vazba .....	9
4.2.1	Služba IPF .....	9
4.2.2	Cílová služba .....	10
4.2.3	Složky .....	10
4.2.4	API a služba.....	11
4.2.5	Životní cyklus služby .....	11
4.2.6	Verzování služeb .....	12
4.2.7	Podporované znakové sady .....	13
4.3	Spolupráce při tvorbě služeb.....	13
4.3.1	Scénář procesu tvorby/změny služby IPF .....	14
4.3.2	Scénář procesu nové konzumace služby IPF .....	16
4.3.3	Změna existující služby IPF .....	16
4.4	Předpoklady pro realizaci služeb IPF .....	17
4.5	Prostředí IPF.....	17
<b>5</b>	<b>Návrh .....</b>	<b>18</b>
5.1	Znovupoužitelnost služeb .....	18
5.2	Jmenná konvence služeb .....	18
5.2.1	SOAP a AQ služby.....	18
5.2.2	REST API .....	18
5.2.3	Další druhy služeb .....	19
5.3	Synchronní a asynchronní režim.....	19
5.4	Druhy služeb IPF .....	20
5.5	Rozhraní .....	21
5.5.1	SOAP služby.....	22
5.5.2	REST .....	26
5.5.3	Messaging .....	28
5.5.4	Asynchronní SOAP služby .....	31
5.5.5	SDI .....	31
5.6	Autentizace a autorizace.....	31
5.6.1	SOAP služby.....	31
5.6.2	REST API .....	32
5.7	Zabezpečení služeb.....	32
5.8	Objem přenášených dat.....	32
5.9	Dokumentace v Evidenci služeb .....	32
5.10	Testovací plán .....	33
5.11	Homologace návrhu.....	33
<b>6</b>	<b>Implementace.....</b>	<b>34</b>

6.1	Messaging .....	34
6.2	Ošetření nestandardních stavů .....	35
6.2.1	SOAP a AQ.....	35
6.2.2	REST .....	35
6.3	MDS .....	36
6.3.1	Úložiště XSD.....	36
6.3.2	Úložiště WSDL.....	36
6.4	BPEL .....	36
6.4.1	Standardní kroky BPEL procesů .....	37
6.4.2	Parametry BPEL procesů.....	37
6.4.3	REST služby.....	37
6.5	Oracle Service Bus .....	38
6.6	SDI .....	39
6.7	Validace.....	39
6.8	Šifrování údajů.....	39
6.8.1	Šifrování XML elementů v rámci IPF .....	39
6.9	Konfigurace služeb .....	39
6.9.1	BPEL .....	39
6.9.2	OSB.....	41
6.9.3	Doplnění konfiguračních plánů.....	41
6.10	Logování.....	42
6.10.1	Využití API NTL ve službách IPF .....	44
6.11	Homologace implementace.....	45
<b>7</b>	<b>Dokumentace .....</b>	<b>47</b>
7.1	Zdrojové kódy .....	47
<b>8</b>	<b>Testování.....</b>	<b>48</b>
8.1	Funkční testy .....	48
8.1.1	Využití testovacích scénářů .....	48
8.1.2	Struktura SoapUI projektu.....	49
8.1.3	Konfigurace prostředí.....	50
8.1.4	Asynchronní SOAP služby .....	51
8.2	Integrační testy .....	51
8.3	Výkonové testy .....	52
8.4	Protokol o provedení testu .....	52
<b>9</b>	<b>Předání k nasazení.....</b>	<b>53</b>
<b>10</b>	<b>Nasazení do provozu .....</b>	<b>54</b>
<b>11</b>	<b>Rozvoj služby .....</b>	<b>55</b>
<b>12</b>	<b>Provoz.....</b>	<b>56</b>
<b>13</b>	<b>Ukončení provozu služby.....</b>	<b>57</b>
<b>14</b>	<b>Přílohy .....</b>	<b>58</b>
14.1	Odkazy .....	58
14.2	VZPIPFZPRAVA.....	58
14.3	Povinné parametry XML zpráv .....	58

## 2.1 Seznam tabulek

Tabulka 1: Seznam zkratk a pojmů .....	7
Tabulka 2 Rozdělení služeb IPF do složek.....	10
Tabulka 3 Fáze životního cyklu služby .....	12
Tabulka 4 Životní cyklus služby IPF .....	12
Tabulka 5 Účastníci integrací .....	14
Tabulka 6 Scénář procesu tvorby/změny služby IPF .....	15
Tabulka 7 Proces nové konzumace služby IPF .....	16
Tabulka 8 Používaná prostředí IPF a jejich využití .....	17
Tabulka 9 Volba typu rozhraní .....	21
Tabulka 10 Standardní elementy požadavku .....	24
Tabulka 11 Standardní elementy odpovědi .....	24
Tabulka 12 Zasílání nestrukturovaných dat v XML.....	25
Tabulka 13 Parametry HTTP požadavku REST API vystaveného na IPF .....	27
Tabulka 14 Popis atributů VZPIPFZPRAVA .....	30
Tabulka 15 Pravidla pro vytváření a pojmenování AQ front.....	30
Tabulka 16 Limity objemů dat dle druhu komunikace .....	32
Tabulka 17 Předání návrhu služby IPF k homologaci.....	33
Tabulka 18 HTTP kódy pro nestandardní stavy REST služeb IPF.....	36
Tabulka 19 Parametry BPEL procesů .....	37
Tabulka 20 Standardní adresáře ve zdrojovém kódu OSB služby .....	38
Tabulka 21 - Konfigurační parametry služeb .....	41
Tabulka 22 NTL - Plnění atributů transakce (transaction) NTL .....	43
Tabulka 23 Plnění atributů události (event) NTL.....	44
Tabulka 24 Plnění atributů reference NTL.....	44
Tabulka 25 Homologace implementace .....	46
Tabulka 26 Umístění služeb v úložišti GIT .....	47
Tabulka 27 Standardní struktura testovacích projektů SoapUI .....	49
Tabulka 28 Rozvoj služby a dopad na verzování .....	55
Tabulka 29 Aktivity při ukončení provozu služby .....	57
Tabulka 30 Odkazy .....	58

## 2.2 Seznam obrázků

Obrázek 1 Schéma integrační vazby.....	9
Obrázek 2 Synchronní komunikace.....	19
Obrázek 3 Znázornění asynchronní komunikace .....	20
Obrázek 4 Schéma typické AQ služby .....	28
Obrázek 5 Příklad realizace REST služby.....	38
Obrázek 6 Typické využití API NTL.....	44
Obrázek 7 Zápis do NTL v SOA službách.....	45
Obrázek 8 Využití SoapUI testovacích scénářů.....	48
Obrázek 9 Struktura standardního SoapUI projektu .....	50
Obrázek 10 Nastavení prostředí v SoapUI projektu .....	51

## 2.3 Seznam zkratk a pojmů

Zkratka	Význam
AQ	Advanced Queuing, technologie zasílání zpráv s pomocí Oracle DB
API	Application Programming Interface – programové rozhraní aplikace, typicky obsahuje jednu či více služeb.
Base64	Kódování, které převádí binární data do posloupnosti tisknutelných znaků.
BPEL	Business Process Execution Language – jazyk pro tvorbu webových služeb používaný v SOA Suite.
DB	Databáze
IPF	Integrační platforma VZP ČR
IS VZP ČR	Informační systém VZP ČR, v textu míněn jako celek zahrnující dílčí komponenty (aplikace, informační systémy, integrace a pomocné nástroje).
KK	Konzumující komponenta
MDS	Metadata Services, úložiště metadat služeb v rámci Oracle SOA Suite
NTL	NIS Transakční Log, komponenta, která zajišťuje jednotný formát logování transakcí.
ORI	Oddělení rozvoje integrací VZP
OSB	Oracle Service Bus, komponenta z Oracle SOA Suite pro vytváření synchronních služeb
PK	Poskytující komponenta
PTP	Poskytovatel technické podpory IPF – dodavatel nebo oddělení, které je pověřeno technickou podporou IPF
REST	REpresentational State Transfer, koncept tvorby programového rozhraní (API)
SDI	Silná datová integrace
SOA	<i>Service Oriented Architecture</i> , architektura orientovaná na služby
SOAP	<i>Simple Object Access Protocol</i> , komunikační protokol webových služeb.
OpenAPI	Standard pro popis REST rozhraní
WSDL	Web Services Description Language – XML jazyk pro popis webových služeb

**Tabulka 1: Seznam zkratk a pojmů**

## 3 Úvod

Tento dokument obsahuje metodické pokyny pro realizaci integrací v rámci informačního systému VZP ČR.

### 3.1 Rozsah a cíle metodiky

Metodika pokrývá oblasti:

- technologické standardy, jakým způsobem integrace vytvářet
- pracovní postupy zúčastněných stran
- použití, zabezpečení, správa a monitoring integračních služeb

Metodika navazuje na dokument *Standardy IS VZP – NIS*, který se integracemi zabývá v obecné rovině, a doplňuje integrační standardy na úroveň detailu potřebnou pro návrh, implementaci a správu integrací.

Cílem metodiky je seznámit pracovníky VZP ČR a její dodavatele s možnostmi integrace v IS VZP, určit jednotná pravidla pro tvorbu integrací, sjednotit související pracovní postupy a tím přispět k efektivnímu rozvoji a správě IS VZP.

### 3.2 Odchytky od standardu

Správce a garantem standardu je Oddělení architektury ICT.

Pokud dodávka není v souladu s integračními standardy uvedenými v tomto dokumentu, je zapotřebí ji uvést do souladu se standardy nebo vyžádat výjimku od Oddělení architektury ICT.



## 4 Integrace v IS VZP ČR

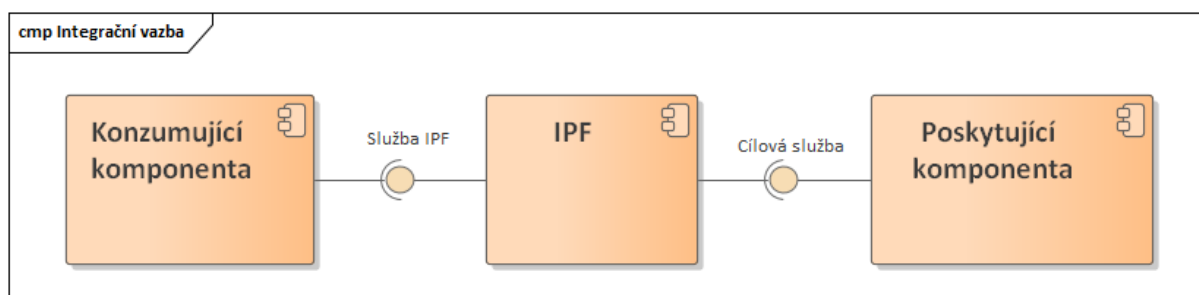
### 4.1 Integrační platforma

Integrační platforma VZP ČR (dále IPF) propojuje jednotlivé komponenty IS VZP pomocí integračních vazeb. Poskytující komponenta vystavuje službu v rámci svého aplikačního rozhraní (API).

Komponenta, která chce službu konzumovat, se však nepřipojí k této službě napřímo, nýbrž ke službě IPF, která komunikaci zprostředkuje.

Každá integrační vazba se tedy skládá ze:

- služby poskytující komponenty
- služby IPF
- konzumující komponenty, která službu využívá



**Obrázek 1 Schéma integrační vazby**

Smyslem zapojení IPF do integrační vazby je:

- vymezit technologie integrací, tím zjednodušit konzumaci služeb a zvýšit znovupoužitelnost služeb
- sjednotit zabezpečení služeb a omezit bezpečnostní riziko neautorizované a nemonitorované konzumace služeb
- sjednotit monitoring a řešení chyb provozu integrací
- správa a využívání katalogu sdílených služeb v rámci IS VZP

### 4.2 Integrační vazba

Integrační vazbou míníme propojení konzumenta s poskytovatelem pomocí služby IPF.

Příklad:

*Komponenta FINANCE konzumuje službu IPF DetailSubjektuZP, jejíž zdrojovou službu poskytuje komponenta RSZP.*

#### 4.2.1 Služba IPF

Služba je v rámci IPF základní jednotka pro integrace. Služba má svůj název, verzi a množinu funkcí, které poskytuje. Služba může mít více operací (např. SOAP) a poskytovat komplexní funkčnost (např. REST API).

Konzument si vybere vhodnou službu na základě svých potřeb dle popisu v Evidenci služeb. Konzument se nemusí zajímat o implementační detaily na straně IPF nebo poskytující komponenty.

## 4.2.2 Cílová služba

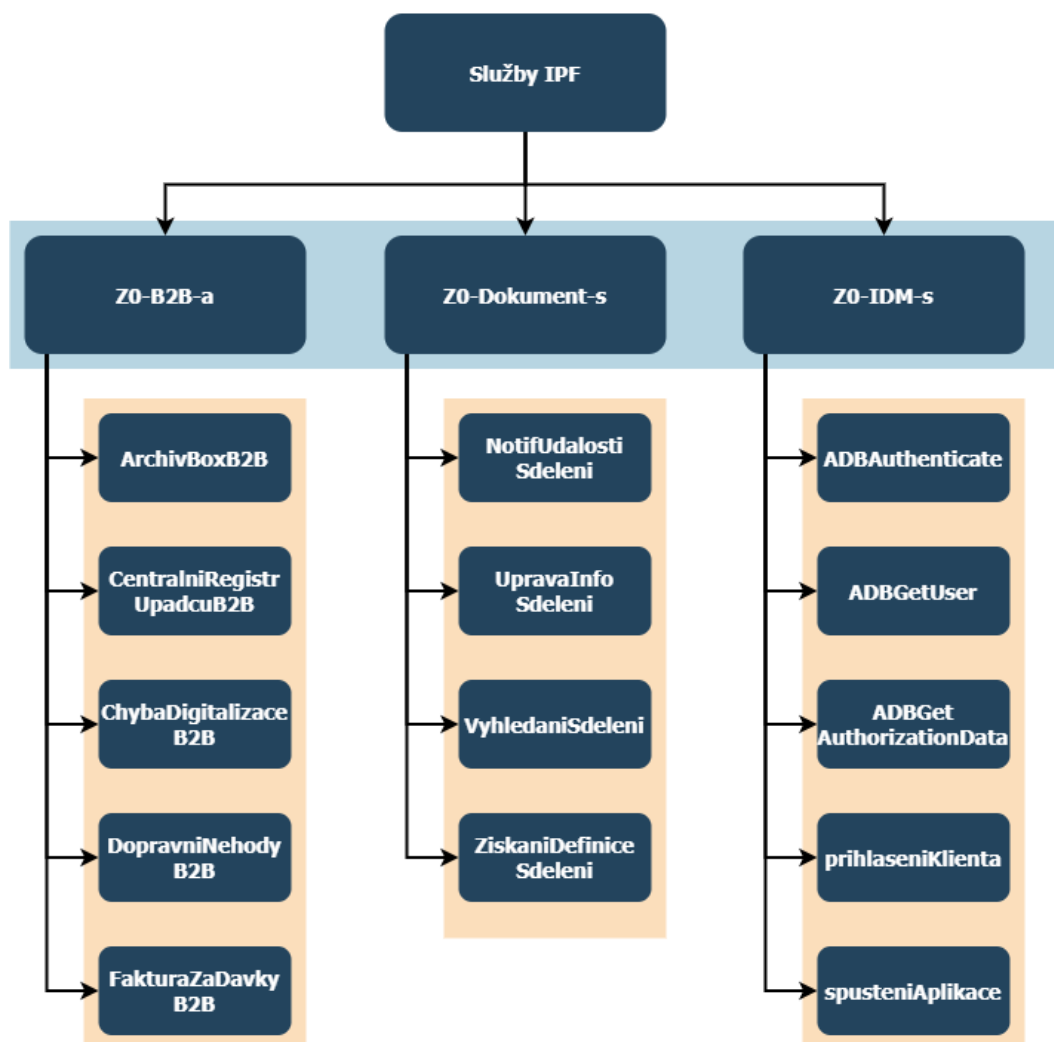
Skutečnou práci obvykle odvádí cílová služba. Ve většině případů je cílová služba realizována v některé z komponent IS VZP, pracuje s daty a programovou logikou cílové komponenty. Služba IPF v takovém případě přešle požadavek na cílovou službu, přijme její odpověď a tu opět vrátí konzumentovi. Toto je nejběžnější scénář integračních vazeb v rámci IS VZP.

Složitější služby IPF provádí transformaci předávaných zpráv nebo koordinují komunikaci s více dalšími službami (tuto funkci IPF říkáme *orchestrace*).

Cílová služba může být poskytována některou z komponent IS VZP nebo externím systémem (např. Informačním systémem základních registrů).

## 4.2.3 Složky

Služby jsou v rámci IPF rozděleny do složek. Složka je logické členění, podobně jako složky (adresáře) na disku organizují soubory (viz následující obrázek).



**Tabulka 2 Rozdělení služeb IPF do složek**

Název složky je ve tvaru: *XY-Komponenta-Typ*, kde

- X: písmeno určující dodavatele (VZP náleží písmeno A)
- Y: číslice 0-9, rozlišení složek v rámci jednoho dodavatele

- *Komponenta*: název aplikační komponenty IS, která službu poskytuje případně obchodní domény, k níž služba náleží, pokud aplikační komponenta nemá svoji složku nebo je služba vystavena pro více aplikačních komponent.
- *Typ*: označení režimu služby:
  - s: synchronní
  - a: asynchronní

Příklady složek:

- *A0-EVZP-s*: dodavatel integračních služeb VZP, komponenta eVZP, synchronní služby
- *B0-RSZP-a*: dodavatel DXC, komponenta RSZP, asynchronní služby

Na úrovni složky lze na IPF nastavit specifická přístupová oprávnění (pro pracovníky VZP a příslušného dodavatele) a přidělovat zdroje IPF.

Požadavek na založení, změnu a zrušení složky je potřeba vždy odsouhlasit s Architektury VZP již ve fázi návrhu služby.

#### 4.2.4 API a služba

Pojmem API rozumíme kompletní rozhraní vystavené aplikací. V kontextu SOAP může být API realizováno jednou SOAP službou s mnoha operacemi různého významu, typicky však více SOAP službami.

REST rozhraní téměř vždy obsahuje více funkcí, které jsou v rámci API reprezentovány cestami (path). V rámci integrační metodiky IS VZP považujeme celé REST API za jednu službu s různými operacemi.

#### 4.2.5 Životní cyklus služby

Služba IPF postupně prochází následujícími fázemi:

Fáze	Popis
<b>Navrhovaná</b>	Služba (verze služby) je ve fázi návrhu, návrh dosud nebyl schválen v procesu homologace návrhu.  Fáze zahrnuje období od vzniku motivace vytvořit novou službu (popřípadě novou verzi) až po úspěšné dokončení homologace návrhu.
<b>Vyvíjená</b>	Fáze začíná v okamžiku, kdy je schválen návrh služby. Dodavatel službu vyvíjí a provádí interní a integrační testy.  V závěru fáze je služba předána k nasazení na akceptační prostředí.
<b>Testovaná</b>	Služba je nasazena na akceptační prostředí, kde probíhají akceptační testy.  Souběžně probíhá homologace implementace (služby IPF).  Fáze je zakončena úspěšnou realizací akceptačních testů a předáním k nasazení do provozu.
<b>Nasazená</b>	Služba je nasazena na provozní prostředí. Služba může být využívána konzumenty.

<b>Ukončená</b>	Služba je ukončena, vyřazena z provozu. Jde o konečný stav životního cyklu služby. Služba je odebrána z příslušné komponenty IS VZP.
-----------------	---

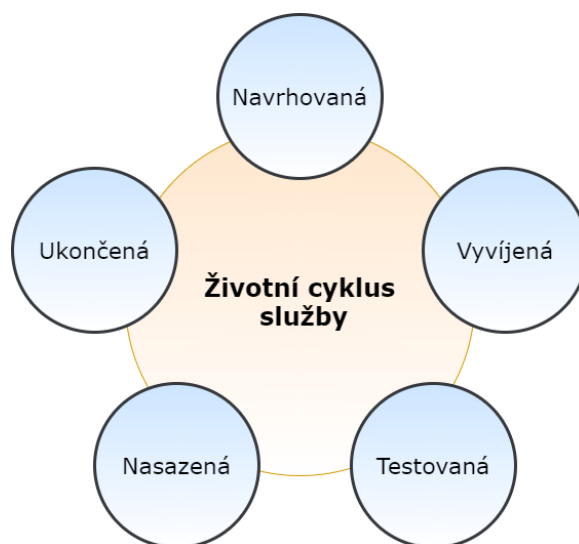
### Tabulka 3 Fáze životního cyklu služby

Souhrnně tyto fáze nazýváme životním cyklem služby.

Zjednodušeně hovoříme (v rámci lepší srozumitelnosti) o fázích v životním cyklu *služby*. Životní cyklus se však vztahuje k *verzi služby*, tj. každá verze služby postupně prochází jednotlivými fázemi.

### Příklad

Služba *Pojistenec* ve verzi 1.0 je ve fázi Nasazená a nově vytvářená verze 1.1 je teprve ve fázi Navrhovaná.



Tabulka 4 Životní cyklus služby IPF

## 4.2.6 Verzování služeb

Verze služeb je označena dvěma čísly: hlavním a vedlejším, např. 1.5 (hlavní číslem je zde 1, vedlejší číslo 5).

Pokud dojde ke změně služby, která není zpětně kompatibilní, musí být zvýšena hlavní verze. V ostatních případech je zvýšena vedlejší verze.

Příklady zpětně nekompatibilních změn:

- Přidání povinného elementu do požadavku služby
- Změna odpovědi služby (struktury, datového typu aj.)
- Odebrání operace služby

Příklady zpětně kompatibilních změn:

- Přidání nové operace služby
- Přidání nepovinného elementu do požadavku služby
- Oprava programové chyby

Pro všechny verze služby je uložen zdrojový kód v úložišti GIT, viz 7.1 Zdrojové kódy.

#### 4.2.6.1 Služby IPF

Ve většině případů nahrazuje nová verze služby předchozí verze a je tedy dostupná pouze poslední verze služby IPF. Nasazením nové verze služby IPF se nemění její adresa.

Pokud změna služby není zpětně kompatibilní (je zvýšena hlavní verze služby), musí zároveň se změnou služby dojít ke změnám souvisejících komponent (typicky poskytovatel a konzument služby) nebo je nová verze služby nasazena jako nová služba (viz kapitola 11 Rozvoj služby).

#### 4.2.6.2 REST API

Pro REST API je v URL API umístěna pouze hlavní verze (*http://host/nazevAPI/verze*), např. *http://server.vzp.cz/Aplikace/1*.

Pokud je vhodné využívat vedlejší verzi API, slouží pouze pro účely dokumentace (k jakým zpětně kompatibilním změnám došlo apod.) a není součástí URL.

#### 4.2.7 Podporované znakové sady

Služby IPF využívají pro komunikaci standardní protokoly (HTTP, SOAP apod.) a datové formáty (XML apod.), které umožňují určit znakovou sadu přenášených dat.

Omezením je datový přenos realizovaný pomocí databáze IPF (např. AQ komunikace nebo SDI), která využívá kódování ISO-8859-2. U těchto druhů služeb nejsou podporovány znaky, které nejsou obsaženy v ISO-8859-2. Plnou znakovou sadu Unicode kromě databáze IPF nepodporují rovněž některé cílové systémy.

Do doby, než bude systémově zajištěna podpora Unicode znaků, je doporučeno pro integrace využívat pouze znaky dostupné v sadě ISO-8859-2.

### 4.3 Spolupráce při tvorbě služeb

Na realizaci integrační vazby typicky spolupracuje více stran: dodavatel poskytující komponenty, dodavatel služeb IPF a dodavatel konzumující komponenty.

Zatímco projekt je často zaměřen na dílčí cíle a naplnění aktuálních potřeb, homologační aktivity slouží k tomu, aby integrace byly efektivní a spolehlivé i v dlouhodobém horizontu (znovupoužitelnost služeb, efektivní podpora apod.)

Na tvorbě služeb IPF se podílejí následující role:

Role	Popis	Organizace
<b>Architekt KK</b>	Architekt konzumující komponenty	VZP
<b>Dodavatel KK</b>	Dodavatel konzumující komponenty	VZP/Dodavatel
<b>Dodavatel služby IPF</b>	Dodavatel služby IPF, která bude zprostředkovávat konzumaci cílové služby.	VZP/Dodavatel
<b>Architekt PK</b>	Architekt poskytující komponenty, která vystaví cílovou službu. Architekt je garantem služeb poskytovaných komponentou. Věcné (obchodní) náměty řeší s věcným garantem komponenty.	VZP
<b>PTP (Poskytovatel technické podpory) IPF</b>	Poskytovatel technické podpory IPF.	Dodavatel
<b>ORI</b>	Oddělení rozvoje integrací, zajišťuje homologaci služeb za VZP	VZP

<b>Administrátor IPF</b>	Administrátor zajišťující provoz služeb IPF a platformy IPF.	VZP
<b>Administrátor PK</b>	Administrátor, který nasazuje změny PK a s nimi i cílové služby potřebné pro služby IPF.	VZP
<b>Garant PK</b>	Věcný garant služby, zodpovídá za rozhodnutí ohledně funkčnosti služby (vytvoření služby, poskytované funkce, změny rozhraní, ukončení služby).	VZP
<b>Dodavatel PK</b>	Dodavatel poskytující komponenty	Dodavatel
<b>Projektový vedoucí, koordinátor změny</b>	Koordinuje zúčastněné role. Zajišťuje a zodpovídá za hladký průběh vytvoření integrační vazby. Účastní se všech níže uvedených kroků.	VZP
<b>Oddělení bezpečnosti</b>	Posouzení bezpečnosti využívaných datových aktiv.	VZP
<b>Oddělení klientské podpory</b>	Oddělení VZP, které poskytuje klientskou podporu. V některých případech hlášené incidenty z aplikací souvisí s integracemi. Pracovníci klientské podpory pak analyzují integrační vazby, včetně instancí služeb IPF a údajů v Evidenci služeb.	VZP
<b>Garant služby</b>	Odpovídá za funkčnost služby z věcného hlediska. Rozhoduje o změnách služby, případně o ukončení služby.	VZP

Tabulka 5 Účastníci integrací

### 4.3.1 Scénář procesu tvorby/změny služby IPF

Nové služby IPF nebo významné změny služby IPF jsou realizovány následujícím procesem. Každá změna je evidována v Servis desku a řízena procesem Delivery Framework. Tento proces popisuje návaznost na Delivery Framework a vlastní realizaci integrační služby:

<b>Aktivita</b>	<b>Popis</b>	<b>Zodpovědnost</b>
<b>Příprava koncepce řešení</b>	V rámci přípravy koncepce řešení potřebujeme integraci s další komponentou IS VZP. Žádná ze stávajících služeb naše potřeby nenaplnuje. Vzniká požadavek na novou službu (změnu služby) a její základní popis.	Solution architekt (A, R) Správce apl. Domény (C) Integrační architekt (C)
<b>Posouzení koncepce řešení</b>	Oddělení architektury posoudí záměr vytvořit novou (změnit) službu, dojde k vyjasnění potřeb konzumujícího systému, identifikace poskytujících komponent. OAR si vyžádá od KCI souhlasné stanovisko.	OAR (A, R) KCI (C)
<b>Návrh rozhraní cílové služby</b>	Dodavatel poskytujícího systému vytvoří návrh nové (změny) služby (název, rozhraní, technologie, režim).	Dodavatel PK (A, R)

	Dodavatel popíše návrh cílové služby v Evidenci služeb.	
<b>Návrh služby IPF</b>	Dodavatel služby IPF může návrh připomínkovat a vyžádat úpravy vzhledem ke standardům služeb IPF. Každá změna se vrací k posouzení návrhu.  Dodavatel služby IPF zaeviduje návrh služby IPF do Evidence služeb. Cílovou službu zdokumentuje v Evidenci služeb dodavatel poskytující komponenty.	Dodavatel služby IPF (A, R)
<b>Homologace návrhu služby</b>	ORI a PTP posoudí a případně připomínkují návrh služby v procesu homologace návrhu (více viz 5.11 Homologace návrhu). Solution architekt nebo PM zajistí vypořádání případných připomínek.	ORI (A, R) PTP (R) KCI (I)
<b>Implementace? cílové služby</b>	Dodavatel PK implementuje službu se schváleným rozhraním. Provede jednotkové testy služby pro každý z případů použití služby (pozitivní i vhodné negativní scénáře). Doplní údaje služby v rámci Evidence služeb.	Dodavatel PK (A, R)
<b>Implementace služby IPF</b>	Dodavatel služby IPF implementuje na základě specifikace cílové služby (WSDL, OpenAPI nebo jiná, dle typu služby) novou (změněnou) službu IPF. Doplní údaje služby v Evidenci služeb. Dodavatel služby předá zdrojový kód a další náležitosti do Azure DevOps.	Dodavatel služby IPF (A, R)
<b>Integrační test</b>	Služba IPF a Cílová služba poskytující komponenty je ověřena integračním testem. Po úspěšném ukončení testů dodavatelé aktualizují Evidenci služeb.	Dodavatel PK (A, R) Dodavatel služby IPF (R)
<b>Homologace implementace služby IPF</b>	Služba IPF prochází homologací implementace (více viz 6.11 Homologace implementace). Dodavatel služby IPF zapracuje případné připomínky (otestuje provedené změny) a aktualizuje zdrojový kód v Azure DevOps.	ORI (A, R) PTP (R) KCI (I) OŘ testování (I)
<b>Akceptační test</b>	OTP nasadí změnu komponenty a službu IPF v akceptačním prostředí. Služba je ověřena v rámci akceptačních testů.	OŘT (A, R) OTP (R)
<b>Nasazení do provozu</b>	OTP nasadí změnu komponenty a službu IPF v provozním prostředí dle schváleného harmonogramu nasazování na produkční prostředí.	OTP (A, R)

**Tabulka 6 Scénář procesu tvorby/změny služby IPF**

Zkratky R, A, C, I uvádějí zodpovědnost dotyčné role:

- R = Responsible / Realize = provádí
- A = Accountable = odpovídá
- C = Consulted = připomínkuje
- I = Informed = je informován

### 4.3.2 Scénář procesu nové konzumace služby IPF

Scénář procesu nové konzumace služby IPF bez změny služby IPF nebo Cílové služby. Každá změna je evidována v Servis desku a řízena procesem Delivery Framework. Tento scénář popisuje návaznost na Delivery Framework v případě nové konzumace již existující služby IPF:

Aktivita	Popis	Zodpovědnost
<b>Klasifikace požadavku</b>	Správce apl. Domény (poskytující komponenty) posoudí a schválí požadavek. V případě souhlasu za poskytující komponentu předá požadavek na posouzení dopadu do architektury.	Správce apl. Domény (A, R)
<b>Posouzení dopadu do architektury</b>	Oddělení architektury posoudí dopad do architektury. OAR si vyžádá KCI souhlasné stanovisko.	OAR (A, R) KCI (C)
<b>Integrační test</b>	Po úspěšném ukončení testů aktualizuje Evidenci služeb.	Dodavatel PK (A, R) Dodavatel služby IPF (R)
<b>Akceptační test</b>	OTP nasadí změnu, změna je ověřena v rámci akceptačních testů.	OŘT (A, R) OTP (R)
<b>Nasazení do provozu</b>	OTP nasadí změnu v provozním prostředí dle schváleného harmonogramu nasazování na produkční prostředí.	OTP (A, R)

Tabulka 7 Proces nové konzumace služby IPF

### 4.3.3 Změna existující služby IPF

Změny služeb rozlišujeme dle typu na:

- **Změny rozhraní:** dochází ke změně vystaveného rozhraní.
- **Změny implementace:** změna vnitřní implementace služby. Může dojít k náhradě některé z cílových služeb, transformaci a mapování údajů v rámci služby IPF, změně logování apod.

Dle rozsahu rozlišujeme změny:

- **Významné:** doplňují nebo odebírají funkčnost, mění chování služby nebo změny rozhraní.
- **Drobné:** změny v konfiguračních parametrech, opravy transformací zpráv, přidání logování a další změny, které nemohou mít negativní dopad na provoz IPF.

Drobné změny nemusí procházet homologací rozhraní a implementace, musí být však náležitě zdokumentovány v Evidenci služeb.

V ostatních ohledech je proces realizace změny služby analogický k postupu uvedenému výše v 4.3.1 a řídí se pravidly Change a Release managementu VZP.



## 4.4 Předpoklady pro realizaci služeb IPF

Vývojáři služeb IPF musí být před zapojením do prací na službách IPF náležitě proškoleni. Cílem je omezit riziko, že neproškolený vývojář ohrozí provoz platformy IPF nebo jejích služeb. Před zahájením prací na službách IPF musí být vývojář:

- Dostatečně proškolen v oblasti technologií a standardů využívaných při integracích: HTTP, SOAP, WSDL, XML, XML Schema, XSLT, REST, JSON, Messaging, relační DB/Oracle DB.
- Proškolen pro vývoj v prostředí produktů Oracle na nichž je postavena IPF: Oracle SOA Suite/Oracle Service Bus, BPEL, Oracle DB, Oracle Advanced Queuing. Tato školení jsou zajišťována společností Oracle a jejími partnery.
- Absolvovat školení pro vývojáře IPF VZP, v rámci něhož získá znalosti integrací v rámci VZP, platformy IPF a standardů pro vývoj služeb.

## 4.5 Prostředí IPF

Při vytváření služeb IPF jsou nejčastěji využívána tato prostředí:

Prostředí	Využití
PROD	Provozní prostředí, produkční běh služeb IPF. Na IPF nasazují služby administrátoři IPF.
Před-prod	Předprodukční prostředí, v maximální míře podobné prostředí produkčnímu, slouží pro ověření funkčnosti bez dopadu na produkční procesy a data.
Prostředí pro UAT	Akceptační prostředí. V tomto prostředí bývají realizovány akceptační testy. Na IPF nasazují služby administrátoři IPF.
Integrační	Integrační prostředí. Na IPF nasazují dodavatelé služeb IPF.
DEV	Vývojové prostředí dodavatele. Dodavatel služeb IPF jej používá pro vývoj služeb a interní testy.
SIMU	Speciální simulační prostředí pro komunikaci s komponentou B2B. Využívá se pouze služby typu B2B adaptér.

**Tabulka 8** Používaná prostředí IPF a jejich využití

## 5 Návrh

### 5.1 Znovupoužitelnost služeb

Před zahájením realizace nové služby je nutné prověřit, zda ve stávajícím portfoliu služeb IPF existují služby, které příslušné požadavky naplňují:

- pokud existující služby vyhovují plně, jsou pro integraci použity
- pokud vyhovují částečně, je zváženo, zda by požadovaná úprava nebyla vhodná i pro další (stávající či plánované) konzumenty služeb – řešením je v takovém případě vytvořit novou verzi existující služby
- pokud nelze nebo není vhodné využít stávající službu, může být vytvořena služba nová.

Nové služby jsou navrhovány takovým způsobem, aby byly obecně použitelné. Návrh služby nemá být závislý na konkrétním konzumentu.

### 5.2 Jmenná konvence služeb

#### 5.2.1 SOAP a AQ služby

Název služby je vytvářen dle vzoru *AaaBbbCcc*, kde *Aaa/Bbb/Ccc* jsou slova (obvykle tedy 1-3 podstatná jména) vyjadřující co nejpřesněji účel nebo funkci služby. Z názvu služby má být na první pohled jasné, k čemu služba slouží. Pokud by byl název příliš dlouhý, je možno slova zkrátit. Při použití zkratky je však zapotřebí, aby název služby zůstal srozumitelný a jednoznačný.

Příklady názvů služeb: *AktualizaceCiselniku*, *PausalLimit*.

SOAP služba může mít jednu či více operací. Operaci pojmenujeme dle vzoru *aaaBbb*, tj. jedním či dvěma slovy, které operaci nejlépe vystihují.

Příklad názvů operací: *smazDokument*, *vytvorDokument*, *aktualizujVyuctovani*.

Název služby posuzuje a schvaluje v době návrhu Integrovační architekt (cílem je zajistit, aby byl název jednoznačný a výstižný).

#### 5.2.2 REST API

Při pojmenování se držíme konceptu REST, ve kterém data reprezentují zdroj (resource). Zdroj může být technické povahy (dokument, uživatel, účet apod.) nebo reprezentovat objekt skutečného světa (např. osoba).

Zdroje adresujeme pomocí URI (*Uniform Resource Identifier*). URI jsou pojmenovány tak, aby srozumitelně určovaly příslušný zdroj.

V případě, že je zdroj:

- kolekce, pojmenujeme ji podstatným jménem v množném čísle: např. */subjekty*, */faktury* apod. Na konkrétní zdroj se odkážeme identifikátorem: */subjekty/{subjektId}*
- jedinec (singleton), pojmenujeme jej jednotným číslem, např. */konfigurace*

Zdroj může obsahovat podřízené kolekce, které uspořádáváme hierarchicky, např. */faktury/{fakturald}/polozky*.

Akce (kontrolery) jsou nazvány slovesem, např. kontroler pro uhrazení faktury bude */faktura/{fakturald}/uhrad*.

V případě, že je vhodné zdroj pojmenovat víceslovně, oddělujeme jednotlivá slova pomlčkou, např. */archivni-vina*.

### 5.2.3 Další druhy služeb

Ostatní druhy služeb (SDI, HTTP apod.) pojmenováváme dle vzoru *AaaBbbCcc*, jak je popsáno v odstavci 5.2.1 SOAP a AQ služby.

## 5.3 Synchronní a asynchronní režim

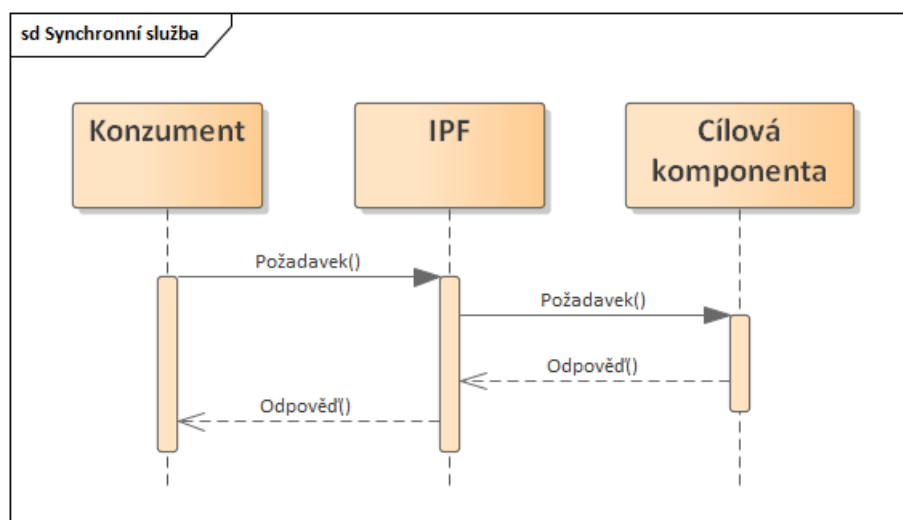
Při návrhu režimu služeb zohledňujeme:

- **zpracování na straně poskytovatele služby:** poskytující systém musí být schopen službu zpracovat a vrátit odpověď dostatečně rychle a zároveň ne být zpracováním služby přetížen
- **jednoduchost (efektivitu) konzumace služby:** jednoduché služby mají v rámci IS výrazně vyšší využitelnost a snižují náklady na rozvoj IS

Služby navrhujeme jako **synchronní** (*syn – spolu, chronos – čas: komunikace ve společném čase*) v případech, kdy platí:

- konzument z povahy služby potřebuje odpověď k pokračování svého procesu (např. interakce uživatele v GUI nebo časově vymezené úlohy)
- je výpočetně a časově nenáročné poskytnout odpověď

Synchronní služby umožňují jednodušší a efektivnější napojení konzumujícího systému. Konzument však musí být připraven na nedostupnost služby a vhodně ji ošetřit. Synchronní služby jsou rovněž jednodušší z hlediska provozu.



**Obrázek 2 Synchronní komunikace**

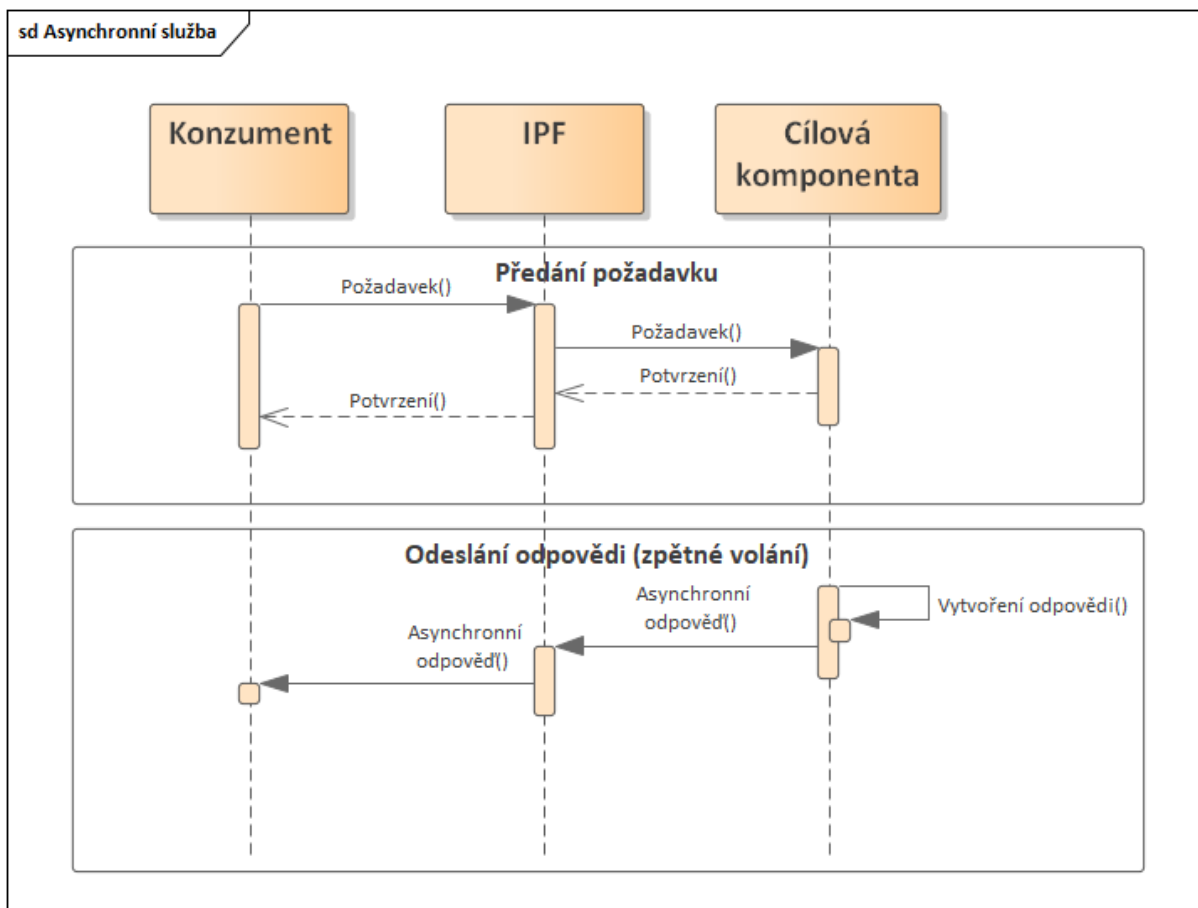
Služby navrhujeme jako **asynchronní**, pokud:

- zpracování je asynchronní ze své povahy (např. proces schválení, který obsahuje lidskou činnost)
- zpracování služby a vytvoření odpovědi je časově nebo výpočetně náročné (např. dávkové zpracování)
- konzument může pokračovat v procesu i bez odpovědi služby

Asynchronní služby umožňují časově oddělit odeslání požadavku od zpracování odpovědi. Konzumující systém musí vyřešit propojení požadavku s pozdějším zpracováním odpovědi služby (tzv. korelaci), což znamená vyšší složitost zpracování.

Asynchronní služby samy o sobě negarantují pořadí zpracování zpráv na straně poskytovatele služby nebo konzumenta. Zpracování zpráv v určeném pořadí lze zajistit přidanou aplikační logikou.

Asynchronní zpracování umožňuje volnou vazbu mezi systémy, pokud je zvolen vhodný přenosový protokol (např. messaging).



Obrázek 3 Znáznornění asynchronní komunikace

## 5.4 Druhy služeb IPF

Služby IPF dělíme do následujících druhů:

- **Atomické:** služba IPF zajišťuje pouze směrování na cílovou službu, případně provádí jednoduchou transformaci.
- **Kompozitní:** služba IPF provádí transformaci zpráv, komunikuje s více než jednou cílovou službou nebo obsahuje specifickou programovou logiku

Preferované jsou služby atomické. Pokud je služba navržena jako kompozitní nebo choreografie, musí být schválena jako výjimka.

Služby kompozitní, které obsahují aplikační logiku, budou evidovány v aplikačním portfoliu VZP a budou mít přiřazeného správce a vlastníka.

## 5.5 Rozhraní

Typ rozhraní volíme dle povahy a účelu služby. Pro synchronní služby volíme mezi SOAP případně REST, při volbě zohledníme, jaký typ rozhraní podporuje poskytovatel a předpokládání konzumenti služby. Je-li podporován SOAP, má tento přednost před REST rozhraním služeb.

Pokud je do IS VZP nasazován hotový (krabicový) systém, služby IPF by měly odpovídat rozhraní poskytovacího systému (bez změn názvů a typů rozhraní).

Pro asynchronní služby, kde chceme realizovat volnou vazbu mezi komponentami, volíme messaging. Je-li systémem podporován Oracle Advanced Queuing (AQ) je preferován. Není-li AQ podporován poskytovacím nebo konzumujícím systémem, bude messaging vnitřním rozhraním služby a pro vnější rozhraní služby zvolíme asynchronní SOAP.

Pro asynchronní integrační vazby, kde není vhodné použít messaging může být vytvořena asynchronní SOAP služba. Asynchronní vazba realizována dvěma synchronními SOAP službami není podporováním návrhovým vzorem.

Pokud konzument často využívá rozsáhlou množinu dat, která primárně udržuje poskytovatel a konzument i poskytovatel ukládají data v Oracle databázi, lze využít SDI (Silná Datová Integrace). SDI se však využije pouze v opodstatněných případech, kde není vhodné zvolit některý z výše uvedených způsobů integrace. Vytvoření nové vazby SDI podléhá schválení integračního architekta.

Volbu typu rozhraní shrnuje následující tabulka.

Režim	Podmínky	Typ
Synchronní	Pro konzumenta a poskytovatele je efektivnější využít SOAP standard.	SOAP
	Pro konzumenta a poskytovatele je efektivnější využít REST.	REST
Asynchronní	Potřebujeme zajistit doručitelnost zpráv. Mezi komponentami má být volná vazba, aby při nedostupnosti jedné z komponent integrační vazby mohlo pokračovat zpracování. Asynchronní je celá integrační vazba.	AQ/Messaging
	Pro konzumenta a poskytovatele není vhodné využívat Messaging (např. nedisponují technologií Oracle AQ). Odeslání, zpracování a potvrzení požadavku jsou synchronní. Následuje asynchronní příprava odpovědi poskytovacím systémem. Zaslání odpovědi v opačném směru je opět synchronní.	Asynch. SOAP
Velký rozsah dat	Konzument potřebuje pro své funkce velký rozsah dat. Data využívá často a potřebuje je stále synchronizovat. Typickým případem jsou číselníky a registry.	SDI

**Tabulka 9 Volba typu rozhraní**

Objekty přenášené pomocí SDI by měly mít velikost do 100 MB. Pro replikace větších objemů dat mezi komponentami by měly být využívány ETL procesy. ETL procesy nejsou realizovány službami IPF a jejich popis není součástí této metodiky.

## 5.5.1 SOAP služby

SOAP služby jsou definované WSDL dokumentem, který se může odkazovat na další XSD definice. Formátem přenášených dat je vždy XML.

### 5.5.1.1 Pravidla pro WSDL

Atribut *name* v elementu *definitions* obsahuje název služby (viz 5.2 Jmenná konvence služeb).

Namespace služby je ve tvaru <http://vzp.cz/RozlisujiciOznaceni/NazevSluzby>, kde:

- *RozlisujiciOznaceni* je nepovinné, určuje aplikaci nebo doménu, ze které služba pochází, např. Finance
- *NazevSluzby*: udává samotný název služby, např. *IndividualniHodnotaBodu*

Příklad namespace: <http://vzp.cz/Finance/IndividualniHodnotaBodu>

Služba může mít jednu či více operací, všechny operace však musí funkčně náležet k dané službě. Služba nemá v rámci svých operací sdružovat nesourodé funkce nebo příliš mnoho funkcí.

U služeb IPF musí být XSD importováno z Oracle MDS (umístění začíná předponou *oramds*). Příklad importu XSD:

```
<xsd:schema>  
  <xsd:import namespace="http://vzp.cz/Finance/1"  
  schemaLocation="oramds:/apps/vzp.cz/Finance/1/Finance.xsd"/>  
</xsd:schema>
```

### 5.5.1.2 Pravidla pro XML

XSD a všechny datové položky jím definované musí být zahrnuty do příslušného namespace udaném v atributu „targetNamespace“.

Atribut *elementFormDefault* elementu *schema* je nastaven na hodnotu *qualified* – pro všechny elementy je určen jejich jmenný prostor (namespace).

XML elementy musí být v rámci XSD popsány ve formě komentářů. Výjimkou jsou jednoduché elementy, jejichž význam je jednoznačný z jejich názvu, např. *rodneCislo* nebo *rodnePrijmeni*.

Elementy musí mít určen konkrétní datový typ – např. výčty, délka textu, číselný rozsah apod. Nepoužíváme datové typy bez omezení (např. string).

#### 5.5.1.2.1 Jmenný prostor XML

Namespace XML zpráv využívaných službou je ve tvaru <http://vzp.cz/RozlisujiciOznaceni/NazevSluzby/HlavniVerze>, kde:

- *RozlisujiciOznaceni* je nepovinné, určuje aplikaci nebo doménu, ze které služba pochází, např. Finance
- *NazevSluzby*: udává samotný název služby, např. *IndividualniHodnotaBodu*
- *HlavniVerze*: číslo hlavní verze služby, tj. pro službu ve verzi 2.1 bude mít hlavní verze hodnotu 2

Příklad namespace: <http://vzp.cz/Finance/IndividualniHodnotaBodu/2>

#### 5.5.1.2.2 Pojmenování požadavku a odpovědi

Element požadavku služby je pojmenován stejně jako příslušná *operace*. Element s odpovědí služby má ve svém názvu navíc příponu *Odpoved*.

**Příklad:**

- Máme operaci *zalozDokument*.
- Element s požadavkem bude nazván: *zalozDokument*
- Element s odpovědí bude nazván: *zalozDokumentOdpoved*

### 5.5.1.2.3 Standardní integrační údaje

Standardní elementy XML zpráv usnadňují zpracování a provoz IPF služeb – nesou typické údaje jako určení konzumenta, referenci konzumenta, výsledek služby a další.

Standardní elementy XML zpráv jsou definovány ve jmenném prostoru *http://xmlns.vzp.cz/PovinneParametry/1*. Každá služba nesoucí XML data musí standardní elementy využívat. Standardní elementy jsou dostupné v rámci IPF na adrese *orams:/apps/xmlns.vzp.cz/PovinneParametry/1/PovinneParametry.xsd*.

Požadavek služby rozšiřuje element *parametryPozadavku*, viz následující příklad služby Monitoring IPF:

```
<xsd:element name="monitoringIPF">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="p1:parametryPozadavku"
xmlns:p1="http://xmlns.vzp.cz/PovinneParametry/1">
      <xsd:sequence>
        <xsd:element name="teloPozadavku" type="teloPozadavku">
          <xsd:annotation>
            <xsd:documentation>Tělo požadavku</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Obdobně element obsahující odpověď služby rozšiřuje element *parametryOdpovedi*:

```
<xsd:element name="monitoringIPFOdpoved">
  <xsd:complexType>
    <xsd:complexContent>
      <xsd:extension base="p1:parametryOdpovedi"
xmlns:p1="http://xmlns.vzp.cz/PovinneParametry/1">
      <xsd:sequence minOccurs="1">
        <xsd:element name="teloOdpovedi" type="teloOdpovedi"
minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>Tělo odpovědi</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
</xsd:element>
```

#### 5.5.1.2.3.1 Standardní elementy požadavku

Standardní elementy požadavku jsou:

Element	Účel
---------	------

<i>konzument</i>	Označení aplikace, která službu konzumuje, povinný údaj
<i>referenceKonzumenta</i>	Identifikátor přidělený konzumentem, je využíván pro rozlišení jednotlivých instancí služby, povinný údaj. Reference musí být unikátní v rámci hodnot používaných konzumentem.
<i>uzivatellid</i>	Identifikátor uživatele, který službu inicioval (např. spustil aplikační funkci, která konzumuje službu). Uživatel se neuvádí v případech, kdy službu konzumuje úloha na pozadí nebo jiný automatický proces. V případech, kdy lze jednoznačně určit zodpovědného uživatele, se tato položka vyplňuje.
<i>modul</i>	Upřesnění konzumenta (identifikace formuláře/sestavy/programového modulu apod.), nepovinná položka.
<i>duvodVolani</i>	Textový popis důvodu volání služby, slouží především pro ochranu osobních údajů.
<i>byznysId</i>	Identifikátor obchodního procesu, v rámci něhož je služba konzumována. Pokud je v rámci jednoho procesu konzumováno více služeb, měla by konzumující aplikace používat tentýž identifikátor.
<i>casPozadavku</i>	Čas odeslání požadavku.

**Tabulka 10 Standardní elementy požadavku**

#### 5.5.1.2.3.2 Standardní elementy odpovědi

Odpověď služby obsahuje stav zpracování služby:

<b>Element</b>	<b>Účel</b>
<i>konzument</i>	Stejná hodnota jako v požadavku.
<i>referenceKonzumenta</i>	Stejná hodnota jako v požadavku.
<i>byznysId</i>	Stejná hodnota jako v požadavku.
<i>casOdpovedi</i>	Čas odeslání odpovědi.
<i>pocetZaznamu</i>	Počet záznamů v odpovědi (používá se pouze u služeb, kde má takový údaj význam).
<i>stavVyrizeniPozadavku</i>	Typ Complex, který sdružuje:
<i>kod</i>	Kódové označení výsledku služby. Seznam kódů je standardizován v XSD a je rovněž uveden níže v 6.2 Ošetření nestandardních stavů.
<i>popis</i>	Doplňující popis výsledku služby, služba jej musí vyplnit, pokud je kód výsledku jiný než 1 (1: nalezen jeden záznam nebo obecná návratová hodnota OK).
<i>mistoChyby</i>	Udává komponentu, ve které došlo k případné chybě, např. IPF nebo RSZP

**Tabulka 11 Standardní elementy odpovědi**

#### 5.5.1.2.3.3 Zasílání nestrukturovaných dat v rámci XML

Pokud jsou v rámci XML zasílány dokumenty/soubory, jsou kódovány v Base64 a standardně v rozhraní uvedeny takto:



Element	Účel
<i>dokument</i>	typ Complex, který obsahuje
<i>nazev</i>	název souboru
<i>mime</i>	typ souboru
<i>obsah</i>	obsah souboru zakódovaný v Base64

**Tabulka 12 Zaslání nestrukturovaných dat v XML**

### 5.5.1.2.4 Struktura požadavku a odpovědi

XML požadavek obsahuje:

- Povinné elementy požadavku
- Tělo požadavku: XML elementy s obchodními daty služby

Příklad:

```
<p:zalozDokument xmlns:p="http://vzp.cz/Sluzba/1"
xmlns:p1="http://xmlns.vzp.cz/PovinneParametry/1">
  <p1:konzument>KONZ</p1:konzument>
  <p1:referenceKonzumenta>8796528</p1:referenceKonzumenta>
  <p1:uzivatelId>holyr19</p1:uzivatelId>
  <p1:modul>F-Registrace</p1:modul>
  <p1:duvodVolani>Kontrola vykázané péče</p1:duvodVolani>
  <p1:byznysId>BM86-4256</p1:byznysId>
  <p1:casPozadavku>2020-12-31T12:00:00</p1:casPozadavku>
  <p:teloPozadavku>
    XML struktura specifická pro službu
  </p:teloPozadavku>
</p:zalozDokument>
```

XML odpověď odpovídá formátu:

- Povinné elementy odpovědi
- Tělo odpovědi

Příklad:

```
<?xml version="1.0" encoding="UTF-8"?>
<p:zalozDokumentOdpoved xmlns:p="http://vzp.cz/Sluzba/1"
xmlns:p1="http://xmlns.vzp.cz/PovinneParametry/1">
  <p1:konzument>KONZ</p1:konzument>
  <p1:referenceKonzumenta>8796528</p1:referenceKonzumenta>
  <p1:byznysId>BM86-4256</p1:byznysId>
  <p1:casOdpovedi>2020-12-31T12:00:00</p1:casOdpovedi>
  <p1:pocetZaznamu>1</p1:pocetZaznamu>
  <p1:stavVyrizeniPozadavku>
    <p1:kod>1</p1:kod>
    <p1:popis>Nalezen 1 záznam</p1:popis>
  </p1:stavVyrizeniPozadavku>
  <p:teloOdpovedi>
    XML struktura specifická pro službu
  </p:teloOdpovedi>
</p:zalozDokumentOdpoved>
```

### 5.5.1.2.5 Pojmenování XML elementů a atributů

Elementy a atributy v rámci XML schémat pojmenováváme podle těchto pravidel:

- jednoslovné názvy: začínají malým písmenem, všechna písmena jsou malá (dle vzoru aaaa)
  - Příklad: *konzument*
- víceslovné názvy: dle vzoru aaaaBbbb, tj. druhé a následující slova začínají velkým písmenem
  - Příklad: *referenceKonzumenta*

Při návrhu XML schémat preferujeme vkládat obchodní (věcná) data do XML elementů a atributů využívat pouze pro pomocné údaje (metadata). Využití elementů zlepšuje čitelnost dokumentů a usnadňuje práci s nimi.

### 5.5.1.3 Operace služeb

Služba by měla sdružovat související funkce a vystavit je jako různé SOAP operace. Při návrhu operací je zapotřebí střídmost, abychom:

- nevytvářeli pro každou funkci novou službu (tj. každá služba by měla pouze jednu operaci)
- nevytvářeli službu s mnoha operacemi, které spolu přímo nesouvisí (návrhový anti-vzor *superslužba*)

Při návrhu operací zvažujeme kritéria:

- Všechny operace by měly náležet ke stejné funkci.
- Služba by měla typicky poskytnout všechny operace pro jedno datové aktivum nebo datovou entitu.

Hlavními kritérii při návrhu operací jsou tedy jednoduchost a srozumitelnost. Operace navrhujeme tak, aby budoucí konzumenti nemuseli vyhledávat různé související služby v Evidenci služeb, pokud bychom zvolili příliš drobnou granularitu, nebo naopak nemuseli číst příliš dlouhý popis *superslužby* a hledat v něm patřičnou operaci.

**Poznámka:** historicky vzniklo množství IPF služeb s jednou operací, jelikož BPEL komponenta IPF neměla vhodnou podporu pro více operací v rámci jedné služby. Od verze SOA Suite 11g je toto omezení odstraněno.

## 5.5.2 REST

Typický scénář využití REST na IPF:

- Aplikace má připravené REST API, které mají využívat další komponenty IS VZP.
- IPF vstupuje do role prostředníka integrací (funkce proxy), nemění chování API ani obsah zpráv.
- IPF provádí podpůrné funkce jako je standardní logování instancí služeb, zobrazení v Dohledu apod.
- Pouze v opodstatněných případech REST API na IPF vykonává vedlejší činnosti, jako například konzumace služeb jiných systémů.

Datové zprávy mají v REST rozhraních charakter zdroje (resource) a neměly by být obohacovány o integrační informace (konzument, reference konzumenta apod.) Pro integrační údaje jsou využívány HTTP hlavičky.

### 5.5.2.1 OpenAPI specifikace

REST rozhraní je popsáno ve formátu OpenAPI (preferována verze 3, popřípadě verze 2). OpenAPI specifikace musí být validní v nástrojích Swagger Editor<sup>1</sup> nebo SwaggerHub<sup>2</sup>.

V rámci OpenAPI specifikaci musí být v dostatečné míře detailu popsáno:

- Účel a využití API
- Pro všechny služby API:
  - Účel služby
  - Parametry služby a způsob jejich předání
  - Možné odpovědi služby, včetně chybových HTTP kódů a zpráv
- Pro všechny datové struktury (modely):
  - Popis datových typů včetně případných omezení
  - Příklady dat

Základním kritériem je, aby specifikace API byla bez dalších informací srozumitelná pro vývojáře konzumujících systémů a bylo zřejmé, jaká data služba očekává a vrací.

### 5.5.2.2 Standardní integrační údaje

Integrační údaje vkládáme do HTTP hlaviček požadavku, jedná se o:

Element	Účel
<i>konzument</i>	Označení aplikace, která službu konzumuje, povinný údaj.
<i>referenceKonzumenta</i>	Identifikátor přidělený konzumentem, je využíván pro rozlišení jednotlivých instancí služby, povinný údaj. Reference musí být unikátní v rámci hodnot používaných konzumentem.
<i>iv-user</i>	Identifikátor uživatele, který volání API spustil (např. spustil aplikační funkci, která konzumuje službu). Uživatel se neuvádí v případech, kdy službu konzumuje úloha na pozadí nebo jiný automatický proces. V případech, kdy lze jednoznačně určit zodpovědného uživatele, se tato položka vyplňuje.
<i>modul</i>	Upřesnění konzumenta (identifikace formuláře/sestavy/programového modulu apod.), nepovinná položka. Nepovinný parametr.
<i>duvodVolani</i>	Textový popis důvodu volání služby, slouží především pro ochranu osobních údajů. Nepovinný parametr.
<i>byznysId</i>	Identifikátor obchodního procesu, v rámci něhož je služba konzumována. Pokud je v rámci jednoho procesu konzumováno více služeb, měla by konzumující aplikace používat tentýž identifikátor. Nepovinný parametr.
<i>casPozadavku</i>	Čas odeslání požadavku. Nepovinný parametr.

**Tabulka 13 Parametry HTTP požadavku REST API vystaveného na IPF**

<sup>1</sup> Dostupný na adrese <https://editor.swagger.io/>

<sup>2</sup> Dostupný na adrese <https://swagger.io/tools/swaggerhub/>

IPF tyto údaje využívá pro své účely (logování apod.) a posílá je dále do cílového systému (pokud není dohodnuto mezi dodavatelem poskytujícího systému a dodavatelem služby IPF jinak).

### 5.5.2.3 Formát dat

V rámci VZP je preferován pro přenos strukturovaných informací formát JSON.

V případě rozsáhlých dokumentů, které je nutné validovat, může být využit formát XML (poskytuje lepší podporu pro definici a kontrolu dokumentů).

### 5.5.2.4 HATEOAS

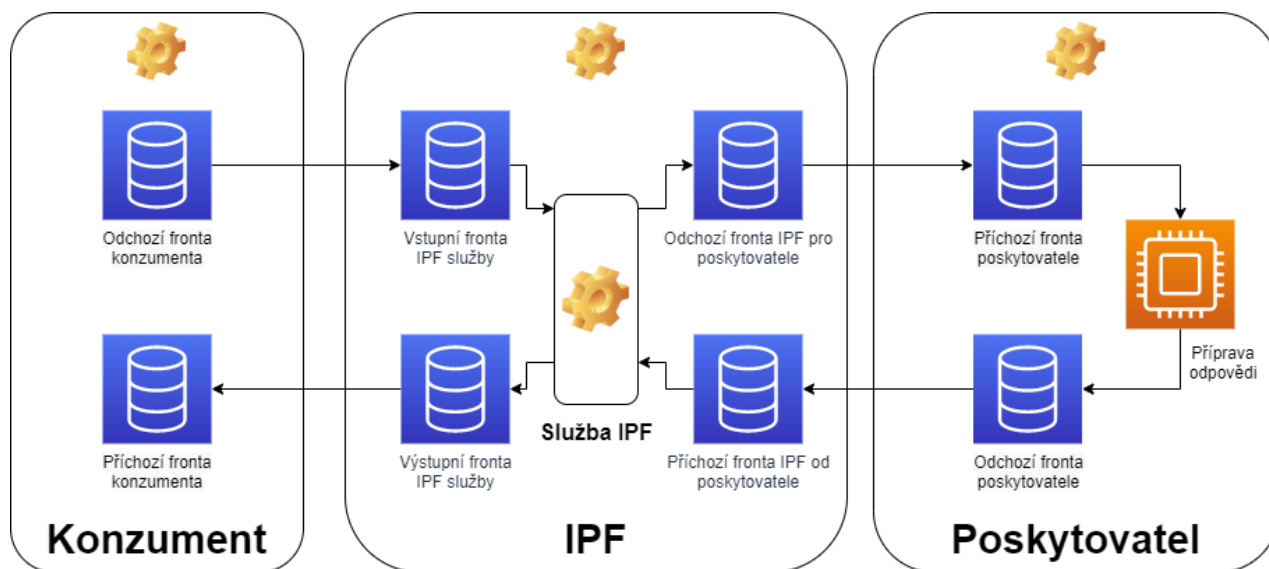
Je-li realizováno nové REST API, je upřednostňován formát HATEOAS (*Hypermedia as the Engine of Application State*), který pro získaný zdroj uvádí, jaké je s ním možné provádět operace ve formě odkazů.

## 5.5.3 Messaging

Komunikace prostřednictvím zpráv je ve stávajícím IS VZP realizována technologií Oracle Advanced Queuing (AQ). Zprávy jsou službami ukládány do AQ front, ze kterých jsou následně přebírány dalšími procesy ke zpracování nebo odesílány do jiných komponent (tzv. propagace zpráv).

Pokud připojovaný systém nepodporuje AQ fronty, ale jiný messaging např. JMS bude vnitřní rozhraní služby JMS a vnější asynchronní SOAP. Stávající systém v případě potřeby vystaví službu, kde vnitřní rozhraní zůstane AQ, ale vnější bude asynchronní SOAP.

Vložení zprávy je označováno jako enqueue, vyjmutí zprávy jako dequeue.



Obrázek 4 Schéma typické AQ služby

### 5.5.3.1 VZPIPFZPRAVA

V rámci AQ je přenášena standardizovaná VZPIPFZPRAVA (její SQL definice je uvedena v příloze 14.2 VZPIPFZPRAVA).

Samotná obchodní data jsou přenášena v elementech DataC (strukturovaná textová data) nebo DataB (binární data). Ostatní údaje slouží pro účely integrace.

<b>Atribut</b>	<b>Popis</b>
MsgID	ID zprávy přidělené odesílatelem (není to ID zprávy ve frontě AQ).
CorrID	Korelační ID: <ul style="list-style-type: none"> <li>• pokud odpovídám na zprávu, která měla vyplněný atribut CorrID, uvedu do něj ve své odpovědi totéž.</li> <li>• pokud přijde zpráva s prázdným CorrID, vyplním do CorrID své odpovědi hodnotu MsgID zprávy příchozí</li> </ul>
Puvodce	Označení aplikace, kde vznikla komunikace.
Odesilatel	Odesílatel této zprávy, např. <i>IPF</i> .
OdesilatelDoplněk	Upřesnění odesílatele, např. pokud existuje více instancí aplikace (jinak konstanta „9900“).
Prijemce	Adresát zprávy – aplikace, které je zpráva určena.
PrijemceDoplněk	Upřesnění adresáta zprávy.
Prostredi	Prostředí: <i>PROD</i> , <i>TEST</i> nebo <i>DEV</i> .
NazevSluzby	Název služby, která zprávu zpracovává, např. <i>StornoVyuctovani</i> .
NazevZpravy	Název zprávy, např. <i>stornoVyuctovani</i> , <i>stornoVyuctovaniAck</i> .
VerzeZpravy	Verze zprávy.
Reference	Obchodní/věcný identifikátor přidělený ke komunikaci.
ParametrV1	Doplňkový parametr typu VARCHAR2 k volnému použití, slouží pouze pro provozní či komunikační informace. Není určen pro obchodní data.
ParametrV2	Doplňkový parametr typu VARCHAR2 k volnému použití, slouží pouze pro provozní či komunikační informace. Není určen pro obchodní data.
ParametrD1	Doplňkový parametr typu DATE k volnému použití, slouží pouze pro provozní či komunikační informace. Není určen pro obchodní data.
ParametrD2	Doplňkový parametr typu DATE k volnému použití, slouží pouze pro provozní či komunikační informace. Není určen pro obchodní data.
ParametrN1	Doplňkový parametr typu NUMBER k volnému použití, slouží pouze pro provozní či komunikační informace. Není určen pro obchodní data.
ParametrN2	Doplňkový parametr typu NUMBER k volnému použití, slouží pouze pro provozní či komunikační informace. Není určen pro obchodní data.
<b>DataC</b>	<b>Obchodní data v textové formě (preferováno XML).</b>
<b>DataB</b>	<b>Obchodní data v binární formě (např. PDF).</b>

**Tabulka 14 Popis atributů VZPIPFZPRAVA**

U každé služby musí být v Evienci služeb popsáno, jaký obsah přenáší v elementech DataC a DataB. Strukturovaný obsah musí být popsán příslušnou specifikací (např. XML pomocí XSD).

Doporučení pro plnění parametrů objektu VZPIPFZPRAVA:

- Parametr *NazevZpravy* se shoduje s kořenovým elementem XML zprávy obsaženým v *DataC*
- Binární data primárně vkládat do položky DataB
- Atribut PUVODCE obsahuje hodnotu XML elementu *konzument*
- Atribut REFERENCE obsahuje hodnotu XML elementu *referenceKonzumenta*
- Při poskytování služby přes AQ musí být do všech odpovědních zpráv přenášeny hodnoty atributů „konzument“ a *referenceKonzumenta* z původního požadavku, který je vyřizován.

### 5.5.3.2 Pravidla pro přenášené XML dokumenty

Pro přenášené XML dokumenty platí stejná pravidla jako pro zprávy SOAP služeb (viz 5.5.1.2 Pravidla pro XML). Namísto SOAP obálky se používá VZPIPFZPRAVA a XML dokument vkládáme do elementu DataC.

### 5.5.3.3 Standardní integrační údaje

XML zprávy obsahují standardní elementy jako zprávy u SOAP komunikace (viz 5.5.1.2.3 Standardní ).

### 5.5.3.4 Pravidla pro pojmenování front

Abychom mohli AQ služby efektivně spravovat, jsou zavedena následující pravidla pro pojmenování AQ front.

Vzor názvu fronty obsahuje následující zkratky

- XX identifikuje dodavatele
- YYY je identifikátor systému

Účel fronty	Vzor názvu	Příklad
Příchozí ze systému na IPF	XXYYYY_Q	GMSAP_Q
Vstupní pro IPF službu (vnější rozhraní IPF služby)	XXIPFYYYYNA_Q	GMIPFFINNA_Q
Výstupní pro IPF službu (vnější rozhraní IPF služby)	XXIPFYYYYZ_Q	GMIPFFINZ_Q
Odchozí pro komunikaci s poskytovatelem (vnitřní rozhraní IPF služby)	XXYYYYNA_Q	GMFINNA_Q
Příchozí pro komunikaci s poskytovatelem (vnitřní rozhraní IPF služby)	XXYYYYZ_Q	GMFINZ_Q

**Tabulka 15 Pravidla pro vytváření a pojmenování AQ front**

### 5.5.4 Asynchronní SOAP služby

Pro asynchronní SOAP služby platí relevantní pravidla uvedená výše v 5.5.1 SOAP služby.

Konzument asynchronní služby vždy uvede adresu pro zpětné volání pomocí WS-Addressing elementu *ReplyTo*. Služba IPF na tuto adresu zašle asynchronní odpověď.

Příklad:

```
<wsa:ReplyTo>  
  <wsa:Address>http://komponenta.dc.vzp.cz/cesta</wsa:Address>  
</wsa:ReplyTo>
```

### 5.5.5 SDI

Silná datová integrace je přenos dat formou databázových tabulek a pohledů. Tabulka nebo pohled na straně poskytovatele dat je prostřednictvím IPF replikována do cílové aplikace.

Technologicky lze SDI vazby realizovat pouze mezi databázemi Oracle DB.

SDI vazby jsou určeny pouze pro specifické případy, kdy nelze potřebu informací uspokojit SOAP nebo AQ službami. SDI je realizováno pouze pokud jsou naplněny podmínky:

- Konzumující komponenta vyžaduje kompletní data, která mají primární zdroj v jiné komponentě.
- Konzumující komponenta vyžaduje okamžitou synchronizaci ze zdrojové komponenty.

#### 5.5.5.1 Specifikace SDI

V návrhu pro novou SDI integrační vazbu musí být určeno:

- Komponenta, která je primárním zdrojem dat (např. RSZP)
- Výčet atributů, které jsou požadovány v konzumující komponentě

## 5.6 Autentizace a autorizace

Konzumenti služeb IPF mohou být identifikováni dvěma způsoby:

- Službu konzumuje komponenta
- Službu konzumuje komponenta s uvedením uživatele (skutečné osoby)

Identifikace komponenty a konzumenta je připravena pro autentizaci na straně poskytujícího systému, případně pro budoucí zavedení autentizačních a autorizačních mechanismů IPF. Aktuálně IPF autentizaci a autorizaci neprovádí.

Pro identifikaci uživatele se používá uživatelské jméno osoby v rámci IS VZP, např. melich99.

Poznámka:

*V rámci modernizace IPF bude přidáno ověřování certifikátu konzumenta při konzumaci služby IPF. Pro platné certifikáty VZP bude povolena konzumace služby IPF.*

### 5.6.1 SOAP služby

Pro identifikaci konzumenta využijeme standardní povinné elementy (viz 5.5.1.2.3 Standardní):

- *konzument*: identifikace konzumující komponenty
- *uzivateld*: identifikace uživatele

## 5.6.2 REST API

Pro REST API se údaje vkládají do HTTP hlaviček (headers) požadavku:

- *konzument*: identifikace konzumující komponenty
- *iv-user*: identifikace uživatele (dle standardu NIS)

Služba IPF zkopíruje tyto hlavičky z přijatého požadavku do požadavku pro cílovou komponentu.

## 5.7 Zabezpečení služeb

Služby IPF jsou aktuálně zabezpečeny pouze na síťové úrovni (síťový přístup ke službám).

Poznámka:

*V rámci modernizace IPF je plánováno zabezpečení na transportní úrovni pomocí TLS.*

## 5.8 Objem přenášených dat

Služby by měly být navrhovány takovým způsobem, aby nepřenášely příliš objemná data.

Při přenosu mnoha záznamů je vhodné zapojit princip stránkování, kdy služba vrátí pouze část záznamů a konzument další části odpovědi získá opakovanou konzumací služby.

Druh komunikace	Typická velikost	Limit	Popis
SOAP	do 200 kB	2 MB	Přenos velkých zpráv trvá příliš dlouho pro interaktivní scénáře. Velké XML dokumenty zatěžují aplikace (paměť, procesor), které je zpracovávají.
REST	do 200 kB	2 MB pro strukturovaná data	Důvod limitu pro strukturovaná data (JSON, XML) je stejný jako pro SOAP.
AQ	do 2 MB	10 MB	Přenos dat běží asynchronně, lze tedy přenášet větší data.
Binární data	do 10 MB	20 MB	V rámci výše uvedených protokolů lze přenášet také binární data (např. v rámci REST API nebo jako datový obsah AQ komunikace).  Limit pro binární data by měl podporovat běžné typy přenášených souborů (dokumenty, obrázky apod.)

**Tabulka 16 Limity objemů dat dle druhu komunikace**

Při návrhu služby by měla být respektována výše uvedená *typická velikost*. Přenášená data mohou být v krajních případech větší, jejich objem by však neměl překročit hodnotu *limit*.

Větší data (souborový nebo ETL přenos) by měla být přenášena mimo IPF. Popis souborového a ETL přenosu není součástí této metodiky.

## 5.9 Dokumentace v Evidenci služeb

Služba musí být již ve fázi návrhu zapsána do Evidenci služeb, včetně:



- Základních parametrů služby (název, režim, apod.)
- Popisu služby (obchodní, technický)
- Specifikace rozhraní
- Relevantní příložené dokumenty

Dokumentaci služeb podrobněji popisuje Metodika dokumentace služeb IPF (viz 14.1 Odkazy).

## 5.10 Testovací plán

Při tvorbě nové služby nebo v případě významné změny služby (viz výše 4.3.3 Změna existující služby IPF) dodavatel služby IPF rovněž dodá testovací plán.

Testovací plán bude obsahovat seznam testů, jimiž bude změna ověřena.

Pro každý test bude uvedeno ve stručnosti:

- funkční popis testu (jaké funkce testujeme)
- služba a operace, která bude konzumována
- popis vstupních dat
- očekávaný výsledek a očekávaný obsah odpovědi

Testovací plán má pokrýt všechny nové službu a operace a rovněž významné změny služeb.

Testovací plán pokrývá všechny scénáře využití služby, ke kterým může v běžném provozu dojít (data nalezena, nenalezena, apod.).

## 5.11 Homologace návrhu

V okamžiku, kdy je připraven návrh nové služby resp. významné změny služby, je předán k homologaci. Cílem homologace návrhu je ověřit, že uvedený návrh odpovídá integračním standardům IS VZP.

Jelikož se návrh služby IPF odvíjí od návrhu cílové služby, je zapotřebí zajistit, aby i **cílová služba odpovídala integračním standardům VZP**.

Akce	Obsah	Zodpovědnost
Předání návrhu k homologaci	<ul style="list-style-type: none"> <li>• Popis služby v rámci Evidence služeb (služba je ve stavu navrhovaná), včetně specifikace rozhraní</li> <li>• Testovací plán</li> </ul>	Dodavatel služby IPF
První kolo homologace	<ul style="list-style-type: none"> <li>• Schválení v případě, že je návrh vyhovující</li> <li>• Seznam připomínek v případě, že návrh nevyhovuje integračním standardům</li> </ul>	ORI VZP
Druhé kolo homologace	<ul style="list-style-type: none"> <li>• Totožné jako v prvním kole homologace.</li> </ul>	PTP IPF

**Tabulka 17 Předání návrhu služby IPF k homologaci**

Homologaci návrhu podrobněji popisuje Metodika homologace služeb IPF (viz 14.1 Odkazy).

## 6 Implementace

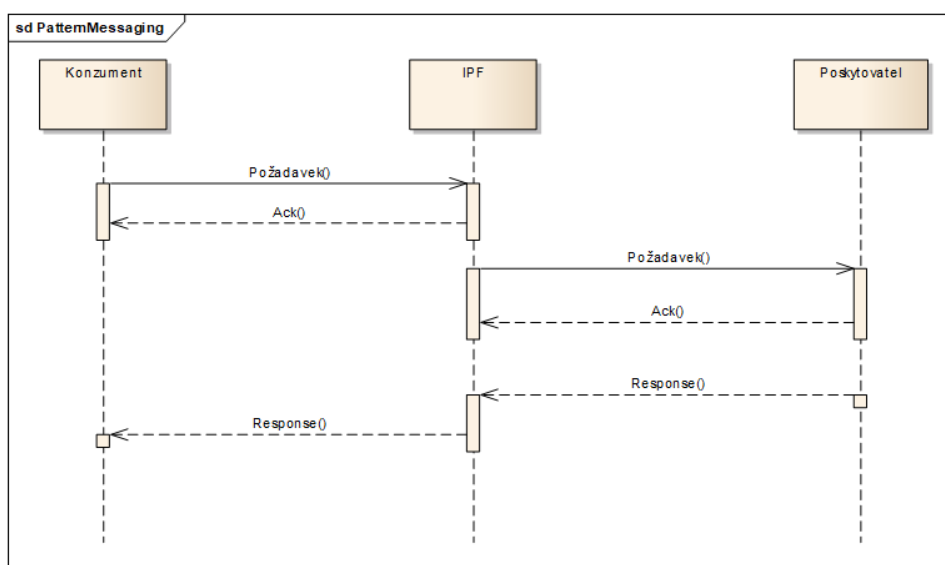
Tato kapitola uvádí pravidla pro implementaci služeb IPF. Kompozitní, asynchronní služby nebo služby, které jsou atomické, ale lze u nich v budoucnu předpokládat přidání programové logiky, realizujeme jako BPEL služby v komponentě SOA.

Jednoduché atomické služby jsou realizovány v komponentě SOA nebo OSB. SOA poskytuje větší možnosti trasování služeb (zobrazení instance služby, identifikace místa, kde došlo k chybě apod.) a také umožňuje důslednější oddělení jednotlivých složek (řízení přístupu na úrovni složky, řízení výkonu na úrovni složky).

### 6.1 Messaging

Asynchronní komunikace pracuje kromě obchodních zpráv nesoucích požadavky a odpovědi také s pomocnými zprávami Ack a Log. Názvy zpráv jsou odvozeny od názvu služby takto:

- Požadavek na službu má název shodný jako kořenový element XML Dokumentu reprezentujícího požadavek, např. *dokladyVyuctovani*
- Potvrzení přijetí požadavku je odvozeno od názvu požadavku a má příponu "Ack" (např. *dokladyVyuctovaniAck*); jeho odesláním dává poskytovatel najevo konzumující straně, že má smysl očekávat zprávu s výsledkem vyřizování požadavku. Jedná se o provozní zprávu, která nenese žádná obchodní data.
- Odmítnutí přijetí požadavku je odvozeno od názvu požadavku a má příponu "Log" (např. *dokladyVyuctovaniLog*); jeho odesláním dává poskytovatel najevo konzumující straně, že již nebude odeslána žádná další zpráva vztahující se k tomuto požadavku. Jedná se o provozní zprávu, která nenese žádná obchodní data.
- Odpověď má název shodný jako kořenový element XML Dokumentu reprezentujícího odpověď (např. *dokladyVyuctovaniOdpoved*). Poskytovatel služby může poslat libovolný počet takto pojmenovaných zpráv v závislosti na charakteru poskytované služby (u poloautomatického zpracování požadavků je těmito zprávami konzumující strana informována o průběhu zpracování, který závisí na aktivitě uživatelů). Indikace poslední zprávy vztahující se k danému požadavku, je uváděna jako součást obchodních dat.



Obrázek 5 AQ Komunikace – Pozitivní scénář

O korektní přenos zpráv mezi frontami procesovými a propagačními se na IPF stará Směrovací aparát popsáný v dokumentu “Integrační platforma: Programová dokumentace/Příručka administrátora”.

Komponenty mimo Integrační platformu mohou pro odstínění od mechanismů AQ komunikace využít modul Komunikační framework, kterému se věnuje samostatný dokument “Komunikační framework”.

## 6.2 Ošetření nestandardních stavů

### 6.2.1 SOAP a AQ

Chyby nejsou konzumujícím systémům předávány prostřednictvím SOAP Fault. Služby odchycují výjimky a transformují je do příslušného chybového kódu předávanému prostřednictvím elementu *stavVyrizeniPozadavku/kod*.

Množina návratových kódů je definována následovně:

- 0 – data nenalezena
- 1 – nalezen jeden záznam nebo obecná návratová hodnota OK
- 2 – částečně zpracováno (část požadavku zpracována v pořádku, část odmítnuta)
- 9 – nalezeno více záznamů
- -1 – technická chyba
- -2 – chyba při validaci vstupních dat
- -3 – chyba při validaci odpovědi

Je-li chyba odchycena či nalezena na v rámci zpracování na IPF, služba doplní do odpovědi element *stavVyrizeniPozadavku/mistoChyby* s hodnotou „IPF“.

### 6.2.2 REST

REST služby IPF předávají odpovědi z poskytující komponenty (včetně HTTP status kódu nebo výjimek).

Pokud je nestandardní stav rozpoznán na IPF (např. nevalidní požadavek nebo neplatná URL zdroje), reaguje služba podle pravidel REST příslušným HTTP kódem (viz následující tabulka).

HTTP kód	Popis
400 BAD REQUEST	Nevalidní požadavek, URL nebo HTTP hlavička.
401 UNAUTHORIZED	Neúspěšná autorizace. Týká se pouze služeb IPF, které budou autorizaci zajišťovat (v běžných případech by ji měla zajistit cílová služba).
403 FORBIDDEN	Nepodporovaný standardní parametr požadavku (např. HTTP metoda).
404 NOT FOUND	Požadovaný zdroj nenalezen.
500 INTERNAL SERVER ERROR	Interní chyba služby IPF.
503 SERVICE UNAVAILABLE	Služba nedostupná. Chybový kód nevrací služba IPF (je nedostupná), ale platforma IPF.

## Tabulka 18 HTTP kódy pro nestandardní stavy REST služeb IPF

### 6.3 MDS

MDS (Oracle Metadata Service) slouží jako vysoce dostupné úložiště pro metadata, která jsou následně k dispozici všem částem Oracle SOA Suite. Jako metadata lze ukládat XML soubory, XSD schémata, XSL transformace, WSDL soubory, BPEL soubory a další artefakty využívané službami IPF.

V případě IPF je MDS primárně používána k ukládání souborů XSD, aby nebylo nutné u jednotlivých kompozitních aplikací držet tyto soubory lokálně. Stejně tak jsou zde pro závislé služby k dispozici abstraktní WSDL zdrojových služeb.

Správa MDS je prováděna prostřednictvím skriptů dodávaných v rámci instalačních balíčků.

#### 6.3.1 Úložiště XSD

XSD se ukládají do stromové struktury sestavené dle jmenného prostoru (namespace) služby, která je vložena do adresáře „apps“.

Například pro „ZakladniRegistry.xsd“, jehož jmenný prostor je:

- `http://xmlns.gemsystem.cz/ZakladniRegistry/1`

je cesta v rámci MDS následující.

- `oramds:/apps/xmlns.gemsystem.cz/ZakladniRegistry/1/ZakladniRegistry.xsd`

Odpovídající adresářová struktura (při uložení na souborový systém, např. v GIT) je:

- apps
  - xmlns.gemsystem.cz
    - ZakladniRegistry
      - 1

V posledním podadresáři „1“ je uložen soubor *ZakladniRegistry.xsd*.

#### 6.3.2 Úložiště WSDL

MDS rovněž poskytuje abstraktní WSDL služeb IPF. Abstraktní WSDL je definice rozhraní služby specifikující datové typy, zprávy a operace, ale nemá určen binding a endpoint.

Struktura je „oramds:/apps/abstract/folder/sluzba.wsdl“, kde „folder“ je název složky, ve které je služba provozována, a „sluzba“ je název služby.

Například:

- `oramds:/apps/abstract/Z0-Pojistenci-a/Oznamenizamestnavatele.wsdl`

Při potřebě konzumace IPF služby jinou IPF službou, pracuje konzumující služba pouze s tzv. „Wrapper WSDL“. Jedná se o rozšíření abstraktního WSDL, které je do Wrapper WSDL importováno z MDS, o binding a endpoint.

### 6.4 BPEL

Nové služby jsou implementovány v BPEL 2.0. Stávající služby mohou zůstat v BPEL 1.1, v případě větších změn je doporučen převod do BPEL 2.0.

Služba může obsahovat jednu nebo více souvisejících operací. Každá operace má vlastní BPEL proces. Mapování operací zajistí komponenta Mediator.

Jméno BPEL procesu je ve tvaru *NazevOperace* s velkým písmenem na začátku.

Všechny aktivity v rámci BPEL procesu jsou srozumitelně pojmenovány v českém jazyce s malým počátečním písmenem. Následující slova v názvu jsou psána s velkým počátečním písmenem (tzv. velbloudí písmo).

Pro pojmenování proměnných platí podobná pravidla jako pro aktivity. Volíme název tak, aby vypovídal o jejich použití.

- Vstupní a výstupní proměnné aktivity *Invoke* mají v názvu operaci či systém, se kterým komunikují, a příponu *Input/Output* udávající směr komunikace.
- Pro vstupní a výstupní proměnné ponecháme implicitní název (*inputVariable*, *outputVariable*).

### 6.4.1 Standardní kroky BPEL procesů

BPEL proces musí nastavit flow a kompozitní titulek ve tvaru "operace:konzument:referenceKonzumenta".

- Nastavení flow titulku se provádí pouze, je-li prázdný.
- Titulky mohou být rozšířeny o další reference či informace usnadňující orientaci mezi jednotlivými instancemi služby.
- Ořezávat titulek na maximálně 100 znaků.

Služby validují požadavky a odpovědi.

V procesu nastavujeme kompozitní senzory pro elementy *konzument* a *referenceKonzumenta*.

Výjimky jsou ošetřeny v *CatchAll* bloku, kde dojde k nastavení chybové odpovědi.

### 6.4.2 Parametry BPEL procesů

Služby pracují s následujícími parametry:

Parametr	Význam
preference.Prostredi	Nabývá hodnot DEV/TEST/PROD/SIMU
bpel.config.inMemoryOptimization	Konfigurace hydratace a ukládání instancí synchronních služeb
bpel.config.completionPersistPolicy	
preference.AckTimeout	Timeout potvrzení AQ zpráv
preference.RetryTime	Čas, po kterém dojde k opakování volání
preference.RspRetryTime	Čas, po kterém dojde k opakování odeslání odpovědi.

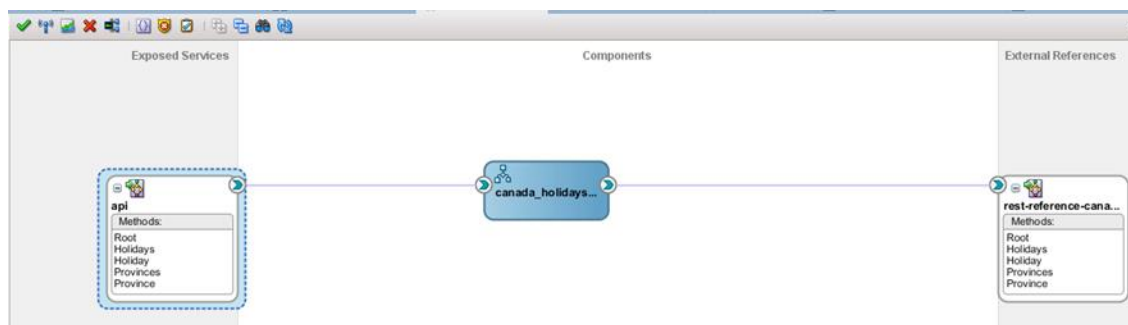
Tabulka 19 Parametry BPEL procesů

### 6.4.3 REST služby

Implementace REST služeb na IPF se provádí prostřednictvím kompozitní aplikace podobně jako u SOAP služby. Rozdílem oproti SOAP je využití komponenty *REST Binding*. Pomocí této komponenty se vystaví jak služba, kterou poskytuje IPF, tak i reference na REST API poskytovacího systému.

*REST Binding* komponenta umožňuje definovat rozhraní služby manuálně (definice jednotlivých resource, method, atd.) nebo importem souboru s definicí *WADL*. K propojení

reference a vystavené cílové služby je možné využít standardních komponent pro kompozitní aplikace, jako je třeba BPEL process na uvedeném příkladě (viz obrázek).



**Obrázek 5** Příklad realizace REST služby

## 6.5 Oracle Service Bus

OSB neumožňuje vytvářet složky jako komponenta SOA. Pro zvýšení přehlednosti jsou služby organizovány do složek pomocí jmenné konvence. Služby na OSB obsahují ve svém názvu složku, do níž patří.

Příklad názvu služby: *B0-Techws.MonitoringOSB*

- *B0-Techws*: označení složky
- *MonitoringOSB*: označení samotné služby

Pro každou operaci služby je vytvořen samostatný proces zpracování (tzv. pipeline).

Vícenásobné přiřazování hodnot (např. naplnění povinných elementů v XML požadavku) provádíme pro větší přehlednost pomocí XQuery šablon.

Pro uspořádání zdrojového kódu služby používáme následující adresáře:

Adresář	Popis
jar	Knihovny použité v rámci OSB služby (jar soubory).
resources	Artefakty použité při vývoji služby, např. XSD, která nenní vhodné sdílet v MDS.
transformations	Všechny transformace ukládáme do adresáře <i>transformations</i> . Jednotlivé transformace pojmenujeme názvem, který jednoznačně určuje využití transformace, např. NTL_ZalozTransakciLogu.xqy.

**Tabulka 20** Standardní adresáře ve zdrojovém kódu OSB služby

Cílovou službu konzumujeme v rámci OSB služby pomocí aktivity *Routing*. Ostatní pomocné služby (zápis do NTL apod.) realizujeme pomocí:

- *Service Callout*: umožňuje zjistit výsledek volání a vyhodnotit jej
- *Publish*: spuštění procedury, neumožňuje získat výsledek operace

## 6.6 SDI

Integrační vazby SDI jsou implementovány na úrovni Oracle DB a zajišťují přenos mezi zdrojovou a cílovou databází.

Silné datové integraci je věnován samostatný dokument pojmenovaný “Silná datová integrace”.

## 6.7 Validace

Služby validují vstupní i výstupní zprávy. V případě validační chyby služba vrací konzumentovi odpověď s příslušným chybovým kódem (-2 pro nevalidní požadavek, -3 pro chybu validace odpovědi) a popisem odpovídajícím nalezené nesrovnalosti.

Kromě kontroly, zda zprávy odpovídají předepsané definici (XSD v případě XML zpráv), mohou služby volitelně provádět také validaci obchodních dat.

AQ služby validují požadavky/odpovědi přenášení v atributu VZPIPFZPRAVA.DATAC.

Poznámka: Většina historických služeb zprávy nevaliduje a integrované systémy s tím počítají.

Pokud služba nevaliduje požadavek nebo odpověď, je zapotřebí na to upozornit v technickém popisu služby.

## 6.8 Šifrování údajů

Pro citlivé údaje, ke kterým musí být odepřen přístup v rámci celé integrační vazby, je nutné využívat koncové šifrování pomocí PKI infrastruktury a standardu WS-Security/XML Encryption.

Pokud je dostatečné zašifrovat přístup k citlivým údajům v rámci IPF, lze využít následující postup.

### 6.8.1 Šifrování XML elementů v rámci IPF

SOA Suite 12c umožňuje zašifrovat určené elementy XML zpráv v rámci IPF. K takto zašifrovaným elementům pak nelze přistupovat v rámci nástrojů SOA Suite (Enterprise Manager). Při přijetí zprávy na IPF je citlivý obsah zašifrován a při odeslání zprávy vrácen do původního stavu.

Pro službu je definována politika ochrany citlivých údajů a XPath výraz, který určuje cestu k chráněnému XML elementu.

V rámci nástrojů SOA Suite je pak namísto citlivého údaje zobrazen text ve formátu:

```
encrypted:<CIPHER_TEXT>;keyname:<CSF_KEY_NAME>
```

Zde je potřeba mít na paměti, že XML obsahující šifrovaný text již nemusí být validní vůči XSD služby. Tento způsob šifrování je tedy zapotřebí používat uvážlivě.

Podrobný popis nastavení politiky je popsán v dokumentaci SOA Suite, viz odkaz Popis šifrování XML elementů v 14.1 Odkazy.

## 6.9 Konfigurace služeb

### 6.9.1 BPEL

Při nasazení kompozitních aplikací využíváme konfigurační plány, které v závislosti na prostředí.

- Náhradu URL endpointu serveru ve WSDL volaných služeb

- Nastavení parametrů služeb (kapitola 6.4.2)
- Nastavení názvů AQ front B2B služeb

Aby bylo použití konfiguračních plánů nezávislé na prostředí, používají se v konfiguračních plánech parametry, které jsou před instalací služby nahrazovány na základě souboru preferencí. Tento soubor je uložený lokálně na aplikačním serveru, na kterém probíhá instalace služeb, což umožňuje používat různé nastavení preferencí pro různá prostředí.

Úkázka části konfiguračního plánu nahrazující endpoint zdrojové služby:

```
<wsdlAndSchema name="IndividualniHodnotaBodu-FIN.wsdl">
  <searchReplace>
    <search>http://finws.dc.vzp.cz:8888</search>
    <replace>###url.fin###</replace>
  </searchReplace>
</wsdlAndSchema>
```

Příklad záznamu souboru preferencí obsahující endpoint zdrojové aplikace v testovacím prostředí:

```
\###url.fin###=http://tfin.dc.vzp.cz:8888
```

Seznam dostupných parametrů:

Parametr	Popis
Prostredi	Název prostředí, do něž je služba instalována (DEV, TEST, PROD, SIMU).
AckTimeout	Čas, po který služba po příjmu odpovědi čeká na obdržení zprávy Ack.
Timeout	Čas, po který služba čeká na obdržení odpovědi.
PocetPokusu	Maximální počet pokusů nějaké opakované činnosti (např. počet pokusů opakování odeslání odpovědi).
RetryTime	Čas, po kterém dojde k dalšímu pokusu vykonání aktivity, která selhala.
RspRetryTime	Čas, po kterém dojde k dalšímu pokusu odeslání odpovědi, které selhalo.
B2BTimeout	Čas, po který služba čeká na odpověď z B2B.
inMemoryOptimization	Zapíná možnost optimalizace tím, že nebudou instance služby ukládány do databáze (True, False).
completionPersistPolicy	Definuje, které instance používající inMemoryOptimization budou ukládány do databáze (On, Faulted, Deferred, Off).
syncAuditLevel	Nastavuje úroveň auditu synchronních služeb, u nichž je umožněno vypnutí auditování (Off, Production, Development, Inherit).
url.ipf	Adresa a číslo portu aplikačního serveru SOA Suite 12c
url.rszp	Adresa a číslo portu aplikačního serveru aplikace RSZP
url.rszp.zrk	Adresa a číslo portu aplikace ZR Konektor
url.idm	Adresa a číslo portu aplikačního serveru aplikace ID
url.vpproxy	Adresa a číslo portu komponenty VPProxy



url.sova	Adresa a číslo portu aplikačního serveru aplikace SOA
url.esss	Endpoint služby ESSS
url.mx	Adresa MX
url.crp	Adresa a číslo portu aplikačního serveru aplikace CRP
url.bambi	Adresa a číslo portu aplikačního serveru aplikace BAM BI
url.fin	Adresa a číslo portu aplikačního serveru aplikace Finance
url.uvv	Adresa a číslo portu aplikačního serveru aplikace UVV
url.optiarchiv	Adresa a číslo portu aplikace OptiArchiv
url.adb	Adresa a číslo portu aplikačního serveru komponenty ADB
url.cvon	Adresa a číslo portu aplikačního serveru aplikace CVC online
url.cvoff	Adresa a číslo portu aplikačního serveru aplikace CVC offline
url.kc	Adresa a číslo portu aplikačního serveru aplikace Kapitační centrum
url.evpr	Adresa a číslo portu aplikačního serveru aplikace EVPR
url.fama	Adresa a číslo portu aplikačního serveru aplikace FaMa
aq.b2bna	Název AQ fronty pro odesílání zpráv do B2B
aq.b2bz	Název AQ fronty pro příjem zpráv z B2B
aq.b2b.ipfna	Název AQ fronty, kterou technologický adaptér přijímá odpovědi podkladové služby
aq.b2b.ipfz	Název AQ fronty, kterou technologický adaptér posílá požadavky podkladové službě
url.sap.spf	Endpoint SAP služby SchvaleniProvozniFaktury
url.sap.ksm	Endpoint SAP služby KatalogSkupinMaterialu
url.sap.kzm	Endpoint SAP služby KmenoveZaznamyMaterialu
url.sap.oup	Endpoint SAP služby OtvorenePolozkyUcetnictvi
url.sap.ud	Endpoint SAP služby UcetniDoklad
url.sap.nz	Endpoint SAP služby NotifikaceZmen

**Tabulka 21 - Konfigurační parametry služeb**

Kromě parametrů v tabulce existují také parametry individuální pro jednotlivé služby. Jejich kompletní výčet je k dispozici v dokumentu „Integrační platforma: Programová dokumentace/Příručka administrátora”.

## 6.9.2 OSB

Konfiguraci pro různá prostředí určujeme pomocí konfiguračních plánů, analogicky jako je tomu u BPEL služeb.

V konfiguračních plánech využíváme stejné parametry jako u BPEL, viz výše.

## 6.9.3 Doplnění konfiguračních plánů

Konfigurační plány jsou postupně doplňovány o nové parametry:

- pro nové komponenty nasazené do IS VZP
- při takové změně stávající komponenty, která přinese potřebu zavést nový parametr (vystavení nové URL, nová AQ fronta apod.).

Konfigurační plán je společný pro všechny služby daného prostředí. Dodavatel služby IPF požádá o doplnění konfiguračního plánu Administrátora IPF a s požadavkem předá:

- návrh názvu nového parametru
- jeho hodnoty pro všechna relevantní prostředí.

## 6.10 Logování

Každá služba IPF zapisuje o svém běhu do NTL (NIS Transakční Log). Služby IPF pro zápis do NTL využívají PLSQL API.

NTL obsahuje následující druhy údajů:

- **Transakce:** odpovídá instanci služby IPF, obsahuje základní údaje instance služby. Každá služba má vytvořit jeden záznam typu Transakce. Poznámka: transakce je míněna v obchodním smyslu, tj. množina procesů, která má nějaký obchodní účel. Transakcí v kontextu NTL nemíníme databázovou ACID transakci.
- **Událost:** událost při běhu služby, kterou dává smysl logovat. Příklady událostí: odeslání požadavku a přijetí odpovědi u asynchronních služeb, konzumace jiné služby u kompozitních služeb, chybové odpovědi služeb. Pro událost je v opodstatněných případech možné uložit také přenášená data, typicky celý datový obsah požadavku či odpovědi.
- **Reference:** přenášená data mohou obsahovat řetězce klíč-hodnota, které jsou následně využívány při vyhledávání instancí služeb IPF. Příkladem může být číslo jednacích, číslo pojišťovny apod. Zapisování referencí je volitelné, služba může zapisovat libovolný počet párů klíč-hodnota. Příklad: *CisloPojistence, 123456789*.

Vždy je nutné zvážit, jaké informace má smysl do NTL zapisovat. Každý zápis do NTL prodlužuje zpracování služby, neměli bychom tedy zapisovat údaje neužitečné.

Na druhé straně vhodným zápisem událostí můžeme zjednodušit podporu služeb a zpřehlednit integrační vazby. Reference zapisujeme tehdy, pokud víme, že podle určených údajů bude zapotřebí vyhledávat.

Následující tabulka uvádí, jaké údaje vkládat do atributů transakce.

Atribut	Význam	Příklady
CREATED_TIME	Datum spuštění služby IPF. Vypĺňováno interně při volání API NTL.	
FINISHED_TIME	Datum ukončení běhu služby IPF. Vypĺňováno interně při volání API NTL.	
ORIGIN_FQDN	Plně kvalifikované doménové jméno uzlu IPF, na němž běží služba.	<i>soa1.dc.vzp.cz</i>
ORIGIN_IP	IP adresa uzlu IPF, na němž běží služba.	<i>10.0.4.35</i>
APPLICATION	Označení aplikace, pro nás vždy "IPF".	<i>IPF</i>

APP_INTERFACE	Druh rozhraní aplikace, pro IPF není nutné vyplňovat.	REST, SOAP, AQ
SERVICE_ID	Název služby IPF	PrehledUhradZP, DetailSubjektuZP
INSTANCE_ID	Identifikátor instance služby v rámci integrační platformy IPF (pro SOA flow ID).	9856175
COM_PARTNER	Konzument služby	RSZP, FIN, CVC
TRANSACTION_ID	Business identifikátor transakce	M65824
PARTNER_ID	Reference konzumenta	RB5893845
SRC_USER	Uživatelské jméno uživatele, který způsobil volání služby (pokud je relevantní).	novop19
RESULT_CODE	Výsledek služby IPF, hodnota ze stavVyrizeniPozadavku.kod.	1
RESULT_TEXT	Popis výsledku služby, bývá vyplněn v případě nestandardního výsledku nebo chyby.  Hodnota ze stavVyrizeniPozadavku.popis.	Pojištěnec s danými údaji nenalezen.
DOMAIN	Složka (folder) služby IPF	X0-Pojistenec-s
MODULE	Modul IPF: "OSB" nebo "SOA", dle komponenty, na které běží služba.	SOA
OPERATION	Operace služby, která byla spuštěna.	vyhledejPojistence
VERSION	Verze služby.	2.0
COMPOSITE	Název kompozitní SOA aplikace, v rámci níž je služba nasazena.	ISZRSlužby

**Tabulka 22 NTL - Plnění atributů transakce (transaction) NTL**

Služby IPF plní atributy událostí následovně.

Atribut	Význam	Příklady
CREATED_TIME	Datum události	
NAME	Název události	OdeslaniPozadavku
DESCRIPTION	Popis události.	Odeslání požadavku na službu ABC.
PAYLOAD	Obsah zprávy související s událostí. Typicky požadavek nebo odpověď služby. Plnění je volitelné, je však třeba dbát zásad ochrany osobních údajů.	XML zpráva požadavku.
STATE	Výsledek události, analogicky k výsledku transakce.	1

**Tabulka 23 Plnění atributů události (event) NTL**

Pro reference do NTL vkládáme páry klíč-hodnota.

Atribut	Význam	Příklady
CREATED_TIME	Datum vložení reference	
NAME	Název reference	<i>CisloPojistence</i>
VALUE	Hodnota reference	<i>12345678</i>

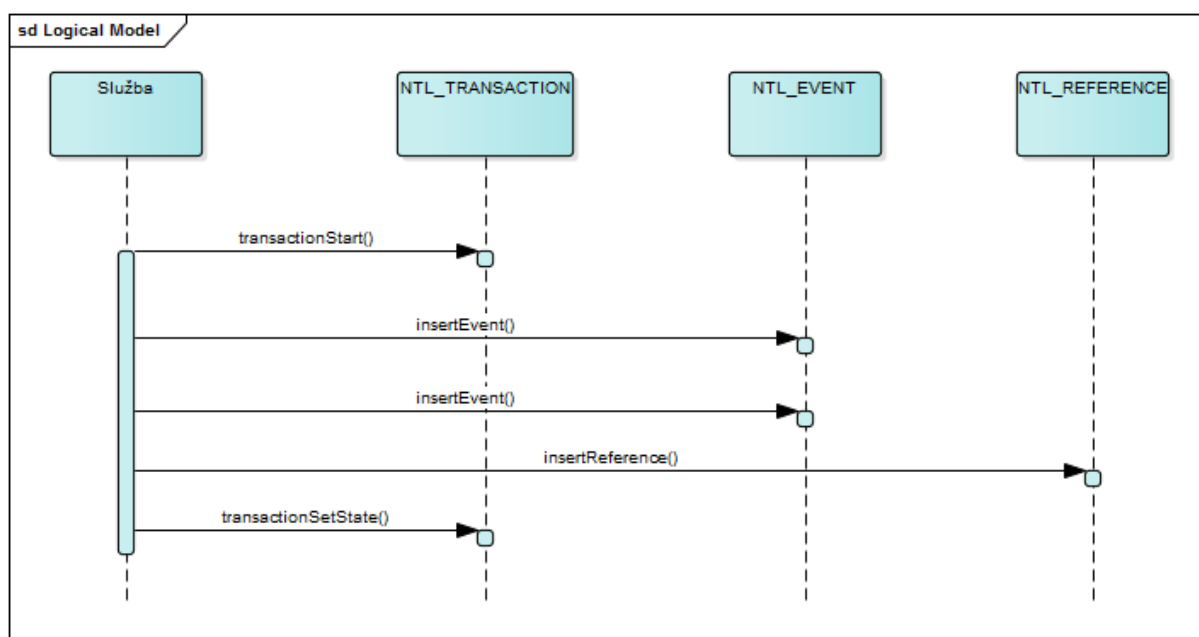
**Tabulka 24 Plnění atributů reference NTL**

### 6.10.1 Využití API NTL ve službách IPF

Služby IPF využívají následující funkce NTL API:

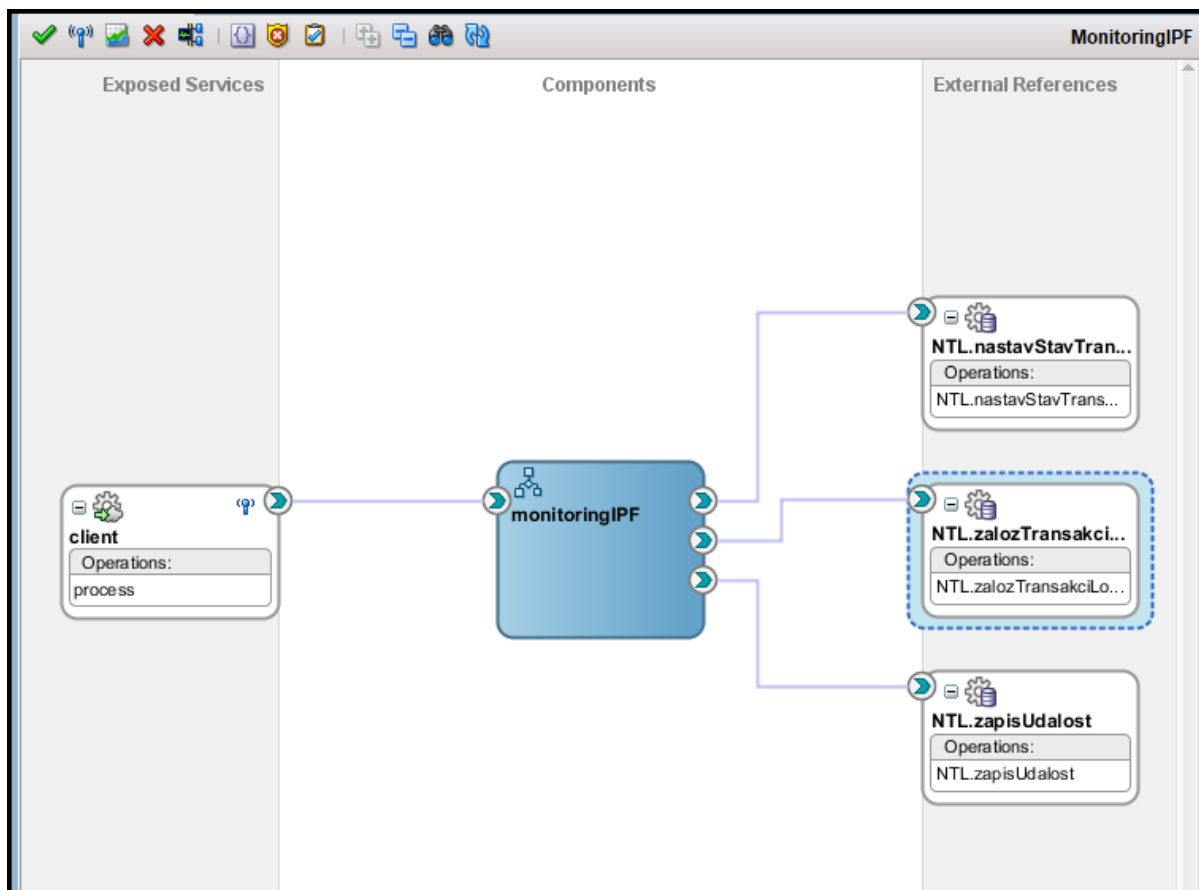
- **transactionStart:** zahájení instance služby
- **transactionSetState:** změna stavu instance služby, využívá se pro zápis výsledku služby
- **insertEvent:** vložení události
- **insertReference:** vložení reference
- **transactionComplete:** zjednodušený zápis do NTL v rámci jednoho volání. Funkce je vhodná pro jednoduché atomické služby a kde je zapotřebí vysoký výkon.

Následující diagram shrnuje typické využití API NTL ve službách IPF.


**Obrázek 6 Typické využití API NTL**

Kompletní popis NTL a jeho API je dostupný v dokumentaci NTL.

Zápis do NTL je v SOA službách realizován pomocí databázových adaptérů, viz následující obrázek.



Obrázek 7 Zápis do NTL v SOA službách

## 6.11 Homologace implementace

Cílem homologace implementace je zajistit, aby služby IPF splňovaly implementační standardy.

**Homologace implementace se týká pouze služby IPF.** Implementace cílové služby (poskytované cílovou komponentou) není předmětem homologace implementace a implementační pokyny v této metodice se jí netýkají. Zde je rozdíl oproti návrhu služby, kde návrh cílové služby zásadně ovlivňuje návrh služby IPF a celou integrační vazbu.

Homologace implementace typicky probíhá v rámci UAT testů, tj. služba již úspěšně prošla:

- fází návrhu a návrh byl úspěšně homologován
- fází implementace a funkčními testy
- integračními testy v prostředí VZP

Akce	Obsah	Zodpovědnost
Předání implementace k homologaci	<ul style="list-style-type: none"> <li>• Služba je kompletně dokumentována v Evidenci služeb (služba je ve stavu <b>testovaná</b>), včetně adresy služby v testovacím prostředí</li> <li>• Testovací protokol z integračních testů</li> </ul>	Dodavatel služby IPF

	<ul style="list-style-type: none"> <li>• Testovací scénáře</li> </ul>	
První kolo homologace	<ul style="list-style-type: none"> <li>• Schválení v případě, že je implementace vyhovující</li> <li>• Seznam připomínek v případě, že implementace nevyhovuje implementačním zásadám služeb IPF</li> </ul>	ORI VZP
Druhé kolo homologace	<ul style="list-style-type: none"> <li>• Totožné jako v prvním kole.</li> </ul>	PTP IPF

**Tabulka 25 Homologace implementace**

Dodavatel služby IPF je povinen upravit implementaci služby v případě, že implementace homologací neprojde. Na provozní prostředí IPF nemůže být nasazena služba bez úspěšného dokončení homologace implementace. Toto opatření má za cíl ochránit provoz IPF a dostupnost jejích služeb a také umožnit efektivní rozvoj a případné předání služeb IPF jinému dodavateli.

Homologaci návrhu podrobněji popisuje Metodika homologace služeb IPF (viz 14.1 Odkazy).

## 7 Dokumentace

Hlavním zdrojem dokumentace služeb IPF je Evidence služeb (ESL). ESL eviduje služby IPF i služby poskytujících komponent (např. RSZP).

Dokumentací služeb se podrobněji zabývá Metodika dokumentace služeb (viz 14.1 Odkazy).

Dokumentace služby v ESL je vstupem pro homologaci návrhu i implementace.

### 7.1 Zdrojové kódy

Dodavatel předává zdrojové kódy služeb IPF do úložiště GIT:

Druh	Cesta v úložišti
SOA/BPEL	ipf-sluzby/SOA/slozka/sluzba Příklad: ipf-sluzby/SOA/Z0-B2B-a/CentralniRegistrUpadcuB2B
OSB	ipf-sluzby/OSB/sluzba Příklad: ipf-sluzby/OSB/B0-Techws-MonitoringOSB

Tabulka 26 Umístění služeb v úložišti GIT

## 8 Testování

Funkčnost služeb IPF ověřujeme následujícími testy:

- **funkční:** ověření směrování, transformací, logiky a dalších aspektů služby IPF
- **integrační:** ověření kompletního spojení mezi konzumentem a zdrojovou službou
- **výkonové:** ověření výkonových parametrů služby

### 8.1 Funkční testy

Každá služba IPF musí být po dokončení implementace ověřena funkčním testem. Testovací scénáře vytváříme v nástroji SoapUI.

Scénáře musí pokrývat všechny operace a pokud služba obsahuje složitější logiku, musí být funkčním testem ověřena každá z větví služby.

#### 8.1.1 Využití testovacích scénářů

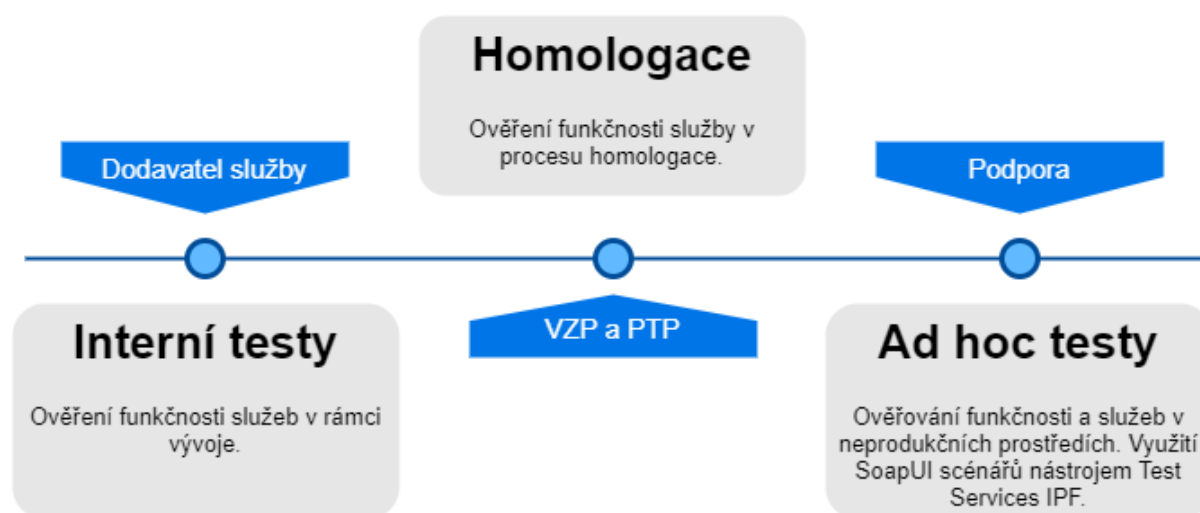
V první řadě testovací scénáře využívá Dodavatel služby IPF pro své interní testy. Následně testovací scénáře předává k homologaci implementace jako součást dodávky služeb IPF.

Testovací scénáře jsou využity při homologaci implementace pracovníky VZP a PTP (Poskytovatel Technické Podpory) IPF.

Po ukončení akceptačních testů jsou scénáře uloženy do společného úložiště testů služeb IPF. Dále jsou testovací scénáře využívány pracovníky podpory pro ad hoc ověřování funkčnosti služeb:

- ověření, zda je služba funkční v testovacím prostředí,
- k získání příkladů požadavků a odpovědí,
- k reprodukci chyb nalezených v produkci apod.

Scénáře rovněž využívá nástroj Test Services IPF.



**Obrázek 8 Využití SoapUI testovacích scénářů**

Aby mohl výše uvedený postup efektivně fungovat, je potřeba vytvářet SoapUI projekty s jednotnou strukturou a ovládáním.

Dodavatel odevzdává testovací scénáře ve verzi SoapUI OpenSource 5.5. Scénáře vytvořené v novější verzi by nemusely být kompatibilní s Test Services IPF a také s prostředím dalších uživatelů (testerů). Proto metodika určuje verzi SoapUI.



Dodavatel služeb IPF si může od VZP vyžádat vzorový testovací SoapUI projekt a využít jej jako vzor pro vlastní testovací scénáře.

### 8.1.2 Struktura SoapUI projektu

Testovací scénáře služeb sdružujeme do SoapUI projektů dle složek IPF. Např. scénáře pro službu *DetailSubjektuZP* jsou umístěny do SoapUI projektu *Z0-Pojistenci-s*.

Projekt obsahuje dvě testovací sady (TestSuite):

- sada pro pozitivní testovací scénáře: *{NazevSlozky}\_RequestSuccess*
- sada pro negativní testovací scénáře: *{NazevSlozky}\_RequestFault*

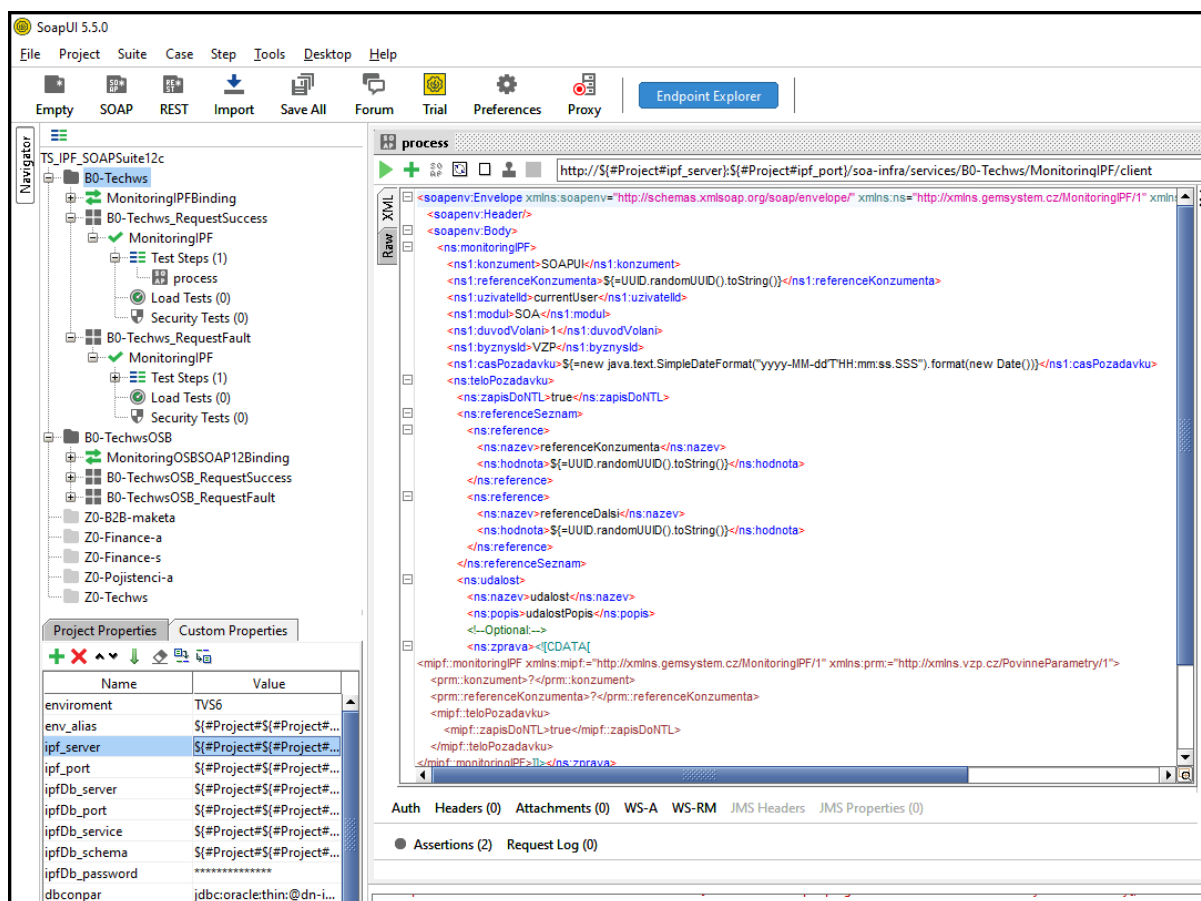
Pro každou ze služeb je vytvořen testovací případ (TestCase).

Úroveň v SoapUI	Využití	Příklad
Projekt	Složka IPF	Z0-Pojistenci-s
TestSuite	Sada pro pozitivní testovací scénáře Sada pro negativní testovací scénáře	Z0-Pojistenci-s_RequestSuccess Z0-Pojistenci-s_RequestFault
TestCase	Služba IPF	DetailSubjektuZP
TestStep	Jednotlivé testy pro ověření funkčnosti služby. Testovací kroky pokrývají všechny operace služby a všechny scénáře, které při běžném provozu služby mohou nastat.	

**Tabulka 27 Standardní struktura testovacích projektů SoapUI**

Názvy testovacích kroků (TestStep) neobsahují mezery.

Uvedenou strukturu ilustruje následující snímek ze SoapUI.

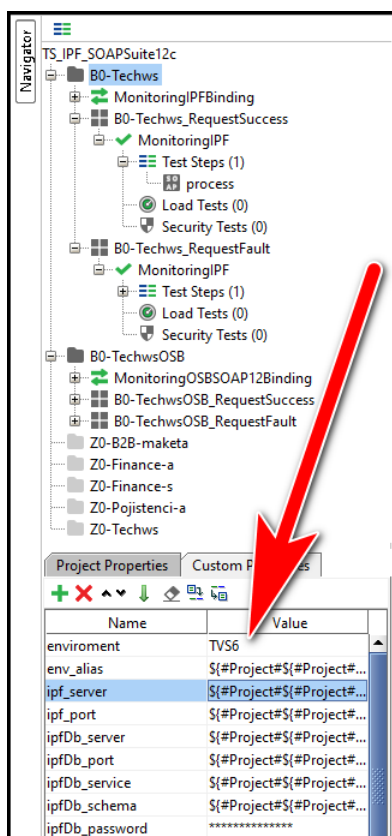


Obrázek 9 Struktura standardního SoapUI projektu

### 8.1.3 Konfigurace prostředí

Konfigurace pro jednotlivá prostředí je udržována v externích souborech (tzv. *property* soubory, které lze exportovat ze SoapUI a importovat do SoapUI). Soubory s konfigurací jsou nahrány do projektu při jeho startu.

Při přechodu mezi jednotlivými prostředími (TVS1, TVS6 apod.) pak stačí zadat identifikátor prostředí do proměnné *environment* na úrovni SoapUI projektu (viz snímek níže). Testovací scénáře jsou pak směrovány na adresy služeb IPF v daném prostředí.



Obrázek 10 Nastavení prostředí v SoapUI projektu

### 8.1.4 Asynchronní SOAP služby

Pro asynchronní SOAP služby je zapotřebí vytvořit příslušné mock služby, které jsou využity pro příjem asynchronní odpovědi.

Testovací kroky jsou seřazeny do sekvence, abychom postupně:

- zavolali službu,
- aktivovali mock službu pro přijetí asynchronní odpovědi a
- přijali a vyhodnotili asynchronní odpověď.

## 8.2 Integrovaní testy

Předpokladem pro spuštění integračního testu služby IPF je připravenost cílové služby.

V rámci integračního testu je v ideálním případě ověřeno celé propojení:

- konzument
- služba IPF
- zdrojová služba.

V odůvodněných případech (např. když není konzument připraven) může být konzumace služby provedena ze SoapUI pomocí připravených scénářů funkčních testů.

V rámci integračních scénářů mají být otestovány pozitivní i negativní scénáře zpracování služby.

## 8.3 Výkonové testy

Výkonové testy provádíme v případě, že služba vykonává výkonnostně náročné operace nebo jsou na službu kladené specifické výkonnostní nároky.

Typickými ověřovanými metrikami výkonových testů jsou:

- Odezva služby
- Nároky na zdroje (RAM, CPU)
- Chybovost služby

Při výkonovém testu obvykle nastavujeme proměnné:

- Počet vláken, ze kterých je služba volána (obvyklé a maximální očekávané zatížení), a dobu počátečního náběhu zátěže
- Četnost volání služby za jednotku času (obvyklé a maximální očekávané zatížení)
- Velikost/složitost požadavku na službu

Pro tvorbu automatizovaných testovacích scénářů, spouštění a vyhodnocování testů je preferován program jMeter. Pokud je výkonový test realizovatelný na běžné pracovní stanici v nástroji SoapUI, může být použit SoapUI, pro vyšší zátěž (desítky či stovky vláken) však nástroj SoapUI není doporučen.

## 8.4 Protokol o provedení testu

Výsledky testů jsou zachyceny v testovacím protokolu.

Testovací protokol obsahuje:

- Datum provedení testu
- Výčet provedených testovacích scénářů, jejich výsledky a odkazy na testovací data (názvy souborů)
- Osoba zodpovědná za provedení testu
- Příložená testovací data (požadavky a odpovědi)

## 9 Předání k nasazení

K nasazení jsou předávány:

- Soubory metadat určené pro nasazení do MDS, tj. popisy rozhraní služeb (XSD a abstraktní WSDL). Dodávka je ve formě jar balíčků.
- Jednotlivé kompozitní aplikace jsou ve formě jar balíčků rozdělených do adresářů podle složek, do kterých se budou instalovat.
- Instalační skripty zajišťující automatickou instalaci všech dodaných součástí
- Průvodky k instalaci popisující postup a předpoklady instalace

Příklad dodávky služby:

- B0-Techws/sca\_MonitoringIPF\_rev1.0.jar (zdrojový kód služby)
- B0-Techws/MonitoringIPF\_cfgplan.xml (konfigurační plán)
- mds/mds-B0-Techws.jar (abstraktní WSDL služby a její XSD)
- GMIPF-12.0.0-upgrade.pdf, GMIPF-12.0.0-upgrade-obsah.pdf (instalační průvodky)

Postup předávání se řídí standardním release procesem.

## 10 Nasazení do provozu

Před nasazením do provozu musí mít služba IPF homologovanou implementaci.

Administrátor VZP nasadí novou službu (resp. novou verzi) na IPF dle postupu uvedeného v instalační průvodce.

Následně v Evidenci služeb nastaví administrátor IPF stav služby na hodnotu: *Nasazená*.

## 11 Rozvoj služby

Pokud je zapotřebí službu změnit, je potřeba určit, zda:

- změna služby bude zpětně kompatibilní,
- všichni konzumenti služby přejdou na novou verzi
- je nutné zachovat více verzí služby souběžně.

Následující tabulka shrnuje možnosti vytvoření nové verze služby.

Zpětně kompatibilní	Nutné obě verze	Řešení
Ano	<i>Nová verze nevzniká</i>	Pro službu je navýšena vedlejší verze služby (např. z verze 1.0 vznikne 1.1). Služba v nové verzi nahradí předchozí verzi služby, na IPF je ve výsledku jedna služba v nové verzi. Stávající konzumenti nemusí provádět úpravy.
Ne	Ne	Je navýšena hlavní verze služby (např. z verze 1.5 vzniká 2.0). Služba v nové verzi nahradí předchozí verzi služby. Všichni konzumenti služby musí nasadit své změny současně se změnou služby IPF.
Ne	Ano	Vznikne nová služba, která má hlavní číslo verze ve svém názvu: <i>NazevSluzby-v{HlavniVerze}</i> , např. <i>PrehledVydaju-v2</i> Konzumenti, kteří vyžadují změnu, konzumují novou službu. Konzumenti, kteří zůstávají u staré služby, žádné změny nasazovat nemusí.

**Tabulka 28 Rozvoj služby a dopad na verzování**

Rozlišení změn dle zpětné kompatibility a formát verzování jsou popsány výše v 4.2.6 Verzování služeb.

### Příklad 1

*Na IPF běží služba PrehledVydaju ve verzi 1.4, kterou konzumuje komponenta A. Komponenta B vyžaduje v této službě změny, kterým se komponenta v daném čase nedokáže přizpůsobit. Na IPF bude nasazena nová služba s názvem PrehledVydaju-v2. V Evidenci služeb budou služby PrehledVydaju a PrehledVydaju-v2 evidovány samostatně.*

### Příklad 2

*Na IPF běží služba DetailSubjektuZP ve verzi 2.0, kterou konzumuje komponenta A. Komponenta B vyžaduje přidat do odpovědi služby tři nové elementy. Tato změna je přínosná rovněž pro komponentu A a obě komponenty mohou být upraveny dle společného plánu.*

*Na IPF bude nasazena služba DetailSubjektuZP ve verzi 3.0 a komponenty A a B společně přecházejí na konzumaci této verze. Služba DetailSubjektuZP ve verzi 2.0 není nadále potřebná a může být ukončena.*

## 12 Provoz

Služba IPF přechází nasazením na provozní prostředí do fáze provozu. Služba může být konzumována a lze její instance prohlížet v Dohledu Evidence služeb, popřípadě v administrátorských nástrojích SOA Suite.

V případě, kdy dojde k chybě služby, je zapotřebí rozlišovat následující příčiny:

- chyba konzumenta: konzument zaslal chybný požadavek
- chyba cílové služby: cílová služba nedokázala zpracovat požadavek a vrátila jinou než očekávanou odpověď
- chyba služby IPF: služba IPF obdržela správnou odpověď, ale nedokázala ji korektně zpracovat a předat konzumentovi služby
- chyba infrastruktury: chyba má příčinu na infrastrukturní úrovni, např. síťové prostupy, nedostupnost komunikačních prvků apod.

Typickými chybami služeb IPF jsou chyby v transformacích zpráv nebo pokud u kompozitních služeb není ošetřen konkrétní stav. Jelikož služby IPF v prostředí IS VZP ve většině případů neobsahují složitou programovou logiku, bývají častěji příčinou chyby na straně konzumenta nebo cílové služby.

V případě chyby v integrační vazbě bývá v Service Desku VZP založen IM požadavek, v rámci něhož je chyba následně řešena.

Pokud pracovník podpory IPF vyhodnotí, že informace o incidentu může pomoci jejím konzumentům nebo k hodnocení služby, doplní ji do Evidence služeb (sekce Dokumenty, seznam IM).

Zapisování IM požadavků ke službě pomáhá při vyhodnocování služby.

Podobně mohou být do seznamu IM v Evidenci služeb zapsány změnové požadavky a další informace, které se služby bezprostředně týkají.



## 13 Ukončení provozu služby

Pokud služba IPF není používána a není perspektivní do budoucna (např. kvůli změně procesů VZP nebo legislativy), je ukončena.

### Postup ukončení služby

Aktivita	Popis	Zodpovědnost
<b>Návrh na ukončení služby</b>	Návrh na ukončení služby může podat kdokoliv, kdo zjistí, že služba již není využívána a není perspektivní do budoucna.  Je provedena kontrola, zda pro danou službu není na IPF aktivní komunikace.	Administrátor IPF, Garant služby, Oddělení rozvoje integrací, Architekt
<b>Schválení ukončení</b>	Garant služby posoudí návrh a ukončení služby schválí, nebo zamítne.  Pokud Garant schválí ukončení služby, změní její stav (nebo pověří konkrétního pracovníka) v Evidenci služeb, uvede důvod ukončení a dá pokyn Administrátorovi IPF, aby zajistil odebrání služby z IPF.	Garant služby
<b>Požadavek na odebrání služby</b>	Administrátor vytvoří požadavek na dodavatele služby IPF, který dodá skripty pro odinstalaci služby a souvisejících konfigurací (směrovací aparát, B2B apod.).	Administrátor IPF
<b>Příprava upgrade</b>	Dodavatel služby IPF dodá upgrade pro odebrání služby (popřípadě více služeb).	Dodavatel služby IPF
<b>Instalace změny</b>	Administrátor IPF postupně aplikuje upgrade na jednotlivá prostředí IPF. Tímto způsobem lze zajistit konzistenci napříč prostředími.	Administrátor IPF

Tabulka 29 Aktivity při ukončení provozu služby

## 14 Přílohy

### 14.1 Odkazy

Dokument	Verze	Popis
Metodika dokumentace integračních vazeb	1.3	Metodika popisuje postupy pro dokumentaci služeb.
Metodika homologace integračních vazeb	1.3	Metodika popisuje postup homologace služeb.

Tabulka 30 Odkazy

### 14.2 VZPIPFZPRAVA

Definice objektu VZPIPFZPRAVA je přiložena v následujícím souboru:



VZPIPFZPRAVA.sql

### 14.3 Povinné parametry XML zpráv

Definice XSD šablony s povinnými parametry.



PovinneParametry.x  
sd