

Popis integračních vazeb prostřednictvím IPF a metodika realizace integračních vazeb

Příloha standardů a podmínek dodávek
informačního systému VZP ČR

UPOZORNĚNÍ:

Tento dokument je zpracován Všeobecnou zdravotní pojišťovnou České republiky (dále též jen „VZP ČR“ nebo „VZP“). Všeobecná zdravotní pojišťovna České republiky jej uveřejňuje v rámci zadávací dokumentace jí zadávaných veřejných zakázek. Tento dokument umožňuje vytvořit si představu o standardech informační architektury ICT VZP ČR. Účelem jeho uveřejnění je poskytnout informace nezbytné pro integraci dodávané komponenty se stávajícím informačním systémem v souladu se Standardy ICT- VZP- NIS.

Uveřejněním tohoto dokumentu není dotčena právní odpovědnost spojená s jeho zneužitím.

V tomto dokumentu bylo použito názvů subjektů a názvů produktů, které mohou být chráněny příslušnými právními předpisy.

Otevřením tohoto dokumentu berete výše uvedené skutečnosti na vědomí.

Obsah

1. Úvod	9
2. Popis integračních vazeb prostřednictvím IPF	9
2.1 Webové služby, IPF procesy	9
2.1.1 Pojmenování	9
2.1.2 Styly	9
2.1.3 Standardy	10
2.1.4 SOAP protokol a webové služby	10
2.1.5 WSDL	10
2.1.6 Význam elementů WSDL	10
2.1.7 Podporované WS standardy	12
2.2 Definice XSD schémat	13
2.3 Standardní datové XML elementy	13
2.4 Synchronní požadavky	14
2.5 Práce s frontou požadavků	14
2.5.1 Pojmenování Jmenné konvence názvů front	14
2.5.1.1 Názvy front aplikací (v aplikacích)	15
2.5.1.2 Názvy front na IPF	15
2.5.2 Odlišení prostředí	16
2.5.3 Jednotná zpráva pro AQ	16
2.6 Pravidla pro stavbu zprávy VZPIPFZPRAVA	18
2.7 Posílání zpráv	18
2.7.1 Směrovací aparát	18
2.8 Dokumentace webových služeb	20
2.9 Další vlastnosti dodržované v souvislosti s IPF procesy	20
2.9.1 Předepsané vlastnosti procesů	20
2.9.2 Vlastnosti aplikací komunikujících s IPF	20
2.9.3 Instalace procesů do testovacího prostředí IPF	20
2.9.4 Instalace do produkčního prostředí IPF	21
2.10 IPF partitions	21
2.10.1 Jmenné konvence pro BPEL partitions	21
2.10.2 Konvence pro umístění služeb do partitions	21
2.10.3 ESB jmenné konvence	22
3. Metodika realizace integračních vazeb	22
3.1 Účel metodiky	22

3.2	Návrh architektury integrace aplikační komponenty	22
3.3	Popis integračních vazeb v analytickém projektu	22
3.4	Testovací scénáře integračních vazeb	23
3.5	Zavedení integračních služeb do aplikace Evidence služeb	23
3.6	Popis integračních vazeb v administrátorské dokumentaci	23
3.7	Workshop k objasnění realizovaných integračních vazeb	23
3.8	Realizace změn integračních vazeb	23

Seznam obrázků

Obrázek 1- jmenné konvence front vzhledem ke směrovacímu aparátu	16
------------------------------------------------------------------------	----

Historie dokumentu

Verze	Datum	Autor	Popis
1.06	17.10.2017	ÚICT VZP ČR	

1. Úvod

Dokument obsahuje sadu standardů pro vybudování integračních vazeb nově dodávaných komponent informačního systému se stávajícími komponentami prostřednictvím integrační platformy v souladu se Standardy ICT VZP ČR. Vytvořené standardy jsou základem pro další rozšiřování systému zaváděním nových komponent a to jak „standardních“, tak i vytvářených dle specifických požadavků VZP ČR. Tento dokument je součástí výše uvedených Standardů ICT.

V případě specifikace rozšíření informačního systému zaváděním nových komponent ve smlouvě s dodavatelem, má specifikace uváděná v této smlouvě přednost před Standardy.

Cílem tohoto dokumentu je popis služeb a procesů vytvářených při realizaci integračních vazeb mezi komponentami informačního systému VZP ČR prostřednictvím integrační platformy a definování metodiky realizace integračních vazeb.

2. Popis integračních vazeb prostřednictvím IPF

2.1 Webové služby, IPF procesy

2.1.1 Pojmenování

Webové služby i služby obecně by měly zapadat do této jmenné konvence:

Název služby: ssssAaaaaaBbbbbBcccc

sss - sloveso, rozkazovací způsob, malá písmena (včetně počátečního písmene!), bez diakritiky, vyjadřuje podstatu služby, přičemž za obdobné aktivity se používá stejné označení podle následující tabulky:

AaaaaaBbbbbBcccc - další slova, první písmeno velké, bez diakritiky, převážně podstatná jména v jednotném čísle v prvním pádě, přídavná jména,...

Datové položky: AaaaaaBbbbbBcccc

AaaaaaBbbbbBcccc - několik slov, první písmeno velké, bez diakritiky, převážně podstatná jména v jednotném čísle v prvním pádě, přídavná jména,...

Pokud položka názvu nebo datová položka obsahuje zkratky, pak jsou všechny znaky velké.

např. dejNazevZZP

2.1.2 Styly

RPC style

Pro svoji jednoduchost je RPC styl blízký všem, kdo začínají pracovat s webovými službami. RPC styl byl od prvních verzí SOAPu podporován v Java nástrojích. V systémech VZP ČR bude RPC styl podporován jen ve velmi odůvodnitelných případech. V ostatních bude použit Document style.

Document style

Dokument styl je druhý ze stylů podporovaných v protokolu SOAP ve webových službách. Oproti RPC stylu, je v SOAP zprávě zaslán celý XML dokument nikoliv „pouze“ parametry volané funkce nebo procedury a samozřejmě toto volání není vázáno na konkrétní funkci nebo proceduru. XML fragment může být samozřejmě popsán ve WSDL. Návratovou hodnotou z volané služby je také XML fragment.

Dokument styl je preferovanou variantou webových služeb. RPC styl používáme jen tam, kde se ukáže obhajitelně opodstatněný.

2.1.3 Standardy

2.1.4 SOAP protokol a webové služby

SOAP protokol pro komunikaci webových služeb (WS) je nadstavbou HTTP protokolu. Fyzicky je to standardem definovaná část XML dokumentu – obálka, do které se vloží vlastní data, a vše se pak posílá protokolem HTTP. Standard pro SOAP definuje také další hlavičky HTTP.

Komunikace pomocí SOAP webovými službami je definovaná pomocí WSDL (Web Service Definition Language) a struktura vlastních dat požadavku a odpovědi pomocí XSD (XML schema definition).

2.1.5 WSDL

WSDL definuje způsob komunikace mezi službami, tj. názvy a definice posílaných zpráv, pojmenování operací, styly, název služby, u nasazené služby pak také její endpoint. Dále obsahuje definice dalších rozšíření, jako WS-Addressing, WS-Security, WS-Reliability apod.

Pro další popis je nutné vysvětlit, co znamená „namespace“ (volně také jmenný prostor). Je to definice prostředí, která označuje logickou skupinu unikátních identifikátorů (např. deklarací elementů, atributů apod. v XML). Identifikátor definovaný v namespace je s ním pak pevně asociován. Lze tak nezávisle definovat stejný identifikátor v rámci několika rozdílných namespace. Namespace může mít teoreticky jakékoliv unikátní označení, v XML se obvykle používá notace urn:JmenoNamespace nebo ještě častěji název obdobný URL - <http://LibovolnaDomenaAutora/rozlisujiciOznaceni/verze>. Volba namespace musí být unikátní, proto se používá uvedená notace, kde doménou je doména autora/dodavatele služby. V naprosté většině je to xmlns následované tečkou a doménou druhého řádu dodavatele. Za lomítkem následuje další dělení, které je již v kompetenci dodavatele a jeho interních standardů. Definice namespace se v rámci XSD definicí udává atributem **targetNamespace**.

Namespace lze definovat také explicitním uvedením xmlns atributu u každého elementu. Tento způsob definice namespace však není preferovanou variantou.

2.1.6 Význam elementů WSDL

<definitions> - tento element je kořenový element dokumentu WSDL. Obsahuje deklaraci WSDL namespace jakožto základního namespace pro dokument, takže všechny elementy patří do tohoto namespace neobsahují-li jinou deklaraci namespace. Základní namespace je definován atributem targetNamespace elementu. Atribut name je společný názvu webové služby.

<types> - vnitřní element <definitions>. Obsahuje definici dále používaných typů (dat obvykle obsažených v požadavku nebo odpovědi). Lze definovat několik schémat.

<schema> - vnitřní element <types>. Obsahuje konkrétní definici typů nebo odkaz na ně. Element obsahuje platné XSD schéma nebo odkaz na něj do souboru pomocí dalšího elementu <import>.

<message> - vnitřní element <definitions>. Pro popis struktury zpráv se používá element <message>, který obsahuje nula nebo více elementů <part>. Element <part> odpovídá parametru nebo návratové hodnotě vzdálené procedury.

<operation> - vnitřní element <portType>. Operace slouží pro přiřazení páru požadavek-odpověď k volání metody. Operace se definuje elementem <operation>, který specifikuje, která zpráva je vstupem a která výstupem pomocí podřízených elementů <input> a <output>.

<portType> - vnitřní element <definitions>. Kolekce všech operací (t.j. i metod) se nazývá portType. Element <operation> je tedy prvkem elementu <portType>

Parametry typicky definované aplikačním serverem

<binding> - vnitřní element <definitions>. Element se používá pro převod z abstraktních datových typů, zpráv a operací do konkrétní fyzické reprezentace zpráv. Definuje konkrétní aspekty operací.

Názvem binding může být cokoliv. Ovšem tento název musí být obsažen v atributu binding elementu <port> (viz definice služby výše).

<soap:binding> - vnitřní element <binding>. Uvnitř elementu <binding> může být SOAP extension element <soap:binding>, který specifikuje použitý přenosový protokol (SOAP může používat HTTP, SMTP nebo jiný protokol) a styl požadavku (např. rpc nebo dokument). Ve VZP se v naprosté většině používá **styl dokument** resp. document.

Další podřízené elementy:

Pro každou operaci, kterou služba exponuje, je třeba specifikovat hodnotu SOAPAction HTTP hlavičky. SOAPAction je HTTP hlavička, kterou klient posílá, když vyvolává službu. Server SOAP může tuto hlavičku použít pro stanovení služby nebo jí využít jiným způsobem. SOAPAction se specifikuje následujícím způsobem:

```
<binding name='WeatherSoapBinding' type='wsdl:WeatherSoapPort'>
<soap:binding style='document' transport='http://schemas.xmlsoap.org/soap/http/'>
  <operation name='GetTemperature'>
    <soap:operation soap:Action='http://tempuri.org/action/Weather.GetTemperature/'>
    ...
  </operation>
</binding>
```

V zásadě se přidává element <operation> se stejným jménem, jako operace definovaná dříve. Dvnitř elementu <operation> se vkládá element <soap:operation> s atributem soapAction. Nakonec musí být specifikováno, jako jsou kódovány vstupní a výstupní zprávy této operace.

Uvnitř elementu <operation> jsou elementy <input> a <output>, které obsahují element <soap:body>. Ten specifikuje způsob kódování dat. URI http://schemas.xmlsoap.org/soap/encoding/ určuje styl kódování, který je popsán ve specifikaci SOAP 1.1.

<service> - vnitřní element <definitions>. Každá služba je popsána prostřednictvím elementu <service>. Uvnitř elementu <service> jsou specifikovány porty, na kterých je tato služba přístupná. Port specifikuje adresu služby např.: " http://localhost/demos/wsdl/devxpert/weatherservice". Definice portu může vypadat například takto:

```
<port name='WeatherSoapPort' binding='wsdl:WeatherSoapBinding'>
<soap:address location='http://localhost/demos/wsdl/devxpert/weatherservice.asp' />
</port>
```

Každý port má unikátní jméno a atribut binding. O atributu binding viz výše. V případě používání SOAP obsahuje element port element <soap:address/> s aktuální adresou služby - URL.

<plink:partnerLinkType> - vnitřní element <definitions>. Element partnerLinkType je specifický pro integrační platformu a definuje „partner link“ v rámci BPEL procesů – viz BPEL standard.

Příklad:

```
<definitions name="NotifikujPrijem"
targetNamespace="http://wsdl.gemsystem.cz/NotifikujPrijem"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:ns4="http://xmlns.oracle.com/pcbpel/adapter/qa/outbound/"
  xmlns:ns0="http://xmlns.gemsystem.cz/NotifikujPrijem"
  xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
  xmlns:ns3="http://wsdl.gemsystem.cz/PoskytniSoubor"
  xmlns:ns2="http://xmlns.gemsystem.cz/RZPTypes"
  xmlns:client="http://wsdl.gemsystem.cz/NotifikujPrijem">
  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ns0="http://xmlns.gemsystem.cz/NotifikujPrijem"
      xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
      xmlns:client="http://wsdl.gemsystem.cz/NotifikujPrijem"
xmlns:ns2="http://xmlns.gemsystem.cz/RZPType"
      xmlns:ns3="http://wsdl.gemsystem.cz/PoskytniSoubor">
      <import namespace="http://xmlns.gemsystem.cz/RZPTypes"
schemaLocation="RZPTypes.xsd"/>
      <import namespace="http://xmlns.gemsystem.cz/NotifikujPrijem"
schemaLocation="NotifikujPrijem.xsd"/>
      <import namespace="http://wsdl.gemsystem.cz/PoskytniSoubor"
location="PoskytniSouborSOAP.wsdl"/>
    </schema>
  </types>
  <message name="NotifikujPrijemResponseMessage">
    <part name="payload" element="ns0:NotifikujPrijemResponse"/>
  </message>
  <message name="NotifikujPrijemRequestMessage">
    <part name="payload" element="ns0:NotifikujPrijem"/>
  </message>
  <portType name="NotifikujPrijem">
    <operation name="process">
      <input message="client:NotifikujPrijemRequestMessage"/>
      <output message="client:NotifikujPrijemResponseMessage"/>
    </operation>
  </portType>
  <plnk:partnerLinkType name="NotifikujPrijem">
    <plnk:role name="NotifikujPrijemProvider">
      <plnk:portType name="client:NotifikujPrijem"/>
    </plnk:role>
  </plnk:partnerLinkType>
</definitions>
```

2.1.7 Podporované WS standardy

Podporovány jsou tyto standardy:

- WS-Security
- WS-Adressing
- WS-Corelation
- WS-Eventing
- WS-Policy
- WS-Reliability

2.2 Definice XSD schémat

XSD definice jsou nedílnou součástí definice celé služby uvedené ve WSDL. Tak, jako WSDL definuje samotnou službu a způsob její komunikace, tak XSD definuje data přenášená touto službou. VZP standard nedefinuje, zda musí být definice dat uvedené v rámci WSDL souboru nebo jako samostatný soubor, který se do WSDL pomocí příslušných definujících elementů importuje.

XSD a **všechny** datové položky jím definované však **musí být zahrnuty** do příslušného namespace udaném v atributu „**targetNamespace**“ (viz 2.1.3 Standardy

SOAP protokol a webové služby).

Používá-li se atribut targetNamespace, kterým se definuje výchozí namespace, je dále povinné použít atribut elementFormDefault=“qualified“. Tím se definuje, že všechny elementy (i ty definované v potomcích rodičovských elementů (komplexní typy) budou kvalifikované, tj. budou přiřazeny do výchozího definovaného namespace. Bez atributu elementFormDefault nastaveným na qualified vzniknou v definic XML elementy, které mají anonymní namespace, což je ve VZP nepřipustné.

Je doporučeno při vystavování webových služeb aplikací prostřednictvím IPF nepublikovat datové elementy služby aplikace ve službě IPF, ale vytvořit si vlastní rozhraní (i když se bude třeba lišit jenom hodnotou namespace) a data do nového rozhraní transformovat. Předejde se tak problémům s případnými úpravami rozhraní IPF v budoucnu.

XSD definice je vhodné definovat alespoň v rámci jedné služby na jednom místě.

2.3 Standardní datové XML elementy

Elementy zpráv jsou definovány s malým počátečním písmenem. Následující slova v názvu elementu jsou psána s velkým písmenem (velbloudí nebo také camel písmo).

V komunikaci webových služeb byly ve VZP prozatím stanoveny tyto standardní elementy zpráv:

Elementy v **požadavku i odpovědi** zprávy:

<konzument>, typ string - element obsahuje název komponenty IS VZP ČR, která reprezentuje vstupní bod pro vyřizování obchodních případů daného typu. Název je jediný pro danou komponentu bez ohledu na počet typů obchodních případů iniciovaných z dané komponenty. Název komponenty by měl být výstižný, stručný a bez použití diakritiky. Např. Portál VZP ČR používá hodnotu “Portal”.

<referenceKonzumenta>, typ string - element obsahuje unikátní identifikaci volání služby na straně konzumenta. Jednoznačně identifikuje případ založený danou komponentou. V případě Portálu VZP ČR je používáno ID volání.

V případě komunikace přes AQ jsou **navíc** tyto údaje obsaženy v metadatových attributech objektu VZPIPFZPRAVA:

- atribut PUVODCE obsahuje hodnotu elementu konzument
- atribut REFERENCE obsahuje hodnotu elementu referenceKonzument

Elementy v odpovědi zprávy:

<stavVyřízeníPozadavku>, typ Complex

<kod>, typ integer

<popis>, typ string

Element obsahuje každá odpověď webové služby. V podřízených parametrech služby bez ohledu na typ, technologii použitou pro komunikaci s konzumentem, či technologii použitou pro implementaci poskytující komponenty. se uvádí stav vyřízení požadavku služby, kde „kod“ je číselný údaj číselníku:

-2 – chybná vstupní data

-1 – služba není funkční

0 – služba nevrátila žádný záznam nebo na základě požadavku nebyla provedena žádná akce

1 – výstup služby je v pořádku nebo služba vrátila právě jeden záznam.

9 – služba vrátila oproti očekávání více záznamů

Element popis může uvádět slovní interpretaci vyřízení požadavku nebo popis chybového stavu. Není však nutně povinný jej vyplňovat vždy, nemusí být uveden, pokud kod=1.

Dalším elementem, který je nutné respektovat jako standard **v rámci jakéhokoliv datového XML přenosu** je:

<dokument>, typ Complex

<nazev>, typ string

<mime>, typ string

<obsah>, typ base64String

Element dokument má pevně definovanou strukturu, kde „nazev“ je název dokumentu, „mime“ je mime typ dokumentu (viz HTTP MIME typy) a element „obsah“ obsahuje vlastní dokument v kódování Base64.

2.4 Synchronní požadavky

Komunikace synchronním způsobem webovými službami není preferovaným standardem. V některých případech je použití synchronní komunikace nevyhnutelné nebo velmi výhodné (k použití synchronní komunikace **musí být vždy oprávněný důvod**). Při rozhodování, zda a kterou komunikaci použít, je nutné respektovat základní pravidla. Při výběru se rozhodujeme podle níže uvedených parametrů komunikace:

- synchronní komunikace – odpověď vrácena v rámci aktuálně navázané relace na úrovni TCP/IP. Komunikace proběhne do cca 60 vteřin, malé zprávy.
- asynchronní komunikace – odpověď vrací volaný systém aktivním navázáním komunikace (nová relace TCP/IP) se systémem volajícím (callback). Komunikace probíhá cca déle jak 60 vteřin a/nebo jde o přenos velkých zpráv (~ jednotky MB a více).

2.5 Práce s frontou požadavků

Většina asynchronních požadavků a odpovědí bude realizována prostřednictvím JMS nebo AQ. Pokud se nejedná o existující systém bez možnosti konfigurace je vhodné, aby byly použity standardizované jmenné konvence a obsah zpráv.

Aplikace může mít libovolný počet front. Doporučujeme však minimalizovat jejich počet a typ služby odlišovat v těle zprávy.

2.5.1 Pojmenování Jmenné konvence názvů front

Název fronty: [PREFIX]JMENO_Q

Například PBVYDAJ_Q

Název Queue Table: JmenoFronty_T

Například PBVYDAJ_T

2.5.1.1 Názvy front aplikací (v aplikacích)

Názvy front AQ podléhají jmenným konvencím aplikace. Název fronty, kterou aplikace přijímá zprávy od IPF (to jest Fronty v aplikaci) má maximální **povolenou délku 10 znaků** (delší řetězec způsobí nefunkčnost směrovacího aparátu).

2.5.1.2 Názvy front na IPF

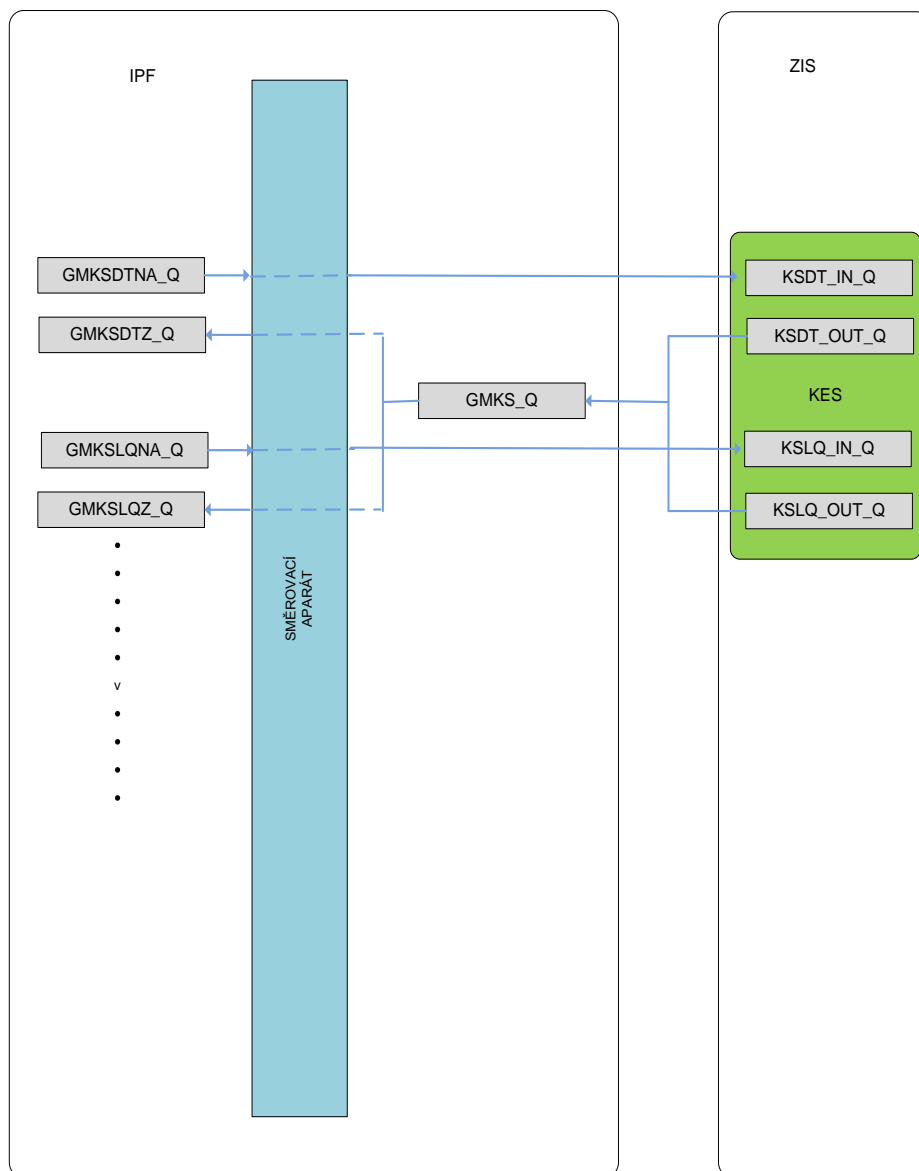
Názvy front, nad kterými je postaven kombinovaný AQ adapter, má tvar **WWXXXYYYYZZ_Q**.
Maximální délka 30 znaků.

- Část WW je tvořena prefixem dodavatele (např. GM pro GEM System International s.r.o., PB pro PIKE ELECTRONICS s.r.o., Brno).
- Část XXX je název aplikace, ze které přicházejí zprávy. Délka 3 znaky (např. KES pro aplikaci KES v ZIS).
- Část YYYYY slouží pro rozlišení front komunikující se stejnou aplikací. Je možné vynechat, pokud s danou aplikací nekomunikuje více služeb. Tato možnost není doporučena z důvodu budoucího rozšíření komunikace aplikace s dalšími službami. Délka 0 – 5 znaků.
- Část ZZ tvoří buď výraz NA, pokud se jedná o AQ frontu určenou pro zprávy posílané ven z IPF nebo výraz Z pro AQ frontu určenou pro zprávy posílané do IPF.

Název fronty, do které jsou propagovány všechny zprávy z dané aplikace umístěné v ZISu, má tvar **WWXXX_Q**.

- Část WW je tvořena prefixem dodavatele.
- Část XXX je název aplikace, ze které přicházejí zprávy. Délka 3 znaky (např. KES pro aplikaci KES v ZIS).

Všechny zde uvedené počty znaků jsou doporučené hodnoty.



Obrázek 1- jmenné konvence front vzhledem ke směrovacímu aparátu

2.5.2 Odlišení prostředí

Tělo zprávy obsahuje atribut, který umožňuje odlišit prostředí, z kterého zpráva vznikla a pro které je určena. Tento atribut se ve standardní zprávě nazývá „Prostředí“. Může nabývat hodnot „PROD“, „DEV“ nebo „TEST“. Každá instance komponenty musí být konfigurovatelná podle toho o jaké prostředí se jedná. Zprávy, které by do tohoto prostředí neměly přicházet, musí být logovány a jejich zpracování ignorováno.

2.5.3 Jednotná zpráva pro AQ

```
CREATE TYPE VZPIPFZPRAVA IS OBJECT (
  -- $Id: VZPIPFZPRAVA.sql,v 1.2 2006/09/29 07:37:17 sherry Exp $
  -- ID zpravy pridane nekterym z odesilatelu (neni to ID zpravy ve fronte AQ)
  MsgID RAW(16),
  -- Korelacni ID:
```


-- pokud odpovídám na zprávu, která měla vyplněný atribut CorrID, uvedu do něj ve své odpovědi totožnost
-- pokud přijde zpráva s prázdným CorrID, vyplním do CorrID své odpovědi hodnotu MsgID zprávy přicházející

CorrID RAW(16),
-- To kde vznikl celý tok zpráv, například "PORTAL"
Původce VARCHAR2(30),
-- Odesílatel této zprávy například "IPF" nebo "VZPP2-Outcome"
Odesílatel VARCHAR2(30),
-- Upřesnění odesílatele, např. "7200" (rozlišení pro případ, že je více instancí aplikace)
OdesílatelDoplnek VARCHAR2(30),
-- Adresát této zprávy, např. "VZPP2/Outcome"
Příjemce VARCHAR2(30),
-- Upřesnění adresáta zprávy, např. "7200" (z IPF bude obsahovat číslo okresu) je to doplněk kdy máme více instancí aplikací
PříjemceDoplnek VARCHAR2(30),
-- Prostředí: "PROD" pro produkci, "TEST" testovacího prostředí,
-- "DEV" vývojové prostředí
Prostředí VARCHAR2(30),
-- Název služby na okres, např. "PrijemDavek"
NázevSluzby VARCHAR2(30),
-- Název zprávy, například "KDavka"
NázevZpravy VARCHAR2(30),
-- Verze zprávy například "3" (v budoucnu se může měnit, aplikace musí znát verzi komunikace)
VerzeZpravy VARCHAR2(30),
-- např. ID podání z PORTALU (obecně nějaká reference původního systému)
Reference VARCHAR2(128),
-- k volnému použití (obsahuje pouze provozní/komunikační informace ne obchodní data)
ParametrV1 VARCHAR2(128),
-- k volnému použití (obsahuje pouze provozní/komunikační informace ne obchodní data)
ParametrV2 VARCHAR2(128),
-- k volnému použití (obsahuje pouze provozní/komunikační informace ne obchodní data)
ParametrD1 DATE,
-- k volnému použití (obsahuje pouze provozní/komunikační informace ne obchodní data)
ParametrD2 DATE,
-- k volnému použití (obsahuje pouze provozní/komunikační informace ne obchodní data)
ParametrN1 NUMBER,
-- k volnému použití (obsahuje pouze provozní/komunikační informace ne obchodní data)
ParametrN2 NUMBER,
-- znaková obchodní data (např. XML)
DataC CLOB,

```
-- binární obchodní data (napr. Datové rozhraní)
```

```
DataB BLOB
```

```
);
```

```
/
```

2.6 Pravidla pro stavbu zprávy VZPIPFZPRAVA

Existují doporučená pravidla, která svazují některé parametry typu VZPIPFZPRAVA:

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

2.7 Posílání zpráv

K posílání zpráv mezi systémy se primárně užívá Směrovacího aparátu (viz také příručka administrátora IFP - ProgramAdmin). Definice AQ front resp. queue table na IPF je proto vždy vytvářena s parametrem `MULTICONSUMER = FALSE` (výchozí nastavení).

2.7.1 Směrovací aparát

Směrovací aparát přenáší zprávy z front, do nichž zapisují procesy, do front, z nichž jsou dále propagovány systémem AQ do vzdálených databází. Tento směr je označen jako *outbound*. Směrovací záznamy jsou ukládány do tabulky `PBIPFOUTBOUND`.

Sloupce tabulky PBIPFOUTBOUND

Název sloupce	Hodnota	Příklady
<code>SRC_QUEUE</code>	Název fronty, do níž zapisuje proces	<code>PBVYUPOJNAUP_Q</code> , <code>PBVALPOJNA_Q</code>
<code>PRIJEMCE</code>	Název aplikace, pro kterou jsou zprávy určeny	<code>VZPP2-Outcome</code> , <code>VZPP2-Income</code> , <code>PSi</code>
<code>PRIJEMCEDOPLNEK</code>	Odlišení instancí aplikací – číslo ÚP	1900, 7200, ...; u ústředních aplikací 9900
<code>PROSTREDI</code>	Prostředí IPF	<code>DEV</code> , <code>TEST</code> nebo <code>PROD</code>
<code>NAZEVSLUZBY</code>	Název služby	<code>VytvorSestavu</code> , <code>VyuctovaniPojistence</code>
<code>NAZEVZPRAVY</code>	Název zprávy	<code>VytvorSestavu</code> , <code>VyuctovaniPojistence</code>
<code>VERZEPRAVY</code>	Verze zprávy	Výchozí verze všech zpráv je 1.0. Dále se pokračuje 1.1, 1.2, ..., 2.0 (dle míry provedených úprav).
<code>DEST_QUEUE</code>	Název fronty ve vzdálené databázi, do níž mají být propagovány zprávy z IPF.	<code>PBPSIDEJ_Q</code> , <code>PEIPFIN_Q</code>

Název sloupce	Hodnota	Příklady
	Maximální délka názvu je 10 znaků.	
DEST_QUEUE_OWNER	Vlastník fronty ve vzdálené databázi, do níž mají být propagovány zprávy z IPF	PVZP, CVZP
PBIPFDBLINK	Klíč databázového linku, který má být při propagaci do vzdálené databáze použit	VZP_OP19_HP1, VZP_OP72_HP1, VZP_OP99_HP2
KORELOVAT	Ano nebo Ne	Pokud směrovací záznam definuje odeslání požadavku z kombinovaného AQ adaptéru (invoke – receive), pak Ano. V ostatních případech Ne.

Po vložení záznamů do tabulky PBIPFOUTBOUND je nutné spustit proceduru GMIPFAQDISPATCH_ADM.CREATE_QUEUEES, která vytvoří fronty, a propagační joby do vzdálené databáze.

Směrovací aparát přenáší zprávy z front, do nichž propagují zprávy vzdálené aplikace, do front, z nichž jsou vybírány procesy. Tento směr je označen jako *inbound*. Směrovací záznamy jsou ukládány do tabulky PBIPFINBOUND.

Sloupce tabulky PBIPFINBOUND

Název sloupce	Hodnota	Příklady
SRC_QUEUE	Název fronty, do níž vzdálená aplikace propaguje odchozí zprávy	PBVDYD_Q, PBPRI_Q, GMKES_Q
PROSTREDI	Prostředí IPF	DEV, TEST nebo PROD
NAZEVSUZBY	Název služby	VytvorSestavu, VyuctovaniPojistence
NAZEZPRAVY	Název zprávy	VytvorSestavu, VyuctovaniPojistenceAck
VERZEZPRAVY	Verze zprávy	Výchozí verze všech zpráv je 1.0. Dále se pokračuje 1.1, 1.2, ..., 2.0 (dle míry provedených úprav).
DEST_QUEUE	Fronta, kterou čte proces.	PBVDYUPOJZUP_Q, PBVALPOJZ_Q
KORELOVAT	Ano nebo Ne	Pokud směrovací záznam definuje příjem odpovědi do kombinovaného AQ adaptéru (invoke – receive), pak Ano. V ostatních případech Ne.

Před vložením záznamu do tabulky PBIPFINBOUND je nutno zkontrolovat, zda fronta, jejíž název je uveden ve sloupci SRC_QUEUE, existuje a lze z ní vyzvedávat zprávy.

2.8 Dokumentace webových služeb

Každá webová služba bude už od počátku svého vývoje evidována a dokumentována v Evidenci služeb VZP.

2.9 Další vlastnosti dodržované v souvislosti s IPF procesy

2.9.1 Předepsané vlastnosti procesů

1. Proces, který komunikuje asynchronně přes Advanced Queuing (dále AQ), definuje preferenci *Prostředí*. Preference nabývá hodnot DEV (vývoj na prostředcích dodavatele), TEST (testování v testovacím prostředí VZP ČR) nebo PROD (provoz v produkčním prostředí VZP ČR). Dle hodnoty preference *Prostředí* lze větvit činnost procesu, hodnota preference je též hodnotou atributu VZPIPFZPRAVA.PROSTREDI všech zpráv, které proces odesílá i přijímá. Uvedené dále v přiměřené míře (vždy kdy je to alespoň trochu smysluplné) platí i u ostatních služeb.
2. Persistentní data proces ukládá do Integrovaní cache. Názvy tabulek jsou odlišeny prefixy, aby nedošlo ke konfliktu názvů: PB – PIKE ELECTRONIC, GM – GEM SYSTEM. K datům a uloženým procedurám přistupuje proces pomocí lokátoru eis/DB/IntegracniCache.
3. Fronty AQ jsou definovány v Integrovaní cache. K frontám proces přistupuje pomocí lokátoru eis/AQ/IntegracniCacheOutbound (pro vkládání zpráv do front) a eis/AQ/IntegracniCacheInbound (pro výběr zpráv z front).
4. Preference Debug používaná na SOA 10g pro volitelné ukládání instancí synchronních služeb do dehydration store nelze v případě SOA 11g z principu využít a ani pro něj neexistuje adekvátní náhrada. Z migrovaných služeb je tak parametr odstraněn.
5. Z provozních důvodů nesmí jedna instance daného procesu obsahovat více jak 1 000 aktivit.

2.9.2 Vlastnosti aplikací komunikujících s IPF

1. Aplikace komunikující s IPF přes rozlehlou síť (WAN), komunikuje zásadně asynchronně prostřednictvím front AQ – jak v případě, že Integrovaní platformě poskytuje službu, tak v případě, že konzumuje službu Integrovaní platformy.
2. Aplikace propaguje všechny zprávy všech odesílacích front do jediné společné fronty v IPF. Vlastník fronty je ICACHE. Aplikace se připojuje prostřednictvím databázového linku pod identifikací vzdáleného uživatele ICACHEPROPAG.
3. Aplikace propaguje zprávy do IPF přes port 1501.

2.9.3 Instalace procesů do testovacího prostředí IPF

1. Instalaci nových komponent do testovacího prostředí IPF provádí na základě dohody s administrátory IPF buď dodavatel nebo administrátor IPF.
2. Při instalaci do testovacího prostředí se ověřuje postup instalace, který bude následně použit při instalaci do produkčního prostředí.
3. Ověřují se skripty pro vytváření záznamů do směrovacích tabulek, vytváření nových front a databázových objektů v Integrovaní cache.
4. Ve „wrapper WSDL“ jsou použity URL endpointů testovacího prostředí.
5. Po deploymentu procesů, u nichž je to třeba, se mění hodnota preference *Prostředí* na TEST.

2.9.4 Instalace do produkčního prostředí IPF

1. Instalaci nových komponent do produkčního prostředí IPF instaluje výhradně administrátor IPF na základě návodu dodavatele. Návod je ověřen při instalaci do testovacího prostředí.
2. Pokud je třeba provést zásah do konfigurace produkčního prostředí (parametry BPEL PM, parametry domény apod.), provádí je výhradně administrátor IPF.
3. Pro instalaci objektů v Integrační cache se zásadně používají skripty ověřené při instalaci do testovacího prostředí IPF.
4. Pro založení záznamů v tabulce PBIPFINBOUND se zásadně používají skripty ověřené při instalaci do testovacího prostředí, v nichž se před instalací do produkčního prostředí změní pouze hodnota pro sloupec PROSTREDI z TEST na PROD.
5. Pro založení záznamů v tabulce PBIPFOUTBOUND se zásadně používají skripty ověřené při instalaci do testovacího prostředí, v nichž se před instalací do produkčního prostředí změní pouze hodnota pro sloupec PROSTREDI z TEST na PROD. V případě, že IPF bude nově zasílat zprávy do aplikace ÚP, se skript obohatí o cyklus přes všechna ÚP (PRIJEMCEDOPLNEK).
6. Pro deployment procesů do produkčního prostředí se zásadně používají instalační balíčky (jar) ověřené v testovacím prostředí.
7. Po deploymentu procesů, u nichž je to třeba, se mění hodnota preference Prostredi na PROD.

2.10 IPF partitions

2.10.1 Jmenné konvence pro BPEL partitions

- Název partition bude vystihovat oblast řešenou umístěvanými procesy.
- Rozdělení podle synchronnosti bude rozlišeno na konci názvu pomlčkou a přidáním písmene „s“ pro služby synchronní a písmenem „a“ pro služby asynchronní.
- Pro založení nové partition je nutné včas podat návrh na založení nové partition. Návrh obsahuje název, účel a nastavení partition v testovacím a produkčním prostředí.
- Dokud nebude návrh odsouhlasen všemi zúčastněnými stranami, není dovoleno partition založit.
- Zúčastněnými stranami jsou zástupci společností VZP, HP, GEM, PIKE.

Po odsouhlasení návrhu všemi zúčastněnými stranami je třeba vytvořit partition ve všech prostředích (a případně do ní přesunout procesy z partící Vyvoj-s a Vyvoj-a).

Pro umístění technologických služeb slouží partition „techws“. Tato partition obsahuje služby a procesy zřízené pouze z technologických důvodů. Tyto procesy nejsou určeny pro užívání libovolnými konzumenty a nebudou tedy publikovány.

2.10.2 Konvence pro umístění služeb do partitions

- Související procesy se budou umísťovat do stejné partition (např. procesy určené pro získávání údajů o pojištěncích, bez ohledu na to, zda jsou údaje získávány z Registru subjektů zdravotního pojištění nebo z Centrálního registru pojištěnců).
- Procesy se budou umísťovat do partitions podle synchronnosti. Snaha o oddělení synchronních a asynchronních služeb do separátních partitions pro potřeby přehlednosti.
- Pokud není žádná z již vytvořených partitions vyhovující pro umístění daného procesu, je třeba dodržet pravidla pro vytváření partitions.
- Do partition techws jsou umísťovány procesy s nulovou business funkcí, určené ke konkrétnímu řešení. Tyto procesy by neměly být nijak veřejně používány.
- Procesy specifické pro B2B jsou ujišťovány do partition B2B.

- Pro vývoj procesů určených do partition, která dosud nebyla odsouhlasena a vytvořena, jsou určeny partitions Vyvoj-s a Vyvoj-a, které lze založit v případě potřeby na vývojovém prostředí kdykoliv.

Při jakékoli sebemenší pochybnosti o správnosti umístění BPEL procesu do konkrétní partition je nutné kontaktovat osobu zodpovědnou za správu příslušné partition.

2.10.3 ESB jmenné konvence

Současné řešení ESB obsahuje pro umístění WSDL a jejich směřování Systémy (system) a skupiny (group). Systémy jsou nadřazeny skupinám. ESB systémy odpovídají BPEL partitions. Skupiny jsou použitelné pro další volné třídění. Pro ESB systémy platí výše uvedené konvence (zejména jmenné) jako pro BPEL partitions.

3. Metodika realizace integračních vazeb

3.1 Účel metodiky

Tato metodika vymezuje základní rámec procesů, které probíhají v rámci realizace integračních vazeb mezi aplikačními komponentami informačního systému VZP ČR. Procesy jsou zastřešovány Kompetenčním centrem integrace (dále KCI) ÚICT, které bylo zřízeno pokynem náměstka ředitele VZP ČR pro informatiku.

Mezi základní činnosti KCI patří dohled nad integračními vazbami mezi aplikačními komponentami informačního systému:

- Posouzení a schválení integračních vazeb mezi aplikačními komponentami
- Posouzení a schválení navrhovaných služeb realizujících integrační vazby
- Správa integračních vazeb v aplikaci Evidence služeb
- Posouzení a schválení testovacích scénářů integračních testů
- Kontrola kvality popisu integračních vazeb v administrátorské dokumentaci

Metodika realizace integračních vazeb stanovuje závazné postupy, které musí být dodržovány ve vztahu ke KCI a jsou popsány v následujících kapitolách.

3.2 Návrh architektury integrace aplikační komponenty

V rámci procesu dohledu nad integračními vazbami mezi aplikačními komponentami uspořádá dodavatel příslušné komponenty workshop, na kterém představí KCI předpokládanou architekturu začlenění dodávané aplikační komponenty do IS VZP ČR s důrazem na navrhované integrační vazby.

V rámci workshopu se dodavatel zaměří zejména na popis předpokládaných integračních vazeb v souladu se Standardy IS VZP ČR. Budou popisovány služby požadované od IPF i služby nabízené komponentou. Součástí popisu bude i druh předpokládaných integračních vazeb (synchronní služby, asynchronní služby, silná datová integrace,...).

Navrhovaná architektura integračních vazeb podléhá posouzení a schválení Kompetenčním centrem integrace.

3.3 Popis integračních vazeb v analytickém projektu

V rámci etapy zpracování analytického projektu popíše dodavatel v samostatné kapitole předpokládané integrační vazby formou služeb v souladu se Standardy IS. Příslušná kapitola analytického projektu musí být předložena Kompetenčnímu centru integrace k posouzení a schválení.

3.4 Testovací scénáře integračních vazeb

V rámci projektu implementace příslušné aplikační komponenty do IS VZP ČR zpracuje dodavatel testovací scénáře integračních testů. Tyto scénáře musí být předloženy Kompetenčnímu centru integrace k posouzení a schválení.

3.5 Zavedení integračních služeb do aplikace Evidence služeb

V rámci projektu implementace příslušné aplikační komponenty do IS VZP ČR zavede dodavatel příslušné aplikační komponenty všechny integrační služby do aplikace Evidence služeb. Tento krok učiní dodavatel nejpozději v etapě instalace aplikační komponenty do provozního prostředí IS.

Potvrzení úplnosti služeb v Evidenci služeb KCI bude součástí akceptační procedury v rámci projektu implementace příslušné aplikační komponenty dodavatelem.

3.6 Popis integračních vazeb v administrátorské dokumentaci

Realizované integrační vazby v rámci implementace aplikační komponenty popíše dodavatel v administrátorské dokumentaci. V samostatné kapitole se zaměří zejména:

- Kompletní seznam služeb poskytovaných implementovanou aplikační komponentou
- Popis realizovaných AQ front
- Přehled konzumovaných služeb v prostředí IPF

Potvrzení úplnosti popisu integračních služeb v administrátorské dokumentaci KCI bude součástí akceptační procedury v rámci projektu implementace příslušné aplikační komponenty dodavatelem.

3.7 Workshop k objasnění realizovaných integračních vazeb

V rámci závěrečného testování implementované aplikační komponenty uspořádá dodavatel aplikační komponenty workshop, na kterém objasní implementované integrační vazby s ostatními aplikačními komponentami zejména pro KCI. Ke každé integrační vazbě uvede, kde v předané dokumentaci je popsána.

3.8 Realizace změn integračních vazeb

Tato metodika se týká rovněž všech procesů změn aplikačních komponent s dopadem na integrační vazby, a to i těch, které jsou realizovány mimo rámec projektového řízení (projektové aktivity, změny, opravy, servisní zásahy). Výše uvedené odstavce 3.2 – 3.7 se použijí přiměřeně.