**PROVÁDĚCÍ KONCEPT SW ŘEŠENÍ (PK)**

projektu

Národní informační systém integrovaného záchranného systému (NIS IZS)

část

**B. Dodávaný SW**

Dokument obsahuje: Popis softwarové architektury NIS IZS.

Verze: 5.1b

Schválil za Dodavatele: RNDr. Vladimír Příbramský

Datum předání k připomínkám: 17/4/2014

Obsah

[1 Architektura softwarové části 4](#_Toc370070600)

[1.1 Architektura CDC 4](#_Toc370070601)

[1.1.1 Architektura CDC – detail IPL 5](#_Toc370070602)

[1.2 Architektura KDC 6](#_Toc370070603)

[1.2.1 Architektura KDC – detail GIS 7](#_Toc370070604)

[1.2.2 Architektura KDC – detail miniIPL 8](#_Toc370070605)

[1.3 Schéma komunikace 9](#_Toc370070606)

[1.4 Vývojová platforma 12](#_Toc370070607)

[1.4.1 Technologie podporující vývoj 12](#_Toc370070608)

[1.5 Testovací a ověřovací platforma 13](#_Toc370070609)

[1.6 Školící platforma 13](#_Toc370070610)

[1.7 Datová věta 13](#_Toc370070611)

[1.8 Datový Model 14](#_Toc370070612)

[1.8.1 Metodika popisu datového modelu 14](#_Toc370070613)

[1.9 Integritní kritéria 15](#_Toc370070614)

[2 Technické řešení SW komponent 16](#_Toc370070615)

[2.1 NSPTV aplikace 16](#_Toc370070616)

[2.2 Architektura klienta 17](#_Toc370070617)

[2.2.1 Tlustý klient 17](#_Toc370070618)

[2.3 Integrační platforma (IPL) 17](#_Toc370070619)

[2.3.2 Failover 19](#_Toc370070620)

[2.3.3 Balancing 19](#_Toc370070621)

[2.4 MiniIPL na KDC 21](#_Toc370070622)

[2.4.1 Ostrovní režim 22](#_Toc370070623)

[2.5 Komponenty pro centrální GIS v rámci NIS IZS 22](#_Toc370070624)

[2.5.1 Popis komponent centrálního GIS 23](#_Toc370070625)

[2.6 Komponenty pro GIS v rámci KDC 26](#_Toc370070626)

[2.7 Databáze 27](#_Toc370070627)

[2.7.1 Replikace DB 27](#_Toc370070628)

[2.7.2 Návrh lokálního fail-over řešení pro PostgreSQL databázi 29](#_Toc370070629)

[2.7.3 Návrh řešení geografické konfigurace database 30](#_Toc370070630)

[2.7.4 Další původně zvažované varianty implementace databázové vrstvy: 30](#_Toc370070631)

[2.7.5 Replikační technologie: 31](#_Toc370070632)

[2.7.6 Příklady replikačních řešení pro konkrétní produkty 31](#_Toc370070633)

[2.7.7 Analýza zvažovaných řešení: 32](#_Toc370070634)

[2.8 Licence 33](#_Toc370070635)

[2.8.1 OS – Operační systém 33](#_Toc370070636)

[2.8.2 DBMS – Databáze 33](#_Toc370070637)

[2.8.3 BIRT - Business Intelligence and Reporting Tools 33](#_Toc370070638)

[2.8.4 JBoss Fuse – IPL 33](#_Toc370070639)

[2.8.5 ACTIVITI - BPM systém 34](#_Toc370070640)

[2.8.6 DROOLS - BR systém 34](#_Toc370070641)

[2.8.7 MYBATIS - Objektově relační mapování 34](#_Toc370070642)

[2.8.8 GISSRV – GIS server 34](#_Toc370070643)

[2.8.9 GISADM – Administrátorský nástroj GIS 34](#_Toc370070644)

[2.8.10 METAGIS - Metainformační systém pro GIS 34](#_Toc370070645)

[2.8.11 ETL - Transformace dat 35](#_Toc370070646)

[2.8.12 VFR - RUIAN import 35](#_Toc370070647)

[2.8.13 Mapové podklady CEDA 35](#_Toc370070648)

[2.8.14 JSDI - Příjem a zpracování DDR JSDI 35](#_Toc370070649)

[2.8.15 HELPER - MST Helper service 36](#_Toc370070650)

[2.9 Reporting 36](#_Toc370070651)

[2.10 Reference na produkty 36](#_Toc370070652)

[2.10.1 JBoss Fuse 36](#_Toc370070653)

[2.10.2 Activiti 36](#_Toc370070654)

[2.10.3 PostgreSQL 36](#_Toc370070655)

[2.11 Open source 37](#_Toc370070656)

[2.12 Koncept provozu 37](#_Toc370070657)

[2.12.1 Instalace serverové části 37](#_Toc370070658)

[2.12.2 Instalace klientské aplikace 37](#_Toc370070659)

[2.13 Aktualizace 38](#_Toc370070660)

[2.13.1 Aktualizace serverové části 38](#_Toc370070661)

[2.13.2 Aktualizace klientské aplikace 38](#_Toc370070662)

[2.14 Monitoring 38](#_Toc370070663)

[2.14.1 Monitoring serverové části 38](#_Toc370070664)

[2.14.2 Monitoring koncových pracovišť 38](#_Toc370070665)

[2.15 Release a Incident Management 38](#_Toc370070666)

# Architektura softwarové části

Softwarové řešení dodávané aplikace NIS IZS je členěno do několika logických celků, které jsou rozmístěny jednak v centrálních datových centrech a jednak v krajských datových centrech. Navržené řešení a použité technologie jsou vybrány s ohledem na požadovanou robustnost výsledného řešení. CDC představuje logickou jednotku, ve které je řízen samotný proces příjmu tísňového volání včetně ukládání primárních dat o událostech a informacích s nimi spojených. Softwarové řešení v CDC je navrženo tak, aby mohlo fungovat jak v paralelním režimu pro rozložení zátěže, tak v režimu, kdy je funkcionalita v CDC omezena.

Krajské datové centrum z pohledu softwaru představuje hlavní vstupní komunikační uzel. Do KDC přechází telefonní hovory z VTS a jsou dále směřovány na odpovídající pracoviště složek IZS. KDC jako takové poskytuje funkcionalitu i pro OŘ a umožňuje v rámci složek komunikaci mezi OŘ a NSPTV. Při ostrovním režimu zajišťuje KDC mezisložkovou komunikaci v rámci OŘ a následnou aktualizaci událostí s NSPTV po přechodu do normálního režimu. Složkami IZS se dále v textu myslí „základní složky IZS“.

Softwarové řešení NIS IZS je děleno do následujících funkčních bloků:

**Telefonie a nahrávání** – skládá se z produktů realizující zpracování telefonních hovorů, ovládání telefonu a managementu telefonních ústředen. Dále je součásti toho bloku i řešení pro nahrávání a přehrávání hovorů (viz kapitola Telefonie).

**GIS** – reprezentuje produkty pro práci s mapovými podklady a vrstvami informací nad nimi. Funkcionalita GISu je formou mapových služeb na protokolu REST. GISová komponenta má vlastní databázi oddělenou od databáze událostí.

**IPL** – představuje logický celek, který propojuje logickou komunikaci v systému a směřuje požadavky na informace. Jádrem IPL je opensource technologie JBoss Fuse. Součástí IPL je také integrace na procesní engine realizující samotný proces příjmu tísňového volání opět na opensource technologii Activiti a JBoss Drools. IPL jako takové je pouze na CDC.

**miniIPL** – plní obdobnou roli jako IPL na CDC s tím rozdílem, že neobsahuje procesní engine. Řešení je analogicky postavena na technologii JBoss Fuse.

**Databáze událostí** – slouží k ukládání primárních dat NSPTV. Databázové řešení je realizováno pomocí opensource produktu PostreSQL a komerční nástavby enterpriseDB (nebo jiné s minimálně odpovídající 100% funkcionalitou a spolehlivostí), které realizuje multi-master replikaci mezi jednotlivými CDC.

## Architektura CDC

Centrální datové centrum poskytuje z pohledu NSPTV centrální logické uložiště pro práci s jednotlivými událostmi. Dále je v CDC řízen samotný proces přístup tísňového volání a přístup k mapovým pokladům pro NSPTV. Telefonie a nahrávání jsou popsány v samostatné kapitole.

Obrázek 1 popisuje kompletní pohled na rozložení řešení v rámci CDC do detailních logických bloků. SW řešení v rámci jednoho CDC obsahuje dvě funkční větve, které jsou logicky oddělené a v případě výpadku jednoho celku, přejímá bez výpadku funkcionality CDC, činnosti větev druhá. Toto zdvojení je důležité pro zajištění běhu řešení při výpadku na úrovni operačních systému či jednotlivých aplikací.

**Legenda**

* Oranžově jsou na obrázku zobrazeny jednotlivé SW řešení
* Modře jsou zobrazeny logické celky na úrovni virtualizovaných operačních systémů. Modrá propojení reprezentují komunikaci mezi jednotlivými systémy.
* Zeleně jsou zobrazeny HW segmenty, které jsou důležité z pohledu komunikace mezi softwarovými celky
* Fialová a žlutá znázorňují komunikace uvnitř jednotlivých clusterů.



Obrázek 1. Detail architektury CDC na úrovni logických bloků

Rozdělení na dvě funkční větve v rámci CDC je zajištěno na úrovni balancingu a konfigurace aplikační logiky, které sama o sobě podporuje běh v clusteru.

Při nedostupnosti některé ze služeb CDC (tzn. po výpadku dvojené funkcionality) je CDC do opravení výpadku odpojeno a jeho funkcionalitu přebírá další CDC. Dle typu výpadku je odstavení zajištěno kombinací automatizovaných procesů a dohledového centra (detailně v Recovery scénářích).

### Architektura CDC – detail IPL

Druhou velkou části SW řešení na CDC je samotná integrační platforma včetně podpůrného SW. Jádrem IPL části je komplexní řešení JBoss Fuse (detailně v kapitole IPL), které poskytuje směrování informací a poskytování služeb v rámci NIS IZS. IPL CDC poskytuje operátorské aplikace NSPTV přístup k informacím událostech. V prvním kroku se na CDC řeší autentizace a autorizace uživatelů v rámci IDM (Microsoft Active Directory). IDM řešení je zapojeno v clusteru tak, aby byli dostupné jednotné informace v rámci všech CDC.

JBoss Fuse je zapojen také v cluster módu – technologie Fuse Fabric, rámci CDC je IPL rozdělena na dvě větve, které obsahují každá po dvou uzlech v módu master/slave. Díky tomuto zapojení je řešení odolné proti pádu jednotlivých uzlů Fuse boxu. Na CDC je s technologií Fuse integrováno opensource BPM řešení Actitivi s JBoss Drools enginem. Tento funkční celek zajišťuje proces příjmu tísňového volání. Funkcionalita procesního enginu je v rámci každého CDC zdvojena z důvodu vysoké dostupnosti. IPL je napojena na databázové rozhraní, kde jsou ukládány veškeré informace NSPTV spojené s událostmi. Databáze funguje modelu vysoké dostupnosti, aby byla odolná proti výpadku databázového uzlu v rámci CDC (detailněji v části Recovery scénáře). Neaktuální události jsou periodicky archivovány, tak aby jejich přítomnost v aktivní databázi nezpomalovala vyhledávání a velikost databáze byla kapacitně únosná.



Obrázek 3. Detail IPL části v rámci CDC

## Architektura KDC

Krajské datové centrum poskytuje z pohledu softwaru systému NIS IZS dvojí základní funkcionalitu:

1. Poskytování mapových pokladů pro NSPTV a OŘ všech složek. Tato funkcionalita je dále dělena na část zajišťující služby pro NSPTV a OŘ a část zajištující analytické služby na GISem také pro OŘ. Tyto dvě funkcionality (NSPTV+OŘ vs. analýza) jsou od sebe odděleny, tak aby nedošlo k ohrožení dostupnosti GISového řešení v rámci NSPTV a OŘ jinými časově a výkonnostně náročnými činnosti analytického charakteru.
2. Směrování informací týkajících se událostí mezi OŘ s NSPTV včetně ostrovního provozu, kdy není na kraji dostupná funkcionalita NSPTV.

Další důležitou funkcionalitou je telefonie a nahrávání (v části Telefonie). Dále zajišťuje KDC směřování operátorů na CDC a je součástí rozhodovacího mechanizmu pro přepojování operátoru při nedostupnosti aktuálního CDC.

Architektura miniIPL části KDC je v redundantním provedení a je podobně jako v CDC odolná proti pádu jedné z běžících instancí. V případě nedostupnosti funkcionality na KDC dojde k přesměrování na záložní KDC. Jednotlivé KDC jsou kapacitně navrženy tak, aby zastoupili v případě potřeby jiné KDC (detailně v Recovery scénářích).



Obrázek 4. Detail architektury KDC na úrovni logických bloků

### Architektura KDC – detail GIS

Pro lepší přehlednost je v této kapitole rozkreslena detailní struktura GISové části KDC. Oba GIS servery budou zapojeny do GIS server site (tímto je zajištěn jejich soulad a vzájemná spolupráce). Oba GIS servery sdílejí databázové připojení, uložení mapových dlaždic a taktéž na vysoce dostupném souborovém datovém úložišti (SSD disky) sdílejí konfigurační soubory “configuration store”. Výkon serverů GIS v KDC je dimenzován dostatečně tak, že není třeba load balancing. Kapacita HW části pro běh GIS serverů byla koncipována v rámci funkčního prototypu dle doporučení firmy ESRI ve virtualizovaném prostředí (http://www.vmware.com/files/pdf/techpaper/Esri-ArcGIS-Server-10-for-VMware-Infrastructure.pdf)

Pro KDC jsou navrženy dvě samostatné instance mapového serveru GIS, jejich role a způsob využití je odlišný.

**GIS I**

publikace analytických služeb pro OŘ a NSPTV,

publikace mapových služeb specifických pro KDC

geo-databáze, kterou plně kontroluje správce GIS na úrovni kraje (GIS Admin)

oddělené testovací prostředí pro krajského správce GIS

**GIS II**

publikace základních mapových služeb pro OŘ, tzn. identické se službami CDC. Obsah geo-database bude synchronizovaný s databází na CDC.

Definice služeb bude synchronizována z CDC



Obrázek 5. Detail GISové části v rámci KDC

### Architektura KDC – detail miniIPL

Jádro miniIPL na KDC tvoří opět Fuse Fabric cluster, který zabezpečuje vysokou dostupnost řešení. Obecně se miniIPL v KDC lidí od IPL v CDC tím, že miniIPL neobsahuje procesní engine a není napojena přímo na databázi události. Technologicky staví na stejném nástroji: JBoss Fuse, který je nakonfigurován na jinou funkcionalitu KDC.

MiniIPL na KDC komunikuje s telefonní ústřednou a umožňuje propojení hovoru a aplikace NSPTV operátorů. MiniIPL zpracovává požadavky na přihlášení jednotlivých operátorů, respektive vybírá CDC, na které se operátoři hlásí. Pro výběr je použitý jednoduchý algoritmus Round Robin (http://en.wikipedia.org/wiki/Round-robin\_scheduling). Tímto mechanizmem, je zaručeno, že operátoři dané složky v daném kraji nejsou v jeden okamžik připojeni na stejné CDC a snižuje se tím dopad případného výpadku konkrétního CDC na zabezpečení funkcionality příjmu tísňového volání v daném kraji.

MiniIPL také publikuje služby NSPTV jako pro příjem tísňového volání, tak pro OŘ. Stejně tak miniIPL slouží jako komunikační gateway mezi systémy NIS a KSP. Datově věty jsou směrovány přes miniIPL na KDC. Funkcionalita KDC je zastupitelná jiným KDC. Aplikace OŘ mohou v případě nedostupnosti mateřského KDC použít jiné KDC pro doručení datových vět na CDC.



Obrázek 6. Detail miniIPL části v rámci KDC

**Legenda**

* Oranžově jsou na obrázku zobrazeny jednotlivé SW řešení
* Modře jsou zobrazeny logické celky na úrovni virtualizovaných operačních systémů. Modré propoje reprezentují komunikaci mezi jednotlivými systémy.
* Zeleně jsou zobrazeny HW segmenty, které jsou důležité z pohledu komunikace mezi softwarovými celky
* Fialová a žlutá znázorňují komunikace uvnitř

## Schéma komunikace

Obrázek 7 znázorňuje komunikaci mezi CDC, KDC a OŘ v rámci jednoho kraje na úrovni protokolů.

Do KCD jsou připojeny tísňové linky z VTS, v případě nedostupnosti ITS v ostrovním režimu jsou linky směrovány z KDC na jednotlivé OS v kraji. Telefonie je zapojena do clusteru a v rámci telefonního řízení (umbrella management) jsou vyměňovány informace telefonii v krajích do centra. Telefonní část ovládá pomocí CTI telefony jednotlivých operátorů NSPTV. MiniIPL pomocí technologie obrserveru komunikuje s ústřednou na KDC a zjišťuje existenci příchozích hovorů. Problematika telefonie je detailně zpracována v samostatném dokumentu PK.

GISová část KDC komunikuje na úrovni REST protokolu s pracovišti OŘ (operační řízení a analytické úlohy, včetně administrace). Operátoři NSPTV a dispečeři OŘ jsou připojeni ke GISové části v KDC. Odkaz na GISové služby získají při přihlášení.

Aplikace NSPTV komunikuje iniciálně s miniIPL a dostává tak pokyn, ke kterému CDC se má připojit. MiniIPL dává informaci aplikaci NSPTV o stavu sítě a rozhoduje o chování v nestandardních situacích. Dále miniIPL zprostředkovává komunikaci mezi CDC a OŘ pro přenos datové věty.

V případě, že dojde k běhu systému v ostrovním režimu, tak miniIPL zajištuje komunikaci s relevantními složkami OŘ v rámci daného kraje. Po obnovení normální stavu zajišťuje miniIPL synchronizaci uložených informací o událostech při ostrovním režimu do centrální databáze v CDC. Navržený protokol pro přenos datových vět je messaging, konkrétně standard JSR 914 (http://jcp.org/en/jsr/detail?id=914).



Obrázek 7. Schéma komunikace v rámci NIS a KSP

Sekvenční diagram příjmu nové události (Obrázek 8) začíná přihlášením operátora k aplikaci, která jeho identitu ověří v centrálně vedené správě identit. Následně obdrží data o přihlášených operátorech a otevřených událostech a dalších přehledech dostupných na 3 okně aplikace. Sekvenční diagram končí přiřazením nové události operátorovi.



Obrázek 8. Přihlášení operátora

Níže uvedený sekvenční diagram konferenčního hovoru (Obrázek 9) zobrazuje jednotlivé zprávy, které si mezi sebou vyměňují účastníci komunikace při vytváření konferenčního hovoru. Z důvodu přehlednosti je zde zjednodušené navázání spojení mezi ústřednou a operátorem. Problematika příjmu hovoru je detailněji popsána na následujícím sekvenčním diagramu.



Obrázek 9. Konference dvou operátorů

## Vývojová platforma

Pro vývoj řešení byly použity nástroje, které zajišťují kontinuální uchování informací o vývoje sw. Tato řešení zamezují ztrátě informací a umožňují efektivní rozvoj řešení. Jednotlivé zadání je formalizováno do podoby „ticketů“, které jsou následně zpracovány vývojovým týmem. Výsledky vývoje jsou uloženy v systému na správu zdrojového kódu a jsou provázány se zadáním.

### Technologie podporující vývoj

Během vývoje je využito několika nástrojů podporujících vyšší kvalitu doručeného software, které pokrývají celý životní cyklus vývoje.

#### Java (http://java.oracle.com/)

Pro běh aplikace NIS ISZ je využívána Java ve verzi 7. Pro vývoj i nasazení je využívána distribuce JVM od společnosti Oracle (dříve Sun), která zastřešuje celou Java platformu.

#### GIT (http://github.com/)

Git slouží ke správě zdrojových kódů, ale i k verzování libovolného jiného typu dokumentů. Pro hostování zdrojových kódů NIS ISZ je zvoleno cloudové řešení github.com, které je široce využívané jak opensource komunitou pro veřejné projekty, tak i komerční sférou pro projekty soukromé. NIS ISZ je druhým příkladem.

Zdrojové kódy jsou komentovány a prolinkovány s funkčními i systémovými požadavky dle specifikace. Jednotlivé nasazované verze jsou tagovány, aby bylo možné sledovat přidanou funkcionalitu mezi verzemi.

#### YouTrack (http://jetbrains.com/youtrack/)

YouTrack je systém pro správu projektu, řízení vývoje a změnových požadavků. V systému je plánována práce podle iterací a požadavků ze strany zákazníka. YouTrack zároveň podporuje integraci s GitHubem, takže je možné sledovat změny v kódu přímo v založené události/chybě.

#### TeamCity (http://jetbrains.com/teamcity/)

TeamCity je serverem průběžné integrace (angl. Continuous Integration), který zajišťuje spouštění buildu po každé změně kódu v GitHubu. Po úspěšném buildu je taktéž spuštěna sada jednotkových testů, které ověřují správnou funkčnost jednotlivých komponent v systému NIS IZS. TeamCity taktéž podporuje statickou analýzu kódu a tedy nejen předcházet chybovosti aplikace, ale i analyzovat případné budoucí problémy, které neodhalí jednotkové testy. V případě označení verze jako stabilní je verze přidána do repozitáře pro nasazení v testovacím/produkčním prostředí.

#### Nexus (http://sonatype.org/nexus/)

Nexus server je repozitář buildů vytvořených během vývojového cyklu a označených jako stabilní verze.

#### Maven (http://maven.apache.org/)

Aplikace pro správu závislostí externích komponent a tvoření jednotlivých buildů. Maven je používán v TeamCity pro vytvoření buildu, ale i na vývojářských stanicích pro vytvoření experimentálních verzí. Verze označená jako stabilní je pomocí Mavenu přidávána do centrálního repozitáře Nexus.

## Testovací a ověřovací platforma

Testovací a ověřovací platforma bude sdílena a bude řešena virtualizací kopie stávající infrastruktury CDC a vybraných KDC. Tím docílíme téměř identické infrastruktury, která je předpokladem k pilotnímu provozu.

Testováním a následným ověřovacím provozem musí projít všechny updaty aplikace NSPTV stejně jako serverové částí na CDC i KDC.

## Školící platforma

Školící platforma bude provozována v odděleném virtualizovaném prostředí. Všechny události pro operátory budou simulovány. Budou vytvořeny simulace pro všechny čtyři druhy školení: operátoři, supervizoři, vedoucí složky a vedoucí IZS.

## Datová věta

Datová věta slouží k obousměrné výměně dat mezi NSPTV a krajskými OŘ všech tří zúčastněných složek. Její struktura je definována pomocí XML Schema Definiton (XSD) formátu, viz příloha (NISIZS\_PKv50\_CAST\_B\_SW\_ veta\_ornew.xsd). Pomocí datové věty jsme schopni zasílat nové události i updaty událostí. Z OŘ do NSPTV směřují informace o stavech výjezdů k jednotlivým událostem a operační situaci včetně nových událostí. Dále se datovou větou přenášení informace o stavech SaP – informace o stavu SaP se neukládají do databáze, pouze se zobrazují.

Technologicky bude komunikace realizována formou zasílání asynchronních zpráv do front hostovaných backendovou částí aplikace. Na straně NSPTV to bude IPL, na straně OŘ je tato funkcionalita součástí KSP projektů a model komunikace bude součástí další fáze projektu. Výhodou frontované asynchronní komunikace je její robustnost a snadná implementace v případě, kdy se liší příjemci na základě obsahu zpráv (různá OŘ podle nastavené součinnosti). Robustnost řešení využívá rozdílné náročnosti zpracování jedné zprávy ve prospěch asynchronní komunikace z důvodů absence okamžitého zpracování a odpovědi.

Každá zpráva bude korelována s událostí pomocí jejího UUID (http://en.wikipedia.org/wiki/Universally\_unique\_identifier).

## Datový Model

Grafické zobrazení datového modelu je uvedeno v příloze. V průběhu tohoto projektu a následného provozu bude udržován a aktualizován.

V rámci této kapitoly následuje popis datového modelu, integritních kritérií a výtah hlavních entit modelu.

Kompletní datové modely IPL a GISu jsou součástí přílohy v čitelné podobě. (NISIZS\_Provadeci\_koncept\_CAST\_B\_SW\_verze\_50\_IPL\_ERD.pdf, NISIZS\_Provadeci\_koncept\_CAST\_B\_SW\_verze\_50\_GIS\_ERD.png)

### Metodika popisu datového modelu

V rámci deskripce datového modelu byla pro popis tabulek použita následující jmenná konvence:

* + prefix TD - datová tabulka
  + prefix TC - tabulka číselníku
  + prefix TDH - historizační tabulka datové tabulky

Ústředními tabulkami modelu jsou TD\_UDALOST, TD\_MISTO a TD\_HOVOR\_OBCAN. Obsah těchto tabulek je zřejmý z přiloženého datového modelu. Pro lepší čitelnost níže přikládáme detailní popis k tabulkám a návazným číselníkům, které vybočují z průměru a mohly by tedy být bez popisu složité:

* + Tabulka TD\_UDALOST, na kterou jsou navázány tabulky číselníků definující atributy Události:
  + TC\_STAV\_UD
  + TC\_STAV\_SLOZKY
  + TC\_HZS\_TYP\_UD
  + TC\_ZZS\_TYP\_UD
  + TC\_NEODKLADNOST\_UD
  + TC\_PCR\_TYP\_UD
  + TC\_TYP\_UD
  + TC\_ZAVAZNOST\_UD
  + Číselník TC\_STAV\_SLOZKY slouží k definici tzv. agregovaných stavů celých složek IZS. Například agregovaný stav celé složky HZS je "poslední jednotka opustila místo" ve chvíli, kdy poslední zúčastněné OŘ HZS zaslalo status o opuštění Místa události.
  + Tabulka TD\_UDALOST je navázána vaznou 1:N na tabulku TD\_MISTO, jelikož Míst události může být definováno více.
  + Tabulka TD\_HOVOR\_OBCAN je na tabulku TD\_UDALOST navázána přes vazební tabulku TD\_HOVOR\_UDALOSTI, do které se kromě jiného zapisuje identita Operátora NSPTV, který hovor k dané události přiřadil.
  + Na tabulku TD\_HOVOR\_OBCAN jsou navázány tabulky číselníků definující atributy Hovoru občana:
  + TC\_VOLANE\_CISL
  + TC\_SIT\_VOLAJICIHO
  + TD\_MISTO, kde je uložena strukturovaná poloha volajícího občana
  + TC\_TYP\_HOVORU je na TD\_HOVOR\_OBCAN navázáno přes vazební tabulku TD\_TYP\_HOVORU, kde se kromě jiného zapisuje identita "subjektu", který tento typ určil. Subjektem může být buď Operátor NSPTV (v případě, že hovor byl spojen na operátora a ten určil jeho typ), nebo samotná IPL (v případě, že hovor nebyl spojen na Operátora NSPTV - Whitelist, Blacklist). V tabulce TD\_TYP\_HOVORU je pak zapsán celý životní cyklus odbavení Hovoru občana (např. IPL nejprve vyhodnotila příchozí hovor jako "blacklist s upozorněním" - tedy upozornila Operátora NSPTV, že volající byl v definované minulosti označen za zlomyslného volajícího, a po té byl Operátorem NSPTV tentýž hovor označen např. "hovor nové události").
  + Do tabulky TD\_ZLOMYSL\_HOVOR se duplicitně zapisují informace o Zlomyslných hovorech (v tabulce TD\_TYP\_HOVORU je tatáž informace). Tabulka TD\_ZLOMYSL\_HOVOR však tyto informace uchovává pouze po dobu, po kterou se na tato data může dotazovat Engine pravidel a vyhodnocovat z nich pravidla blacklistování a blacklistování s upozorněním.
  + Tabulka TD\_HOVOR\_OPERATOR slouží k záznamům hovorů Operátorů NSPTV, včetně hovorů Operátor-Operátor. Navázaný číselník TC\_UKONCENI\_HOVORU slouží k definici atributu id\_ukonceni\_hovoru, kterým je zaznamenáno, v jaké situaci a který účastník ukončil hovor, nebo opustil konferenci. Tabulka TD\_VYZADANE\_ZNALOSTI pak zaznamenává, jaké znalosti si Operátor NSPTV vyžádal, pokud žádal operátora se specifickými znalostmi (vhodné pro vyhodnocování, plánování rozšiřování řad Operátorů a plánování vzdělávání).
  + Tabulky TD\_POZNAMKA\_UDALOST a TD\_POZNAMKA\_HOVOR\_OBCAN slouží k zápisu Poznámky. Do tabulky TD\_POZNAMKA\_UDALOST se zapisují Poznámky vedené k Události a do tabulky TD\_POZNAMKA\_HOVOR\_OBCAN se zapisují poznámky k hovorům, na jejichž základě nebyla založena Událost.
  + Tabulka TD\_UCAST\_SLOZEK slouží k záznamům stavů účasti na Události jednotlivých Operačních řízení všech složek. Z těchto stavů je pak dovozován agregovaný stav celé složky IZS (číselník TC\_STAV\_SLOZKY).

## Integritní kritéria

Integritní kritéria jsou definována datovým modelem uvedeným v příloze a jejich dodržování zaručuje DB server svými vlastnostmi (Aktivní referenční integrita).

* + DB server neumožní zápis řádku, který nemá úplný primární klíč; ani zápis řádku s primárním klíčem, který je již v téže tabulce použit u jiného řádku (Entitní integritní omezení).
  + DB neumožní zápis hodnoty jiného datového typu, než je pro atribut definován (Doménové integritní omezení).
  + DB server hlídá Referenční integritní omezení a nedovolí zápis hodnoty cizího klíče, pokud neexistuje jeho hodnota v roli primárního klíče v tabulce, na kterou tento cizí klíč odkazuje.

Pro primární a cizí klíče jsou použita tato označení:

* + PK - primární klíč
  + FK - cizí klíč
  + pfK - cizí klíč sloužící jako součást složeného primárního klíče

# Technické řešení SW komponent

## NSPTV aplikace

NSPTV aplikace bude tvořena třemi hlavními okny normalizovanými na rozhraní zobrazovacího zařízení 1920:1200. První okno představuje formulář pro vyplnění údajů získaných při vytěžování hovoru, druhé okno je tvořeno GIS mapovou komponentou a třetí okno obsahuje přehledy a stavy.

Všechny tři okna jsou tvořeny aplikací, která ke svému běhu vyžaduje Java Virtual Machine (JVM). Veškerá data jsou z aplikace odesílána do IPL, která zajištuje distribuci aktualizace dat v rámci NSPTV a OŘ.

Každý operátor má své unikátní číslo v rámci NSPTV, aplikace zajistí, že po přihlášení operátora do aplikace NSPTV je operátor automaticky spřažen s terminálem u daného pracoviště a není potřeba se zvlášť hlásit k telefonnímu terminálu.

První okno aplikace je složeno z pěti hlavních grafických komponent. První komponentou je lišta záložek, ve které jsou řazeny otevřené události. Druhá komponenta obsahuje rozdělení příchozích hovorů do kategorií: Nová událost, informační hovor, zlomyslný hovor, předání výhradně pro jinou složku a FHQ. Třetí komponenta obsahuje samotný formulář, pomocí něhož jsou vytěžená data o hovoru zaznamenána a liší se podle kategorie příchozího hovoru. Čtvrtou komponentou je místopisný helper, který slouží k lokalizaci události a úzce spolupracuje s GIS komponentou. Poslední je SW telefon, který ovládá konference a zobrazuje signalizační informace od operátora.

Druhé okno představuje mapovou komponentu, pomocí které je operátor schopen lokalizovat místo události, které automaticky převezme do formuláře v prvním okně. Dále okno zobrazuje ostatní aktivní události a výsledky vyhledávání v místopisném helperu umístěném v prvním okně aplikace.

Třetí přehledové okno zobrazuje přehled všech operátorů, přehled událostí (vlastních a sdílených), telefonní seznam.

## Architektura klienta

Operátorská aplikace NSPTV je postavená na technologii Java Swing a připojuje se přes JMS a HTTP protokol na IPL servery. Umístění aplikace je složkově závislé (instalace přímo na koncové stanici nebo VDI server). Z hlediska dostupnosti služeb KDC/GIS/CDC pro aplikaci NSPTV není rozdíl mezi spuštěním z koncové stanice a VDI serveru.

Telefonie je nezávislá na typu instalace aplikace NSPTV. HW telefon fyzicky umístěný u klientské aplikace je ovládán z miniIPL přes JTAPI. Přiřazení telefonního přístroje k  operátorovi (direct number) se provádí na úrovni IPL během přihlášení operátora do aplikace.

Cílová platforma (koncová stanice nebo VDI server) musí splňovat minimálně následující parametry:

* Technologie x86
* Instalace Java 1.7.0\_45 (nebo vyšší)
* Paměť dostupná pro aplikaci NSPTV min. 2 GB
* Podpora 3 monitorů (a to včetně připojení přes VDI), rozlišení 1920 x 1200
* Výstup zvuku z aplikace
* Prostup minimálně na servery KDC (služby GIS a miniIPL) a CDC (služby IPL a nahrávací server telefonie).

### Tlustý klient

Aplikace NSPTV se bude instalovat přímo na koncové zařízení, které musí splňovat uvedené parametry.

Instalace aplikace NSPTV jako „Tlustý klient“ bude používána pro složku HZS.

### VDI server

Aplikace NSPTV se bude instalovat na VDI server. Použitý server musí splňovat uvedené parametry a umožňovat práci s více monitory na klientské stanici.

Klientská stanice vyžaduje pouze přístup na VDI server a musí zajistit připojení na záložní řešení v případě jeho výpadku.

Instalace aplikace NSPTV na VDI server bude používána pro složky ZZS a PČR.

## Integrační platforma (IPL)

Systém IPL je srdcem celého projektu NIS ISZ. Slouží jako zdroj dat pro klientské aplikace, mediátor krajských operačních řízení nebo pro správu telefonního subsystému. Dále zajišťuje perzistenci dat z NSPTV a operačního řízení. Dostupnost tohoto systému je kritická pro celkovou funkčnost NIS IZS.

V každém datovém centru existují dvě instance IPL, které jsou navzájem nezávislé a tedy při kompletním výpadku (viz Failover) jedné z nich zůstává stále funkční jedna větev. Ve standardním režimu se zátěž rozkládá mezi obě instance IPL (viz Balancing).



Obrázek 14. Detail IPL

#### JBoss Fuse (http://www.redhat.com/products/jbossenterprisemiddleware/fuse/)

JBoss Fuse je ESB (Enterprise Service Bus) implementace, která je využívaná v mnoha organizacích pro svoje vlastnosti jako je podpora clusteringu (cloudové řešení), jednoduchost konfigurace, nízká náročnost atd. JBoss Fuse integruje několik opensourcových technologií jako Apache Camel, Apache CXF, Apache Karaf, Apache ActiveMQ a další. Vývoj a podporu zaštiťuje jedna z největších společností v IT sektoru společnost RedHat. Jedna její vývojová pobočka se nalézá i v České republice, což dává záruku flexibilního řešení našich požadavků případně problémů.



Funkční komponenty JBoss Fuse jsou:

* **kontejner** – je základem JBoss Fuse. Tato komponenta je založena na Apache Karaf a dále rozšířena Fabric Management konzolí, která zjednodušuje správu velkého množství distribuovaných kontejnerů
* **integrační framework** - tato vrstva je založena na Apache Camel a umožňuje využití velkého množství standardních metod a protokolů
* **framework pro webové služby** – umožňuje proměnit aplikaci ve službu a tedy zajistit její interoperabilitu s dalšími systémy a aplikacemi. Tato vrstva využívá Apache CXF
* **spolehlivé zprávy** - vrstva zajišťující asynchronní zpracování zpráv z ostatních systémů. Tato vrstva využívá Apache ActiveMQ
* **správa a monitoring** - správa produkčního prostředí je podporována ve Fabric Management konzoli. Pro monitoring systému je možné využit JBoss Operations Network nebo jiné řešení

### Failover

V rámci zajištění vysoké dostupnosti systému je každý uzel stínován v režimu (master-slave). Slave uzel čeká na nedostupnost mastera a v případě jeho výpadku přebírá veškeré jeho služby. Pro urychlení obnovení funkčností je slave uzel připraven v tzv. "hot" režimu, kdy jsou nastartovány všechny systémové služby a v případě přebírání funkčnosti se startují pouze služby aplikace IPL.

### Balancing

Pro rozdělení zátěže je systém nasazen v clusteru (Fuse Fabric), který se stará o transparentnost celého systému. Konzistence jednotlivých uzlů je zajištěna pomocí algoritmu Quorum (http://en.wikipedia.org/wiki/Quorum\_(distributed\_computing))

#### Activiti / Drools

Activiti/Drools jsou systémy podporující byznys modelování procesů, které budou používány v rámci konfigurace a nastavení pravidel v NIS IZS.

Procesní engine spolu s Business rule enginem obsahuje veškerá vyšší logika systému, dá se tedy jednotně spravovat a případně i měnit. Business rule engine je jednou ze složek procesního enginu a je volán vždy před rozvětvením procesu, aby vyhodnotil pravidla větvení respektive chování procesu. Výhodou business rules je možnost dynamicky za chodu měnit množinu pravidel, popřípadě jejich atributy tak, aby se změnilo chování aplikace podle potřeb situace.

Procesní engine bude plnit roli orchetratora v SOA architektuře. Nasazen bude jako jedna ze služeb do aplikačního prostředí Fuse Fabric. Dle obrázku (viz níže) bude volat jednotlivé business služby poskytované integračním jádrem vybudovaným pomocí Apache Camelu.



Obrázek 15. Schéma napojení Procesního enginu na IPL

Camel jako jádro integrační platformy disponuje dvěma silnými stránkami.

1. Implementací Enterpise Integration Patterns (http://www.eaipatterns.com), které umožnují komplexní choreografii zpráv zasílaných IPL mezi jednotlivými integrovanými systémy. Hojně využívaným je např:
   1. Message router
   2. Pipes and filters 
   3. Message bus



1. Širokou škálou konektorů, které usnadňují implementaci integračního scénáře i v silně heterogenním prostředí. Pro Messaging a webové služby jsou ve Jboss Fuse dedikované frameworky (viz kap. JBoss Fuse), ostatní konektory: JDBC, filesystem, email a FTP zajišťuje Apache Camel sám. Procesní engine používá notaci BPMN 2.0, kterou lze použít jak pro popis chování procesů, tak pro samotný běh procesu, kdy je ovšem nutné jednotlivá systémová volání nakonfigurovat. Pro modelování procesů lze použít Activiti Modeller, pro konfiguraci procesů nutnou pro spuštění lze použít nástroj Activiti Designer. Další vlastností BPMN 2.0 notace je standardizovaná reprezentace v XML, která zajišťuje snadnou přenositelnost modelu mezi různými nástroji.

## MiniIPL na KDC



Obrázek 16. Detail miniIPL

V krajském datovém centru je taktéž nasazena IPL avšak ve standardním režimu funguje pouze jako mediátor do hlavních datových center. Až v případě nedostupnosti žádného centrálního datového centra přebírá plnohodnotnou funkcionalitu IPL a zajišťuje veškeré služby potřebné k provozu. K tomuto směrování požadavků na CDC se používají definice ve frameworku Apache Camel. Rozfázování směrování požadavků lze nalézt v části dokumentu Schéma komunikace.

### Ostrovní režim

V případě nedostupnosti CDC je provoz transparentně přesměrován na další CDC. V případě nedostupnosti všech CDC (ať již výpadkem datového centra nebo veškeré konektivity do KDC) se KDC přepíná do tzv. „ostrovního" režimu. V něm IPL zabezpečuje komunikaci na úrovni kraje mezi jednotlivými operačními řízeními všech složek. Zajišťuje též vzájemnou komunikaci pro všechna operační řízení. V tomto režimu nejsou ostatní KDC informována o stavu jednotlivých operátorů ani řešených událostech, ale tento stav je viditelný pouze v rámci kraje. KDC se periodicky snaží obnovit spojení na CDC. Po obnovení spojení jsou stavy operátorů a řešené události synchronizovány do CDC a ostatní KDC. Systém se tak dostává do svého standardního režimu.

## Komponenty pro centrální GIS v rámci NIS IZS

Centrální GIS na vstupu přebírá data externí (RUIAN, JSDI, ČHMÚ…) i složková. Pomocí automatizovaných nástrojů typu ETL udržuje v aktuálním stavu centrální databázi GIS. Tato databáze obsahuje prostorová i neprostorová data v optimalizovaném tvaru, kdy jsou některé vazby již předpočítány (materializované) tak aby lokalizace událostí byla co nejrychlejší a nejpřesnější. Tuto optimalizaci provádějí (kromě samotné kontroly vstupních dat a vlastního importu) nástroje ETL.

Centrální GIS poskytuje služby pro NSPTV a slouží jako referenční geo-datová základna (C-GDB) pro krajské GIS (KDC). Do geo-databáze KDC je stav C-GDB průběžně přenášen pomocí aplikační one-way replikace pomocí technologie GIS. Složková geo-databáze je v rámci centrálního GIS buď udržovaná replikací geodatabáze složkového správce GIS (pro HZS je to pracoviště IOO Lázně Bohdaneč a pro PČR je to Oddělení krizového řízení PP) nebo nástroji správce GIS (GIS ADM). Také tato geodatabáze je replikována na všechna KDC. Složková data existují pro každou složku IZS separátně.



Obrázek 17. Schéma návrhu centrálního GIS a jeho komponent

### Popis komponent centrálního GIS

* **Základní GIS technologie**
  + **GIS SRV (4)**
    - GIS server poskytující kromě dalšího
      * plnou implementaci GeoServices REST API
      * plnou implementaci OGC standardů WMS, WFS, WCS, WMTS
      * Administrátorské API (REST)
      * Databázové uložení prostorových dat (geodatabáze) nezávislé na konkrétním dodavateli DB (podporuje Oracle, MS SQL Server 2008, PostgreSQL)…
  + **GIS ADM (6)**
    - Administrátorské nástroje pro správu geodatabáze a GIS serveru na bázi výkonného desktop GIS nástroje
    - Požadavky
      * Plná integrace s technologií GIS serverGIS server
      * Přípravu map a jejich publikaci
      * Administrace služeb GIS server
  + Map cache
    - Obsahuje optimalizované mapové dlaždice s předpřipravenou základní mapovou kompozicí (společnou pro všechny složky) a optimalizované mapové dlaždice obsahující ortofotomapu
  + C-GDB
    - Geodatabáze (vektorová data pro GIS) uložená v „enterprise relační databázi“
      * Základní vrstvy dle požadavků NSPTV a společných požadavků OŘ
      * Obsah registru RUIAN optimalizovaný pro potřeby GIS, včetně reprezentace prostorových prvků
      * Vrstvy optimalizované pro vyhledávání, používané komponentou o MST Helper service
  + **Metainfo služby (5)**
    - Metainformační systém pro GIS
    - Funkčnost
      * Uložení metadatových záznamů v souladu s platnou legislativou, směrnicí INSPIRE, Metadatovým profilem ČR a normami ISO 19115, ISO 19119 a ISO 19139
      * Rozhraní katalogové služby dle standardu OGC CSW 2.0.2
      * Webová aplikace pro evidenci metadat geografických dat a služeb založených na geografických datech
      * Umožňuje vytváření a ukládání, správu, vyhledávání, zobrazování, stahování a publikování metadat
      * Podpora vydávání dat
      * Plná integrace s GIS serverem
      * Umožňuje automatizovaný harvesting (zpracováná změny dat v geodatabázi)
      * Je možné konfigurovat propojení s dalšími katalogy metadat (dle příslušných metadatových standardů)
* **Transformace dat - ETL**
  + **ETL (1)**
    - Konfigurovatelná Spatial ETL a geoprocessing technologie.
    - Požadavky-funkčnost:
      * Automatizované transformační procesy, které zpracují vstupní data a provádí kontrolu a import dat do pracovní geodatabáze
      * Transformují data do schématu databáze místopisného helperu, dopočítají položky pro fulltext vyhledávání, prostorové vazby a indexy
      * Integrace s metainformačním systémem pro GIS
      * Řízené konfigurací (obsahuje nástroj pro přípravu ETL procesů), plně automatizované
      * Provádí logování činnosti a prováděných změn
  + **RUIAN import (2)**
    - Importér dat RUIAN ve formátu VFR (pro efektivní práci GIS je nutný přímý přístup do databáze obsahující vybrané prvky registru v podobě optimalizované pro použitou technologii)
    - Požadavky-funkčnost
      * Aplikace umožňující ruční nebo automatizovaný, úplný nebo změnový import dat z registru územní identifikace, adres a nemovitostí (RUIAN). Vytváří kompletní stav registru včetně reprezentace bodových liniových a plošných geometrií ve formě nativní a optimalizované geodatabáze pro „základní GIS technologii“. Importuje a aktualizuje geometrie včetně ulic a stavebních objektů. Vytváří pohledy usnadňující práci s komplexním schématem databáze a zajišťuje podporu pro inteligentní fulltextové vyhledávání nad registrem.
      * Poskytuje uživatelské rozhraní pro „ruční aktualizaci“ a zároveň umožňuje plně automatizovaný provoz
      * Řízena konfigurací (konfigurace se připravuje pomocí uživatelského rozhraní), plně automatizované
  + **Příjem a zpracování DDR JSDI (3)**
    - Příjem a zpracování dopravních informací z JSDI včetně informací o uzavírkách. Získané informace jsou reprezentované do datového modelu C-GIS, kde slouží pak také jako doplňkový zdroj informací pro routovací služby.
    - Požadavky-funkčnost
      * Na serverové straně se udržuje stav JSDI (zpracovávají se přicházející updaty, dochází k expiraci událostí atd.). DDR JSDI je postaveno tak, že externí strana posílá informace konzumentovi (ne naopak). Na straně konzumenta musí být otevřené rozhraní (např. servlet) a potřebný port.
      * Udržovaný stav JSDI (dopravní informace) je následně prostřednictvím „GIS server“ publikován ve formě kterou může klient využívat už standardním způsobem (GeoServices REST API)
      * Plně automatizovaný provoz
* **Služby pro místopisný helper**
  + **MST Helper service (7)**
    - Konfigurovatelná služba pro vyhledávání v místopise a vyhledávacích vrstvách včetně administrátorského rozhraní
    - Požadavky-funkčnost:
      * Umožňuje vyhledávání podle atributů a vazeb, podle úplného datového modelu RUIAN
      * Přímá podpora práce s geodatabází, úzká vazba na datový model GIS
      * Vyspělé fulltextové vyhledávání
        + Podporuje prostorové omezování územní jednotkou nebo obecně geometrií
        + Podporuje synonyma, thesaurus, vyhledává na základě neúplných nebo i gramaticky nesprávných vstupů
      * Správcovské rozhraní pro údržbu synonym, výjimek a thesauru, s možností konfigurace doplňujících vyhledávacích vrstev
      * Rozhraní SOAP, REST

Proces aktualizace dat je tvořen dvěma procesy.

1) Aktualizace místopisných dat z registru RUIAN. Tento proces bude realizován pomocí komponenty RUAIN import. Vstupní data (po primárním naplnění) budou přebírány a zpracovány jako každodenní přírůstky do geo-databáze. Součástí procesu bude také aktualizace odvozených tabulek používaných pro místopisné vyhledávání.

2) Aktualizace vrstev a vyhledávacích databází daty poskytovanými správcem GIS – IOO Lázně Bohdaneč. Toto proces bude realizován pomocí ETL nástroje, kde bude na základě konfigurace (skripty, modely, konfigurace) automatizovaně provádět kontrolu vstupních dat a provádět import do produkční geo-databáze CDC. Součástí procesu bude transformace dat do schématu databáze místopisného helperu, dopočítání položek pro fulltext vyhledávání, doplnění prostorových vazeb a indexů.

## Komponenty pro GIS v rámci KDC

V rámci KDC je GIS určen pro podporu OŘ všech základních složek IZS, společná geodata (centrální a složková) přebírá z centrálního GIS pomocí aplikační replikace. GIS v rámci KDC má zároveň možnost vytvářet vlastní (místní) geodatabázi a těmito daty a službami nad nimi postavenými řešit specifické potřeby OŘ každé složky IZS v daném kraji s využitím místně dosažitelných dat.



Obrázek 18. Schéma návrhu component GIS v rámci KDC

GIS v rámci KDC je budován ze stejných komponent jako centrální GIS, s některými odlišnostmi.

Na úrovni KDC pro GIS není navržena komponenta pro import dat z RUIAN, ani JSDI – tento import probíhá na úrovni CDC a změny jsou přenášeny replikačním mechanismem.

ETL nástroj umožní, podobně jako je to navrženo pro CDC, automatizovat zpracování vstupních dat –vrstev (zde specifických pro konkrétní kraj). Administrátor GIS bude schopen pomocí ETL tyto činnosti popsat, automatizovat a zabezpečit kontrolami na správnou formu a obsah vstupních dat.

Nástroje pro administrátora (KGIS ADM) budou bohatší a určené pro aktivní práci správce krajského GIS.

* + Příprava specifických dat, mapových kompozic a služeb (správce KGIS bude mít v tomto široké možnost)
  + Příprava specifických "vyhledávácích" vrstev a jejich registrace pro MST Helper service
  + Předávání vybraných vrstev z místního geodatabáze správci složkového GIS ke kontrole a začlenění do složkové geodatabáze.
  + Administrátor bude využívat plnou funkčnost vyspělého GIS desktop nástroje, doplněného o specifickou nadstavbu

Krajské služby GIS budou rozděleny do dvou skupin.

* Služby základní (mapové služby, mapové cache, vyhledávací služby) – GIS server I.
* Služby analytické (routování, …) – GIS server II.

Aby nebylo možné provozem analytických služeb výkonnostně ovlivnit poskytování základních služeb, budou analytické služby poskytovány samostatným serverem. Tento server zároveň může sloužit pro přípravu a ověřování dalších specifických služeb, které bude připravovat podle potřeb správce GIS.

V případě přetížení serveru se základními službami bude možné provést přepnutí GIS server II využívat jako zálohu GIS server I.

GIS server je možné s využitím komponenty GIS Web Adaptor integrovat s krajským webovým serverem (který je mimo rozsah NIS IZS) a tímto způsobem řešit nastavení oprávnění přístupu ke službám pro uživatele, přebírané z místního (krajského) Active Directory.

## Databáze

Zvolené řešení je založené na PostgreSQL databázi rozšířené o replikační vrstvu. Replikační vrstva bude implementována s použitím technologie „Slony-I“. Toto řešení bylo vybráno na základě referenčního testu několika technologií.

### Replikace DB

Pro řešení replikací DB bylo zvoleno řešení založené na PostgreSQL databázi rozšířené o replikační vrstvu. Replikační vrstva bude implementována s použitím technologie xDB Replication Server v5.0 (nebo jiné s minimálně odpovídající 100% funkcionalitou a spolehlivostí) v režimu Multi-Master.

Zvolená koncepce systému plní především tyto požadavky:

* Multimaster db systém
* Plně nezávislý provoz databáze
* Datová synchronizace dat mezi jednotlivými nody
* Potenciálně replikace do jiných lokalit mimo hlavní datové centrum

Jednotlivé nody databáze jsou clustery, které pokrývají běžné výpadky HW.



Obrázek 19. Replikační schéma

**Technický detail řešení:**

* Replikace probíhá mezi jednotlivými tabulkami
* Primární klíč slouží k identifikaci daných záznamů
* Pokud by nebylo možné zvolit unikátní klíč pro daný záznam, pak by nebylo možné zajistit párování záznamů mezi DB systémy
* Vytvoření procedury pro řešení konfliktů pro každou replikovanou tabulku

**Datová izolace:**

* Replikační vrstva provádí replikaci DDL operací z jednoho Master Definition Node.

**Dopady na výkonnost databáze a odezvu:**

Z důvodu požadavku na vysokou nezávislost jednotlivých databází, byl zvolen systém využívající asynchronní replikace. Toto řešení má výhodu, oproti synchronnímu, že odezva databáze na uživatelský požadavek není závislá na dalších nodech.

Nicméně replikační vrstva přidá dodatečnou zátěž systému:

* uskladnění záznamu o replikaci (na úrovni dodatečného insertu)
* transportu dat do jiné databáze
* zpracování replikací z ostatních nodů

**Recovery scénář:**

Jednotlivé DB si ukládají seznam položek, které je nutné replikovat i stavovou informaci, které systémy již tyto data obdržely. V případě výpadku dojde k ukládání těchto informací uvnitř DB a po obnovení spojení dojde k zahájení synchronizace mezi jednotlivými databázemi. Vzhledem ke zvolené technologii asynchronní replikace jsou jednotlivé nody plně nezávislé.

**Nákladová studie pro řešení založené na EnterpriseDB:**

Tato technologie je dostupná jako open source řešení, placená je podpora zvolené technologie.

Licenční pohled:

Replikační nadstavba dodávaná společností EnterpriseDB je poskytována bez základních licenčních poplatků. SW je distribuován pod dvěma různými licencemi. Komponenty, které nejsou produktem společnosti, spadají pod opensource licenci a jimi poskytované komponenty jsou distribuované pod „EnterpriseDB Limited Use License“. Licence nebrání využívání řešení bez zakoupení podpory. V případě zakoupení SW podpory, je poté poskytována licence „EnterpriseDB Full Use License“ a podle „Subscription Terms and Conditions“.

Příklad srovnání nákladů pro jednotlivá databázová řešení pro vybranou konfiguraci:



Obrázek 20. Porovnání nákladů na replikaci DB

### Návrh lokálního fail-over řešení pro PostgreSQL databázi

Návrh řešení lokálního fail-over clusteru viz Obrázek 21. pro databázi PostgreSQL zohledňuje následující požadavky:

* Externí load balance se připojí na virtuální IP adresu služby v clusteru (VIP)
* Veškerý příchozí provoz musí být směrován na VIP tak aby bylo transparentně zajištěna dostupnost služby. Toto platí pro služby jako je datová replikace, kdy musí být zajištěno spojení na konkrétní cluster službu
* Databáze využívá failover mechanizmu pro řešení výpadku HW.
* Cluster služba musí zajistit přepnutí file systému, VIP IP adresy a jednotlivé služby (procesy) databáze a replikační vrstvy.



Obrázek 21. Model HA databázového uzlu CDC

### Návrh řešení geografické konfigurace database

* Geografické řešení vysoké dostupnosti database je zajištěno plnou multi - master replikací dat mezi jednotlivými DB clustery
* Směrování provozu bude zajištěno pomocí load balance zařízení, které se zároveň zajišťuje směrování příchozích hovorů operátora
* V případě nedostupnosti vybrané lokace, jsou data pro replikace ukládána v jednotlivých databázích a po obnovení spojení dojde k synchronizaci dat podle replikačního schématu

## Licence

### NSPTV a GIS klient NSPTV

Budou dodány jako multilicence vč. Majetkových autorských práv (viz kap. 07. Čl. 9. Smlouvy o dílo na dodávku pro projekt NIS IZS).

### OS – Operační systém

Budou pořízeny OS dvojího typu – pro servery po běh IPL, GIS, TEL a serverové části NSPTV. Konkrétní operační systém a počet licencí bude zřejmý po realizaci výběrového řízení.

### DBMS – Databáze

Specifikace: PostgreSQL verze 9

Forma licencování: BSD/MIT

### BIRT - Business Intelligence and Reporting Tools

Specifikace: BIRT 4.3.0

Forma licencování: Eclipse

### JBoss Fuse – IPL

Integrační platforma postavená na produktu JBoss Fuse.

Specifikace: Fuse

Forma licencování: Apache License Version 2.0

### ACTIVITI - BPM systém

Procesní vrstva je postavena na Business Process Management (BPM) řešení Activiti (<http://www.activiti.org/> ). Toto řešení umožňuje jak samotný proces namodelovat ve standardu BPMN 2.0 za pomocí nástroje Activiti Modeler, tak jej přímo z tohoto modelu spustit a propojit s dalšími komponentami řešení.

Specifikace: Activiti 5

Forma licencování: Apache License Version 2.0

### DROOLS - BR systém

V rámci procesní vrstvy je začleněna z pohledu BPM stěžejní komponenta, kterou je Business Rule Engine. V navrženém řešení jsme zvolili produkt Drools (<http://www.jboss.org/drools/>). Business Rule Engine umožňuje za běhu měnit nastavení stěžejních rozhodovacích kroků v procesech dle nastavení systému.

Specifikace: Drools 5/6

Forma licencování: Apache License Version 2.0

### MYBATIS - Objektově relační mapování

Aplikační vrstva je napojená na databázi pomoci Object-relational Mapping (ORM) na databázi, pro toto mapovaní, je využito řešení MyBatis (<http://mybatis.github.io/mybatis-3/>)

Specifikace: MyBatis verze 3

Forma licencování: Apache License Version 2.0

### GISSRV – GIS server

Serverové řešení („GIS server“) poskytující plnou implementaci GeoServices REST API (OGC Candidate), plnou implementaci OGC standardů WMS, WFS, WCS, administrátorské API (REST), databázové uložení prostorových dat (geodatabáze) nezávislé na konkrétním dodavateli DB (musí podporovat minimálně Oracle, MS SQL Server 2008 a Postgre SQL).

Počet licencí: 2 pro 1 KDC (1 licence je na 4 fyzické CPU)

Počet licencí celkem: 34

### GISADM – Administrátorský nástroj GIS

Administrátorské nástroje pro správu geodatabáze a GIS serveru s plnou integrací s technologií GIS serveru umožňující přípravu map a jejich publikaci a administrace služeb GIS Serveru.

Technologie vytvořená nad standardní funkčností „krabicového“ desktop GIS nástroje v podobě aplikační nadstavby bude dodaná včetně zdrojových kódů a v licenčním modelu multilicence (viz čl. 2.8.1. tohoto dokumentu).

Počet licencí: 1 pro 1 KDC

Počet licencí celkem: 14

### METAGIS - Metainformační systém pro GIS

Zajišťuje uložení metadatových záznamů v souladu s platnou legislativou, směrnicí INSPIRE, Metadatovým profilem ČR a normami ISO 19115, ISO 19119 a ISO 19139, poskytuje rozhraní katalogové služby dle standardu OGC CSW 2.0.2. Obsahuje webové aplikace pro evidenci metadat geografických dat a služeb založených na geografických datech. Umožňuje vytváření a ukládání, správu, vyhledávání, zobrazování, stahování a publikování metadat, podporuje vydávání dat. Zaručuje plnou integraci s GIS serverem. Umožňuje automatizovaný harvesting (zpracovává změny dat v geodatabázi). Je realizován technologii Java a jeho zdrojové kódy jsou bez licenčních omezení.

Specifikace: včetně zdrojových kódů

Počet licencí celkem: 1 pro neomezený počet uživatelů

### ETL - Transformace dat

Automatizované transformační procesy, které zpracují vstupní data a provádí kontrolu a import dat do pracovní geodatabáze, transformují data do schématu databáze místopisného helperu, dopočítají položky pro fulltext vyhledávání, prostorové vazby a indexy. Zaručuje plnou integraci s metainformačním systémem pro GIS. Má řízené konfigurací (obsahuje nástroj pro přípravu ETL procesů), plně automatizované. Je realizován v technologii Java a zdrojové kódy jsou bez licenčních omezení.

Specifikace: včetně zdrojových kódů

Počet licencí celkem: 1 pro neomezený počet uživatelů

### VFR - RUIAN import

Importér dat RUIAN ve formátu VFR. Aplikace umožňuje ruční nebo automatizovaný, úplný nebo změnový import dat z registru územní identifikace, adres a nemovitostí (RUIAN). Vytváří kompletní stav registru včetně reprezentace bodových liniových a plošných geometrií ve formě geodatabáze nativní a optimalizované pro „základní GIS technologii“. Importuje a aktualizuje geometrie včetně ulic a stavebních objektů. Vytváří pohledy usnadňující práci s komplexním schématem databáze a zajišťuje podporu pro inteligentní fulltextové vyhledávání nad registrem. Poskytuje uživatelské rozhraní pro „ruční aktualizaci“ a zároveň umožňuje plně automatizovaný provoz. Má řízenou konfigurací (konfigurace se připravuje pomocí uživatelského rozhraní), plně automatizované. Je realizován v technologii Java a zdrojové kódy jsou bez licenčních omezení.

Specifikace: včetně zdrojových kódů

Počet licencí celkem: 1 pro neomezený počet uživatelů

### Mapové podklady CEDA

V rámci projektu je počítáno s nákupem multilicencí pro složky IZS, tak aby bylo umožněno tyto mapové podklady pro potřeby NSPTV a OŘ.

### JSDI - Příjem a zpracování DDR JSDI

Na serverové straně se udržuje stav JSDI (zpracovávají se přicházející updaty, dochází k expiraci událostí atd.). DDR JSDI je postaveno tak, že externí strana posílá informace konzumentovi (ne naopak). Na straně konzumenta musí být otevřené rozhraní (např. servlet) a potřebný port. Udržovaný stav JSDI (dopravní informace) je následně prostřednictvím „GIS server“ publikován ve formě kterou může klient využívat už standardním způsobem (GeoServices REST API). Umožňuje plně automatizovaný provoz. Je realizován v technologii Java a zdrojové kódy jsou bez licenčních omezení.

Specifikace: včetně zdrojových kódů

Počet licencí celkem: 1 pro neomezený počet uživatelů

### HELPER - MST Helper service

Konfigurovatelná služba pro vyhledávání v místopise a vyhledávacích vrstvách včetně administrátorského rozhraní. Umožňuje vyhledávání podle atributů a vazeb, podle úplného datového modelu RUIAN. Má přímou podporu práce s geodatabází a úzkou vazbu na datový model GIS. Zajišťuje vyspělé fulltextové vyhledávání. Podporuje prostorové omezování územní jednotkou nebo obecně geometrií. Podporuje synonyma, thesaurus, vyhledává na základě neúplných nebo i gramaticky nesprávných vstupů. Má správcovské rozhraní pro údržbu synonym, výjimek a thesauru, s možností konfigurace doplňujících vyhledávacích vrstev. Rozhraní SOAP, REST.

Specifikace: včetně zdrojových kódů

Počet licencí celkem: 1 pro neomezený počet uživatelů

## Reporting

Reporting bude zastřešovat nástroj Eclipe BIRT (http://www.eclipse.org/birt/), který je složen ze dvou hlavních částí: Report designer, který slouží k navrhování reportů a report engine, který reporty generuje ze šablon z připojených dat. BIRT nabízí kromě vlastních standardních konektorů (Web services, XML, JDBC) také možnost psaní vlastních konektorů, čímž umožní zahrnout pod jeden reporting celé NIS-IZS a déle ten reporting rozvíjet s tím, jak se bude rozvíjet aplikace NIS-IZS. Nedílnou součástí reportovacího řešení je těž transformace dat a logika jejich zpracování.

## Reference na produkty

### JBoss Fuse

Fuse ESB je založeno na běžně používaných standardech, čímž zajišťuje snadnou rozšiřitelnost či napojení na řešení třetích stran. Celý balík Fuse ESB je složen z několika rozšířených frameworků, které jsou základem funkční SOA architektury.

Sabre Holdings uspěšně nasadil Jboss Fuse v roce 2013 aby čelil návalu půl milionu transakcí za den pomocí řešení, které je kompatibilní se servisně orientovanou architekturou.

### Activiti

Activiti má téměř 10 letou historii (dříve jako jBPM) a patří k nejvyspělejším BPM enginům. Jeho vyspělost dokládá i několik derivátů jako je stávající jBPM v5, Kamunda a Effektiv. Activiti má dlouhodobě silné vazby na Alfresco, které tento engine využívá ve svém ECM řešení pro workflow.

Momentálně není na trhu tak verzatilní engine jako Activiti, který lze programově ovládat. Většina dnešních řešení je zaobalena spoustou dalších nástrojů, které znesnadňují hlubší integraci s výše popsaným Fuse ESB.

### PostgreSQL

PostgreSQL je objektově-relační databázový systém (ORDBMS). Vydáván je pod licencí typu MIT a tudíž se jedná o free a open source software. Společnost EnterpriseDB vyvíjí Postgres Plus což je plnohodnotná distribuce PostgreSQL, zahrnující mnoho contrib modulů a doplňkových komponent. Postgres Plus je dostupný ve dvou verzích: Postgres Plus Standard Server, který má všechny vlastnosti PostgreSQL, je dodatečně QA testován, obsahuje integrované komponenty, možnosti ladění a jednoduchou instalaci, a Postgres Plus Advanced Server, který kromě vlastností produktu Postgres Standard Server navíc disponuje i kompatibilitou s Oracle, vlastnostmi pro zlepšení škálovatelnosti a nástroji pro DBA a vývojáře.

Referenční nasazení PostgreSQL:

* Yahoo - provozuje řešení pro analýzu chování uživatelů na webu, obsahuje dva petabyty dat
* MySpace - populární sociální síť, používá Aster nCluster Database pro data warehousing, který je postavený na nemodifikované verzi PostgreSQL
* BASF - nákupní platforma pro jejich zemědělský portál
* Mezi další významné uživatele patří např. Cisco, Fujitsu nebo NTT Data.

## Open source

Výhoda využití open source spočívá zejména v možnosti upravit funkcionalitu podle potřeby projektu díky dostupnému zdrojovému kódu. Komunita je schopna sebeposilujícím mechanizmem vytřídit nejúspěšnější projekty.

Open source model přináší i možnost rychlé opravy v produktu a i možnost backportování této opravy do starší verze produktu (menší riziko - není nutné přejít na novější verzi, u které může být odlišná funkcionalita).

Snížená pořizovací cena, při stejně kvalitní podpoře díky business partnerům, kteří nabízejí konzultace i hotová řešení využívající OSS. Díky komerční podpoře (placené služby) OSS produktů je poskytnutá podpora kvalitní srovnatelně s komerčními produkty (placená je licence i podpora).

Z pohledu bezpečnosti přináší open source model větší kontrolu nad případnými backdoory oproti proprietárnímu řešení, kdy je dodáván pouze binární kód.

## Koncept provozu

### Instalace serverové části

Předpokládáme nainstalovaný operační systém.

#### Postup instalace

* Získání aktuální verze JBossFuse z http://www.jboss.org/products/fuse
* Proběhne instalace dle https://access.redhat.com/site/documentation/en-US/JBoss\_Fuse/6.0/html-single/Installation\_Guide/index.html
* Následující kroky budou probíhat podle manuálu https://access.redhat.com/site/documentation/en-US/JBoss\_Fuse/6.0/html-single/Configuring\_and\_Running\_JBoss\_Fuse/index.html
* Konfigurace a spuštění služby (kapitola 2)
* V případě že se jedná o první uzel, vytvoří se nový Fuse Fabric (kapitola 6)
* V opačném případě je uzel přidán do již existujícího Fuse Fabric (kapitola 7)
* Nasazení IPL systému a závislých komponent podle https://access.redhat.com/site/documentation/en-US/JBoss\_Fuse/6.0/html-single/Deploying\_into\_the\_Container/index.html

### Instalace klientské aplikace

Instalace klientské aplikace bude realizována pomocí síťové instalace (tlustý klient i VDI server) prováděné standardními prostředky cílové platformy.

Na cílovém OS je nutné mít nainstalovánu Java JRE 1.7.0\_45 nebo vyšší (Java Runtime Environment). Aplikace NSPTV se spouští zástupcem z plochy OS.

## Aktualizace

### Aktualizace serverové části

Aplikace nahraná v JBoss Fuse kontejneru je kontrolována Fabric agentem. Agent prozkoumává profily aplikované na kontejner, aby rozhodl, jaké verze aplikace má načítat. Update typicky obsahuje novou verzi jedné nebo více aplikací, proto je třeba upravit profil aplikovaný na tento kontejner. To způsobí načtení nové verze agentem. Doporučeným nástrojem pro update je využití Management console.

#### Postup aktualizace

* update (systémový nebo aplikační) je nahrán do maven repozitáře
* pomocí Management console je upraven profil verzí
* je zmigrován jeden nebo dva kontejnery pro ověření správné funkčnosti
* následně je zmigrován celý cluster

### Aktualizace klientské aplikace

#### Aktualizace OS

Pro vzdálenou aktualizaci operačního systému klientské stanice a VDI budou využity standardní nástroje systému dle výsledku VŘ.

#### Aktualizace aplikace NSPTV

Pro instalaci nové verze aplikace je potřeba aktualizovat instalační složku s aplikací NSPTV novými soubory. Tento proces je nutné aplikovat pomocí standardního distribučního systému cílového systému dle výsledku VŘ..

## Monitoring

### Monitoring serverové části

JBoss Fuse nabízí pro správu a metriky svého použití přístup přes JMX (Java Management Extensions) lze tedy použít již existující aplikace (JConsole, MC4J), které lze pomocí pluginů rozšířit o vlastní pohledy, použít aplikaci Fuse HQ (komerční, monitoring včetně vlastního webového rozhraní), JBoss ON (JBoss Operations Network) nebo vytvořit vlastní monitorovací nástroj.

Bude se monitorovat aktuální stav jednotlivých kontejnerů, jejich vytížení a spotřeba prostředků.

### Monitoring koncových pracovišť

Operátorské terminály se monitorují pomocí observerů definovaných v JTAPI. Aplikace NSPTV neustále monitoruje stavy všech terminálů a pracovišť, ke kterým jsou přihlášeni operátoři. Zajišťuje tak, aby žádná nová událost nebyla směrována na pracoviště, kde není přihlášený operátor a telefon je schopen přijímat hovory.

Kromě indikace koncových pracovišť, které jsou schopny přijímat hovory, systém při každém přihlášení koncového pracoviště k IPL monitoruje aktuální verzi klienta.

## Release a Incident Management

Nové verze a aktualizace aplikace NSPTV budou v předem oznámených intervalech za součinnosti zadavatele. Při nasazování nových verzí budou operátoři přehlášení z CDC, které je aktualizováno a další CDC. Po nasazení aktuální verze NSPTV na CDC bude CDC zapojeno do provozu a instalace proběhne analogicky na zbývajících CDC.

Při aktualizaci klientské části NSPTV budou operátoři postupně odstaveni, dle plánu aktualizace, od koncových pracovišť tak, aby nedošlo k ohrožení funkcionality příjmu tísňového volání. Aktualizace bude probíhat v dohodnutých časových intervalech se zadavatelem. Aktualizaci bude možné v případě rozhodnutí zadavatele přerušit a obnovit standardní provoz.

V případě výskytu chyby v aplikaci může operátor chybu zaznamenat do helpdeskového systému. Vedoucí pracoviště následně rozhodne o validaci požadavku na opravu a dle definovaného workflow dojde k opravení chyby a nasazení nové verze.