Říjen 2018

Integrační standardy

Pro:

Magistrát hlavního města Prahy

IBM Česká republika, spol. s r.o.
V Parku 2294/4, The Park, 148 00 Praha 4 - Chodov

Historie dokumentu

Datum	Autor	Verze	Popis změny
30.10.2018	Jiří Melichna	0.01	Založení dokumentu a iniciální naplnění
13.11.2018	Jiří Melichna	1.00	Vytvoření první verze

Obsah

2. Seznam použitých termínů a zkratek

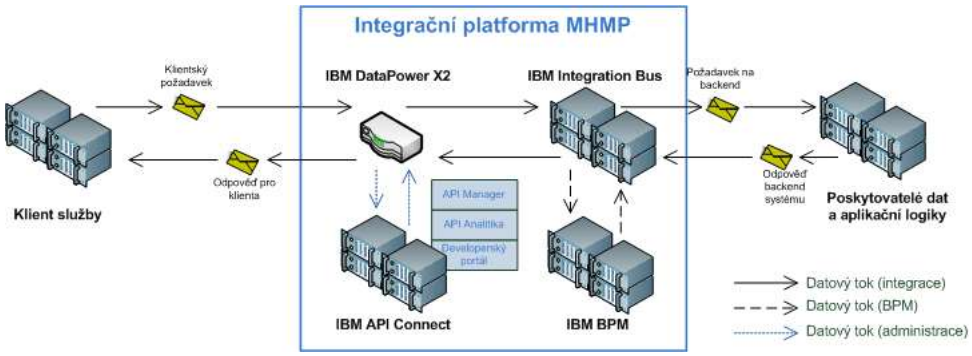
Termín / Zkratka	Vysvětlení
MHMP	**M**agistrát **h**lavního **m**ěsta **P**rahy
SOA	**S**ervice **O**riented **A**rchitecture – architektura orientovaná na služby. Jedná se o architektonický styl, kdy je snaha rozdělit na venek informační systémy z pohledu poskytovaných dat a aplikační logiky do služeb, které mohou být následně opakovaně použity
API	**A**pplication **P**rogramming **I**nterface – definované a dokumentované rozhraní komponenty určené pro vývoj systémů
IIB	**I**BM **I**ntegration **B**us
BPM	**B**usiness **P**rocess **M**anagement
ESB	**E**nterprise **S**ervice **B**us - sběrnice služeb, místo, kde se realizuje integrační logika propojení systémů
ETL	**E**xtract **T**ransform **L**oad – nástroje pro dávkové přenosy a zpracování dat vhodné
SLA	**S**ervice **L**evel **A**greement – SLA ve smyslu tohoto dokumentu představuje do značné míry tzv. nefunkční požadavky na systém jako je jeho výkonnost a dostupnost
QoS	**Q**uality **of S**ervice – řízení datového toku mezi systémy

3. Úvod

Dokument popisuje integrační standardy, které budou použity pro integrace jednotlivých informačních systémů přes integrační vrstvu Magistrátu hlavního města Prahy (zkráceně MHMP).
Dokument navazuje na dokument *MHMP – Rámcové integrační standardy*.

4. Představení integrační platformy MHMP

Integrační platforma MHMP je postavena na technologiích IBM. Jádrem integrační platformy je *IBM Integration Bus v10* (zkráceně *IIB*), který provádí případnou integrační logiku. Hlavním přístupovým bodem k integrační platformě z pohledu klientů služeb je *API Gateway* na technologii *IBM DataPower Gateway X2*, která zajišťuje bezpečný přístup klientských informačních systémů ke službám poskytovaným na integrační platformě. Pokud je součástí integrace systémů i spolupráce s koncovými uživateli (například úpravy dat), poskytuje integrační platforma také procesní vrstvu *IBM Business Process Management* (zkráceně IBM BPM). Na následujícím obrázku je naznačen koncept použití integrační platformy MHMP:



4.1 Vlastnosti integrační platformy

Integrační platforma MHMP nabízí následující funkcionality:

- Integrační vrstvu na principech ESB realizovanou na technologii *IBM Integration Bus* včetně podpory orchestrace více rozhraní backend systémů
- *API Gateway* na technologii *IBM DataPower Gateway X2*, která zpřístupňuje vystavené služby klientským systémům
- Procesní vrstvu pro komplexní integrace zahrnující lidské aktivity realizovanou na technologii *IBM Business Process Manager*
- Management vrstvu *API Manager* pro governance API a služeb
- Developerský portál, kde mohou vývojáři informačních systémů vyhledávat dostupné integrační služby a evidovat své aplikace jako konzumenty

Z pohledu formátu přenášených zpráv jsou primárně podporovány:

- SOAP 1.1, případně SOAP 1.2
  - Pro přenos binárních dat (například dokumentů) bude použit standard MTOM
- XML
- JSON
- CSV, TXT (pevná struktura)

Pro popis kontraktů služeb a rozhraní budou využity:

- WSDL 1.1 pro popis SOAP služeb
- XSD – XML Schema 1.1 pro popis datových objektů
- Swagger 2.0 pro popis REST rozhraní

Pro komunikaci s integrační platformou mohou být využity primárně následující protokoly:

- HTTP 1.1 přes TLS
- IBM MQ, JMS, AMQP, MQTT
- SMTP, IMAP, POP3
- ODBC, JDBC

Doplňkové standardy webových služeb:

- WS-Security 1.1

- WS-Addressing (pro asynchronní komunikace)
- WS-ReliableMessaging 1.2

Integrační platforma poskytuje:

- Transformaci obsahu zpráv (např. JSON <-> SOAP, SOAP <-> CSV…) nebo pro transformaci (přizpůsobení) datového modelu
    - Včetně obohacování zpráv nebo transformací například číselníkových hodnot při přizpůsobování datového modelu mezi jednotlivými informačními systémy
- Transformaci komunikačních protokolů (https <-> JMS, FTP <-> HTTP…)
- Směrování zpráv na základě komunikačních hlaviček nebo obsahu zpráv (tzv. Content Based Routing)
- Orchestraci více poskytovatelů služeb
- Možnost asynchronní komunikace přes fronty a témata (Topic) pro přizpůsobení SLA jednotlivých informačních systémů
- Logování a monitoring transakcí
- Vynucení QoS pro integrační služby pro ochranu systémů, které poskytují svou aplikační logiku a data na integrační vrstvu
- Cache dat typu klíč - hodnota

5. Pravidla integrace

Integrační platforma vystavuje své služby konzumentům (klientské informační systémy). Sama integrační vrstva je však závislá na poskytujících informačních systémech, které poskytují aplikační logiku a data klientským informačním systémům. Aby integrace systémů správně fungovala, musí všechny strany ctít určité zásady – pravidla integrace.

5.1 Základní doporučení pro výběr integračních scénářů pro Integrační platformu

Integrační služby na integrační platformě jsou určeny k propojení systémů. Základním smyslem zavedení integrační vrstvy je izolace systémů (izolace při změnách nebo z pohledu rozdílných SLA jednotlivých systémů) a pokud možno vystavovat služby, které budou opakovatelně použité v klientských informačních systémech. Základním cílem využití integrační platformy je tedy zpřístupnění vybrané aplikační logiky a dat jednoho (poskytujícího) informačního systému návazným systémům přes dokumentované rozhraní s jasným SLA. Obecná doporučení pro integrační scénáře pro integrační platformu:

- Integrace v rámci prostředí MHMP by měla procházet přes Integrační vrstvu
- Integrační platforma je primárně orientována na přenos zpráv (různé formáty i komunikační protokoly) synchronním nebo asynchronním způsobem
- Integrační platforma díky komponentě *API Gateway* může vytvářet bezpečnou bránu pro připojení systémů vně prostředí MHMP
- Integrační platforma se zaměřuje na on-line nebo téměř on-line komunikace

- Integrační platforma by neměla nahrazovat nástroje pro ETL – nástroje pro dávkové přenosy a zpracování dat pro plnění datových skladů nebo manažerských informačních systémů a podobných
- Integrační platforma není určená pro migrace dat

5.2 Postup při návrhu integračního rozhraní

1. Při návrhu integračního rozhraní většinou vznikají požadavky na data a aplikační logiku na straně nového nebo měněného informačního systému (např. ve vazbě na změny legislativy). Na základě těchto funkčních požadavků je pak možno identifikovat systémy, které požadovanou aplikační logiku a data poskytnou. Výsledkem analýzy je datový model rozhraní integračních služeb a případně požadavky na mapování při přizpůsobování datových modelů mezi doménami a informačními systémy (teoreticky může mít jiný pohled na osobu systém, který řeší oblast personalistiky a jiný pohled má agendový systém).
2. Dále je potřeba získat a posoudit nefunkční požadavky, a to zejména:
   - četnost volání integračních služeb (výkonnostní pohled a stabilita systémů)
   - dostupnost integrovaných systémů
   - bezpečnostního model integrovaných systémů – například nutnost propagace uživatelské identity
3. Pokus o vyhledání existujících integračních služeb na základě sebraných požadavků. Výsledkem je vyhodnocení možnosti použít existující integrační rozhraní nebo návrh nového (případně změna – verzování existujícího rozhraní).
4. Při posouzení nefunkčních parametrů se navrhne použití správných integračních vzorů (pattern) a identifikují se tak případné doplňkové komponenty jako dočasná datová úložiště nebo nutnost lidské interakce vedoucí na použití procesní vrstvy v BPM.
5. Pokud se realizuje nové rozhraní, nebo se mění rozhraní stávající, je vhodné udělat výhledovou analýzu přepoužitelnosti integračního rozhraní – zamyšlení, zda by jiný systém také mohl využít data nebo aplikační logiku, pokud by došlo k malému rozšíření implementace (například mírné rozšíření datového modelu služby).
6. Vytvoření detailního návrhu realizace integračního rozhraní.
7. Je-li vyvíjena nová verze rozhraní je potřeba navrhnout správné využití již existující verze (ponechání nebo migrace klientů rozhraní).
8. **Schválení nového nebo změněného integračního rozhraní na MHMP v rámci SOA kompetenčního centra/enterprise architektury**.

5.3 Postup při konzumaci integračního rozhraní na integrační platformě

Postup při konzumaci integračních rozhraní je zajímavý pro architekty, analytiky a vývojáře informačních systémů. Tento základní přístup je obecně platný pro nejběžnější principy integrace, tedy použití webových služeb (SOAP a REST):

1. Odpovědná osoba dodavatele se jako uživatel zaregistruje do Developer portálu MHMP a stane se tak správcem developerské organizace v prostředí MHMP.

2. Následně správce developerské organizace pozve své vývojáře informačního systému do Developer portálu.
3. Vývojáři informačního systému zaevidují svůj informační systém v Developer portálu. Výsledkem je jednoznačný identifikátor aplikace – tzv. client_id.
4. Vývojáři informačního systému si zaregistrují použití integračního rozhraní pro svůj registrovaný informační systém. **V některých případech může být vyžadováno schválení registrace SOA kompetenčním centrem MHMP**.
5. Po registraci integračního rozhraní pro informační systém mohou vývojáři následně používat toto rozhraní v testovacím prostředí.
6. Při volání integračních rozhraní z integrovaného systému je každý požadavek doplněn:
   - o hodnotu client_id – pro SOAP služby je client_id odesíláno jako query parametr v URL
   - o unikátní identifikátor transakce (požadavku) – tato hodnota je důležitá v případě, že selže volání služby a je potřeba řešit problém v rámci technické podpory
7. Po otestování informačního systému v testovacím prostředí zažádá vývojář informačního systému o jeho přechod do provozního prostředí.
8. **Odpovědná osoba v SOA kompetenčním centru MHMP schválí přechod do produkčního prostředí.**
9. Informační systém může volat integrační rozhraní v produkčním prostředí.

5.4 Postup při tvorbě rozhraní pro integrační platformu

Postup, který by měli aplikovat tvůrci rozhraní na straně informačního systému, které volá integrační platforma (v při aplikačním vývoji na míru):

1. Vývojář rozhraní si nejen vlastní aplikační logiku, ale zapracuje také podporu pro převzetí a logování identifikátoru transakce, který předá integrační logika z integrační platformy. Do budoucna tak bude usnadněna komunikace mezi týmy při podpoře informačních systému MHMP.

5.5 Jmenné konvence

Jmenné konvence jsou z větší části převzaty z dokumentu *MHMP – Rámcové integrační standardy*.

Návrh jmenných konvencí bude upřesněn v průběhu implementace IP jakožto součást podrobných integračních standardů. Veškeré názvy služeb, operací, atributů apod. budou uvedeny v českém jazyce, příp. je-li to vhodné v anglickém jazyce.

5.5.1 Jméno služby

- Jméno služby je unikátní, mělo by být vytvořeno na základě jejího účelu a musí být nezávislé na poskytovateli a konzumentovi služby.
- Notace Upper-CamelCase (CamelCase notace s velkým počátečním písmenem prvního slova).

5.5.2 Název operace

- Jméno operace musí být unikátní v rámci služby.
- Notace Lower-CamelCase (camelCase notace s malým počátečním písmenem prvního slova).
- Nejčastěji se skládá ze slovesa a podstatného jména.

5.5.3 Namespace služby

Namespace služby (targetNamespace) vzniká složením následujících částí:

- prefix,
- doména určující oblast, do které služba patří (např. ekonomika apod.),
- jméno služby,
- verze služby

5.5.4 Datové elementy

- Elementy (publikované root elementy) – Upper-CamelCase notace.
- Elementy (uvnitř definice typů) – Lower-CamelCase notace.
- Komplexní typy – Upper-CamelCase notace, končí sufixem „Type“.
- Request – Lower-CamelCase notace, končí sufixem „Request“.
- Response – Lower-CamelCase notace, končí sufixem „Response“.
- Fault – Lower-CamelCase notace, končí sufixem „Fault“.

5.5.5 Datový model

Datový model rozhraní služby musí vycházet ze jmenných konvencí. Všechny nově vznikající webové služby musí používat společný datový model zpráv, který bude upřesněn v průběhu implementace IP. Datový model definuje vstupní (request), výstupní (response) a chybové (fault) zprávy webových služeb. Každá request/response/fault zpráva obsahuje stejnou hlavičku requestHeader/responseHeader/faultHeader a dále komplexní datový typ requestBody/responseBody/faultBody, který obsahuje samotný obsah zprávy specifický pro každou službu a její operaci. Hlavička je obsažena i v chybové fault odpovědi z důvodu jednotného logování.

5.5.6 Hlavička zprávy

Komplexní datový typ requestHeader/responseHeader/faultHeader bude obsahovat minimálně tyto elementy:

- unikátní ID zprávy (request, response i fault zprávy mají svá různá ID),
- korelační ID (všechny zprávy v řetězci jednoho volání mají stejné korelační ID),
- časové razítko zprávy, které označuje čas odeslání zprávy klientem (response a fault header pak obsahují čas odeslání odpovědi, resp. čas vygenerování chyby),
- zdrojový systém, který vytváří volání webové služby (unikátní označení systému/aplikace) – evidenci zajišťuje MHMP (SOA kompetenční centrum/Enterprise Architecture),
- fyzický zdroj (FQDN stroje, IP adresa),

- fyzický cíl (FQDN stroje, IP adresa) – do response hlavičky se uvede původní fyzický zdroj.

5.5.7 Datové elementy

- Všechny elementy MUSÍ být povinně definovány jako „qualified“.
- Všechny jednoduché datové typy s omezením by měly být definovány jako „xsd:simpleType“ v root elementu schématu.
- Všechny komplexní datové typy musí být povinně definovány jako „xsd:complexType“ v root elementu schématu.

5.5.8 Chybové odpovědi

Chybové odpovědi mohou být trojího druhu (jiné typy odpovědí nejsou povoleny):

- BusinessLogicFault – v případě chyby vzniklé uvnitř business logiky integrovaného systému/aplikace (např. záznam nenalezen) – musí se jednat o chybu definovanou (rozpoznatelnou) na aplikační logice backend systému.
- SecurityFault – v případě porušení bezpečnosti (např. během autentizace nebo autorizace).
- SystemFault – v případě výskytu systémové chyby (tj. žádný z výše uvedených typů).

Chybová odpověď bude mít následující strukturu:

- typ chyby (viz výše),
- číselný kód chyby – kódy bude definovat/schvalovat MHMP (SOA kompetenční centrum/Enterprise Architecture),
- textový popis chyby,
- příčinu chyby (detail)
  - odkaz na chybu, která je původcem vyvolání výjimky.
  - ID transakce – některé komunikační SOAP knihovny mají problém se SOAP hlavičkami v chybové (fault) odpovědi

Všechny chybové (fault) odpovědi od všech služeb vystavených na IP budou mít jeden společný namespace. Konkrétní namespace bude definován v průběhu implementace IP.

5.6 Validace zpráv

IP bude umožňuje validaci zpráv proti XML Schématu (WSDL a XSD) požadavků i odpovědí. Nastavení validací (zapnutí/vypnutí validace) je možné pro jednotlivá prostředí IP (vývojové, testovací, produkční) na úrovni konkrétní aplikace, která poskytuje definovanou sadu integračních služeb.

5.7 Zabezpečení služeb

Volání služeb bude primárně zabezpečeno na úrovni transportní vrstvy (HTTPS). Bude využit minimálně protokol TLS 1.1. Budou využívány TLS certifikáty v režimu:

- jednocestné (one-way) autentizace

- dvoucestné (two-way, mutual) autentizace – preferovaná varianta pro vytvoření důvěryhodného komunikačního kanálu mezi systémy

Dále bude komunikace zabezpečena na úrovni SOAP buď jako Basic authentication nebo jako Client cert authentication.
Při Basic authentication bude konzument při komunikaci s IP vždy v rámci SOAP hlavičky posílat jméno a heslo uživatele. Autentizace proběhne proti MS Active Directory.
Při Client cert authentication bude systém komunikovat s IP za použití certifikátu (namísto jména a hesla).
SSL certifikáty (serverové i klientské) bude vydávat MHMP prostřednictvím své interní certifikační autority a prostřednictvím technického garanta za každý systém/aplikaci poskytující službu. Technický garant bude dále kontrolovat exspiraci vydaných certifikátů.

5.8 Verzování

5.8.1 Verzování služeb

Pro účely odlišení jednotlivých verzí jedné služby se bude používat trojice čísel, které dohromady tvoří jednoznačné označení konkrétní verze:

- <Major> – změna v čísle znamená nekompatibilní změnu rozhraní s předchozí verzí služby (např. odebrání operací či atributů, změna namespace, změna datových typů apod.). Major verze služby je také součástí URL služby
- <Minor> – změna v čísle znamená kompatibilní změnu rozhraní s předchozí verzí služby (např. přidání operací, přidání nového datového typu apod.).
- <Micro> – změna v čísle nepředstavuje žádnou změnu rozhraní oproti předchozí verzi služby, ale jen menší implementační úpravu (např. oprava chyb, nastavení zabezpečení služby apod.). Číslo verze odpovídá číslu buildu.

První dvě čísla (MajorVerze, MinorVerze) se vkládají do vybraných elementů WSDL:

- targetNamespace pro datové typy (WSDL <types>),
- portType,
- service name,
- endpoint.

IBM Integration Bus bude podporovat souběžný běh více verzí jedné webové služby.

5.8.2 Verzování balíčků

Jména balíčků služeb/procesů pro jejich nasazení na IP budou mít následující strukturu:
<JménoSlužby>_v_<MajorVerze>.<MinorVerze>.<MicroVerze>.bar

5.9 Transakce

Pro zajištění konzistence a integrity dat by měli poskytovatelé služeb při návrhu respektovat následující pravidla:

- Primárně by se měly využívat transakční mechanismy jednotlivých systémů, např. aplikačních serverů, relačních databází apod. – nikoliv IP.

- Granularita služeb by měla být dostatečně „hrubá“ tak, aby zapouzdřila i ošetření transakcí.
- V případech, kde to je relevantní, implementovat tzv. kompenzační služby, které umožňují návrat do původního stavu, pokud selže volání primární služby.

5.10 SLA

Pro každou službu by měly být definovány provozní parametry dle příslušné SLA, které vycházejí z požadavků MHMP. Tato oblast se patrně bude řešit později – nikoliv v prvotní implementaci IP.
V rámci SLA bude vhodné specifikovat (dle charakteru služby/procesu/systému/aplikace):

- kategorii služby (např. kritická/standardní/podpůrná);
- dostupnost, výjimky z dostupnosti (např. okna pro údržbu);
- rychlost odezvy (např. průměrná, maximální);
- plánovaný počet přenosů (např. počet zpráv za vteřinu/minutu/hodinu/den);
- plánovaná velikost přenosů (např. velikost zprávy);
- maximální možný čas na zprovoznění systému v případě jeho havárie;
- způsob řešení případných incidentů (např. způsob oznámení, způsob eskalace).

Dále by měl být připravený postup, jak monitorovat a následně reagovat v případě porušení SLA.

V prostředí MHMP je navrženo využít prostředků integrační platformy pro monitoring SLA a vynucení QoS integračních služeb. Řekněme, že služba poskytovaná poskytovatelem služby má kapacitu 80 transakcí za 1s. Máme čtyři systémy a každý potřebuje volat službu až 30x za 1s. V takovém případě jsou schopny teoreticky vytvořit zátěž až 120 transakcí za 1s. To znamená, že poskytovatel by byl vytížen na 150% a to je již značné přetížení a bude hrozit jeho zahlcení a zhroucení. Je však možno odhadovat, že všechny 4 systémy nebudou mít špičkovou zátěž právě ve stejném okamžiku.

- Informační systémy, které jsou klienty služeb se připojují na *API Gateway* pro registraci aplikace v *Developer portálu*. Zde je pro každý systém nastaven měkký (soft) a případně i tvrdý (hard) limit, jak často mohou volat danou službu nebo konkrétní operaci služby. Ve vztahu k příkladu výše by zde by byl nastaven limit v rámci plánu použití API na 25 požadavků za 1s jako soft limit, který způsobí alert do dohledového systému a 30 požadavků za 1s jako hard limit (31. požadavek v okně 1s nebude propuštěn).
- Následně IBM Integration Bus obsahuje tzv. workload policy. Ve vztahu k příkladu výše by zde by byl nastaven limit 80 (pokud systém snese mírné přetížení, pak i např. 90) transakcí za 1s. Backend systém tak bude ochráněn před přetížením nadměrným počtem požadavků.

6. Metodika a způsob poskytování služeb IP městským částem HMP

Při zpřístupňování integračních rozhraní městským částem, zřizovaným organizacím HMP a organizacím zřizovaným MČ by mělo být v maximální možné míře využito možností *API Geateway* na technoloigii *IBM DataPower Gateway X2*.

API Gateway má možnost být zapojena do řady sítí (fyzických i virtuálních). Díky tomu je možno zpřístupnit vybrané služby i městským částem a dalším organizacím. V rámci technologie *IBM API Connect* by byl vytvořen oddělený katalog, kde by byly služby vystavené pro účely informačních systémů zmíněných organizací. Díky tomu boudou mít tyto organizace svou virtuální instanci *Developer portálu* a bude pro ně možno nastavit jiná pravidla použití služeb než pro interní systémy.