ČESKÁ SPRÁVA SOCIÁLNÍHO ZABEZPEČENÍ
ÚSTŘEDÍ - SEKCE INFORMAČNÍCH A KOMUNIKAČNÍCH TECHNOLOGIÍ

Křížová 25, 225 08 Praha 5

Standard požadavků na aplikace při integraci do AAA portálu

verze 9.02

Historie verzí

Číslo verze	Datum verze	Vypracoval	Popis	Jméno souboru
1	21. 4. 2006	MV		pozadavky_na_aplikace.doc
2	24. 4. 2006	MV		pozadavky_na_aplikace_v2.doc
3	3. 5. 2006	MV		pozadavky_na_aplikace_v3.doc
4	22. 5. 2006	MV		pozadavky_na_aplikace_v4.doc
5	1. 6. 2006	MV	Přidání podpory pro fyzické role	pozadavky_na_aplikace_v5.doc
6	18. 8. 2006	MV	Odstranění skupin aplikací	pozadavky_na_aplikace_v6_nonfinal.doc
7	13. 3. 2006	MV	Doplnění možností komunikace mezi aplikacemi, revize příslušných částí	Pozadavky_na_aplikace_v7.doc
7.01	26. 3. 2006	MS	Doplnění organizačních opatření pro integraci nových aplikací	Pozadavky_na_aplikace_v7.doc
7.02	10. 10. 2013	Jana Fuková, Petr Soukup	Aktualizace kapitoly 13	Pozadavky_na_aplikace_v7.doc
7.03	5. 3. 2014	Jana Fuková, Petr Soukup, Petr Němec	Úprava dokumentu na základě migrace na nový sw	Pozadavky_na_aplikace_v7.doc
8.0	10. 4. 2014	Petr Němec	Celková revize dokumentu	Pozadavky_na_aplikace_v8.doc
8.01	10. 2. 2015	Petr Němec	Úprava VIP role	Pozadavky_na_aplikace_v8.doc
9.00	20. 7. 2015	Jana Fuková, Petr Soukup, Petr Němec	Celková revize dokumentu	Pozadavky_na_aplikace_v9.doc
9.01	29. 9. 2015	Petr Němec Zdeněk Rampas	Databázový audit	Pozadavky_na_aplikace_v9.doc
9.02	9. 2. 2016	Petr Němec, Jana Fuková, Petr Soukup	Technologické účty Změna názvu	Std_Pozadavky_na_aplikace_v9.doc

Obsah

1. Úvod

AAA portál je nástroj pro řízení přístupových oprávnění a audit přístupu do jednotlivých aplikací a databází v prostředí Integrovaného informačního systému ČSSZ (dále jen „IIS ČSSZ"). IIS ČSSZ byl určen usnesením vlády České republiky č. 576 dne 25. 5. 2015, jako systém kritické informační infrastruktury. ČSSZ se nabytím právní moci tohoto usnesení stala správcem informačního systému kritické informační infrastruktury podle § 3 písm. c) zákona č. 181/2014 Sb., o kybernetické bezpečnosti. V IIS ČSSZ je autentizace prováděna prostřednictvím zaměstnanecké čipové karty, autorizace je zabezpečena přidělováním rolí pro jednotlivé uživatele v grafickém rozhraní subsystému IBM Security Identity Manager (dále jen „ISIM GUI"). Audit přístupu k aplikacím je zabezpečen systémy CARS a audit přístupu k jednotlivým databázím je zajištěn prostřednictvím systému IBM InfoSphere Guardium.
Architektura v IIS ČSSZ je postavena na třívrstvé architektuře, která zaručuje dostatečnou bezpečnost dat a dovoluje aplikovat bezpečnostní principy řízení přístupu k aplikacím a datům. Základním principem tohoto uspořádání je stanovení pravidel pro komunikaci mezi vrstvami, které současně nastavují základní bezpečnost této komunikace pomocí nastavení principů a oprávnění komunikace mezi vrstvami. Tyto vazby jsou realizovány pomocí rozhraní na bázi webových služeb zčásti pomocí sběrnicových technologií (ESB IKR a ESB Backend).

Tento dokument patří mezi schválené standardy ČSSZ a je pro zhotovitele aplikací závazný.

2. Účel dokumentu

Účelem tohoto dokumentu je vymezit model komunikace pro aplikace a strukturu přístupových oprávnění. S tímto modelem musí tvůrci aplikací a AAA portálu ve svých projektech počítat jako se závazným.

Každá nová aplikace musí být nasazena do integračního prostředí, kde bude ověřena funkčnost v AAA portálu a splnění požadavků tohoto dokumentu.

3. Rozsah a působnost standardu

Tento standard se vztahuje na všechny aplikace v IIS ČSSZ, výjimkou jsou aplikace, které z historických důvodů nebyly zařazeny pod systém AAA Portál.

Aplikace, které nesplní požadavky dané tímto standardem, nebude možné v prostředí IIS ČSSZ provozovat.

Dokument se dělí na několik základních oblastí:
- požadavky na aplikace z hlediska komunikace
- popis struktury rolí v aplikacích
- požadavky z hlediska databázového auditu

4. Funkčnost

AAA portál je souhrn několika bezpečnostních systémů, mezi základní komponenty z hlediska aplikací patří tyto komponenty: WebSeal (autentizace, autorizace), AAA API (autorizace) a Audit (IBM InfoSphere Guardium, Datová vrstva).

4.1. Funkčnost - WebSeal

1. Klient spustí klientskou část aplikace (KA).

2. KA získá Kerberos ticket pro komunikaci s AAA portálem z Kerberos Key Distribution Center, který je na Domain Controlleru.

3. KA pošle dotaz (např. HTTP GET) na URL, pod kterým serverová část aplikace vystupuje na serveru WebSeal (ten funguje podobně jako proxy, přičemž skutečná IP adresa serverové části aplikace je z KA nedosažitelná).

4. WebSeal vrátí odpověď o požadavku na autentizaci klienta. Spolu s odpovědí WebSeal zašle KA také cookie relace.

5. KA pošle znovu dotaz na URL. Spolu s ním zašle WebSealu Kerberos ticket pro komunikaci s AAA portálem a taky cookie relace, které KA obdržela v kroku 4 od WebSealu.

6. V případě úspěšného ověření Kerberos ticketu bude uživatel po zbytek relace považován za autentizovaného, a žádná další autentizace již nebude požadována (KA posílá Kerberos ticket jenom při navazování relace s WebSealem (relace udržované pomocí cookies).

7. WebSeal odpoví KA přeposláním odpovědi z aplikačního serveru spolu s novou cookie relace V následné komunikaci musí být použita tato cookie. (Cookie obdržená v kroku 4 musí být v rámci další komunikace KA - Webseal nahrazena na straně KA novou cookie). (WebSeal udržuje s KA relaci pomocí session cookies).

8. Veškerá další komunikace mezi klientskou a serverovou částí aplikace prochází přes WebSeal. Ten ke každému HTTP požadavku z KA (GET i POST) přidá HTTP hlavičku obsahující identifikaci uživatele.

9. Pokud daný uživatel nemá právo přístupu k aplikaci (jinými slovy nemá žádnou roli v dané aplikaci), WebSeal tuto komunikaci nezprostředkuje.

10. Serverová část aplikace posílá své odpovědi WebSealu, který je přeposílá KA.

11. V případě vypršení relace požádá WebSeal klienta o opětovné zaslání Kerberos ticketu. To znamená, že na jakýkoliv další klientův požadavek odpoví WebSeal chybovou hláškou HTTP 401 Unauthorized. Klientská část aplikace pak musí (pro uživatele transparentně) serveru WebSeal odpovědět opětovným zasláním Kerberos ticket relace s dalším požadavkem. Cookie relace je vytvořena analogicky dle odstavců 4 – 7.

4.2. Funkčnost - AAA API

1. Serverová část aplikace převezme identitu uživatele z konkrétního spojení se serverem WebSeal.

2. Oprávnění uživatele a další informace o aktuálním uživateli (dle bodu 1) aplikace získá voláním metod služby AAA API, (jednotlivé metody jsou popsány v kapitole 7).

3. Aplikace nastaví prostředí aktuálního uživatele podle oprávnění získaných v bodu 2.

4.3. Funkčnost – Audit - datová vrstva

1. Serverová část aplikace převezme identitu uživatele z konkrétního spojení se serverem WebSeal dle zvolené metody (viz kapitola 15).

2. Serverová část aplikace vloží identitu získanou v bodě 1 do komunikace směrem k databázovému serveru.

5. Požadavky na komunikaci

Každá aplikace, která je integrována do AAA portálu, musí splnit požadavky popsané v následujících kapitolách.

5.1. Klientská část aplikace

- Klientská část aplikace nikdy nepřistupuje přímo k serverové části aplikace, ale vždy přistupuje k serveru WebSeal. Pro všechny WebSeal servery je v rámci sítě ČSSZ zřízena VIP adresa wapp.cssz.cz. Příklad: pokud se aplikace jmenuje *aplikace1*, její zkratka (tří případně čtyř písemná) je *ap1*, adresa její serverové části je *www.ap1.cssz.cz,* bude klient přistupovat k adrese *wapp.cssz.cz/ap1,* nikoliv k *www.ap1.cssz.cz.* WebSeal prověří, zda je uživatel oprávněn přistupovat k aplikaci, a pokud ano, provede přesměrování.

- Klientská část aplikace musí být kerberizována, tj. musí umět získat z Domain Controleru Kerberos ticket pro komunikaci s AAA portálem, konkrétně s webovou službou wapp.cssz.cz. (WebSealy jsou replikovány, ale mají jen jeden účet v Active Directory). Tento ticket pak musí odeslat WebSealu jako odpověď na jeho zprávu „401 Unauthorized". Ticket pošle aplikace v HTTP hlavičce v položce „Authorization" (viz příklad komunikace MS Internet Exploreru s WebSeal dále v tomto dokumentu).

- Klientská část aplikace musí komunikovat přes HTTP nebo HTTPS.

- Klientská část aplikace musí umět pracovat s cookies. Pomocí cookies se udržují relace mezi klientskou částí aplikace a WebSealem.

- Pokud klientská část aplikace potřebuje znát seznam rolí daného uživatele, poskytne jí ho serverová část aplikace.

5.2. Serverová část aplikace - webová služba - „střední vrstva"

- Webová služba musí umět zpracovat HTTP hlavičky přidávané do komunikace WebSealem. Hlavičky budou standardně obsahovat identifikace uživatele (*iv-user*), jeho Windows login name. Tento záznam HTTP hlavičky mohou používat CGI programy jako proměnnou prostředí HTTP_IV_USER (viz příklad komunikace s aplikačním serverem).

- V případě komunikace webové služby s jinou webovou službou, musí tato volající webová služba posílat v rámci dotazu identifikaci koncového uživatele obdrženou z WebSealu (iv-user). Volající webová služba musí předat jiné (volané) Webové službě jméno koncového uživatele. Pro tyto

účely, je stanovený název pole pro předávání informace o identitě (původci) dotazu v případě mezi aplikační komunikace, takto: HTTPHeaderAuditUser= "audit-user-id"

- Pokud bude serverová část aplikace posílat uživateli libovolné URL (například v rámci www stránky), musí toto být v relativním tvaru (například: *abc.html* nebo *../abc.html* nebo *./abc.html* nebo *program/abc.html*). Nikoliv ve tvaru relativním pro server (například: */abc.html* nebo */program/abc.html*), ani ve tvaru absolutním (například: *www.cssz.cz/program/abc.html*). Všechna URL posílaná uživateli budou WebSealem modifikována tak, aby odpovídala přístupovému bodu dané aplikace na WebSeal.

5.3. Příklad komunikace Microsoft Internet Exploreru (klient) s WebSealem

Komunikace směrem Klient IE -> WebSeal je kurzívou.

GET /nem/nem/main.jsp HTTP/1.1
*Accept: text/html, application/xhtml+xml, */**
Accept-Language: cs-CZ
User-Agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate, peerdist
Host: wapp.cssz.cz
Connection: keep-alive
X-P2P-PeerDist: Version=1.0

HTTP/1.1 401 Unauthorized
connection: close
content-length: 2185
content-type: text/html
date: Thu, 16 Feb 2016 15:23:19 GMT
p3p: CP="NON CUR OTPi OUR NOR UNI"
server: WebSeal/7.0.0.0 (Build 121024)
www-authenticate: Negotiate
cache-control: no-cache
pragma: no-cache
Set-Cookie: PD-H-SESSION-ID=1_4_0_T0gcYDQqZhhZj-rsso8aGQAuGU0i6-uC7-oYuioB0cNFvSOE; Path=/

<Následuje text chyby (html)……>

GET /nem/nem/main.jsp HTTP/1.1
*Accept: text/html, application/xhtml+xml, */**
Accept-Language: cs-CZ
User-Agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate, peerdist
Host: wapp.cssz.cz
Connection: keep-alive
X-P2P-PeerDist: Version=1.0
Cookie: PD-H-SESSION-ID=1_4_0_T0gcYDQqZhhZj-rsso8aGQAuGU0i6-uC7-oYuioB0cNFvSOE
Authorization: Negotiate: YIIHowYG KwYBBQU CoIIHlz CCB5OgM DAuBgkqhkiC9x IBAgI GCSqGSIb3Eg ECAg….

Poznámka: Položka Negotiate zde není uvedena celá.

HTTP/1.1 200 OK
connection: close

content-type: text/html;charset=UTF-8
date: Thu, 16 Feb 2016 15:23:20 GMT
p3p: CP="NON CUR OTPi OUR NOR UNI"
Server: Apache-Coyote/1.1
Set-Cookie: PD-H-SESSION-ID=1_4_0_T0gcYDQqZhhZj-rsso8aGQAuGU0i6-uC7-oYuioB0cNFvSOE; Path=/

< Zde je v odpovědi serveru umístěna požadovaná WWW stránka.....>

5.4. Příklad komunikace WebSealu s aplikačním serverem

Komunikace směrem WebSeal -> Aplikační server je kurzívou.

GET /tcl/MasterPage/gma_style.css HTTP/1.1
*accept: */**
accept-language: cs-CZ
host: 10.200.15.134
if-modified-since: Wed, 13 Mar 2013 10:16:32 GMT
if-none-match: "08073d5d31fce1:50d5"
iv-user: zngolma2
referer: http://wapp.cssz.cz/tcl/
user-agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; InfoPath.3; .NET4.0C; .NET4.0E)
via: HTTP/1.1 va2x005p03:8585
iv_server_name: ws02-WebSeald-va2x005p03.app.cssz.cz
Cookie:__utma=21455273.1223407899.1392722890.1396524821.1396854883.5; __utmz=21455273.1396524821.4.2.utmcsr=intranet.cssz.cz|utmccn=(referral)|utmcmd=referral|utmcct=/; ASP.NET_SessionId=vlfqhh3blwl02fndi031me55

HTTP/1.1 200 OK
Content-Length: 28341
Content-Type: text/css
Last-Modified: Wed, 13 Mar 2013 11:16:32 GMT
Accept-Ranges: bytes
ETag: "0e83737dc1fce1:4b4c"
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Date: Thu, 10 Apr 2014 13:48:53 GMT

< Zde je v odpovědi serveru umístěna požadovaná WWW stránka.....>

5.5. Příklad komunikace s databází

Databázové volání musí obsahovat informaci o koncovém uživateli, který databázové volání inicioval. Toto lze realizovat ve dvou krocích:

- Získat informace o koncovém (přihlášeném) uživateli
- Upravit existující databázové volání, a vložit do volání informace o koncovém uživateli, například vložením SQL příkazu (viz níže).

SQL příkaz na začátku transakce:

```
SELECT 'GuardAppEvent:Start','GuardAppEventUserName:<username>'
FROM SYSIBM.SYSDUMMY1
```

SQL příkaz na konci transakce:

```
SELECT 'GuardAppEvent:Released','GuardAppEventUserName:<username>'
FROM SYSIBM.SYSDUMMY1
```

Příklad komunikace aplikace PSL:

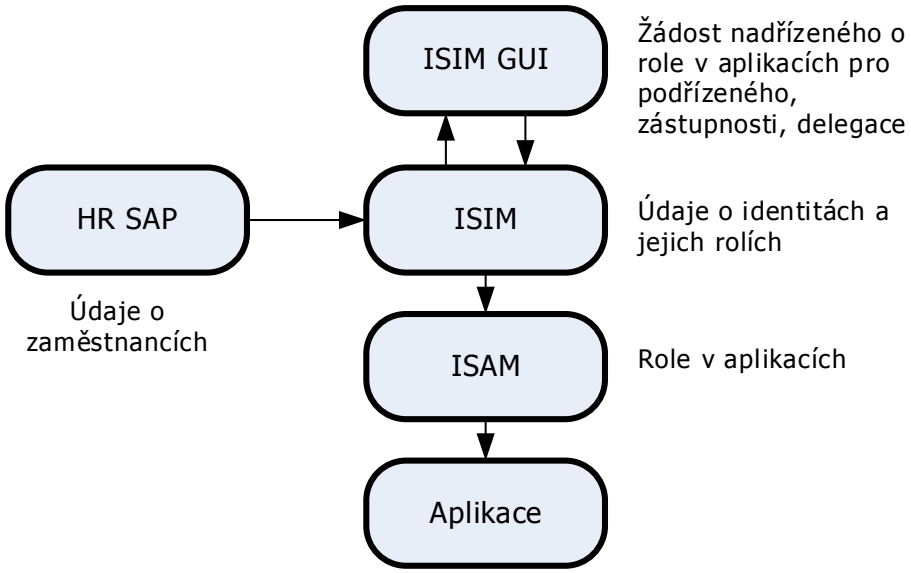
```
DBMS_SESSION_SET_IDENTIFIER('xxuzifra::1234567890');
UPDATE tbPER SET PER_Lock=1, PER_DatLock=sysdate,
PER_UsrLock='uzifra_00/XXXXXXXX' WHERE tbPER.PER_Id=12345
DBMS_SESSION_SET_IDENTIFIER('xxuzifra::0987654321');
SELECT * FROM tbRIZ
DBMS_SESSION_SET_IDENTIFIER('xxuzifra::0987654321');
UPDATE tbPER SET PER_Lock=0, PER_DatLock=sysdate, PER_UsrLock='' WHERE
tbPER.PER_Id=12345
DBMS_SESSION_SET_IDENTIFIER('xxuzifra::5432109876');
UPDATE tbPER SET PER_Lock=1, PER_DatLock=sysdate,
PER_UsrLock='xxuzifra_05/YYYYYYYY' WHERE tbPER.PER_Id=67890
```

6. Struktura rolí v aplikacích (uživatelská přístupová oprávnění)

Role Based Access Control (dále jen „RBAC“) je základní princip (přístupový model), na němž je AAA portál postaven. Uživatelé jsou mapováni do skupin, které mají základ v jejich pracovním zařazení v organizaci nebo úloze v aplikaci. RBAC model rolí je implementován v systému ISIM, který si udržuje aktuální model rolí pro každou aplikaci nebo cílový systém. Pro oblast rolí je cílovým systémem komponenta ISAM, která zajišťuje autorizaci (povoluje uživatelům přístup do aplikací na základě rolí). Informace o přidělených oprávněních získávají aplikace prostřednictvím AAAAPI právě ze systému ISAM.

Základní rozdělení rolí v systému AAA portál

- Logické role (úloha uživatele v aplikaci)
- Fyzické role (funkčnost/práva konkrétní aplikace)



Obrázek 1 *Schéma jednotlivých komponent systému AAA portál*

6.1. Struktura rolí v systému ISAM

6.1.1. LOGICKÉ ROLE

Jsou role vyššího stupně, které určují roli - úlohu daného uživatele v aplikaci. Příklad takové role je role „referent". Tyto role mohou být konkrétnější, například může existovat aplikace, která bude mít jenom 3 role: čtení, zápis, mazání. V takovém případě jsou tyto role považovány za logické.

O logické role žádá nadřízený přes aplikaci ISIM GUI a ve schvalovacím procesu je přidělení schváleno pověřenými osobami.

Logické role mají tvar:

xxx_yyy,

kde xxx je alfanumerický řetězec (malé délky) jednoznačně identifikující konkrétní aplikaci, yyy je alfanumerický řetězec (malé délky) jednoznačně identifikující konkrétní logickou roli v aplikaci.

Systém AAA část ISAM umožní uživateli přístup jenom k datům, na která ho opravňují jeho aplikační logické role.

6.1.2. FYZICKÉ ROLE

Tyto role jsou navázány na konkrétní funkčnosti dané aplikace, např. role: ap1_F_1 „mazání z tabulky A5". Tyto role se aplikacím také předávají prostřednictvím AAA API. Pro fyzické role je zřízena zvláštní instance LDAP AAA, která zajišťuje mapování logických rolí na role fyzické.

Fyzické role mají tvar:

xxx_F_yyy,

kde xxx je alfanumerický řetězec (malé délky) jednoznačně identifikující konkrétní aplikaci, yyy je alfanumerický řetězec (malé délky) jednoznačně identifikující konkrétní fyzickou roli v aplikaci.

6.1.3. Vztah logických a fyzických rolí

Aplikacím předává AAA portál jak logické role, tak fyzické role. Je na tvůrcích jednotlivých aplikací, aby navrhli konkrétní logické a fyzické role a jejich vzájemnou vazbu. Role se pak zanesou do AAA portálu. Tyto konkrétní role navržené aplikací musí ale odpovídat struktuře rolí popsané v tomto dokumentu.

Logické role jsou vztaženy v ISAM vždy ke konkrétní identitě (zaměstnanci). Přidělují se na základě žádosti vytvořené nadřízeným přes ISIM GUI a po schválení se promítají do ISAM.

Fyzické role jsou nahrány do instance LDAP AAA pomocí XML souborů, který předávají garanti/tvůrci aplikací. Neexistuje žádná přímá vazba zaměstnanec – fyzická role.

Zaměstnanec -> přidělená logická role -> fyzické role mapované na danou logickou roli.

AAA portál může tedy předat aplikacím také vazbu mezi logickými a fyzickými rolemi.

6.1.4. Další členění logických rolí pro aplikace

- **Role lokalizační**

Tyto role jsou ve tvaru:

***xxx*_L_*yyy*,**

kde xxx je alfanumerický řetězec (malé délky) jednoznačně identifikující konkrétní aplikaci. yyy je alfanumerický řetězec (malé délky) jednoznačně identifikující lokalitu (okres), ke které se role uživatele v dané aplikaci vztahují, tj. ke kterým datům (míněno ve smyslu lokalizačního členění dat nebo původu dat) má daný uživatel právo přistupovat prostřednictvím dané aplikace. Existuje také role tvaru xxx_L_0 (ALL), která určuje, že daný uživatel může na všechna data, ke kterým ho opravňují jeho aplikační logické role bez omezení na lokalitu.
Je zřejmé, že tvůrci aplikací by měly používat lokalizační role, jen pokud to je nezbytné. Pro lokalizační role je v AAA zaveden jednotný číselník lokalit, který je uveden v dokumentu AAA_rollout_support , který je dostupný na Intranetu ČSSZ v menu Aplikace/ ISIM GUI/Dokumenty – Podpora pro nasazení).

- **VIP role**

VIP role je pro aplikace nepovinná. Může být přidělena uživateli, který může pracovat s daty VIP osob. Tato role je ve tvaru:

***xxx_viplegacy*,**

kde xxx je alfanumerický řetězec (malé délky) jednoznačně identifikující konkrétní aplikaci a viplegacy je vyhrazený identifikátor.

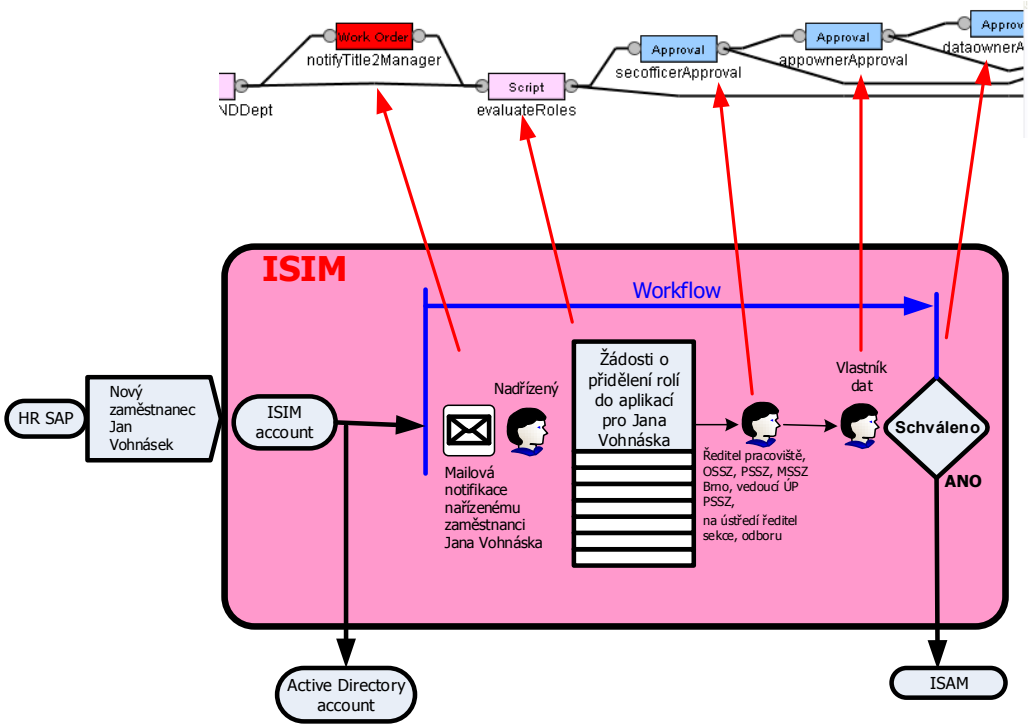
Tvůrci aplikací si určí tvar řetězců identifikujících logické role a výkon role (nesmí se shodovat s lokalizačními rolemi ani s VIP rolí). Ze strany AAA portálu jim bude přidělen řetězec identifikující danou aplikaci. Tvůrci aplikací také určí popisy jednotlivých rolí a jejich názvy.

- **Role ESS_výjimka**

ESS_výjimka je sada specifických rolí nově zavedených do AAA portálu pouze pro aplikaci ESS.

Tyto role se používají POUZE v případě, že podřízený pracuje v sekundární instanci ESS nebo potřebuje umožnit přístup do ESS uživateli ze skupiny: „externí pracovníci", viz dále. Role ESS_výjimka vyjadřuje „sdílené funkční místo" – práce v sekundární instanci ESS. Číselník rolí ESS_výjimka udržuje tým AAA ve spolupráci s provozním týmem aplikace ESS. První případ použití této role je pro sdílená pracoviště (sekundární instance ESS, např. jedna podatelna v rámci OSSZ/PSSZ). Druhý případ použití této role je pro skupinu externí pracovníci (externisté - dodavatelé – 97, externisté stážisté – 98, agenturní zaměstnanci – 99, DPČ - vlastní; DPČ - cizí; DPP - cizí).
Systém AAA Portál umožňuje nastavit maximálně dvě výjimky (pro každého uživatele) pro aplikaci ESS.

6.2. Role z pohledu GUI nad ISIM



Obrázek 2 Zjednodušené workflow nástupu nového zaměstnance a přidělení rolí

Nativní rozhraní ISIM umožňuje pouze základní přizpůsobení vzhledu, proto byla nad ISIM vytvořena grafická nadstavba – ISIM GUI.
ISIM GUI je webová aplikace, která využívá jako zdroj i cíl dat ISIM. Přes ISIM API si vyzvedává seznam aplikací, seznam rolí, zaměstnanců apod. a formuje z nich formuláře pro nadřízeného, který žádá o nějakou roli v aplikaci pro svého podřízeného. Vyplnění a odeslání formuláře se žádostí spouští v ISIM schvalovací workflow. Pověřené osoby (schvalovatelé) mají přístup do ISIM GUI, kde mohou příslušnou žádost schválit nebo zamítnout.
Informace o rolích v aplikacích musí být proto pro ISIM GUI co nejpodrobnější, musí obsahovat názvy a popisy položek apod. tak, aby jim rozuměli netechničtí zaměstnanci.

Hlavním úkolem ISIM GUI je nabídnout nadřízenému a později jednotlivým schvalovatelům seznam aplikačních rolí ve vhodné struktuře. Nadřízenému se zobrazí všechny evidované role k aplikacím. Schvalovatelům, poté už pouze obyčejný seznam vybraných rolí, které jako celek buď schválí, nebo odmítnou.

6.3. Schéma názvů rolí v ISIM

6.3.1. Logické role

NAZEV APLIKACE|NAZEV ROLE|UPRESNENI

„NAZEV ROLE" logické role se musí skládat z třípísmenné nebo čtyřpísmenné zkratky aplikace, podtržítka a volitelného sufixu, nesmí obsahovat řetězec "VIPLEGACY" (např. NEM1_VIPLEGACY) a nesmí obsahovat samotné písmeno „L" (vyhrazeno pro lokalizační role) např.: NEM_L_KUKATKO. Pokud je v ID logické role vyhodnocen řetězec „VIPLEGACY", pak je schvalovací workflow automaticky

směrováno na ředitele odboru bezpečnostní politiky ČSSZ (12) z důvodu vyhodnocení, že se jedná o údaje osob VIP. (Popsáno v dokumentu AAA_rollout_support , který je dostupný na Intranetu ČSSZ v menu Aplikace/ ISIM GUI – Podpora pro nasazení).

6.3.2. Lokalizační role

Pro aplikace, jejichž role jsou lokalizované, existuje v ISIMu 105 rolí (podle počtu lokalit) pro každou aplikaci, které jsou ve tvaru:

NAZEV APLIKACE|_NAZEV LOKALITY

6.3.3. VIP role

Pro aplikace, které rozlišují VIP přístup ke zvláště chráněným datům, jsou vytvořeny role ve tvaru:

CSSZ|NAZEV APLIKACE|_VIPLEGACY

V názvu VIP rolí se nesmí vyskytovat znaky „_" (vyčleněn pro rozpoznání lokalizační role) a „|" (vyčleněn pro oddělení úrovně ve stromu). V ISIM GUI nadřízený nejprve vybere jednotlivé role a poté má možnost lokalizovat aplikace, které tuto možnost podporují.

6.4. Vazba rolí v ISIM na ISAM

Typ role	ISIM	ISAM
Lokalizační	CSSZ|NAZEV APLIKACE|_ALL CSSZ|NAZEV APLIKACE|_NAZEV LOKALITY	xxx_L_0 *např. NEM_L_0* xxx_L_yyy *např. NEM_L_101*
VIP	CSSZ|NAZEV APLIKACE|_VIPLEGACY	xxx_VIPLEGACY

Kvůli jednoznačnosti vazby rolí v ISIM a skupin v ISAM je třeba do atributu description role uvádět i její identifikátor v ISAM (ve tvaru xxx_yyy_zzz). Hodnotu atributu description získá ISIM z ISAM.

xxx_yyy_zzz |UPŘESŇUJÍCÍ POPIS ROLE
např. NEM_Referent_12|Referent EPN

V atributu description rolí se nesmí vyskytovat znak „|" (vyčleněn pro oddělení popisu a ISAM označení role). Maximální délka popisu role je 200 znaků včetně mezer.

6.5. Vazba logických rolí na fyzické

Fyzické role a jejich vazba na logické role je udržována ve strukturách LDAP AAA, nepodléhá tedy schvalovacímu procesu ISIM/ISAM. Pro role je v AAA portálu vytvořen číselník. Jednoznačná je vždy vazba aplikace_role např. NEM_1.

Příklad definic pro aplikaci NEM:

Logické role

ID role	Název	Popis
NEM_1	Referent EPN	Provádí změnu v zaevidovaných případech NP, udržuje registr lékařů, vydává formuláře PN
NEM_2	Referent DNP	Rozhoduje o dávkách nemocenského pojištění.
NEM_3	Aprobant	Aprobuje dávky nem. pojištění.
NEM_4	Vedoucí pracovník	Uvolňuje dávky k výplatě.
NEM_5	Nadřízený orgán	Přezkoumává reklamace.
NEM_7	Administrátor	Speciální role -spouští dávkové úlohy, administruje uživatelské filtry.
NEM_8	Referent KLR	Eviduje podněty pro kontroly KLR (přiřazuje k případům PN), vytváří podněty pro kontroly KLR, vytváří plány kontrol KLR, vytváří def. statistiky.
NEM_9	Vedoucí referent KLR	Schvaluje plány kontrol KLR, sleduje lhůty související s kontrolami KLR.

Fyzické role

ID role	Popis
NEM_F_1	Právo zobrazit a vyhledat dokument (menu dokument, odkazy na dokument)
NEM_F_2	Právo zobrazit záložku Stav dokumentu
NEM_F_3	Právo zobrazit záložku Detail
NEM_F_4	Právo editovat poznámku
NEM_F_5	Právo zobrazit záložku Kontroly
NEM_F_6	Právo změnit stav na zpracován
NEM_F_7	Právo změnit stav na storno
NEM_F_8	Právo předat dokument
NEM_F_9	Právo propojit dokument s případem
NEM_F_10	Právo odpojit dokument od případu
NEM_F_11	Právo vytvořit případ
NEM_F_12	Právo provést hromadné předání dokumentů
NEM_F_20	Právo zobrazit a vyhledat dávku (menu dávka, odkazy na dávku)
NEM_F_21	Právo zobrazit záložku Výpočet

Mapování rolí

Logická role	Fyzické role
NEM_1	NEM_F_1, NEM_F_2, NEM_F_3, NEM_F_4, NEM_F_5, NEM_F_6, NEM_F_7
NEM_2	NEM_F_1, NEM_F_100, NEM_F_102, NEM_F_103, NEM_F_104, NEM_F_111, NEM_F_113, NEM_F_114, NEM_F_115, NEM_F_116, NEM_F_119, NEM_F_12, NEM_F_120, NEM_F_126, NEM_F_2, NEM_F_21, NEM_F_26, NEM_F_29, NEM_F_3, NEM_F_30, NEM_F_4, NEM_F_5, NEM_F_51, NEM_F_54, NEM_F_59, NEM_F_6, NEM_F_61, NEM_F_62, NEM_F_63, NEM_F_64, NEM_F_65, NEM_F_67, NEM_F_68, NEM_F_69, NEM_F_7, NEM_F_70, NEM_F_76, NEM_F_78, NEM_F_79, NEM_F_82, NEM_F_83, NEM_F_84, NEM_F_85, NEM_F_86, NEM_F_87, NEM_F_88, NEM_F_89, NEM_F_90, NEM_F_99
NEM_3	NEM_F_1, NEM_F_2, NEM_F_3, NEM_F_30, NEM_F_31, NEM_F_37, NEM_F_38, NEM_F_4, NEM_F_40, NEM_F_42, NEM_F_44, NEM_F_45, NEM_F_5, NEM_F_53, NEM_F_54, NEM_F_55, NEM_F_56, NEM_F_57, NEM_F_58, NEM_F_59, NEM_F_60, NEM_F_61, NEM_F_62, NEM_F_63, NEM_F_64, NEM_F_84, NEM_F_98
NEM_4	NEM_F_1, NEM_F_115, NEM_F_117, NEM_F_126, NEM_F_2, NEM_F_26, NEM_F_3, NEM_F_30, NEM_F_31, NEM_F_33, NEM_F_4, NEM_F_5, NEM_F_51, NEM_F_53, NEM_F_54, NEM_F_55, NEM_F_56, NEM_F_57, NEM_F_58, NEM_F_59, NEM_F_60, NEM_F_61, NEM_F_62, NEM_F_63, NEM_F_64, NEM_F_65, NEM_F_66, NEM_F_67

Fyzické role nemají na rozdíl od logických rolí přímou vazbu na uživatele. Jsou namapovány na logické role.

Pro zjištění vazeb fyzických rolí na logické role byly do AAA API přidány metody **getPhysicalRoles (logicalRole)** a **getLogicalRoles (physicalRole)**.

6.6. Struktura rolí aplikace

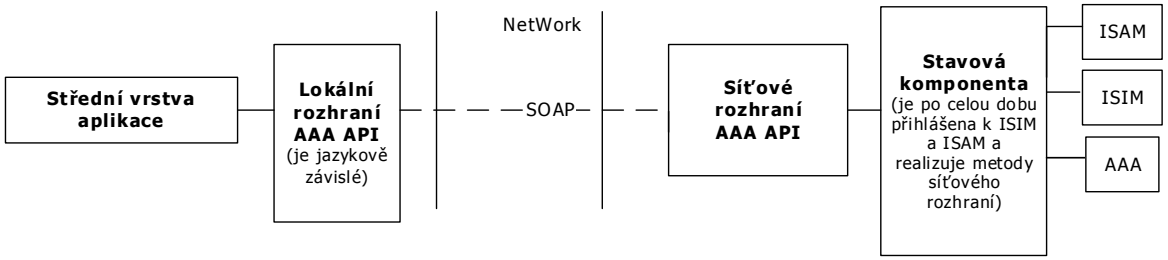
Každá aplikace může obsahovat pouze svoje vlastní role. Multiplikace (aplikace složené z více modulů/rolí) AAA portál nepodporuje. Všechny role aplikace schvaluje vždy jeden schvalovatel. Pokud se aplikace skládá z více modulů/rolí, je nutné každý modul integrovat samostatně. Takto lze integrovat aplikace pouze ve výjimečných a odůvodněných případech.

6.7. Nástroj pro vytváření aplikačních rolí

Z důvodu zjednodušení celého procesu zavedení aplikace do AAA portálu je k dispozici nástroj (průvodce tvorbou rolí) A3Editor, pro tvorbu XML souborů. Pomocí tohoto nástroje je nutné vytvořit dva XML soubory (jeden pro TP a PP, druhý pro IP), který obsahuje definici aplikačních rolí, logické role, fyzické role jejich mapování a další údaje potřebné pro integraci aplikace. Nástroj A3Editor je k dispozici u garanta AAA portálu.

7. AAA API

AAA API umožňuje aplikacím získávat údaje z databází ISIM a ISAM pomocí volání metod tohoto API. Jednotlivé metody jsou popsány v tomto dokumentu.



Obrázek 3 Implementace komponenty AAAAPI

Jako zdroj informací slouží:

- ISIM LDAP pro informace o zaměstnancích,
- ISAM LDAP pro informace o logických rolích zaměstnanců,
- AAA LDAP pro informace o fyzických rolích a jejich mapování na logické role.

7.1. Specifikace metod rozhraní (příklady)

Method Summary	
java.util.List	**getInferiors**(java.lang. String logonName) Gets inferiors of the users specified by their logon name.
RoleInfo	**getLogicalRoleInfo**(java.lang.String role) Gets logical role info by the given logical role code.
java.util.List	**getLogicalRoles**(java.lang.String physicalRole) Gets logical roles assigned to the specified physical role.
java.lang.String	**getLogonName**(long personalId) Gets logon name for the given user ID.
java.lang.Long	**getPersonalId**(java.lang.String logonName) Gets user ID for the given logon name.
java.util.List	**getPhysicalRoles**(java.lang.String logicalRole)

	Gets physical roles assigned to the specified logical role.
UserAppRoles	**getRolesByLogon**(java.lang.String appIdjava.lang.String logonName) Determines set of roles of the user specified by his logon name within the given application.
java.lang.String	**getSuperior**(java.lang.String logonName) Gets a superior user of the users specified by their logon name.
UserInfo	**getUserInfo**(java.lang.String logonName) Determines user info for the users specified by their logon name.
java.util.List	**getUserInfoEmailsByPersonalIds**(java.util.Collection personalIds) Determines e-mails from the user info using a collection of personal ids.
java.util.List	**getUserInfosInRoles**(java.util.Collection roles) The same as getUsersInRoles(Collection), but also eager-fetches UserInfo objects - use only when necessary.
java.util.List	**getUsersInRoles**(java.util.Collection roles) Determines a set of users having all of the specified roles.
java.util.List	**synchronizeUsers**(java.lang.String lastTriggeredChangeDate) Gets UserInfo objects for users which has been modified after the specified date.

- ***Metoda getInferior***

  Volání

  getInferior (string logonName)

  Odpověď

  Seznam LogonName uživatelů podřízených uživateli indexovanému pomocí LogonName.

  Doba držení v cache

  120 sekund

- ***Metoda getLogicalRolesInfo***

  Volání

  getLogicalRolesInfo (String role)

  Parametry

  role – identifikátor logické role

  Odpověď

  Vrací informace o logické roli.

  Doba držení v cache

  120 sekund

- ***Metoda getLogicalRoles***

  Volání

  getLogicalRoles (String physicalRole)

  Parametry

physicalRole – identifikátor fyzické role

Odpověď

Vrací seznam identifikátorů logických rolí nebo null, když zadaná fyzická role neexistuje.

Doba držení v cache

120 sekund

- ***Metoda getLogonName***

Volání

getLogonName (personalId)

Odpověď

LogonName uživatele indexovaného pomocí personalId. Pokud daný uživatel neexistuje, metoda vrací „null".

Doba držení v cache

120 sekund

- ***Metoda getPersonalId***

Volání

getPersonalId(String logonName)

Parametry

logonName – uživatelův logon name

Odpověď

Osobní číslo uživatele nebo null

Doba držení v cache

120 sekund

- ***Metoda getPhysicalRoles***

Volání

getPhysicalRoles (String logicalRole)

ParametrylogicalRole – identifikátor logické role

Odpověď

Seznam identifikátorů fyzických rolí nebo null, jestliže logická role neexistuje.

Doba držení v cache

120 sekund

- ***Metoda getRolesByLogon***

  Volání

  getRolesByLogon(String appId, String logonName)

  Parametry

  appId – identifikátor aplikace
  logonName – logon name uživatele

  Odpověď

  Seznam rolí uživatele v dané aplikaci; pokud se uživatel nenajde, vrací se prázdný seznam.

  Doba držení v cache

  120 sekund

- ***Metoda getSuperior***

  Volání

  getSuperior (string logonName)

  Odpověď

  logonName, resp. personalId uživatele nadřízeného uživateli indexovanému pomocí logonName.

  Doba držení v cache

  120 sekund

- ***Metoda getUserInfo***

  Volání

  getUserInfo (string logonName)

  Odpověď

  Řádek z tabulky USERS_INFO z databáze DB2 indexován logonName. Řádek obsahuje položky jako jméno, příjmení, číslo kanceláře, e-mail, telefon apod.

  Doba držení v cache

  120 sekund

- ***Metoda getUserInfoEmailsByPersonalIds***

  Volání

  getUserInfoEmailsByPersonalIds (string personalIds)

  Odpověď

  Vrací e-mail pro seznam personalIds.

  Doba držení v cache

120 sekund

- ***Metoda getUsersInRoles***

  Volání
  getUsersInRoles (Collection roles)

  Odpověď
  Seznam logonName uživatelů majících danou roli

  Doba držení v cache
  120 sekund

7.2. Metoda zjišťující mapování logických rolí na fyzické role

Aby bylo možné jednoznačně určit vztah mezi rolemi fyzickými a logickými, byla do prostředí AAA přidaná funkčnost mapování logických rolí na fyzické role (role zobrazené v ISIM GUI jsou role logické).

V LDAP serveru AAA jsou uloženy informace o fyzických rolích, jejich popis, jméno apod. Ve druhé části jsou uloženy vazby mezi fyzickou rolí s příslušnými logickými rolemi a naopak. V tomto LDAP serveru jsou udržovány pouze vazby mezi logickými a fyzickými rolemi. Vazba fyzické role přímo na uživatele se neudržuje, ta je řešena vždy prostřednictvím logické role.

- **Metoda List getPhysicalRoles(string logicalRole)**

  Volání
  getPhysicalRoles (string logicalRole)

  Odpověď
  Seznam fyzických rolí příslušejících k dané logické roli

  Doba držení v cache
  30 minut

- **Metoda List getLogicalRoles(string physicalRole)**

  Volání
  getLogicalRoles (string physicalRole)

  Odpověď
  Seznam logických rolí, které obsahují danou fyzickou roli

  Doba držení v cache
  30 minut

Projektový tým AAA portálu je schopen na základě požadavků jednotlivých aplikací přidávat do AAA API další metody.

8. Integrace softwaru vyvíjeného na základě obecných požadavků (krabicový sw)

Financování integrace aplikace pod AAA portál je plně v režii přistupující aplikace.

1. V případě, že lze software vyvíjený na základě obecných požadavků (dále jen „krabicový SW) upravit dle kapitoly 7 (AAA API), pak je integrace pod AAA portál splněna.

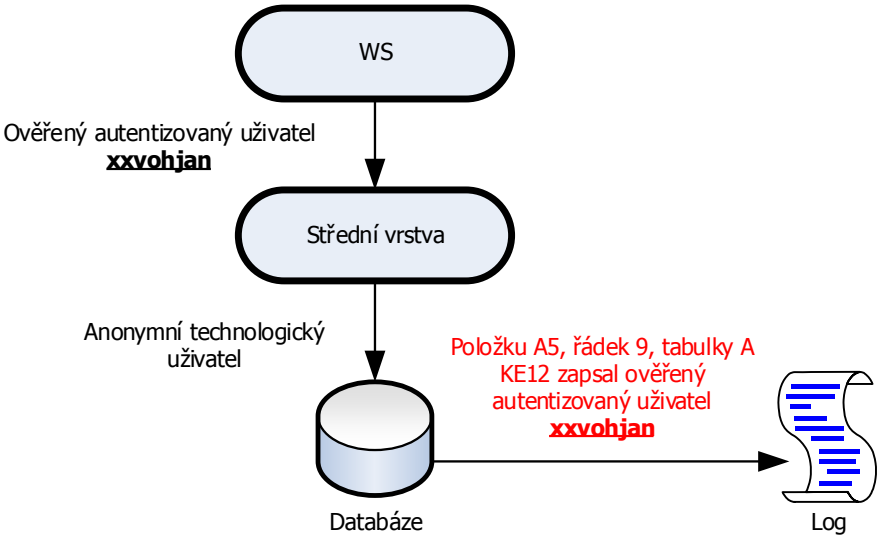
2. V případě, že krabicový SW nelze upravit, je nutné dohodnout s týmem AAA způsob integrace pod AAA. Řešení pomocí konektoru.

   - Integrace konektorem od IBM – pokud takový konektor pro krabicový SW již existuje, lze tuto aplikaci integrovat pomocí tohoto konektoru. Je řešeno integrací a konfigurací.

   - Integrace vývojem – Část konektoru (TDI) má k dispozici AAA portál a druhou část je potřeba vždy vyvinout na straně aplikace. Je řešeno vývojem, integrací a konfigurací.

9. Technologický uživatel a datová vrstva

Hlavním cílem systému "AAA portál" je vytvoření takového prostředí, aby byl každý přístup k aplikaci jednoznačně dohledatelný a spojený s konkrétním autentizovaným uživatelem. Z tohoto důvodu není možné vytvářet anonymní uživatele. AAA portál dle svého zadání končí na aplikační vrstvě a do datové vrstvy nezasahuje, nevytváří tedy ani uživatele v databázích.



Obrázek 4 Schéma komunikace aplikační části s databázovou vrstvou

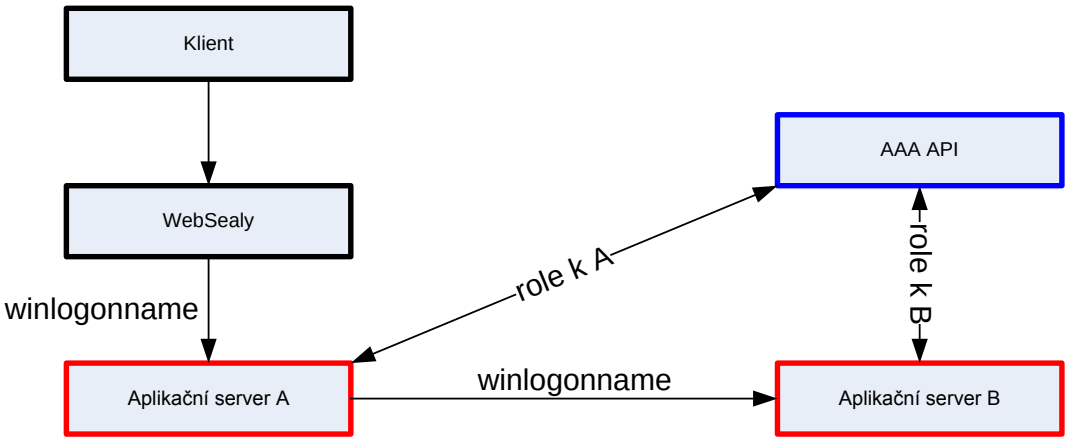
Aby celé řešení bezpečnosti mělo smysl, musí střední vrstva předat identitu autentizovaného uživatele datové vrstvě – neboli **aplikace musí zajistit**, aby v logových záznamech bylo vždy uvedeno, že autentizovaný uživatel provedl akci – nikoliv technologický uživatel.

Aplikace NEM používá pro připojení do DB tzv. Database Connection Pool, který umožňuje sdílení databázových spojení více uživatelům. Databázové spojení je vytvořeno pomocí technologického uživatele (user/password), a je sdíleno jednotlivými identifikovanými uživateli.

9.1. Komunikace aplikace s datovou vrstvou

Produkt IBM InfoSphere Guardium podporuje několik metod identifikace aplikačního uživatele v rámci komunikace mezi aplikačním a databázovým serverem. Způsob integrace aplikace na úrovni DB vrstvy je detailně popsán v kapitole 15 Metody identifikace aplikačního uživatele.

10. Komunikace mezi aplikačními servery

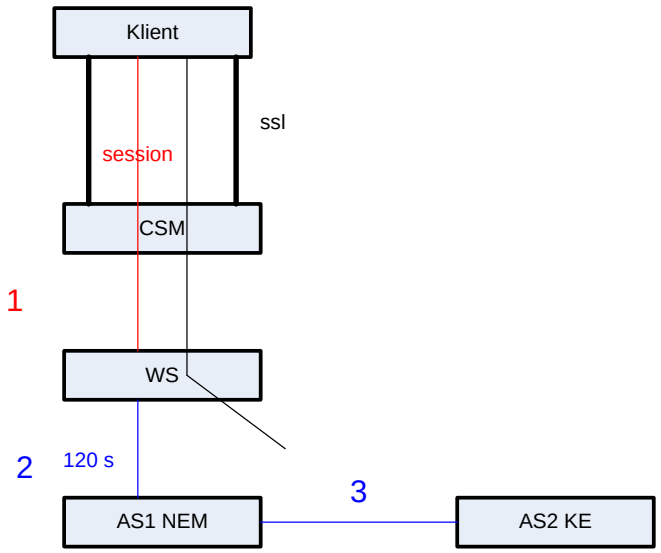


Obrázek 7 Schéma komunikace mezi aplikačními servery

Pokud Aplikační server aplikace A potřebuje informace od Aplikačního serveru aplikace B, musí mu předat winlogonname, který získal z http header ze serveru WebSealu. Aplikační server B předá identitu uživatele směrem k databázovému serveru. Tímto bude také zalogován přístup k datům – neanonymní přístup.

11. Reakční časy aplikací integrovaných do AAA portálu

Komunikace Klient – Aplikační server prochází přes komponentu Webseal. Maximální doba čekání klientské části na vyřízení požadavku je 120 sekund.



Obrázek 5 Schéma komunikace klientské části s aplikační vrstvou

1. Klient navazuje spojení (session) přes Content Switch na WebSeal.
2. WebSeal přesměruje komunikaci např. na aplikaci NEM (AS1 NEM).
3. Aplikace NEM vyžaduje další informace od jiné aplikace, např. KE (AS2 KE).

WebSeal má nastaveno udržovat relaci s aplikačním serverem po dobu 120 s. Pokud není do této doby vyřízen požadavek, WebSeal přeruší spojení a zobrazí uživateli chybové hlášení.

„Protože aplikace *XYZ* neodpověděla do 120 sekund, byla odpojena".

Pokud aplikace není schopná dotaz vyřídit synchronně v limitu 120 sekund musí být upravena na asynchronní volání/odpověď.

Doporučení pro aplikace:

a) optimalizace aplikace na nižší odezvu
Doporučené maximum pro výjimečně náročné operace je navrženo na 30 sekund, po 120 sekundách WebSeal spojení ukončí.

b) v případě, že aplikace závisí na spolupráci jiné aplikace, u níž nelze zaručit dobu odpovědi, neměly by být operace prováděné synchronně s požadavkem. Aplikační server může dát takovou operaci do fronty (např. JMS) a oznámit, že se operace vyřizuje. Dále je vhodné poskytnout uživateli rozhraní ke sledování průběhu operace/operací.

12. Integrace aplikací na platformě .NET do AAA portálu

Řešení, které zpřístupňuje standardním způsobem autentizační a autorizační údaje aplikacím v. NET, umožňuje standardní využití deklarativní Role Based Access Control a standardní autorizaci v ASP.NET. Přesný popis je možné předat na vyžádání.

V ČSSZ existuje schválený standard Programátorské konvence .NET. Tento dokument je určen pro programátory .NET. Jeho cílem je standardizovat kód pro zvýšení kvality vytvářených aplikací, kódu, snadnější zaškolení uživatelů (jednotné prostředí), snadnější převzetí kódu, rychlou orientaci v kódu při vyhledávání chyb a řešení problémů.

13. Více aplikací na jednom serveru

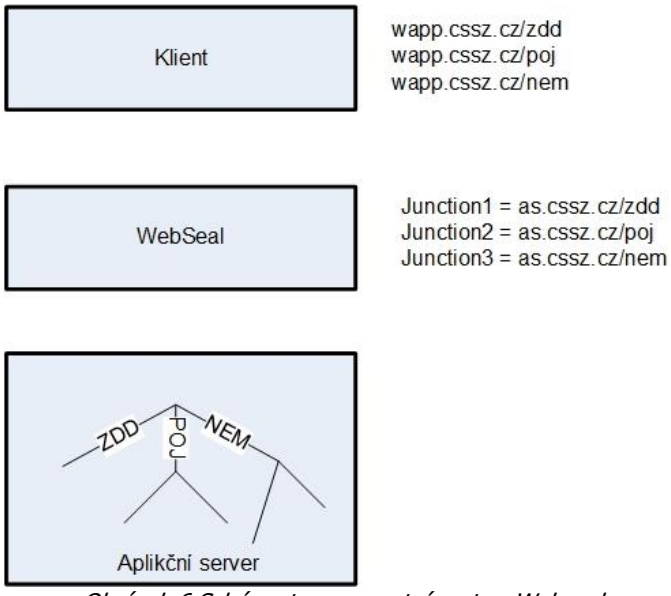
AAA portál ve výjimečných případech umožňuje, aby byla více než jedna aplikace na jednom aplikačním serveru (přesněji na skupině serverů pod stejnou virtuální adresou – VIP adresa). Aplikace mohou být odlišeny např. různým portem nebo různou virtuální IP adresou, na které běží.
Servery WebSeal směrují junction na VIP na portu 80. Jednou z možností, jak obsloužit více aplikací na jednom serveru, je tzv. transparentní cesta, která se dá nastavit ve vlastnostech junction. Pokud je na aplikačním serveru (skupině serverů pod stejnou virtuální adresou) více aplikací, je řešením právě nastavení transparentní cesty pro junction.

Aplikace musí tuto logiku v odkazech striktně dodržovat. Použití transparentní cesty zajistí zabezpečený přístup do jednotlivých aplikací na základě přiřazení logických rolí jednotlivým uživatelům.

13.1. Transparentní cesta

Pokud se nastaví u junction transparentní cesta, pak musí nakonfigurované jméno spojení odpovídat jménu podadresáře v adresářovém kořeni aplikačního serveru. Jméno spojení a jméno podadresáře se musí shodovat.

Jestliže je např. nakonfigurované jméno spojení /docs, pak všechny zdroje řízené tímto spojením, musí být umístěny na serveru typu back-end v podadresáři /docs.
Transparentní cesta a více aplikací na aplikačním serveru (farmě serverů) – příklad pro aplikace NEM, POJ a ZDD:



Obrázek 6 Schéma transparentní cesty - Webseal

Pokud je na aplikačním serveru jedna aplikace lze transparentní cestu nastavit na základě žádosti, v případě více aplikací na jednom aplikačním serveru je nastavení transparentní cesty vyžadováno pro každou aplikaci.

14. Organizační opatření při integraci aplikace do AAA portálu

- Požadavek na začlenění nové aplikace do AAA portálu pro všechna prostředí musí garant aplikace za ČSSZ předložit prostřednictvím spisu v DocuLive dle vnitřní organizační směrnice „AAA portál a jeho využití pro autentizaci a autorizaci uživatelů". Součástí spisu musí být soubory:
    1. "AAA Portál - Integrace nové / úprava stávající aplikace"
    2. příslušné xml soubory pro produkční a integrační prostředí, vygenerované pomocí nástroje A3Editor, který je k dispozici u garanta systému AAA Portál

  Součástí tohoto požadavku je:
  - průkazná informace o požadavku na integraci aplikace ze strany ČSSZ (formou e-mailu, resp. jeho kopie),
  - 3-písmenná nebo 4-písmenná zkratka aplikace pro jednoznačnou identifikaci v rámci aplikací a systémů ČSSZ, musí vycházet ze schváleného seznamu aplikací,
  - název aplikace pro zobrazení názvu a významu aplikace v rámci ISIM GUI, (stejný název musí být použit v příslušném XML souboru)
  - popis aplikace – pro zobrazení upřesňujícího významu, případně další pokyny
  - upozornění aplikace – nová verze ISIM GUI umožňuje zobrazovat upozornění, že byla vytvořena nepřípustná kombinace rolí nebo, že k dané roli je nutné přidat i roli z jiné aplikace apod.
  - definice potřebných aplikačních rolí (logické, fyzické, mapování fyzických rolí na role logické, dle potřeby role lokalizační a VIP) dle dokumentu Požadavky na aplikace pro integraci do AAA portálu, role a jejich struktura budou odsouhlaseny týmem AAA portálu,

- definice směrovacího bodu (junction) a IP adresy virtuálního aplikačního serveru příslušné aplikace,
- informace o vyjednaném a odsouhlaseném schvalovateli 2. úrovně (garantu aplikace) ze strany ČSSZ. Všechny role aplikace schvaluje vždy jeden schvalovatel. Jde o vedoucího příslušného útvaru ČSSZ, tým AAA portálu nebude toto ověřovat, bez definovaného schvalovatele nebude aplikace do produkčního prostředí AAA portálu zanesena.

- Další požadavky na změnu již integrované aplikace musí garant aplikace za ČSSZ předložit prostřednictvím spisu v DocuLive dle Směrnice AAA portál a jeho využití pro autentizaci a autorizaci uživatelů. Požadavky budou zpracovány v došlém pořadí dle předpokládaného režimu provozní podpory AAA portálu v rámci servisních oken AAA portálu (každé úterý a čtvrtek od 16:00 hod.)

- Na základě nastavení bude aplikace zobrazena v prostředí ISIM GUI, a bude možné pro ni přidělovat uživatelská oprávnění. Bude vydána aktualizovaná verze dokumentu Podpora pro nasazení (AAA_rollout_support) se seznamem aplikací a aplikačních rolí, který je k dispozici v aplikaci ISIM GUI.

- Informaci o provedení požadavku podá garant aplikace AAA příslušnému garantovi aplikace za ČSSZ.

- Aby mohla být aplikace integrována pod AAA portál, musí kromě funkčních požadavků definovaných v předchozích kapitolách splnit či specifikovat:

  - testování tlustého klienta, (pokud existuje), týmem AAA portálu,
  - způsob přebírání uživatelských účtů,
  - metody AAA API využívané aplikací, případně přidání dalších metod požadovaných aplikací.

  Kontrola výše uvedených bodů bude provedena pro každou aplikaci.

15. Metody identifikace aplikačního uživatele

Produkt IBM InfoSphere Guardium podporuje několik metod identifikace aplikačního uživatele v rámci komunikace mezi aplikačním a databázovým serverem. Tyto metody znamenají zásah do kódu aplikace na úrovni příslušného Frameworku nebo přímo kódu aplikace. Pro některé aplikace lze použít Database Helper (hook), který se zavolá při každé databázové transakci. Helper do transakce na začátek a konec přidá SQL query, kde identifikuje uživatele.

Pro každou aplikaci to tedy bude obdobné:
1. Zavést Helper, který přizpůsobí všechna DB volaní v rámci Framework, nebo to bude umět případně přímo aplikační server. V nejhorším případě vsak bude nutné upravit všechny databázové dotazy.
2. Do databázového volaní zavést dummy SQL query s identifikací uživatele, uživatel se získá z aplikačního kontextu nebo bude muset byt předán jako parametr.

15.1. Identifikace aplikačního uživatele pomocí Application Event API

Tato metoda využívá Guardium Application Event API, které umožňuje definovat hranice mezi aplikačními uživateli v jednom spojení do DB (session). Toto API reaguje na specifické no-op (bez účinku) SQL dotazy do tabulky dual. Pomocí těchto SQL dotazů a parametru „GuardAppEvent„ aplikace ohraničí začátek a konec DB operací jednoho konkrétního uživatele. Parametr „GuardAppEvent:Start" zajistí spuštění příslušného programu přes Guardium API, který zaznamená průběh celé komunikace až po její ukončení pomocí parametru „GuardAppEvent:Released", nebo do

ukončení spojení (session). Parametr „GuardAppEventUserName" obsahuje identifikátor uživatele a parametr „GuardAppEventType" obsahuje tříznakový kód aplikace ČSSZ.

Příklad syntaxe nastavení uživatele:
SELECT 'GuardAppEvent:Start', 'GuardAppEventUserName:uzivatel', 'GuardAppEventType:APP' FROM dual;

Pokud aktuálně ohlášený uživatel již nebude původcem následujících DB příkazů, je nutné kontext uživatele zrušit pomocí SQL dotazu:

SELECT 'GuardAppEvent:Released', 'GuardAppEventUserName:uzivatel', 'GuardAppEventType:APP' FROM dual;

Pokud po ukončení uživatele okamžitě navazuje činnost nového uživatele lze kontext přenastavit bez spouštění 'GuardAppEvent:Released'.

Příklad DB komunikace jednoho aplikačního uživatele v jednom spojení (session) tedy může vypadat takto:

SELECT 'GuardAppEvent:Start', 'GuardAppEventUserName:xxuzivatel1','GuardAppEventType:POJ' FROM dual;
..
-- SQL dotazy aplikace POJ v kontextu uživatele xxuzivatel1
..
SELECT 'GuardAppEvent:Released', 'GuardAppEventUserName:xxuzivatel1', 'GuardAppEventType:POJ' FROM dual;

DB příkazy mezi 'Start' a 'Released' budou v IBM Guardium logovány s textem "xxuzivatel1" v atributu "Event User Name" a textem "POJ" v atributu "Event Type" v entitě "Application Events". Hodnota z atributu "Event User Name" se též propíše do atributu "App User Name"

Timestamp	Full Sql	Event User Name	Event Type	App User Name
2016-01-19 15:38:46.0	SELECT 'Konec session' FROM dual			
2016-01-19 15:38:35.0	SELECT 'GuardAppEvent:Released', 'GuardAppEventUserName:xxuziv2', 'GuardAppEventType:POJ' FROM dual			
2016-01-19 15:38:24.0	SELECT 'Databazova aktivita uzivatele xxuziv2' FROM dual	xxuziv2	POJ	xxuziv2
2016-01-19 15:38:22.0	SELECT 'Databazova aktivita uzivatele xxuziv2' FROM dual	xxuziv2	POJ	xxuziv2
2016-01-19 15:38:18.0	SELECT 'GuardAppEvent:Start', 'GuardAppEventUserName:xxuziv2', 'GuardAppEventType:POJ' FROM dual	xxuziv2	POJ	xxuziv2
2016-01-19 15:38:13.0	SELECT 'GuardAppEvent:Released', 'GuardAppEventUserName:xxuziv1', 'GuardAppEventType:POJ' FROM dual			
2016-01-19 15:38:08.0	SELECT 'Databazova aktivita uzivatele xxuziv1' FROM dual	xxuziv1	POJ	xxuziv1
2016-01-19 15:38:06.0	SELECT 'Databazova aktivita uzivatele xxuziv1' FROM dual	xxuziv1	POJ	xxuziv1
2016-01-19 15:38:01.0	SELECT 'GuardAppEvent:Start', 'GuardAppEventUserName:xxuziv1', 'GuardAppEventType:POJ' FROM dual	xxuziv1	POJ	xxuziv1
2016-01-19 15:37:55.0	SELECT 'GuardAppEvent:Released', 'GuardAppEventUserName:xxuziv2', 'GuardAppEventType:POJ' FROM dual			
2016-01-19 15:37:51.0	SELECT 'Databazova aktivita uzivatele xxuziv2' FROM dual	xxuziv2	POJ	xxuziv2
2016-01-19 15:37:50.0	SELECT 'Databazova aktivita uzivatele xxuziv2' FROM dual	xxuziv2	POJ	xxuziv2
2016-01-19 15:37:49.0	SELECT 'Databazova aktivita uzivatele xxuziv2' FROM dual	xxuziv2	POJ	xxuziv2
2016-01-19 15:37:45.0	SELECT 'GuardAppEvent:Start', 'GuardAppEventUserName:xxuziv2', 'GuardAppEventType:POJ' FROM dual	xxuziv2	POJ	xxuziv2
2016-01-19 15:37:40.0	SELECT 'Databazova aktivita nezavisla na cinnostech uzivatelu' FROM dual			
2016-01-19 15:37:39.0	SELECT 'Databazova aktivita nezavisla na cinnostech uzivatelu' FROM dual			
2016-01-19 15:37:38.0	SELECT 'Databazova aktivita nezavisla na cinnostech uzivatelu' FROM dual			
2016-01-19 15:37:34.0	SELECT 'GuardAppEvent:Released', 'GuardAppEventUserName:xxuziv1', 'GuardAppEventType:POJ' FROM dual			
2016-01-19 15:37:28.0	SELECT 'Databazova aktivita uzivatele xxuziv1' FROM dual	xxuziv1	POJ	xxuziv1
2016-01-19 15:37:28.0	SELECT 'Databazova aktivita uzivatele xxuziv1' FROM dual	xxuziv1	POJ	xxuziv1
2016-01-19 15:37:27.0	SELECT 'Databazova aktivita uzivatele xxuziv1' FROM dual	xxuziv1	POJ	xxuziv1
2016-01-19 15:37:26.0	SELECT 'Databazova aktivita uzivatele xxuziv1' FROM dual	xxuziv1	POJ	xxuziv1
2016-01-19 15:37:21.0	SELECT 'GuardAppEvent:Start', 'GuardAppEventUserName:xxuziv1', 'GuardAppEventType:POJ' FROM dual	xxuziv1	POJ	xxuziv1
2016-01-19 15:37:06.0	SELECT 'Zacatek session - bez kontextu uzivatele' FROM dual			

Records 1 to 24 of 24

Obrázek 8 Příklad komunikace Guardium Application Event API

Při změně aplikačního uživatele využívajícího dané spojení do DB, se nejdříve spustí 'GuardAppEvent:Released' nebo 'GuardAppEvent:Start' s loginem nového uživatele.

Při změně spojení využívaného aplikačním uživatelem (přívlastek) se na starém spojení spustí příkaz 'GuardAppEvent:Released', nebo 'GuardAppEvent:Start' s novým uživatelem starého spojení a na novém spojení se spustí příkaz 'GuardAppEvent:Start'.

15.2. Identifikace aplikačního uživatele pomocí volání uložených procedur (stored procedure)

Tato metoda ohlášení uživatele aplikačního serveru je obdobou Application Event API. Místo noop SQL dotazu je identita uživatele aplikačního serveru vložena jako parametr do volání uložených procedur. Komunikace může probíhat například takto:

BEGIN set_application_property('user_name', 'xxuzivatel1', 'application', 'POJ'); END;
..
-- SQL dotazy aplikace POJ v kontextu uživatele xxuzivatel1
..
BEGIN del_application_property('user_name', 'xxuzivatel1', 'application', 'POJ'); END;

DB příkazy mezi 'set_application_property' a 'del_application_property' budou v IBM Guardium logovány s textem "xxuzivatel1" v atributu "Event User Name" a "POJ" v atributu "Event Type" v entitě "Application Events". Hodnota z atributu "Event User Name" se též propíše do atributu "App User Name"

Timestamp	Full Sql	Event User Name	Event Type	App User Name
2016-01-19 15:28:35.0	SELECT 'Konec session' FROM dual			
2016-01-19 15:28:27.0	SELECT 'Databazova aktivita nezavisla na cinnostech uzivatelu' FROM dual			
2016-01-19 15:28:24.0	SELECT 'Databazova aktivita nezavisla na cinnostech uzivatelu' FROM dual			
2016-01-19 15:28:22.0	SELECT 'Databazova aktivita nezavisla na cinnostech uzivatelu' FROM dual			
2016-01-19 15:28:19.0	SELECT 'Databazova aktivita nezavisla na cinnostech uzivatelu' FROM dual			
2016-01-19 15:28:14.0	BEGIN del_application_property('user_name', 'xxuziv2', 'application', 'POJ'); END;	xxuziv2	POJ	xxuziv2
2016-01-19 15:28:06.0	SELECT 'Databazova aktivita uzivatele xxuziv2' FROM dual	xxuziv2	POJ	xxuziv2
2016-01-19 15:28:03.0	SELECT 'Databazova aktivita uzivatele xxuziv2' FROM dual	xxuziv2	POJ	xxuziv2
2016-01-19 15:27:58.0	SELECT 'Databazova aktivita uzivatele xxuziv2' FROM dual	xxuziv2	POJ	xxuziv2
2016-01-19 15:27:49.0	BEGIN set_application_property('user_name', 'xxuziv2', 'application', 'POJ'); END;	xxuziv2	POJ	xxuziv2
2016-01-19 15:27:40.0	BEGIN del_application_property('user_name', 'xxuziv1', 'application', 'POJ'); END;	xxuziv1	POJ	xxuziv1
2016-01-19 15:27:34.0	SELECT 'Databazova aktivita uzivatele xxuziv1' FROM dual	xxuziv1	POJ	xxuziv1
2016-01-19 15:27:29.0	BEGIN set_application_property('user_name', 'xxuziv1', 'application', 'POJ'); END;	xxuziv1	POJ	xxuziv1
2016-01-19 15:27:18.0	SELECT 'Zacatek session - bez kontextu uzivatele' FROM dual			

Records 1 to 14 of 14

Obrázek 9 Příklad komunikace Guardium Stored Procedure

Tato metoda je vhodná zejména pro aplikace, které nejsou dostatečně flexibilní pro nasazení Application Event API, a již obsahují volání procedur s uživatelským jménem. Například aplikace PSL spouští proceduru DBMS_SESSION.SET_IDENTIFIER s identitou uživatele před každým databázovým dotazem. Informaci o ukončení činnosti uživatele neposílá.

Příklad komunikace aplikace PSL:

DBMS_SESSION_SET_IDENTIFIER('xxuzifra::1234567890');
UPDATE tbPER SET PER_Lock=1, PER_DatLock=sysdate, PER_UsrLock='uzifra_00/XXXXXXXX' WHERE tbPER.PER_Id=12345
DBMS_SESSION_SET_IDENTIFIER('xxuzifra::0987654321');
SELECT * FROM tbRIZ
DBMS_SESSION_SET_IDENTIFIER('xxuzifra::0987654321');
UPDATE tbPER SET PER_Lock=0, PER_DatLock=sysdate, PER_UsrLock='' WHERE tbPER.PER_Id=12345
DBMS_SESSION_SET_IDENTIFIER('xxuzifra::5432109876');
UPDATE tbPER SET PER_Lock=1, PER_DatLock=sysdate, PER_UsrLock='xxuzifra_05/YYYYYYYY' WHERE tbPER.PER_Id=67890

Příklad jak by měla vypadat komunikace ve dvou databázových spojeních, která si „přehazují" pracující aplikační uživatele (connection pool) v pseudo SQL.

Čas spuštění	Spojení 1	Spojení 2
14:24:00	Spojení 1 navázáno	
14:24:01	Select ‘test_spojeni_na_db’ from dual	
14:24:05	SELECT 'GuardAppEvent:Start', 'GuardAppEventUserName:Uziv_1', 'GuardAppEventType:APP1' FROM dual;	
14:24:06	Update REFERENT set Zamek=ANO where UID=1234567890	
14:24:30	Select * from PRIPAD where Pracovník=Uziv_1	
14:24:31	Select * from CIS_TYPU_PRIPADU where typ in (51,93,135)	
14:24:49	Update PRIPAD Set STAV=‘ZAMITNUTO‘ Where ID=125456	
14:25:15	Update PRIPAD Set STAV=‘SCHVALENO‘ Where ID=154586	
14:25:49	Update PRIPAD Set STAV=‘ODLOZENO‘ Where ID=369854	
14:27:12	Update PRIPAD Set STAV=‘SCHVALENO‘ Where ID=456741	
14:28:48	Update PRIPAD Set STAV=‘ODLOZENO‘ Where ID=658891	
14:30:00		Spojení 2 navázáno
14:30:01		Select ‘test_spojeni_na_db’ from dual
14:30:11	Update PRIPAD Set STAV=‘ZAMITNUTO‘ Where ID=156874	SELECT 'GuardAppEvent:Start', 'GuardAppEventUserName:Uziv_2', 'GuardAppEventType:APP1' FROM dual;
14:30:12		Update REFERENT set Zamek=ANO where UID=4444333221
14:30:16		Select * from PRIPAD where Pracovník=Uziv_2
14:30:58	Update PRIPAD Set STAV=‘SCHVALENO‘ Where ID=854796	Select * from CIS_TYPU_PRIPADU where typ in (256,258,263)
14:31:18		Update PRIPAD Set STAV=‘PRIJATO‘ Where ID=785324
14:32:15	Update REFERENT set Zamek=NE where UID=1234567890	
14:32:16	SELECT 'GuardAppEvent:Released' 'GuardAppEventUserName:Uziv_1', 'GuardAppEventType:APP1' FROM dual	Update PRIPAD Set STAV=‘ZAMITNUTO‘ Where ID=654878
14:33:25		Update PRIPAD Set STAV=‘DODAT_INFO‘ Where ID=854732
14:33:26		SELECT 'GuardAppEvent:Released', 'GuardAppEventUserName:Uziv_2', 'GuardAppEventType:APP1' FROM dual
14:33:27	SELECT 'GuardAppEvent:Start', 'GuardAppEventUserName:Uziv_2', 'GuardAppEventType:APP1' FROM dual;	
14:34:12	Update PRIPAD Set STAV=‘DODAT_INFO‘ Where ID=812532	
14:34:59	Update PRIPAD Set STAV=‘ZAMITNUTO‘ Where ID=812874	Select count(*) from PRIPAD where STAV=ZAMITNUTO
14:35:00		INSERT into STATISTIKA_ZAMITNUTYCH values (‚14:34:59‘, 4)
		Spojení 2 uzavřeno
14:35:38	Update PRIPAD Set STAV=‘PRIJATO‘ Where ID=458963	
14:36:47	Update PRIPAD Set STAV=‘PRIJATO‘ Where ID=788952	
14:38:08	Update PRIPAD Set STAV=‘PRIJATO‘ Where ID=457415	
14:39:55	Update REFERENT set Zamek=NE where UID=4444333221	
14:40:15	SELECT 'GuardAppEvent:Released', 'GuardAppEventUserName:Uziv_2', 'GuardAppEventType:APP1' FROM dual	
14:40:16	Spojení 1 uzavřeno	

ČSSZ

16. Správa technologických účtu v AAA portálu

AAA portál zabezpečuje komplexní správu technologických účtů, tzn. takových přístupových účtů, které využívá konkrétní služba (proces) v aplikaci a nejsou bezprostředně odvozeny od uživatele v systému HR SAP.

Rozdělení podle typů účtu

- Administrátorské účty
- Servisní účty
- Provozní účty
- Lokální účty

Administrátorský účet
- přístupový účet, který využívají pouze administrátoři cílových systému
- je striktně oddělen od uživatelského účtu
- je přidělován pověřeným zaměstnancům na základě Organizačního řádu

Servisní účet
- účet, který není přímo spojen s reálnou identitou,
- zřizuje se pro specifické účely aplikace, provozu nebo testování
- přidělují se technologické oprávnění (např. skupiny v AD, apod.),
- autentizace jménem a heslem nebo certifikátem,
- je reprezentován virtuální identitou, která má konkrétního vlastníka (nadřízeného),

Každý servisní účet musí mít definovaného svého vlastníka/garanta. Zřízení technologického účtu je možné na základě vyplněného formuláře „Žádost o technologický účet", který je pro uživatele ČSSZ dostupný v ISIM GUI.

Technologická role
- je reprezentována skupinou v AD nebo ISIM,
- na role/skupiny se vážou práva v konkrétním systému,
- žádost o založení/zrušení přes service desk (xls formulář),
- role založí určená skupina administrátorů ISIM.

17. Schvalovací doložka a platnost standardu

Standard byl schválen dne 25 -02- 2016 na základě spisu čj. 502 - 7001 - 15.12.2015/4260.

.....
Ing. Milan Shrbený
ředitel sekce IKT

Standard nabývá účinnosti dnem schválení ředitelem sekce IKT (5).