

**VALSTS INFORMĀCIJAS SISTĒMU SAVIETOTĀJA  
(VISS) UN VIENOTĀ VALSTS UN PAŠVALDĪBU  
PAKALPOJUMU PORTĀLA WWW.LATVIJA.LV  
PILNVEIDOŠANA UN UZTURĒŠANA**

**DATU APMAIŅAS IZVEIDES VADLĪNIJAS**

VADLĪNIJAS

**VRAA-13\_7\_17\_41-VISS\_2016-D\_APM-VDL**

28.02.2024. versija 2.23



Rīgā 2024

# DOKUMENTA IDENTIFIKĀCIJA

<b>DOKUMENTA ID:</b>	VRAA-13_7_17_41-VISS_2016-D_APM-VDL-V2.23-28.02.2024.
<b>DOKUMENTA NOSAUKUMS:</b>	Valsts informācijas sistēmu savietotāja (VISS) un Vienotā valsts un pašvaldību pakalpojumu portāla www.latvija.lv pilnveidošana un uzturēšana. 3.daļa "VISS un Portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju". Datu apmaiņas izveides vadlīnijas. Vadlīnijas.
<b>DOKUMENTA KODS:</b>	VRAA-13_7_17_41-VISS_2016-D_APM-VDL
<b>VERSIJA:</b>	Versija 2.23, Laidiens 28.02.2024.

## SASKAŅOJUMI

ORGANIZĀCIJA	VĀRDS, UZVĀRDS, AMATS	DATUMS	PARAKSTS
Valsts reģionālās attīstības aģentūra	Atbildīgā persona no Pasūtītāja puses		
SIA "ABC software"	J.Korņijenko, projekta vadītājs par tehniskiem jautājumiem no Izpildītāja puses	06.10.2023.	
SIA "ABC software"	M.Pētersons, projekta vadītājs par administratīviem jautājumiem no Izpildītāja puses	06.10.2023.	
SIA "ABC software"	E.Blumberga, projekta kvalitātes kontroles vadītāja	06.10.2023.	

# IZMAIŅU VĒSTURE

VERSĪJA	DATUMS	APRAKSTS	ORGANIZĀCIJA	AUTORS
1.0	09.10.2015.	Izveidota dokumenta sākotnējā versija.	SIA „ABC software”	J.Korņijenko
1.01	30.11.2015	Iestrādāti Pasūtītāja iesniegtie komentāri	SIA „ABC software”	J.Korņijenko
1.02	16.10.2017	Dokuments pārņemts uz VISS_2016 veidni	SIA „ABC software”	M.Gasparoviča-Asīte
1.03	10.12.2018	Drošības standartu aktualizācija – 23. saistītais avots, 1. un 3.pielikums. Gramatiski labojumi visā dokumentā.	SIA „ABC software”	J.Korņijenko, M.Gasparoviča-Asīte
2.00	24.09.2019	Papildinājumi saistībā ar IP Nr. 26 “VISS Pieprasījumu servisa modernizācija”.	SIA „ABC software”	A.Mačiņš
2.01	01.11.2019	Papildināts atbilstoši pasūtītāja komentāriem IP Nr. 26 “VISS Pieprasījumu servisa modernizācija” ietvaros.	SIA „ABC software”	E.Stāmurs
2.02	13.11.2019	Papildināta sadaļa 2.1. API Pārvaldnieka izmantošana atbilstoši pasūtītāja komentāriem.	SIA „ABC software”	E.Stāmurs
2.03	26.11.2019	Izņemtas salauztās atsauces uz attēliem.	SIA „ABC software”	E.Stāmurs
2.04	22.04.2020	Papildināta nodaļa “5.4.4. Lietojuma pielāgošana servisa izsaukšanai” (izmaiņas “Pieprasījumu servisa modernizācijas 2.posms” ietvaros).	SIA „ABC software”	E.Stāmurs
2.05	25.05.2020	Papildināta sadaļa 5.4.4. API Pārvaldnieka izmantošana atbilstoši pasūtītāja komentāriem.	SIA „ABC software”	E.Stāmurs
2.06	05.06.2020	Papildināta sadaļa 5.4.1. Pieejas tiesību pieprasīšana atbilstoši pasūtītāja komentāriem.	SIA „ABC software”	E.Stāmurs
2.07	06.07.2020	Papildinātas instrukcijas attiecībā uz EDK izmantošanu datu apmaiņās atbilstoši 07.06.2020 sapulcei ar pasūtītāju.	SIA „ABC software”	E.Stāmurs
2.08	15.07.2020	Pievienota nodaļa “5.5. Transakciju API izmantošana” atbilstoši pasūtītāja komentāriem.	SIA „ABC software”	E.Stāmurs
2.09	13.08.2020	Sniegtas atbildes uz pasūtītāja komentāriem.	SIA „ABC software”	E.Stāmurs
2.10	26.08.2020	Precizēta nodaļa “5.5. Transakciju API izmantošana” atbilstoši pasūtītāja komentāriem.	SIA „ABC software”	E.Stāmurs
2.11	07.09.2020	Nodalījumam “5. API Pārvaldnieks” pievienots API Pārvaldnieka PR dokumenta saturs (globālas izmaiņas), kā arī atbilstoši IP veiktajām izmaiņām veiktas korekcijas saturā (PFAS operāciju vietā tiek izmantoti scopes).	SIA „ABC software”	A.Mačiņš
2.12	14.09.2020	Precizēta nodaļa “5.5. Transakciju API izmantošana” atbilstoši pasūtītāja komentāriem.	SIA „ABC software”	V.Orlovs
2.13	01.10.2020	Papildinājumi nodaļā “5. API Pārvaldnieks” līguma “Iestādes darbinieka darba vietas pilnveide” ietvaros	SIA „ABC software”	A.Mačiņš E.Stāmurs

2.14	02.11.2020	Veikti precizējumi atbilstoši Pasūtītāja komentāriem	SIA „ABC software”	E.Stāmurs
2.15	10.11.2020	Veikti precizējumi atbilstoši Pasūtītāja komentāriem	SIA „ABC software”	E.Stāmurs
2.16	12.11.2020	Veikti precizējumi atbilstoši Pasūtītāja komentāriem	SIA „ABC software”	E.Stāmurs
2.17	06.01.2021	Aktualizēta nodaļa “5.6. API izsaukumu piemēri”.	SIA „ABC software”	J.Gekišs
2.18	29.01.2021	Pilnveidota informācija par lauka “Visible on Developer portal” aizpildīšanu.	SIA „ABC software”	E.Stāmurs
2.19	01.11.2021	Aktualizēta nodaļa “5. API Pārvaldnieks” līguma Nr. 5-1/2021-34/EIS-54 “API pārvaldniekā izmantojamo WSO2 komponentu versiju aktualizāciju uz versiju v4.0.0.” ietvaros.	SIA „ABC software”	E.Stāmurs
2.20	16.11.2021	Veikti precizējumi atbilstoši Pasūtītāja komentāriem	SIA „ABC software”	E.Stāmurs
2.21	01.12.2021	Sniegtas atbildes uz Pasūtītāja komentāriem.	SIA „ABC software”	E.Stāmurs
2.22	06.10.2023	Aktualizēts darba uzdevuma “API Pārvaldnieka funkcionālie uzlabojumi VRAA EIS 2020/13/AK/CI-118-2-PKP” ietvaros.	SIA „ABC software”	E.Stāmurs
2.23	28.02.2024	Aktualizētas drošības talonu iegūšanas adreses 5. nodaļā	Valsts reģionālās attīstības aģentūra	N.Priževoits

## Satura rādītājs

<b>ATTĒLU SARAKSTS .....</b>	<b>8</b>
<b>1. IEVADS.....</b>	<b>10</b>
<b>1.1. Termini un pieņemtie apzīmējumi .....</b>	<b>10</b>
1.1.1. Termini .....	10
1.1.2. Apzīmējumi.....	11
<b>1.2. Saistība ar citiem dokumentiem .....</b>	<b>12</b>
<b>1.3. Dokumenta pārskats .....</b>	<b>13</b>
<b>2. VISS NODROŠINĀTĀS DATU APMAIŅAS.....</b>	<b>14</b>
<b>2.1. API Pārvaldnieka izmantošana .....</b>	<b>15</b>
<b>2.2. Pieprasījumu servisa izmantošana .....</b>	<b>17</b>
<b>2.3. DIT izmantošana datu apmaiņai .....</b>	<b>18</b>
<b>2.4. Datu servisi un nozares integratori.....</b>	<b>19</b>
<b>3. VISS PIEPRASĪJUMU SERVISS .....</b>	<b>20</b>
<b>3.1. Pieejamo valsts reģistru servisu katalogs.....</b>	<b>20</b>
<b>3.2. Tiesību pieprasīšanas un piešķiršanas kārtība Integrācijas IS servisu izsaukšanai .....</b>	<b>21</b>
<b>3.3. Pieprasījumu servisa lietošanas scenārijs .....</b>	<b>21</b>
<b>3.4. Pieprasījumu servisa darbības principi .....</b>	<b>22</b>
<b>3.5. StartTransaction metode .....</b>	<b>25</b>
<b>3.6. EndTransaction metode.....</b>	<b>26</b>
<b>3.7. Sinhronie pieprasījumi.....</b>	<b>26</b>
<b>3.8. Asinhronie pieprasījumi.....</b>	<b>28</b>
<b>3.9. SaveResponse.....</b>	<b>30</b>
<b>3.10. „GetStatus” metode .....</b>	<b>31</b>
<b>3.11. Poll .....</b>	<b>31</b>
3.11.1. Pieprasījumu servisa kļūdas .....	32
<b>3.12. Datu struktūru apraksts .....</b>	<b>33</b>
3.12.1. IVISRequest struktūra .....	33
3.12.2. IVISResponse struktūra.....	35
3.12.2.1. IVISResponse struktūra veiksmīgas izpildes gadījumā.....	35
3.12.2.2. IVISResponse struktūra neveiksmīgas izpildes gadījumā.....	36

<b>3.13. Universālā Pieprasījuma servisa asinhronās saskarnes izmantošana.....</b>	<b>36</b>
3.13.1. Sinhronā Integrācijas IS servisa asinhronais izsaukums.....	38
3.13.2. Pieprasījuma ievietošana un rezultāta saņemšana, izmantojot BizTalk adapteri.....	38
<b>4. DATU IZPLATĪŠANAS TĪKLS .....</b>	<b>41</b>
<b>4.1. Nepieciešamās tiesības DIT izmantošanai .....</b>	<b>42</b>
<b>4.2. DIT lietošanas scenārijs.....</b>	<b>42</b>
4.2.1. Parakstīšanās uz esošu citas iestādes kanāla plūsmu .....	42
4.2.2. Jaunā kanāla izveidošana .....	42
<b>5. API PĀRVALDNIEKŠ .....</b>	<b>43</b>
<b>5.1. API Pārvaldnieka darbības principi.....</b>	<b>44</b>
<b>5.2. API Pārvaldnieka lietošanas scenārijs.....</b>	<b>44</b>
<b>5.3. Izstrādātāju portāls (Developer portal) .....</b>	<b>45</b>
5.3.1. Publicēta biznesa servisa izmantošana .....	45
5.3.1.1. Publicēto API pārlūkošana .....	46
5.3.1.2. Pieejas tiesību pieprasīšana no VRAA.....	47
5.3.1.3. Klienta lietojuma reģistrācija.....	47
5.3.1.4. Servisu abonēšana.....	49
5.3.1.5. Servisa izmantošanas tiesību pieprasīšana no datu devēja .....	50
5.3.1.6. Lietojuma pielāgošana servisa izsaukšanai .....	51
5.3.2. Publicēta biznesa servisa izmantošanas pārtraukšana .....	54
<b>5.4. Publicētāju portāls (API Publisher) .....</b>	<b>54</b>
5.4.1. Pieejas tiesību pieprasīšana no VRAA .....	54
5.4.2. Servisu (API) izstrāde.....	55
5.4.3. Servisu (API) reģistrēšana un publicēšana .....	55
5.4.3.1. Aizsargāts API.....	56
5.4.3.2. REST servisu reģistrācija .....	57
5.4.3.3. Integrācijas Servisu reģistrācija .....	58
5.4.3.4. GraphQL servisu reģistrācija.....	58
5.4.3.5. Scope pievienošana .....	59
5.4.4. Pieejas servisam (API) piešķiršana .....	62
5.4.5. Pieejas servisam (API) apturēšana.....	63
5.4.6. Transakcijas obligātuma konfigurēšana .....	63
<b>5.5. Transakciju API izmantošana .....</b>	<b>65</b>
5.5.1. Transakcijas uzsākšanas metode.....	65
5.5.1.1. Piemēri ar datu nodošanu body daļā .....	66
5.5.1.2. Piemēri ar datu nodošanu query daļā .....	67
5.5.2. Transakcijas apturēšanas metode .....	68
5.5.2.1. Piemēri ar datu nodošanu body daļā .....	68
5.5.2.2. Piemēri ar datu nodošanu query daļā .....	70
5.5.3. Transakcijas statusa metode .....	71

<b>5.6. API izsaukumu piemēri .....</b>	<b>73</b>
5.6.1. Priekšnosacījumi .....	73
5.6.2. Izsaukumu Curl piemēri .....	73
5.6.2.1. OAuth2 drošības talona pieprasījums bez scope aizpildīšanas .....	74
5.6.2.2. OAuth2 drošības talona pieprasījums ar norādītiem scope .....	74
5.6.2.3. Datu izgūšana par drošības talonu .....	75
5.6.2.4. REST servisa izsaukums .....	77
5.6.2.5. SOAP servisa izsaukšana .....	78
5.6.3. .NET piemēri .....	81
5.6.3.1. SOAP un REST servisu izsaukšana un talona pieprasīšana ar sertifikātu. ..	81
5.6.3.2. SOAP un REST servisu izsaukšana un references talona pieprasīšana .....	82
5.6.4. Java piemēri .....	83
5.6.5. PHP piemēri .....	85
<b>1. PIELIKUMS. IS AUTENTIFIKĀCIJAS UN AUTORIZĀCIJAS PIEPRASĪJUMS NO DROŠĪBAS TALONU SERVISA .....</b>	<b>88</b>
<b>2. PIELIKUMS. ATBILDE UZ IS AUTENTIFIKĀCIJAS UN AUTORIZĀCIJAS PIEPRASĪJUMU .....</b>	<b>90</b>
<b>3. PIELIKUMS. PIEPRASĪJUMU SERVISA BIZNESĀ TRANSĀKCIJAS IZSAUKŠANAS PIEPRASĪJUMS.....</b>	<b>92</b>
<b>4. PIELIKUMS. PIEPRASĪJUMU SERVISA BIZNESĀ TRANSĀKCIJAS IZSAUKŠANAS PIEPRASĪJUMA REZULTĀTS.....</b>	<b>94</b>
<b>5. PIELIKUMS. INTEGRĀCIJAS IS SERVISA IZSAUKUMS.....</b>	<b>95</b>
<b>6. PIELIKUMS. INTEGRĀCIJAS IS SERVISA IZSAUKUMA REZULTĀTS.....</b>	<b>97</b>
<b>7. PIELIKUMS. BIZNESĀ PROCESU MODELĒŠANAS VALODAS NOTĀCIJA .....</b>	<b>98</b>

## Attēlu saraksts

1.attēls. VISS centrālās datu apmaiņas platformas galvenās komponentes .....	14
2.attēls. Biznesa scenārijs API Pārvaldnieka izmantošana sinhrono pieprasījumu realizācijai 16	
3.attēls. Biznesa scenārijs API pārvaldnieka servisa izmantošana sinhrono pieprasījumu realizācijai, kad sagatavojamā atbilde ir lielāka par 4 MB .....	16
4.attēls. Biznesa scenārijs Pieprasījumu servisa izmantošana sinhrono pieprasījumu realizācijai .....	17
5.attēls. Biznesa scenārijs Pieprasījumu servisa izmantošana asinhrono pieprasījumu realizācijai .....	17
6.attēls. Biznesa scenārijs Pieprasījumu servisa izmantošana sinhrono pieprasījumu realizācijai, kad sagatavojamā atbilde ir lielāka par 4 MB .....	18
7.attēls. Biznesa scenārijs DIT izmantošana datu izplatīšanai .....	18
8.attēls. IS servisu katalogs .....	20
9.attēls. Pieprasījumu servisā publicēta Integrācijas IS servisa izsaukums, izmantojot sertifikātu vai VISS lietotāja vārdu/paroli .....	22
10.attēls. Pieprasījuma servisa detalizēts plānojums .....	23
11.attēls. StartTransaction metode .....	25
12.attēls. Sinhrons pieprasījums .....	27
13.attēls. Pieprasījuma pāradresācija .....	28
14.attēls. Asinhrons pieprasījums .....	29
15.attēls. SaveResponse serviss .....	30
16.attēls. Poll serviss .....	32
17.attēls. Tilts starp sinhroniem un asinhroniem izsaukumiem .....	37
18.attēls. Saskarnes darbības algoritms .....	38
19.attēls. SaveResponse paplašināšana .....	39
20.attēls. Datu izplatīšanas kanāls (DIK).....	41
21.attēls. Datu savākšanas kanāls (DSK) .....	41
22.attēls. API Pārvaldnieka risinājuma koncepts.....	43
23.attēls. API Pārvaldniekā publicēta biznesa servisa izsaukums.....	45
24.attēls. Izstrādātāju portālā (Developer portal) publicētie API .....	46
25.attēls. Rest servisa detalizētais skats.....	47
26.attēls. Izstrādātāju portālā (Developer portal) reģistrētie klienta lietojumi .....	48
27.attēls. Klienta lietojuma izveides forma .....	48
28.attēls. Izveidotā klienta lietojuma/PFAS uzticamās puses talona izgūšanas dati .....	49
29.attēls. Publicēta servisa (API) kartiņa.....	49
30.attēls. Publicēta servisa (API) piesaiste lietojumam - abonēšana.....	50
31.attēls. Datu devēja kontaktinformācija.....	50
32.attēls. Datu devēja kontaktinformācija.....	51
33.attēls. PFAS AUTH lietojuma PFAS uzticamās puses detalizētais skats .....	52
34.attēls. Klienta lietojuma detalizētais skats API Pārvaldnieka saskarnē .....	52
35.attēls. REST servisa swagger definīcija .....	53
36.attēls. SOAP servisa WSDL lejupielāde .....	53
37.attēls. Izstrādātāju portālā (Developer portal) reģistrētā klienta lietojuma kartiņa .....	54
38.attēls. API Publicētāja saskarne.....	55
39.attēls. API reģistrēšanas veidi .....	56
40.attēls. Oauth2 mediatora pielikšana .....	57



41.attēls. Autoritātes identifikators leštādes un struktūrvienības klasifikatorā.....	57
42.attēls. Servisa rediģēšanas skata atvēršana .....	60
43.attēls. Scope datu aizpildīšana.....	61
44.attēls. Scope pievienošana .....	61
45.attēls. Transakcijas uzsākšana .....	65
46.attēls. Koda izmaiņas JWT talona pieprasīšanai ar sertifikātu .....	81
47.attēls. .NET piemēra izpildes rezultāts .....	82
48.attēls. Koda izmaiņas references talona pieprasīšanai.....	83
49.attēls. .NET piemēra izpildes rezultāts .....	83
50.attēls. Koda izmaiņas talona veida maiņai, attēla pieprasījums ar sertifikātu.....	84
51.attēls. Java piemēru izpildes rezultāts.....	85
52.attēls. Koda izmaiņas talona veida maiņai, attēla pieprasījums ar sertifikātu.....	86
53.attēls. PHP piemēra izpildes rezultāts .....	87

## 1. Ievads

Šī dokumenta mērķis ir aprakstīt datu apmaiņas iespējas “sistēma-sistēma integrācijas” un e-pakalpojuma izstādes gadījumos izmantojot Valsts informācijas sistēmu savietotāju (VISS) kā integrācijas platformu no datu saņēmēja (klienta) viedokļa.

Dokumenta auditoriju veido:

- VRAA darbinieki, kuri ir atbildīgi par VISS nodevumu un integrācijas procesu pieņemšanu un izvērtēšanu;
- Valsts, pašvaldību un komerciestāžu izstrādātāji un sistēmu uzturētāji;
- Valsts un pašvaldību iestāžu reģistru e-pakalpojumu izstrādātāji;
- VISS administratori un uzturētāji.

Paredzēts, ka šie norādījumi tiks pastāvīgi papildināti ar jauniem ieteikumiem, t.sk. arī no dokumenta lasītāju puses.

### 1.1. Terminu un pieņemtie apzīmējumi

#### 1.1.1. Terminu

Dokumentā izmantotie termini ir apkopoti 1.tabulā.

1.tabula

#### Termini

TERMINS	APRAKSTS
Informācijas sistēma	Datu bāzes, lietotāja programmatūras, procedūru un funkciju kopums un to savstarpējās attiecības, kas nodrošina sistēmas funkcionēšanu atbilstoši tās pamatmērķiem un pamatuzdevumiem.
Integrācijas IS serviss	Standartizēts informācijas sistēmas pieejas veids (serviss), kas pie informācijas resursa nodrošina ārējo saskarni. Integrācijas IS serviss parasti sastāv no divām daļām: servera daļas un klienta daļas. Servera daļa – IS servisa realizācija iestādes pusē (realizācijas veida izvēle ir iestādes kompetencē, saskarnes aprakstīšanas vēlamais standarts ir WSDL); klienta daļa – IS servisa saskarnes realizācija VISS IS servisu kataloga pusē (nodrošina nepieciešamo auditāciju, kontroli).
Uz servisiem orientēta arhitektūra	Uz servisiem orientēta arhitektūra sastāv no virknes brīvi savienotu, sadalītu, lietišķu komponentu, kuras kopā var ietilpt vienā ar uzņēmumu saistītājā procesā.
Vārdtelpa (namespace)	XML shēmas vārdtelpa nosaka noslēgtu nosaukumu kopu vai vietu, kur shēma (nosaukumu kopa) tiek glabāta. Vārdtelpas definē ar URI (piemēram, URL) un, tās paredzētas ar nosaukumu piešķiršanu saistītu problēmu risināšanai. Vienas vārdtelpas iekšienē visiem nosaukumiem jābūt unikāliem, lai gan citā vārdtelpā attiecīgajam nosaukumam var būt cita nozīmē.
Web Serviss vai IS serviss	Web Servisi ir apzīmējums tādām lietojumprogrammām, kuras ir paredzētas izsaukumiem no citu lietojumprogrammu puses un kuras saņem datus, un sniedz atbildes uz tiem standartizētā formātā (XML), parasti izmantojot SOAP protokolu. Web Servisi ir uz servisiem orientētas arhitektūras (SOA) sastāvdaļu konkrēta realizācija.
Datu devējs	Juridiska persona, kas tur citiem interesējošus datus, un ir kompetenta uz saviem nosacījumiem tos izplatīt citām juridiskām vai fiziskām personām.
Datu izplatīšanas kanāls (DIK)	Komunikācijas veids, kad ziņojumi plūst no viena sūtītāja, kas ir kanāla īpašnieks, vairākiem saņēmējiem, kas ir kanāla dalībnieki
Datu ņēmējs	Juridiska vai fiziska persona, kas saņem datu devēja datus, piekrītot to saņemšanas nosacījumiem.

TERMINS	APRAKSTS
Datu savākšanas kanāls (DSK)	Komunikācijas veids, kad ziņojumi plūst no vairākiem sūtītājiem, kas ir kanāla dalībnieki, vienam saņēmējam – kanāla īpašniekam
Biznesa ziņojums	Parasts ziņojums, kas nesaistās ar citiem ziņojumiem kanāla ietvaros
Transakcijas ziņojums	Standarta (klasificēta) sakārtota operāciju kopa, kas tiek pielietota noteikta datu masīva aktualizācijai. Katrs transakcijas ziņojuma ieraksts parasti satur DML ( <i>Data manipulation language</i> ) operācijas kodu ( <i>insert, update, delete</i> )
Kļūdas ziņojums	Atbildes ziņojums uz iepriekš izsūtīto biznesa vai transakcijas ziņojumu, ja tiek konstatētas problēmas ar to apstrādi. Kļūdas ziņojums vienmēr satur atsauci uz saistīto biznesa vai transakcijas ziņojumu.
DIT pasta kastīte	Pasta kastītē (virtuāla ziņojumu glabātava) tiek saglabāti visi nosūtītie un saņemtie ziņojumi.

### 1.1.2. Apzīmējumi

Dokumentā izmantotie apzīmējumi ir apkopoti 2.tabulā.

2.tabula

#### Apzīmējumi

APZĪMĒJUMS	APRAKSTS
ATVK	Administratīvi teritoriālo vienību klasifikators
DoS attack	Pakalpojumatteices uzbrukums (Denial of service attack)
DIT	Datu izplatīšanas tīkls
DIK	Datu izplatīšanas kanāls
DSK	Datu savākšanas kanāls
FAQ	Saraksts „Bieži uzdodamie jautājumi” (Frequently asked questions)
HTTP	Hiperteksta pārvietošanas protokols (Hyper Text Transfer Protocol)
HTTPS	Hiperteksta drošas pārsūtīšanas protokols (Secure Hypertext Transfer Protocol)
IeM	Iekšlietu ministrija
IeR	Iedzīvotāju reģistrs
IS	Informācijas sistēma
VRAA	Valsts reģionālās attīstības aģentūra
VISS	Valsts informācijas sistēmu savietotājs (bij.IVIS)
PMLP	Pilsonības un migrācijas lietu pārvalde
QoS	Servisa kvalitāte (Quality of service)
SLA	Pakalpojumu līmeņa vienošanās (Service Level Agreement)
SOA	Servisorientēta arhitektūra (Service Oriented Architecture)
VPN	Virtuāls privāts koptīkls (Virtual Private Network)
WS	Web Serviss (Web Service)
XML	Paplašināmās iezīmēšanas valoda (eXtensible Markup Language)
XSLT	Paplašināmās stila lapas pārveidošanas valoda (Extensible Stylesheet Language Transformation)
TM VVDZ	Tieslietu ministrijas Valsts vienotā datorizētā zemesgrāmata
TM VZD	Tieslietu ministrijas Valsts zemes dienests
VPN	Virtual Private Network

## 1.2. Saistība ar citiem dokumentiem

Dokuments ir izstrādāts, balstoties uz šādiem dokumentiem:

- [1] XML resursu izstrāde. Vadlīnijas (VRAA-6\_15\_11\_58-VISS\_2010-XML-VDL).
- [2] Standarts LVS 72:1996, leteicamā prakse programmatūras projektējuma aprakstīšanai.
- [3] Fizisko personu datu aizsardzības likums (<http://www.dvi.gov.lv/likumdosana/fpda/>).
- [4] Arhitektūras risinājuma vīzija (VRAA-6\_15\_11\_58-VISS\_2010-ARH-VIZ-V1.02-21.03.2012).
- [5] Metadatu un e-pakalpojumu identifikācija. Standarts (VRAA-6\_15\_11\_58-VISS\_2010-MET\_EPAK\_ID-STD-V1.03-21.02.2012).
- [6] <http://www.oasis-open.org/specs/index.php#wssv1.0>.
- [7] <http://www.oasis-open.org/specs/index.php#wssv1.1>.
- [8] E-pakalpojumu arhitektūras izstrāde. Vadlīnijas. (VRAA-13\_7\_17\_41-VISS\_2016-EPAK\_ARH-VDL).
- [9] Datu izplatīšanas tīkls: ārējās saskarnes. Programmatūras saskarnes projektējums. (VRAA-13\_7\_17\_41-VISS\_2016-DIT\_AS-PPA).
- [10] Elektronisko dokumentu krātuve: ārējās saskarnes. Programmatūras projektējuma apraksts (VRAA-13\_7\_17\_41-VISS\_2016-EDK-PPA\_AS).
- [11] IS servisu izstrādes standarts
- [12] SOAP v1.2. <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>.
- [13] WS-Addressing. <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509>.
- [14] WS-Security v1.1. <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>.
- [15] XML-Signature. <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>.
- [16] User Name Token Profile v1.1. <https://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-UsernameTokenProfile.pdf>
- [17] X.509 Certificate Token Profile v1.1. <https://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-x509TokenProfile.pdf>
- [18] WS-Trust v1.3. <http://docs.oasis-open.org/ws-sx/ws-trust/200512>.
- [19] SAML v1.1. <http://www.oasis-open.org/committees/download.php/3400/oasis-sstc-saml-1.1-pdf-xsd.zip>.
- [20] SAML v2.0. <http://docs.oasis-open.org/security/saml/v2.0/saml-2.0-os.zip>.
- [21] SAML Token Profile 1.1. <http://docs.oasis-open.org/wss/oasis-wss-SAMLTOKENProfile-1.1>.
- [22] VISS Drošības talonu serviss. Programmētāja rokasgrāmata (VRAA-13\_7\_17\_41-VISS\_2016-DTS-PR).

- [23] Valsts informācijas sistēmu savietotāja (VISS) un Vienotā valsts un pašvaldību pakalpojumu portāla [www.latvija.lv](http://www.latvija.lv) pilnveidošana un uzturēšana. API Pārvaldnieks (WSO2). Programmētāja rokasgrāmata. (VRAA-13\_7\_17\_41-VISS\_2016-API\_MG-PR)
- [24] WSO2 reģistrēto servisu abonēšana un abonentu bloķēšana,  
<https://apim.docs.wso2.com/en/latest/design/advanced-topics/block-subscription-to-an-api/https://docs.wso2.com/display/AM260/Block+Subscription+to+an+API>;
- [25] WSO2 dokumentācija, "Create and Publish an API",  
<https://apim.docs.wso2.com/en/latest/design/create-api/create-rest-api/create-a-rest-api/>;
- [26] WSO2 dokumentācija, "Create and Publish a SOAP API",  
<https://apim.docs.wso2.com/en/latest/design/create-api/create-rest-api/expose-a-soap-service-as-a-rest-api/>;  
<https://docs.wso2.com/display/AM260/Create+and+Publish+a+SOAP+API>
- [27] WSO2 OAuth Mediator, <https://github.com/imesh/wso2-oauth-mediator>
- [28] E-pakalpojuma izveidošanas ceļvedis [latvija.lv](http://latvija.lv) portālam. Programmētāja rokasgrāmata (VRAA-13\_7\_17\_41-VISS\_2016-LVP\_WZD\_3-PR)
- [29] OAuth 2.0 autorizācijas ietvara specifikācija, <https://tools.ietf.org/html/rfc6749>
- [30] JWT standarts, <https://jwt.io>;
- [31] Pieejas tiesību pieprasīšanas procedūra  
[https://viss.gov.lv/lv/Informacijai/partneriem/Sadarbibas\\_proceduras/API-abonesana](https://viss.gov.lv/lv/Informacijai/partneriem/Sadarbibas_proceduras/API-abonesana)

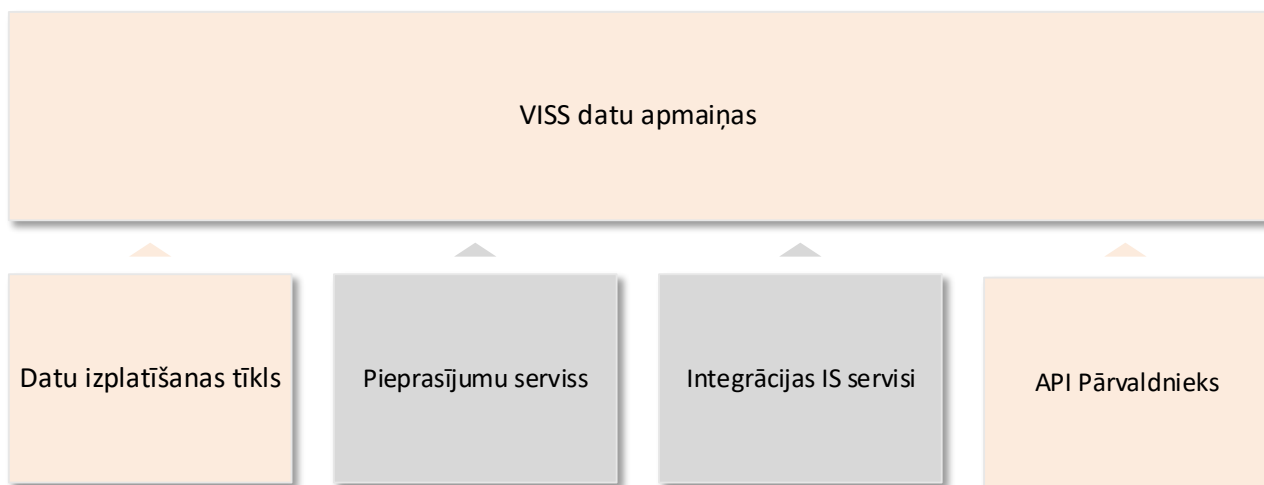
### 1.3. Dokumenta pārskats

Dokumentu veido pieci nodaļumi un pielikumi:

1. nodaļumā – „Ievads” – aprakstīta dokumenta kopējā struktūra, nolūks, izmantotie termini un apzīmējumi, kā arī saistība ar citiem dokumentiem.
2. nodaļumā – “VISS nodrošinātās datu apmaiņas” ir aprakstīti Pieprasījumu servisa un DIT izmantošanas scenāriji datu apmaiņai.
3. nodaļumā – „VISS pieprasījumu serviss” – ir dotas VISS Pieprasījumu servisa izmantošanas rekomendācijas.
4. nodaļumā – „Datu izplatīšanas tīkls” – ir sniegts DIT nepieciešamo tiesību un DIT lietošanas scenāriju apraksti.
5. Nodaļumā – „API Pārvaldnieks” – ir dotas VISS API pārvaldnieka risinājuma izmantošanas rekomendācijas.

## 2. VISS nodrošinātās datu apmaiņas

VISS projekta ietvaros ir īstenotas koplietojamās komunikācijas komponentes, kas nodrošina pilnvērtīgu centralizētu platformu datu sniegšanai iedzīvotājiem (G2C, pastarpināti izmantojot portālu latvija.lv), komersantiem (G2B) un valsts reģistriem (G2G):



### 1.attēls. VISS centrālās datu apmaiņas platformas galvenās komponentes

No pirmā skatupunkta klasiskā “point-to-point” integrācija tās dalībniekiem ir ērtāka un ātrāka, jo pašas iestādes var visu kontrolēt savu informācijas sistēmu ietvaros. Jebkurā gadījumā, arī klasiska “point-to-point” integrācija vienmēr prasa:

- Lietotāju (informācijas sistēmu) tiesību kontroli;
- Datu plūsmas organizāciju: protokoli, drošība, ziņojumu sintakse un semantika, utt.,
- Datu plūsmas monitorings,
- Personas datu audits.

Gadījumā, kad šādu integrāciju ir simtiem, no saimnieciskā viedokļa visi minētie uzdevumi jāuzliek uz centralizēto datu sniegšanas platformu, kas nodrošina katrai iestādei izmaksu ietaupījumu IT infrastruktūrā. VISS centrālo datu sniegšanas platformu veido šādas komponentes:

- Pieprasījumu serviss – klasisks *service bus* risinājums, kas nodrošina publicēto servisu sinhronu un asinhronu plūsmas komunikāciju, drošības slāņa nodrošināšanu, personas pieprasījumu auditāciju, ziņojumu validāciju, kontekstā bāzētu maršrutēšanu un darbību monitoringu. Pieprasījuma servisa tehnoloģiskais pēctecis ir API Pārvaldnieks, kas tiks pakāpeniski attīstīts, paplašinot tā funkcionalitāti, tomēr pārejas posmā abi šie risinājumi darbojas paralēli.
- API Pārvaldnieks – ir uz WSO2 platformas bāzēts *service bus* risinājums. Tas nodrošina iespēju publicēt sinhronus servissus un iespēju reģistrēties to izmantošanai, piekļuves tiesību kontroli, pieprasījumu monitoringu, maršrutēšanu un auditāciju.
- Datu izplatīšanas tīkls – nodrošina virtuālo vidi, kurā iestādes var reglamentēt automatizētu vai manuālu komunikāciju starp to informācijas sistēmām (IS). Informācijas apmaiņa notiek, izmantojot kanālus, kur katrs kanāls nodrošina ziņojumu pārsūtīšanu no viena īpašnieka vairākiem dalībniekiem (DIK kanāls) vai no vairākiem dalībniekiem vienam īpašniekam (DSK kanāls).
- Integrācijas IS servisi – valsts reģistru datu sniegšanas servisi, kas ir eksponēti ar VISS Pieprasījumu servisa palīdzību un izveidoti atbilstoši VISS prasībām (detalizēti skat. [11]).

ieviešot API pārvaldnieku, biznesa servisiem nav nepieciešams veidot integrācijas servisu starpslāni, bet tos izstrādāt, ievērojot Open API 2.0 vai Open API 3.0 specifikāciju.

Katra komponente tiek izmantota noteiktā komunikācijas gadījumā, skat. 3.tabulā.

3.tabula

### VISS centrālo datu sniegšanas platformas komponentes salīdzinājums

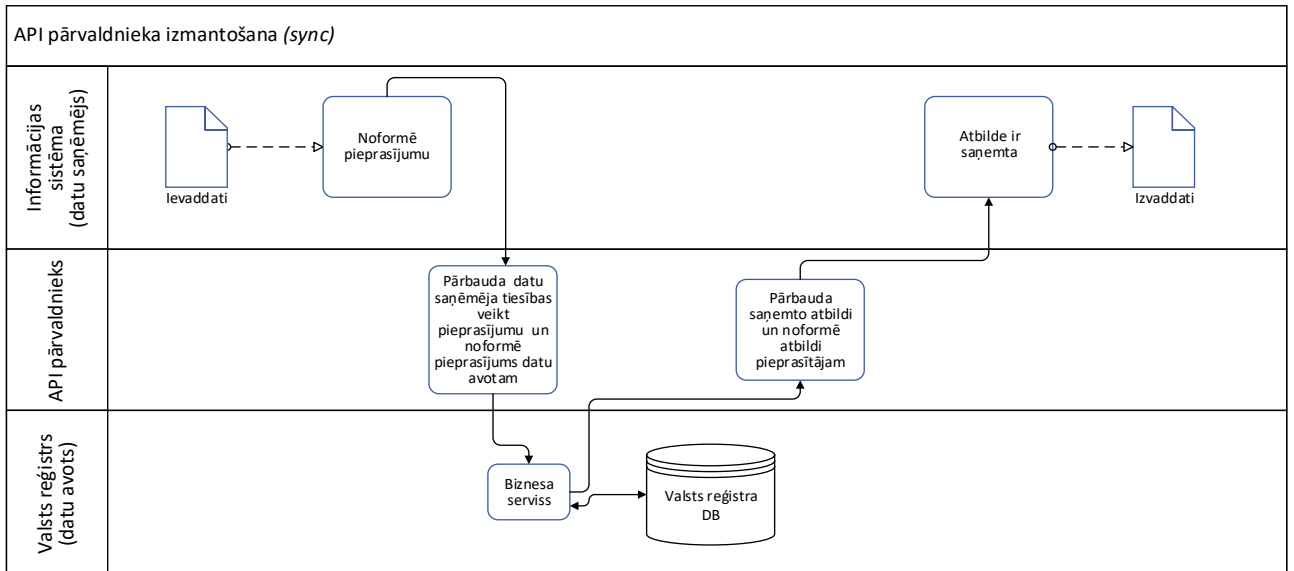
	PIEPRASĪJUMU SERVISS	DIT	API PĀRVALDNIKS
• Integrācija latvija.lv e-pakalpojumos	X		X
• Sinhrona datu apmaiņa	X		X
• Asinhrona datu apmaiņa	X	X	
• Ziņojuma validācija	X	X	
• Ziņojumu apjoma ierobežojums	4 MB <sup>1</sup>	0,5 GB	4 MB
• Lielapjoma (līdz 0,5 GB) datu sūtīšana		X	
• Garantēta piegāde		X	
• Formātu atbalsts:			
• XML	X	X	X
• JSON			X
• Protokolu/API atbalsts:			
• SOAP	X	X	X
• FTPs		X	
• REST			X

Līdzīgi e-pakalpojumiem, tiek uzturēti vairāki datu apmaiņas realizācijas biznesa scenāriji, kas nodrošina “sistēma – sistēma” komunikāciju. Pavisam tiek izšķirti tālāk aprakstītie VISS datu apmaiņas izmantošanas biznesa scenāriji.

## 2.1. API Pārvaldnieka izmantošana

API pārvaldnieka datu apmaiņas scenāriju var uzskatīt par “jautājums – atbilde” komunikāciju, kur iestādes informācijas sistēma (datu saņēmējs) noformē pieprasījumu (“jautājumu”), bet valsts reģistrs (datu avots) sagatavo atbildi, kas tiek sniegta ar API pārvaldnieka palīdzību, skat 2 attēlā.

<sup>1</sup> Gadījumā, ja nepieciešams nosūtīt ziņojumu ar lielāku apjomu, jāizmanto EDK, skat. 6. attēlā

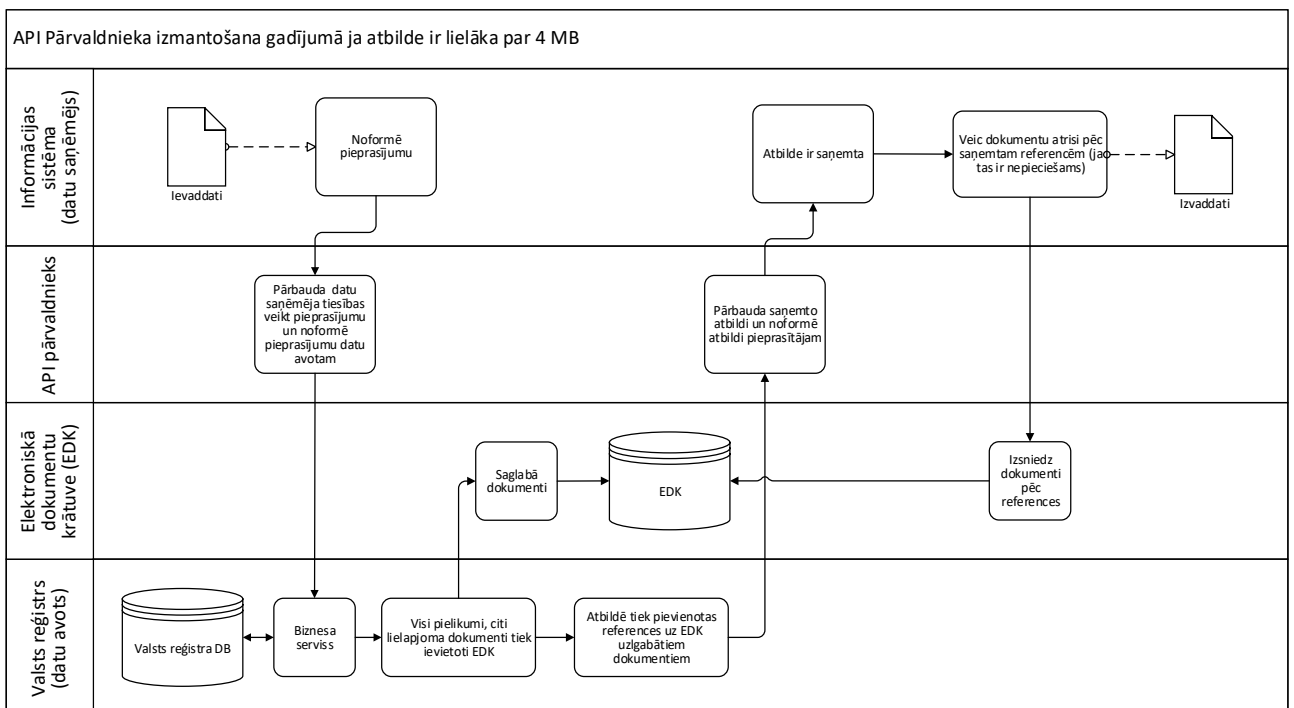


## 2.attēls. Biznesa scenārijs API Pārvaldnieka izmantošana sinhrono pieprasījumu realizācijai

API pārvaldnieks nodrošina tikai sinhronu pieprasījumu apstrādi, bet plānots nodrošināt arī iespēju veikt asinhronus pieprasījumus, kad atbilde datu saņēmējam tiek atgriezta pēc kāda laika.

Gadījumā, kad sagatavojamā atbilde ir lielāka par 4 MB, nepieciešams realizēt vienu no šādiem scenārijiem, kas nodrošina pārsūtāmā datu apjoma samazināšanu:

- Nodrošināt lapošanu datu servisā, lai atgrieztu tikai pieprasīto lappusi;
- Uzglabāt pielikumus, utt Elektroniskajā dokumentu krātuvē (EDK, skat.[10]), kas nodrošina to ilglaicīgu uzglabāšanu. EDK izmantošanas scenāriji jāaskaņo ar VRAA uzsākot datu apmaiņas projektēšanu.



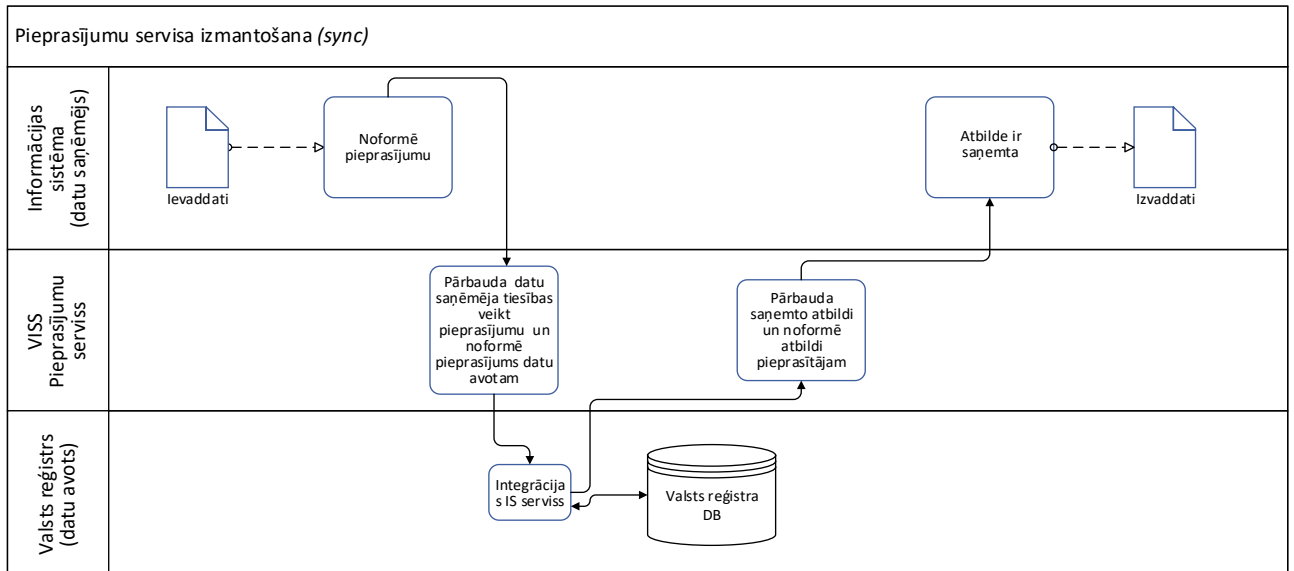
## 3.attēls. Biznesa scenārijs API pārvaldnieka servisa izmantošana sinhrono pieprasījumu realizācijai, kad sagatavojamā atbilde ir lielāka par 4 MB



## 2.2. Pieprasījumu servisa izmantošana

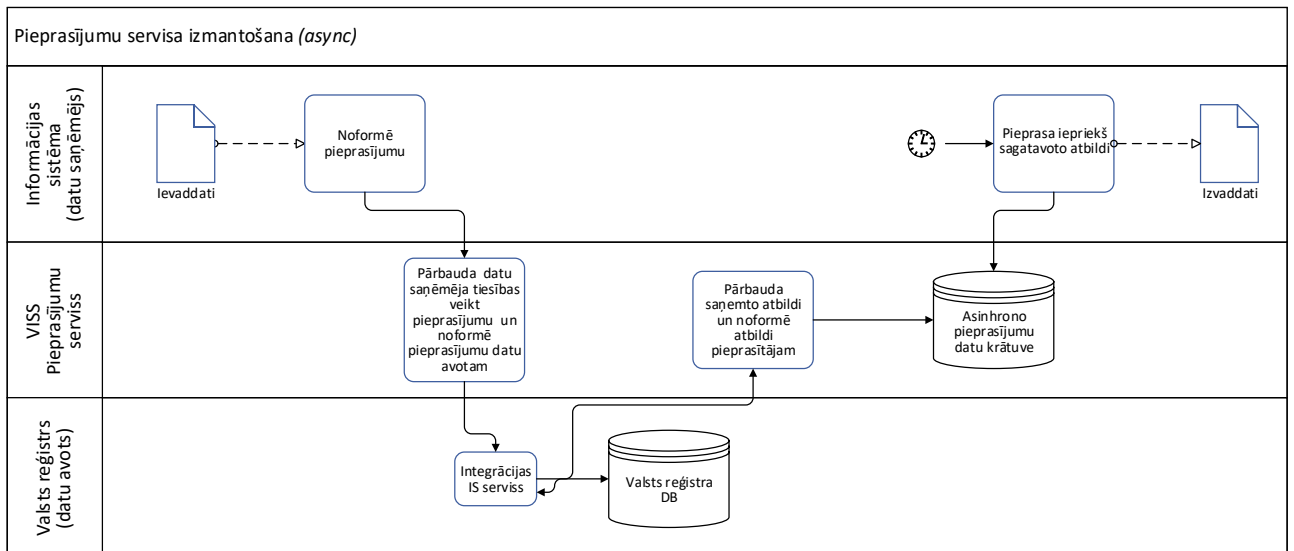
**Jaunu risinājumu izstrāde ir jāveic izmantojot API pārvaldnieku, kas ir Pieprasījuma servisa tehnoloģiskais pēctecis.**

Līdzīgi kā API pārvaldnieka gadījumā arī Pieprasījumu servisa datu apmaiņas scenāriju var uzskatīt par “jautājums – atbilde” komunikāciju, kur iestādes informācijas sistēma (datu saņēmējs) noformē pieprasījumu (“jautājumu”), bet valsts reģistrs (datu avots) sagatavo atbildi, kas tiek sniegta ar Pieprasījumu servisa palīdzību, skat. 4.attēlā.



### 4.attēls. Biznesa scenārijs Pieprasījumu servisa izmantošana sinhrono pieprasījumu realizācijai

Vienkāršākā gadījumā “jautājums – atbilde” komunikācija varētu notikt uzreiz (sinhroni) vai ar aizturi – kad iestādes informācijas sistēma (pieprasītājs) atbildi uz ievietoto pieprasījumu saņem ne uzreiz, bet pēc kāda laika perioda (asinhroni), skat. 5. attēlā.

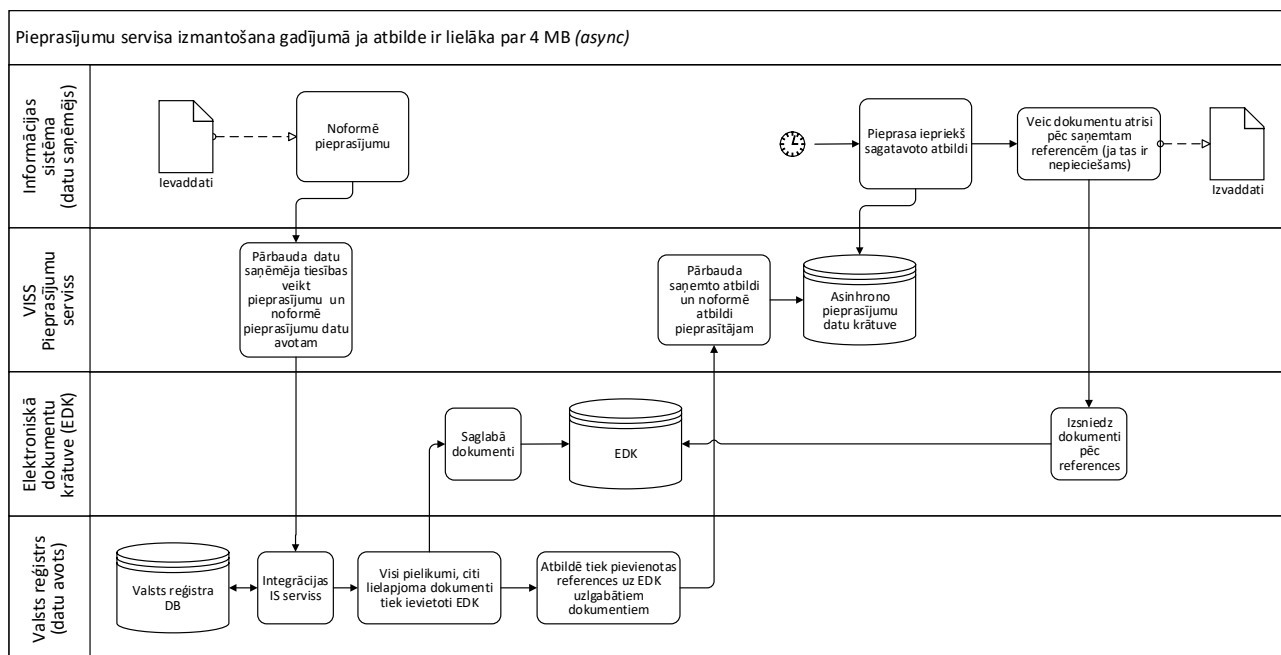


### 5.attēls. Biznesa scenārijs Pieprasījumu servisa izmantošana asinhrono pieprasījumu realizācijai

Gadījumā, kad sagatavojamā atbilde ir lielāka par 4 MB, nepieciešams realizēt vienu no šādiem scenārijiem, kas nodrošina pārsūtāmā datu apjoma samazināšanu:

- Nodrošināt lapošānu datu servisā, lai atgrieztu tikai pieprasīto lappusi;

- Uzglabāt pielikumus, utt Elektroniskajā dokumentu krātuvē (EDK, skat.[10]), kas nodrošina to ilglaicīgu uzglabāšanu. EDK izmantošanas scenāriji jāsaskaņo ar VRAA uzsākot datu apmaiņas projektēšanu.

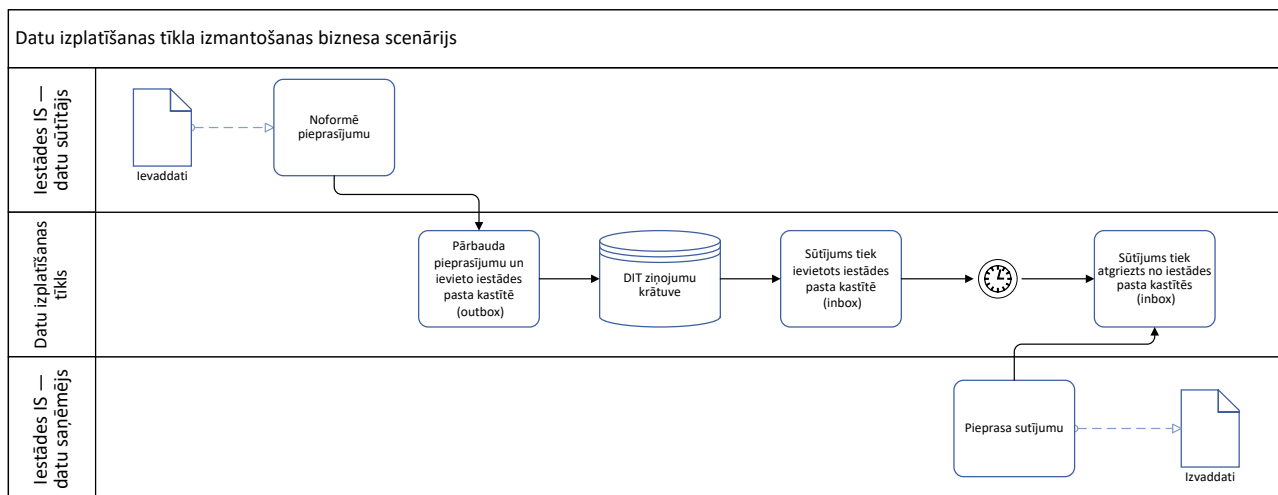


6.attēls. Biznesa scenārijs Pieprasījumu servisa izmantošana sinhrono pieprasījumu realizācijai, kad sagatavojamā atbilde ir lielāka par 4 MB

### 2.3. DIT izmantošana datu apmaiņai

DIT realizētā datu apmaiņa iestādēs varētu interesēt gadījumos, kad datu nodošana, izmantojot DIT, ir ļoti reta, piemēram, vienu reizi gadā vai mēnesī un nav lietderīgi veidot datu servisu vai citu apmaiņas infrastruktūru. Iestāde noformē XML ziņojumu, to piegādā uz DIT, un tas tiek uzglabāts iestādes izejošo sūtījumu mapē (*outbox*). No iestādes izejošās sūtījumu mapes XML ziņojumi nonāk līdz adresātiem viņu ienākošo sūtījumu mapēs (*inbox*), skat. 7.attēlā. Tiek uzturētas divas ziņojumu plūsmas:

- Viens sūtītājs izplata ziņojumu vienam vai vairākiem saņēmējiem (datu izplatīšanas kanāls);
- Viens vai vairāki sūtītāji piegādā ziņojumu vienam vai vairākiem saņēmējam (datu savākšanas kanāls).



7.attēls. Biznesa scenārijs DIT izmantošana datu izplatīšanai

Pateicoties tam, ka tiek uzturēta centralizēta izsūtīto/saņemto ziņojumu DB, katras iestādes inbox/outbox tiek uzglabāta reference (atsauce) uz ziņojumu, nevis ziņojums tiek fiziski kopēts. Katra ziņojuma izmēra ierobežojums ir 0.5 GB. Iestādes inbox/outbox apjoma ierobežojums tiek definēts administratīvi.

## **2.4. Datu servisi un nozares integratori**

Valsts reģistru datu sniegšanas servisi, kas ir eksponēti ar VISS infrastruktūras palīdzību un ir izveidoti atbilstoši VISS prasībām. Detalizēti par datu servisu izstrādi un nozares integratoriem skat. [11] dokumentā.

### 3. VISS Pieprasījumu serviss

**API Pārvaldnieks (skat. 5. nodaļu) pakāpeniski aizstāj VISS Pieprasījumu servisu. Jaunas datu apmaiņas nodrošināšanai nepieciešams izmantot API Pārvaldnieku.**

VISS Pieprasījumu serviss nodrošina iespēju piekļūt valsts reģistru un pašvaldību informācijas sistēmu sniegtajiem datu servisiem standartizētā veidā. VISS Pieprasījumu servisa risinājums pēc arhitektūras atbilst *enterprise service bus*<sup>2</sup> standarta risinājumam, kas nodrošina:

- publicēto servisu sinhronu un asinhronu plūsmas komunikāciju,
- drošības slāņa nodrošināšanu,
- personas pieprasījumu auditāciju,
- ziņojumu validāciju,
- konteksta bāzēto maršrutēšanu un
- darbību monitoringu.

Šī nodaļuma ietvaros tiek nodrošināts "road map", kas apraksta nepieciešamo darbību uzskaitījumu, lai varētu izmantot VISS Pieprasījumu servisā pieejamus Integrācijas IS servisi:

- Sameklēt sev nepieciešamo servisu, izmantojot integrācijas servisu katalogu, skat. 3.1. nodaļu.
- Pieprasīt tiesības IS servisu izsaukumam, skat. 3.3. nodaļu.
- Veikt iestādes Informācijas sistēmas programmatūras pielāgošanu, skat. 3.4. nodaļu.

#### 3.1. Pieejamo valsts reģistru servisu katalogs

Ar Pieprasījumu servisa palīdzību iespējams piekļūt publicētiem servisiem, kurus VISS infrastruktūrā dēvē par Integrācijas IS servisiem. Visu publicēto servisu apraksts ir pieejams VISS Informācijas sistēmu servisu katalogā (<https://lvp.viss.gov.lv/VISS.ISSK>), skat. 8.attēlā.

The screenshot shows the 'Par Informācijas sistēmu servisu katalogu' page. It contains a search bar, a list of services, and a table of 'Jaunākie IS servisi' (Latest IS services). The table has columns: Nosaucums, Urn, Versija, Izveidots, Atjaunots, and Īpašnieks. The services listed include EP136SubmitReqRequestIS, EP138SubmitReqRequestIS, EP138SubmitReqPaymentTransactIS, EP138GetReqTypesIS, EP138GetReqPlacesIS, EP138GetPaymentSettingsIS, EP138GetMoneyReceiverByIS, EP138GetLanguagesIS, PaymentStatusIS, and EP141UpdateCVIS.

8.attēls. IS servisu katalogs

<sup>2</sup> [https://en.wikipedia.org/wiki/Enterprise\\_service\\_bus](https://en.wikipedia.org/wiki/Enterprise_service_bus)

VISS IS servisu katalogs nodrošina valsts reģistru un pašvaldību servisu meklēšanas iespējas un servisu detalizētu aprakstu.

Ar Pieprasījumu servisa palīdzību iespējams izsaukt jebkādu Integrācijas servisu katalogā publicēto Integrācijas servisu (ja, protams, izsaukums saturēs atbilstošu autentifikācijas un autorizācijas informāciju). Katrs Integrācijas serviss katalogā ir aprakstīts ar šādiem svarīgiem atribūtiem:

- Identifikators – IS servisa globāli unikāls numurs IVIS URN formātā<sup>3</sup>;
- Autentifikācijas līmenis – kāda autentifikācijas un autorizācijas informācija ir nepieciešama servisa izsaukumam;
- Ieejas parametri – atsauce uz XML ziņojumu shēmas elementu, pēc kura ir jānoformē pieprasījums;
- Izejas parametri – atsauce uz XML ziņojumu shēmas elementu, pēc kura tiks noformēta servisa sniegtā atbilde.

### **3.2. Tiesību pieprasīšanas un piešķiršanas kārtība Integrācijas IS servisu izsaukšanai**

Tiklīdz, izmantojot IS servisu katalogā pieejamo saskarni tiks atrasts nepieciešamais Integrācijas IS serviss, datu ņēmējam nepieciešams pieprasīt tiesības publicētā servisa izsaukšanai sadarbībā ar datu turētāju un VRAA. VRAA datu ņēmējam piekļuvi pie datu devēja servisa dod tikai tad, kad VRAA no datu devēja ir saņēmis rakstisku piekļuves pieprasījumu. Pirms tam datu ņēmējs ar datu devēju savstarpēji ir vienojušies par datu izmantošanas nosacījumiem (līgums vai normatīvo aktu regulējums).

Pavisam tiek uzturēti vairāki autentifikācijas un autorizācijas scenāriji:

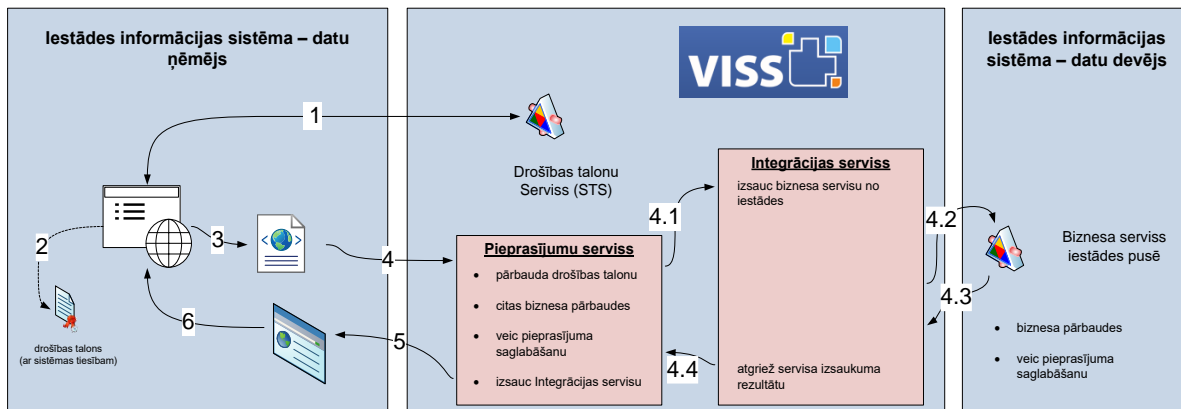
- “Sistēma – sistēma” komunikācijas gadījumā jāizmanto VISS reģistrētās iestādes autentifikācijas sertifikātu, kuram tiek piešķirtas tiesības noteikta Integrācijas IS servisa izsaukšanai<sup>4</sup>.
- Izmantot VISS vai Vienoto pieteikšanos lietotāju autorizācijai. VISS autorizēto lietotāju gadījumā, tiesības izsaukt noteikto Integrācijas IS servisu tiek piešķirtas lietotājam. Vienotās pieteikšanās autentifikācijas gadījumā izsaukumi tiek veikti iedzīvotāja vārdā.

### **3.3. Pieprasījumu servisa lietošanas scenārijs**

Tiklīdz tiek saņemtas tiesības uz Pieprasījumu servisā eksponēto Integrācijas servisu izsaukšanu, jārealizē informācijas sistēmas (portāla) pielāgošana, kas atbilst lietošanas scenārijam 9. attēlā.

<sup>3</sup> <https://tools.ietf.org/html/rfc4617>

<sup>4</sup> Izņemot DIT FTPs pieslēgšanu, kur jāizmanto lietotāja vārdu/paroli



### 9.attēls. Pieprasījumu servisā publicēta Integrācijas IS servisa izsaukums, izmantojot sertifikātu vai VISS lietotāja vārdu/paroli

Scenārijs ietver šādus soļus, kas ir paskaidroti ar XML ziņojumu fragmentiem<sup>5</sup>:

1. IS autentifikācija un autorizācija VISS drošības talonu servisā (STS), kur tiek saņemts autentifikācijas un autorizācijas apgalvojums (drošības talons), skat. 1. un 2. pielikumā. Drošības talons ir šifrēts un var tikt atšifrēts tikai ar Pieprasījumu servisa sertifikāta privāto atslēgu.
2. Drošības talons tiek pieglabāts atmiņā topošiem izsaukumiem.
3. Notiek pieprasījuma formēšana atbilstoši 3.4. nodaļā aprakstītām darbības metodēm.
4. Datu ņēmējs veic divus pieprasījumus VISS pieprasījumu servisam:
  - biznesa transakcijas reģistrācijai (skat. 3. pielikumā);
  - IS servisa izsaukums (skat. 5.pielikumā):
    - 4.1 –pieprasījumu serviss nodod pieprasījumu integrācijas servisam;
    - 4.2 –integrācijas serviss izveido pieprasījumu biznesa servisam;
    - 4.3 – integrācijas serviss saņem biznesa servisa atbildi;
    - 4.4 – pieprasījumu serviss saņem integrācijas servisa atbildi.
5. Datu ņēmējs saņem atbildi:
  - Uz biznesa transakcijas reģistrāciju (skat. 4.pielikumā)
  - Uz IS servisa izsaukumu (skat. 6.pielikumā)

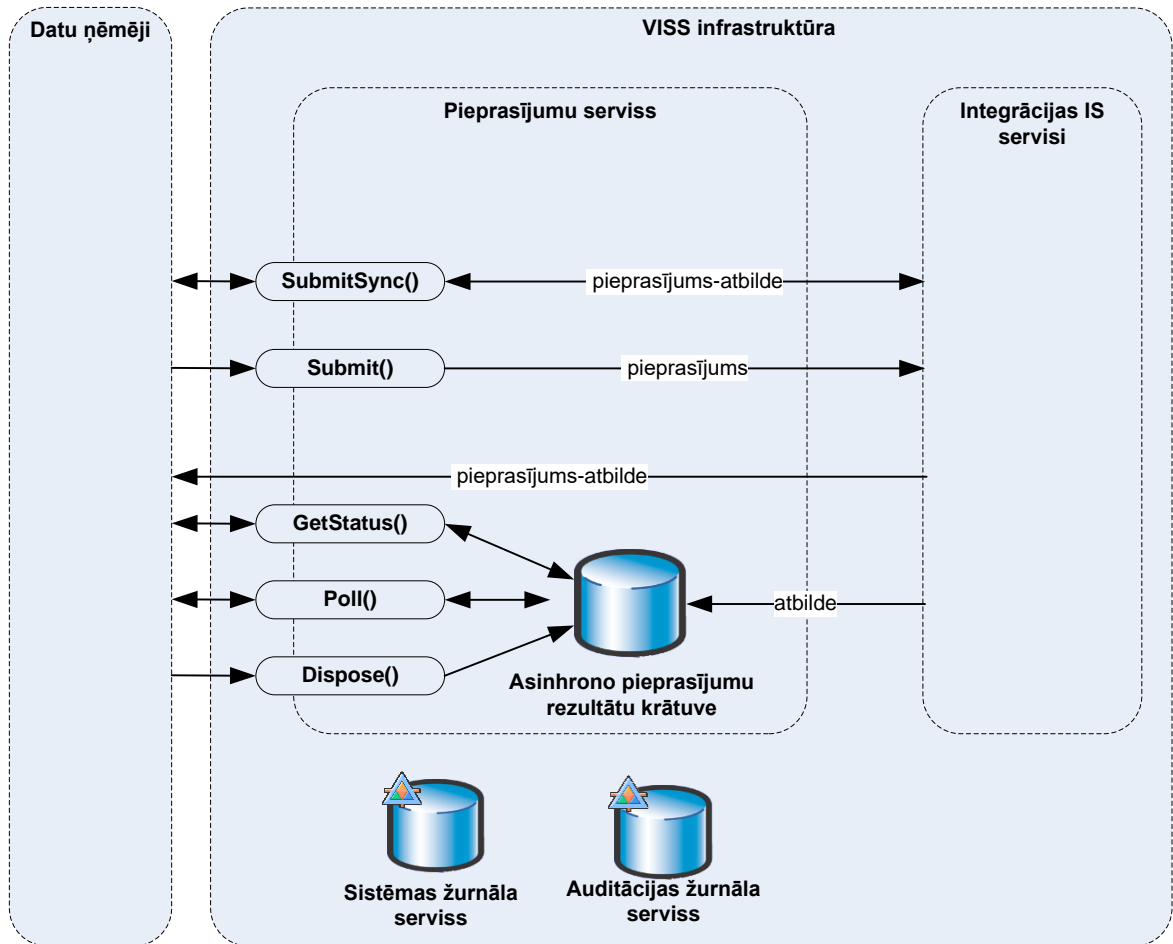
### 3.4. Pieprasījumu servisa darbības principi

Pieprasījumu serviss nodrošina pieeju Integrācijas IS servisiem, izpildot pieprasījumu auditācijas un kontroles uzdevumus.

**Pieeja Integrācijas IS servisiem, kā arī pastarpināti pie biznesa IS servisiem un citiem iestādes datu-sniedzējiem ārējām saskarnēm ir iespējama, tikai lietojot Pieprasījumu servisu, kas savukārt veido centrālo ieejas punktu VISS infrastruktūrā.**

Pieprasījumu servisa komponenti veido XML Web servisa veidā būvēta ārēja saskarne, BizTalk servera komponentes (Universālā asinhronā saskarne) un datu bāzes (skat. 10.attēlu).

<sup>5</sup> Šiem XML ziņojumu fragmentiem ir iepazīšanas nolūks, un tie nevar tikt lietoti reālā komunikācijas gadījumā



10.attēls. Pieprasījuma servisa detalizēts plānojums

Pieprasījumu serviss esošajā realizācijā atbilst gan Web Services Security v1.0<sup>6</sup> [6], gan Web Services Security v1.1<sup>7</sup> [7] standartiem. Pieprasījumu servisa saskarni veido metodes (Microsoft .NET C# sintaksē), kas ir aprakstītas 4.tabulā.

4.tabula

## Pieprasījuma servisa metodes

PIEPRASĪJUMA SERVISĀ METODE	APRAKSTS
IVISResponse SubmitSync (IVISRequest request)	Ar metodi SubmitSync tiek nodrošināts sinhrons izsaukums Integrācijas IS servisam, kura identifikators ir pievienots IVISRequest.Destination atribūtā. Sinhrona izsaukuma gadījumā par parametru nepieciešams izmantot IVISRequest struktūru (skat. 3.12.1. sadaļā). Metode atgriež arī IVISResponse struktūru, kas attiecīgā IS servisa darbības rezultātā parasti tiek papildināta ar kādiem biznesa datiem <Body> elementā.
IVISResponse Submit (IVISRequest request)	Asinhrons izsaukums tiks uzsākts, izsaucot metodi Submit Integrācijas IS servisam, kura identifikators ir pievienots IVISRequest.Destination atribūtā. Kā metodes parametru nepieciešams izmantot IVISRequest struktūru (skat. 3.12.1. sadaļā). Veiksmīga izsaukuma gadījumā, izsaucējam tiks atgriezta IVISResponse struktūra ar aizpildītu CorrelationID – konkrētā

<sup>6</sup> OASIS standarts "Web Services Security v1.0" (WS-Security 2004), kas ietver SOAP Message Security 1.0 (WS-Security 2004)

<sup>7</sup> OASIS standarts "Web Services Security v1.1", kas ietver WS-Security Core Specification 1.1

PIEPRASĪJUMA SERVISA METODE	APRAKSTS
<pre>int GetStatus (CorrelationID requestID)</pre>	<p>pieprasījuma identifikatoru. Šis elements tiks lietots, lai sasaistītu oriģinālo pieprasījumu ar „back-end” sistēmas sniegto atbildi. Tas var tikt lietots kā parametrs Pieprasījumu servisa metodēm <code>Poll</code>, <code>GetStatus</code> un <code>Dispose</code>. Asinhronajā scenārijā <code>&lt;Body&gt;</code> elements netiks atgriezts.</p> <p><code>GetStatus</code> metode atgriezīs sagatavotās atbildes statusu. Tā ir jāizsauc pēc <code>Submit</code> metodes izsaukuma, lai uzzinātu, vai atbilde jau ir sagatavota. Kā parametrs tai jānodod <code>CorrelationID</code>.</p> <p>Pieprasījuma servisa <code>GetStatus</code> metodes implementācijai tiks izmantota Asinhrono pieprasījumu rezultātu krātuve (speciāla datu bāze asinhrono rezultātu glabāšanai).</p>
<pre>IVISResponse Poll (CorrelationID requestID)</pre>	<p><code>Poll</code> metode atgriezīs sagatavoto atbildes ziņojumu. To nepieciešams izsaukt pēc <code>Submit</code> metodes izsaukuma, lai iegūtu atbildes ziņojumu. Vispirms ar <code>GetStatus</code> metodi ir jānoskaidro, vai atbilde jau ir sagatavota. Ja atbilde jau ir sagatavota, tad to var iegūt ar <code>Poll</code> metodi. <code>Poll</code> metodi var izsaukt vairākas reizes. Kamēr atbilde ar <code>Dispose</code> metodi nebūs dzēsta, to vēl joprojām varēs saņemt. Kā parametrs tai jānodod <code>CorrelationID</code>. <code>Poll</code> metode atgriež <code>IVISResponse</code> struktūru.</p>
<pre>int Dispose (CorrelationID requestID)</pre>	<p><code>Dispose</code> metode tiek lietota asinhrona scenārija gadījumā. Šī metode obligāti jāizsauc, lai Pieprasījumu servisam paziņotu, ka atbilde ir veiksmīgi saņemta un turpmāk to var neuzglabāt, pretējā gadījumā - atbilde tiks automātiski izdzēsta pēc noteikta laika perioda. Pieprasījuma servisa <code>Dispose</code> metode izmantos Asinhrono pieprasījumu rezultātu datu bāzi atbildes meklēšanai.</p>
<pre>URNType StartTransaction (URNType eServiceID)</pre>	<p>Metode <code>StartTransaction</code> tiek izsaukta, kad nepieciešams uzsākt konkrētu e-pakalpojuma instanci. Parametrs <code>eServiceID</code> ir e-pakalpojuma identifikators. Metode atgriež e-pakalpojuma instances (transakcijas) URN identifikatoru.</p>
<pre>void EndTransaction (URNType transactionID)</pre>	<p>Metode <code>EndTransaction</code> tiek izsaukta, kad nepieciešams pabeigt konkrētu e-pakalpojuma instanci. Parametrs <code>TransactionID</code> ir e-pakalpojuma instances (transakcijas) identifikators.</p>

Pieprasījumu un vaicājumu protokolā tiek izmantoti divi ziņojumu veidi – `IVISRequest` un `IVISResponse`. Kļūdu gadījumā izsaucējs saņems `<Errors>` struktūru, skat. 3.12.2.2.paragrāfā. Pieprasījumu serviss nodrošina šādas kopīgas funkcijas:

- Personalizē lietotāju, kura vārdā ir izsaukts serviss;
- Veic pieprasījuma auditāciju;
- Nodrošina tālāku *back-end* sistēmas Integrācijas IS servisu izsaukumu un atbildes ziņojumu saņemšanu;
- Sinhroniem IS servisiem atgriež Integrācijas IS servisa izpildes rezultātu.

Anonīmiem (bez drošības talona) pieprasījumiem tiek noteiktas šādas tiesības:

METODE	AUTORIZĒTAIS LIETOTĀJS	ANONĪMAIS LIETOTĀJS
<code>StartTransaction</code>	bez ierobežojuma	bez ierobežojuma
<code>EndTransaction</code>	bez ierobežojuma	tikai anonīmo lietotāju transakcijas
<code>DisposeRequest</code>	bez ierobežojuma	tikai anonīmo lietotāju pieprasījumi
<code>GetStatus</code>	bez ierobežojuma	tikai anonīmo lietotāju atbildes
<code>Poll</code>	bez ierobežojuma	tikai anonīmo lietotāju atbildes
<code>SubmitSync</code>	bez ierobežojuma	tikai uz anonīmajiem integrācijas servisiem



METODE	AUTORIZĒTAIS LIETOTĀJS	ANONĪMAIS LIETOTĀJS
Submit	bez ierobežojuma	tikai uz anonīmajiem integrācijas servisiem
SaveResponse (ar drošību)	bez ierobežojuma	tikai anonīmo lietotāju pieprasījumi
SaveResponse (bez drošības VISS infrastruktūrā)	-	bez ierobežojuma

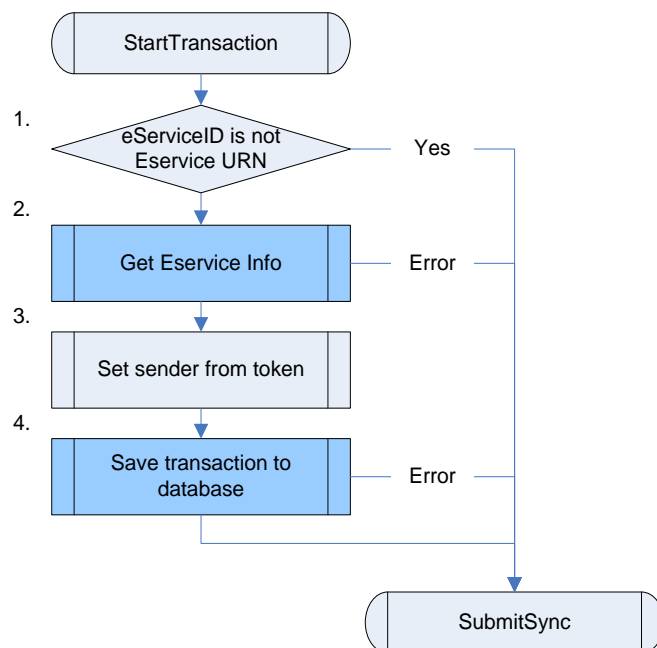
**Pieprasījumu apjoma ierobežojums ir 4 MB. Ja nepieciešams pārsūtīt lielākus pielikumus, datnes, utt., ir jālieto DIT (skat. 2.3. nodaļu) vai Elektroniskās dokumentu krātuves (EDK) risinājums (skat. [10] saistīto dokumentu). EDK izmantošanas scenāriji jāaskaņo ar VRAA uzsākot datu apmaiņas projektēšanu.**

**Metodi StartTransaction nepieciešams izsaukt no e-pakalpojuma vai integrācijas procesa tikai pirms pirmā datu (biznesa) pieprasījuma veikšanas, izmantojot Submit vai SubmitSync. Var izmantot vienu biznesa transakcijas identifikatoru vairākiem izsaukumiem, īpašie, ja tie ir saistīti savā starpā pēc biznesa nozīmes.**

### 3.5. StartTransaction metode

Metode StartTransaction tiek izsaukta, kad nepieciešams uzsākt konkrētu e-pakalpojuma vai datu apmaiņas procesa biznesa instanci. Parametrs eServiceID ir e-pakalpojuma vai datu apmaiņas procesa identifikators. Lai iegūtu datu apmaiņas identifikatoru (URN) ir jāsazinās VRAA un tā piešķirs konkrētai datu apmaiņai paredzētu identifikatoru. Metode atgriež e-pakalpojuma (datu apmaiņas) instances (transakcijas) URN identifikatoru un, izsaucot metodi SetProcessStatus, saglabā e-pakalpojuma vai datu apmaiņas procesa statusu „Sākums” (skat. 11.attēlu):

6. Tiek pārbaudīts, vai padotais „eServiceID” ka tas ir IVIS:URN;
7. Tiek iegūta informācija no e-pakalpojumu kataloga un e-formu kataloga;
8. “Sender” tiek aizpildīts no drošības talona;
9. Tiek izveidots transakcijas identifikators un saglabāts datu bāzē.



11.attēls. StartTransaction metode

Biznesa transakcija parasti nepieciešama komunikācijas identifikācijas gadījumā. Iespējamo e-pakalpojumu URN tipu uzskaitījums biznesa transakcijas uzsākšanai:

- EP – e-pakalpojums latvija.lv portālā (piemēram, URN:IVIS:100001:EP-EP00-V1-2);

- EP.VISS – e-pakalpojums VISS portālā (piemēram, URN:IVIS:100001:EP.VISS-ErrorReport-V1-2);
- EP.DA – sistēma–sistēma datu sniegšanas pakalpojums (piemēram, URN:IVIS:100001:EP.DA-DA29-V1-0);
- EF – e-forma (piemēram, URN:IVIS:100001:EF-Form01-V1-0).

**Sistēma–sistēma datu sniegšanas pakalpojuma gadījumā (EP.DA) papildus tiek pārbaudīta autentificētā lietotāja un izmantotā EP.DA URN sasaiste, kā papildus sistēmu pieprasījumu autorizācijas līdzeklis.**

Kļūdas gadījumā tiek atgriezts tukšais identifikators.

### 3.6. EndTransaction metode

Metode `EndTransaction` tiek izsaukta, kad nepieciešams pabeigt konkrētu e-pakalpojuma datu vai datu apmaiņas procesa instanci. Parametrs `transactionID` ir e-pakalpojuma vai datu apmaiņas procesa instances (transakcijas) identifikators. Metode izsauks metodi `SetProcessStatus` ar statusu „Nobeigums”.

Pārējās metodes darbojas vai nu sinhronā, vai asinhronā variantā.

Papildus ir pieejams arī nedrošais `SaveResponse` serviss, kas ir izsaukams tikai no VISS vides. Ar šī servisa palīdzību, iespējams pabeigt e-pakalpojuma vai datu apmaiņas procesa instanci no Integrācijas IS servisa, kas ir izmitināts VISS infrastruktūrā. Serviss implementē metodes:

```
void SaveResponse (IVISResponse response)
int EndTransaction(URNType transactionID).
```

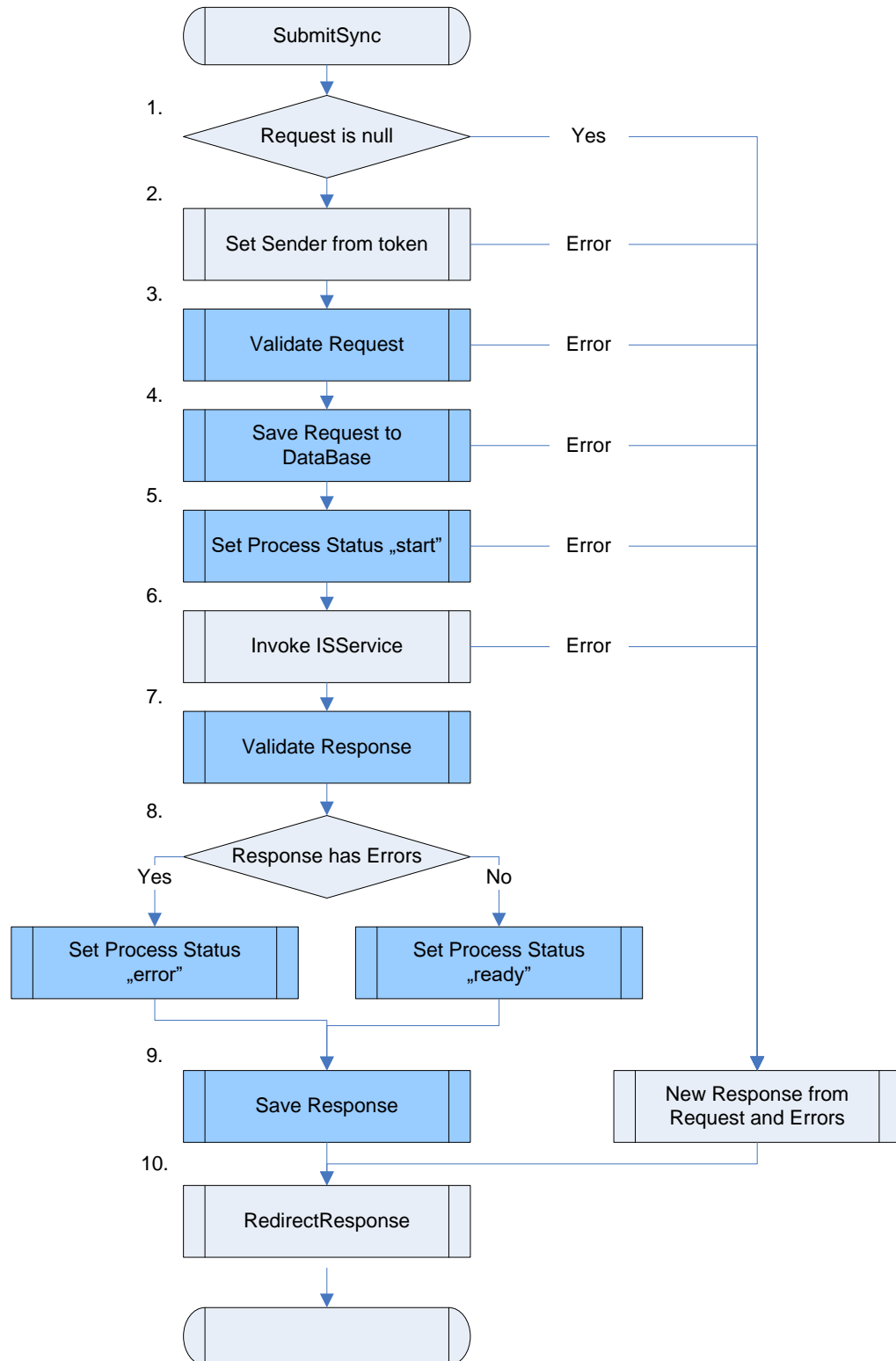
Metodē `EndTransaction` tiek atgriezta „0”, ja transakcijas pabeigšanas rezultāts bija neveiksmīgs.

**Visiem nepabeigtajiem sinhronajiem pieprasījumiem transakcija tiek izbeigta automātiski par iepriekšējo dienu, visiem asinhroniem pieprasījumiem pēc 12 mēnešiem.**

### 3.7. Sinhronie pieprasījumi

Sinhrons izsaukums tiek nodrošināts, izsaucot metodi `SubmitSync`. Kā metodes parametru nepieciešams izmantot struktūru `IVISRequest`. Sinhrona izsaukuma gadījumā metode atgriezīs arī struktūru `IVISResponse`, kas attiecīgā IS servisa darbības rezultātā parasti tiek papildināta ar kādiem datiem. 12.attēlā ir parādīta sinhrona pieprasījuma blokshēma:

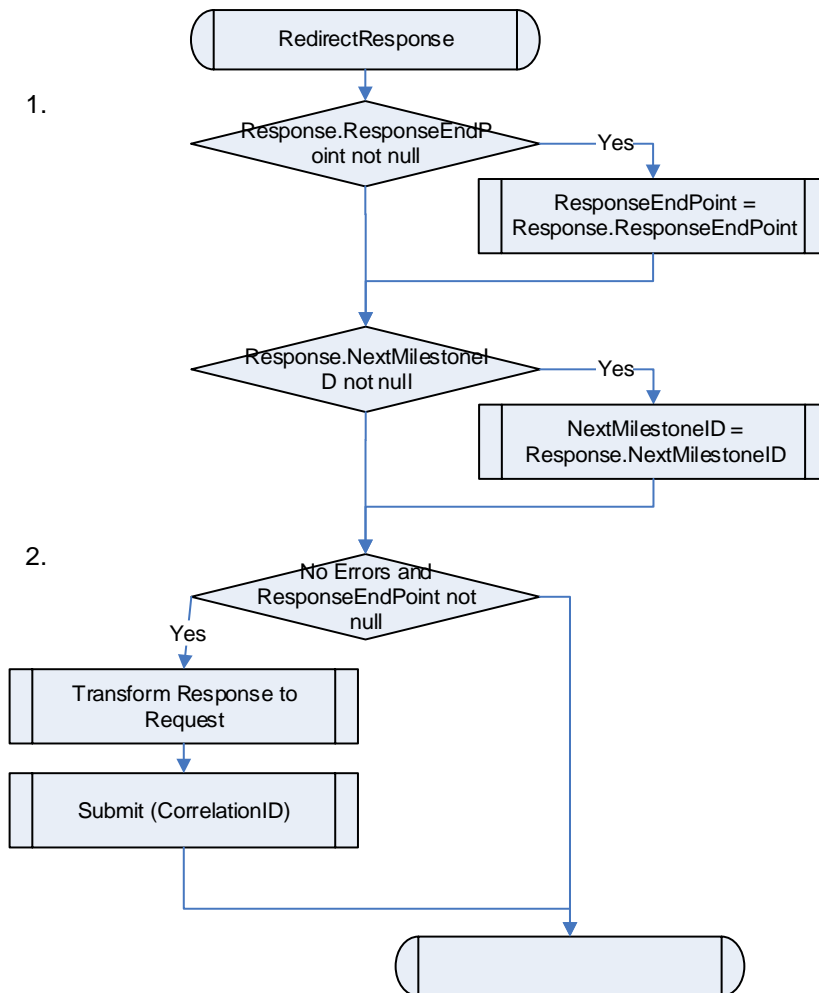
1. Tiek pārbaudīts, vai padotais „Request” ir tukšs (*is NULL*);
10. “Sender” tiek aizpildīts no drošības talona;
11. Tiek pārbaudīts pieprasījums, kā arī tiek pārbaudīts, vai lietotājs ir saglabājis attiecīgo atbildi;
12. Pieprasījums tiek saglabāts datu bāzē;
13. Process tiek uzstādīts uz statusu “start”;
14. Tiek izsaukts IS serviss vai IDDV forma;
15. Tiek pārbaudīta atbilde;
16. Atkarība vai atbilde bija kļūdas; ja tā bija, tad process tiek uzstatīts statusā “error” vai “ready”;
17. Tiek saglabāta atbilde;
18. Atbildes pāradresācija (ja nepieciešams) `RedirectResponse` tiek veikta pāradresācija uz citu transakciju, kas atbilst diagrammai 13.attēlā.



## 12.attēls. Sinhrons pieprasījums

Katrā pieprasījuma solī (ja nepieciešams) `RedirectResponse` tiek veikta pārdresācija uz citu transakciju, kas atbilst diagrammai 13.attēlā:

1. Ja atbildē tiek norādīts `"ResponseEndPoint"` vai `"NextMilestoneID"`, tad jāveic pārdresācija;
2. Ja atbildē nav kļūdu un jāveic pārdresācija, atbilde tiek pārveidota uz pieprasījumu, un tiek izsaukta metode `Submit`, kas atbilst diagrammai 14.attēlā.



13.attēls. Pieprasījuma pāradresācija

### 3.8. Asinhronie pieprasījumi

Asinhrons izsaukums tiks uzsākts, izsaucot metodi `Submit`. Kā metodes parametru nepieciešams izmantot struktūru `IVISRequest`. Veiksmīga izsaukuma gadījumā, izsaucējam tiks atgriezta struktūra `IVISResponse` ar aizpildītu `CorrelationID`. Šis elements tiks lietots, lai oriģinālo pieprasījumu sasaistītu ar *back-end* sistēmas sniegto atbildi. To var lietot arī kā parametru Pieprasījumu servisa metodēm `Poll`, `GetStatus` un `Dispose`. Asinhronajā scenārijā elements `<Body>` netiks atgriezts.

Kļūdas gadījumā struktūra `<Errors>` saturēs vienu vai vairākus kļūdas paziņojumus, kas sastāvēs no kļūdas identifikatora `<ErrorCode>` un kļūdas ziņojuma `<ErrorDescription>`.

Metode `GetStatus` atgriezīs sagatavotās atbildes statusu. To nepieciešams izsaukt pēc metodes `Submit` izsaukuma, lai uzzinātu, vai atbilde jau ir sagatavota. Par parametru tai jānodod `CorrelationID`. Pieprasījumu servisa metodes `GetStatus` implementācijai paredzēts izmantot Asinhrono pieprasījumu rezultātu krātuvi.

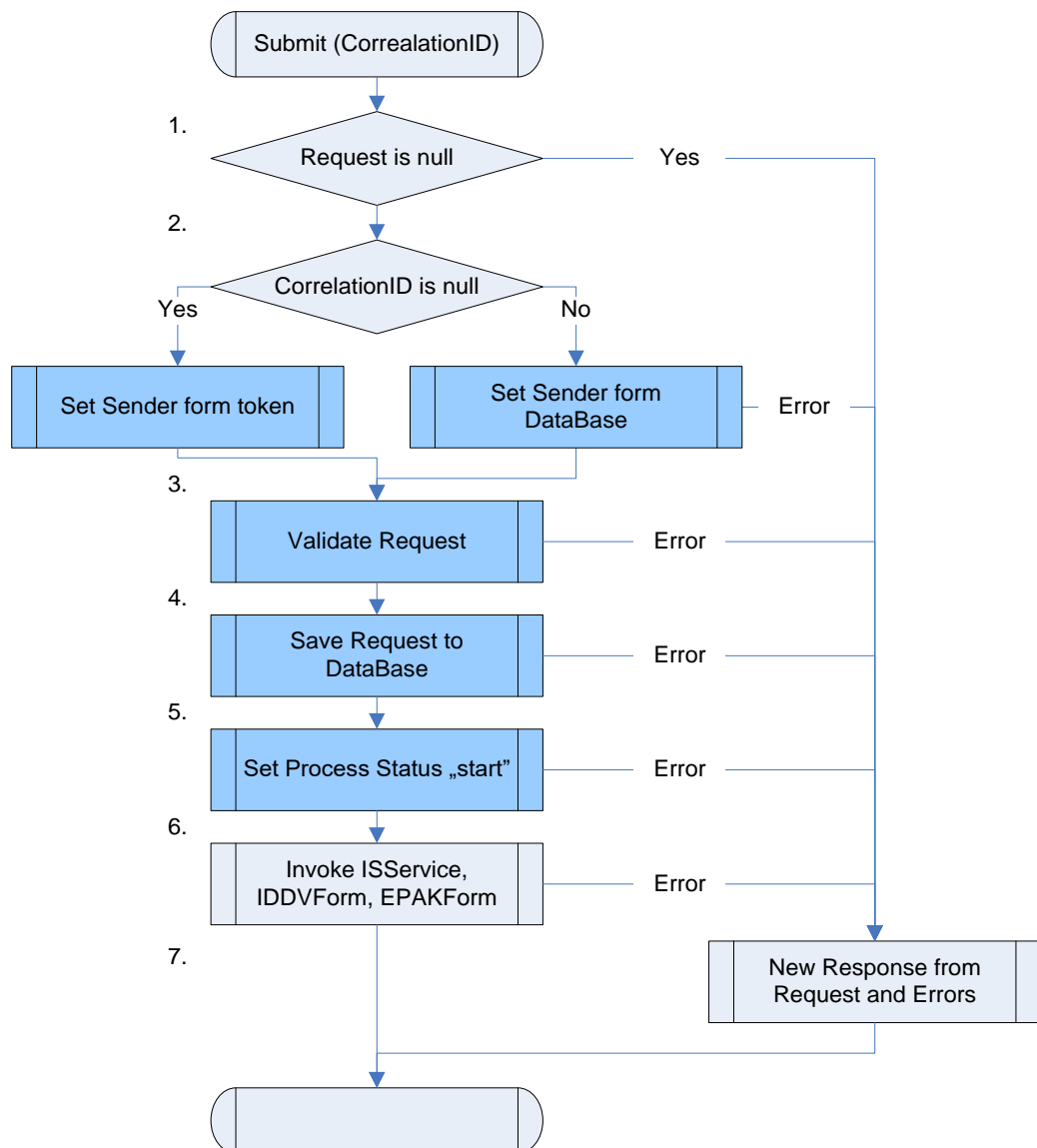
Metode `Poll` atgriezīs sagatavoto atbildes ziņojumu. To nepieciešams izsaukt pēc metodes `Submit` izsaukuma, lai iegūtu atbildes ziņojumu. Vispirms ar metodi `GetStatus` noskaidro, vai atbilde jau ir sagatavota: ja atbilde ir sagatavota, tad to iespējams iegūt ar metodi `Poll`. Metodi `Poll` var izsaukt vairākas reizes. Kamēr atbilde nebūs dzēsta ar metodi `Dispose`, to joprojām varēs saņemt. Par parametru tai jānodod `CorrelationID`. Metode `Poll` atgriež struktūru `IVISResponse`. Kļūdas

gadījumā tiks aizpildīts elements <Errors>. Pieprasījuma servisa metode Poll izmantos Asinhrono pieprasījumu rezultātu krātuvi.

Metode Dispose tiek lietota asinhrona scenārija gadījumā. Šī metode obligāti jāizsauc, lai Pieprasījumu servisam paziņotu, ka atbilde ir veiksmīgi saņemta un to var turpmāk neuzglabāt, pretējā gadījumā – atbilde pēc noteikta laika perioda tiks automātiski izdzēsta. Pieprasījuma servisa metode Dispose izmantos Asinhrono pieprasījumu rezultātu datu bāzi.

14.attēlā ir parādīta asinhrona pieprasījuma blokshēma:

1. Tiek pārbaudīts, vai padotais „Request” ir tukšs (is NULL);
2. “Sender” tiek aizpildīts no drošības talona;
3. Tiek pārbaudīts pieprasījums, kā arī tiek pārbaudīts, vai lietotājs ir saglabājis attiecīgo atbildi;
4. Pieprasījums tiek saglabāts datu bāzē;
5. Process tiek uzstādīts uz statusu “start”;
6. Tiek izsaukts IS serviss vai IDDV forma;

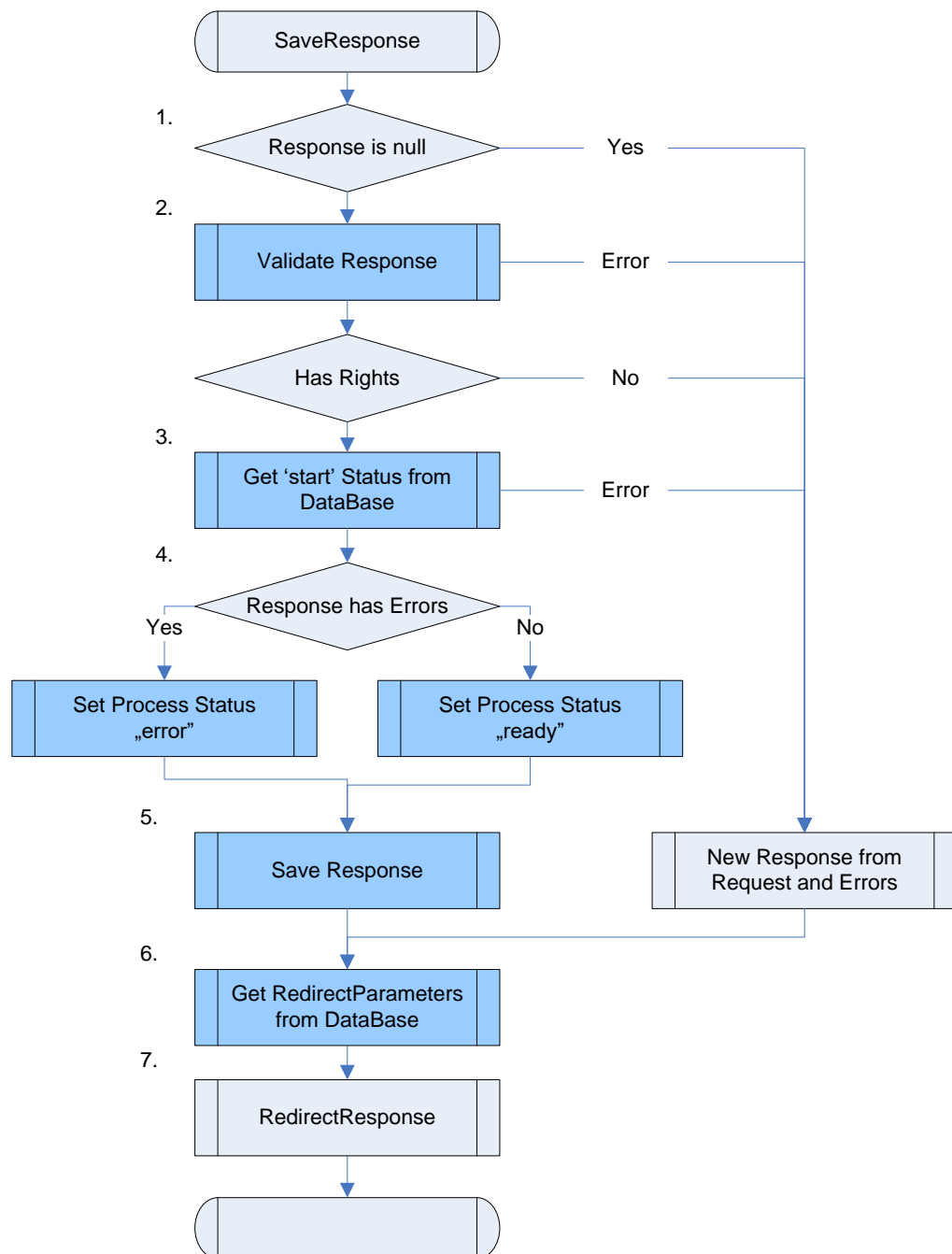


14.attēls. Asinhrons pieprasījums

### 3.9. SaveResponse

Serviss `SaveResponse` veic atbildes saglabāšanu Pieprasījumu servisa Asinhrono pieprasījumu krātuvē. 15. attēlā ir parādīta servisa darbības blokshēma:

1. Tiek pārbaudīts, vai padotais „Response” ir tukšs (*is NULL*);
19. Tiek pārbaudīta atbilde; arī tiek pārbaudīts, vai lietotājs ir saglabājis attiecīgo atbildi;
20. Tiek iegūts procesa sākotnējais stāvoklis;
21. Atkarība, vai atbildē bija kļūdas; ja jā - tad process tiek uzstāpīts statusā “error” vai “ready”;
22. Tiek saglabāta atbilde;
23. Tiek iegūti pārdresācijas parametri.
24. Atbildes pārdresācijas (ja nepieciešams); `RedirectResponse` tiek veikta pārdresācija uz citu transakciju, kas atbilst diagrammai 13.attēlā.



15.attēls. SaveResponse serviss

### 3.10. „GetStatus” metode

GetStatus metode atgriezīs sagatavotās atbildes statusu. Tā ir jāizsauc pēc „Submit” metodes izsaukuma, lai uzzinātu, vai atbilde jau ir sagatavota. Kā parametrs tai jānodod „CorrelationID”.

Pieprasījuma servisa „GetStatus” metodes implementācijā notiek šādas darbības:

- Tiek pārbaudīts, vai lietotājs ir tiesīgs saņemt datus par attiecīgo pieprasījumu;
- Pieprasījumu stāvokļu datu bāzē pārbaudīs, vai atbilde jau ir saņemta;
- Ja atbilde tiks saņemta, atgriezīs izsaucējam „ResponseReady” (5);
- Ja atbilde netiks saņemta, atgriezīs izsaucējam „ResponseNotReady” (6);
- Gadījumā, kad nav atrasts pieprasījuma „CorrelationID”, tiek atgriezta kļūda „CorrelationIdNotFound”.

Iespējamie <ErrorCode> kodi neveiksmīga „GetStatus” izsaukuma gadījumā ir apkopoti 3.11.1. sadaļā.

### 3.11. Poll

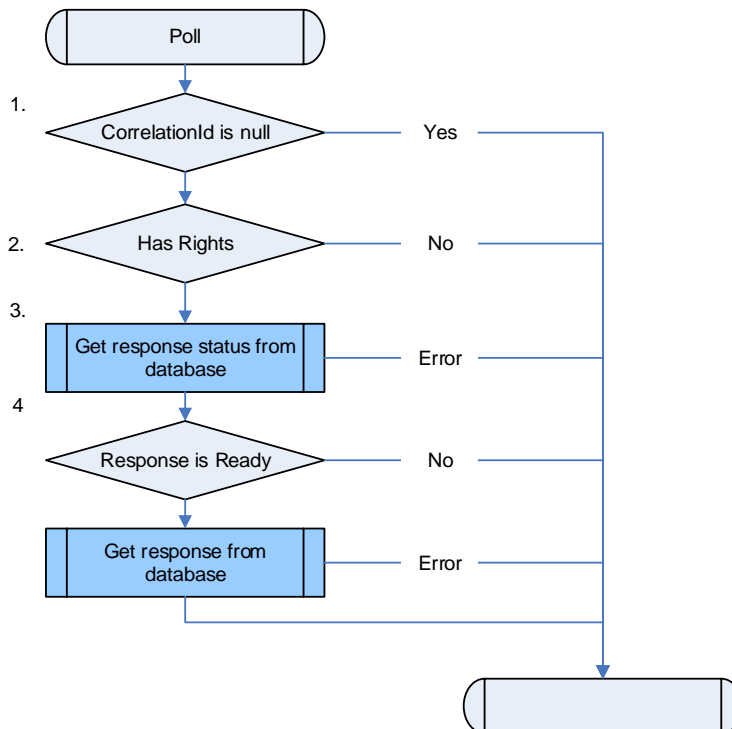
Serviss Poll veic atbildes izgūšanu no Pieprasījumu servisa Asinhrono pieprasījumu krātuves. To nepieciešams izsaukt pēc „Submit” metodes izsaukuma, lai iegūtu atbildes ziņojumu. Vispirms ar „GetStatus” metodi ir jānoskaidro, vai atbilde jau ir sagatavota. Ja atbilde jau ir sagatavota, tad to var iegūt ar „Poll” metodi. „Poll” metodi var izsaukt vairākas reizes. Kamēr atbilde ar „Dispose” metodi nebūs dzēsta, to vēl joprojām varēs saņemt. Kā parametrs tai jānodod „CorrelationID”. „Poll” metode atgriež „IVISResponse” struktūru. Kļūdas gadījumā tiks aizpildīts <Errors> elements. Pieprasījuma servisa „Poll” metode izmantos Asinhrono pieprasījumu rezultātu krātuvi un izpildīs šādas darbības (skat. 16.attēlā):

1. Tiek pārbaudīts, vai padotais „CorrelationID” nav tukšs (*not NULL*)
  25. Tiek pārbaudīts, vai lietotājs ir tiesīgs saņemt datus par attiecīgo pieprasījumu;
  26. Pieprasījumu stāvokļu datu bāzē pārbaudīs, vai atbilde jau ir saņemta;
  27. Ja atbilde nebūs saņemta vai citas kļūdas gadījumā, atgriezīs kādu no vērtībām, kas ir sniegtas 5'.tabulā;
  28. Ja atbilde ir saņemta, nolasīs ziņojumu no datu bāzes un atgriezīs izsaucējam.
- 16.attēlā ir parādīta servisa darbības blokshēma.

Veiksmīga izsaukuma gadījumā <Result> elements saturēs „Success”, neveiksmīga izsaukuma gadījumā <Result> elements saturēs „Failure”, kā arī <Errors> struktūra saturēs kļūdas paziņojumus.

Iespējamie <ErrorCode> kodi neveiksmīga „Poll” izsaukuma gadījumā ir apkopoti 3.11.1.sadaļā.

**Sagatavotās atbildes “apjautāšanās” biežums ir atkarīgs no biznesa scenārija, bet tas nevar būt mazāks par 10 sek. Gadījumā, ja saskaņā ar biznesa scenāriju, atbilde netiek sniegta pakalpojumā sagaidāmajā laikā, nepieciešams izbeigt “apjautāšanās” procesu šim ievietotajam pieprasījumam.**



16.attēls. Poll serviss

### 3.11.1. Pieprasījumu servisa kļūdas

Iespējamās Pieprasījumu servisa kļūdas ir apkopotas 5.tabulā.

5.tabula

#### Pieprasījumu servisa operāciju rezultātu kodi

KODS	NUMURS	APRAKSTS
MessageIdNotSpecified	0	Nav norādīts MessageID.
CorrelationIdNotSpecified	1	Nav uzdots korelācijas numurs.
CorrelationIdNotFound	2	Atbilde ar norādīto CorrelationID '{korelācijas vērtība}' neeksistē vai ir nekorekta.
TransactionIdincorrect	3	Nepareizs transakcijas identifikators.
TransactionNotStarted	4	Transakcija '{transakcijas vērtība}' nav uzsākta vai pabeigta.
SenderIncorrect	5	EmployeePoll nevar būt ivisRequest sūtītājs.
DestinationIsNull	6	Destination nav norādīts.
DestinationIncorrectType	7	Nepareizi norādīts Destination vai ResponseEndPoint. Ir jālieto sinhronam uzsaukumam URN:IVIS ISService, bet asinhronam izsaukumam ISService, IDDVForm vai EPAKForm identifikatoru.
DestinationNotFound	8	Norādītais Destination '{saņēmēja vērtība}' neeksistē.
DestinationNotInterationServiss	9	ISServiss nav reģistrēts kā integrācijas serviss.
DestinationWrongServiceCallingType	10	Destination '{saņēmēja vērtība}' nav reģistrēts ISServisu katalogā kā '{pieslēgšanās veida vērtība}' serviss.
DestinationUserRightToLow	11	Lietotājam nav tiesību izsaukt ISServisu.
DestinationIncorrectAddress	12	DestinationID '{saņēmēja vērtība}' nav uzdots servisa adrese.



KODS	NUMURS	APRAKSTS
DestinationIncorrectTestAddress	13	DestinationID '{saņēmēja vērtība}' nav uzdota testa servisa adrese.
MessageTypeNotSet	14	Pieprasījumā ir Body daļa, bet Header daļā nav norādīts MessageType.
BodyNotSet	15	Pieprasījuma Header daļā ir norādīts MessageType, bet nav Body daļas.
BodyIncorrect	16	{Vaicājuma tipa vērtība: request vai response} Body serializācijas kļūda {xml validācijas kļūdas apraksts}.
BodyToBig	17	{Vaicājuma tipa vērtība: request vai response} pieprasījuma kopējais garums pārsniedz atļauto.
MessageTypeIncorrect	18	{Ziņojuma tipa vērtība} messageType nav URN:IVIS shēmas tipa identifikatoros.
BodyWrongFormat	19	{XML elementa nosaukums} pieprasījuma Body elements neatbilst MessageType laukā norādītajai shēmai.
MessageTypeNotFound	20	{Ziņojuma tipa vērtība} xml shēma XML shēmu katalogā neeksistē.
MilestoneIdIncorrect	21	Nepareizs pieturpunkta identifikators.
MilestoneIdNotFound	22	Norādītais Milestone identifikators nav reģistrēts.
CorrelationIdsNotEqual	23	IVISResponse korelācijas identifikators nav vienāds ar IVISRequest identifikatoru.
TransactionIdsCopied	24	Atbildes transactionID tiek aizvietots ar pieprasījuma transactionID.
GeneratedMessageId	25	Tika noģenerēts jauns messageId.
GeneratedCorrelationid	26	Tika noģenerēts jauns corellationID.
DestinationMustHaveRealm	27	IS Servisam jābūt uzdotam realm atribūtam.
RequestIsNull	100	Pieprasījums vai pieprasījuma Header struktūra ir tukša.
ResponselsNull	101	Atbilde vai atbildes Header struktūra ir tukša.
ErrorSaveRequest	102	Kļūda saglabājot pieprasījumu datubāzē.
ErrorSaveResponse	103	Kļūda saglabājot atbildi datubāzē.
ErrorSetStatus	104	Kļūda saglabājot procesa statusu datubāzē.
ErrorInvoke	105	Kļūda izsaucot ISServisu vai IDDV formu. Apraksts: {servisa izsaušanas kļūdas apraksts}.
ErrorRestoreSender	106	Kļūda atjaunojot sūtītāju.
ErrorRestoreMilestoneId	107	Kļūda atjaunojot pieturpunktu.
ErrorUserRightToLow	108	Lietotājam nav tiesību izpildīt operāciju.
ErrorSetRequestStatus	109	Kļūda saglabājot ziņojuma statusu datubāzē.

## 3.12. Datu struktūru apraksts

### 3.12.1. IVISRequest struktūra

Struktūras `IVISRequest` apraksts ir pieejams 6. tabulā. Obligātos elementus varētu noskaidrot shēmas tipa datnē (skat. <https://ivis.eps.gov.lv/RC.WebApp/Resource/Details/294>).

## 6.tabula

## IVISRequest struktūras elementu uzskaitījums

ELEMENTS	KUR TIEK AIZPILDĪTS	APRAKSTS
MessageID	Submit vai SubmitSync	Globāli unikāls identifikators, kas identificē konkrēto ziņojumu. Elements, kas nepieciešams, lai identificētu konkrēto ziņojumu. Lauka formāts atbilst VISS UUIDType datu tipam.
MessageType	Ārējos servisos un biznesa loģikas bibliotēkās	<MessageType> ir primārais elements, kas nosaka, kāda veida datus satur ziņojuma <Body> elements; tas ir atbilstošās XML shēmas URN:IVIS identifikators XML shēmu katalogā. <MessageType> nosaka <Body> elementa XML shēmu. Šis elements ir obligāts.
TransactionID	Ārējos servisos un biznesa loģikas bibliotēkās	Elements <TransactionID> tiek lietots, lai visus Integrācijas IS servisu pieprasījumus un atbildes sasaistītu vienā biznesa transakcijā (e-pakalpojuma vai datu apmaiņas procesa instance). Šis elements nav obligāts. <TransactionID> Pieprasījumu servisam nodod izsaucejs, savukārt Pieprasījumu serviss šo pašu <TransactionID> atgriež visās atbildēs, ko tas sniedz. To saņem, izsaucot metodi StartTransaction, un tas atbilst URN:IVIS shēmai.
CorrelationID	SubmitSync	Šis elements tiks lietots, lai sasaistītu oriģinālo pieprasījumu ar back-end sistēmas sniegto atbildi vienas „pieprasījums-atbilde” transakcijas ietvaros. Lauka formāts atbilst VISS (IVIS) UUIDType datu tipam.
Sender	Submit vai SubmitSync	Elements <Sender> ļauj identificēt ziņojuma sūtītāju un nosaka ziņojuma tālāko maršrutēšanu. Elementa saturs parasti tiek ņemts no drošības talona. Elements ir izmantojams auditācijai un citiem līdzīgiem nolūkiem.
Destination	Ārējos servisos un biznesa loģikas bibliotēkās	Elements <Destination>. Tas kopā ar elementiem <Sender> un <MessageType> nosaka ziņojuma tālāko maršrutēšanu. Elements satur IS servisa URN:IVIS identifikatoru IS servisu katalogā vai IDDV (KDV) formas identifikatoru. Šis elements ir obligāts.
ResponseEndPoint	Ārējos servisos un biznesa loģikas bibliotēkās	Sinhronā izsaukuma gadījumā šis elements netiek izmantots (izņemot Maksājuma moduļa EPAKPaymentSync servisu), tāpēc pieprasījuma ziņojumā to neiekļauj. Šis elements satur adresi, kurā ziņojuma sūtītājs vēlas saņemt atbildi - IS servisa URN:IVIS identifikatoru IS servisu katalogā vai IDDV (KDV) formas identifikators.
Test	Ārējos servisos un biznesa loģikas bibliotēkās	Elements <Test> norāda, ka šis ir testa ziņojums. Ja elements nav norādīts vai tā vērtība ir „false”, tad ziņojums netiek uzskatīts par testa ziņojumu. Ja elementa vērtība ir „true”, tas ir testa ziņojums. Testa ziņojums tiek sūtīts uz testa IS servisu, ja Destination laukā ir norādīta IS servisa adrese. IS servisa testa adrese ir definēta IS servisu katalogā lauka „IS servisa testa versijas URL”.
MilestoneID	Ārējos servisos un biznesa loģikas bibliotēkās	Pieturpunkta IVIS:URN identifikators nosaka stāvokli, kuru sasniegs e-pakalpojums, ja konkrētais pieprasījums tiks izpildīts veiksmīgi. Respektīvi, tiek saglabāti katra pieturpunkta divi stāvokļi: pieprasījuma sākums un pieprasījuma beigas. Elements netiek lietots datu apmaiņas procesa gadījumā.

ELEMENTS	KUR TIEK AIZPILDĪTS	APRAKSTS
NextMilestoneID	Ārējos servisos un biznesa loģikas bibliotēkās	Sinhronā izsaukuma gadījumā šis elements netiek izmantots, tāpēc pieprasījuma ziņojumā to neiekļauj. Šis elements satur pieturpunkta IVIS:URN identifikatoru, kas nosaka nākamo stāvokli, kuru uzsāks e-pakalpojuma izpildes process, ja konkrētais pieprasījums tiks izpildīts veiksmīgi. Elements netiek lietots datu apmaiņas procesa gadījumā.
Body	Ārējos servisos un biznesa loģikas bibliotēkās	Elements <Body> satur konkrētā servisa pieprasījuma datus, kas ir atkarīgi no elementa <MessageType>. Elements <Body> drīkst saturēt jebkādu XML datus.

### 3.12.2. IVISResponse struktūra

IVISResponse struktūras atgriežamais datu apjoms mainās veiksmīga vai neveiksmīga izsaukumu gadījumā. Obligātos elementus var noskaidrot pēc shēmas tipa datnē (skat. <https://lvp.viss.gov.lv/RC.WebApp/Resource/Details/295>).

#### 3.12.2.1. IVISResponse struktūra veiksmīgas izpildes gadījumā

Struktūrā IVISResponse veiksmīgas izpildes gadījumā obligāti tiks atgriezti elementi, kas ir redzami 7. tabulā.

7.tabula

#### IVISResponse struktūras elementu uzskaitījums (veiksmīgs izsaukums)

ELEMENTS	KUR TIEK AIZPILDĪTS	APRAKSTS
MessageID	IS serviss	Globāli unikāls identifikators, kas identificē konkrēto ziņojumu. Elements, kas nepieciešams tikai BizTalk, lai identificētu konkrētu ziņojumu. Lauka formāts atbilst VISS (IVIS) UUIDType datu tipam.
MessageType	IS serviss	<MessageType> ir primārais elements, kas nosaka, kāda veida datus satur ziņojuma <Body> elements; tas ir atbilstošās XML shēmas URN:IVIS identifikators XML shēmu katalogā. <MessageType> nosaka <Body> elementa XML shēmu. Šis elements ir obligāts.
TransactionID	Nokopēt no IVISRequest	Elements <TransactionID> tiek lietots, lai visus IS servisu vai e-pakalpojumu servisu pieprasījumus un atbildes sasaistītu vienā biznesa transakcijā (e-pakalpojuma instance). Šis elements nav obligāts. <TransactionID> Pieprasījumu servisam nodod izsaucējs, savukārt Pieprasījumu serviss šo pašu <TransactionID> atgriež visās atbildēs, ko tas sniedz. To saņem, izsaucot StartTransaction metodi, un tas atbilst URN:IVIS shēmai.
CorrelationID	Nokopēt no IVISRequest	Sinhronā izsaukuma gadījumā šo lauku nelieto, savukārt asinhronā izsaukumu gadījumā šis elements tiks lietots, lai oriģinālo pieprasījumu sasaistītu ar back-end sistēmas sniegto atbildi vienas „pieprasījums-atbilde” transakcijas ietvaros. Lauka formāts atbilst VISS (IVIS) UUIDType datu tipam.
PollInterval	IS serviss	Asinhronā izsaukuma gadījumā šis elements norādīs laika intervālu sekundēs, pēc kāda izsaucējam turpmāk ir jāizsauc metode GetStatus.
TimeStamp	IS serviss	Pieprasījuma saņemšanas laiks.
Result	IS serviss	Success
ErrorList	IS serviss	Atstāt tukšu.

ELEMENTS	KUR TIEK AIZPILDĪTS	APRAKSTS
Body	IS serviss	Elements <Body> satur konkrētā servisa pieprasījuma datus, kas ir atkarīgi no elementa <MessageType>. Elements <Body> drīkst saturēt jebkādu XML datus. Netiek atgriezts asinhrono pieprasījumu gadījumā (Submit).

### 3.12.2.2. IVISResponse struktūra neveiksmīgas izpildes gadījumā

Struktūrā IVISResponse neveiksmīgas izpildes gadījumā tiks atgriezti elementi, kas ir redzami 8. tabulā.

8.tabula

#### IVISResponse struktūras elementu uzskaitījums (neveiksmīgs izsaukums)

ELEMENTS	KUR TIEK AIZPILDĪTS	APRAKSTS
MessageID	Nokopēt no IVISRequest	Globāli unikāls identifikators, kas identificē konkrēto ziņojumu. Elements, kas nepieciešams tikai BizTalk, lai identificētu konkrētu ziņojumu. Lauka formāts atbilst VISS (IVIS) UUIDType datu tipam.
MessageType	IS serviss	<MessageType> ir primārais elements, kas nosaka, kāda veida datus satur ziņojuma <Body> elements; tas ir atbilstošs XML shēmas URN:IVIS identifikators XML shēmu katalogā. <MessageType> nosaka <Body> elementa XML shēmu. Šis elements ir obligāts.
TransactionID	Nokopēt no IVISRequest	<TransactionID> elements tiek lietots, lai visus IS servisu vai e-pakalpojumu servisu pieprasījumus un atbildes sasaistītu vienā biznesa transakcijā (e-pakalpojuma instance). Šis elements nav obligāts. <TransactionID> Pieprasījumu servisam nodod izsaucējs, savukārt Pieprasījumu serviss šo pašu <TransactionID> atgriež visās atbildēs, ko tas sniedz. To saņem, izsaucot StartTransaction metodi, un tas atbilst URN:IVIS shēmai.
CorrelationID	Nokopēt no IVISRequest	Sinhronā izsaukuma gadījumā šo lauku nelieto savukārt asinhronā izsaukuma gadījumā šis elements tiks lietots, lai oriģinālo pieprasījumu sasaistītu ar back-end sistēmas sniegto atbildi vienas „pieprasījums-atbilde” transakcijas ietvaros. Lauka formāts atbilst VISS (IVIS) UUIDType datu tipam.
PollInterval	IS serviss	Atstāt tukšu.
TimeStamp	IS serviss	Pieprasījuma saņemšanas laiks.
Result	IS serviss	Failure
ErrorList	IS serviss	Aizpilda vienu vai vairākas kļūdas
Body	IS serviss	Elements <Body> satur konkrētā servisa pieprasījuma datus, kas ir atkarīgi no elementa <MessageType>. Elements <Body> drīkst saturēt jebkādu XML datus. Netiek atgriezts asinhrono pieprasījumu gadījumā (Submit).

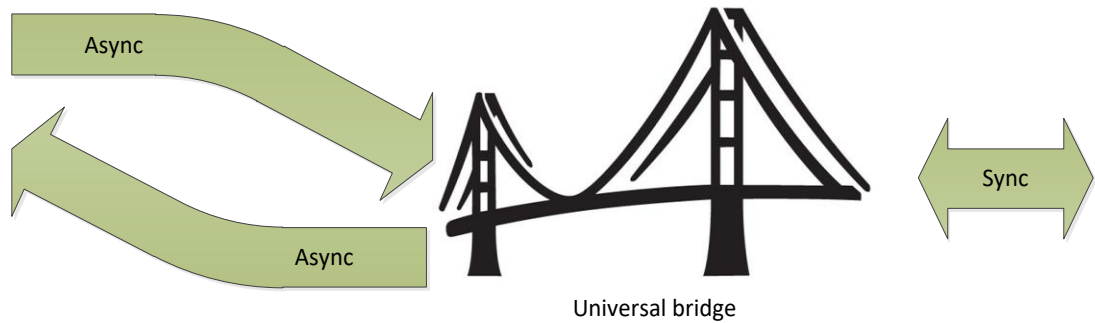
### 3.13. Universālā Pieprasījuma servisa asinhronās saskarnes izmantošana

Universālā Pieprasījuma servisa asinhronā saskarne (tālāk Universālā asinhronā saskarne) dod iespēju izmantot sinhronus servisu asinhronā interfeisā, kas nodrošina šādas priekšrocības:

1. Izmantot asinhronās sūtīšanas principus bez BizTalk (BPEL) programmēšanas, atvieglot asinhronu servisu izstrādāšanas procesu;

29. Salīdzinājumā ar parasto .NET servisa realizāciju, tiks nodrošināta iespēja izmantot *BizTalk* rīkus, lai kontrolētu darba plūsmas izpildi;
30. Iespēja nodrošināt papildu biznesa atskaites izveidošanu ar *BAM* rīkiem;
31. Iespēju konfigurēt asinhronus izsaukumus ar dažādām atkārtotības iespējām, dažādos adapteros un protokolos, kas būtiski palielina sistēmas efektivitāti.

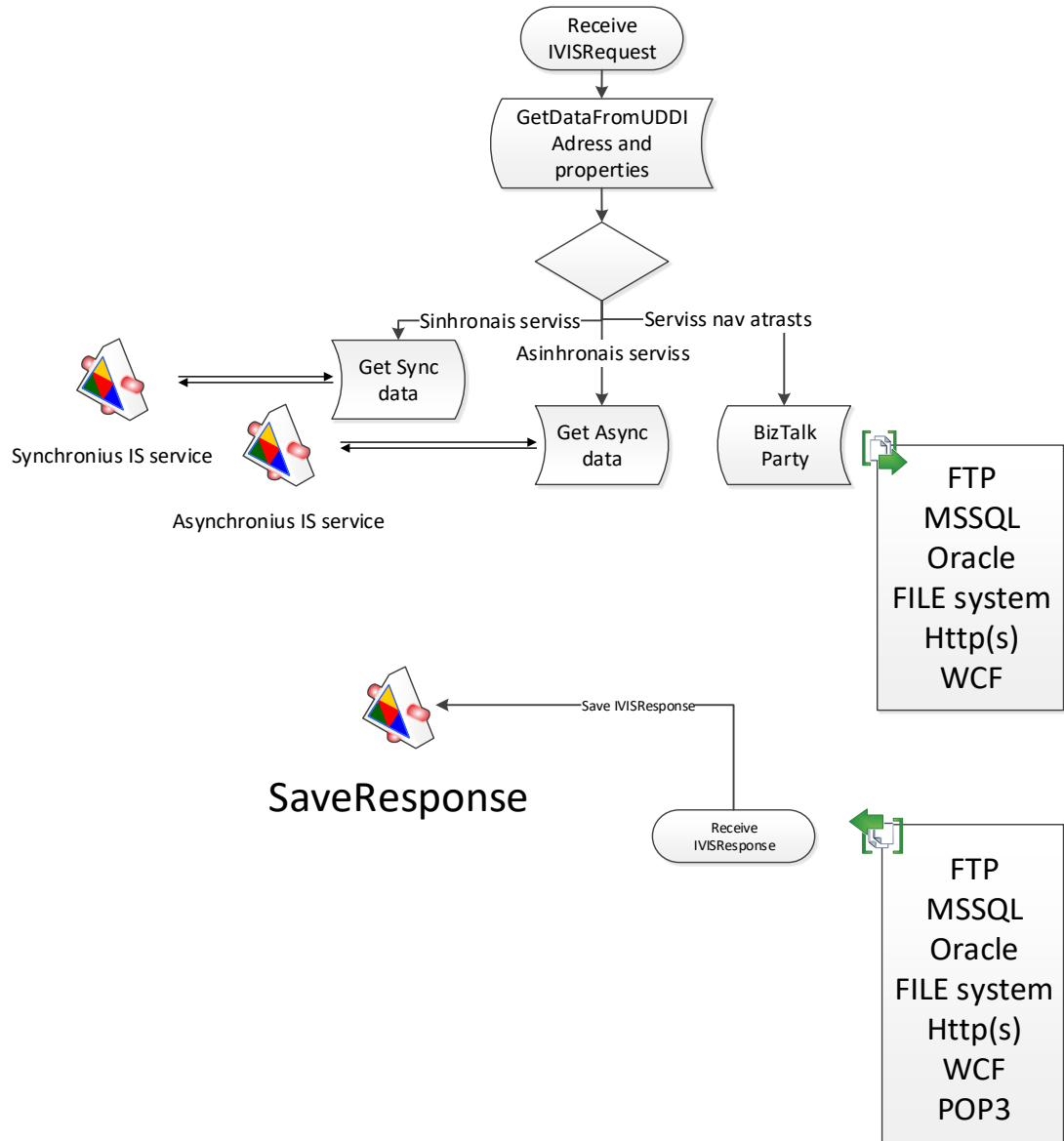
Universālo saskarni var attēlot kā tiltu starp sinhroniem un asinhroniem izsaukumiem, jo tas dot iespēju izsaukt sinhronus servissus, izmantojot asinhronu interfeisu, skat. 17.attēlā.



### 17.attēls. Tilts starp sinhroniem un asinhroniem izsaukumiem

Universālās asinhronās saskarnes izmantošanai ir divas alternatīvas iespējas, skat. 18.attēlā:

1. Sinhronā Integrācijas IS servisa asinhronais izsaukums („Get Sync data”);
2. Pieprasījuma ievietošana (tai skaita, izmantojot asinhrono integrācijas IS servisu) un rezultāta saņemšana, izmantojot BizTalk adapteru (Get Async data un BizTalk Party).
3. Gadījuma, ja netiek atrasts atbilstošais sinhronais vai asinhronais Integrācijas IS serviss, tiek uzskatīts, ka pieprasījums jāveic izmantojot publicēto BizTalk party ierakstu.



18.attēls. SASKARNES DARBĪBAS ALGORITMS

### 3.13.1. *Sinhronā Integrācijas IS servisa asinhronais izsaukums*

Lai nodrošinātu Sinhronā Integrācijas IS servisa asinhrono izsaukumu ir nepieciešams (1. un 2. punktu izpilda Integrācijas IS servisa izstrādātājs – datu devējs):

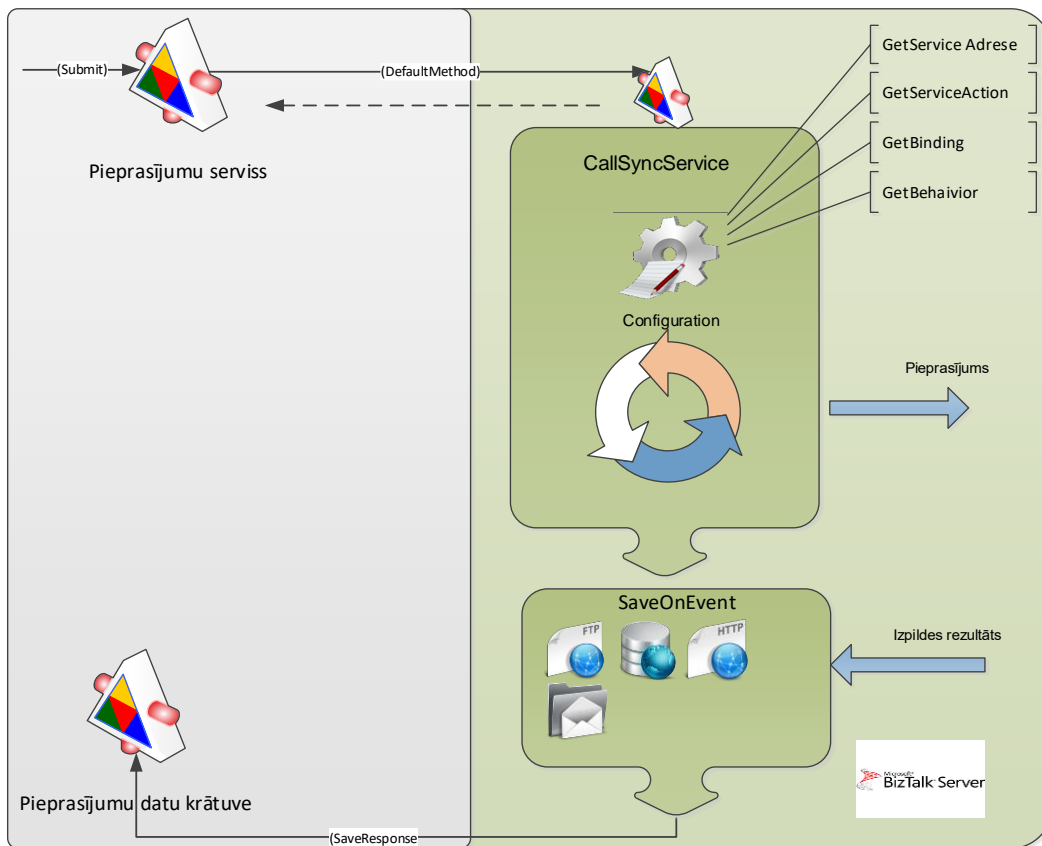
1. Izveidot sinhronu Integrācijas IS servisu;
2. Piereģistrēt to IS servisu katalogā;
3. Veikt `Submit` metodes izsaukumu, `Destination` atribūtā norādot pierēģistrētā **sinhronā** Integrācijas IS servisa URN.

Gadījumā, ja sinhronais izsaukums neizpildās laicīgi, VRAA sistēmas administratoram vienmēr būs iespēja veikt servisa darbības atjaunošanu, izmantojot BizTalk `Resume` komandu.

### 3.13.2. *Pieprasījuma ievietošana un rezultāta saņemšana, izmantojot BizTalk adapteri*

Pieprasījuma ievietošana un rezultāta saņemšana, izmantojot BizTalk adapteri, ir nepieciešama gadījumā, kad tiek veidots, integrācijas process VISS infrastruktūras ietvaros. Pieprasījuma

ievietošana un rezultāta saņemšana notiek kā divas neatkarīgas darbības, skat. 19.attēlā: CallSyncService un SaveOnEvent.



19.attēls. SaveResponse paplašināšana

Pieejamo BizTalk adapteru uzskaitījums ir aprakstīts 9.tabulā.

9.tabula

SaveResponse adapteru uzskaitījums

ADAPTERI	ADAPTERU APRAKSTS	CALLSYNCSERVICE	SAVEONEVENT
SOAP	Asinhrons Integrācijas IS serviss.	X	X
SQL	Adapteris saņēma/nosūtīja informāciju no MSSQL servera balstoties uz savienošanas rindu (connection string) un tabulu, no kuras ņem informāciju	X	X
HTTP	Saņēma/nosūtīja informāciju caur HTTP	X	X
FTP	Saņēma/nosūtīja informāciju no FTP	X	X
POP3	Nosūta informāciju caur POP3 protokolu		X
Oracle	Adapteris saņēma/nosūtīja informāciju no Oracle servera, balstoties uz savienošanas rindu (connection string) un tabulu, no kuras ņem informāciju	X	X

Abas darbības varētu tikt veiktas, izmantojot jebkādu adapteru kombināciju, piemēram, pieprasījuma sūtīšana varētu tikt veikta izmantojot SOAP, bet atbildes saņemšana izmantojot FTP.

Pieprasījuma ievietošanas gadījumā (CallSyncService), kā ienākošais parametrs tiek padots VISS (IVIS) „IVISRequest” pakotne.

Atbildes saņemšanas gadījumā (SaveOnEvent) adapteris, kā ienākošo parametru saņem „IVISResponse” ziņojuma pakotni. Saņemot ziņojumu, tas to pārsūta uz SaveResponse servisu

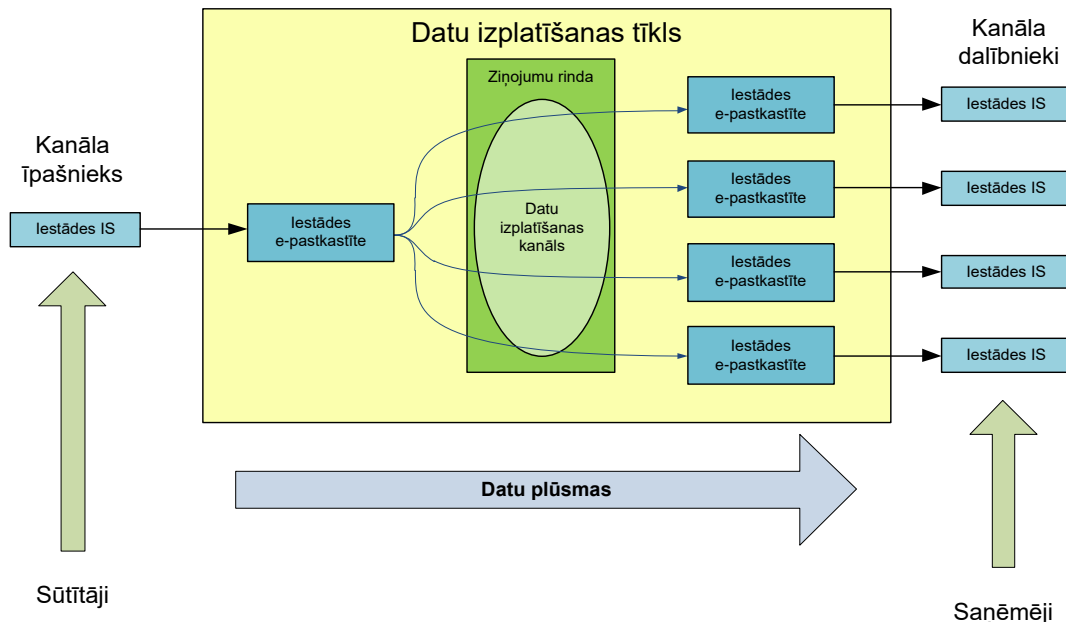
(asinhronu datu krātuvē) metodei *SaveResponse*. Servisa izsaukšana var ievilkties līdz brīdim, kad serviss būs pieejams. VRAA administrators var atkārtot izsaukšanu, ja serviss nedarbojas.



## 4. Datu izplatīšanas tīkls

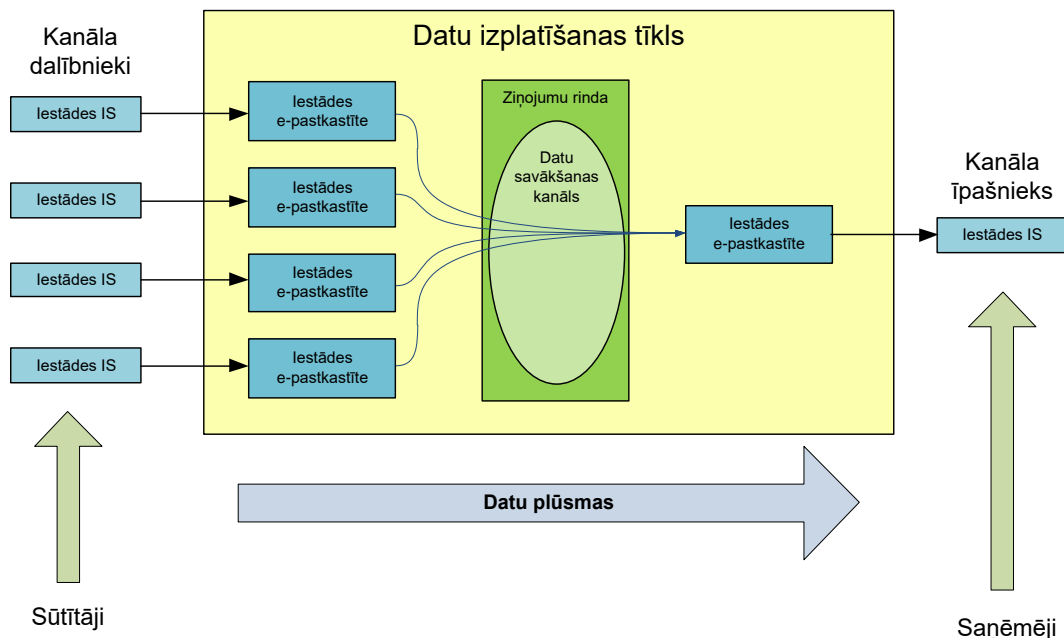
Datu izplatīšanas tīkla risinājuma ietvaros informācijas apmaiņa notiek, izmantojot kanālus, kur katrs kanāls nodrošina ziņojumu pārsūtīšanu no viena īpašnieka vairākiem dalībniekiem (DIK kanāls) vai no vairākiem dalībniekiem vienam īpašniekam (DSK kanāls):

- Datu izplatīšanas kanāls (DIK), skat. 20.attēlā, pa kuru ziņojumi plūst no viena sūtītāja, kas ir kanāla īpašnieks, vairākiem saņēmējiem, kas ir kanāla dalībnieki.



20.attēls. Datu izplatīšanas kanāls (DIK)

- Datu savākšanas kanāls (DSK), pa kuru ziņojumi plūst no vairākiem sūtītājiem, kas ir kanāla dalībnieki, vienam saņēmējam – kanāla īpašniekam.



21.attēls. Datu savākšanas kanāls (DSK)

Jebkuram kanālam ir konfigurēts (kanāla izveidošanas laikā), kāda tipa ziņojumus pa to var sūtīt. Katrs kanāls nodrošina Biznesa ziņojumu vai Transakcijas ziņojumu sūtīšanu.

- Biznesa ziņojums. Parastais ziņojums, kas nesaistās ar citiem ziņojumiem kanāla ietvaros.
- Transakcijas ziņojums. Standarta (klasificētā) sakārtota operāciju kopa, kas tiek pielietota noteiktam datu masīvam tā aktualizācijai (replikācijai). Katrs transakcijas ziņojuma ieraksts parasti satur DML (*Data manipulation language*) operācijas kodu (*insert, update, delete*). Viena kanāla versija varēs nodrošināt darbu vienai datu replikācijai.

Detalizēti skat. [9] dokumentā.

#### **4.1. Nepieciešamās tiesības DIT izmantošanai**

Tiesības DIT infrastruktūras izmantošanai VRAA piešķir iestādes lietotājam pēc oficiālas tiesību pieprasījuma veidlapas saņemšanas. Tiek nodefinētas vismaz divas lomas DIT iestāžu lietotājiem:

- DIT operators - var veidot jaunus kanālus, pārvaldīt esošos pieslēgumus, sūtīt un saņemt ziņojumus utt.
- DIT lasītājs - var sūtīt un saņemt ziņojumus.

Izmantot DIT eksponēto FTPs saskarni var tikai VISS lietotāji, kam tiek piešķirta VISS lietotāja vārda/parole autentifikācija.

#### **4.2. DIT lietošanas scenārijs**

DIT lietošanā parasti ir iespējami divi scenāriji:

- Parakstīšanās uz esošu citas iestādes kanāla plūsmu;
- Jaunā kanāla izveidošana, lai to varētu lietot citas iestādes.

##### ***4.2.1. Parakstīšanās uz esošu citas iestādes kanāla plūsmu***

Parakstīšanās uz esošu citas iestādes kanāla plūsmu varētu notikt atbilstoši šādiem scenārijiem:

- Iestāde-kanāla īpašnieks, pievieno iestādi-pretendentu pie tiem, kas pierakstījušies kanālam. To var izdarīt, izmantojot VISS portālā pieejamo DIT lietotni.
- Iestāde-pretendents izsūta pieprasījumu par pievienošanu noteiktam DIT kanālam, ko pēc tam iestāde-kanāla īpašnieks apstiprina.

Pēc parakstīšanās uz DIT kanālu, iestādes IS var sūtīt un saņemt ziņojumus, izmantojot DIT web servisu saskarnes vai FTPs protokolu (skat. [9] dokumentā).

##### ***4.2.2. Jaunā kanāla izveidošana***

Jaunā DIT kanāla izveidošana notiek izmantojot VISS portālā pieejamo DIT lietotni.

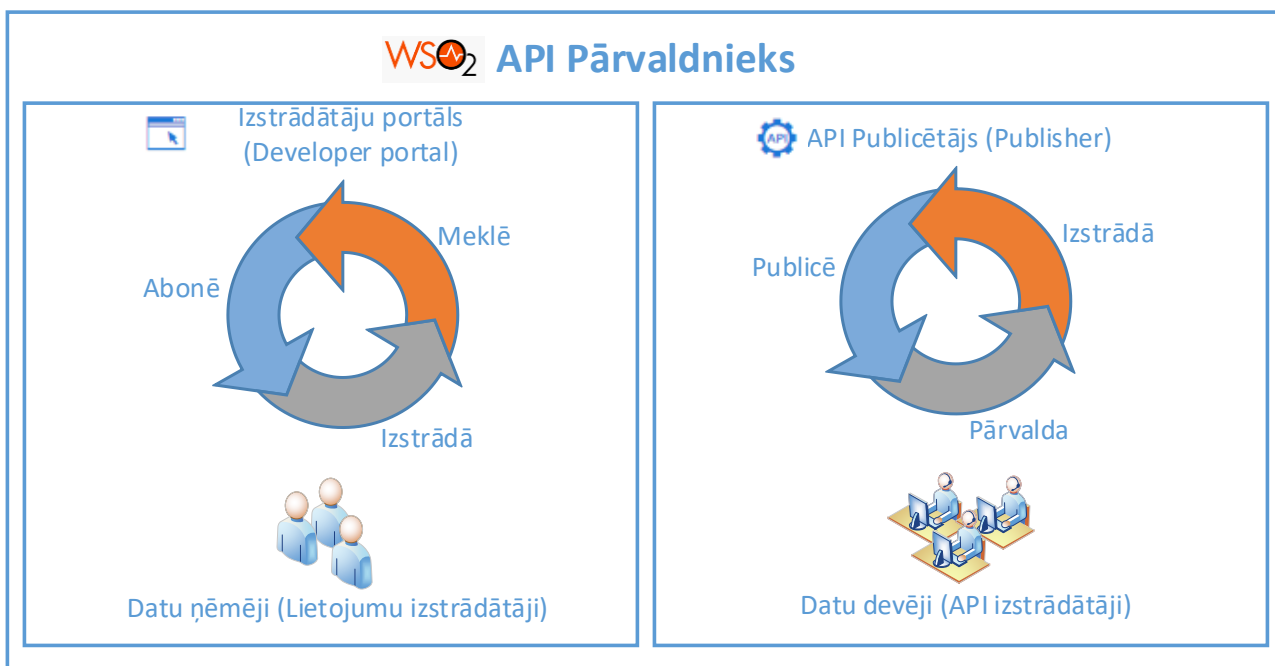
## 5. API Pārvaldnieks

API Pārvaldnieks ir Pieprasījuma servisa (skat. 3. nodaļumu) tehnoloģiskais pēctecis, tomēr pārejas posmā abi šie risinājumi darbojas paralēli.

**Pieeja IS servisiem un citām iestādēm datu-sniedzējiem, ārējām saskarnēm nākotnē būs iespējama, tikai lietojot API, kas veidos centrālo ieejas punktu VISS infrastruktūrā.**

API Pārvaldnieks ir uz WSO2 *enterprise service bus*<sup>8</sup> standarta risinājuma bāzēts, un tas nodrošina izstrādātājiem divas galvenās pamatfunkcijas:

1. Iespēju API izstrādātājiem publicēt vienuviet izstrādātos servisos un nodrošināt centralizētu to pārvaldību un monitoringu.
2. Iespēju lietojumu izstrādātājiem atrast nepieciešamos servisos, lai integrētu tos savos risinājumos.



### 22.attēls. API Pārvaldnieka risinājuma koncepts

Šī nodaļuma ietvaros tiek nodrošināts:

- API Pārvaldnieka darbības principu ieskats, skat. 5.1. nodaļu;
- API Pārvaldnieka tipisko lietošanas scenāriju, skat. 5.2. nodaļu;
- Nepieciešamo darbību kopumus ("road map"), lai datu ņēmēji varētu izmantot API Pārvaldniekā publicētos servisos, skat. 5.3. nodaļu.
- Nepieciešamo darbību kopumus ("road map"), lai datu devēji varētu publicēt un pārvaldīt savus servisos API Pārvaldniekā, skat. 5.4. nodaļu;

Pieeja pie izstrādātāju portāla VRAA testa vidē: <https://apitest.vraa.gov.lv/devportal/apis>.

Pieeja pie publicētāju portāla VRAA testa vidē: <https://apitest.vraa.gov.lv/publisher>.

Pieeja pie izstrādātāju portāla VRAA produkcijas vidē: <https://api.viss.gov.lv/devportal>.

Pieeja pie publicētāju portāla VRAA produkcijas vidē: <https://api.viss.gov.lv/publisher>.

<sup>8</sup> [https://en.wikipedia.org/wiki/Enterprise\\_service\\_bus](https://en.wikipedia.org/wiki/Enterprise_service_bus)

Drošības talonu var pieprasīt no šādas adreses VRAA testa vidē:

- <https://ha.vraa.gov.lv/STS/VISS.Pfas.STS/oauth2/token>

Drošības talonu var pieprasīt no šādas adreses VRAA produkcijas vidē:

- <https://ha.viss.gov.lv/STS/VISS.Pfas.STS/oauth2/token>

## 5.1. API Pārvaldnieka darbības principi

API Pārvaldnieka komponenti veido pielāgots WSO2 risinājums, un tas nodrošina:

- Iespēju datu devējiem reģistrēt savus REST un SOAP servišus API publicētāja (API publisher) saskarnē.
- Iespēju datu ņēmējiem reģistrēties Izstrādātāju portālā (Developer portal), lai izmantotu publicētos servišus.
- Drošības slāni izmantojot OAuth 2.0 protokolu ar VISS PFAS autentifikāciju.
- Pieprasījumu un atbilžu auditēšanu.
- Iespēju nodot OAuth2 references talonu, kas tiek izmantots API publicētājā (API publisher) publicēto servišu(API) izsaukšanai līdz pat biznesa servišam, lai veiktu papildus pieejas kontroli datiem biznesa serviša līmenī. Atkarīgs no konkrētā biznesa serviša realizācijas un konfigurācijas.

Pieprasījumu veikšanai nav nepieciešams izmantot noteiktas ziņojumu struktūras (kā tas bija Pieprasījumu serviša gadījumā).

***Pieprasījumu apjoma ierobežojums ir 4 MB. Ja nepieciešams pārsūtīt lielākus pielikumus, datnes utt., ir jālieto DIT (skat. 2.3.nodaļu) vai Elektroniskās dokumentu krātuves (EDK) risinājums (skat. [10] saistīto dokumentu). EDK izmantošanas scenāriji jāaskaņo ar VRAA uzsākot datu apmaiņas projektēšanu.***

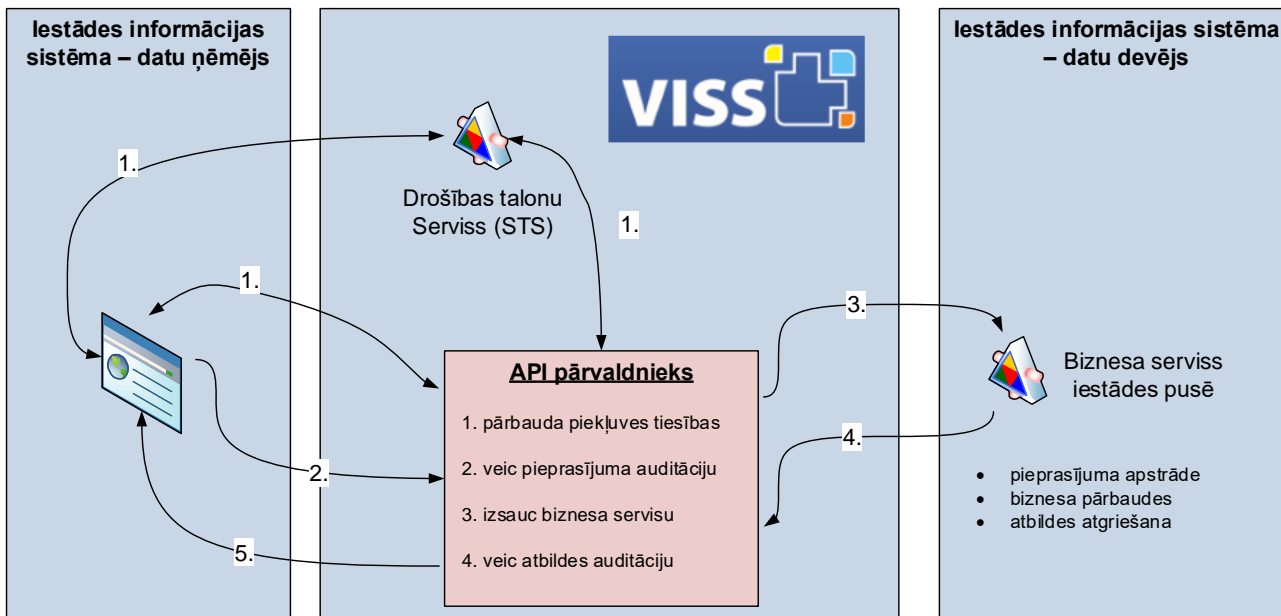
Anonīmi (bez drošības talona) pieprasījumi netiek nodrošināti.

API Pārvaldniekā publicētie (statusā - Published) serviši (API) ir publiski pieejami Izstrādātāju portāla (Developer portal) saskarnē (skat. 5.3. nodaļu), bet to reģistrēšana un publicēšana tiek veikta, izmantojot API publicētāja (API Publisher) saskarni (skat. 5.4. nodaļu), kas pieejama tikai autentificētiem lietotājiem.

VISS Integrācijas servišu kataloga papildināšana nav nepieciešama, izstrādājot servišus (API), kurus paredzēts eksponēt, izmantojot API Pārvaldnieku.

## 5.2. API Pārvaldnieka lietošanas scenārijs

API Pārvaldnieks nodrošina iespēju datu ņēmējiem veikt sinhronu biznesa servišu (API) izsaukumus. Visi publicētie API ir aizsargāti ar PFAS lietotāja autentifikāciju, un to izsaukšanai ir nepieciešams drošības talons.



**23.attēls. API Pārvaldniekā publicēta biznesa servisa izsaukums**

Sinhronais izsaukums ietver šādus soļus:

1. Autentifikācija un autorizācija VISS drošības talonu servisā (vai nu vēršoties pa tiešo pie PFAS AUTH vai arī pie PFAS AUTH caur API Pārvaldnieku), kur tiek saņemta autentifikācijas un autorizācijas apgalvojuma atsauce (drošības talons - reference);
2. Notiek pieprasījuma formēšana un publicētā biznesa servisa izsaukums nodod drošības talona referenci;
3. API Pārvaldnieks veic drošības talona pārbaudi griežoties pie PFAS STS ar saņemto talona referenci, pieprasījuma auditēšanu un datu devēja biznesa servisa izsaukumu;
4. Biznesa serviss veic pieprasījuma apstrādi un atgriež datus uz API Pārvaldnieku;
5. API Pārvaldnieks veic atbildes auditēšanu un atgriež to datu pieprasītājam.

### 5.3. Izstrādātāju portāls (Developer portal)

Datu ņēmēju (Developer portal) portālā ir pieejami API Pārvaldniekā publicētie (statusā - Published) servisi (API). Šajā nodaļā ir uzskaitīti soļi, kā datu ņēmējs (Lietojumu izstrādātājs) var izmantot publicētos sevisus (API).

#### 5.3.1. Publicēta biznesa servisa izmantošana

Lai izmantotu Izstrādātāju portālā (Developer portal) reģistrētus servisi, nepieciešams veikt šādas darbības:

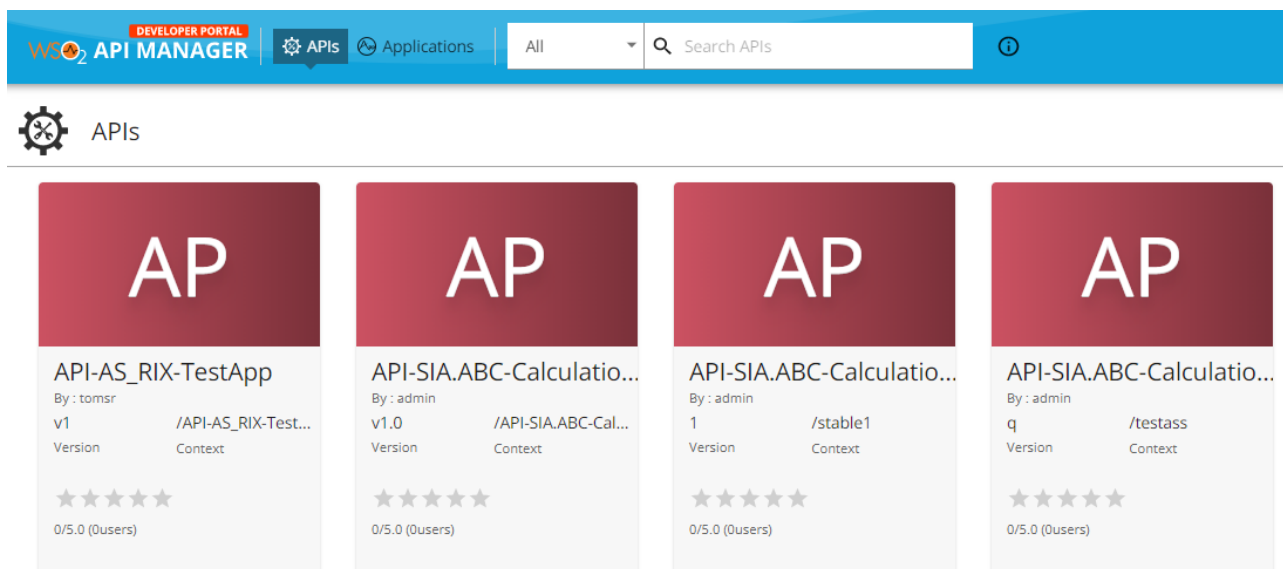
1. Jāsameklē sev nepieciešamo servisu Izstrādātāju portālā (Developer portal), skatīt 5.3.1.1 nodaļu.
2. Jāsaņem pieejas tiesības no VRAA izmantot Izstrādātāju portālā (Developer portal) publicētos servisi skatīt 5.3.1.2 sadaļu
3. Izstrādātāju portālā (Developer portal) jāreģistrē datu ņēmēja Klienta lietojums, kas izmantos API Pārvaldniekā eksponētos servisi, skatīt 5.3.1.3. sadaļu;
4. Jāveic servisu abonēšana, skatīt 5.3.1.4. sadaļu;
5. Nepieciešamības gadījumā jāaskaņo ar servisa īpašnieku tiesības izmantot izvēlēto API, skatīt 5.3.1.5;
6. Jāpielāgo Klienta lietojums servisa izsaukšanai, skatīt 5.3.1.6. sadaļu;

**E-pakalpojumu izstrādātājiem biznesa servisu izsaukšanai jāizmanto e-pakalpojumu platformas LvpContext.Request funkcionalitāte skat. e-pakalpojuma izveides ceļveža programmētāja rokasgrāmatu [28]. No šīs sadaļas e-pakalpojumu izstrādātājiem saistoša ir tikai pieejas tiesību pieprasīšana no VRAA (skatīt 5.3.1.2) un datu devējiem (skat. 5.3.1.5).**

### 5.3.1.1. Publicēto API pārlūkošana

Lai aplūkotu Izstrādātāju portālā (Developer portal) publicētos servisos (API), nepieciešams veikt šādas darbības:

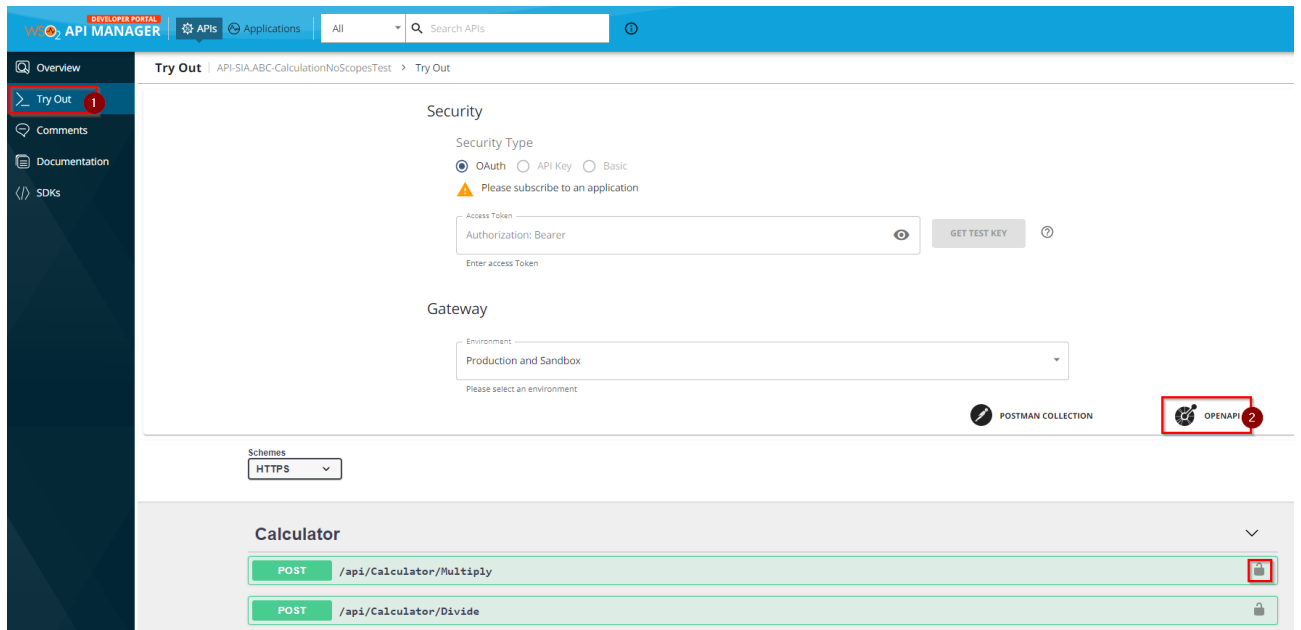
- Atvērt Izstrādātāju portāla (Developer portal) sadaļu APIS, tā pieejama arī neautenticētiem lietotājiem.
- APIS sadaļā pārlūkošanai ir pieejami visi publicētie servisi (API). Saskaņā nodrošina iespēju filtrēt publicētos servisos pēc statusa un nosaukuma.



#### 24.attēls. Izstrādātāju portālā (Developer portal) publicētie API

- iespējams atvērt servisa detalizētās informācijas skatu, cilnē Try Out (1) pieejama REST servisa swagger definīcija (2), pēc kuras var redzēt kuras metodes ir aizsargātas ar atļaujām (scope) un kuras nav. Tas tiek attēlots arī vizuāli ar atslēdzes ikonu ja izsaukumam ir nepieciešama atļauja (scope).

**Try Out cilni nevar izmantot servisu izsaukumu testēšanai, jo tā nenodrošina visu nepieciešamo header atribūtu aizpildīšanu, piemēram transakcijas identifikatora.**



25.attēls. Rest servisa detalizētais skats

### 5.3.1.2. Pieejas tiesību pieprasīšana no VRAA

Pieejas pieprasīšana un sniegšana Izstrādātāju portāla (Developer portal) servisu izmantošanai tiek veikta pēc noteiktas procedūras, skatīt [31].

Ja tiek veikta e-pakalpojuma izstrāde tiek izsniegts:

- E-pakalpojuma identifikators (URN);
- Klienta identifikatoru (clientid) un atslēga (secret) kas jāizmanto autentifikācijas nodrošināšanai;

Ja tiek veikta tehniskā procesa izstrāde tiek izsniegts:

- Pēc nepieciešamības, datu apmaiņas identifikators (URN);
- API pārvaldnieka lietotājs ar piekļuves iespēju pie izstrādātāju portāla (Developer portal);
- Pēc nepieciešamības PFAS lietotājs – nepieciešamo PFAS autentifikāciju izveidei vai autentifikācijas realizēšanai nepieciešamie rekvizīti;

***E-pakalpojumu gadījumā datu apmaiņai tiek izmantots e-pakalpojuma identifikators.***

***Servisu izsaukšanai no e-pakalpojuma jāizmanto LVP IDS izsniegto oauth2 references vai JWT (access) talonu.***

***E-pakalpojumi API pārvaldnieku sauc pastarpināti izmantojot e-pakalpojumu platformas mikroservisu, tāpēc tie nav jāreģistrē kā klienta lietojumi izstrādātāju portālā (Developer portal).***

***E-pakalpojumu platformas mikroservis ir reģistrēti kā klienta lietojums un starp to un visu e-pakalpojumu izmantotajiem servisiem ir jābūt izveidotiem abonementiem.***

***Aizsargāta API gadījumā datu devējam jāpiešķir atļaujas (scopes) e-pakalpojuma IDS uzticamajai pusei atbilstoši 5.4.4. sadaļai.***

### 5.3.1.3. Klienta lietojuma reģistrācija

Pēc pieejas tiesību saņemšanas, datu ņēmējs reģistrē savu Klienta lietojumu izstrādātāju portālā (Developer portal). Reģistrējot Klienta lietojumu, tiek automātiski izveidota atbilstoša uzticamā puse (Relying Party) PFAS STS. Lai reģistrētu jaunu Klienta lietojumu, jāveic šādas darbības:

1. Jāatver Izstrādātāju portālā (Developer portal) Applications saraksts;

Name ↑	Owner	Policy	Workflow Status	Subscriptions	Actions
AlvisTest	admin	Unlimited	ACTIVE	1	
APP-SIA_ABC-ApiManagementSamples-v1_0	admin	Unlimited	ACTIVE	21	
AlexTest66	admin	Unlimited	ACTIVE	2	
APP-SIA_ABC-RegulatedNameTest	admin	Unlimited	ACTIVE	0	
APP-SIA_ABC-MyApplication-v1_0	admin	Unlimited	ACTIVE	0	
APP-PAP_Madonas_PD-TestApp-v1	admin	Unlimited	ACTIVE	0	

## 26.attēls. Izstrādātāju portālā (Developer portal) reģistrētie klienta lietojumi

- Jānospiež saiti “ADD NEW APPLICATION”;
- Jānorāda Klienta lietojuma nosaukums, (PFAS STS tiks izveidota atbilstoša uzticamā puse ar šādu nosaukumu), pieprasījumu limitu, aprakstu un jānospiež poga “Save”.

**Klienta lietojuma nosaukums jāveido vadoties pēc Klienta lietojuma nosaukuma formāta notācījas: “APP- $\{$ Autoritātes identifikators (visscore:authorityIdent) no iestāžu un struktūrvienību klasifikatora (Authority), izmanto apakšvītru “\_” punktu vietā, piemēram, “SIA\_lestade1” $\}$ - $\{$ datu ņēmēja klienta lietojuma nosaukums bez speciālajiem simboliem, kurā konkrētais serviss (API) tiks izmantots, piemēram, “TapisUI” u.tml. $\}$ ”. Piemērs: “APP-SIA\_lestade1-TapisUI”.**

## Create an application

Create an application providing name and quota parameters. Description is optional.  
Required fields are marked with an asterisk ( \* )

Application Name \*

Enter a name to identify the Application. You will be able to pick this application when subscribing to APIs

Shared Quota for Application Tokens \*

Assign API request quota per access token. Allocated quota will be shared among all the subscribed APIs of the application.

Application Description

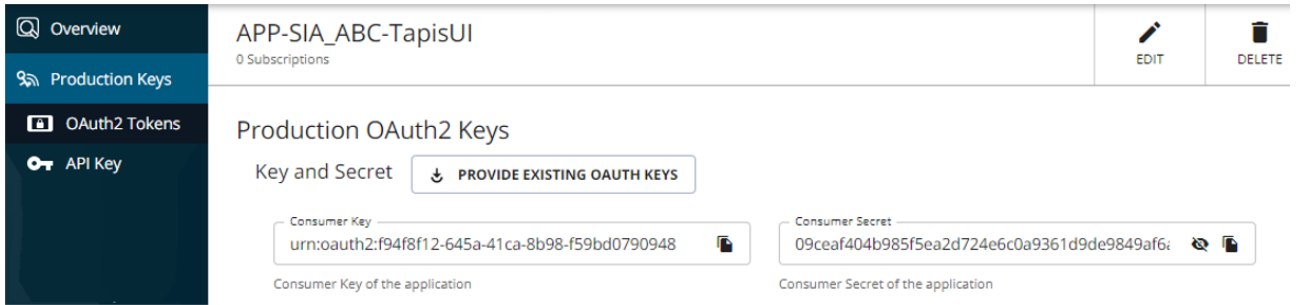
( 512 ) characters remaining

SAVE
CANCEL

## 27.attēls. Klienta lietojuma izveides forma

- Pēc Klienta lietojuma izveides, tiks atvērts tā Overview skats. Jāpāriet uz cilni “Production Keys”, token type jā saglabā vērtība DEFAULT un jānospiež poga “Generate keys”. Tiks uzģenerēti un PFAS saglabāti Consumer Key un Consumer Secret (skat. 28.attēlu). Šie atribūti vēlāk būs nepieciešami, lai pieprasītu talonu pirms servisa (API) izsaukšanas.





## 28.attēls. Izveidotā klienta lietojuma/PFAS uzticamās puses talona izgūšanas dati

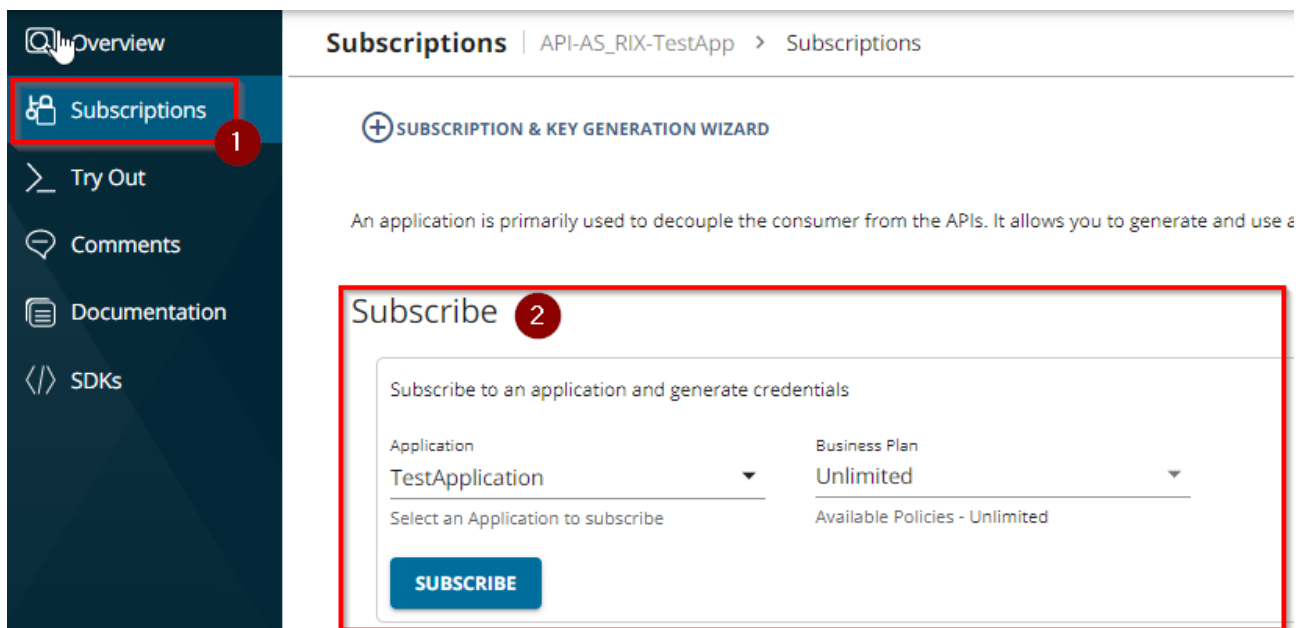
5. Pēc klienta lietojuma reģistrācijas klienta lietojuma kartiņā un atslēgu noģenerēšanas. Ir iespējams iegūt talonu no PFAS STS, skatīt 5.6.2 sadaļu.
6. Ja Klienta lietojums paredzēts servisa (API) metožu, kas aizsargātas ar scopes (pielaidēm), izsaukšanai, tad šajā brīdī (atbilstoši VISS programsaskarnes (API) abonēšanas procedūrai) ir nepieciešams vērsties pie attiecīgā Datu devēja ar lūgumu pievienot scopes reģistrētajam Klienta lietojumam, skatīt 5.3.1.5.

### 5.3.1.4. Servisu abonēšana

Pēc Klienta lietojuma reģistrācijas datu ņēmējs veic Klienta lietojuma sasaisti (abonē) ar publicēto servisu (API), izmantojot Izstrādātāju portālu (Developer portal). Viens Klienta lietojums var izmantot vairākus servisu (API) un katrs serviss (API) ir jāpiesaista Klienta lietojumam – šādā gadījumā visi šie servisi (API) tiks saukti no viena un tā paša klienta lietojuma. Pieprasot talonu vienam klienta lietojumam lai izsauktu servisu no tā var izmantot dažādus PFAS lietotājus/sertifikātus. Ja autentifikācijai tiek izmantots sertifikāts tad rekomendējams katrā klienta lietojumā izmantot vienu sertifikātu.

Lai veiktu publicēta servisa abonēšanu:

1. Jāatver Izstrādātāju portālā (Developer portal) publicēto servisu (API) saraksts, skatīt 5.3.1.1.;
2. Jāizvēlas vienu publicēto servisu (API) no saraksta un jāatver tā kartiņas cilne Subscriptions, skatīt 29. attēlu;



## 29.attēls. Publicēta servisa (API) kartiņa

- Laukā Application jāizvēlas Klienta lietojums, kurš izmantos šo servisu un jānospiež poga Subscribe, skatīt 30. attēlu;

Application	Business Plan
MyApplication	Unlimited
Select an Application to subscribe	Available Policies - Unlimited

**SUBSCRIBE**

### 30.attēls. Publicēta servisa (API) piesaiste lietojumam - abonēšana

#### 5.3.1.5. Servisa izmantošanas tiesību pieprasīšana no datu devēja

Ja pieejas tiesības jāpiešķir e-pakalpojumam:

- Pēc tam kad no VRAA ir saņemts e-pakalpojumam piešķirtais klienta identifikators (clientId – IDS uzticamā puse) datu ņēmējam ir jākontaktējas ar datu devēju, kontaktinformācija ir pieejama servisa (API) kartiņas detalizētā skata Overview cilnes Business Info laukā.

The screenshot shows the 'API Manager' interface. The 'Overview' tab is selected, displaying details for the API 'API-SIA.ABC-CalculationNoScopesTest'. The 'Business Info' section is highlighted with a red box, showing the contact email 'info@vraa.lv'. Other visible elements include the 'TRY OUT' button, the URL 'https://wsogw1-dev-vraa.abc:8244/API-SIA.ABC-CalculationNoS...', and the 'Business Plans' section showing 'Unlimited'.

### 31.attēls. Datu devēja kontaktinformācija

- Pieprasot tiesības no datu devēja jānorāda:
  - Servisa (API) kuru vēlaties izmantot nosaukums, piemēram, API-SIA\_ABC-TestApi;
  - Servisa (API) kuru vēlaties izmantot versija, piemēram, v1\_0;
  - Klienta (clientId) jeb IDS uzticamās puses identifikators, piemēram, urn:oauth2:test-vraa:myapplication;
  - Saraksts ar atļauju (scope) identifikatoriem, pieejami swagger definīcijā, piemēram, API-SIA\_ABC-TestApi-v1\_0-Scope1
- Datu devējs izmantojot PFAS administratora saskarni pēc norādītā klienta identifikatora sameklē IDS uzticamo pusi un piesaista tai norādītās atļaujas, scope.

Ja pieejas tiesības jāpiešķir tehniskajam procesam:

- Pēc tam, kad datu ņēmējs izstrādātāju portālā (Developer portal) ir reģistrējis klienta lietojumu un noģenerējis atslēgas, datu devējs var veikt pieejas tiesību piešķiršanu konkrētajam klienta lietojumam.

2. Datu ņēmējs kontaktējas ar datu devēju, kontaktinformācija ir pieejama servisa (API) kartiņas detalizētā skata Overview cilnes Business Info laukā.

### Business Info

info@vraa.lv

#### 32.attēls. Datu devēja kontaktinformācija

3. Pieprasot tiesības no datu devēja jānorāda:
  - a. Servisa (API) kuru vēlaties izmantot nosaukums, piemēram, API-SIA\_ABC-TestApi;
  - b. Servisa (API) kuru vēlaties izmantot versija, piemēram, v1\_0;
  - c. Klienta (Consumer Key no 5.3.1.3 nodaļas) jeb PFAS uzticamās puses identifikators, piemēram, urn:oauth2: 6a3a2483-6666-4444-2222-60a3aef21363;
  - d. Saraksts ar atļauju (scope) identifikatoriem, pieejami swagger definīcijā, piemēram, API-SIA\_ABC-TestApi-v1\_0-Scope1
4. Datu devējs izmantojot PFAS administratora saskarni pēc norādītā klienta identifikatora sameklē PFAS uzticamo pusi un piesaista tai norādītās atļaujas, scope.

#### 5.3.1.6. Lietojuma pielāgošana servisa izsaukšanai

Pēc servisa (API) abonēšanas (skat. 6. sadaļā) servisu var izsaukt no reģistrētā Klienta lietojuma. Publicētie servisi ir izsaukami tikai drošā veidā, izmantojot OAuth2 autorizāciju ar PFAS "Lietotāja vārds un parole" vai sertifikāta autentifikāciju, detalizētai informācijai skatīt .Net, PHP un Java izsaukumu piemērus 5.6 nodaļā.

*E-pakalpojumu izstrādātājiem biznesa servisu izsaukšanai jāizmanto e-pakalpojumu platformas LvpContext.Request funkcionalitāte skat. e-pakalpojuma izveides ceļveža programmētāja rokasgrāmatu [28]. No šīs sadaļas e-pakalpojumu izstrādātājiem saistoša ir tikai pieejas tiesību pieprasīšana no VRAA (skatīt 5.3.1.2) un datu devējiem (skat. 5.3.1.5).*

Lai iegūtu OAuth2 drošības talonu ar lietotāja autentifikāciju datu ņēmējam ir nepieciešama šāda informācija:

- consumer-key – API Pārvaldniekā reģistrētā Klienta lietojuma atslēga, piemēram „urn:oauth2:fa49e5c0-2345-4fa1-b9a3-08ccf46e2d3e”. Tā PFAS administratoriem tiek attēlota PFAS uzticamās puses detalizētajā skatā (29.attēls) kā „Identifikators” vērtība un datu ņēmējiem reģistrētā Klienta lietojuma detalizētajā skatā API Pārvaldnieka Izstrādātāju portālā (Developer portal) saskarnē (30.attēls). Tas tiek ģenerēts, reģistrējot klienta lietojumu, skatīt 5.3.1.3. sadaļu

Vis > Administrēšana > PFAS > Uzticamās puses > PFAS uzticamo pušu pārvaldība >

## Detalizēta informācija par PFAS uzticamo pusi

Izmantojiet šo lapu uzticamās puses informācijas detalizētai apskatei.

Uzticamās puses īpašības	Galapunkti	Sertifikāti	Pielaiides
Nosaukums:	ApiManagementSamples		
Identifikators:	urn:oauth2:825e807d-207f-4793-98b4-d8a1f20c9a51		
Aktīva:	<input checked="" type="checkbox"/>		
Drošības talona noklusētais tips:	OAuth 2.0 Reference Token		
Drošības talona dzīves laiks:			
Atslēga:	uGTzNdMEAO56ITy3VZgo2F5ePjvLAbsuQ02h8mfaZvk=		
Atslēgas derīguma termiņš:	20.08.4757		
Tēma:			
Apraksts:			

### 33.attēls. PFAS AUTH lietojuma PFAS uzticamās puses detalizētais skats

### 34.attēls. Klienta lietojuma detalizētais skats API Pārvaldnieka saskarnē

- consumer-secret - API Pārvaldniekā reģistrētā klienta lietojuma noslēpums, piemēram „jpXV42ntpHYfdObK2cissI2XdHmsFyAmbw3PNZvn”. Tas tiek attēlots PFAS administratoriem VISS portālā PFAS uzticamās puses detalizētajā skatā (29.attēls) kā „Atslēga” vērtība un datu ņēmējiem reģistrētā klienta lietojuma detalizētajā skatā API Pārvaldnieka Izstrādātāju portāla (Developer portal) saskarnē (30.attēls). Tas tiek ģenerēts, reģistrējot klienta lietojumu, skatīt 5.3.1.3.sadaļu.
- username – PFAS lietotāja vārds, piemēram „T2T\_ABerzins” To veido VRAA, balstoties uz datu ņēmēja pieteikumu, skatīt 5.3.1.1.sadaļu Pēc pieteikuma apstiprināšanas, tiek nosūtīts pieteikumā norādītajā e-pastā.
- password – PFAS lietotāja parole, piemēram „Dr0\$@\_p@r0le\_I2E”. Pēc pieteikuma apstiprināšanas tiek nosūtīta pieteikumā norādītajā e-pastā, skatīt 5.3.1.1.sadaļu.
- scope – Ja tiek izsaukts ar scopes – neaizsargāts serviss (API), šis parametrs nav jānorāda. Ja tiek izsaukts serviss (API), kas tiek aizsargāts ar scopes tad OAuth2 drošības talonu datu pieprasījumā jānorāda nepieciešamie scopes. Ja scopes netiks norādītas, tad saņemtajam talonam šīs scopes nebūs pievienotas un ar šo talonu servisu (API) izsaukt nebūs iespējams.

Lai iegūtu OAuth2 references talonu no PFAS STS, nepieciešams izpildīt šādu pieprasījumu, norādot iepriekš uzskaitīto parametru vērtības:

```
curl -X POST -v https://ha.viss.gov.lv/STS/VISS.Pfas.STS/oauth2/token -u
" fa49e5c0-2345-4fa1-b9a3-08ccf46e2d3e:
jpXV42ntpHYfdObK2cissI2XdHmsFyAmbw3PNZvn" -d "grant_type=password" -data-
urlencode "username=T2T_ABerzins" -data-urlencode
"password=Dr0$@_p@r0le_I2E" --data-urlencode
"scope=VISS.Module1_CreateSomeObject"
```

Šis pieprasījums atbildē atgriezīs references talona identifikatoru atribūtā „access\_token”:

```
{
  "access_token": "urn:uuid:34c9bc8a-ffb1-46cf-9133-cb276888b6c9",
  "expires_in": 3600,
  "scp": "VISS.Module1_CreateSomeObject",
  "token_type": "bearer"
}
```

Izmantojot saņemto talona referenci, iespējams izsaukt API Pārvaldniekā reģistrēto servisu (API), piemēram, tādu, kas sareizina divus skaitļus, norādot tā ieejas parametros 3 un 4 (transakcijas identifikators ir obligāts, obligātumu iespējams atslēgt konkrētam API, to iepriekš saskaņojot ar VRAA):

```
curl -H "Authorization: Bearer urn:uuid:34c9bc8a-ffb1-46cf-9133-cb276888b6c9" -H "Content-Type: application/json" -H "x-transactionId: URN:IVIS:100001:EP-EP186-v1-0-TR-1584" -X POST -d '{"Multiplier": 3, "Multiplicand": 4}'
"http://192.168.102.36:8281/VISS.ApiManagement/Extensions/CalculationApi/table/v1/api/Calculator/Multiply"
```

Lai varētu aizpildīt biznesa datus, jāizmanto API Pārvaldniekā Izstrādātāju portālā (Developer portal) pieejamo informāciju vēlamā servisa (API) kartiņā, REST servisiem skatīt cilnē „Try Out” pieejamo swagger definīciju, SOAP servisiem izmantot cilnē Overview pieejamo WSDL (32.attēls).

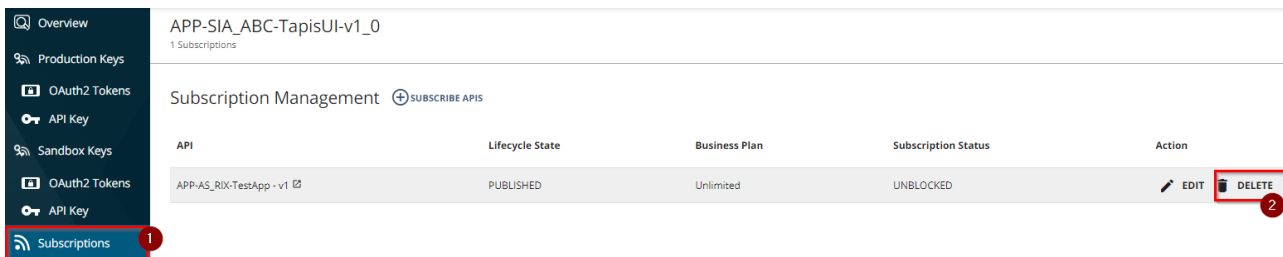
### 35.attēls. REST servisa swagger definīcija

### 36.attēls. SOAP servisa WSDL lejupielāde

### 5.3.2. **Publicēta biznesa servisa izmantošanas pārtraukšana**

Klienta lietojums var pārtraukt kāda konkrēta servisa izmantošanu, piemēram, notiek pāreja uz jaunāku servisa versiju vai citu servisu, jo mainās lietojuma biznesa procesi.:

1. Ja notiek pārslēgšanās uz citu servisu vai tā versiju, tai jābūt nopublicētai attiecīgajā vidē.
2. Tiek veiktas izmaiņas klienta lietojuma kodā, un tas izmitināts attiecīgajā vidē.
3. Pārtrauc servisa (API) abonēšanu atbilstošajā vidē, veicot šādas darbības:
  - a. Atver Izstrādātāju portālā (Developer portal) reģistrēto klienta lietojumu (Application) sarakstu, skatīt 26.attēls;
  - b. Izvēlas reģistrēto klienta lietojumu (Application) no saraksta un jāatver tā kartiņu. Kartiņā pāriet uz cilni Subscriptions skatīt 37.attēls;



#### 37.attēls. Izstrādātāju portālā (Developer portal) reģistrētā klienta lietojuma kartiņa

- c. Sarakstā sameklē servisu (API), kura izmantošana ir pārtraukta, nospiež kolonnā Actions redzamo saiti Delete un pārtraukšanas apstiprinājuma paziņojumā nospiež pogu Delete.

## 5.4. **Publicētāju portāls (API Publisher)**

Datu devēju (API Publisher) portālā tiek reģistrēti API Pārvaldniekā eksponētie servisi (API). Pēc to publicēšanas tie ir publiski pieejami Izstrādātāju portāla (Developer portal) saskarnē. Šajā nodaļā ir uzskaitītas prasības un soļi, kā datu devējs (API izstrādātājs) var reģistrēt un publicēt savus servisi (API) API Pārvaldniekā.

Lai datu devējs eksponētu savu servisu caur API Pārvaldnieku jāveic šādas darbības:

1. Jāizstrādā servisi atbilstoši specifikācijai, skatīt 5.4.2;
2. Jāpieprasa VRAA pieejas tiesības pie API Publisher portāla, skatīt 5.4.1;
3. Jāreģistrē serviss (API) API Publisher portālā;
4. Jāpublicē reģistrētais serviss, lai tas būtu atrodams Izstrādātāju portālā (Developer portal);

### 5.4.1. **Pieejas tiesību pieprasīšana no VRAA**

Tā kā datu devēju / publicētāju (API Publisher) portāls ir pieejams tikai autentificētiem API Pārvaldnieka lietotājiem, tad, lai varētu pieslēgties portālam, reģistrēt un publicēt servisi (API), no VRAA ir jāpieprasa piekļuves tiesības:

1. Datu devēji rakstiski pieprasa tiesības servisu izmantošanai no VRAA, atbilstoši VISS Programsaskarnes (API) publicēšanas procedūrai.
2. VRAA datu devējiem piekļuvi pie Publicētāju portāla (API Publisher) dod tikai tad, kad VRAA no datu devēja ir saņēmis rakstisku pieprasījumu.
3. VRAA saņemot datu devēja tiesību pieteikumu:
  - a. Reģistrē datu devēja iestādi lestažu un struktūrvienību (Authority) klasifikatorā.

- b. Izveido API Pārvaldnieka risinājuma lietotāju. Piešķir tam tiesības Publicētāju portālā (API Publisher) reģistrēt un publicēt servišus (API).
  - c. Izveido datu devēju iestādei atbilstošu PFAS iestādes administratoru, kurš savas iestādes ietvaros varēs piešķirt datu ņēmēju klientu lietojumiem (PFAS un IDS uzticamajām pusēm) atļaujas izmantot savus servišus.
4. VRAA informē datu devēju par veiktajām darbībām un tam izsniedz:
    - a. Piešķirto iestādes 6 zīmju identifikatoru un iestādes kodu.
    - b. API publicētāja piekļuves.
    - c. PFAS piekļuves datus.

**E-pakalpojumu gadījumā Klienta lietojumus kā IDS uzticamās puses PFAS reģistrē VRAA.**

### 5.4.2. Servišu (API) izstrāde

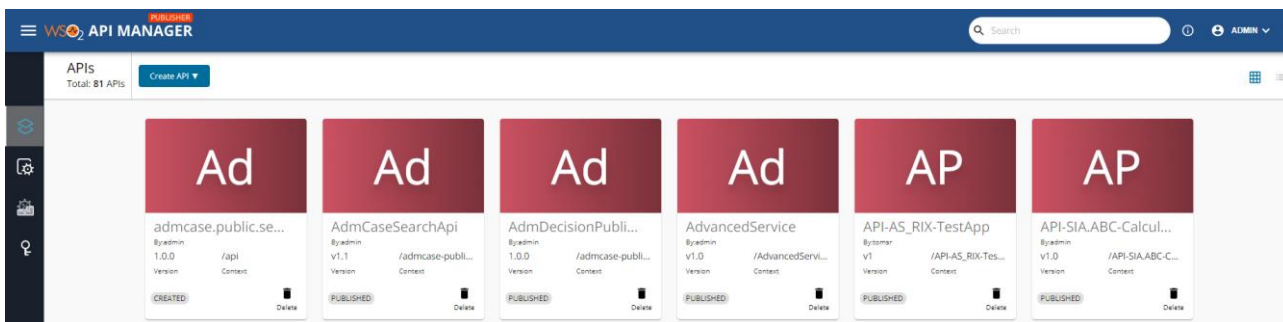
Izmantojot API Pārvaldnieku, biznesa servišiem nav nepieciešams veidot integrācijas servišu starpslāni, bet biznesa servišus izstrādāt, ievērojot **Open API Specification 2.0** vai **Open API Specification 3** specifikāciju un reģistrējot API Publicētājā (API Publisher).

Specifikācija nosaka, ka katram servišam (API) jābūt aprakstītam izmantojot Swagger, veidojot Swagger ieteicams metodes aizsargāt ar scope (skat. 5.4.3.2. paragrāfu). Tās publicēto servišu (API) metodes, kas nav aizsargātas ar scope ir pieejamas jebkuram Klientu lietojumam pēc šī serviša (API) abonēšanas Izstrādātāju portālā (Developer portal). Jebkurš Klientu lietojums var abonēt jebkuru no publicētajiem servišiem (API), bet serviša (API) publicētājam ir iespējams šo abonentu bloķēt (skat. 5.4.4. sadaļu), ja nav saskaņota tā izmantošana.

### 5.4.3. Servišu (API) reģistrēšana un publicēšana

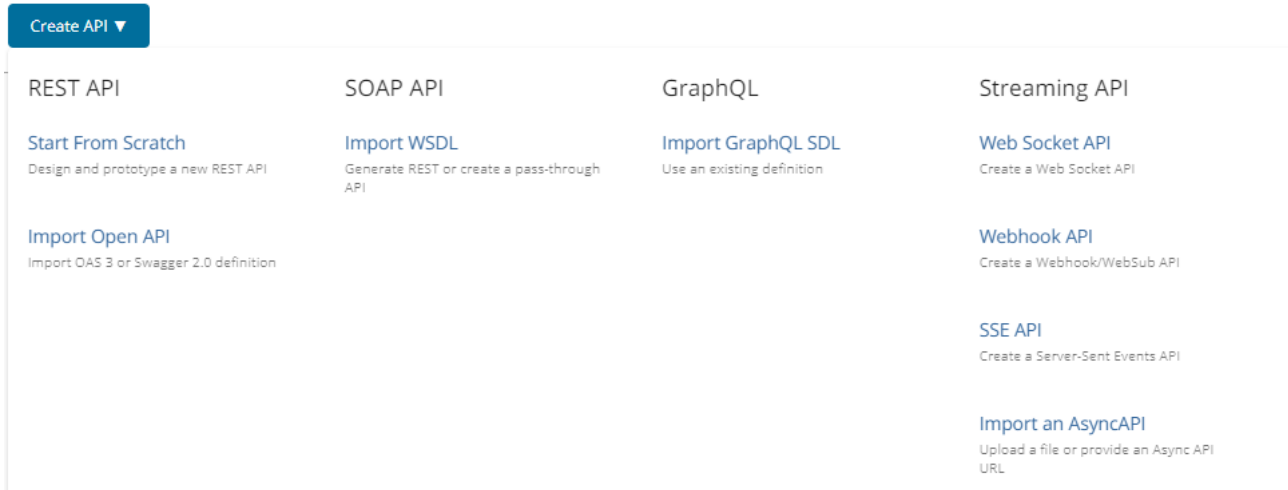
Lai reģistrētu API publicētājā (API Publisher) jaunu servišu (API), jāveic šādas darbības:

- 1) Jāautenticējas API publicētājā (API Publisher);



38.attēls. API Publicētāja saskarne

- 2) Jānospiež saite "Create API";
- 3) Jāizvēlas vienu no serviša (API) reģistrēšanas veidiem. Lai pievienotu REST servišu, tam jābūt aprakstītam ar swagger. Lai validētu swagger korektumu, jāizmanto <https://editor.swagger.io/>.



### 39.attēls. API reģistrēšanas veidi

- 4) Jāveic servisa (API) konfigurēšana atbilstoši API Pārvaldnieka dokumentācijai (REST [25], SOAP [26]) un zemāk esošiem paragrāfiem, tajā skaitā par nosaukumu veidošanu skat. 5.4.3.3 paragrāfu. (SOAP) vai skat. 5.4.3.2 paragrāfu (REST) šī soļa beigās serviss (API) jau ir publicēts datu ņēmējiem (Developer portal).
- 5) Nopublicētajiem servisiem (API) iespējams ierobežot rediģēšanas piekļuvi Publicētāju portālā iestādes ietvaros. Pēc jauna servisa reģistrācijas ir jāsaazinās ar VRAA un jāinformē, ka iestāde (norādot iestādi) ir reģistrējusi jaunu servisu (API) un nepieciešams ierobežot pieeju. Vai arī publicējot servisu (API) Basic Info sadaļā, laukā Publisher Access Control norādīt "Restrict by role(s)" un laukā Roles norādīt lomu, kuru VRAA izveido priekš iestādes un nosūta kopā ar piekļuves tiesībām e-pastā.

***Datu devējs nedrīkst ierobežot servisa pieejamību Izstrādātāju portālā (Developer portal), īpašībai "Visibility on Developer portal" vienmēr jābūt ar vērtību "Public".***

#### 5.4.3.1. Aizsargāts API

Pēc noklusējuma pievienotam servisam (API) tiek padots ienakošais OAuth2 talons, bet ja pievienojamais serviss (API) papildus ir aizsargāts ar cita identity servisa talonu vai nepieciešams cits OAuth2 talons (iestādes sistēmas pusē), tam jāpievieno "Request Message Mediation Policy" atbilstoši dokumentācijai [27]. Mediatora pievienošana notiek servisa detalizētā skata "Runtime" sadaļā (40.attēls). Tiek nodrošināts mediators, kas veic jauna talona pieprasīšanu un nodošanu uz aizsargāto servisu (API). Ja tiek izmantots mediators uz servisu (API), netiks nodots sākotnējais OAuth2 talons, ar kuru tiek izsaukts API Pārvaldniekā publicēts serviss (API), kas nozīmē, ka nevarēs iegūt datus par sākotnējo datu pieprasītāju. Jaunu mediatoru reģistrāciju API Pārvaldniekā veic VRAA administratori pēc pieprasījuma. Mediatoru iespējams veidot izmantošanai vienā vai vairākos servisos (API).



#### 40.attēls. Oauth2 mediatora pielikšana

### 5.4.3.2. REST servisu reģistrācija

Reģistrējot REST servisu, jāņem vērā tas, ka servisa (API):

- Pilnais servisa nosaukums veidojas noteiktā formāta notācijā – “**API**-{Autoritātes identifikators (visscore:authorityIdent) no iestāžu un struktūrvienību klasifikatora (Authority), izmanto apakšvītru “\_” punktu vietā, piemēram, “AS\_RIX”}-**{servisa nosaukums, piemēram, “CalculationServiceSync”}**”, piemēram, “**API-AS\_RIX-CalculationServiceSync**”;
- Izveides formā jāievada šādi dati:
  - Iestādes identifikatora laukā “Authority” jāievada iestādes identifikācijas kods, šajā jautājumā ievadāmā vērtība ir reģistrējama un jāievada tāda kāda tā ir norādīta Klasifikatoru katalogā, piemēram: “**AS.RIX**”;

**iestādes identifikācijas kods ir iegūstams “Iestādes un struktūrvienības” “Klasifikatoru katalogā (skat. 41.attēls) kā īpašības visscore:authorityIdent vērtība.**

Jēdziena "100002" īpašības

Izmantojiet šo lapu jēdziena pārvaldībai

Īpašība	Vērtība	Lokalizācija	Īpašības semantiskais tips
Velamais nosaukums	A/S "RIX Technologies"	Latviešu	skos:prefLabel
Uri	http://www.rixtechnologies.lv	Latviešu	visscore:uri
Burtu kods	RIX	Latviešu	vissorg:code
Īsais nosaukums	RIX Technologies	Latviešu	vissis:shortname
Iestādes reģistrācijas kods	40003548427	Latviešu	visscore:RegCode
Autoritātes identifikators	AS.RIX	Latviešu	visscore:authorityIdent

#### 41.attēls. Autoritātes identifikators iestādes un struktūrvienības klasifikatorā

- Nosaukuma laukā “Name” jāievada pakalpes nosaukums, tajā atļauts izmantot burtus, ciparus un punktu, piemēram, “**CalculationServiceSync**”;
- Versijas laukā “Version” vērtībai jābūt norādītai noteiktā formāta notācijā – “**v{Major}\_{Minor}**”, piemēram, “**v1\_0**”;

Ja REST servisu paredzēts izmantot e-pakalpojumos, tad tam ir jābūt abonētiem (*subscribed*) no viena kopīgā jauno e-pakalpojumu klienta lietojuma, kuru izmanto LvpContext.Request (saucot WSO2 reģistrēto servisu, e-pakalpojumu platforma apmaina e-pakalpojuma lietotāja IDS access\_token pret PFAS talonu pēc ActAs principa. PFAS talonā atļaujas (scopes) tiek ņemtas no

IDS access\_token scp pielaides (claim), t.i., e-pakalpojumam nav nepieciešams (un nevar) pieprasīt šīs scp atļaujas kā IDS scope (piem., profile, openid, context\_api). IDS talona scp pielaidē tiek nodotas visas vērtības, kas ir piesaistītas IDS uzticamajai pusei PFAS lietotnē kā atļaujas.

### 5.4.3.3. Integrācijas Servisu reģistrācija

SOAP servisi (IS) jāreģistrē kā SOAP API, izmantojot WSDL. Reģistrējot SOAP IS servisu, jāņem vērā:

- Pilnais servisa nosaukums veidojas noteiktā formāta notācijā – “**ISS**-{Autoritātes identifikators (visscore:authorityIdent) no iestāžu un struktūrvienību klasifikatora (Authority), izmanto apakšvītru “\_” punktu vietā, piemēram, “SIA\_ABC”}-{servisa nosaukums, piemēram, “CalculationServiceSync”}”, piemēram, “**ISS-SIA\_ABC-CalculationServiceSync**”;

**Ja IS serviss ir reģistrēts arī IS servisu katalogā, nosaukumu var paņemt no šī servisa URN beigu daļas aizstājot punktus ar apakšvītrām “\_”. E-pakalpojuma konfigurācijā URN tiek norādīts ar punktiem tāda formātā kā reģistrēts servisu katalogā’.**

- Izveides formā jāievada šādi dati:
  - Iestādes identifikatora laukā “Authority” jāievada iestādes identifikācijas kods, šajā jautā ievadāmā vērtība ir reģistrjūtīga un jāievada tāda kāda tā ir norādīta Klasifikatoru katalogā, piemēram: “**SIA.ABC**”;
  - **Autoritātes identifikators ir iegūstams “Iestādes un struktūrvienības” ”klasifikatoru katalogā (skat. 41.attēls) kā īpašības visscore:authorityIdent vērtība;**
    - Nosaukuma laukā “Name” jāievada pakalpes nosaukums, tajā atļauts izmantot burtus, ciparus un punktu, piemēram, “**CalculationServiceSync**”;
    - Versijas laukā “Version” vērtībai jābūt norādītai noteiktā formāta notācijā – “**v{Major}\_{Minor}**”, piemēram, “**v1\_0**”;

Ja SOAP IS paredzēts izmantot vecajos e-pakalpojumos pārejas periodā, tad tam ir jābūt abonētam (*subscribed*) no viena kopīgā e-pakalpojumu klienta lietojuma “APP-VIDM\_VRAA-VISS.ApiManagement.ISServiceInterceptor-v1\_0” un šie servisi nedrīkst būt aizsargāti ar atļaujām (scope).

**Netiek rekomendēta ISServiceInterceptor izmantotošana, jo tā dod iespēju no jebkura klienta lietojuma veikt IS servisu izsaukumus veicot to abonēšanu.**

Ja SOAP IS paredzēts izmantot jaunajos e-pakalpojumos, tad tam ir jābūt abonētam (*subscribed*) no viena kopīgā jauno e-pakalpojumu klienta lietojuma, kuru izmanto LvpContext.Request (saucot WSO2 reģistrēto servisu, e-pakalpojumu platforma apmaina e-pakalpojuma lietotāja IDS access\_token pret PFAS talonu pēc ActAs principa. PFAS talonā atļaujas (scopes) tiek ņemtas no IDS access\_token scp pielaides (claim), nevis scope, t.i., e-pakalpojumam nav nepieciešams (un nevar) pieprasīt šīs scp atļaujas kā IDS scope (piem., profile, openid, context\_api). IDS talona scp pielaidē tiek nodotas visas vērtības, kas ir piesaistītas IDS uzticamajai pusei PFAS lietotnē kā atļaujas.

### 5.4.3.4. GraphQL servisu reģistrācija

Reģistrējot GraphQL servisu, jāņem vērā tas, ka servisa (API):

- Pilnais servisa nosaukums veidojas noteiktā formāta notācijā – “**GQL**-{Autoritātes identifikators (visscore:authorityIdent) no iestāžu un struktūrvienību klasifikatora (Authority), izmanto apakšvītru “\_” punktu vietā, piemēram, “AS\_RIX”}-{servisa nosaukums, piemēram, “CalculationServiceSync”}”, piemēram, “**GQL-AS\_RIX-CalculationServiceSync**”;
- Izveides formā jāievada šādi dati:

- Iestādes identifikatora laukā "Authority" jāievada iestādes identifikācijas kods, šajā jaukā ievadāmā vērtība ir reģistrjūtīga un jāievada tāda kāda tā ir norādīta Klasifikatoru katalogā, piemēram: "AS.RIX";

**Iestādes identifikācijas kods ir iegūstams "Iestādes un struktūrvienības" klasifikatoru katalogā (skat. 41.attēls) kā īpašības visscore:authorityIdent vērtība.**

- Nosaukuma laukā "Name" jāievada pakalpes nosaukums, tajā atļauts izmantot burtus, ciparus un punktu, piemēram, "CalculationServiceSync";
- Versijas laukā "Version" vērtībai jābūt norādītai noteiktā formāta notācijā – "v{Major}\_{Minor}", piemēram, "v1\_0";

Ja REST servisu paredzēts izmantot e-pakalpojumos, tad tam ir jābūt abonētiem (*subscribed*) no viena kopīgā jauno e-pakalpojumu klienta lietojuma, kuru izmanto LvpContext.Request (saucot WSO2 reģistrēto servisu, e-pakalpojumu platforma apmaina e-pakalpojuma lietotāja IDS access\_token pret PFAS talonu pēc ActAs principa. PFAS talonā atļaujas (scopes) tiek ņemtas no IDS access\_token scp pielaides (claim), t.i., e-pakalpojumam nav nepieciešams (un nevar) pieprasīt šīs scp atļaujas kā IDS scope (piem., profile, openid, context\_api). IDS talona scp pielaidē tiek nodotas visas vērtības, kas ir piesaistītas IDS uzticamajai pusei PFAS lietotnē kā atļaujas.

#### 5.4.3.5. Scope pievienošana

API Publicētājā (API Publisher) reģistrētajiem servisiem iespējams ierobežot pieeju izmantojot scopes. Reģistrējot SOAP servisu, tam iespējams norādīt tikai vienu kopīgu scope uz visu servisu, bet, reģistrējot REST servisu, katrai metodei ir iespējams norādīt savu scope.

**Ja nepieciešams izsaukt ar scopes aizsargātu servisu (API), pieprasot talonu (skat. 5.3.1.5. paragrāfu), ir jāpieprasa arī atbilstošie scope.**

Scope pievienošana servisiem (API) notiek manuāli (skatīt 5.4.3.5.1. punktu) vai automatiski (skatīt 5.4.3.5.2. punktu), papildinot importējamo swagger datni (tikai REST gadījumā). Tiek rekomendēts, izstrādājot REST servisu, scope iekļaut swagger datnē, lai atvieglotu servisu reģistrēšanu API publicētājā (API Publisher).

Viedojot scope, to nosaukums jāveido atbilstoši noteiktai formāta notācijai, lai pēc šī nosaukuma būtu iespējams noteikt, kura datu devējam servisa (API) versijai tas pieder:

- SOAP gadījumā "ISS-{autoritātes identifikators, kura publicē servisu (API), izmanto apakšsvītru, piemēram, "SIA\_lestade1"}-{servisa (API) nosaukums, piemēram, "VissIntegrācijasServiss"}-{servisa (API) versija, izmanto apakšsvītru, piemēram, "v1\_0"}-{scope nosaukums bez speciālajiem simboliem, piemēram, "Manage"}". Piemērs: "ISS-SIA\_lestade1-VissIntegrācijasServiss-v1\_0-Manage";
- REST gadījumā: "API-{autoritātes identifikators, kura publicē servisu (API), izmanto apakšsvītru, piemēram, "SIA\_lestade1"}-{servisa (API) nosaukums, piemēram, "VissRestApi"}-{servisa (API) versija, izmanto apakšsvītru, piemēram, "v1\_0"}-{scope nosaukums bez speciālajiem simboliem, piemēram, "Manage"}". Piemērs: "API-SIA\_lestade1-VissRestApi-v1\_0-Manage".

**autoritātes identifikatorā un servisa (API) versijā tiek izmantota apakšsvītra, nevis punkts, jo WSO2 neļauj izmantot punktu scope nosaukumos.**

**Autoritātes identifikators ir iegūstams "Iestādes un struktūrvienības" klasifikatoru katalogā (skat. 41.attēls) kā īpašības visscore:authorityIdent vērtība.**

##### 5.4.3.5.1. Manuāli

Lai norādītu scopes manuāli ir jāveic šādas darbības:

Dokumenta kods: VRAA-13_7_17_41-VISS_2016-D_APM-VDL	Datums: 06.10.2023.	Versija: 2.23
Datne: VISS_2016.VDL.D_APM_v2.23	Izstrādāja: J.Kornijenko	Lpp.: 59 (99)

1. Jāatver API publicētājā (API Publisher) vēlamā servisa (API) rediģēšanas skata (skatīt, 42.attēlu) cilni Local Scopes.

The screenshot shows the API Publisher interface for the API 'API-SIA\_ABC-SwaggerPetstore :v1\_5'. The left sidebar contains a navigation menu with 'Local Scopes' highlighted. The main content area shows the 'Local Scopes' management page with a table listing existing scopes.

Name	Description	Roles
read:pets	read your pets	

#### 42.attēls. Servisa rediģēšanas skata atvēršana

2. Cilnē Local Scopes vispirms ir jādefinē visas servisā izmantotās scopes, nospiežot pogu "Add New Local Scope", skatīt 44.attēlu.
3. Javeic scope pievienošana, norādot tās identifikatoru un nosaukumu, skatīt 43.attēlu. Nosaukuma laukā jāievada tikai pašas atļaujas (scope) nosaukums, servisa nosaukums un versija tika pievienota automātiski, lai tas atbilstu scope formāta notācijai (skat. 5.4.3.2. paragrāfu):

Scopes > Create New Scope

Name  
CreatePet  
Enter Scope Name ( E.g.: creator )

Display Name  
CreatePet  
Enter Scope Name ( E.g.: creator )

Description  
Reģistrēt mājdzīvnieku  
Short description about the scope

Roles  
Enter roles and press Enter  
Enter a valid role and press "Enter".

Save Cancel

#### 43.attēls. Scope datu aizpildīšana

- Jāveic servisa (API) metožu sasaiste ar iepriekšējos soļos izveidotajiem scope. Cilnē Resources jāizvēlas metode kuras pieeju nepieciešams ierobežot un laukā "Operation scope" izvēlas kādu no pieejamajām scope (44.attēls).

The screenshot shows the API Gateway console interface. On the left, the 'Resources' tab is selected and highlighted with a red box and a red circle with the number 1. The main area displays the configuration for the /pet endpoint. The endpoint is highlighted with a red box and a red circle with the number 2. Below the endpoint, the 'API Scopes' section is visible, showing a list of scopes. The scope 'API-SIA\_ABC-SwaggerPetstore-v1\_5-CreatePet' is selected and highlighted with a red box and a red circle with the number 3. The 'Rate limiting policy' is set to 'Unlimited'. The 'Summary & Description' section is also visible, with a description field containing 'Add a new pet to the store'.

- Jāsaglabā veiktās izmaiņas nospiežot pogu Save lapas lejasdaļā.

#### 44.attēls. Scope pievienošana

##### 5.4.3.5.2. Automātiski

Reģistrējot jaunu REST servisu (API), izmantojot swagger datni, to var papildināt ar sekciju "x-wso2-security", kurā tiek aprakstītās visas servisā izmantotās scopes, un papildināt katras metodes (paths) konfigurāciju, norādot elementā "x-scope" kādu no servisā izmantotajiem scope, kas būs nepieciešams, lai izsauktu metodi.

```
{
```

```
"swagger": "2.0",
"info": {
  "version": "v1_0",
  "title": "API-VIDM_VRAA-Calculatation"
},
"host": "viss.gov.lv",
"basePath": "/VISS.ApiManagement/Extensions/CalculatationApi/stable",
"schemes": [
  "https"
],
"paths": {
  "/api/Calculator/Multiply": {
    "post": {
      ...
      "x-scope": "API-VIDM_VRAA-Calculatation-v1_0-Divide"
    }
  },
},
"definitions": {
  ...
},
"x-wso2-security": {
  "apim": {
    "x-wso2-scopes": [
      {
        "name": "Divide",
        "description": "Dalīt skaitļus",
        "key": "API-VIDM_VRAA-Calculatation-v1_0-Divide",
        "roles": ""
      }
    ]
  }
}
```

#### 5.4.4. Pieejas servisam (API) piešķiršana

Ja serviss (API) ir aizsargāts ar scopes, tad pēc tam, kad ir saņemts pieprasījums no Datu ņēmēja (atbilstoši VISS programsaskarnes (API) abonēšanas procedūrai), servisa (API) publicētājam (datu devējām) ir jāpiešķir tiesības (scopes) atbilstoši to formāta notācijai (skat. 5.4.3.2. paragrāfu) datu ņēmēja Klienta lietojumam (PFAS vai IDS uzticamajai pusei) PFAS saskarnē, izmantojot VRAA nodrošināto PFAS lietotāju (skat. 5.4. nodaļu).

**Datu devējs PFAS uzticamajai pusei var pievienot tikai tādus scope, kuru nosaukums satur datu devēja pārstāvētās autoritātes identifikatoru un kuri attiecīgajā brīdī eksistē API Pārvaldniekā (tiek veikta validācija).**

**E-pakļojumu ietvara mikroservisam, kas nodrošina API pārvaldniekā reģistrēto servisu (API) izsaukumus nav nepieciešams piešķirt atļaujas (scopes).**

### 5.4.5. Pieejas servisam (API) apturēšana

Datu devējam atkarībā no tā, vai serviss (API) ir vai nav aizsargāts, ir vairāki veidi, kā ierobežot piekļuvi servisam (API):

- Gan aizsargāta, gan neaizsargāta Servisa (API) publicētājam ir iespējams bloķēt abonentu (datu ņēmēju) [24], ja abonentam šī servisa izmantošana nav atļauta.

**Aizliegts bloķēt šo klienta lietojumu abonementus: APP-VIDM\_VRAA-EservicePlatformRequestApi-v1\_0, APP-VIDM\_VRAA-VISS.ApiManagement.ISServiceInterceptor-v1\_0**

- Ja serviss (API) vai metode ir aizsargāta, tad servisa (API) publicētājs var:
  - PFAS saskarnē, izmantojot VRAA nodrošināto PFAS lietotāju (skat. 5.4. nodaļu), noņemt pilnībā vai daļēji piesaistītos scopes Klienta lietojuma PFAS un IDS uzticamajai pusei. (ietekmē konkrēto servisa (API) izmantotāju (datu ņēmēju));

**Datu devējs PFAS un IDS uzticamajām pusēm var noņemt tikai tādus scope, kuru nosaukums satur datu devēja pārstāvētās autoritātes identifikatoru.**

- Datu devēju (API Publisher) portālā mainīt servisam (API) vai metodei piesaistītos scopes. (ietekmē visus servisa (API) izmantotājus (datu ņēmējus)).

**Datu devējs nedrīkst ierobežot servisa pieejamību Izstrādātāju portālā (Developer portal), īpašībai "Visibility on Developer portal" vienmēr jābūt ar vērtību "Public".**

### 5.4.6. Transakcijas obligātuma konfigurēšana

Datu devējam atkarībā no biznesa prasībām vai konkrētā servisa darbības nodrošināšanai nepieciešamās informācijas apjoma ir iespējams katram tā pārvaldītajam servisam individuāli norādīt vai transakcijas identifikators ir obligāts parametrs tā servisa izsaukšanai:

1. Publicētāju portālā jāatver izvēlētā servisa cilne Properties un ja transactionRequired īpašība vēl nav pievienota, tā jāpievieno. Īpašībai var norādīt vienu no divām vērtībām:
  - a. false – transakcija nav obligāta, ja tā tiks norādīta tiks veikta tās vērtības validācija;
  - b. true – transakcija ir obligāta;
2. Veidojot īpašību jānorāda pazīme "Show in devportal", lai datu ņēmēji būtu informēti par šo ierobežojumu.

The screenshot shows the 'API Properties' configuration page for an API named 'API-SIA,ABC-CalculationScopesTestV3 :v3.0'. The page has a sidebar with navigation options like 'Overview', 'Develop', 'Portal Configurations', 'Basic Info', 'Business Info', 'Subscriptions', 'Documents', 'Comments', 'API Configurations', 'Runtime', 'Resources', 'API Definition', 'Endpoints', 'Local Scopes', 'Properties', and 'Monetization'. The 'Properties' section is active, showing a table with columns for 'Property Name', 'Property Value', and 'Visibility'. A new property 'transactionRequired' is being added with a value of 'false' and the 'Show in devportal' checkbox checked. The 'Add' button is highlighted with a red box, and the 'Save' button is also highlighted with a red box.

**Pēc noklusējuma transakcija ir obligāta un tipiskā scenārijā šo parametru nav nepieciešams norādīt. Ja šī parametra vērtība tiek mainīta par to iepriekš ir jābrīdina datu ņēmēji īpaši gadījumā ja tā no neobligātas tiek nomainīta uz obligātu.**





## 5.5. Transakciju API izmantošana

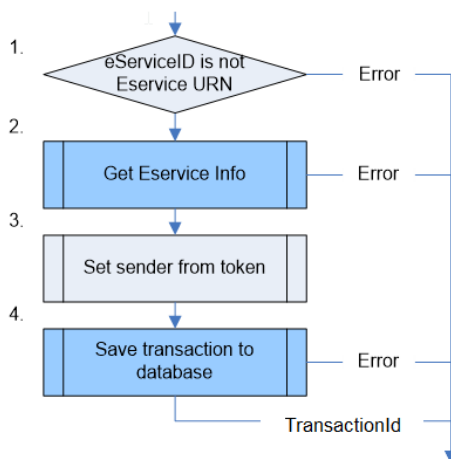
Transakciju API ir REST serviss, un tas ir paredzēts, lai aizvietotu esošā pieprasījumu servisa metodes StartTransaction un EndTransaction. Transakciju API ir pieejamas anonīmiem un ar PFAS STS OAuth2 references vai JWT talonu autentificētiem lietotājiem.

***Ja transakciju ir nepieciešams uzsākt e-pakalpojuma ietvaros, tad nepieciešams izmantot LvpContext.Request funkcionalitāti skat. e-pakalpojuma izveides ceļveža programmētāja rokasgrāmatu [28]***

### 5.5.1. Transakcijas uzsākšanas metode

Metode tiek izsaukta, kad nepieciešams uzsākt konkrētu datu apmaiņas procesa biznesa instanci. Parametrs eServiceId ir datu apmaiņas procesa identifikators. Lai iegūtu datu apmaiņas identifikatoru (URN) ir jāsaņem VRAA un tā piešķirs konkrētai datu apmaiņai paredzētu identifikatoru. Metode atgriež datu apmaiņas instances (transakcijas) URN identifikatoru (skat. 11.attēlu):

1. Tiek pārbaudīts, vai padotais „eServiceId” atbilst IVIS URN formātam;
2. Tiek iegūta informācija no e-pakalpojumu kataloga un e-formu kataloga;
3. Tiek apstrādāts talons un aizpildīti “sender” dati;
4. Tiek izveidots transakcijas identifikators un saglabāts datu bāzē.



#### 45.attēls. Transakcijas uzsākšana

Biznesa transakcija parasti nepieciešama komunikācijas identifikācijas gadījumā. Iespējamo e-pakalpojumu URN tipu uzskaitījums biznesa transakcijas uzsākšanai:

- EP – e-pakalpojums latvija.lv portālā (piemēram, URN:IVIS:100001:EP-EP00-V1-2);
- EP.VISS – e-pakalpojums VISS portālā (piemēram, URN:IVIS:100001:EP.VISS-ErrorReport-V1-2);
- EP.DA – sistēma–sistēma datu sniegšanas pakalpojums (piemēram, URN:IVIS:100001:EP.DA-DA29-V1-0);
- EF – e-forma (piemēram, URN:IVIS:100001:EF-Form01-V1-0).

***Sistēma–sistēma datu sniegšanas pakalpojuma gadījumā (EP.DA) papildus tiek pārbaudīta autentificētā lietotāja un izmantotā EP.DA URN sasaiste – tie piederīgi vienai iestādei, kā papildus sistēmu pieprasījumu autorizācijas līdzeklis.***

### 5.5.1.1. Piemēri ar datu nodošanu body daļā

#### Adrese

POST /transactions

#### HEADER parametri

- Authorization - *PFAS STS Bearer OAuth2 (JWT vai references)* talons (autenticētiem izsaukumiem) vai nenorādīts (anonīms izsaukums).
- Content-Type - pieprasījuma body formāts, serviss nodrošina *application/json* vai *application/xml*.
- Accept - vēlamais atbildes formāts, serviss nodrošina *application/json*, *application/xml* un *text/plain* formātus.
- Accept-Language - vēlamā atbildes valoda, noklusētā vērtība *en*, serviss nodrošina *lv*, *en* un *ru* valodas.

#### PATH parametri

Nav.

#### QUERY parametri

Nav.

#### BODY parametri

- eServiceId - datu apmaiņas procesa vai e-paklpojuma identifikators (*URN*).

#### Piemēri

```
POST /transactions HTTP/1.1

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InNEWX..
Content-Type: application/json

{
  "eServiceId": "URN:IVIS:100001:EP-EP96-V1-0"
}
```

```
POST /transactions HTTP/1.1

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InNEWX..
Content-Type: application/xml
Accept: application/xml

<StartTransactionRequest>
  <eServiceId>URN:IVIS:100001:EP-EP96-V1-0</eServiceId>
</StartTransactionRequest>
```

#### IZVADDATI

Izveidotās datu apmaiņas procesa transakcijas identifikators (*URN*) ar statusu vai HTTP 400, ja transakcija netika izveidota (piemēram, nepareiza e-palpojuma identifikatora dēļ).

#### Kļūdas

- HTTP 400 - norādīts nederīgs datu apmaiņas identifikators;
- HTTP 403 - nav tiesību veikt operāciju.

#### Piemēri

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"result": "URN:IVIS:100001:EP-EP96-V1-0-TR-336"}
```

```
HTTP/1.1 200 OK
content-type: application/xml; charset=utf-8

<StartTransactionResponse      xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><Result>URN:IVIS:100001:EP-EP96-
V1-0-TR-341</Result></StartTransactionResponse>
```

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8

URN:IVIS:100001:EP-EP96-V1-0-TR-336
```

### 5.5.1.2. Piemēri ar datu nodošanu query daļā

#### Adrese

POST /transactions/

#### HEADER parametri

- Authorization - *PFAS STS Bearer OAuth2 (JWT vai references)* talons (autenticētiem izsaukumiem) vai nenorādīts (anonīms izsaukums).
- Accept – vēlamais atbildes formāts, serviss nodrošina application/json, application/xml un text/plain formātus.
- Accept-Language – vēlamā atbildes valoda, noklusētā vērtība en, serviss nodrošina lv, en un ru valodas.

#### PATH parametri

Nav.

#### QUERY parametri

- eServiceId - datu apmaiņas procesa vai e-paklpojuma identifikators (*URN*).
- api-version – servisa versija, lai nodrošinātu datu apstrādi no query daļas jānorāda 1.2 versija.

#### BODY parametri

Nav.

#### Piemēri

```
POST      /transactions?api-version=1.2&eServiceId=URN:IVIS:100001:EP-EP96-V1-0
HTTP/1.1

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InNEWX..
Content-Type: application/json
```

```
POST /transactions?api-version=1.2&eServiceId=URN:IVIS:100001:EP-EP96-V1-0
HTTP/1.1

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InNEWX..
Accept: application/xml
```

## IZVADDATI

Izveidotās datu apmaiņas procesa transakcijas identifikators (*URM*) ar statusu vai HTTP 400, ja transakcija netika izveidota (piemēram, nepareiza e-palpojuma identifikatora dēļ).

### Kļūdas

- HTTP 400 - norādīts nederīgs datu apmaiņas identifikators;
- HTTP 403 - nav tiesību veikt operāciju.

### Piemēri

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"result": "URN:IVIS:100001:EP-EP96-V1-0-TR-336"}
```

```
HTTP/1.1 200 OK
content-type: application/xml; charset=utf-8

<StartTransactionResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><Result>URN:IVIS:100001:EP-EP96-
V1-0-TR-341</Result></StartTransactionResponse>
```

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8

URN:IVIS:100001:EP-EP96-V1-0-TR-336
```

## 5.5.2. Transakcijas apturēšanas metode

Metode tiek izsaukta, kad nepieciešams pabeigt konkrētu datu apmaiņas procesa instanci.

**Transakcija tiek izbeigta automātiski dienu pēc tās uzsākšanas.**

### 5.5.2.1. Piemēri ar datu nodošanu body daļā

#### Adrese

DELETE /transactions

#### HEADER parametri

- Authorization - *PFAS STS Bearer OAuth2 (JWT vai references)* talons (autenticētiem izsaukumiem) vai nenorādīts (anonīms izsaukums).
- Accept – vēlamais atbildes formāts, serviss nodrošina application/json, application/xml un text/plain formātus.

- Accept-Language – vēlamā atbildes valoda, noklusētā vērtība en, serviss nodrošina lv, en un ru valodas.
- Content-Type – pieprasījuma body formāts, serviss nodrošina application/json vai application/xml.

### PATH parametri

Nav.

### QUERY parametri

Nav.

### BODY parametri

- transactionId - datu apmaiņas procesa vai e-paklpojuma transakcijas identifikators (URN), kuru ir nepieciešams apturēt.

### Piemēri

```
DELETE /transactions HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InNEWX..
Content-Type: application/json

{
  "transactionId": "URN:IVIS:100001:EP-EP01-v1-0-TR-43"
}
```

```
DELETE /transactions HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InNEWX..
Content-Type: application/xml
Accept: application/xml

<EndTransactionRequest>
  <TransactionId>URN:IVIS:100001:EP-EP01-v1-0-TR-43</TransactionId>
</EndTransactionRequest>
```

### IZVADDATI

Veiksmīgas datu apmaiņas transakcijas apturēšanas gadījumā tiek izvadīts 1. Ja transakcija jau ir apturēta vai norādīta neeksistējoša, tad tiek izvadīta 0.

### Kļūdas

- HTTP 400 - norādīts nederīgs transakcijas identifikators
- HTTP 403 - nav tiesību veikt operāciju.

### Piemēri

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"result":1}
```

```
HTTP/1.1 200 OK
content-type: application/xml; charset=utf-8

<EndTransactionResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><Result>1</Result></EndTransactio
nResponse>
```

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8

1
```

### 5.5.2.2. Piemēri ar datu nodošanu query daļā

#### Adrese

DELETE /transactions

#### HEADER parametri

- Authorization - *PFAS STS Bearer OAuth2 (JWT vai references)* talons (autenticētiem izsaukumiem) vai nenorādīts (anonīms izsaukums).
- Accept – vēlamais atbildes formāts, serviss nodrošina application/json, application/xml un text/plain formātus.
- Accept-Language – vēlamā atbildes valoda, noklusētā vērtība en, serviss nodrošina lv, en un ru valodas.

#### PATH parametri

Nav.

#### QUERY parametri

- transactionId - datu apmaiņas procesa vai e-paklpojuma transakcijas identifikators (URN), kuru ir nepieciešams apturēt.
- api-version – servisa versija, lai nodrošinātu datu apstradi no query daļas jānorāda 1.2 versija.

#### BODY parametri

Nav.

#### Piemēri

```
DELETE /transactions?api-version=1.2&transactionId=URN:IVIS:100001:EP-EP01-v1-
0-TR-61050 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InNEWX..
Accept: application/json
```

```
DELETE /transactions?api-version=1.2&transactionId=URN:IVIS:100001:EP-EP01-v1-
0-TR-61050 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InNEWX..
Accept: text/plain
```

```
DELETE /transactions?api-version=1.2&transactionId=URN:IVIS:100001:EP-EP01-v1-0-TR-61050 HTTP/1.1
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InNEWX..
Accept: application/xml
```

## IZVADDATI

Veiksmīgas datu apmaiņas transakcijas apturēšanas gadījumā tiek izvadīts 1. Ja transakcija jau ir apturēta vai norādīta neeksistējoša, tad tiek izvadīta 0.

### Kļūdas

- HTTP 400 - norādīts nederīgs transakcijas identifikators
- HTTP 403 - nav tiesību veikt operāciju.

### Piemēri

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"result":1}
```

```
HTTP/1.1 200 OK
content-type: application/xml; charset=utf-8

<EndTransactionResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><Result>1</Result></EndTransactionResponse>
```

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8

1
```

### **5.5.3. Transakcijas statusa metode**

Metode tiek izsaukta, kad nepieciešams parbaudīt transakcijas statusu.

#### Adrese

GET /transactions

#### HEADER parametri

- Authorization - *PFAS STS Bearer OAuth2 (JWT vai references)* talons (autentificētiem izsaukumiem) vai nenorādīts (anonīms izsaukums).
- x-transactionId - datu apmaiņas procesa transakcijas identifikators (URN), kuru ir nepieciešams apturēt.
- Accept – vēlamais atbildes formāts, serviss nodrošina application/json, application/xml un text/plain formātus.
- Accept-Language – vēlamā atbildes valoda, noklusētā vērtība lv, serviss nodrošina lv, en un ru valodas.

#### PATH parametri

Nav.

#### QUERY parametri

Nav.

## BODY parametri

Nav.

## Piemērs

```
GET /transactions HTTP/1.1

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6InNEWX..
x-transactionId: URN:IVIS:100001:EP-EP01-v1-0-TR-43
```

## IZVADDATI

Veiksmīgas datu apmaiņas transakcijas statusa izgušanas gadījumā tiek izvadīts transakcijas identifikators (*URN*) ar statusu.

Statusa varianti, ja atbilde tiek pieprasīta *application/json* formātā:

- *notstarted* – transakcijas nav uzsākta;
- *started* – transakcija ir uzsākta;
- *ended* – transakcija ir pabeigta.

Statusa varianti, ja atbilde tiek pieprasīta *application/xml* vai *text/plain* formātā:

- *TransactionNotStarted* – transakcijas nav uzsākta;
- *TransactionInWork* – transakcija ir uzsākta;
- *TransactionIsEnded* – transakcija ir pabeigta.

## Kļūdas

- HTTP 400 - norādīts nederīgs transakcijas identifikators
- HTTP 403 - nav tiesību veikt operāciju.

## Piemēri

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{
  "transactionId": "URN:IVIS:100001:EP-EP01-v1-0-TR-43",
  "status": "ended"
}
```

```
HTTP/1.1 200 OK
content-type: application/xml; charset=utf-8

<TransactionStatusResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"><TransactionId>
URN:IVIS:100001:EP-EP01-v1-0-TR-
43</TransactionId><Status>TransactionIsEnded</Status></TransactionStatusRespon
se>
```



```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8

TransactionIsEnded
```

## 5.6. API izsaukumu piemēri

Piemēri demonstrē iespēju iegūt PFAS OAuth2 drošības talonu un WSO2 publicēta API servisa izsaukumu. Piemēri ir vienkāršoti, lai uzskatāmi demonstrētu API pārvaldnieka un PFAS risinājumu funkcionalitāti, veicot programmatūras izstrādi, ņem vērā labās izstrādes prakses (neiešūt proles kodā, realizēt kļūdu apstrādi, veikt talona kešošanu u.c).

**Jauno e-pakalpojumu izstrādātājiem biznesa servisu izsaukšanai jāizmanto e-pakalpojumu platformas *LvpContext.Request* funkcionalitāte skat. e-pakalpojuma izveides ceļveža programmētāja rokasgrāmatu [28].**

### 5.6.1. Priekšnosacījumi

- 1) VRAA uzstāda atbilstošās vidēs CalculationApi REST servisu izmantojot `Viss.ApiManagement.Extensions.CalculationApi.Setup.msi`
- 2) VRAA reģistrē API publicētājā (API Publisher) CalculationApi (REST) un CalculationDataSync (esošais IS serviss URN:IVIS:100001:ISS-SIA.ABC-CalculationDataSync-v1-0) servisu saskaņā ar 5.4.3. sadaļas vadlīnijām;
- 3) Reģistrēt jaunu Klienta lietojumu Izstrādātāju portālā (Developer portal) saskaņā ar 5.3.1.3. paragrāfa vadlīnijām;
- 4) Izveidot abonementu starp reģistrēto Klienta lietojumu un servisiem CalculationApi un CalculationDataSync saskaņā ar 5.3.1.4. paragrāfa vadlīnijām;
- 5) Izveidot PFAS AUTH lietotāju saskaņā ar 5.3.1.1. paragrāfa vadlīnijām.

### 5.6.2. Izsaukumu Curl piemēri

Pieprasot talonu ir iespējams nodot CLIENTID un SECRET vērtības vairākos veidos:

Izmantot OAuth2 specifikācijā RFC6749 noteikto sintaksi – key var tikt norādīts ar vai bez prefiksa „urn:oauth2:”, bet katram no parametriem ir nepiešams veikt URL enkodēšanu.

Drošības talonu var pieprasīt no šādas adreses VRAA testa vidē:

- <https://ha.vraa.gov.lv/STS/VISS.Pfas.STS/oauth2/token>

Drošības talonu var pieprasīt no šādas adreses VRAA produkcijas vidē:

- <https://ha.viss.gov.lv/STS/VISS.Pfas.STS/oauth2/token>

Pieprasījuma piemērs:

```
curl -X POST -v https://ha.viss.gov.lv/STS/VISS.Pfas.STS/oauth2/token -u
"urn%3Aoauth2%3A892df848-ad6f-458a-b77b-
435d98e7dbc1:onTcdfDagb63f61euIo4obzjCJgScdkrlU6ZDVeKJAc%3D" -d
"grant_type=password" --data-urlencode "username=VRAATestUser" --data-urlencode
"password=TheBestPsw123"
```

- Izmantot RFC2617 specifikācija noteikto sintaksi – key jānorāda bez prefiksa „urn:oauth2:”.
- Vai nodot tos kā atribūtus *client\_id* un *client\_secret* iekš body – key var tikt norādīts ar vai bez prefiksa „urn:oauth2:”

```
curl -X POST -v https://ha.viss.gov.lv/STS/VISS.Pfas.STS/oauth2/token -d
"grant_type=password" --data-urlencode "username=VRAATestUser" --data-urlencode
"password=TheBestPsw123" --data-urlencode "client_id=urn:oauth2:892df848-ad6f-
```

```
458a-b77b-435d98e7dbc1" --data-urlencode  
"client_secret=onTcdfDagb63f61euIo4obzjCJgScdkrlU6ZDVeKJAc="
```

Dokumentā esošie piemēri veidoti pēc RFC2617 specifikācijas.

### 5.6.2.1. OAuth2 drošības talona pieprasījums bez scope aizpildīšanas

Piemērs OAuth2 drošības talona atsauces pieprasīšanai no PFAS.STS bez scope norādīšanas. Šāds drošības talons derīgs pieprasījumiem kuros izmantotie servisi nav aizsargāti ar scope.

```
curl -X POST -v <TOKEN_ENDPOINT> -u "<CLIENTID>:<SECRET>" -d  
"grant_type=password" --data-urlencode "username=<USERNAME>" --data-urlencode  
"password=<PASSWORD>"
```

Parametri:

- TOKEN\_ENDPOINT - STS OAuth2 drošības talona pieprasījuma adrese;
- CLIENTID - izveidotā Klienta lietojuma Consumer Key (bez urn:oauth2: prefiksa), skatīt 5.3.1.3;
- SECRET - izveidotā Klienta lietojuma Consumer Secret, skatīt 5.3.1.3;
- USERNAME - pieprasītā PFAS AUTH lietotāja nosaukums, skatīt 5.3.1.1;
- PASSWORD - pieprasītā PFAS AUTH lietotāja parole, skatīt 5.3.1.1;

Drošības talonu var pieprasīt no šādām adresēm VRAA testa vidē:

Drošības talonu var pieprasīt no šādas adreses VRAA testa vidē:

- <https://ha.vraa.gov.lv/STS/VISS.Pfas.STS/oauth2/token>

Drošības talonu var pieprasīt no šādas adreses VRAA produkcijas vidē:

- <https://ha.viss.gov.lv/STS/VISS.Pfas.STS/oauth2/token>

Pieprasījuma piemērs:

```
curl -X POST -v https://ha.viss.gov.lv/STS/VISS.Pfas.STS/oauth2/token-u  
"0f2053f6-95bf-401d-8e01-  
0f8a49537b43:BqNTJgY5dtmcKi7zoRBZl/hRXj0a3mSPvyrUtlx+cDM=" -d  
"grant_type=password" --data-urlencode "username=VRAATestUser" --data-urlencode  
"password=TheBestPsw123"
```

Atbildes piemērs:

```
{"access_token":"urn:uuid:ddf7a566-afff-4a58-9a30-2f0859165d5c","expires_in":360  
0,"token_type":"bearer"}
```

Atribūts access\_token satur drošības talona atsauci, kas būs jāizmanto servisu izsaukšanai. Detalizētu informāciju par OAuth2 drošības talonu skatīt [29].

### 5.6.2.2. OAuth2 drošības talona pieprasījums ar norādītiem scope

Piemērs OAuth2 references talona pieprasīšanai no PFAS.STS ar norādītu operāciju/scope. Pieprasījumā ir iespējams norādīt vairākus scope atdalot tos ar atstarpi. Atbildē tiks atgriezti tikai tie scope uz kuriem lietotājam būs piešķirtas tiesības PFAS.

```
curl -X POST -v <TOKEN_ENDPOINT> -u "<CLIENTID>:<SECRET>" -d  
"grant_type=password" --data-urlencode "username=<USERNAME>" --data-urlencode  
"password=<PASSWORD>" --data-urlencode "scope=SCOPE"
```

Parametri:

- TOKEN\_ENDPOINT - STS OAuth2 drošības talona pieprasījuma adrese;
- CLIENTID - izveidotā Klienta lietojuma Consumer Key (bez urn:oauth2: prefiksa), skatīt 5.3.1.3;

- SECRET - izveidotā Klienta lietojuma Consumer Secret, skatīt 5.3.1.3;
- USERNAME - pieprasītā PFAS AUTH lietotāja nosaukums, skatīt 5.3.1.1;
- PASSWORD - pieprasītā PFAS AUTH lietotāja parole, skatīt 5.3.1.1;
- SCOPE – Saraksts ar scope/operācijām, kas nepieciešamas, lai ar drošības talonu izsauktu vēlamos servisu.

Drošības talonu var pieprasīt no šādas adreses VRAA testa vidē:

- <https://ha.vraa.gov.lv/STS/VISS.Pfas.STS/oauth2/token>

Drošības talonu var pieprasīt no šādas adreses VRAA produkcijas vidē:

- <https://ha.viss.gov.lv/STS/VISS.Pfas.STS/oauth2/token>

Pieprasījuma piemērs:

```
curl -X POST -v https://ha.viss.gov.lv/STS/VISS.Pfas.STS/oauth2/token -u
"0f2053f6-95bf-401d-8e01-
0f8a49537b43:BqNTJgY5dtmcKi7zoRBZ1/hRXj0a3mSPvyrUtlx+cDM=" -d
"grant_type=password" --data-urlencode "username=VRAATestUser" --data-urlencode
"password= TheBestPsw123" --data-urlencode "scope=CreateNewObject"
```

Atbildes piemērs:

```
{"access_token":"urn:uuid:f5c74f49-fcbf-4f94-b8f6-892292e448c5","expires_in":359
9,"scp":"CreateNewObject","token_type":"bearer"}
```

Atribūts access\_token satur drošības talona atsauci, kas būs jāizmanto servisu izsaukšanai. Detalizētu informāciju par OAuth2 drošības talonu skatīt [29].

### 5.6.2.3. Datu izgūšana par drošības talonu

Ja nepieciešams biznesa servisā (API) izgūt datus par lietotāju, kas veic izsaukumu ar OAuth2.0 references piekļuves (access) talonu, to ir iespējams veikt, izsaucot PFAS STS ieejas punktu /oauth2/introspect.

Datus talonā par lietotāju var izmantot, lai papildus ierobežotu pieeju pie datiem, t.i., nodrošinātu, ka katrs lietotājs var izgūt tikai viņam piederošos datus. Ja datu atlase tiek veikta pēc lietotāja identifikatora, to rekomendējams lasīt no talona, nevis veidot atsevišķu ieejas parametru.

Par lietotāju, kas autentificējies, izmantojot IDS vai PFAS STS, un veic izsaukumu iespējams iegūt šādus datus atkarībā no tā kādai auditorijai talona pieprasītājs pieprasa talonu:

- Fiziskas personas access talonā pēc noklusējuma būs iekļautas šādas pielaišanas (claims) no IDS izsniegtā talona:
  - given\_name – vārds;
  - family\_name – uzvārds;
  - ppid – personas kods;
  - sub – lietotāja vienotais identifikators;
- Juridiskas personas access talonā pēc noklusējuma būs iekļautas šādas pielaišanas (claims) no IDS izsniegtā talona:
  - given\_name – vārds;
  - family\_name – uzvārds;
  - ppid – personas kods;
  - sub – lietotāja vienotais identifikators;
  - legalentity – uzņēmuma reģistrācijas numurs;
  - legalentityname – uzņēmuma pilnais nosaukums;
  - legalentityshortname – uzņēmuma īsais nosaukums;

- legalentityaddress – uzņēmuma adrese;
- legalentityposition – amats;
- legalentityrepresentation – pārstāvniecības veids;
- Pilnvarotās personas access talonā pēc noklusējuma būs iekļautas šādas pielaišanas (claims) no IDS izsniegtā talona:
  - given\_name – vārds;
  - family\_name – uzvārds;
  - ppid – pilnvaras saņēmēja personas kods;
  - sub – lietotāja vienotais identifikators;
  - grantor – pilnvaras devēja personas kods vai reģistrācijas numurs;
  - grantorname – pilnvaras devējs.
- API pārvaldniekā reģistrētajiem Klienta lietojumiem (application) pēc noklusējuma PFAS STS izsniegtajā access talonā būs pievienotas šādas pielaišanas (claims), ja talonu pieprasa ar iestādes vai iestādes darbinieka lietotāju:
  - given\_name – vārds (tikai iestādes darbiniekam);
  - family\_name – uzvārds (tikai iestādes darbiniekam);
  - sub – lietotāja vienotais identifikators;

Veicot uzticamās puses konfigurēšanu, ir iespējams pievienot vai noņemt pielaišanas. Ne visas pielaišanas ir obligātas un var būt neaizpildītas. Uzticamo pušu pielaižu konfigurēšanu veic VRAA administrators PFAS administratora saskarnē.

### Introspect izsaukums:

```
curl -X POST -v <INTROSPECT_ENDPOINT> -u "<CLIENTID>:<SECRET>" --data-urlencode "token=<TOKEN>"
```

Parametri:

- INTROSPECT\_ENDPOINT - STS OAuth2 drošības talona informācijas izgūšanas adrese;
- CLIENTID - izveidotā Klienta lietojuma Consumer Key (bez urn:oauth2: prefiksa), skatīt 5.3.1.3;
- SECRET - izveidotā Klienta lietojuma Consumer Secret, skatīt 5.3.1.3;
- TOKEN – drošības talona atsauce

Drošības talonu var pārbaudīt izsaucot šādu adresi VRAA testa vidē:

- <https://ha.vraa.gov.lv/STS/VISS.Pfas.STS/oauth2/introspect>;

Drošības talonu var pārbaudīt izsaucot šādu adresi VRAA produkcijas vidē:

- <https://ha.viss.gov.lv/STS/VISS.Pfas.STS/oauth2/introspect>;

Pieprasījuma piemērs:

```
curl -X POST -v https://ha.vraa.gov.lv/STS/VISS.Pfas.STS/oauth2/introspect -u "0f2053f6-95bf-401d-8e01-0f8a49537b43:BqNTJgY5dtmcKi7zoRBZ1/hRXj0a3mSPvyrUt1x+cDM=" --data-urlencode "token= urn:uuid:f5c74f49-fcbf-4f94-b8f6-892292e448c5"
```

Atbildes piemērs:

```
{
  "active": true,
  "aud": [
```

```
    "urn:oauth2:0f2053f6-95bf-401d-8e01-0f8a49537b43"
  ],
  "exp": 1571747794,
  "family_name": "Vārds",
  "given_name": "Uzvārds",
  "nbf": 1571733394,
  "scp": [
    "CreateNewObject"
  ],
  "sub": "AU:100001-PK:12345678901"
}
```

Detalizētu informāciju par OAuth2 drošības talonu skatīt [29].

#### 5.6.2.4. REST servisa izsaukums

Reizināšanas metodes izsaukuma piemērs, metode Multiply:

```
curl -H "Authorization: Bearer <TOKEN>" -H "Content-Type: application/json" -H
"x-transactionId: <TRANSACTION>" -X POST -d "{ \"Multiplier\": <MULTIPLIER>,
\"Multiplicand\": <MULTIPLICAND>}" "<ENDPOINT>/api/Calculator/Multiply"
```

Parametri:

- TOKEN - drošības talona atsauce;
- MULTIPLIER - reizinātājs;
- MULTIPLICAND - reizināmais;
- ENDPOINT – servisa CalculationApi adrese no Izstrādātāju portāla (Developer portal) vērtība no Production and Sandbox Endpoints lauka;
- TRANSACTION – transakcijas identifikators, lai iegūtu skatīt 5.5.1.

Piemērs:

```
curl -H "Authorization: Bearer urn:uuid:lafdf241-b671-4ed4-b0e2-c047eb39da37" -H
"Content-Type: application/json" -H "x-transactionId: URN:IVIS:100001:EP-EP186-
v1-0-TR-1584" -X POST -d "{ \"Multiplier\": 12, \"Multiplicand\": 12}"
"http://192.168.102.36:8281/VISS.ApiManagement/Extensions/CalculationApi/stable/
v1/api/Calculator/Multiply"
```

Dalīšanas metodes izsaukuma piemērs, metode Divide:

```
curl -H "Authorization: Bearer <TOKEN>" -H "Content-Type: application/json" "x-
transactionId: <TRANSACTION>" -X POST -d "{ \"Dividend\": <DIVIDEND>,
\"Divisor\": <DIVISOR>}" "<ENDPOINT>/api/Calculator/Divide"
```

Parametri:

- TOKEN - drošības talona atsauce;
- DIVIDEND - dalāmais;
- DIVISOR - dalītājs;
- ENDPOINT - servisa CalculationApi adrese no Izstrādātāju portāla (Developer portal) vērtība no Production and Sandbox Endpoints lauka;
- TRANSACTION – transakcijas identifikators, lai iegūtu skatīt 5.5.1.

**Piemērs:**

```
curl -H "Authorization: Bearer urn:uuid:1afdf241-b671-4ed4-b0e2-c047eb39da37" -H
"Content-Type: application/json" -H "x-transactionId: URN:IVIS:100001:EP-EP186-
v1-0-TR-1584" -X POST -d "{ \"Dividend\": 12, \"Divisor\": 12}"
"http://192.168.102.36:8281/VISS.ApiManagement/Extensions/CalculationApi/stable/
v1/api/Calculator/Divide"
```

**5.6.2.5. SOAP servisa izsaukšana**

Reizināšanas metodes izsaukuma piemērs, action DefaultMethodSync:

```
curl -H "Authorization: Bearer <TOKEN>" -H "Content-Type: text/xml" -H "x-
transactionId: <TRANSACTION>" -H
"SOAPAction:http://ivis.eps.gov.lv/ISS/IVISService/v1-0/DefaultMethodSync" -X
POST -d @request.xml "<ENDPOINT>"
```

**Parametri:**

- TOKEN - drošības talona atsauce;
- ENDPOINT - servisa ISS-SIA.ABC-CalculationDataSync adrese no Izstrādātāju portāla (Developer portal) vērtība no Production and Sandbox Endpoints lauka.
- MULTIPLIER - reizinātājs;
- MULTIPLICAND - reizināmais;
- TRANSACTION – transakcijas identifikators, obligāti jānorada header atribūtā x-transactionId vai pieprasījuma struktūras <IVISRequest><Header><TransactionID> elementā, lai iegūtu skatīt 5.5.1.

**Pieprasījuma xml piemērs:**

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <DefaultMethodSync xmlns="http://ivis.eps.gov.lv/ISS/IVISService/v1-
0">
      <IVISRequest
xmlns="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0">
        <Header>
          <MessageID>6cafe2c6-9834-47ad-b710-
be8033710b02</MessageID>
          <MessageType>URN:IVIS:100001:XSD-Testing-
TestISServise-v1-0-TYPE-Calculation</MessageType>
          <TransactionID>URN:IVIS:100001:EP-EdkTester-v1-0-
TR-1393</TransactionID>
          <CorrelationID>f4ceae8a-dfd9-48b3-a0b6-
5d990aaac86e</CorrelationID>
          <Sender>
            <SystemID>URN:IVIS:100001:PORTAL-
Unknown</SystemID>
          <Inhabitant>
            <PersonID
Scheme="urn:ivis:100001:name.id-viss">PK:12345612345</PersonID>
            <PersonCode>12345612345</PersonCode>
            <FullName>
```

```

        <LastName
xmlns="http://ivis.eps.gov.lv/XMLSchemas/100001/Person/v1-0">Uzvārds</LastName>
        <FirstName
xmlns="http://ivis.eps.gov.lv/XMLSchemas/100001/Person/v1-0">Vārds</FirstName>
        </FullName>
    </Inhabitant>
</Sender>
    <Destination>URN:IVIS:100001:ISS-SIA.ABC-
CalculationDataSync-v1-0</Destination>
    <MilestoneID>URN:IVIS:100001:EP-EdkTester-v1-0-MS-
CallCalcSync</MilestoneID>
</Header>
    <Body>
        <Calculation
xmlns="http://ivis.eps.gov.lv/XMLSchemas/100000/TestISServise/v1-0"
xmlns:ivis="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0"
xmlns:pers="http://ivis.eps.gov.lv/XMLSchemas/100001/Person/v1-0">
            <Number1><MULTIPLIER></Number1>
            <Number2><MULTIPLICAND></Number2>
            <Operation>multiplication</Operation>
        </Calculation>
    </Body>
</IVISRequest>
</DefaultMethodSync>
</s:Body>
</s:Envelope>

```

#### Dalīšanas metodes izsaukuma piemērs:

```

curl -H "Authorization: Bearer <TOKEN>" -H "Content-Type: text/xml" -H "x-
transactionId: <TRANSACTION>" -H
"SOAPAction:http://ivis.eps.gov.lv/ISS/IVISService/v1-0/DefaultMethodSync" -X
POST -d @request.xml "<ENDPOINT>"

```

#### Parametri:

- TOKEN - drošības talona atsauce;
- ENDPOINT - servisa ISS-SIA.ABC-CalculationDataSync adrese no Izstrādātāju portāla (Developer portal) vērtība no Production and Sandbox Endpoints lauka.
- DIVIDEND - dalāmais;
- DIVISOR - dalītājs;
- TRANSACTION – transakcijas identifikators, obligāti jānorada header atribūtā x-transactionId vai pieprasījuma struktūras <IVISRequest><Header><TransactionID> elementā, lai iegūtu skatīt 5.5.1.

#### Pieprasījuma xml piemērs:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <DefaultMethodSync xmlns="http://ivis.eps.gov.lv/ISS/IVISService/v1-
0">

```

```
<IVISRequest
xmlns="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0">
  <Header>
    <MessageID>6cafe2c6-9834-47ad-b710-
be8033710b02</MessageID>
    <MessageType>URN:IVIS:100001:XSD-Testing-
TestISServise-v1-0-TYPE-Calculation</MessageType>
    <TransactionID>URN:IVIS:100001:EP-EdkTester-v1-0-
TR-1393</TransactionID>
    <CorrelationID>f4ceae8a-dfd9-48b3-a0b6-
5d990aaac86e</CorrelationID>
    <Sender>
      <SystemID>URN:IVIS:100001:PORTAL-
Unknown</SystemID>
      <Inhabitant>
        <PersonID
Scheme="urn:ivis:100001:name.id-viss">PK:12345612345</PersonID>
        <PersonCode>12345612345</PersonCode>
        <FullName>
          <LastName
xmlns="http://ivis.eps.gov.lv/XMLSchemas/100001/Person/v1-0">Uzvārds</LastName>
          <FirstName
xmlns="http://ivis.eps.gov.lv/XMLSchemas/100001/Person/v1-0">Vārds</FirstName>
          </FullName>
        </Inhabitant>
      </Sender>
      <Destination>URN:IVIS:100001:ISS-SIA.ABC-
CalculationDataSync-v1-0</Destination>
      <MilestoneID>URN:IVIS:100001:EP-EdkTester-v1-0-MS-
CallCalcSync</MilestoneID>
    </Header>
    <Body>
      <Calculation
xmlns="http://ivis.eps.gov.lv/XMLSchemas/100000/TestISServise/v1-0"
xmlns:ivis="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0"
xmlns:pers="http://ivis.eps.gov.lv/XMLSchemas/100001/Person/v1-0">
        <Number1><DIVIDEND></Number1>
        <Number2><DIVISOR></Number2>
        <Operation>division</Operation>
      </Calculation>
    </Body>
  </IVISRequest>
</DefaultMethodSync>
</s:Body>
</s:Envelope>
```



### 5.6.3. .NET piemēri

#### 5.6.3.1. SOAP un REST servisu izsaukšana un talona pieprasīšana ar sertifikātu.

Ja no backend nepieciešams izsaukt API pārvaldniekā reģistrētus servisu, var izmantot PFAS STS sertifikāta autentifikāciju, lai iegūtu JWT drošības talonu.

.NET Core v2.2.110 piemērā `Viss.ApiManagement.Samples` izmantotie parametri jānorāda `RestExample.cs` un `SoapExample.cs` datnēs:

- `baseAddress` – pievienoto servisu (API) adrese no Izstrādātāju portāla (Developer portal) APIS (Production and Sandbox Endpoints);
- `identityAddress` – PFAS STS vai API pārvaldnieka OAuth2 drošības talonu servisa adrese;
- `introspectionAddress` – PFAS STS OAuth2 talonu introspekcijas adrese;
- `jwtPayloadAudience` – PFAS OAuth2 drošības talonu servisa adrese JWT talona pieprasīšanai ar sertifikātu;
- `clientId` - izveidotā Klienta lietojuma Consumer Key (bez `urn:oauth2:` prefiksa), skatīt 5.3.1.3;
- `clientSecret` - izveidotā Klienta lietojuma Consumer Secret, skatīt 5.3.1.3;
- `pathnameprivkeyfile` – pilns ceļš līdz sertifikātam, tam jābūt derīgam un reģistrētam PFAS kā autentifikācijas sertifikātam, un ja nepieciešams sertifikātam jābūt piešķirtām nepieciešamajām operācijām.
- `passwordprivkey` – Sertifikāta parole.
- `scope` – atļauja ja tiek izsaukts aizsargāts serviss API;
- `transactionApiAddress` – Transakcijas API adrese;
- `eServiceId` – datu apmaiņas vai e-paklpojuma identifikators.

Piemēri satur talona izgūšanu, REST un SOAP servisu izsaukumus. Lai piemēru darbinātu izmantojot sertifikāta autentifikāciju `AuthenticationHandler.cs` un `SoapExample.cs` datnēs jāizmanto `GetAccessTokenClientCredentials`:

```

AuthenticationHandler.cs
protected override async Task<HttpResponseMessage> SendAsync(HttpRequestMessage request, Cancellat
{
    HttpResponseMessage response = null;
    async Task retry()
    {
        var tokenResponse = await identityServerClient.GetAccessTokenClientCredentials();
        //var tokenResponse = await identityServerClient.GetAccessTokenPassword(); //or await iden
        //tokenResponse = null;
    }
}

SoapExamples.cs
private async Task Multiply()
{
    TokenResponse tokenResponse = await identityServerClient.GetAccessTokenClientCredentials();
    //TokenResponse tokenResponse = await identityServerClient.GetAccessTokenPassword(); //or awai
}

private async Task Divide()
{
    //TokenResponse tokenResponse = await identityServerClient.GetAccessTokenPassword();
    TokenResponse tokenResponse = await identityServerClient.GetAccessTokenClientCredentials();
    string accessToken = tokenResponse?.AccessToken;
}

```

46.attēls. Koda izmaiņas JWT talona pieprasīšanai ar sertifikātu

Palaižot Viss.ApiManagement.Samples.Program tiek atspoguļotas atbildes uz veiktajiem pieprasījumiem:



```
Select C:\Program Files\dotnet\dotnet.exe
RestExample
144.0
5.0
SoapExample
<Result xmlns="http://ivis.eps.gov.lv/XMLSchemas/100000/TestISServise/v1-0" xmlns:pers="http://ivis.eps.gov.lv/XMLSchemas/100001/Person/v1-0" xmlns:ivis="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0">
  <CalculationValue>30</CalculationValue>
</Result>
<Result xmlns="http://ivis.eps.gov.lv/XMLSchemas/100000/TestISServise/v1-0" xmlns:pers="http://ivis.eps.gov.lv/XMLSchemas/100001/Person/v1-0" xmlns:ivis="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0">
  <CalculationValue>5</CalculationValue>
</Result>
```

47.attēls. .NET piemēra izpildes rezultāts

### 5.6.3.2. SOAP un REST servisu izsaukšana un references talona pieprasīšana

Ja no frontend vai backend nepieciešams izsaukt API pārvaldniekā reģistrētus servisu, var izmantot PFAS STS izsniegtu OAuth2 references talonu.

.NET Core v2.2.110 piemērā Viss.ApiManagement.Samples izmantotie parametri, jānorāda RestExample.cs un SoapExample.cs datnēs:

- baseAddress – pievienoto servisu (API) adrese no Izstrādātāju portāla (Developer portal) APIS (Production and Sandbox Endpoints);
- identityAddress – PFAS STS vai API pārvaldnieka OAuth2 drošības talonu servisa adrese;
- introspectionAddress – PFAS STS OAuth2 talonu introspekcijas adrese;
- jwtPayloadAudience – PFAS OAuth2 drošības talonu servisa adrese JWT talona pieprasīšanai ar sertifikātu;
- clientId – izveidota application Consumer Key (bez urn:oauth2: prefiksa);
- clientSecret – izveidota application Consumer Secret;
- userName – izveidota PFAS AUTH lietotāja nosaukums;
- password - izveidota PFAS AUTH lietotāja parole;
- scope – atļauja ja tiek izsaukts aizsargāts serviss API;
- transactionApiAddress – Transakcijas API adrese;
- eServiceId – datu apmaiņas vai e-paklpojuma identifikators.

Piemēri satur talona izgūšanu, REST un SOAP servisu izsaukumus. Lai piemēru darbinātu izmantojot OAuth2 references talonu AuthenticationHandler.cs un SoapExample.cs datnēs jāizmanto GetAccessTokenPassword:

```

AuthenticationHandler.cs
protected override async Task<HttpResponseMessage> SendAsync(HttpRequestMessage request, Cancellat
{
    HttpResponseMessage response = null;
    async Task retry()
    {
        //var tokenResponse = await identityServerClient.GetAccessTokenClientCredentials();
        var tokenResponse = await identityServerClient.GetAccessTokenPassword(); //or await identi
    }
}


SoapExamples.cs
private async Task Multiply()
{
    //TokenResponse tokenResponse = await identityServerClient.GetAccessTokenClientCredentials();
    TokenResponse tokenResponse = await identityServerClient.GetAccessTokenPassword(); //or await
    string accessToken = tokenResponse?.AccessToken;
}

private async Task Divide()
{
    TokenResponse tokenResponse = await identityServerClient.GetAccessTokenPassword();
    //TokenResponse tokenResponse = await identityServerClient.GetAccessTokenClientCredentials();
    string accessToken = tokenResponse?.AccessToken;
}

```

#### 48.attēls. Koda izmaiņas references talona pieprasīšanai

Palaižot Viss.ApiManagement.Samples.Program tiek atspoguļotas atbildes uz veiktajiem pieprasījumiem:



```

C:\Program Files\dotnet\dotnet.exe
RestExample
144.0
5.0
SoapExample
<Result xmlns="http://ivis.eps.gov.lv/XMLSchemas/100000/TestISServise/v1-0" xmlns:pers="http://ivis.eps.gov.lv/XMLSchem
s/100001/Person/v1-0" xmlns:ivis="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0">
  <CalculationValue>30</CalculationValue>
</Result>
<Result xmlns="http://ivis.eps.gov.lv/XMLSchemas/100000/TestISServise/v1-0" xmlns:pers="http://ivis.eps.gov.lv/XMLSchem
s/100001/Person/v1-0" xmlns:ivis="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0">
  <CalculationValue>5</CalculationValue>
</Result>

```

#### 49.attēls. .NET piemēra izpildes rezultāts

### 5.6.4. Java piemēri

Java piemēri API pārvaldnieka servisu (API) izsaukšanai implementēti Viss.ApiManagement.Samples lietojumā. Lai darbinātu piemēru ir neieciešams uzstādīt Maven rīku. Tie demonstrē iespēju pieprasīt OAuth2 references talonu un JWT taona pieprasīšanu ar sertifikātu, kā arī REST un SOAP servisu izsaukumus.

Ja no backend nepieciešams izsaukt API pārvaldniekā reģistrētus servisu, var izmantot PFAS STS sertifikāta autentifikāciju, lai iegūtu JWT drošības talonu. Lai iegūtu talonu ar sertifikātu jānorāda šādi RestExample.java un SoapExample.java datnēs:

- baseAddress – pievienoto servisu (API) adrese no Izstrādātāju portāla (Developer portal) APIS (Production and Sandbox Endpoints);
- identityAddress – PFAS STS vai API pārvaldnieka OAuth2 drošības talonu servisa adrese;
- introspectionAddress – PFAS STS OAuth2 talonu introspekcijas adrese;
- jwtPayloadAudience – PFAS OAuth2 drošības talonu servisa adrese JWT talona pieprasīšanai ar sertifikātu;
- clientId - izveidotā Klienta lietojuma Consumer Key (bez urn:oauth2: prefiksa), skatīt 5.3.1.3;
- clientSecret - izveidotā Klienta lietojuma Consumer Secret, skatīt 5.3.1.3;
- certificatePath – pilns ceļš līdz sertifikātam, tam jābūt derīgam un reģistrētam PFAS kā autentifikācijas sertifikātam, un ja nepieciešams sertifikātam jābūt piešķirtām nepieciešamajām operācijām.
- certificatePassword – Sertifikāta parole.
- scope – atļauja ja tiek izsaukts aizsargāts serviss API;
- transactionApiAddress – Transakcijas API adrese;
- eServiceId – datu apmaiņas vai e-paklpojuma identifikators.

Lai piemēru darbinātu izmantojot sertifikāta autentifikāciju RestExample.cs un SoapExample.cs datnēs jāizmanto `getAccessTokenClientCredentials`, bet lai pieprasītu references talonu `getAccessTokenPassword`:

SoapExample.java

```
private void Multiply(){
```

```
CalculationSyncStub client = new CalculationSyncStub(this.baseAddress);
ServiceClient sc = client.getServiceClient();
//String accessToken = identityServerClient.getAccessTokenPassword();
String accessToken = identityServerClient.getAccessTokenClientCredentials();
```

```
private void Divide(){
```

```
try {
    CalculationSyncStub client = new CalculationSyncStub(this.baseAddress);
    ServiceClient sc = client.getServiceClient();
    String accessToken = identityServerClient.getAccessTokenClientCredentials();
    //String accessToken = identityServerClient.getAccessTokenPassword();
```

RestExample.java

```
private void Multiply() {
```

```
HttpPost httpPost = new HttpPost( uri: baseAddress + "api/Calculator/Multiply");
httpPost.setHeader( name: "Content-Type", value: "application/json");
//httpPost.setHeader("Authorization", "Bearer " + identityServerClient.getAccessTokenPassword());
httpPost.setHeader( name: "Authorization", value: "Bearer " + identityServerClient.getAccessTokenClientCredentials());
```

```
private void Divide() {
```

```
httpPost.setHeader( name: "Authorization", value: "Bearer " + identityServerClient.getAccessTokenClientCredentials());
//httpPost.setHeader("Authorization", "Bearer " + identityServerClient.getAccessTokenPassword());
```

**50.attēls. Koda izmaiņas talona veida maiņai, attēla pieprasījums ar sertifikātu.**

Ja no frontend vai backend nepieciešams izsaukt API pārvaldniekā reģistrētus servisu, var izmantot PFAS STS izsniegtu OAuth2 references talonu. Lai iegūtu references talonu jānorāda šādi RestExample.java un SoapExample.java datnēs:

- baseAddress – pievienoto servisu (API) adrese no Izstrādātāju portāla (Developer portal) APIS (Production and Sandbox Endpoints);
- identityAddress – PFAS STS vai API pārvaldnieka OAuth2 drošības talonu servisa adrese;
- introspectionAddress – PFAS STS OAuth2 talonu introspekcijas adrese;
- clientId – izveidota lietojuma Consumer Key (bez urn:oauth2: prefiksa);
- clientSecret – izveidota lietojuma Consumer Secret;
- userName – izveidota PFAS AUTH lietotāja nosaukums;
- password - izveidota PFAS AUTH lietotāja parole;
- scope – atļauja ja tiek izsaukts aizsargāts serviss API;
- transactionApiAddress – Transakcijas API adrese;
- eServiceId – datu apmaiņas vai e-paklpojuma identifikators.

Izpildot piemēru tiek atspoguļotas atbildes uz veiktajiem pieprasījumiem:

```
"C:\Program Files\Java\jdk-12.0.1\bin\java.exe" ...
RestExample
log4j:WARN No appenders could be found for logger (org.apache.http.impl.conn.BasicClientConnectionManager).
log4j:WARN Please initialize the log4j system properly.
144.4
5.0
SoapExample
<Result xmlns="http://ivis.sps.gov.lv/XMISchemas/10000/TestISService/v1-0" xmlns:pers="http://ivis.sps.gov.lv/XMISchemas/10001/Person/v1-0" xmlns:ivis="http://ivis.sps.gov.lv/XMISchemas/10001/IVIS/v1-0"><CalculationValue>30</CalculationValue></Result>
<Result xmlns="http://ivis.sps.gov.lv/XMISchemas/10000/TestISService/v1-0" xmlns:pers="http://ivis.sps.gov.lv/XMISchemas/10001/Person/v1-0" xmlns:ivis="http://ivis.sps.gov.lv/XMISchemas/10001/IVIS/v1-0"><CalculationValue>5</CalculationValue></Result>
```

## 51.attēls. Java piemēru izpildes rezultāts

### 5.6.5. PHP piemēri

PHP v7.x piemēri API pārvaldnieka servisu (API) izsaukšanai implementēti Viss.ApiManagement.Samples lietojumā.

Tie demonstrē iespēju pieprasīt OAuth2 references talonu un JWT taona pieprasīšanu ar sertifikātu, kā arī REST un SOAP servisu izsaukumus.

1. Uzstādīt šādus PHP papildinājumus (extension):
  - a. curl;

Ja no backend nepieciešams izsaukt API pārvaldniekā reģistrētus servisu, var izmantot PFAS STS sertifikāta autentifikāciju, lai iegūtu JWT drošības talonu nepieciešams (index.php) nodefinēt parametrus:

- SERVICE\_BASE\_ADDR – pievienotā servisa (API) adrese no Izstrādātāju portāla (Developer portal) APIS (Production and Sandbox Endpoints);
- SOAP\_SERVICE\_BASE\_ADDR – pievienota SOAP servisa adrese;
- CERTIFICATE\_TOKEN\_URL - PFAS OAuth2 drošības talonu servisa adrese JWT talona pieprasīšanai ar sertifikātu;
- CERTIFICATE\_CLIENT\_ID – izveidota lietojuma Consumer Key (bez urn:oauth2: prefiksa);
- CERTIFICATE\_CLIENT\_SECRET – izveidota lietojuma Consumer Secret;
- CERTIFICATE – sertifikāta nosaukums, sertifikātam jābūt \*.pem formātā, to var konvertēt no \*.pfx atkomentējot index.php datnē rindiņu:

Dokumenta kods: VRAA-13_7_17_41-VISS_2016-D_APM-VDL	Datums: 06.10.2023.	Versija: 2.23
Datne: VISS_2016.VDL.D_APM_v2.23	Izstrādāja: J.Kornijenko	Lpp.: 85 (99)

```
openssl pkcs12 -in c:\ABCTestCert.friendlyname.123-14.10.201919.34.43.pfx -out  
c:\Temp\ABCTestCert.friendlyname.123-14.10.201919.34.43.pem
```

- CERTIFICATE\_PASSWORD – sertifikāta parole.
- CERTIFICATE\_SCOPE – atļauja ja tiek izsaukts aizsargāts serviss API;
- TRANSACTION\_URL – Transakcijas API adrese;
- ESERVICE\_URN – datu apmaiņas vai e-paklpojuma identifikators
- INTROSPECT\_URL - PFAS STS OAuth2 talonu introspekcijas adrese;
- API\_ID – izsaukama API identifikātors (bez urn:oauth2: prefiksa);
- API\_SECRET – izsaukāma API atslēga.

Ja no frontend vai backend nepieciešams izsaukt API pārvaldniekā reģistrētus servisu, var izmantot PFAS STS izsniegtu OAuth2 references talonu nepieciešams (index.php) nodefinēt parametrus:

- SERVICE\_BASE\_ADDR – pievienotā servisa (API) adrese no Izstrādātāju portāla (Developer portal) APIS (Production and Sandbox Endpoints);
- SOAP\_SERVICE\_BASE\_ADDR – pievienota SOAP servisa adrese;
- TOKEN\_URL – PFAS STS vai API pārvaldnieka talona pieprasījuma adrese;
- TOKEN\_CLIENT\_ID – izveidota lietojuma Consumer Key (bez urn:oauth2: prefiksa);
- TOKEN\_CLIENT\_SECRET – izveidota lietojuma Consumer Secret;
- TOKEN\_USERNAME – izveidota PFAS AUTH lietotāja nosaukums;
- TOKEN\_PASSWORD - izveidota PFAS AUTH lietotāja parole;
- TOKEN\_SCOPE – atļauja ja tiek izsaukts aizsargāts serviss API;
- TRANSACTION\_URL – Transakcijas API adrese;
- ESERVICE\_URN – datu apmaiņas vai e-paklpojuma identifikators;
- INTROSPECT\_URL - PFAS STS OAuth2 talonu introspekcijas adrese.
- API\_ID – izsaukama API identifikātors (bez urn:oauth2: prefiksa);
- API\_SECRET – izsaukāma API atslēga.

Lai piemēru darbinātu izmantojot sertifikāta autentifikāciju index.php datnē jāizmanto `getAccessTokenWithSertificate`, bet lai pieprasītu references talonu `getAccessToken`:

```
38 | function __construct(){  
39 |     //$this->token = $this->getAccessToken();  
40 |     $this->token = $this->getAccessTokenWithSertificate();  
41 | }  
42 |
```

## 52.attēls. Koda izmaiņas talona veida maiņai, attēla pieprasījums ar sertifikātu.

Izpildot piemēru (index.php) tiek atspoguļotas atbildes uz veiktajiem pieprasījumiem:

Access token:

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIngldCI6Ij0M2FZWFluWDdldDFHdzVyeK5WUThOaXA3TSJ9.eyJpc3MiOiJodHRwOi866yRUZlI7IK\_8zxCI4qR4dTojRSLhi8jbd3xUupuHoDR6fJsCMC2T5leRc2ex1N98KyIVvuY9g45d0HD640qGFTrPmCiKy18Jy889CUS-uEUqbtokYCeWlicQunG2Nk082Z84oeHvsz65vc8y9u\_mtY\_LfSsFqdShlCWgRPFgljB4F37x4Whbdon66iWH\_xuUjSyzljjP\_PsFs6iGS-cmYB0M-d6cXguRSVvXWUcXT253tyyD-qJUUA7rJLwMjTUM-EFEv7XYs5\_51T7nyEv9Z5CDiIdRtK429HBw

Multiply result:

144.0

Divide result:

1.0

Soap multiply result:

Soap multiplication result:

```
<?xml version='1.0' encoding='utf-8'?><s:Envelope xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'><s:Header><ActivityId
xmlns='http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics' CorrelationId='d41d9cee-83e8-46fc-
96cc-3ab54a9cb7d4'>00000000-0000-0000-0000-000000000000</ActivityId></s:Header><s:Body xmlns:xsd='http://www.w3.org
/2001/XMLSchema' xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'><DefaultMethodSyncResponse
xmlns='http://ivis.eps.gov.lv/ISS/IVISService/v1-0'><IVISResponse xmlns='http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0'>
<Header><MessageID>900a5402-265b-44b1-8e31-2ee00b85748d</MessageID><MessageType>URN:IVIS:100001:XSD-Testing-
TestISServise-v1-0-TYPE-Result</MessageType><TransactionID>URN:IVIS:100001:EP-EdkTester-v1-0-TR-1393</TransactionID>
<CorrelationID>e6813a08-46b3-4249-90ec-aa60b4addb2c</CorrelationID>
<TimeStamp>2020-12-17T21:37:17.6139558+02:00</TimeStamp><Result>success</Result></Header><Body><Result
xmlns='http://ivis.eps.gov.lv/XMLSchemas/100000/TestISServise/v1-0' xmlns:pers='http://ivis.eps.gov.lv/XMLSchemas/100001/Person
/v1-0' xmlns:ivis='http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0'><CalculationValue>30</CalculationValue></Result></Body>
</IVISResponse></DefaultMethodSyncResponse></s:Body></s:Envelope>
```

Introspect result:

```
Array ( [iss] => http://ivis.eps.gov.lv/trust [aud] => urn:oauth2:825e807d-207f-4793-98b4-d8a1f20c9a51 [nbf] => 1608233831 [exp] =>
1608248231 [primarysid] => f768dd90-a3c2-409f-bce4-4772ff1500dd [legalentity] => 40003627089 [ppid] => 08107512812 [family_name]
=> Gekišs [given_name] => Juris [action] => Array ( [0] => apim:api_view [1] => CalcSync [2] => Calculation [3] =>
SearchAPIFullTextSearch.GET [4] => SearchAPIFullTextSearch.POST [5] => SearchAPI_FullTextSearch_GET [6] =>
SearchAPI_FullTextSearch_POST [7] => SearchAPIFullTextSearchGET [8] => SearchAPIFullTextSearchPOST ) [amr] => pwd
[auth_time] => 1608233832 [nameidformat] => urn:ivis:100001:name.id-viss [sub] => AU:100000-PK:08107512812 [active] => true )
```

Transaction:

URN:IVIS:100001:EP-EP186-v1-0-TR-62894

### 53.attēls. PHP piemēra izpildes rezultāts

# 1. Pielikums. IS autentifikācijas un autorizācijas pieprasījums no drošības talonu servisa

IS autentifikācijas un autorizācijas pieprasījuma saturs no drošības talonu servisa atbilst šādiem standartiem:

- Envelope, Header, Body – SOAP v1.2 (skatīt SOAP standarta 5.1, 5.2, 5.3 nodaļas);
- Action, MessageID, ReplyTo, To – WS-Addressing v1.0 (skatīt WS-Addressing standarts 3.2 nodaļu);
- Security, Timestamp – WS-Security v1.1 (5, 10 nodaļumi);
- UsernameToken - User Name Token Profile v1.1 (3.1 nodaļa);
- BinarySecurityToken – X.509 Certificate Token Profile v1.1 (3.3.2 sadaļa);
- Signature – XML Signature;
- RequestSecurityToken – WS-Trust v1.3 (3.1 nodaļa).

XML ziņojuma piemērs, izmantojot lietotāja vārdu un paroli:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:a="http://www.w3.org/2005/08/addressing"
xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-trust/200512/RST/Issue</a:Action>
    <a:MessageID>urn:uuid:fa020542-3cba-4770-8ace-ce79cbe8e494</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To
s:mustUnderstand="1">https://epakvisstv.vraa.gov.lv/STS/VISS.Pfas.STS/STS/Issue.svc/trust/13/usernamemixed</a
:To>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2015-09-17T11:46:52.859Z</u:Created>
        <u:Expires>2015-09-17T11:51:52.859Z</u:Expires>
      </u:Timestamp>
      <o:UsernameToken u:Id="uuid-12a5e50d-0d38-4f54-8924-fb9233c7a771-1">
        <o:Username>
          <!-- Removed-->
        </o:Username>
        <o:Password>
          <!-- Removed-->
        </o:Password>
        <o:Nonce>
          <!-- Removed-->
        </o:Nonce>
        <u:Created>2015-09-17T11:46:52.859Z</u:Created>
      </o:UsernameToken>
    </o:Security>
  </s:Header>
  <s:Body>
    <trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
      <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
        <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
          <wsa:Address>https://ivis.eps.gov.lv/Request.WebService</wsa:Address>
        </wsa:EndpointReference>
      </wsp:AppliesTo>
      <trust:Claims Dialect="http://schemas.xmlsoap.org/ws/2005/05/identity"
xmlns:i="http://schemas.xmlsoap.org/ws/2005/05/identity">
        <i:ClaimType Uri="urn:ivis:100001:name.id-viss" Optional="false"></i:ClaimType>
      </trust:Claims>
      <trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</trust:RequestType>
      <trust:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</trust:TokenType>
    </trust:RequestSecurityToken>
  </s:Body>
</s:Envelope>
```

XML ziņojuma piemērs izmantojot sertifikātu:

Dokumenta kods: VRAA-13_7_17_41-VISS_2016-D_APM-VDL	Datums: 06.10.2023.	Versija: 2.23
Datne: VISS_2016.VDL.D_APM_v2.23	Izstrādāja: J.Kornijenko	Lpp.: 88 (99)



```

<s:Envelope xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd">
      <u:Timestamp u:Id="TS-B65CDDAC66F66F62D314485313824851333">
        <u:Created>2015-11-26T09:49:42.485Z</u:Created>
        <u:Expires>2015-11-26T09:59:42.485Z</u:Expires>
      </u:Timestamp>
      <wsse:BinarySecurityToken>
        <!-- Removed-->
      </wsse:BinarySecurityToken>
      <ds:Signature Id="SIG-B65CDDAC66F66F62D314485313824731332"
xmlns:ds="http://www.w3.org/2000/09/xmldsigs#">
        <!-- Removed-->
      </ds:Signature>
    </wsse:Security>
    <a:Action s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-trust/200512/RST/Issue</a:Action>
    <a:To s:mustUnderstand="1" u:Id="id-
B65CDDAC66F66F62D314485313824731331">https://epak2.abcsoftware.lv/PFAS/Pfas.STS/v1-
2/STS/Issue.svc/trust/13/certificatemixed</a:To>
  </s:Header>
  <s:Body>
    <Body>
      <trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
        <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
          <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
            <wsa:Address>https://ivis.eps.gov.lv/Request.WebService</wsa:Address>
          </wsa:EndpointReference>
        </wsp:AppliesTo>
        <trust:Claims Dialect="http://schemas.xmlsoap.org/ws/2005/05/identity"
xmlns:i="http://schemas.xmlsoap.org/ws/2005/05/identity">
          <i:ClaimType Uri="urn:ivis:100001:name.id-viss" Optional="false"></i:ClaimType>
        </trust:Claims>
        <trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</trust:RequestType>
        <trust:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</trust:TokenType>
      </trust:RequestSecurityToken>
    </s:Body>
  </s:Envelope>

```

Norādītā piemērā STS izsaukums ir veikts, izmantojot STS metodi “../Issue.svc/trust/13/{name}” atbilstoši SAML un WS-\* standartiem:

VERSION	SOAP	WS-TRUST	WS-ADDRESSING	WS-POLICY	WS-SECURITYPOLICY	WS-SECURITY
13	V1.2	V1.3	2005/08	V1.2	V1.2	V1.1

Detalizēti skatiet [22] dokumentā, 3.1. nodaļā.

## 2. Pielikums. Atbilde uz IS autentifikācijas un autorizācijas pieprasījumu

Atbildes saturs uz IS autentifikācijas un autorizācijas pieprasījumu atbilst standartiem:

- Envelope, Header, Body – SOAP v1.2 (skatīt SOAP standarta 5.1, 5.2, 5.3 nodaļas);
- Action, RelatesTo – WS-Addressing v1.0 (3.2 nodaļa);
- Security, Timestamp – WS-Security v1.1 (5, 10 nodaļējumi);
- RequestSecurityTokenResponseCollection – WS-Trust v1.3 (3.2 nodaļa).
- RequestedSecurityToken elementa saturs – SAML1.1 vai SAML2.0 vai XMLEncryption

Ziņojuma piemērs:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-
trust/200512/RSTRC/IssueFinal</a:Action>
    <a:RelatesTo>urn:uuid:fa020542-3cba-4770-8ace-ce79cbe8e494</a:RelatesTo>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2015-09-17T11:46:53.283Z</u:Created>
        <u:Expires>2015-09-17T11:51:53.283Z</u:Expires>
      </u:Timestamp>
    </o:Security>
  </s:Header>
  <s:Body>
    <trust:RequestSecurityTokenResponseCollection xmlns:trust="http://docs.oasis-open.org/ws-
sx/ws-trust/200512">
      <trust:RequestSecurityTokenResponse>
        <trust:KeySize>256</trust:KeySize>
        <trust:Lifetime>
          <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd">2015-09-17T11:46:53.283Z</wsu:Created>
          <wsu:Expires xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wsssecurity-utility-1.0.xsd">2015-09-17T15:46:53.283Z</wsu:Expires>
        </trust:Lifetime>
        <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
          <a:EndpointReference>
            <a:Address>https://ivis.eps.gov.lv/Request.WebService</a:Address>
          </a:EndpointReference>
        </wsp:AppliesTo>
        <trust:RequestedSecurityToken>
          <EncryptedAssertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
            <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
              <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-
cbc"></xenc:EncryptionMethod>
              <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
                <e:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmlenc#">
                  <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-
oaep-mgf1p">
                    <DigestMethod
                      Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod
                    >
                  </e:EncryptionMethod>
                </KeyInfo>
                <o:SecurityTokenReference xmlns:o="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
                  <X509Data>
                    <X509IssuerSerial>
                      <X509IssuerName>CN=IVIS Root CA</X509IssuerName>
                      <X509SerialNumber>526677922334502603980825</X509SerialNu
mber>
                    </X509IssuerSerial>
                  </X509Data>
                </o:SecurityTokenReference>
              </xenc:EncryptedData>
            </EncryptedAssertion>
          </trust:RequestedSecurityToken>
        </trust:RequestSecurityTokenResponse>
      </trust:RequestSecurityTokenResponseCollection>
    </s:Body>
  </s:Envelope>
```

```

</KeyInfo>
<e:CipherData>
  <e:CipherValue>icGvsJvxfHT38V0ZgrMFJMzCOIP1DMqAzK6tv+LpsrbhsILs
  +R6uE2DKfNX3KnA7fkweebNp0hfoFpMtkTU51clHkWQw3sXdhRqeTxJQJN
  ztE1mJp5VjUxVyUKtGFmYt4rdZQZNYo//SIrsB5tmO21L8bb9T590qqP3L8
  LGrwTs=</e:CipherValue>
</e:CipherData>
</e:EncryptedKey>
</KeyInfo>
<xenc:CipherData>
  <xenc:CipherValue>BEmoMtZ+MmXO7+Z1+4bjrVk1RCLRPXhDofOZIIM5emhEnZ
  ai1LJtV9cvLgPVFw6spxe38biWAIrzL1+Wy1NJvJZJjee9EULChYZryZFa00tvItHR
  MG8GEIsdOjr/KgZvKEIeoVHbaS8NyerIWMvQTc2hdi2/OrtSv694/xZsBapBo03
  MY2xJOOmTMideXSKGqVQvHRYgkP0cYHTEwbC4wUIbuUWvXSb93FYtVJyqvrQ8
  NLVX8Irx47cPdVWu8E0kx5tM0hM6wNWIwK4qRfpmKfw+kpbcZN6+
  +6oAgWpdycp+oiIzPyNSEVAIj5GQ7af5vaatXTlv+e5WDM9JSJwWtN9xMc/SqzIP
  dX75pLqrkwb2uoIkhV2Z5+Gy8xTXdHwoLtQbe7UraM3ZyJRuzuuw9r92ddP0szB
  rqqzqAJ7Uv5ITZYuE54oZCIXrMfB2asFVrRrVNGfHHphwYA0milbvuf9P/WrIXm2i
  rRyMonmFRh/q05H/LEvq21Jaaexv6L2510fMP6slZJZhljehgQQ+0JXPko3+BI/H
  M4g8WbqLWem9SfLvDMNlkm/1BambQTpcfYwCAbSH+M8/8555KGRrDTV8EL
  Z1yAbBViBpxqy7hF9gf2U0HUGFFH+uw==</xenc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>
</EncryptedAssertion>
</trust:RequestedSecurityToken>
<trust:RequestedProofToken>
  <trust:BinarySecret>5I1lkTh0fgl1wHTZDKGkTEyo4fbeb4XylbOkRMV8N0Q=</trust:Binary
  Secret>
</trust:RequestedProofToken>
<trust:RequestedAttachedReference>
  <SecurityTokenReference b:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
  token-profile-1.1#SAMLV2.0" xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-
  200401-wss-wssecurity-secext-1.0.xsd" xmlns:b="http://docs.oasis-
  open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd">
    <KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
    profile-1.1#SAMLID">_c5327ddc-689b-4771-b0c4-943bd7ba9e18</KeyIdentifier>
  </SecurityTokenReference>
</trust:RequestedAttachedReference>
<trust:RequestedUnattachedReference>
  <SecurityTokenReference b:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
  token-profile-1.1#SAMLV2.0" xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-
  200401-wss-wssecurity-secext-1.0.xsd" xmlns:b="http://docs.oasis-
  open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd">
    <KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
    profile-1.1#SAMLID">_c5327ddc-689b-4771-b0c4-943bd7ba9e18</KeyIdentifier>
  </SecurityTokenReference>
</trust:RequestedUnattachedReference>
<trust:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
  1.1#SAMLV2.0</trust:TokenType>
<trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-
  trust/200512/Issue</trust:RequestType>
<trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-
  trust/200512/SymmetricKey</trust:KeyType>
</trust:RequestSecurityTokenResponse>
</trust:RequestSecurityTokenResponseCollection>
</s:Body>
</s:Envelope>

```

### 3. Pielikums. Pieprasījumu servisa biznesa transakcijas izsaukšanas pieprasījums

Pieprasījumu servisa biznesa transakcijas izsaukšanas pieprasījuma satura XML elementi tiek aprakstīti atbilstoši šādiem standartiem:

- Envelope, Header, Body – SOAP v1.2 (skatīt SOAP standarta 5.1, 5.2, 5.3 nodaļas);
- Action, MessageID, ReplyTo, To – WS-Addressing v1.0 (3.2 nodaļa);
- Security, Timestamp – WS-Security v1.1 (5, 10 nodaļējumi);
- Security elementa saturs – SAML Token Profile v1.1 (3.3 nodaļa);
- StartTransaction – šī dokumenta 3. sadaļā.

Ziņojuma piemērs:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://ivis.eps.gov.lv/ISS/RequestServiss/v1-
0/StartTransaction</a:Action>
    <a:MessageID>urn:uuid:ff7d698f-f90b-4a1a-be2d-1f7bb0c8afe6</a:MessageID>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">https://ausmatest8.vraa.gov.lv/Request.WebService/v1-
9/WcfService/ws2007FederationNoSct</a:To>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2015-09-09T12:51:19.195Z</u:Created>
        <u:Expires>2015-09-09T12:56:19.195Z</u:Expires>
      </u:Timestamp>
      <EncryptedAssertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
        <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
          <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-
cbc"></xenc:EncryptionMethod>
          <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
            <e:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmlenc#">
              <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-
mgf1p">
                <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
              </e:EncryptionMethod>
            </KeyInfo>
            <o:SecurityTokenReference>
              <X509Data>
                <X509IssuerSerial>
                  <X509IssuerName>CN=IVIS Root CA</X509IssuerName>
                  <X509SerialNumber>526677922334502603980825</X509SerialNumber>
                </X509IssuerSerial>
              </X509Data>
            </o:SecurityTokenReference>
          </KeyInfo>
          <e:CipherData>
            <e:CipherValue>G2TwcGwLwvzZzE/wblyJdBgJtj40BcAaBrljfk4uwLDInq5Lrg+wr
DuS4m7+55W6XJeCYhPjVqRMC9z3ZOCJRa0uGfdAti7n5JcTKBQeEcIQI56Rqglb
d/s1JphTQgST7608n6NAMavOC6+qB8Fpa1TTV9i6i5WJvvSmm0Xr69s=</e:Ciph
erValue>
          </e:CipherData>
        </e:EncryptedKey>
      </KeyInfo>
    </xenc:CipherData>
    <xenc:CipherValue>GZDH0LhHw6j1gnB+rn8mr/LceyXgcKy5DYZKa8kDI27YfQgDddiX
CtjgcFf5a3bdcgU5z+fJY4wDwgDwfipZTHn5fV3ibTA0mtyTrfLOMzcbPvVCztm1cieLKq
MUZdzI47btzVZW0C0WGDa8dH1AGDVHs5YmCenmI+gNzP/G7St6bgopjAtfvvvK/dm
U/FGWaxjDxW/w4rINGN5pnbOcXwgLarIBxxY2zByZEw/12eLvM+mnxhe/8WVC5/SB
Mhc8OQOx2ztTQRhKpUQyCAOf4KcHfttHgfOCsnhahKIIImNXjUoQAFEWmYJw/
/IXp4mB9ibQKeQORxUhZV++ZpmWwWU8a0eOPZwFr5h/i4jCJFgDBhruLOom5dC2
```

```

SgMnZ61kMRJxIYgCYdFjn2PtugcxhLbk2xHymK3FqaIQps5jE9G4lI0J7+QDS5s8Z1RqJ
Qv0Xc2Qf7SWZrkPmvOwwVf6NmNdiFMsnOvmDhnZMsdintQULat5nWidTzProc5J+4p
Dzh3TLolOtpsfst/VYaz7SpXamtRFcgPDyWtG8HOoL233dxfqZy+5TQffA6D5JQ==</xenc:
nc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>
</EncryptedAssertion>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></CanonicalizationMethod>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-
sha1"></SignatureMethod>
    <Reference URI="#_0">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></Transform>
      </Transforms>
      <DigestMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
      <DigestValue>iuT3099mK1e98AyhF61ouy5eydg=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>azVYxgqmP+b6ISPDC3mhdYN9KhW=</SignatureValue>
  <KeyInfo>
    <o:SecurityTokenReference b:TokenType="http://docs.oasis-open.org/wss/oasis-wss-
saml-token-profile-1.1#SAMLV2.0" xmlns:b="http://docs.oasis-open.org/wss/oasis-
wss-wssecurity-secext-1.1.xsd">
      <o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.1#SAMLID">_bbe19297-fda4-4a2b-9b19-62dca967d37b</o:KeyIdentifier>
    </o:SecurityTokenReference>
  </KeyInfo>
</Signature>
</o:Security>
</s:Header>
<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <StartTransaction xmlns="http://ivis.eps.gov.lv/ISS/RequestServiss/v1-0">
    <eServiceID>URN:IVIS:100001:EP-TEST01-v1-0</eServiceID>
  </StartTransaction>
</s:Body>
</s:Envelope>

```

Norādītajā piemērā Pieprasījuma servisa izsaukums ir veikts, izmantojot metodi “../ws2007FederationNoSct” atbilstoši WS-\* standartam:

NAME	SOAP	WS-TRUST	WS-ADDRESSING	WS-SECURITY
ws2007FederationNoSct	V1.2	V1.3	2005/08	V1.1

Detalizēti skatiet [22] dokumentā, 3.3.nodaļā.

## 4. Pielikums. Pieprasījumu servisa biznesa transakcijas izsaukšanas pieprasījuma rezultāts

Pieprasījumu servisa biznesa transakcijas izsaukšanas pieprasījuma satura veidojošie XML elementi tiek aprakstīti atbilstoši šādiem standartiem:

- Envelope, Header, Body – SOAP v1.2 (skatīt SOAP standarta 5.1, 5.2, 5.3 nodaļas);
- Action, RelatesTo – WS-Addressing v1.0 (3.2 nodaļa);
- Security, Timestamp – WS-Security v1.1 (5, 10 nodaļumi);
- StartTransactionResponse – šī dokumenta 3. sadaļā.

Ziņojuma piemērs:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://ivis.eps.gov.lv/ISS/RequestServiss/v1-
0/StartTransactionResponse</a:Action>
    <a:RelatesTo>urn:uuid:ff7d698f-f90b-4a1a-be2d-1f7bb0c8afe6</a:RelatesTo>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2015-09-09T12:51:19.525Z</u:Created>
        <u:Expires>2015-09-09T12:56:19.525Z</u:Expires>
      </u:Timestamp>
    </o:Security>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <StartTransactionResponse xmlns="http://ivis.eps.gov.lv/ISS/RequestServiss/v1-0">
      <StartTransactionResult>URN:IVIS:100001:EP-TEST01-v1-0-TR-861</StartTransactionResult>
    </StartTransactionResponse>
  </s:Body>
</s:Envelope>
```

## 5. Pielikums. Integrācijas IS servisa izsaukums

Integrācijas IS servisa izsaukuma satura XML elementi tiek aprakstīti atbilstoši standartiem:

- Envelope, Header, Body – SOAP v1.2 (skatīt SOAP standarta 5.1, 5.2, 5.3 nodaļas);
- Action, MessageID, ReplyTo, To – WS-Addressing v1.0 (3.2 nodaļa);
- Security, Timestamp – WS-Security v1.1 (5, 10 nodaļējumi);
- Security elementa saturs – SAML Token Profile v1.1 (3.3 nodaļa);
- SubmitSync – šī dokumenta 3. sadaļā.

Ziņojuma piemērs:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://ivis.eps.gov.lv/ISS/RequestServiss/v1-
    0/SubmitSync</a:Action>
    <a:MessageID>urn:uuid:c66456ed-6133-454c-9185-662fe1436d5f</a:MessageID>
    <ActivityId CorrelationId="e434d403-c847-4d07-a9c1-e14d18f22690"
    xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">f79c80da-a502-
    4d4b-b0f1-21c6cdd8ccc4</ActivityId>
    <a:ReplyTo>
      <a:Address>http://www.w3.org/2005/08/addressing/anonymous</a:Address>
    </a:ReplyTo>
    <a:To s:mustUnderstand="1">https://ausmatest8.vraa.gov.lv/Request.WebService/v1-
    9/WcfService/ws2007FederationNoSct</a:To>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
    200401-wss-wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2015-09-09T12:51:19.289Z</u:Created>
        <u:Expires>2015-09-09T12:56:19.289Z</u:Expires>
      </u:Timestamp>
      <EncryptedAssertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
        <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
          <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-
          cbc"></xenc:EncryptionMethod>
          <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
            <e:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmlenc#">
              <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oeap-
              mgf1p">
                <DigestMethod
                Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
              </e:EncryptionMethod>
              <KeyInfo>
                <o:SecurityTokenReference>
                  <X509Data>
                    <X509IssuerSerial>
                      <X509IssuerName>CN=IVIS Root CA</X509IssuerName>
                      <X509SerialNumber>526677922334502603980825</X509SerialNumber>
                    </X509IssuerSerial>
                  </X509Data>
                </o:SecurityTokenReference>
              </KeyInfo>
            <e:CipherData>
              <e:CipherValue>G2TwcGwLwvzZzE/wblyJdBgJtj40BcAaBrljfk4uwLDInq5Lrg+w
              DuS4m7+55W6XJeCYhPjVqRMC9z3ZOCJRa0uGfdAti7n5JcTKBQeEcIQI56Rqglb
              d/s1JphTQgST76O8n6NAMavOC6+qB8Fpa1TTV9i6i5WJvvSmm0Xr69s=</e:Ciph
              erValue>
            </e:CipherData>
          </e:EncryptedKey>
        </KeyInfo>
      <xenc:CipherData>
        <xenc:CipherValue>GZDH0LhHw6j1gnB+rn8mr/LceyXgcKy5DYZKa8kDI27YfQgDddiX
        CtjgcFf5a3bdcgU5z+fJY4wDwgDwfipZTHn5fV3ibTA0mtyTrfLOMzcBPvVCztm1cieLKq
        MUZdzI47btzVZW0C0WGD8dH1AGDVHs5YmCenmI+gNzP/G7St6bgopjAtfvvvK/
        eQORxUhZV++ZpmWwWU8a0eOPZWFr5h/i4jCJFgDbhrruLOom5dC2SgMnZ61kMRJx
        IYgCYdFjn2PtugcxhLbk2xHymK3FqaIQps5jE9G4lI0J7+QDS5s8Z1RqJQv0Xc2Qf7SWZ
```

```

rkPmvOwvVf6NmNdiFMsnOvmDhnZMsdintQULat5nWidTzProc5J+4pDzh3TLolOtpsfst
/VYaz7SpXamtRFcgPDyWtG8HOoL233dxfqZy+5TQffA6D5JQ==</xenc:CipherValue>
  </xenc:CipherData>
</xenc:EncryptedData>
</EncryptedAssertion>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></CanonicalizationMethod>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-
sha1"></SignatureMethod>
    <Reference URI="#_0">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"></Transform>
      </Transforms>
      <DigestMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"></DigestMethod>
        <DigestValue>d9Dc3P5p9R+IH3n++f0lkxgllg0</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>76kp8UR6FXJwiW16byX3zuv8KrE</SignatureValue>
    <KeyInfo>
      <o:SecurityTokenReference b:TokenType="http://docs.oasis-open.org/wss/oasis-wss-
saml-token-profile-1.1#SAMLV2.0" xmlns:b="http://docs.oasis-open.org/wss/oasis-
wss-wssecurity-secext-1.1.xsd">
        <o:KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.1#SAMLID">_bbe19297-fda4-4a2b-9b19-62dca967d37b</o:KeyIdentifier>
      </o:SecurityTokenReference>
    </KeyInfo>
  </Signature>
</o:Security>
</s:Header>
<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SubmitSync xmlns="http://ivis.eps.gov.lv/ISS/RequestServiss/v1-0">
    <IVISRequest xmlns="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0">
      <Header>
        <MessageID>438aa40e-d15b-4dbf-a6eb-9d792882580d</MessageID>
        <MessageType>URN:IVIS:100001:XSD-EP67-EP67Data-v1-0-TYPE-
EP67FullData</MessageType>
        <TransactionID>URN:IVIS:100001:EP-TEST01-v1-0-TR-861</TransactionID>
        <CorrelationID>c90007ac-bf4f-4577-a78c-5752d9d28f88</CorrelationID>
        <Destination>URN:IVIS:100001:ISS-LM.VDEAVK-GetUniversalServiceDataIS-v1-
0</Destination>
      </Header>
      <Body>
        <EP67FullData xmlns="http://ivis.eps.gov.lv/XMLSchemas/100665/EP67/v1-0"
xmlns:ivis="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0"
xmlns:pers="http://ivis.eps.gov.lv/XMLSchemas/100001/Person/v1-0">
          <EP67Data>
            <UniversalServiceDataRequest>
              <IdentityNumber>02010127005</IdentityNumber>
              <InstitutionCode>80</InstitutionCode>
              <AboutIdentityNumber>1118510043</AboutIdentityNumber>
              <SessionId>b5114c290e624be4a884547c2af3bcac</SessionId>
            </UniversalServiceDataRequest>
          </EP67Data>
        </EP67FullData>
      </Body>
    </IVISRequest>
  </SubmitSync>
</s:Body>
</s:Envelope>

```



## 6. Pielikums. Integrācijas IS servisa izsaukuma rezultāts

Integrācijas IS servisa izsaukuma rezultāta XML Elementi tiek aprakstīti atbilstoši standartiem:

- Envelope, Header, Body – SOAP v1.2 (skatīt SOAP standarta 5.1, 5.2, 5.3 nodaļas);
- Action, RelatesTo – WS-Addressing v1.0 (3.2 nodaļa);
- Security, Timestamp – WS-Security v1.1 (5, 10 nodaļījumi);
- SubmitSyncResponse – šī dokumenta 3. sadaļā.

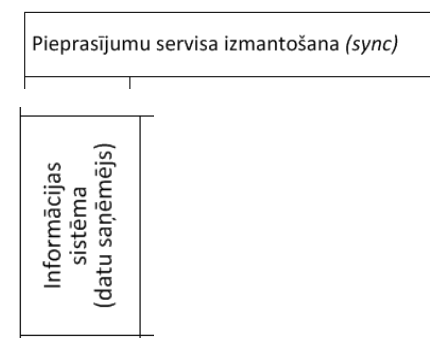
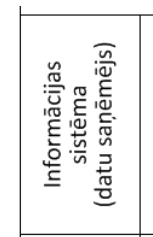



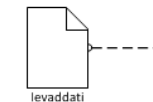
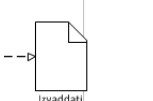

Ziņojuma piemērs:

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
xmlns:a="http://www.w3.org/2005/08/addressing" xmlns:u="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://ivis.eps.gov.lv/ISS/RequestServiss/v1-
0/SubmitSyncResponse</a:Action>
    <ActivityId CorrelationId="a07a1f70-ead-4689-a075-452ef68b01d5"
xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">f79c80da-a502-
4d4b-b0f1-21c6cdd8ccc4</ActivityId>
    <a:RelatesTo>urn:uuid:c66456ed-6133-454c-9185-662fe1436d5f</a:RelatesTo>
    <o:Security s:mustUnderstand="1" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd">
      <u:Timestamp u:Id="_0">
        <u:Created>2015-09-09T12:51:20.071Z</u:Created>
        <u:Expires>2015-09-09T12:56:20.071Z</u:Expires>
      </u:Timestamp>
    </o:Security>
  </s:Header>
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <SubmitSyncResponse xmlns="http://ivis.eps.gov.lv/ISS/RequestServiss/v1-0">
      <IVISResponse xmlns="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0">
        <Header>
          <MessageID>bcf1af67-0245-4933-b97f-f3ff23b36ba5</MessageID>
          <MessageType>URN:IVIS:100001:XSD-EP67-EP67Data-v1-0-TYPE-
EP67FullData</MessageType>
          <TransactionID>URN:IVIS:100001:EP-TEST01-v1-0-TR-861</TransactionID>
          <CorrelationID>c90007ac-bf4f-4577-a78c-5752d9d28f88</CorrelationID>
          <TimeStamp>2015-09-09T15:51:20.071942+03:00</TimeStamp>
          <Result>success</Result>
        </Header>
        <Body>
          <EP67FullData xmlns="http://ivis.eps.gov.lv/XMLSchemas/100665/EP67/v1-0"
xmlns:pers="http://ivis.eps.gov.lv/XMLSchemas/100001/Person/v1-0"
xmlns:ivis="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0">
            <ivis:IVISServiceDataEx>
              <ivis:ServiceInstanceID>URN:IVIS:100001:EP-TEST01-v1-0-TR-
861</ivis:ServiceInstanceID>
              <ivis:EServiceID>URN:IVIS:100001:EP-TEST01-v1-0</ivis:EServiceID>
            </ivis:IVISServiceDataEx>
            <EP67Data>
              <UniversalServiceDataResponse>
                <CertExists>true</CertExists>
                <PersonExists>>false</PersonExists>
              </UniversalServiceDataResponse>
            </EP67Data>
          </EP67FullData>
        </Body>
      </IVISResponse>
    </SubmitSyncResponse>
  </s:Body>
</s:Envelope>
```

# 7. Pielikums. Biznesa procesu modelēšanas valodas notācija

10.tabula

Biznesa procesu modelēšanas valodas notācija

ELEMENTS	NOSAUKUMS	DEFINĪCIJA / APRAKSTS
	Procesa nosaukums	Organizēta darbību kārtība, kas ir uzsākta ar noteiktiem nosacījumiem un ir virzīta uz noteiktu rezultātu vienas vai vairāku organizāciju ietvaros.
	Darbības veicējs	Sistēma vai loma, kas veic savu daļu no kopējā procesa.
	Procesa fāze	Loģiski izdalāmais apakšprocess.
	Darbība	Nominālais apakšprocess. Notikums: <ol style="list-style-type: none"> <li>1. tam aplūkojamā procesa ietvaros, ir nosakāmi spēkā stāšanās nosacījumi (no procesa aspekta);</li> <li>2. to producē darbības veicējs;</li> <li>3. tas realizē stāvokļu pārejas;</li> <li>4. tam stājoties spēkā, rodas: <ul style="list-style-type: none"> <li>• nosacījums citas darbības izpildei;</li> <li>• resurss.</li> </ul> </li> </ol>
	Darbības uzsākšana pēc laika	Atzīmē notikumu, kas atsākas pēc laika (ieplānots process vai lietotāja aktivitāte)
	Ievaddati	Situācija, kas ir izveidota ārpus aplūkojamā procesa un ir nosacījums kādas darbības veikšanai vai kāda resursa pastāvēšanai, aplūkojamā procesa ietvaros. Procesā saistījums ar kontekstu. Procesam var būt vairāki ievadpunkti.
	Izvaddati	Procesa rezultātā situācija, kas ir izveidota aplūkojamā procesā, un ar kuru rīkojās ārpus aplūkojama procesa. Procesā saistījums ar kontekstu. Procesam var būt vairāki tādi izvada punkti.
	Pārbaude	Situācijas novērtēšana: <ol style="list-style-type: none"> <li>1. to uzpilda darbības veicējs;</li> <li>2. tās rezultātā ir iespējami viens vai vairāki rezultāti (novērtējumi).</li> </ol> Parasti pārbaude notiek nosacījumam ar diviem rezultātiem: izpildās/neizpildās (patiess/aplams, jā/nē, ir/nav, pozitīvs/negatīvs vai cits). Tālākā procesa izpilde ir atkarīga no pārbaudes rezultāta. Modelējot procesu, autors var nolemt nerādīt scenārijus visiem pārbaudes rezultātiem, bet tikai

ELEMENTS	NOSAUKUMS	DEFINĪCIJA / APRAKSTS
		tādiem, kas ir strikti nosakāmi, attiecās uz biznesa loģiku un/vai ir svarīgas procesa daļas.