

DATU APMAIŅAS IZVEIDES VADLĪNIJAS

VADLĪNIJAS

VDAA-VDL-D_APM

13.11.2025. versija 2.30



Rīgā 2025

DOKUMENTA IDENTIFIKĀCIJA

DOKUMENTA ID:	VDAA-VDL-D_APM-V2.30-13.11.2025.
DOKUMENTA NOSAUKUMS:	Datu apmaiņas izveides vadlīnijas. Vadlīnijas.
DOKUMENTA KODS:	VDAA-VDL-D_APM
VERSIJA:	Versija 2.30, Laidiens 13.11.2025.

SASKAŅOJUMI

ORGANIZĀCIJA	VĀRDS, UZVĀRDS, AMATS	DATUMS	PARAKSTS
Valsts digitālās attīstības aģentūra	Atbildīgā persona no Pasūtītāja puses		
SIA "ABC software"	J.Korņijenko, projekta vadītājs par tehniskiem jautājumiem no Izpildītāja puses	13.11.2025.	
SIA "ABC software"	M.Pētersons, projekta vadītājs par administratīviem jautājumiem no Izpildītāja puses	13.11.2025.	
SIA "ABC software"	E.Blumberga, projekta kvalitātes kontroles vadītāja	13.11.2025.	

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 “ <i>VISS Pieprasījumu servisa modernizācija</i> ”.	SIA „ABC software”	A.Mačiņš
2.01	01.11.2019	Papildināts atbilstoši pasūtītāja komentāriem IP Nr. 26 “ <i>VISS Pieprasījumu servisa modernizācija</i> ” 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 software	„ABC	E.Stāmurs
2.15	10.11.2020	Veikti precizējumi atbilstoši Pasūtītāja komentāriem	SIA software	„ABC	E.Stāmurs
2.16	12.11.2020	Veikti precizējumi atbilstoši Pasūtītāja komentāriem	SIA software	„ABC	E.Stāmurs
2.17	06.01.2021	Aktualizēta nodaļa “5.6. API izsaukumu piemēri”.	SIA software	„ABC	J.Gekišs
2.18	29.01.2021	Pilnveidota informācija par lauka “Visible on Developer portal” aizpildīšanu.	SIA software	„ABC	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 software	„ABC	E.Stāmurs
2.20	16.11.2021	Veikti precizējumi atbilstoši Pasūtītāja komentāriem	SIA software	„ABC	E.Stāmurs
2.21	01.12.2021	Sniegtas atbildes uz Pasūtītāja komentāriem.	SIA software	„ABC	E.Stāmurs
2.22	06.10.2023	Aktualizēts darba uzdevuma “API Pārvaldnieka funkcionālie uzlabojumi VRAA EIS 2020/13/AK/CI-118-2-CPK” ietvaros.	SIA software	„ABC	E.Stāmurs
2.23	28.02.2024	Aktualizētas drošības talonu iegūšanas adreses 5. nodaļā	Valsts digitālās attīstības aģentūra		N.Priževoits
2.24	27.02.2025	Aktualizēts darba uzdevuma “API Pārvaldnieka un TransactionAPI pielāgojumi VISS IAM ietvaros” ietvaros	SIA software	„ABC	E.Stāmurs
2.25	04.03.2025	Veikti precizējumi atbilstoši Pasūtītāja komentāriem	SIA software	„ABC	E.Stāmurs
2.26	24.04.2025	Papildināta nodaļa “Datu izgūšana par drošības talonu”	SIA software	„ABC	E.Stāmurs
2.27	30.04.2025	Veikti precizējumi atbilstoši Pasūtītāja komentāriem	SIA software	„ABC	E.Stāmurs
2.28	08.05.2025	Veikti precizējumi atbilstoši Pasūtītāja komentāriem	SIA software	„ABC	E.Stāmurs
2.29	29.05.2025	Aktualizēts darba uzdevuma “API Pārvaldnieka versijas atjaunošana un migrācija” ietvaros	SIA software	„ABC	M.Strikāns
2.30	13.11.2025	Papildināts pieteikuma “DAI-831” ietvaros	SIA software	„ABC	E.Stāmurs

Satura rādītājs

ATTĒLU SARAKSTS	7
1. IEVADS.....	8
1.1. Terminu un pieņemtie apzīmējumi	8
1.1.1. Terminu	8
1.1.2. Apzīmējumi.....	9
1.2. Saistība ar citiem dokumentiem	10
1.3. Dokumenta pārskats	11
2. VISS NODROŠINĀTĀS DATU APMAIŅAS.....	12
2.1. API Pārvaldnieka izmantošana	13
2.2. DIT izmantošana datu apmaiņai	14
3. DATU IZPLATĪŠANAS TĪKLS	16
3.1. Nepieciešamās tiesības DIT izmantošanai	17
3.2. DIT lietošanas scenārijs.....	17
3.2.1. Parakstīšanās uz esošu citas iestādes kanāla plūsmu	17
3.2.2. Jaunā kanāla izveidošana	17
4. API PĀRVALDNIKS	18
4.1. API Pārvaldnieka darbības principi	19
4.2. API Pārvaldnieka lietošanas scenārijs.....	19
4.3. Izstrādātāju portāls (Developer portal)	20
4.3.1. Publicēta biznesa servisa izmantošana	20
4.3.1.1. Publicēto API pārlūkošana	20
4.3.1.2. Pieejas tiesību pieprasīšana no VDAA.....	21
4.3.1.3. Klienta lietojuma reģistrācija.....	22
4.3.1.4. Servisu abonēšana.....	24
4.3.1.5. Servisa izmantošanas tiesību pieprasīšana no datu devēja	24
4.3.1.6. Lietojuma pielāgošana servisa izsaukšanai	26
4.3.2. Publicēta biznesa servisa izmantošanas pārtraukšana	28
4.4. Publicētāju portāls (API Publisher)	29
4.4.1. Pieejas tiesību pieprasīšana no VDAA	29
4.4.2. Servisu (API) izstrāde	30
4.4.3. Servisu (API) reģistrēšana un publicēšana	30

4.4.3.1. Aizsargāts API.....	31
4.4.3.2. REST servisu reģistrācija	33
4.4.3.3. Integrācijas Servisu reģistrācija	33
4.4.3.4. GraphQL servisu reģistrācija.....	34
4.4.3.5. Scope pievienošana	35
4.4.4. Pieejas servisam (API) piešķiršana	38
4.4.5. Pieejas servisam (API) apturēšana.....	38
4.4.6. Transakcijas obligātuma konfigurēšana	39
4.5. Transakciju API izmantošana	40
4.5.1. Transakcijas uzsākšanas metode	40
4.5.1.1. Piemēri ar datu nodošanu body daļā	41
4.5.1.2. Piemēri ar datu nodošanu query daļā	42
4.5.2. Transakcijas apturēšanas metode	43
4.5.2.1. Piemēri ar datu nodošanu body daļā	43
4.5.2.2. Piemēri ar datu nodošanu query daļā	45
4.5.3. Transakcijas statusa metode	46
4.6. API izsaukumu piemēri	47
4.6.1. Priekšnosacījumi	48
4.6.2. Izsaukumu Curl piemēri.....	48
4.6.2.1. OAuth2 drošības talona pieprasījums bez scope aizpildīšanas	48
4.6.2.2. OAuth2 drošības talona pieprasījums ar norādītiem scope.....	50
4.6.2.3. Datu izgūšana par drošības talonu.....	52
4.6.2.4. REST servisa izsaukums	56
4.6.2.5. SOAP servisa izsaukšana	57
4.6.3. .NET piemēri	60
4.6.3.1. SOAP un REST servisu izsaukšana un talona pieprasīšana ar sertifikātu	60
4.6.4. Java piemēri	61
4.6.5. PHP piemēri	63
5. PIELIKUMS. BIZNESA PROCESU MODELĒŠANAS VALODAS NOTĀCIJA	65

Attēlu saraksts

1.attēls. VISS centrālās datu apmaiņas platformas galvenās komponentes	12
2.attēls. Biznesa scenārijs API Pārvaldnieka izmantošana sinhrono pieprasījumu realizācijai 13	
3.attēls. Biznesa scenārijs API pārvaldnieka servisa izmantošana sinhrono pieprasījumu realizācijai, kad sagatavojamā atbilde ir lielāka par 4 MB	14
4.attēls. Biznesa scenārijs DIT izmantošana datu izplatīšanai	15
5.attēls. Datu izplatīšanas kanāls (DIK)	16
6.attēls. Datu savākšanas kanāls (DSK)	16
7.attēls. API Pārvaldnieka risinājuma koncepts.....	18
8.attēls. API Pārvaldniekā publicēta biznesa servisa izsaukums	19
9.attēls. Izstrādātāju portālā (Developer portal) publicētie API	21
10.attēls. Rest servisa detalizētais skats.....	21
11.attēls. Izstrādātāju portālā (Developer portal) reģistrētie klienta lietojumi	22
12.attēls. Klienta lietojuma izveides forma	23
13.attēls. Izveidotā klienta lietojuma/Viss.IamIdentity.Server klienta talona izgūšanas dati ..	23
14.attēls. Publicēta servisa (API) kartiņa.....	24
15.attēls. Publicēta servisa (API) piesaiste lietojumam - abonēšana.....	24
16.attēls. Datu devēja kontaktinformācija.....	25
17.attēls. Datu devēja kontaktinformācija.....	25
18.attēls. Viss.IamIdentity lietojuma IAM klienta detalizētais skats.....	26
19.attēls. Klienta lietojuma detalizētais skats API Pārvaldnieka saskarnē	27
20.attēls. REST servisa swagger definīcija	28
21.attēls. SOAP servisa WSDL lejupielāde	28
22.attēls. Izstrādātāju portālā (Developer portal) reģistrētā klienta lietojuma kartiņa	29
23.attēls. API Publicētāja saskarne	30
24.attēls. API reģistrēšanas veidi	31
25.attēls. Oauth2 mediatora politikas izveide	32
26.attēls. Politikas atribūtu aizpildīšana	33
27.attēls. Autoritātes identifikators lestādes un struktūrvienības klasifikatorā.....	33
28.attēls. Servisa rediģēšanas skata atvēršana	36
29.attēls. Scope datu aizpildīšana.....	36
30.attēls. Scope pievienošana	37
31.attēls. Transakcijas uzsākšana	40
32.attēls. Koda izmaiņas JWT talona pieprasīšanai ar sertifikātu	61
33.attēls. .NET piemēra izpildes rezultāts	61
34.attēls. Koda izmaiņas talona veida maiņai, attēla pieprasījums ar sertifikātu.....	62
35.attēls. Java piemēru izpildes rezultāts.....	63
36.attēls. Koda izmaiņas talona veida maiņai, attēla pieprasījums ar sertifikātu.....	63
37.attēls. PHP piemēra izpildes rezultāts	64

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:

- VDAA 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 serviss	IS 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 saistītajā 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
VDAA	Valsts digitālās 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)
TA VVDZ	Tieslietu administrācijas 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] Standarts LVS 72:1996, leteicamā prakse programmatūras projektējuma aprakstīšanai.
- [2] Fizisko personu datu aizsardzības likums (<http://www.dvi.gov.lv/likumdosana/fpda/>).
- [3] Metadatu un e-pakalpojumu identifikācija. Standarts (VRAA-6_15_11_58-VISS_2010-MET_EPAK_ID-STD-V1.03-21.02.2012).
- [4] <http://www.oasis-open.org/specs/index.php#wssv1.0>.
- [5] <http://www.oasis-open.org/specs/index.php#wssv1.1>.
- [6] E-pakalpojumu arhitektūras izstrāde. Vadlīnijas. (VDAA-VDL-EPAK_ARH).
- [7] Datu izplatīšanas tīkls: ārējās saskarnes. Programmatūras saskarnes projektējums. (VRAA-13_7_17_41-VISS_2016-DIT_AS-PPA).
- [8] Elektronisko dokumentu krātuve: ārējās saskarnes. Programmatūras projektējuma apraksts (VDAA-PPA_AS-Viss.EDK).
- [9] VISS Drošības talonu serviss. Programmētāja rokasgrāmata (VDAA-PR-DTS).
- [10] API Pārvaldnieks (WSO2). Programmētāja rokasgrāmata. (VDAA-PR-Viss.ApiManagement)
- [11] WSO2 reģistrēto servisu abonēšana un abonentu bloķēšana, <https://apim.docs.wso2.com/en/latest/design/advanced-topics/block-subscription-to-an-api/>
- [12] WSO2 dokumentācija, "Create and Publish an API", <https://apim.docs.wso2.com/en/latest/design/create-api/create-rest-api/create-a-rest-api/>;
- [13] 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/>;
- [14] WSO2 dokumentācija, "Creating a Common Policy", <https://apim.docs.wso2.com/en/4.3.0/design/api-policies/create-policy/#creating-a-common-policy>
- [15] WSO2 Oauth Mediator, <https://github.com/imesh/wso2-oauth-mediator>
- [16] E-pakalpojuma izveidošanas ceļvedis latvija.lv portālam. Programmētāja rokasgrāmata (VDAA-PR-Lvp.EservicePlatform)
- [17] OAuth 2.0 autorizācijas ietvara specifikācija, <https://tools.ietf.org/html/rfc6749>
- [18] JWT standarts, <https://datatracker.ietf.org/doc/html/rfc7519>;
- [19] Pieejas tiesību pieprasīšanas procedūra https://viss.gov.lv/lv/Informacijai/partneriem/Sadarbibas_proceduras/API-abonesana
- [20] Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants <https://www.rfc-editor.org/rfc/rfc7521.txt>

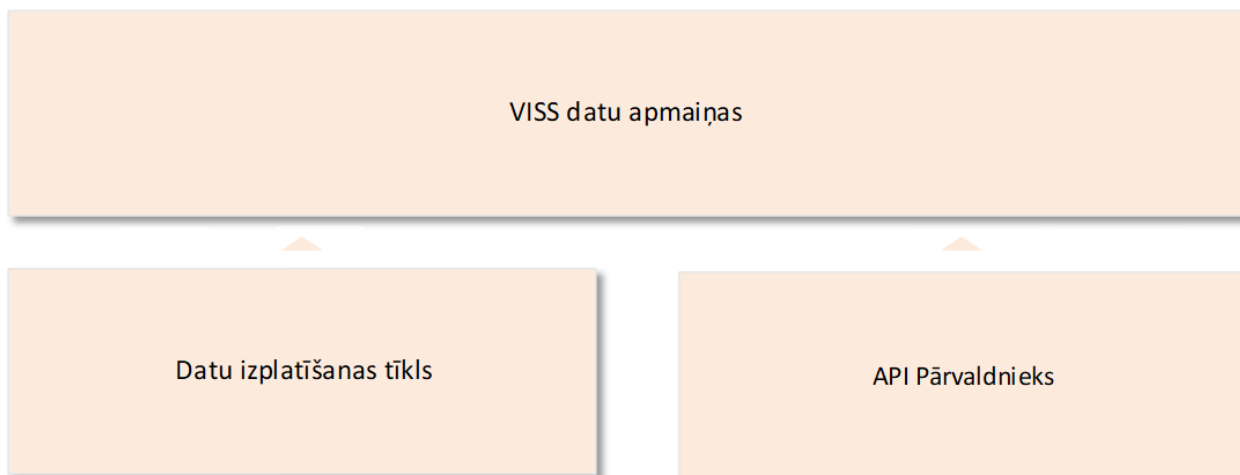
1.3. Dokumenta pārskats

Dokumentu veido pieci nodaļījumi un pielikumi:

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

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:

- 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).

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

3.tabula

VISS centrālo datu sniegšanas platformas komponentes salīdzinājums

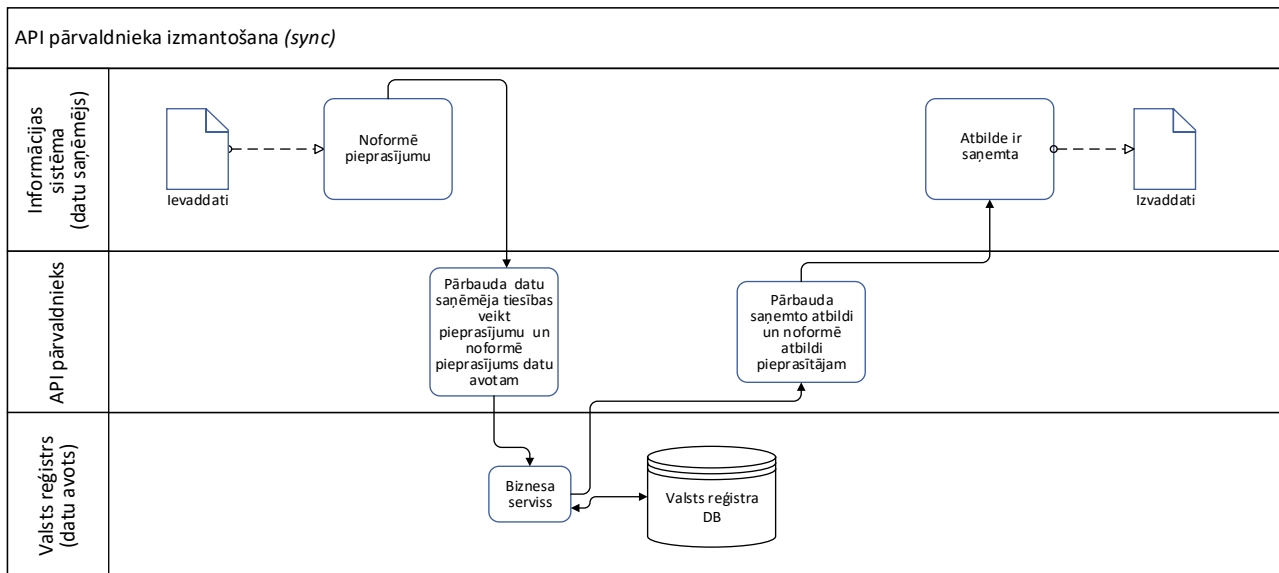
	DIT	API PĀRVALDNIKS
• Integrācija latvija.lv e-pakalpojumos		X
• Sinhrona datu apmaiņa		X

• Asinhrona datu apmaiņa	X	
• Ziņojuma validācija	X	
• Ziņojumu apjoma ierobežojums	0,5 GB	4 MB
• Lielapjoma (līdz 0,5 GB) datu sūtīšana	X	
• Garantēta piegāde	X	
• Formātu atbalsts:		
• XML	X	X
• JSON		X
• Protokolu/API atbalsts:		
• SOAP	X	X
• FTPs	X	
• REST		X
• GraphQL		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.



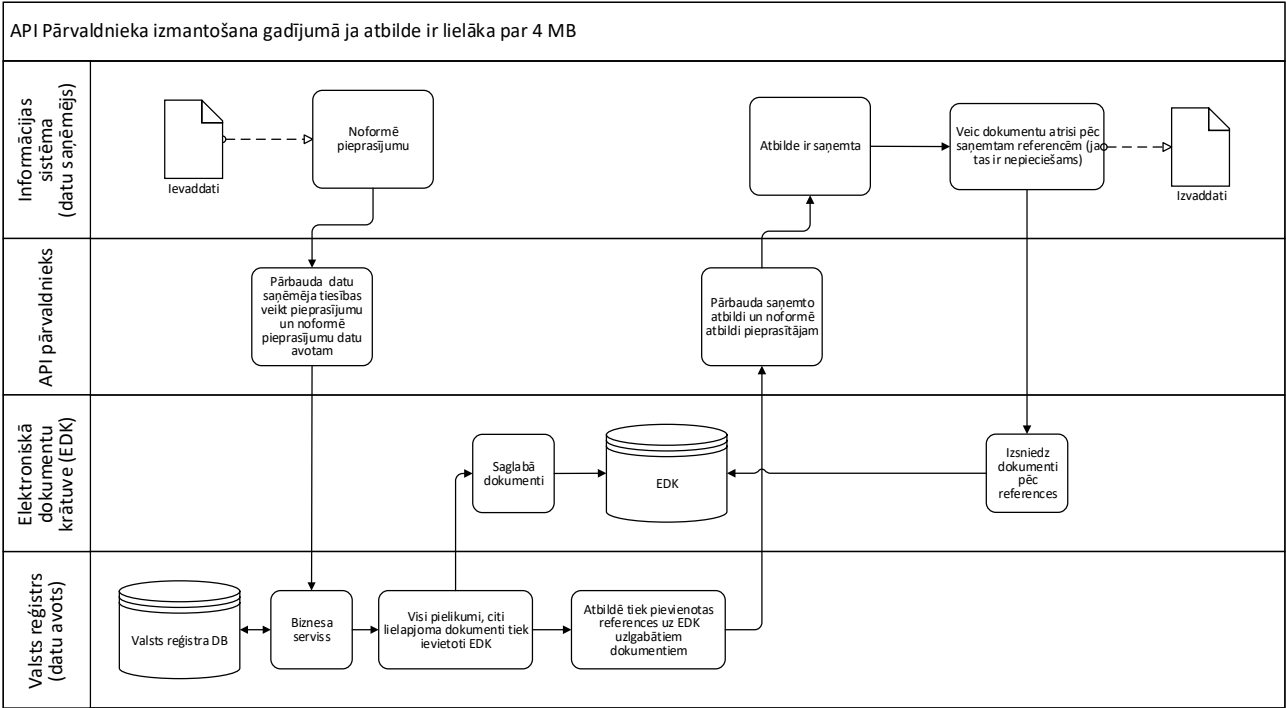
2.attēls. Biznesa scenārijs API Pārvaldnieka izmantošana sinhrono pieprasījumu realizācijai

API pārvaldnieks nodrošina tikai sinhronu pieprasījumu apstrādi.

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, u tt Elektroniskajā dokumentu krātuvē (EDK, skat.[8]), kas nodrošina to ilglaicīgu uzglabāšanu. EDK izmantošanas scenāriji jāsaskaņo ar VDAA uzsākot datu apmaiņas projektēšanu.

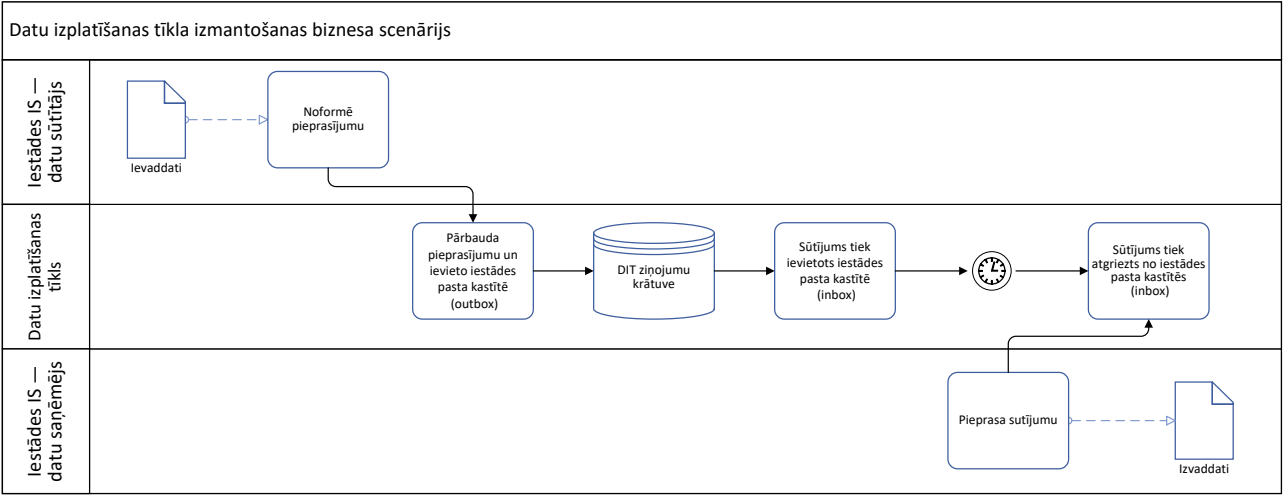


3.attēls. Biznesa scenārijs API pārvaldnieka servisa izmantošana sinhrono pieprasījumu realizācijai, kad sagatavojamā atbilde ir lielāka par 4 MB

2.2. 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. 4.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).



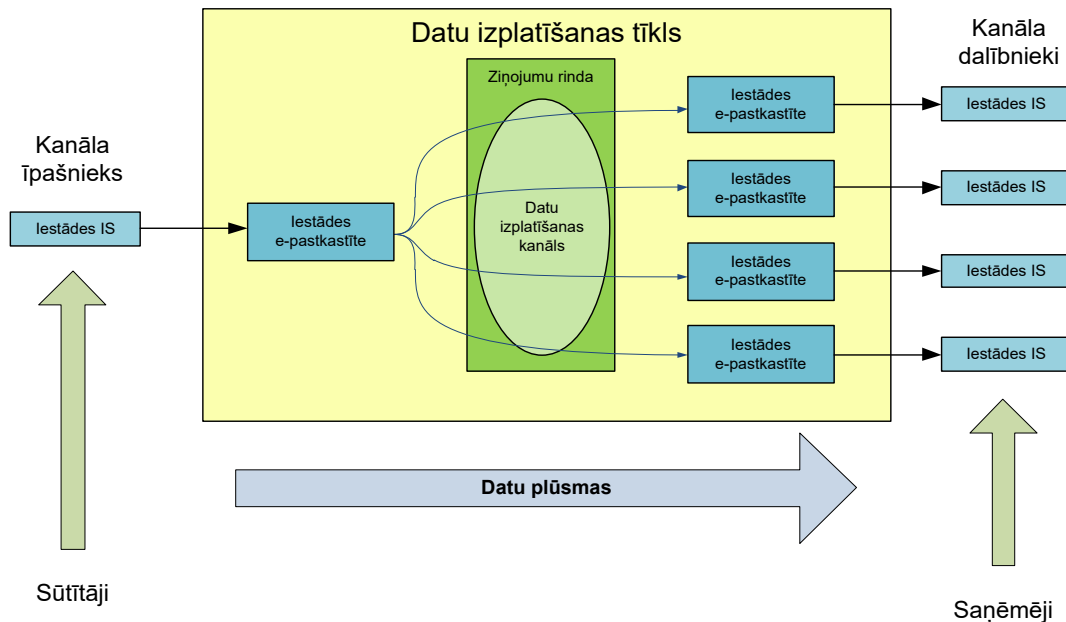
4.attēls. Biznesa scenārijs DIT izmantošana datu izplatīšanai

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.

3. Datu izplatīšanas tīkls

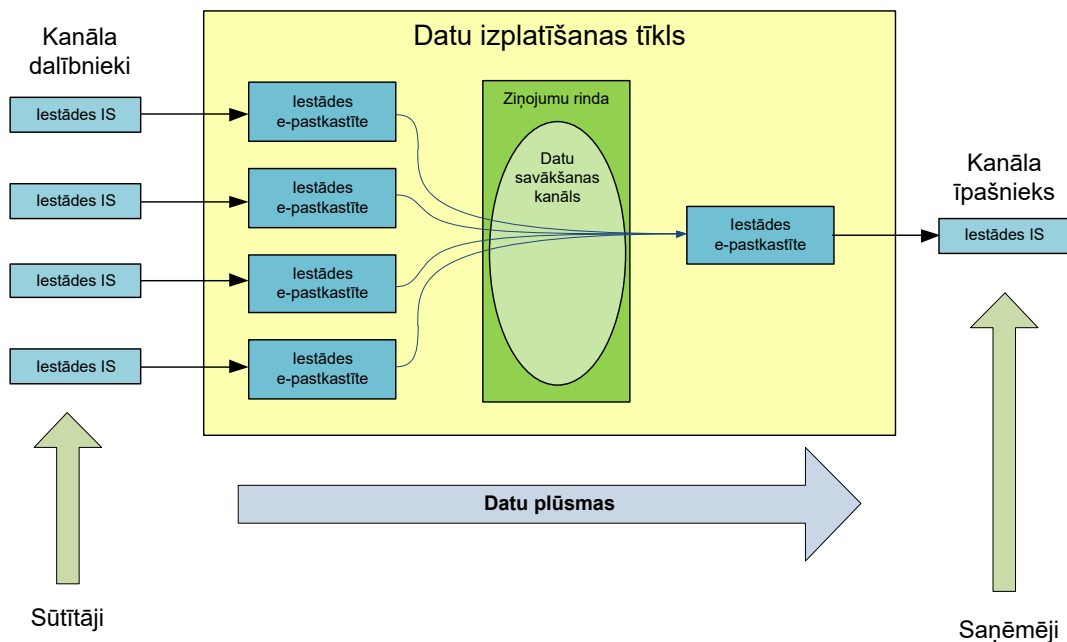
Datu izplatīšanas tīkla risinājuma ietvaros informācijas apmaiņa notiek, izmantojot kanālus, kur katrs kanāls nodrošina ziņojumu pārsūtīšanu no viena īpašnieka vairākiem dalībniekiem (DIK kanāls) vai no vairākiem dalībniekiem vienam īpašniekam (DSK kanāls):

- Datu izplatīšanas kanāls (DIK), skat. 5.attēlā, pa kuru ziņojumi plūst no viena sūtītāja, kas ir kanāla īpašnieks, vairākiem saņēmējiem, kas ir kanāla dalībnieki.



5.attēls. Datu izplatīšanas kanāls (DIK)

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



6.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. [7] dokumentā.

3.1. Nepieciešamās tiesības DIT izmantošanai

Tiesības DIT infrastruktūras izmantošanai VDAA 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.

3.2. 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.

3.2.1. *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 pievienošanos 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. [7] dokumentā).

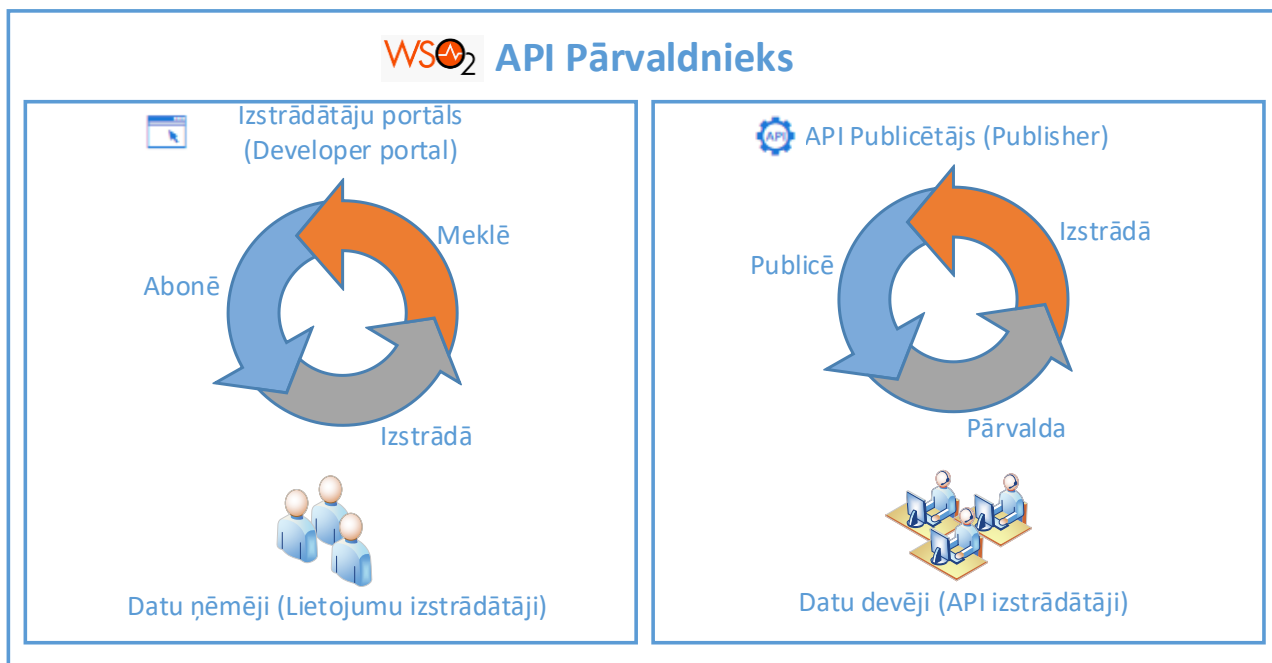
3.2.2. *Jaunā kanāla izveidošana*

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

4. API Pārvaldnieks

API Pārvaldnieks ir uz WSO2 API Manager (*enterprise service bus*¹) standarta risinājuma bāzēts, un tas nodrošina izstrādātājiem divas galvenās pamatfunkcijas:

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



7.attēls. API Pārvaldnieka risinājuma koncepts

Šī nodalījuma ietvaros tiek nodrošināts:

- API Pārvaldnieka darbības principu ieskats, skat. 4.1. nodaļu;
- API Pārvaldnieka tipisko lietošanas scenāriju, skat. 4.2. nodaļu;
- Nepieciešamo darbību kopumus ("road map"), lai datu ņēmēji varētu izmantot API Pārvaldniekā publicētos servissus, skat. 4.3. nodaļu.
- Nepieciešamo darbību kopumus ("road map"), lai datu devēji varētu publicēt un pārvaldīt savus servissus API Pārvaldniekā, skat. 4.4. nodaļu;

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

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

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

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

Drošības talonu var pieprasīt no šādas adreses VDAA testa vidē:

- <https://viss-portal-test.vraa.gov.lv/lamIdentity.Server/connect/token>

Drošības talonu var pieprasīt no šādas adreses VDAA produkcijas vidē:

- <https://viss-portal.viss.gov.lv/lamIdentity.Server/connect/token>

¹ https://en.wikipedia.org/wiki/Enterprise_service_bus

4.1. API Pārvaldnieka darbības principi

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

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

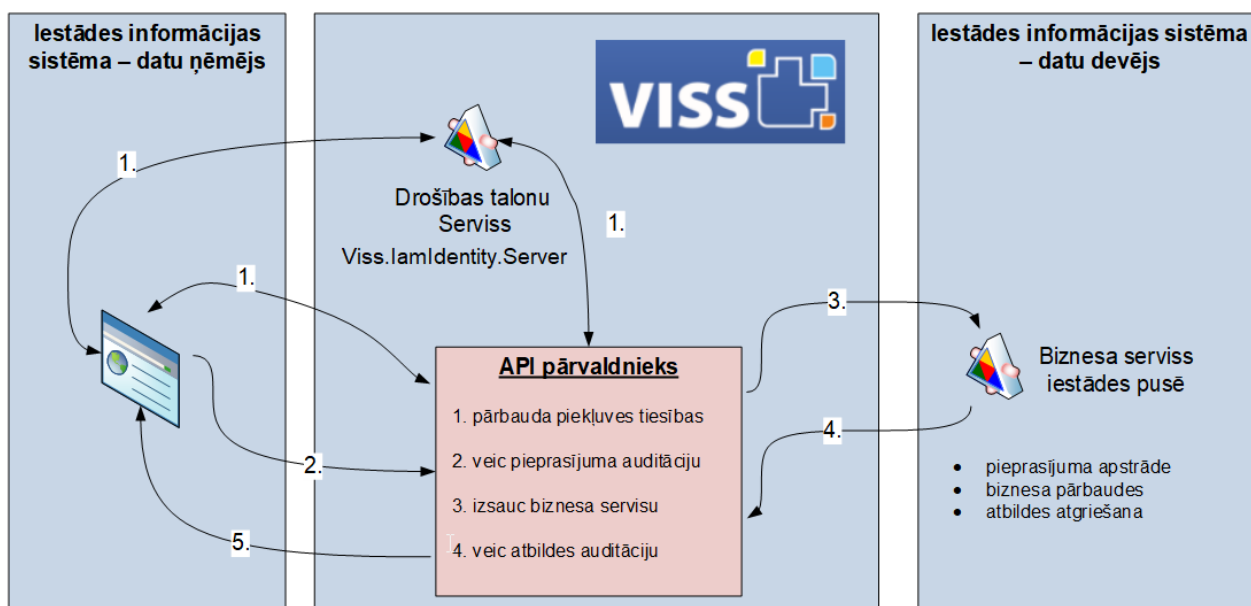
Pieprasījumu apjoma ierobežojums ir 4 MB. Ja nepieciešams pārsūtīt lielākus pielikumus, datnes utt., ir jālieto DIT (skat. 2.2.nodaļu) vai Elektroniskās dokumentu krātuves (EDK) risinājums (skat. [8] saistīto dokumentu). EDK izmantošanas scenāriji jāaskaņo ar VDAA uzsākot datu apmaiņas projektēšanu.

Anonīmi (bez drošības talona) pieprasījumi netiek nodrošināti.

API Pārvaldniekā publicētie (statusā - Published) servisi (API) ir publiski pieejami Izstrādātāju portālā (Developer portal) saskarnē (skat. 4.3. nodaļu), bet to reģistrēšana un publicēšana tiek veikta, izmantojot API publicētāja (API Publisher) saskarni (skat. 4.4. nodaļu), kas pieejama tikai autentificētiem lietotājiem.

4.2. API Pārvaldnieka lietošanas scenārijs

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



8.attēls. API Pārvaldniekā publicēta biznesa servisa izsaukums

Sinhronais izsaukums ietver šādus soļus:

1. Autentifikācija un autorizācija Viss.lamIdentity.Server 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 Viss.lamIdentity.Server 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.

4.3. 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).

4.3.1. Publicēta biznesa servisa izmantošana

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

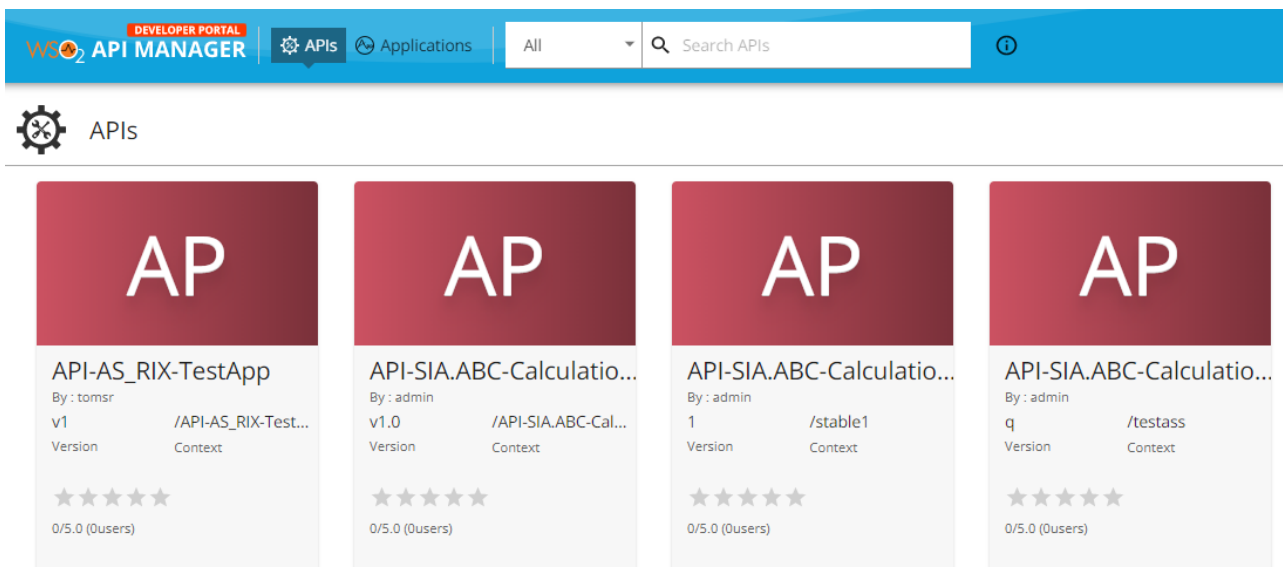
1. Jāsameklē sev nepieciešamo servisu Izstrādātāju portālā (Developer portal), skatīt 4.3.1.1 nodaļu.
2. Jāsaņem pieejas tiesības no VDAA izmantot Izstrādātāju portālā (Developer portal) publicētos servissus skatīt 4.3.1.2 sadaļu
3. Izstrādātāju portālā (Developer portal) jāreģistrē datu ņēmēja Klienta lietojums, kas izmantos API Pārvaldniekā eksponētos servissus, skatīt 4.3.1.3. sadaļu;
4. Jāveic servisu abonēšana, skatīt 4.3.1.4. sadaļu;
5. Nepieciešamības gadījumā jāaskaņo ar servisa īpašnieku tiesības izmantot izvēlēto API, skatīt 4.3.1.5;
6. Jāpielāgo Klienta lietojums servisa izsaukšanai, skatīt 4.3.1.6. sadaļu;

E-pakalpojumu izstrādātājiem biznesa servisu izsaukšanai jāizmanto e-pakalpojumu platformas LvpContext.Request funkcionalitāte skat. e-pakalpojuma izveides ceļveža programmētāja rokasgrāmatu [16]. No šīs sadaļas e-pakalpojumu izstrādātājiem saistoša ir tikai pieejas tiesību pieprasīšana no VDAA (skatīt 4.3.1.2) un datu devējiem (skat. 4.3.1.5).

4.3.1.1. Publicēto API pārlūkošana

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

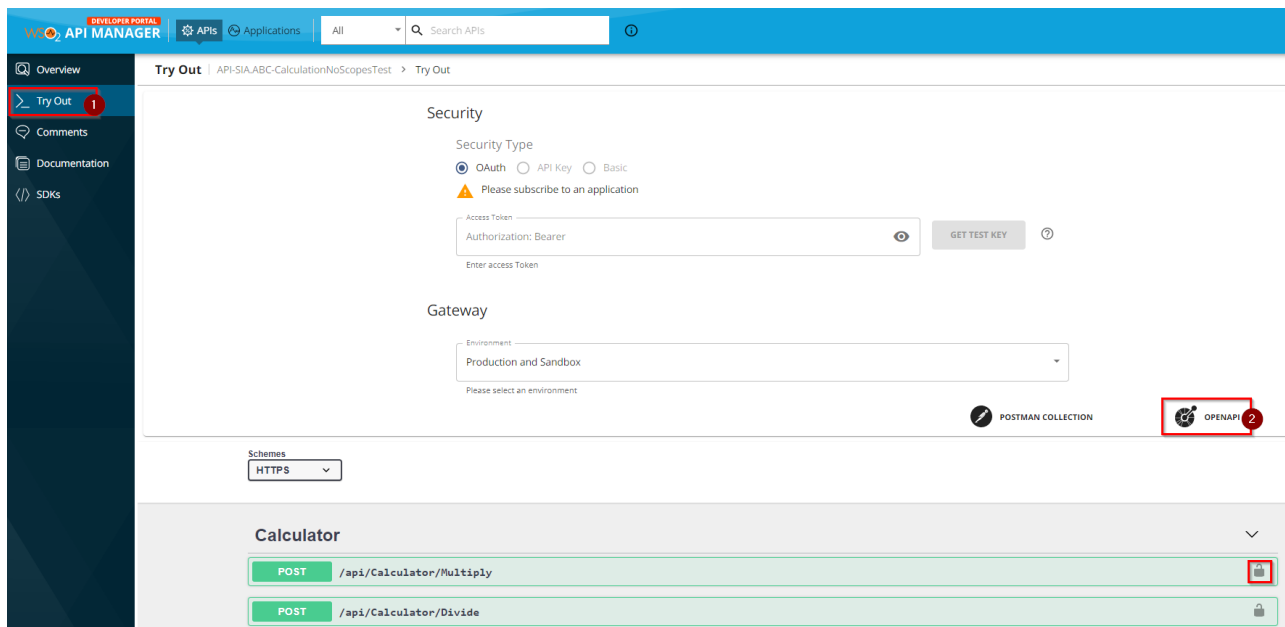
- Atvērt Izstrādātāju portāla (Developer portal) sadaļu APIS, tā pieejama arī neautentificētiem lietotājiem.
- APIS sadaļā pārlūkošanai ir pieejami visi publicētie servisi (API). Saskaņā nodrošina iespēju filtrēt publicētos servissus pēc statusa un nosaukuma.



9.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 atslēdziņas 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.



10.attēls. Rest servisa detalizētais skats

4.3.1.2. Pieejas tiesību pieprasīšana no VDAA

Pieejas pieprasīšana un sniegšana Izstrādātāju portālā (Developer portal) servisu izmantošanai tiek veikta pēc noteiktas procedūras, skatīt [19].

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 Viss.lamIdentity.Server lietotājs – nepieciešamo 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).

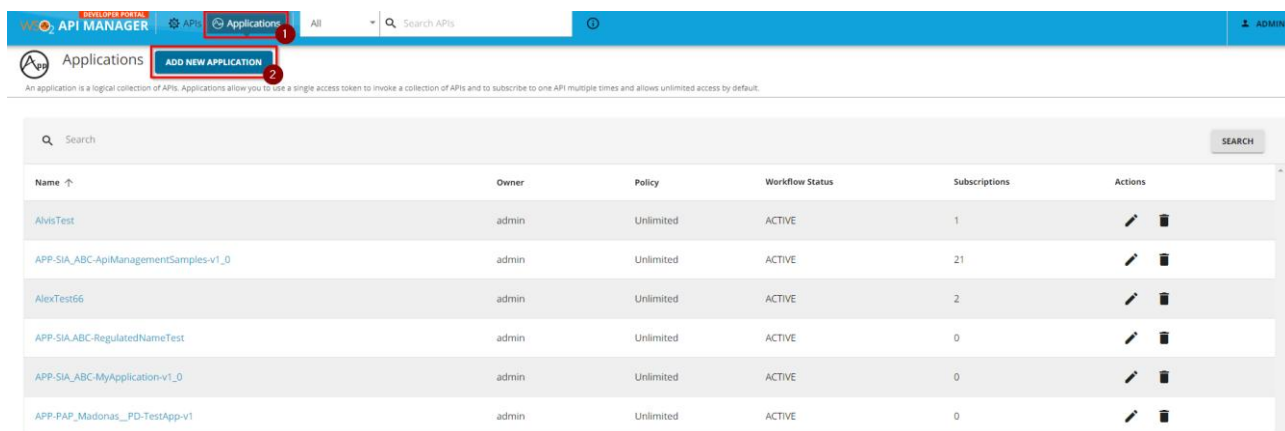
E-pakalpojumu platformas mikroservis ir reģistrēti kā klienta lietojums un starp to un visu e-pakalpojumu izmantotajiem servisiem ir jābūt izveidotiem abonementiem.

Aizsargāta API gadījumā datu devējam jāpiešķir atļaujas (scopes) e-pakalpojuma IDS uzticamajai pusei atbilstoši 4.4.4. sadaļai.

4.3.1.3. Klienta lietojuma reģistrācija

Pēc pieejas tiesību saņemšanas, datu ņēmējs reģistrē savu Klienta lietojumu izstrādātāju portālā (Developer portal). Reģistrējot Klienta lietojumu, tiek automātiski izveidota atbilstošs OIDC klients Viss.lamIdentity.Server. Lai reģistrētu jaunu Klienta lietojumu, jāveic šādas darbības:

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



11.attēls. Izstrādātāju portālā (Developer portal) reģistrētie klienta lietojumi

2. Jānospiež saiti “ADD NEW APPLICATION”;
3. Jānorāda Klienta lietojuma nosaukums, (Viss.lamIdentity.Server tiks izveidots atbilstošs klients ar šādu nosaukumu), pieprasījumu limitu, aprakstu un jānospiež poga “Save”.

Klienta lietojuma nosaukums jāveido vadoties pēc Klienta lietojuma nosaukuma formāta notācības: “APP-{Autoritātes identifikators (visscore:authorityIdent) no iestāžu un struktūrvienību klasifikatora (Authority), izmanto apakšvītru “_” punktu vietā, piemēram, “SIA_ļestade1”}-{datu ņēmēja klienta lietojuma nosaukums bez speciālajiem simboliem, kurā konkrētais serviss (API) tiks izmantots, piemēram, “TapisUI” u.tml.}”. Piemērs: “APP-SIA_ļestade1-TapisUI”.

Create an application

Create an application providing name and quota parameters. Description is optional.

Required fields are marked with an asterisk (*)

Application Name *

My Application

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

Shared Quota for Application Tokens *

10PerMin

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

12.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 Viss.lamIdentity.Server saglabāti Consumer Key un Consumer Secret (skat. 13.attēlu). Šie atribūti vēlāk būs nepieciešami, lai pieprasītu talonu pirms servisa (API) izsaukšanas.

Overview

Production Keys

OAuth2 Tokens

API Key

APP-SIA_ABC-TapisUI

0 Subscriptions

EDIT

DELETE

Production OAuth2 Keys

Key and Secret

PROVIDE EXISTING OAUTH KEYS

Consumer Key

urn:oauth2:f94f8f12-645a-41ca-8b98-f59bd0790948

Consumer Key of the application

Consumer Secret

09ceaf404b985f5ea2d724e6c0a9361d9de9849af6i

Consumer Secret of the application

13.attēls. Izveidotā klienta lietojuma/Viss.lamIdentity.Server klienta talona izgūšanas dati

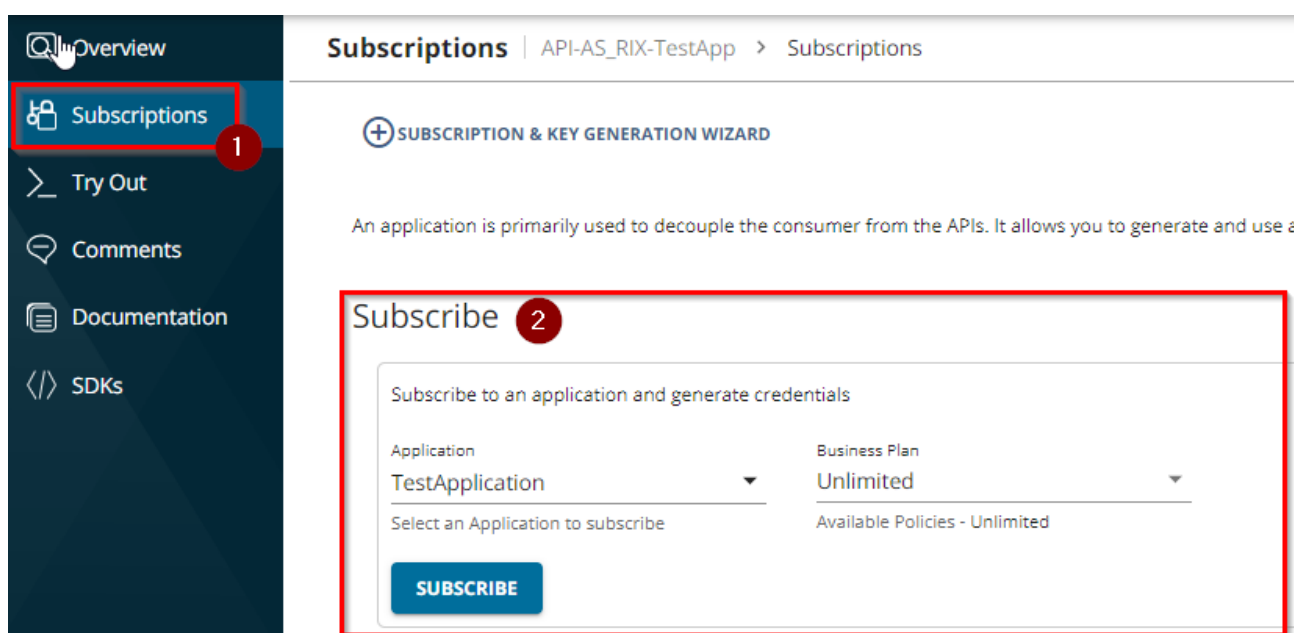
- Pēc klienta lietojuma reģistrācijas klienta lietojuma kartiņā un atslēgu noģenerēšanas. Ir iespējams iegūt talonu no Viss.lamIdentity.Server, skatīt 4.6.2 sadaļu. Piekļuves talona (access_token) iegūšanai neizmantojot saskarnē piedāvāto iespēju “CURL To Generate Access Token/CURL Piekļuves Talona Ģenerēšana”.
- 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 4.3.1.5.

4.3.1.4. 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 servissus (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 servissus no tā var izmantot dažādus Viss.IamIdentity.Server tehniskos lietotājus (sertifikātus). Ja autentifikācijai tiek izmantots sertifikāts tad rekomendājams katrā klienta lietojumā izmantot vienu sertifikātu.

Lai veiktu publicēta servisa abonēšanu:

1. Jāatver Izstrādātāju portālā (Developer portal) publicēto servisu (API) saraksts, skatīt 4.3.1.1; API sadaļā “Izmēģināt/Try Out” iespējams apskatīt konkrētajam servisam pieejamās metodes un izsaukuma piemērus;
2. Jāizvēlas viens publicēto servisu (API) no saraksta un jāatver tā kartiņas cilne Subscriptions, skatīt 14. attēlu;



14.attēls. Publicēta servisa (API) kartiņa

3. Laukā Application jāizvēlas Klienta lietojums, kurš izmantos šo servisu un jānospiež poga Subscribe, skatīt 15. attēlu;

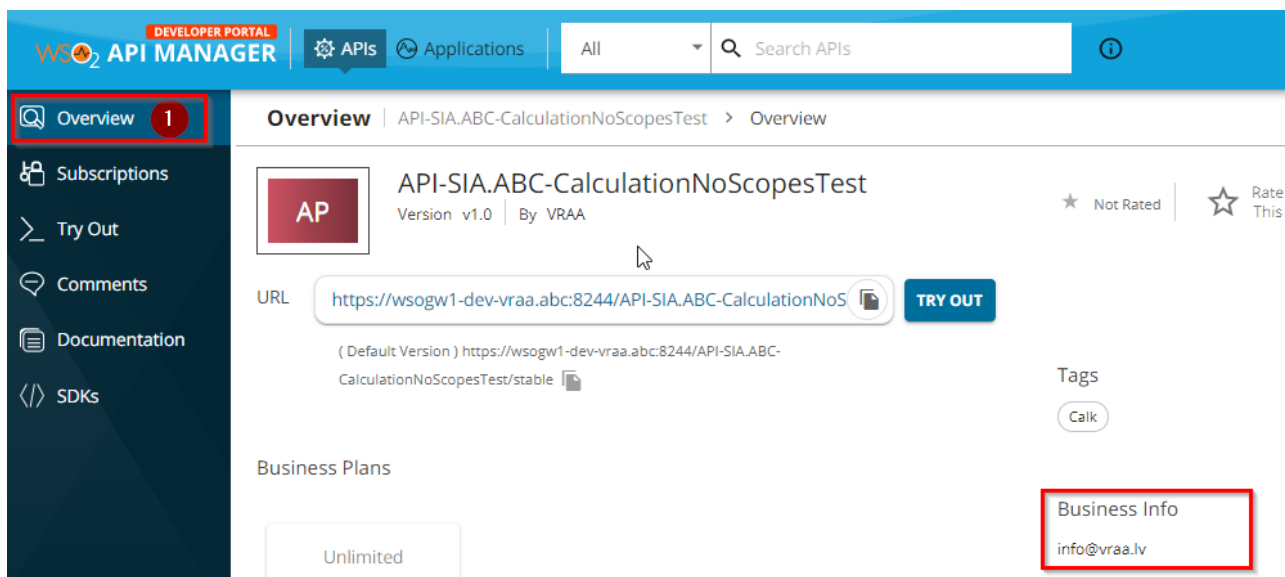


15.attēls. Publicēta servisa (API) piesaiste lietojumam - abonēšana

4.3.1.5. Servisa izmantošanas tiesību pieprasīšana no datu devēja

Ja pieejas tiesības jāpiešķir e-pakalpojumam:

1. Pēc tam kad no VDAA 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.



16.attēls. Datu devēja kontaktinformācija

2. 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 (clientId) jeb IDS uzticamās puses identifikators, piemēram, urn:oauth2:test-vraa:myapplication;
 - d. Saraksts ar atļauju (scope) identifikatoriem, pieejami swagger definīcijā, piemēram, API-SIA_ABC-TestApi-v1_0-Scope1
3. Datu devējs izmantojot Viss.LvplIdentity administratora saskarni pēc norādītā klienta identifikatora sameklē IDS klientu un piesaista tai norādītās atļaujas, scope.

Ja pieejas tiesības jāpiešķir tehniskajam procesam:

1. 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

17.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 4.3.1.3 nodaļas) jeb Viss.lamIdentity.Server klienta 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 Viss.lamIdentity administratora saskarni pēc norādītā klienta identifikatora sameklē IAM klientu un piesaista tai norādītās atļaujas, scope.

4.3.1.6. Lietojuma pielāgošana servisa izsaukšanai

Pēc servisa (API) abonēšanas (skat. 6. sadaļā) servisu var izsaukt no reģistrētā Klienta lietojuma. Publicētie servisi ir izsaucami tikai drošā veidā, izmantojot OAuth2 autorizāciju ar Viss.lamIdentity.Server sertifikāta autentifikāciju, detalizētai informācijai skatīt .Net, PHP un Java izsaukumu piemērus 4.6 nodaļā.

E-pakalpojumu izstrādātājiem biznesa servisu izsaukšanai jāizmanto e-pakalpojumu platformas LvpContext.Request funkcionalitāte skat. e-pakalpojuma izveides ceļveža programmētāja rokasgrāmatu [16]. No šīs sadaļas e-pakalpojumu izstrādātājiem saistoša ir tikai pieejas tiesību pieprasīšana no VDAA (skatīt 4.3.1.2) un datu devējiem (skat. 4.3.1.5).

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

- consumer-key – API Pārvaldniekā reģistrētā Klienta lietojuma atslēga, piemēram „urn:oauth2:fa49e5c0-2345-4fa1-b9a3-08ccf46e2d3e”. Tā Viss.lamIdentity.Server administratoriem tiek attēlota Viss.lamIdentity klienta 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 4.3.1.3. sadaļu

Detalizēta informācija par IAM klientu

Izmantojiet šo lapu IAM klienta informācijas detalizētai apskatei.

IAM klientu īpašības	Atslēgas	API Pārvaldnieka atļaujas
Klienta tips	Tehniskais lietojums (API)	
Talona veids	References	
Nosaukums	API-SIA-ABC-TapisUI	
Identifikators	urn:oauth2:fa49e5c0-2345-4fa1-b9a3-08ccf46e2d48	
Aktīvs	☒	
Piekļuves talona dzīves laiks	3600	
Identitātes talona dzīves laiks	120	
Apraksts		

18.attēls. Viss.lamIdentity lietojuma IAM klienta detalizētais skats

19.attēls. Klienta lietojuma detalizētais skats API Pārvaldnieka saskarnē

- sertifikāts – IAM reģistrēts sertifikāts. To reģistrē VDAA, balstoties uz datu ņēmēja pieteikumu, skatīt 4.3.1.2.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 Viss.IamIdentity.Server, nepieciešams izpildīt šādu pieprasījumu, norādot iepriekš uzskaitīto parametru vērtības:

```
curl --location 'https://viss-portal-
test.vraa.gov.lv/IamIdentity.Server/connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_assertion_type=urn:ietf:params:oauth:client-
assertion-type:jwt-bearer' \
--data-urlencode
'client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIyNjQ0S
IsImp0aSI6IjYxYTRlNzljLWFiZmYtNGE2YS1iY2RiLWlzMjQ3MWU1NGE2OSIsImZlcyI6IjI2
NDg5IiwiaXVkiOiJoidXJuOiJFUlQtdTE5COKRhBR1IuU1RTL2NvbW5lY3QvdG9rZW4iLCJleHAiOi
iXNzQWMDUzNjg5IiwibmJmIjoimTC0MDA1MDA4OSJ9.uP0kXHgzCzL-
W8W5JxEKl7wJmksSGNMPZKCR0dvIn8qzTZ7fQLKwWymEWYMC0_607iAMsXQCKL0aLJWP1EkFfr
ko3YqMW75SKbUz7V68yleYysjJcgrqtxnhUn5FHmUFD5zLpXb91YqejqNm80hLlfkOE3iLUDgA
irKmPgS9fz-
Up79JgXLM8wq_leTAKHRioAsmP7HCBIBcneaexfeTDcbHmDhxqWIPpVRNv92mh4ZnqFrkZ2hIN
91KENTWYhU6EDsupATDS8lUSUBI9gsxkoNbYXCTlaGbT0XNE8kC3IogfrAwDeLas_THxAEXA_k
kL2ksPEngUPrw7-itiKSBU4Q' \
--data-urlencode 'grant_type=client_credentials'
```

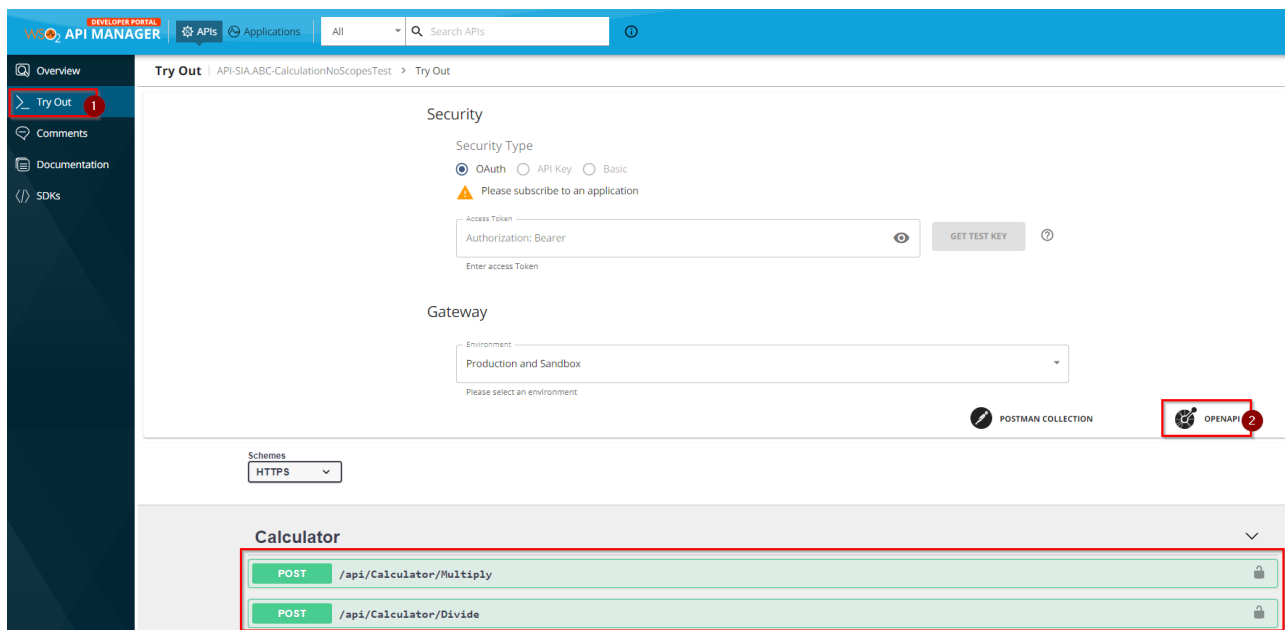
Šis pieprasījums atbildē atgriezīs references talona identifikatoru atribūtā „access_token”:

```
{"access_token":"urn:uuid:34c9bc8a-ffbf1-46cf-9133-
cb276888b6c9","expires_in":3600,"token_type":"bearer"}
```

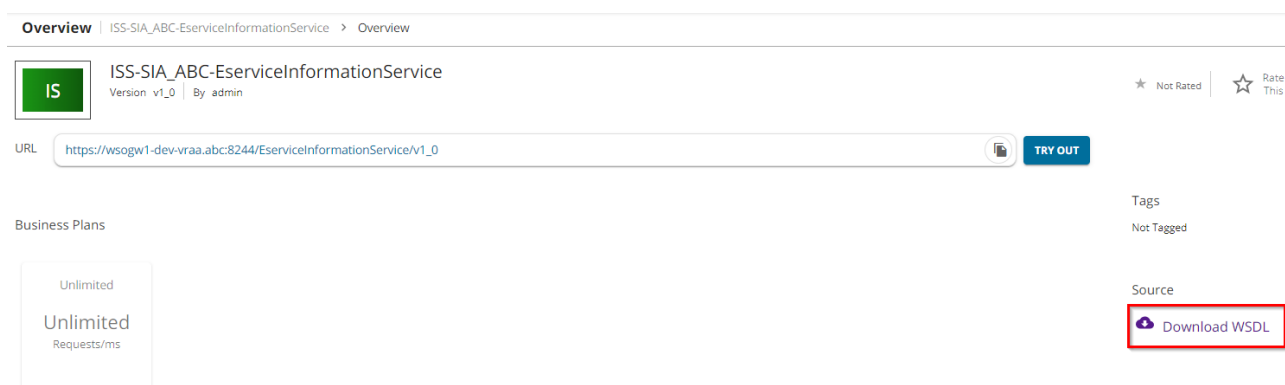
Izmantojot saņemto talona referenci, iespējams izsaukt API Pārvaldniekā reģistrēto servisu (API), piemēram, tādu, kas sareizina divus skaitļus, norādot tā ieejas parametros 3 un 4 (transakcijas identifikators ir obligāts, obligātumu iespējams atslēgt konkrētam API, to iepriekš saskaņojot ar VDAA):

```
curl -H "Authorization: Bearer urn:uuid:34c9bc8a-ffbf1-46cf-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/s
table/v1/api/Calculator/Multiply"
```

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



20.attēls. REST servisa swagger definīcija

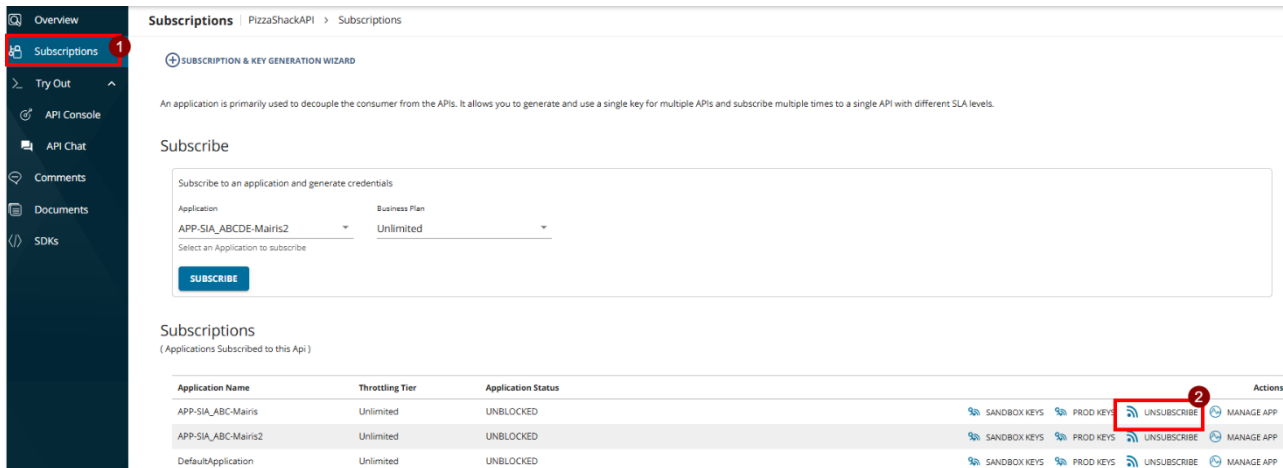


21.attēls. SOAP servisa WSDL lejupielāde

4.3.2. 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 [11.attēls](#) ~~11.attēls~~;
 - b. Izvēlas reģistrēto klienta lietojumu (Application) no saraksta un jāatver tā kartiņu. Kartiņā pāriet uz cilni Subscriptions skatīt [22.attēls](#) ~~22.attēls~~;



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

- c. Sarakstā sameklē servisu (API), kura izmantošana ir pārtraukta, nospiež kolonnā Actions redzamo saiti Unsubscribe.

4.4. 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 servissus (API) API Pārvaldniekā.

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

1. Jāizstrādā servisi atbilstoši specifikācijai, skatīt 4.4.2;
2. Jāpieprasa VDAA pieejas tiesības pie API Publisher portāla, skatīt 4.4.1;
3. Jāreģistrē serviss (API) API Publisher portālā;
4. Jāpublicē reģistrētais serviss, lai tas būtu atrodams Izstrādātāju portālā (Developer portal);

4.4.1. Pieejas tiesību pieprasīšana no VDAA

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

1. Datu devēji rakstiski pieprasa tiesības servisu izmantošanai no VDAA, atbilstoši VISS Programsaskarnes (API) publicēšanas procedūrai.
2. VDAA datu devējiem piekļuvi pie Publicētāju portāla (API Publisher) dod tikai tad, kad VDAA no datu devēja ir saņēmis rakstisku piekļuves pieprasījumu.
3. VDAA saņemot datu devēja tiesību pieteikumu:
 - a. Reģistrē datu devēja iestādi lestāžu un struktūrvienību (Authority) klasifikatorā.
 - b. Izveido API Pārvaldnieka risinājuma lietotāju. Piešķir tam tiesības Publicētāju portālā (API Publisher) reģistrēt un publicēt servissus (API).
 - c. Izveido datu devēju iestādei atbilstošu Viss.lamIdentity un Viss.LvpIdentity iestādes administratorus, kuri savas iestādes ietvaros varēs piešķirt datu ņēmēju klientu lietojumiem (IAM un IDS klientiem) atļaujas izmantot savus servissus.
4. VDAA informē datu devēju par veiktajām darbībām un tam izsniedz:
 - a. Piešķirto iestādes 6 zīmju identifikatoru un iestādes kodu.

- b. API publicētāja piekļuves.
- c. Viss.lamIdentity un Viss.LvplIdentity piekļuves datus.

E-pakalpojumu gadījumā Klienta lietojumus kā IDS klientus Viss.LvplIdentity reģistrē VDAA.

4.4.2. Servisu (API) izstrāde

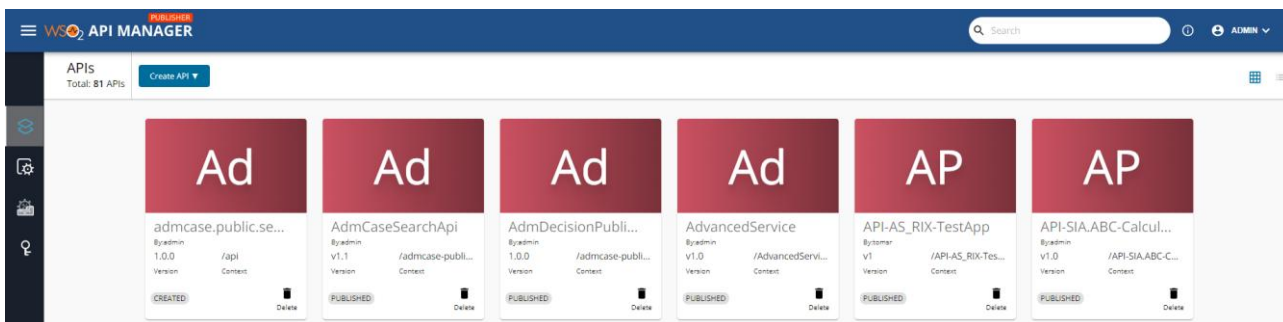
Izmantojot API Pārvaldnieku, biznesa servisu nepieciešams izstrādāt, ievērojot Open API Specification 2.0 vai Open API Specification 3 specifikāciju un to varētu reģistrēt API Publicētājā (API Publisher).

Specifikācija nosaka, ka katram servisam (API) jābūt aprakstītam izmantojot Swagger, veidojot Swagger ieteicams metodes aizsargāt ar scope (skat. 4.4.3.2. paragrāfu). Tās publicēto servisu (API) metodes, kas nav aizsargātas ar scope ir pieejamas jebkuram Klientu lietojumam pēc šī servisa (API) abonēšanas Izstrādātāju portālā (Developer portal). Jebkurš Klientu lietojums var abonēt jebkuru no publicētajiem servisiem (API), bet servisa (API) publicētājam ir iespējams šo abonentu bloķēt (skat. 4.4.4. sadaļu), ja nav saskaņota tā izmantošana.

4.4.3. Servisu (API) reģistrēšana un publicēšana

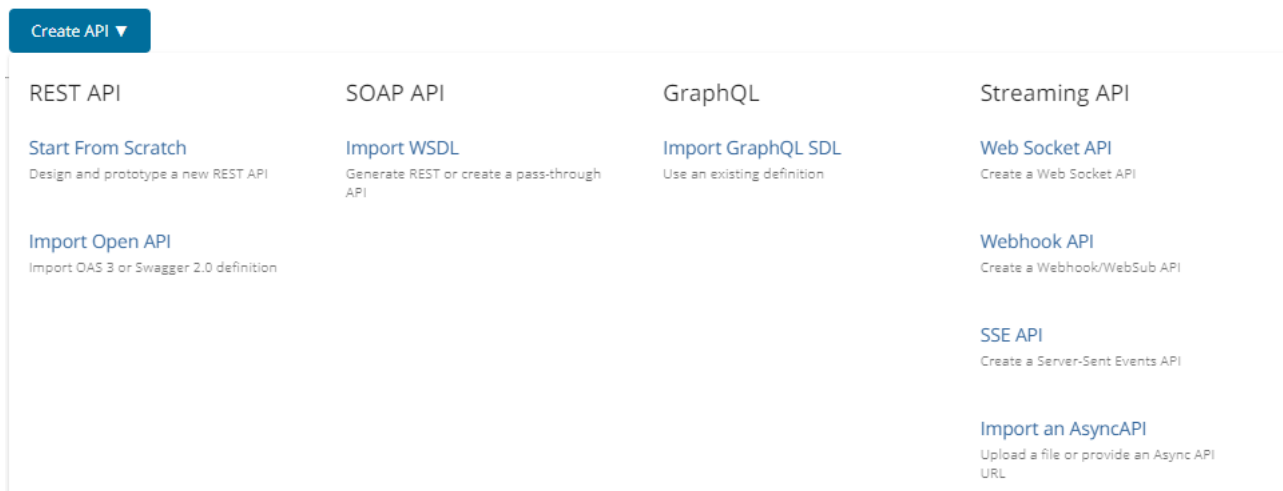
Lai reģistrētu API publicētājā (API Publisher) jaunu servisu (API), jāveic šādas darbības:

- 1) Jāautenticējas API publicētājā (API Publisher);



23.attēls. API Publicētāja saskarne

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



24.attēls. API reģistrēšanas veidi

- 4) Jāveic servisa (API) konfigurēšana atbilstoši API Pārvaldnieka dokumentācijai (REST [12], SOAP [13]) un zemāk esošiem paragrāfiem, tajā skaitā par nosaukumu veidošanu skat. 4.4.3.3 paragrāfu. (SOAP) vai skat. 4.4.3.2 paragrāfu (REST) šī soļa beigās seviss (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 VDAA 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 VDAA 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".

4.4.3.1. 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 politika pieprasījumu pūsmai dokumentācijai [14]. Tā kā šo politiku var izmantot vairāki API vēlams to veidot kā kopēju politiku, lai turpmāk to var pievienot citiem API izmantojot esošo konfigurāciju.

Politikas definēšanā aizpilda vispārējo informāciju par to, kādiem nolūkiem tā paredzēta. Svarīgi nodefinēt politikas atribūtus, lai tie atbilstu mediators [15] sagaidāmajiem dinamiskiem datiem. Kā redzams politikas atribūtu definīcijā (99 attēls), nepieciešamie atribūti atbilst lauku nosaukumiem politikas failā:

```
<sequence xmlns="http://ws.apache.org/ns/synapse" name="oauth-sequence">
  <class name = "Viss.ApiManagement.Extensions.OAuthMediator.OAuthMediator">
    <property name="endpointId" value="{{endpointIdValue}}"></property>
  </class>
</sequence>
```

General Details

Provide the name, description and applicable flows of the policy.

Name *

oauthMediationCommon

Enter Policy Name (E.g.: Add Header)

Version *

v 1

Enter Policy Version (E.g.: v1)

Description

This policy configures OAuth Mediation

Short description about the policy

Applicable Flows *

☒ Request ☐ Response ☐ Fault

Supported API Types *

☒ HTTP ☒ SOAP ☒ SOAPFOREST

Gateway Specific Details

Define the Gateway (s) that will be supporting this policy. Based off of this selection, you can upload the relevant business logic inclusive policy file.

Upload Policy File *

Policy file contains the business logic of the policy

oauthMediation.j2

Policy Attributes

Define attributes of the policy.

+ Add Policy Attribute

Name *

endpointIdValue

Attribute Name

Display Name *

Endpoint id

Attribute Display Name

!

Save

Cancel

25.attēls. Oauth2 mediatora politikas izveide

Politikas pievienošana notiek servisa detalizētā skata “Policies” sadaļā (**Error! Reference source not found.attēls**). Izvēoties atbilstošo politiku “Request Flow” sadaļā tiks pieprasīts aizpildīt nepieciešamos atribūtus, kas atbilst mediatora konfigurācijai. 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 VDAA administratori pēc pieprasījuma. Mediatoru iespējams veidot izmantošanai vienā vai vairākos servisos (API).

Configure oauthMediationCommon : v1

Reset

Description

This policy configures OAuth Mediation

Endpoint Id *

SOME_ID

Cancel

Save

26.attēls. Politikas atribūtu aizpildīšana

4.4.3.2. 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 jaukā ievadāmā vērtība ir reģistrējūtīga un jāievada tāda kāda tā ir norādīta Klasifikatoru katalogā, piemēram: “**AS.RIX**”;

iestādes identifikācijas kods ir iegūstams “Iestādes un struktūrvienības” ”klasifikatoru katalogā (skat. [27.attēls](#)27.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

27.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 Viss.lam talonu pēc ActAs principa. Viss.lam 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 Viss.LvpIdentity lietotnē kā atļaujas.

4.4.3.3. 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 apakšsvītrām “_”. 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 jaukā ievadāmā vērtība ir reģistrējama un jāievada tāda kāda tā ir norādīta Klasifikatoru katalogā, piemēram: **“SIA.ABC”**;
 - **Autoritātes identifikators ir iegūstams “Iestādes un struktūrvienības” “klasifikatoru katalogā (skat. [27.attēls](#)[27.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_VDAA-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 jaunos 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 Viss.lam talonu pēc ActAs principa. Viss.lam 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 Viss.LvpIdentity lietotnē kā atļaujas.

4.4.3.4. GraphQL servisu reģistrācija

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

- SDL datnēm jābūt .graphql paplašinājumā un viņu saturam jāatbilst text/plain formātam.
- Pilnais servisa nosaukums veidojas noteiktā formāta notācijā – **“GQL-{Autoritātes identifikators (visscore:authorityIdent) no iestāžu un struktūrvienību klasifikatora (Authority), izmanto apakšsvītru “_” 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 reģistrējama un jāievada tāda kāda tā ir norādīta Klasifikatoru katalogā, piemēram: **“AS.RIX”**;

Iestādes identifikācijas kods ir iegūstams “Iestādes un struktūrvienības” “klasifikatoru katalogā (skat. [27.attēls](#)[27.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 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 Viss.lam talonu pēc ActAs principa. Viss.lam 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 Viss.LvpIdentity lietotnē kā atļaujas.

4.4.3.5. 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. 4.3.1.5. paragrāfu), ir jāpieprasa arī atbilstošie scope.

Scope pievienošana servisiem (API) notiek manuāli (skatīt 4.4.3.5.1. punktu) vai automatiski (skatīt 4.4.3.5.2. 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”***.

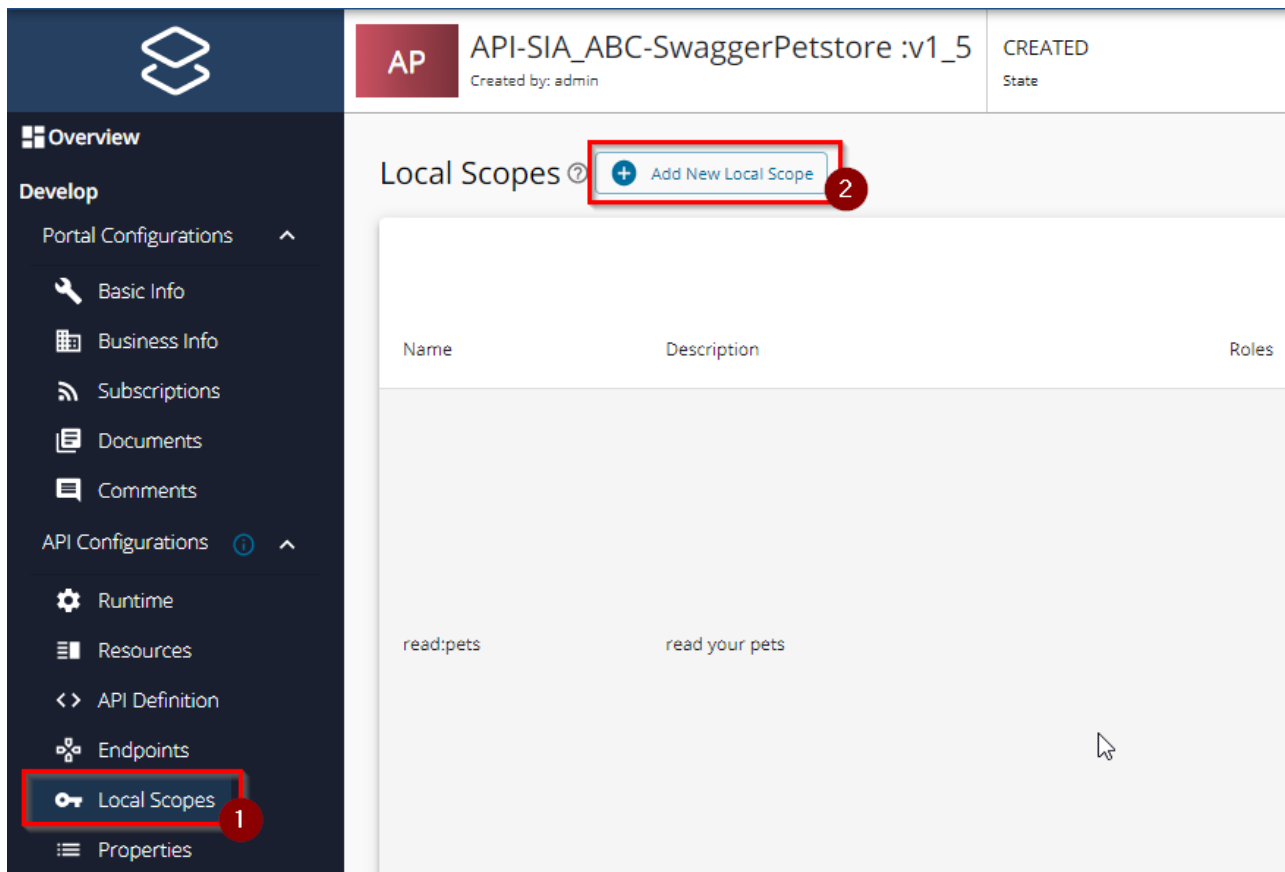
autoritātes identifikatorā un servisa (API) versijā tiek izmantota apakšsvītra, nevis punkts, jo WSO2 neļauj izmantot punktu scope nosaukumos.

Autoritātes identifikators ir iegūstams “Iestādes un struktūrvienības” klasifikatoru katalogā (skat. [27.attēls](#)[27.attēls](#)) kā īpašības visscore:authorityIdent vērtība.

4.4.3.5.1. Manuāli

Lai norādītu scopes manuāli ir jāveic šādas darbības:

1. Jāatver API publicētājā (API Publisher) vēlāmā servisa (API) rediģēšanas skata (skatīt, 28.attēlu) cilni Local Scopes.



28.attēls. Servisa rediģēšanas skata atvēršana

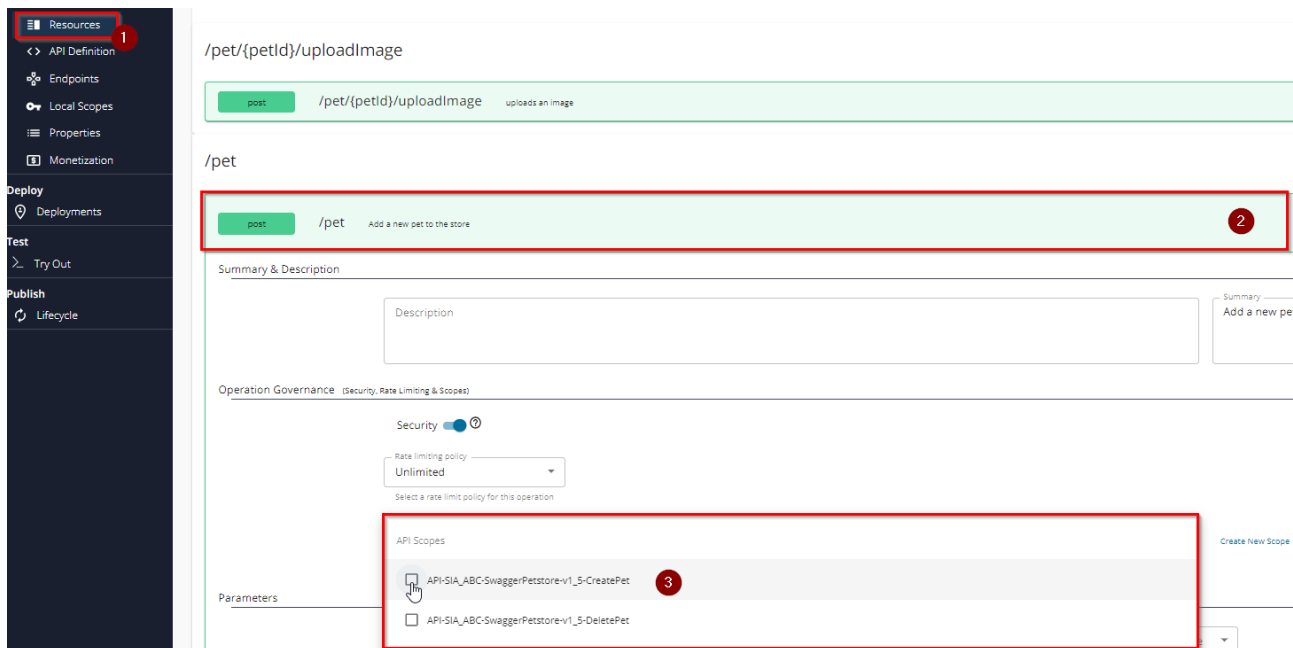
2. Cilnē Local Scopes vispirms ir jādefinē visas servisā izmantotās scopes, nospiežot pogu “Add New Local Scope”, skatīt 30.attēlu.
3. Javeic scope pievienošana, norādot tās identifikatoru un nosaukumu, skatīt 29.attēlu. Nosaukuma laukā jāievada tikai pašas atļaujas (scope) nosaukums, servisa nosaukums un versija tika pievienota automātiski, lai tas atbilstu scope formāta notācijai (skat. 4.4.3.2. 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

29.attēls. Scope datu aizpildīšana

4. 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 (30.attēls).



5. Jāsaglabā veiktās izmaiņas nospiežot pogu Save lapas lejasdaļā.

30.attēls. Scope pievienošana

4.4.3.5.2. 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_VDAA-Calculatation"
  },
  "host": "viss.gov.lv",
  "basePath": "/VISS.ApiManagement/Extensions/CalculationApi/stable",
  "schemes": [
    "https"
  ],
  "paths": {
    "/api/Calculator/Multiply": {
      "post": {
        ...
        "x-scope": "API-VIDM_VDAA-Calculatation-v1_0-Divide"
      }
    }
  }
},
```

```

    },
    "definitions": {
        ...
    },
    "x-wso2-security": {
        "apim": {
            "x-wso2-scopes": [
                {
                    "name": "Divide",
                    "description": "Dalīt skaitļus",
                    "key": "API-VIDM_VDAA-Calculatation-v1_0-Divide",
                    "roles": ""
                }
            ]
        }
    }
}

```

4.4.4. 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. 4.4.3.2. paragrāfu) datu ņēmēja Klienta lietojumam (Viss.lam vai IDS klientam) Viss.lamIdentity vai Viss.LvplIdentity saskarnē, izmantojot VDAA nodrošināto lietotāju (skat. 4.4. nodaļu).

Datu devējs Viss.lam vai IDS klientam var pievienot tikai tādus scope, kuru nosaukums satur datu devēja pārstāvētās autoritātes identifikatoru un kuri attiecīgajā brīdī eksistē API Pārvaldniekā (tiek veikta validācija).

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

4.4.5. 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) [11], ja abonentam šī servisa izmantošana nav atļauta.

Aizliegts bloķēt šo klienta lietojumu abonementus: APP-VIDM_VDAA-EservicePlatformRequestApi-v1_0, APP-VIDM_VDAA-VISS.ApiManagement.ISServiceInterceptor-v1_0

- Ja serviss (API) vai metode ir aizsargāta, tad servisa (API) publicētājs var:
 - Viss.lamIdentity vai Viss.LvplIdentity saskarnē, izmantojot VDAA nodrošināto lietotāju (skat. 4.4. nodaļu), noņemt pilnībā vai daļēji piesaistītos scopes Klienta lietojuma Viss.lam vai IDS klientam. (ietekmē konkrēto servisa (API) izmantotāju (datu ņēmēju));

Datu devējs IAM un IDS klientiem 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”.

4.4.6. 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 displays the 'API Properties' configuration page. The table has columns for 'Property Name', 'Property Value', and 'Visibility'. A new property 'transactionRequired' is being added with a value of 'false'. The 'Show in devportal' checkbox is checked. The 'Add' button is highlighted with a red box. The 'Save' button is also highlighted with a red box. The left sidebar shows the 'Properties' tab selected under 'Local Scopes'.

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.

4.5. 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 ar `Viss.lamIdentity.Server OAuth2` references vai JWT talonu autentificētiem lietotajiem.

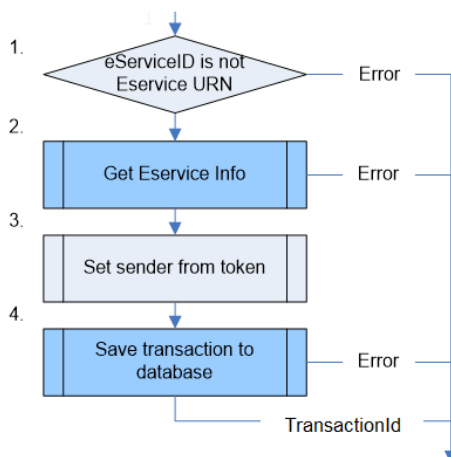
Ja transakciju ir nepieciešams uzsākt e-pakalpojuma ietvaros, tad nepieciešams izmantot `LvpContext.Request` funkcionalitāti skat. e-pakalpojuma izveides ceļveža programmētāja rokasgrāmatu [16]

4.5.1. 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 VDAA un tā piešķirs konkrētai datu apmaiņai paredzētu identifikatoru. Metode atgriež datu apmaiņas instances (transakcijas) URN identifikatoru (skat.

Error! Reference source not found..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.



31.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.

4.5.1.1. Piemēri ar datu nodošanu body daļā

Adrese

POST /transactions

HEADER parametri

- Authorization – *Viss.IamIdentity.Server Bearer OAuth2* (JWT vai references) talons.
- 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 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
```

4.5.1.2. Piemēri ar datu nodošanu query daļā

Adrese

POST /transactions/

HEADER parametri

- Authorization - *Viss.lamIdentity.Server Bearer OAuth2 (JWT vai references)* talons.
- Accept – vēlamais atbildes formāts, serviss nodrošina application/json, application/xml un text/plain formātus.
- Accept-Language – vēlāmā atbildes valoda, noklusētā vērtība en, serviss nodrošina lv, en valodas.

PATH parametri

Nav.

QUERY parametri

- eServiceId - datu apmaiņas procesa vai e-paklpojuma identifikators (*URN*).
- api-version – servisa versija, lai nodrošinātu datu apstradi 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 (*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
```

4.5.2. Transakcijas apturēšanas metode

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

Transakcija tiek izbeigta automātiski dienu pēc tās uzsākšanas.

4.5.2.1. Piemēri ar datu nodošanu body daļā

Adrese

DELETE /transactions

HEADER parametri

- Authorization - *Viss.lamIdentity.Server Bearer OAuth2 (JWT vai references)* talons.
- Accept – vēlamais atbildes formāts, serviss nodrošina application/json, application/xml un text/plain formātus.
- Accept-Language – vēlāmā atbildes valoda, noklusētā vērtība en, serviss nodrošina lv, en 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
```

4.5.2.2. Piemēri ar datu nodošanu query daļā

Adrese

DELETE /transactions

HEADER parametri

- Authorization - *Viss.lamIdentity.Server Bearer OAuth2 (JWT vai references)* talons.
- 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 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></EndTransactio
nResponse>
```

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

1
```

4.5.3. Transakcijas statusa metode

Metode tiek izsaukta, kad nepieciešams parbaudīt transakcijas statusu.

Adrese

GET /transactions/info

HEADER parametri

- Authorization - Viss.lamIdentity.Server Bearer OAuth2 (JWT vai references) talons.
- 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 valodas.

PATH parametri

Nav.

QUERY parametri

Nav.

BODY parametri

Nav.

Piemērs

```
GET /transactions/info 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
```

4.6. API izsaukumu piemēri

Piemēri demonstrē iespēju iegūt *Viss.lamIdentity.Server* OAuth2 drošības talonu un WSO2 publicēta API servisa izsaukumu. Piemēri ir vienkāršoti, lai uzskatāmi demonstrētu API pārvaldnieka un

Viss.*IamIdentity.Server* 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 [16].

4.6.1. Priekšnosacījumi

- 1) VDAA uzstāda atbilstošās vidēs CalculationApi REST servisu izmantojot Viss.ApiManagement.Extensions.CalculationApi.Setup.msi
- 2) VDAA reģistrē API publicētājā (API Publisher) CalculationApi (REST) un CalculationDataSync (esošais IS serviss URN:IVIS:100001:ISS-SIA.ABC-CalculationDataSync-v1-0) servisu saskaņā ar 4.4.3. sadaļas vadlīnijām;
- 3) Reģistrēt jaunu Klienta lietojumu Izstrādātāju portālā (Developer portal) saskaņā ar 4.3.1.3. paragrāfa vadlīnijām;
- 4) Izveidot abonementu starp reģistrēto Klienta lietojumu un servisiem CalculationApi un CalculationDataSync saskaņā ar 4.3.1.4. paragrāfa vadlīnijām;
- 5) Izveidot Viss.*IamIdentity.Server* lietotāju saskaņā ar 4.3.1.1. paragrāfa vadlīnijām.

4.6.2. Izsaukumu Curl piemēri

Drošības talonu var pieprasīt no šādas adreses VDAA testa vidē:

- <https://viss-portal-test.vraa.gov.lv/IamIdentity.Server/connect/token>

Drošības talonu var pieprasīt no šādas adreses VDAA produkcijas vidē:

- <https://viss-portal.viss.gov.lv/IamIdentity.Server/connect/token>

Pieprasījuma piemērs:

```
curl --location 'https://viss-portal-test.vraa.gov.lv/IamIdentity.Server/connect/token' --header 'Content-Type: application/x-www-form-urlencoded' --data-urlencode 'client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer' --data-urlencode 'client_assertion=gVAtutqlizB0kxrogMvdxYwkrH6ea7I5z9PeQWjTfwjcos3Noqdrj-wF-E4Ult_oZxrxVxm5WJspDXQr5tRA72ldN6RdrASiFlsC9EaUOhxNLsLV5DOSEjbMBwVSbosAfrpUyfSIJfnttarta68yzQKHFJhdrQXNyZvq-LePWey3HpecF0ELYVfSpv7YfWCUhIiB_Xp8W_Dj_HmUCXT5LtNX4DhQ5v9L0-_pJEj8wR60Mp5XMwEWJ6V3l6Aw0eS6fhDeBA204GTzy_DY1jccGBk6C' --data-urlencode 'grant_type=client_credentials'
```

4.6.2.1. OAuth2 drošības talona pieprasījums bez scope aizpildīšanas

Piemērs OAuth2 drošības talona atsauces pieprasīšanai no Viss.*IamIdentity.Server* bez scope norādīšanas. Šāds drošības talons derīgs pieprasījumiem kuros izmantotie servisi nav aizsargāti ar scope.

```
curl --location <TOKEN_ENDPOINT> --header 'Content-Type: application/x-www-form-urlencoded' --data-urlencode 'client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer' --data-urlencode 'client_assertion=<ASSERTION>' --data-urlencode 'grant_type=client_credentials'
```

Parametri:

- TOKEN_ENDPOINT - *Viss.lamIdentity.Server* OAuth2 drošības talona pieprasījuma adrese;
- ASSERTION – Vērtība vedojas saskaņā ar [20] specifikāciju. Subject (sub) vērtībā izmantot izveidotā Klienta lietojuma identifikatoru, skatīt 4.3.1.3, bet aud norādīt TOKEN_ENDPOINT parametra vērtību;

Drošības talonu var pieprasīt no šādas adreses VDAA testa vidē:

- <https://viss-portal-test.vraa.gov.lv/lamIdentity.Server/connect/token>

Drošības talonu var pieprasīt no šādas adreses VDAA produkcijas vidē:

- <https://viss-portal.viss.gov.lv/lamIdentity.Server/connect/token>

Pieprasījuma piemērs:

```
curl --location 'https://viss-portal-test.vraa.gov.lv/IamIdentity.Server/connect/token' --header 'Content-Type: application/x-www-form-urlencoded' --data-urlencode 'client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer' --data-urlencode 'client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1YyI6WyJNSU1GcHpDQ0E0K2dBd0lCQWdJS1BrNlg3UUFBQUFBQ0pUQU5CZ2txaGtpRzl3MEJBUXNGQURBd01STXdfUVVlLQlpJbWlaUHlMRlFCRlJZRFlXSmpNUmt3RndZRFZRUURFeEJoWW1NdFZ6SXdnRGhGVmtGTUxVTkJKQjRYRFRJeU1URkdNVEE1TWpZeE9wb1hEVEkzTVRFd01UQTVNell4T1Zvd1dERU1NQW9HQTFVRUNoTURRVUpETVJfFd0R3WU RWUVFMRXdoRVJWXRWbEpCUVRFMU1ETUDBMVVFXhNc1RIWmdMa1Z6WlhKMmFXTMxVR3hoZEdadmNtMH VRbuZqYTJWdVpDNU9iM1JwWmlsal1YUnBiMjVCY0drd2dnR2lnQTBHq1NxR1NJYjNEUUVQCQVFVQUE0SU Jqd0F3Z2dHS0FvSUJnUUNuS293eG5qN1RvL2ZFrfZ5WHYvTGwraGxCShpzTUtUSjr0MmRwaWJMZEpxST ZmR01jWU1UTE0eVBnYTB5al dmc2ZqMzhoVtJNRjRhMmRTTnVpbFBnSVlay3B4N2N6SkxFNETXSGdnRm tqK2g5eCtON1ZzUDBPYm5JUzFPT1JyUmVOK2RWclExekZFetZtLythMEZ1bm9QZXpDM0JEQ0ROWFBuYT hqMDZadDR3REpzcGs2bGlEZC9VbTRReXBpMkVnYjg3dTE3N2U3N2lFMmU4V0VpNENRbDRJcVZ1MVFGNV JEdURDOWxXay96bHBuZzBqV2duOCTacnU4V0lLSmxMcEFYREMyQVB2OXVIRkn3Y2F1Smc2MlJZWnlkU2 FrbTlTNWJUbtmjTWVPalRvbGt0Q2pWVDFFaFZHcjddcmxDME85dkU5N1I2Y09zSjByOU16RUN5TFNpN2 hxNEY1d0w5eHRkcU44UWNDDWU9admUxQ3gzY3BoODdlMG5aM0ZzWGJzYkY5SjZZV2VvUVRlATZvdFgwZ0 JINDE1MitEZVhBQmRaUnZ3UC9BM0NvdK5QT1JZzk5JUld3RkpQcDNRTGhZbzQxbngrejdtUVCWEZxUG 9aWDZjNzZuM0QxMVltWXhPNKx6Um9SRgtSMULPdFJUB0RKZHNyRHZGb1lGRkt3c0NBd0VBQWFPPQ0FSa3 dnZ0VWTUE0R0EXVWRED0VCL3dRRUF3SUZvREFkQmdOVkhRNEVGZ1FVY1IyQXJjUmFyd2ZReHFfaUnZTWD ZWSXhjVvQ0d0h3WURWUjBqQkJud0ZvQVU4azhVeVZ0WDRWaEpLNnN2MFZWY3dmddGwxVE13VFFZRFZSMG ZCRVl3UkRCQ29FQ2dQb1k4YUhSMGNEb3ZMMkZpWTNOemJ5NWZhZbU56YjJaMGQyRnlaUzVzZGk5RFpYSj BSVzV5YjJ4c0wyRmlZeTFYTWPbd09FVLdRVXd0UTBFdVvkzSnNNR1lHQ0nzR0FRVUZCd0VCQkZvd1dEQl dCZ2dyQmdFrkRJRY3dBblpLYUhSMGNEb3ZMMkZpWTNOemJ5NWZhZbU56YjJaMGQyRnlaUzVzZGk5RFpYSj BSVzV5YjJ4c0wxY3lNREE0UlhaaGJDNDWhZbU5mWVdKakxWY3lNREE0UlZaQlRDMURRUzVqY25Rd0RBWU RWUjBUQVFIL0JBSXdbREFOQmdrcWhraUc5dzBCQVfZrKFBT0NBZ0VBalJOcllVZ2lXZklLc1pPT0FYen FzSlJCuuDzSuH0VGnuujV5StleOSS5cvRza0lJY1p2Y3V5UzcxTXZhdklRUmr1RloleHM3WFYraXdsOE ZPZWtabURkrZzK2UyanFqdEVleVluVktuZUw5emViRzM5MUeWeERxeTztUVNhc2ZPUU82QjRUUXNRb3 dlejJUSyt0btZkQnJQVEVBK3hZVGdQRnZfbnZqV3NYK1JVcfUzaWN1ZzlUdTBMjdQN2lYdl0vQkZNaV E5S1NHylA1SWRDou5IEvZuMHLRLnhDUolaHF0beTmqXlJZmlleGV5TUFuWituYmZDWHIzSHlxZVVjQl Fldlo0ekdubVZlQ2prQXo0T04wTVlpSmMrNHhLUhXT2ZFRgp0WE5LeGRDMXg3bk85RuTDsnVvUTzhL1 NJZWRURThGeJR4bzZRYytqY2lSVXRZVWNUbGpxVfVZVZlZwTFVDRVdXTDNHVGd0WmVaZEhCSzVaZvDRYj ZxTWdhQXLEbmsxdDV1UWZRRGHaeZzqmFhJU1BiOG4ru2Q4c0t0VFZSRHZVTWN6SnV6SDZ6alVOVudRen JiaDJvcjJQTWdXZjFfxKzfiekcydHZQBtNvOXRDb082atVTRU1KWxBWukFvRW9VVEezQ0xKc09IZ3lqak 9sQXFpVk84ZCsVWTdpemFOV1BZAwwRYUlNZVcvWTzxewJ0R1Y3T1Vhb2ZpetLDVzm0VkdOSGRpbNnQd3 ZobldYQkclanlXWU0wrW9lMFBEbzZZNktkmJJEYjhpckVCSG5sseZYOHVISG8xdWI2Q3BBVXcrU0l6Mn prV0x0Wkx2N2ZJOeh6MWRWYVZIcklvbHhTR0o1TTN4NFR3ZEJVT1lWbTvMv1hyS3o3Vy9lNmhdQ0Q0an RoWDA9Il19.eyJzdWIiOiJlcM46ZGV2LXZyYWE6d3NvMi1jbGllbnQtand0IiwianRpIjoInjk2NTgOM DgtNmJmNi00YTtklLTg3MmEtZjU3NDU5MTRiMTg5IiwiaXNzIjoidXJuOmRldi12cmFhOnZzbzItY2xpZ W50LWp3dCIscImF1ZCI6Imh0dHBzOi8vdmlzcylwb3J0YWwtZGV2LXZyYWEuYWJjc29mdHdhcmUubHYv
```

Dokumenta kods: VDAA-VDL-D APM	Datums: 13.11.2025.	Versija: 2.30
Datne: VDAA-VDL-D APM v2.30VDAA-VDL-D APM.DOCX	Izstrādāja: J.Kornilenko	Lpp.: 49 (66)

```
WfTSWRlbnRpdHkuU2VydmVyL2NvbW5lY3QvdG9rZW4iLCJleHAiOiIxNzQwNjY1NjkkxiwibmJmIjoim
Tc0MDY2MjA5MSJ9.WhpNA4fQ2NL0n2Iz8gQ-ZlrsfeoU0r10Agz-
omwE7AALqUamU7wsfnteqqZ24uee4sluBoQx3GJJkam7Ip_VeNO2sn0IKuzElCnzNbgY8zKeT6LHe0iZ
chzZ-99FkPEAbC58oeFuc6J0914hVb9mH42mwWlQzfUCIh_EjG_d6St4DOMTdnYxLHqCB5SNWvF-
lXHwH_NGLg8EuKklBS_fxmzkzuiS982EwtBwT2BZgVAtutqlizB0kxrogMvdxyWkrH6ea7I5z9PeQWjTf
wjcos3Noqdrj-wF-
E4Ult_oZxrXVxm5WJspDXQr5tRA72ldN6RdrASiFlsC9EaUOhxNLsLV5DOSEjBmBwVSbosAfrpUyfSIJ
fnttarta68yzQKHFJhdrQXNyZvq-
LePWey3HpecF0ELyVfSpv7YfWCUhIiB_Xp8W_Dj_HmUCXT5LtNX4DhQ5v9L0-
_pJEjt8wR60Mp5XMwEWJ6V3l6Aw0eS6fhDeBA204GTzy_DY1jccGBk6C' --data-urlencode
'grant_type=client_credentials'
```

Atbildes piemērs:

```
{ "access_token": "urn:uuid:ddf7a566-afff-4a58-9a30-2f0859165d5c", "expires_in": 3600, "token_type": "Bearer" }
```

Atribūts `access_token` satur drošības talona atsauci, kas būs jāizmanto servisu izsaukšanai. Detalizētu informāciju par OAuth2 drošības talonu skatīt [17].

4.6.2.2. OAuth2 drošības talona pieprasījums ar norādītiem scope

Piemērs OAuth2 references talona pieprasīšanai no *Viss.IamIdentity.Server* ar norādītu atļauju/scope. Pieprasījumā ir iespējams norādīt vairākus scope atdalot tos ar atstarpi. Atbildē tiks atgriezti tikai tie scope uz kuri būs piesaistīti klientam *Viss.IamIdentity* saskarnē.

```
curl --location <TOKEN_ENDPOINT>
--header 'Content-Type: application/x-www-form-urlencoded'
--data-urlencode 'client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer'
--data-urlencode 'client_assertion=<ASSERTION>'
--data-urlencode 'grant_type=client_credentials'
--data-urlencode 'scope=SCOPE'
```

Parametri:

- `TOKEN_ENDPOINT` - *Viss.IamIdentity.Server* OAuth2 drošības talona pieprasījuma adrese;
- `ASSERTION` – Vērtība vedojas saskaņā ar [20] specifikāciju. Subject (sub) vērtībā izmantot izveidotā Klienta lietojuma identifikatoru, skatīt 4.3.1.3, bet aud norādīt `TOKEN_ENDPOINT` parametra vērtību;
- `SCOPE` – Saraksts ar scope/atļaujām, kas nepieciešamas, lai ar drošības talonu izsauktu vēlamos servisu.

Drošības talonu var pieprasīt no šādas adreses VDAA testa vidē:

- <https://viss-portal-test.vraa.gov.lv/IamIdentity.Server/connect/token>

Drošības talonu var pieprasīt no šādas adreses VDAA produkcijas vidē:

- <https://viss-portal.viss.gov.lv/IamIdentity.Server/connect/token>

Pieprasījuma piemērs:

```
curl --location 'https://viss-portal-test.vraa.gov.lv/IamIdentity.Server/connect/token' --header 'Content-Type: application/x-www-form-urlencoded' --data-urlencode 'client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer' --data-urlencode 'client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1YyI6WyJNSU1GcHpDQ0E0K2
```

dBd01CQWdJS1BrNlg3UUFbQUFBQ0pUQU5CZ2txaGtpRzl3MEJBUXNGQURBd01STXdFUVlLQ1pJbWlaUH
lMR1FCR1JZRfLXSmPNUmt3RndZRFZRURFeEJoWw1NdFZ6SXdnRGhGVmtGTUxVtKJNQjRYRFRJeU1URX
dnVEE1TWpZeE9Wb1hEVEkzTVRFd01UQTVNell4T1Zvd1dERU1NQW9HQTFVRUNoTURRVUpETVJFd0R3WU
RWUVFMRXdoRVJWWXRWbEpCUVRFMU1ETUdBMVVFQXhNclRIWndMa1Z6WlhKMmFXTmxVR3hoZEdadmNtMH
VRbUZqYtJWdVpDNU9iM1JwWm1sallYUnBiMjVCY0drd2dnR21NQTBHQ1NxR1NJYjNEUUVcQVFVQUE0SU
Jqd0F3Z2dHS0FvSUJnUUNuS293eG5qN1RvL2ZFRzZ5WHYvTGwraGxCShpzTUtUSjR0MmRwaWJMZEpxST
ZmR01jWU11UTE0eVBnYTB5aldmc2ZqMzhoVTJNRjRhMmRTTnVpbFBnSVlaY3B4N2N6SkxFNtXSgdnRm
tqK2g5eCtOnLZzUDBPYm5JUzFPT1JyUmVOK2RWclExekZFeTztLythMEZ1bm9QZXPDM0JEQ0ROWFBuYT
hqMDZadDR3REpzcGs2bG1EZC9VbTRReXBpMkVnYjg3dTE3N2U3N2lFMmU4V0VpNENRbDRJcVZ1MVFGNV
JEdURDOWxXay96bHBuZzBqV2duOCtactnU4V01LSmxMcEFYREMYQVB2OXVIRkN3Y2F1Smc2MlJZWnlkU2
FrbTlTNWJUubmtjTWVPalRvbGt0Q2pWVDFFaFZHcjDcmxDME85dkU5N1I2Y09zSjByOU16RUN5TFNpN2
hxNEY1d0w5eHRkcU44UWNDWU9admUxQ3gzY3BoODdlMG5aM0ZzWGJzYkY5SjZVZ2VvUVRiATZvdFgwZ0
JINDE1MitEZVhBQmRaUnZ3UC9BM0NvdK5QT1JZZk5JUld3RkpQcDNRTGhZbzQxbngrejddTUVCWEZxUG
9aWDZjNzZuM0QxMVltWXhPNkx6Um9SRGtSMU1PdFJU0RKZHNyRHZGb1lGRkt3c0NBd0VBQWFPQ0FSa3
dnZ0VWTUE0R0ExVWRED0VCL3dRRUF3SUZvREFkQmdOVkhRNEVGZ1FVY1IyYQXJjUmFyd2ZReHFaUnZTWD
ZWSXhjVQ0d0h3WURWUjBqQkjd0ZvQVU4azhVeVZ0WDRWaEpLNN2MFZWY3dmdGwxVEl3VFFZRFZSMG
ZCRV13UkRCQ29FQ2dQb1k4YUhSMGNEb3ZMMkZpWTNOemJ5NWhZbU56YjJaMGQYRnlaUzVzZGk5RFpYSj
BSVzV5YjJ4c0wyRmlZeTFyTWpBd09FVldRVXd0UTBfdVzkSnNNR1lHQ0NzR0FRVUZCd0VCQkZvd1dEQl
dCZ2dyQmdFRkJRy3dBb1pLYUhSMGNEb3ZMMkZpWTNOemJ5NWhZbU56YjJaMGQYRnlaUzVzZGk5RFpYSj
BSVzV5YjJ4c0wxY3lNREE0UlhagJDNWhZbU5mWVdKakxWY3lNREE0U1ZaQlRDMURRUzVqY25Rd0RBWU
RWUjBUQVFIL0JBSXdbREFOQmdrcWhraUc5dzBCQVZrRkFBT0NBZ0VbalJOcllVZ2lXZklLclpPT0FYen
FzS1JCUUdzSUh0VGNUUjV5STlEOSs5cVRza0lJY1p2Y3V5UzcwTXZhdklRUmR1Rlo1eHM3WFYraXdsOE
ZPZWtabURkRzZkZ2UyanFqdEVleVluVktuZUw5emViRzM5MUEweERxeTztUVNhc2ZPUU82QjRUUXNRb3
dleJjJUSyt0bTzKqNjQVEVBK3hZVGdQrnZfbnZqV3NYK1JVcFUzaWN1ZzlUdTBZMjdQN2lYd1ovQkZNaV
E5S1NHY1A1SWRDOU5IeVZuMHRLRnhDUolaHF0bEtMQXlJZmlleGV5TUfUWituYmZDWHIzShlxZVVjQ1
Flldlo0ekdubVz1Q2pRQXo0T04wTVlpSmMrNHhLZUhXT2ZFRGp0WE5LeGRDMXg3bk85RutDSnVvUTZhL1
NJZWRURThGejR4bzZRYytqY2lSVXRZVWNUbGpxVFVZ1ZwTFVDRVdXTDNHVgd0WmVaZehCSzVaZvDRYj
ZxTWdhQXlEbmsxdDV1UWZRRGhEazZqMFhJU1BiOG4rU2Q4c0t0VFZSRHZVTWN6SnV6SDZ6alVOVUdRen
JiaDJVcjJQTwDXzjFkKzFiekcydHZQbTNvOXRD082aTVTRU1KWXBWUkFvRW9VVEEzQ0xKc09IZ3lqak
9sQXFPV84ZCvWTdpemFOV1BZaWRRYU1NZVcVWTZxewJ0R1Y3T1Vhb2Zpet1DVz0M0VkdOSGRFPbnNQd3
ZoblDYQkclanlXWU0wRW9lMFBEBzZZNktkMjJEYjhpckVCSG5sSEZYOHVVISG8xdWI2Q3BBVXcrU016Mn
pRV0x0Wkx2N2ZJOEH6MWRWYVZICK1vbHhTR0o1TTN4NFR3ZEJVT11WbTVmV1hyS3o3Vy9lNmhqd0Q0an
RoWDA9I119.eyJzdWIiOiJlcm46ZGV2LXZyYWE6d3NvM1ljBGlbnQtand0IiwianRpIjoianNk2NTg0M
DgtNmJmNi00YTklLTg3MmEtZjU3NDU5MTRiMTg5IiwiaXNzIjoiaXNzIiwiaWF0IjoiMTY2MDYyMjMjA5MSJ9.
WhpNA4fQ2NL0n2Iz8gQ-ZlrsfeoU0r10Agz-
omwE7AALqUamU7wsfnteqqZ24uee4sluBoQx3GJJKam7Ip_VeNO2sn0IKuzElCnzNbgY8zKeT6LHe0iz
chzZ-99FkPEAbC58oeFuc6J0914hVb9mH42mwWlQzfUCIh_EjG_d6St4DOMTdnYxLHqCB5SNWvF-
lXHwH_NGLg8EuKklBS_fxmzkzuiS982EwtBwT2BZgVAtutqlizB0kxrogMvdxyWkrH6ea7I5z9PeQWjTf
wjcos3Noqdrj-wF-
E4Ult_oZxrXVxm5WJspDXQr5tRA72ldN6RdrASIfLsC9EaUOhxNLsLV5DOSEjBMBwVSbosAfrpUyfSIJ
fnttarta68yzQKHFJhdrQXNyZvq-
LePWey3HpecF0ELYVfSpv7YfWCUhIiB_Xp8W_Dj_HmUCXT5LtNX4DhQ5v9L0-
_pJEj8wR60Mp5XMwEWJ6V3l6Aw0eS6fhDeBA204GTzy_DY1jccGBk6C' --data-urlencode
'grant_type=client_credentials' --data-urlencode 'scope=API-VDAA-Viss.Api-v1_0-
Scope1

Atbildes piemērs:

```
{"access_token":"urn:uuid:f5c74f49-fcbf-4f94-b8f6-892292e448c5","expires_in":3599,"token_type":"bearer"}
```

Atribūts `access_token` satur drošības talona atsauci, kas būs jāizmanto servisu izsaukšanai. Detalizētu informāciju par OAuth2 drošības talonu skatīt [17].

4.6.2.3. Datu izgūšana par drošības talonu

Primāri datu apmaiņās tiek izmantoti IAM JWT taloni, ja nepieciešams izmantot OAuth2.0 references talonu tad šo scenāriju ir atsevišķi jāaskaņo ar VDAA.

JWT talons kodētā veidā satur autentifikācijas procesa laikā talonā iekļautos datus. Saņemot JWT talonu ir jāveic tā validāciju saskaņā ar specifikāciju [18], validācijas veikšanai var tikt izmantotas dažādas trešo pušu bibliotēkas. Validācijas process tipiski iekļauj šādus soļus:

- Talona atkodēšana;
- Paraksta pārbaude;
- Derīguma termiņa (exp) un spēkā stāšanās termiņa (nbf) pārbaude;
- Talona izsniedzēja (iss) pārbaude – rekomendējams pret konfigurējamu vērtību;
- Talona saņēmēja (aud) pārbaude – rekomendējams pret konfigurējamu vērtību;

Ja nepieciešams biznesa servisā (API) izgūt datus par lietotāju, kas veic izsaukumu ar OAuth2.0 references piekļuves (access) talonu, to ir iespējams veikt, izsaucot *Viss.IamIdentity.Server* ieejas punktu /connect/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 autentificētu Latvija.lv lietotāju, kas veic API izsaukumu no e-pakalpojuma iespējams iegūt šādus datus (izmantojot *Viss.IamIdentity.Server* introspect izsaukumu, e-pakalpojumu ietvars šajā scenārijā nodrošina IDS talona apmaiņu pret *Viss.IamIdentity.Server* talonu saglabājot oriģinālos lietotāja datus) atkarībā no tā kādai auditorijai talona pieprasītājs pieprasa talonu:

- Fiziskas personas access talonā pēc noklusējuma būs iekļautas šādas pielaides (claims) no sākotnēji izsniegtā IDS talona:
 - given_name – vārds;
 - family_name – uzvārds;
 - ppid – personas kods/kodi (teksta virkne vai masīvs ar teksta virknēm);
 - citizenQAALevel – autentifikācijas sniedzēja kvalifikācijas līmenis;
 - ppid_hash – lietotāja vienotais identifikators;
 - scp – atļauju saraksts API pārvaldnieka servisu izsaukšanai;
- Juridiskas personas access talonā pēc noklusējuma būs iekļautas šādas pielaides (claims) no sākotnēji izsniegtā IDS talona:
 - given_name – vārds;
 - family_name – uzvārds;
 - ppid – personas kods/kodi (teksta virkne vai masīvs ar teksta virknēm);
 - ppid_hash – lietotāja vienotais identifikators;
 - citizenQAALevel – autentifikācijas sniedzēja kvalifikācijas līmenis.
 - legalentity – uzņēmuma reģistrācijas numurs;
 - legalentityname – uzņēmuma pilnais nosaukums;
 - legalentityposition – amats;
 - legalentityrepresentation – pārstāvniecības veids;
 - scp – atļauju saraksts API pārvaldnieka servisu izsaukšanai;
- Pilnvarotās personas access talonā pēc noklusējuma būs iekļautas šādas pielaides (claims) no sākotnēji izsniegtā IDS talona:
 - given_name – vārds;
 - family_name – uzvārds;

- ppid – pilnvaras saņēmēja personas kods/kodi (teksta virkne vai masīvs ar teksta virknēm);
 - ppid_hash – lietotāja vienotais identifikators;
 - citizenQAALevel – autentifikācijas sniedzēja kvalifikācijas līmenis;
 - grantor – pilnvaras devēja personas kods vai reģistrācijas numurs;
 - grantorname – pilnvaras devējs;
 - grantor_hash – pilnvaras devēja vienotais identifikators (ja pilnvaras devējs ir fiziska persona);
 - scp – atļauju saraksts API pārvaldnieka servisu izsaukšanai;
- **Gadījumos, ja lietotājs nomainīja personas kodu, tad IDS pielaidē “ppid” var tikt atgrieztas vairākas vērtības patvaļīgā secībā - masīvs ar teksta virknēm kurš satur veco personas kodu un aktuālo personas kodu. Ja personas kods nav mainīts vērtība tiek atgriezta kā teksta virkne Analógiski kā rfc7519 specifikācijā aprakstītajai “aud” pielaidei.**
 - **Pielaide “ppid_hash” satur iedzīvotāja vienoto identifikatoru (unikāla burtu ciparu virkne), identifikators nemainās pat ja lietotājs ir veicis personas koda maiņu. Tā var tikt izmantota personas koda vietā, lai noteiktu datu piederību autentificētajai personai. Ņemt vērā ka tā ir iekļauta tikai IDS izsniegtajos talonos. Tā nevar tikt izmantota, lai identificētu pilnvaras devēju vai pārstāvēto uzņēmumu.**
 - **Pielaide “grantor_hash” fiziskas personas pilnvarotā gadījumā satur pilnvaras devēja vienoto identifikatoru (unikāla burtu ciparu virkne), identifikators nemainās pat ja pilnvaras devējs ir veicis personas koda maiņu. Tā var tikt izmantota personas koda vietā, lai noteiktu datu piederību autentificētajai personai. Ņemt vērā ka tā ir iekļauta tikai IDS izsniegtajos talonos.**
 - **IDS talona apmaiņas pret Viss.lamIdentity.Server talonu procesa ietvaros tiek aktualizētas vairāku pielaižu vērtības atbilstoši tam klientam ar kuru tiek veikta talonu apmaiņa, kā rezultātā sākotnējās IDS izbeigtās vērtības pēc apmaiņas vairs nav pieejamas vismaz šādām pielaidēm: “iss”, “nbf”, “iat”, “exp”, “aud”, “scope”, “amr”, “client_id”, “sub”, “auth_time”, “idp”, “jti”.**
 - **Introspect izsaukums ir paredzēts tikai OAuth2.0 references talona satura iegūšanai, kas praksē netiks izmantoti.**

API pārvaldniekā reģistrētajiem Klienta lietojumiem (application) pēc noklusējuma Viss.lamIdentity.Server izsniegtajā access talonā būs pievienotas šādas pielaides (claims), šajā scenārijā talonu pieprasa ar iestādes izsniegtu sertifikātu un tas tiek izdots šai iestādei (tehniskais lietotājs):

- authority_id – iestādes reģistrācijas numurs;
- authority_name – iestādes nosaukums.

Viss.lamIdentity.Server introspect izsaukums:

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

Parametri:

- INTROSPECT_ENDPOINT - Viss.lamIdentity.Server OAuth2 drošības talona informācijas izgūšanas adrese;
- CLIENTID – Viss.lam API resursa identifikators;
- SECRET - Viss.lam API resursa atslēga;
- TOKEN – Viss.lam drošības talona atsauce

Drošības talonu var pārbaudīt izsaucot šādu adresi VDAA testa vidē:

- <https://viss-portal-test.vraa.gov.lv/IamIdentity.Server/connect/introspect>;

Drošības talonu var pārbaudīt izsaucot šādu adresi VDAA produkcijas vidē:

- <https://viss-portal.viss.gov.lv/IamIdentity.Server/connect/introspect>;

Pieprasījuma piemērs:

```
curl -X POST -v https://viss-portal-test.vraa.gov.lv/IamIdentity.Server/connect/introspect -u "0f2053f6-95bf-401d-8e01-0f8a49537b43:BqNTJgY5dtmcKi7zoRBZ1/hRXj0a3mSPvyrUtlx+cDM=" --data-urlencode "token= urn:uuid:f5c74f49-fcbf-4f94-b8f6-892292e448c5"
```

Atbildes piemērs, ja pieprasījumā tika padots iestādes (tehniskā lietotāja) pieprasīts talons :

```
{
  "iss": "urn:DEV-VDAA:VISS.IAM",
  "active": true,
  "aud": [
    "urn:DEV-VDAA:VISS.IAM/resources"
  ],
  "exp": 1571747794,
  "authority_id": "90001733697",
  "authority_name": "Valsts reģionālās attīstības aģentūra",
  "nbf": 1571733394,
  "sub": "04dcd9fcc1494a314f43483b6a3a45d70673507e5f",
  "scp": [
    "CreateNewObject"
  ]
}
```

Atbildes piemērs, ja pieprasījumā tika padots IAM talons ar fiziskas personas datiem:

```
{
  "iss": "urn:DEV-VRAA:VISS.IAM",
  "nbf": 1746680549,
  "iat": 1746680549,
  "exp": 1746694949,
  "aud": [
    "urn:viss.apimanagement",
    "urn:DEV-VRAA:VISS.IAM/resources"
  ],
  "client_id": "urn:abc.samples:app:core-ref",
  "sub": "016D187118E1DDC0A9C2F9379A9CCF247BC81A5621",
  "auth_time": 1746680549,
  "idp": "local",
  "amr": "urn:ietf:params:oauth:grant-type:token-exchange",
  "scp": [
    "API-SIA.ABC-Viss.Delegation.Api-v1_0-Cancel",
    "API-SIA.ABC-Viss.Delegation.Api-v1_0-CheckAccess"
  ],
  "ppid": "01018512345",
}
```

```
{
  "given_name": "Jānis",
  "family_name": "Bērziņš",
  "citizenQAALevel": 4,
  "ppid_hash": "08WuFoeLPb/1ZC2vW8F4E/TdVFX+s8OBkOEf5HuKpPk=",
  "jti": "B6E6A488D686EE7968653A27315CC132",
  "active": true,
  "scope": "urn:viss.apimanagement-apicall"
}
```

Atbildes piemērs, ja pieprasījumā tika padots IAM talons ar juridiskās personas datiem:

```
{
  "iss": "urn:DEV-VRAA:VISS.IAM",
  "nbf": 1746706959,
  "iat": 1746706959,
  "exp": 1746721359,
  "aud": [
    "urn:viss.apimanagement",
    "urn:DEV-VRAA:VISS.IAM/resources"
  ],
  "client_id": "urn:abc.samples:app:core-ref",
  "sub": "03B3EBC4DB116BDA6FED8BE8BCEB5FB17FDBC1429E",
  "auth_time": 1746706959,
  "idp": "local",
  "amr": "urn:ietf:params:oauth:grant-type:token-exchange",
  "scp": [
    "API-SIA.ABC-Viss.Delegation.Api-v1_0-Cancel",
    "API-SIA.ABC-Viss.Delegation.Api-v1_0-CheckAccess"
  ],
  "ppid": "01018512345",
  "given_name": "Jānis",
  "family_name": "Bērziņš",
  "citizenQAALevel": 4,
  "ppid_hash": "08WuFoeLPb/1ZC2vW8F4E/TdVFX+s8OBkOEf5HuKpPk=",
  "legalentity": "40103303559",
  "legalentityname": "Akciju sabiedrība \"Citadele banka\"",
  "legalentityrepresentation": false,
  "legalentityposition": "Valdes loceklis",
  "jti": "41E251043B6D605CFBC784A59FFAECD4",
  "active": true,
  "scope": "urn:viss.apimanagement-apicall"
}
```

Atbildes piemērs, ja pieprasījumā tika padots IAM talons ar pilnvarotās personas datiem:

```
{
  "iss": "urn:DEV-VRAA:VISS.IAM",
```

```

"nbf": 1746707070,
"iat": 1746707070,
"exp": 1746721470,
"aud": [
  "urn:viss.apimanagement",
  "urn:DEV-VRAA:VISS.IAM/resources"
],
"client_id": "urn:abc.samples:app:core-ref",
"sub": "09ADE1C4DFFE36EF6F00C5040BF4AE0A056E7E8BAA",
"auth_time": 1746707070,
"idp": "local",
"amr": "urn:ietf:params:oauth:grant-type:token-exchange",
"scp": [
  "API-SIA.ABC-Viss.Delegation.Api-v1_0-Cancel",
  "API-SIA_ABC-Viss.Notification.Api-v1_0-LvpIdsSend"
],
"given_name": "Jānis",
"family_name": "Bērziņš",
"citizenQAALevel": 4,
"ppid_hash": "08WuFoeLPb/1ZC2vW8F4E/TdVFX+s8OBkOEf5HuKpPk=",
"grantor_hash": "1UhX/e+GD+XWu19D/PFDaVRMWG1eaDRgWDgm9BV1dw=",
"grantor": "02028012345",
"grantorname": "Anna Bērziņa",
"ppid": "01018512345",
"jti": "59E66D18541361972BDC646E9DD71DB8",
"active": true,
"scope": "urn:viss.apimanagement-apicall"
}

```

Detalizētu informāciju par OAuth2 drošības talonu skatīt [17].

4.6.2.4. 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 4.5.1.

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 '{"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 4.5.1.

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"
```

4.6.2.5. 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 jānorāda header atribūtā x-transactionId vai pieprasījuma struktūras <IVISRequest><Header><TransactionID> elementā, lai iegūtu skatīt 4.5.1.

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 jānorāda header atribūtā x-transactionId vai pieprasījuma struktūras <IVISRequest><Header><TransactionID> elementā, lai iegūtu skatīt 4.5.1.

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>
        </Header>
      </IVISRequest>
    </DefaultMethodSync>
  </s:Body>
</s:Envelope>
```

```

        <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>

```

4.6.3. .NET piemēri

4.6.3.1. SOAP un REST servisu izsaukšana un talona pieprasīšana ar sertifikātu.

Ja no backend nepieciešams izsaukt API pārvaldniekā reģistrētus servisu, jāizmanto *Viss.IamIdentity.Server* 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* – *Viss.IamIdentity.Server* OAuth2 drošības talonu servisa adrese;
- *introspectionAddress* – *Viss.IamIdentity.Server* OAuth2 talonu introspekcijas adrese;
- *clientId* - izveidotā Klienta lietojuma Consumer Key skatīt 4.3.1.3;
- *pathnameprivkeyfile* – pilns ceļš līdz sertifikātam, tam jābūt derīgam un reģistrētam *Viss.IamIdentity.Server* kā tehniskajam lietotājam.
- *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
    }
}

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;
}

```

32.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/TestISService/v1-0" xmlns:pers="http://ivis.eps.gov.lv/XMLSchema
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/TestISService/v1-0" xmlns:pers="http://ivis.eps.gov.lv/XMLSchema
s/100001/Person/v1-0" xmlns:ivis="http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0">
  <CalculationValue>5</CalculationValue>
</Result>

```

33.attēls. .NET piemēra izpildes rezultāts

Ja no frontend vai backend nepieciešams izsaukt API pārvaldniekā reģistrētus servisu, var izmantot Viss.IamIdentity.Server izsniegtu OAuth2 references talonu.

Lai nomainītu IAM klientam izsniegtā talona tipu var izmantot Viss.IamIdentity klientu adminstrēšanas saskarni – izmaiņu veikšanu jāpieprasa VDAA.

4.6.4. 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 *Viss.lamIdentity.Server* 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 – *Viss.lamIdentity.Server* OAuth2 drošības talonu servisa adrese;
- introspectionAddress – *Viss.lamIdentity.Server* OAuth2 talonu introspekcijas adrese;
- clientId - izveidotā Klienta lietojuma Consumer Key (bez urn:oauth2: prefiksa), skatīt 4.3.1.3;
- certificatePath – pilns ceļš līdz sertifikātam, tam jābūt derīgam un reģistrētam *Viss.lamIdentity.Server* kā tehniskajam lietotājam.
- 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`:

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());
```

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

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://visa.spa.gov.lv/XMISchemas/10000/TestISService/v1-0" xmlns:pers="http://visa.spa.gov.lv/XMISchemas/10000/Person/v1-0" xmlns:vis="http://visa.spa.gov.lv/XMISchemas/10000/IVIS/v1-0"><CalculationValue>30</CalculationValue></Result>
<Result xmlns="http://visa.spa.gov.lv/XMISchemas/10000/TestISService/v1-0" xmlns:pers="http://visa.spa.gov.lv/XMISchemas/10000/Person/v1-0" xmlns:vis="http://visa.spa.gov.lv/XMISchemas/10000/IVIS/v1-0"><CalculationValue>5</CalculationValue></Result>
```

35.attēls. Java piemēru izpildes rezultāts

4.6.5. 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 Viss.IamIdentity.Server 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 - Viss.IamIdentity.Server OAuth2 drošības talonu servisa adrese JWT talona pieprasīšanai ar sertifikātu;
- CERTIFICATE_CLIENT_ID – izveidota lietojuma Consumer Key;
- CERTIFICATE – sertifikāta nosaukums, sertifikātam jābūt *.pem formātā, to var konvertēt no *.pfx atkomentējot index.php datnē rindiņu:

```
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 - Viss.IamIdentity.Server OAuth2 talonu introspekcijas adrese;
- API_ID – izsaucama API identifikātors (bez urn:oauth2: prefiksa);
- API_SECRET – izsaucama API atslēga.

Lai piemēru darbinātu izmantojot sertifikāta autentifikāciju index.php datnē jāizmanto `getAccessTokenWithSertificate`:

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

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

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

\

Access token:

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ij0M2FZWFluWDdlbDFHdzVyeK5WUThOaXA3TSJ9.eyJpc3MiOiJodHRwOi8uZGyRUzI7IK_8zxCI4qR4dTojRSLhi8ijbd3xUupuHoDR6fJsCMC2T5leRc2ex1N98KyIVvuY9g45d0HD640qGFTrPmCiKyI8Jy889CUS-uEUqbtoKYCeWticQunG2Nk082Z84oeHvsz65vc8y9u_mtY_LfSsFqdShlCWgRPFgljB4F37x4Whbdon66iWH_xuUjSyzljjP_PsFs6iGSicmYB0M-d6cXguRSVvXWUcXT253tyyD-qJUUA7rlJLwMjTUM-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-44b1-8e31-2ee0b85748d</MessageID><MessageType>URN:IVIS:100001:XSD-Testing-TestISService-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/TestISService/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:

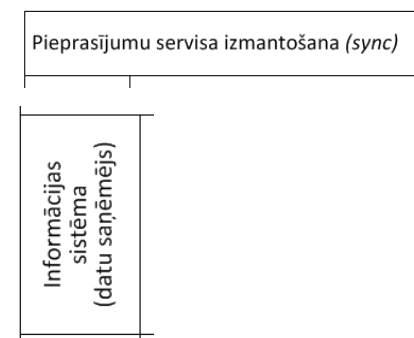
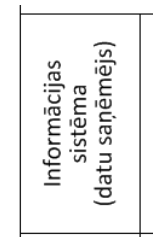
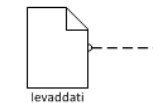
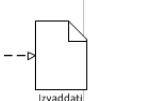
URN:IVIS:100001:EP-EP186-v1-0-TR-62894

37.attēls. PHP piemēra izpildes rezultāts

5. Pielikums. Biznesa procesu modelēšanas valodas notācija

4.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: 1. tam aplūkojamā procesa ietvaros, ir nosakāmi spēkā stāšanās nosacījumi (no procesa aspekta); 2. to producē darbības veicējs; 3. tas realizē stāvokļu pārejas; 4. tam stājoties spēkā, rodas: - nosacījums citas darbības izpildei; - resurss.
	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.
	Pārbaude	Procesam var būt vairāki tādi izvada punkti. Situācijas novērtēšana: 1. to uzpilda darbības veicējs; 2. tās rezultātā ir iespējami viens vai vairāki rezultāti (novērtējumi). 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.