VALSTS DIGITĀLĀS ATTĪSTĪBAS AĢENTŪRA

PROGRAMMĒTĀJA ROKASGRĀMATA

VDAA-PR-DTS

10.10.2024 versija 2.20

Rīgā 2024

Saturs

Attēlu saraksts

NO TABLE OF FIGURES ENTRIES FOUND.

1. Ievads

1.1. Dokumenta nolūks

Šis dokuments ir projekta „Valsts informācijas sistēmu savietotāja (VISS) un Vienotā valsts un pašvaldību pakalpojumu portāla www.latvija.lv pilnveidošana un uzturēšana" 1.2.7.Vienotā pieteikšanās moduļa papildinājumu nodevums.

Programmētāja rokasgrāmata ir domāta e-pakalpojumu un saistīto lietojumu izstrādātājiem, identifikācijas piegādātāju (*provider*) veidotājiem. Pirms šī dokumenta lasīšanas būtu vēlams iepazīties ar [1] un [2] dokumentiem.

Programmētāja rokasgrāmatas nolūks ir:

- izklāstīt PFAS.STS un vienotās pieteikšanās moduļa lietojumu kopējo aprakstu;
- aprakstīt drošā Web Servisa un tā klienta izstrādes procesu;
- aprakstīt bankas identifikācijas piegādātāja izstrādes procesu.

1.2. Darbības sfēra

Dokuments paredzēts lietošanai projekta izstrādes grupas ietvaros, kā arī visām tām personām, kuru darba pienākumi ir tieši saistīti ar projekta realizāciju.

1.3. Termini un saīsinājumi

Visi šajā dokumentā izmantotie termini un saīsinājumi ir apkopoti [3] dokumentā.

1.4. Saistītie dokumenti

Instrukcija ir lietojama kopā ar šādiem dokumentiem:

[1] A GUIDE TO CLAIMS-BASED IDENTITY AND ACCESS CONTROL. Authentication and Authorization for Services and the Web. Dominick Baier. Vittorio Bertocci. Keith Brown. Eugenio Pace. Matias Woloski.

[2] Microsoft Windows Identity Foundation (WIF) Whitepaper for Developers. Keith Brown. Sesha Mani.

[3] IVIS „Terminu un saīsinājumu indekss." (EPS-31/2005-IVIS-OLIMPS-TSI-V0.02 07.03.2006.).

[4] STS konfigurēšana. Administratora rokasgrāmata (VDAA-13_7_17_41-VISS_2016-STS-AR)

[5] eIDAS SAML Attribute Profile. Version 1.2. eIDAS Technical Specifications.

[6] OASIS.SAML V2.0 Attribute Extensions Version 1.0 http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attribute-ext-cd-01.pdf.

[7] OpenID Connect Core 1.0, http://openid.net/specs/openid-connect-core-1_0.html

[8] OpenID Connect Discovery 1.0, http://openid.net/specs/openid-connect-discovery-1_0.html

[9] OpenID Connect Front-Channel Logout 1.0, https://openid.net/specs/openid-connect-frontchannel-1_0.html

[10] OAuth 2.0, http://tools.ietf.org/html/rfc6749

[11] OAuth 2.0 Bearer Token Usage, http://tools.ietf.org/html/rfc6750

[12] JSON Web Token, http://tools.ietf.org/html/rfc7519

[13] OAuth 2.0 Token Revocation, https://tools.ietf.org/html/rfc7009

[14] OAuth 2.0 Token Introspection, https://tools.ietf.org/html/rfc7662

[15] OAuth 2.0 JSON Web Tokens for Client Authentication, https://tools.ietf.org/html/rfc7523

[16] OpenID Connect Session Management 1.0, https://openid.net/specs/openid-connect-session-1_0.html

1.5. Dokumenta pārskats

Dokumentu veido pieci nodalījumi un pielikumi:

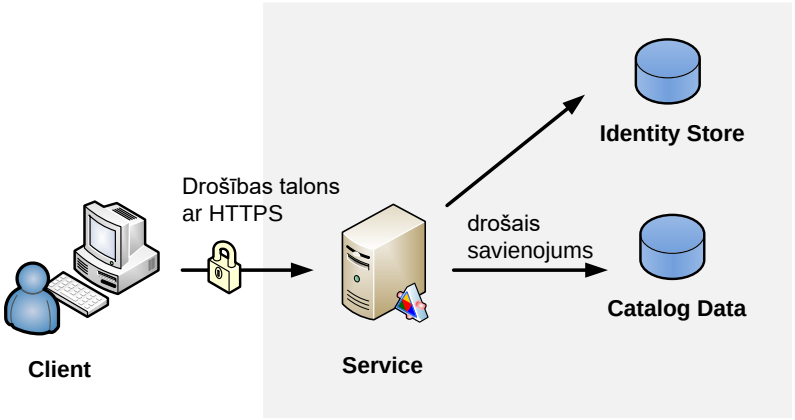
- Ievads – nodaļa satur dokumenta izmantošanas aprakstu, terminu un saīsinājumu definīcijas un saistīto dokumentu sarakstu, kā arī vispārējais apraksts;
- Drošības talonu servisa izmantošana – nodaļa satur PFAS AUTH STS izmantoto deklarāciju apraksts.
- Drošā Web Servisa izsaukšana – nodaļa satur vadlīnijas droša Web Servisa izstrādei un izsaukšanai;
- Bankas identifikācijas piegādātāja izstrādes vadlīnijas – nodaļa satur vadlīnijas Bankas identifikācijas piegādātājam;
- Vienotā pieteikšanās moduļa izmantošana – sniegta informācija, kur detalizēti aprakstīta vienotā pieteikšanās moduļa izmantošana.

1.6. Problēmu ziņošana

Gadījumos, kad lietotāja dokumentācijā vai programmatūrā ir pamanītas problēmas, par tām jāpaziņo uzturošajam personālam.

1.7. Vispārējais apraksts

Lai projektētu, realizētu un uzstrādātu drošos Web servisus, nepieciešams izmantot jaunas tehnoloģijas, kas ļauj nodrošināt funkcionalitāti potenciāli nedrošos tīklos. 1.attēlā parādīts publisko Web servisu droša izsaukuma risinājums, risinājuma detalizēts apraksts ir pieejams [1] un [2] dokumentā.



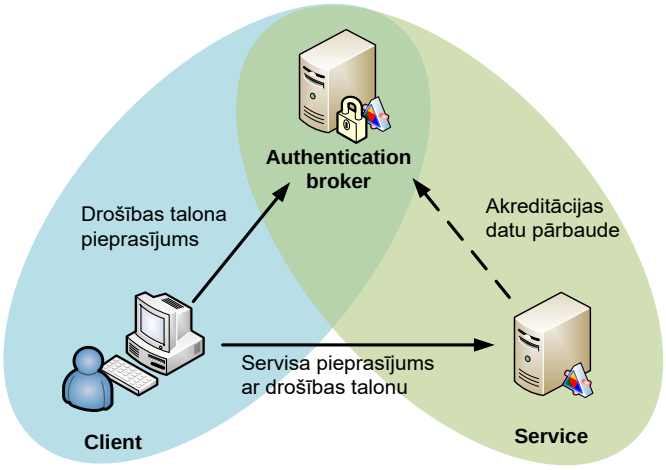
1.attēls. Publisko Web servisu drošā izsaukuma risinājums

Publisko Web servisu drošā izsaukuma risinājums izveidots šādi:

- Publiskais Web serviss izmanto servera sertifikātu drošo savienojumu (SSL) uzstādīšanai, izmantojot HTTPS.
- Klienta aplikācija nosūta drošības talonu, ar kuru tā apstiprina savu identitāti.
- Drošības talona akreditācijas dati tiek izmantoti klienta autorizācijai.
- Publiskais Web serviss izmanto drošo sistēmu piekļuvei pie kataloga datiem.

VISS (bija IVIS) vidē lietojamiem Web servisiem nepieciešams noteikt klienta autentifikāciju heterogēnās vidēs, lai varētu veikt autorizāciju un auditāciju. Šim nolūkam tiek izmantots autentifikācijas brokeri, lai nodrošinātu kopējo piekļuvi aplikāciju grupai. Autentifikācijas brokeris veic sarunas starp klientu aplikācijām un Web servisiem, tas noņem taisnās saites. Autentifikācijas brokeris izdod drošības talonus, kuri tiek izmantoti identifikācijai.

Situācija, kad klients izsauc Web servisu, starp autentifikācijas brokeri parādīta 2.attēlā.

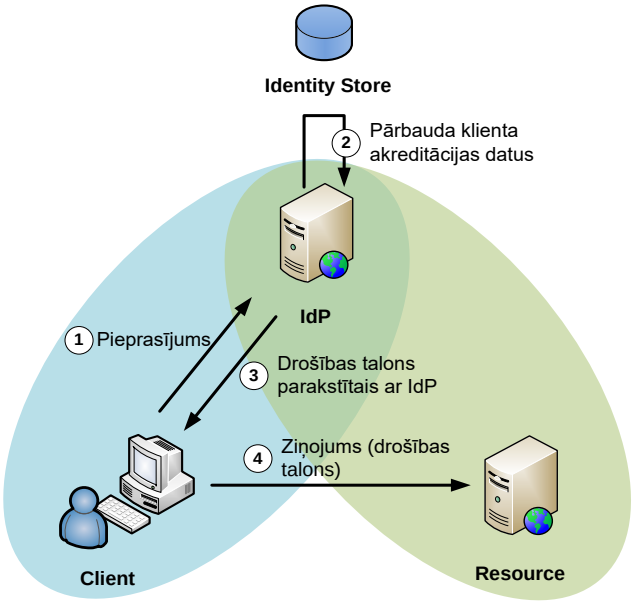


2.attēls. Autentifikācijas brokera izmantošana Web servisa izsaukšanai

Autorizācijas brokeris Web servisa izsaukšanai tiek izmantots šādi:

- Autentifikācijas brokeris pārbauda klienta akreditācijas datus un izdod drošības talonu, kuru izmanto serviss klienta autentifikācijai. Drošības talons tipiski tiek pārbaudīts uz servisa, bet servisam nav obligāti jāgriežas pie autorizācijas brokera klienta akreditācijas datu pārbaudes gadījumā, jo serviss uzticas drošības talonam, kuru izdeva autorizācijas brokeris.

VISS vidē Web lietojumiem nepieciešama autentifikācija. Šim nolūkam izmantojam identifikācijas piegādātājus. Situācija, kad klientam nepieciešana autentificēta pieeja pie Web resursa, izmantojot identifikācijas piegādātāju, parādīta 3.attēlā.



3.attēls. Identifikācijas piegādātāja izmantošana Web lietojuma resursa piekļuvei

Piekļuve resursam realizēta šādi:

- Klients griežas pie resursa. Piekļuvei nepieciešama autentifikācija. Lietotājs tiek pāradresēts uz Identifikācijas piegādātāju.
- Identifikācijas piegādātājā tiek pārbaudīti lietotāja akreditācijas dati. No identifikācijas piegādātāja tiek atgriezts drošības talons, ar to klients griežas pie resursa atkārtoti.

2. Drošības talonu servisu izmantošana

VPM nodrošina autentifikācijas rezultātā sagatavoto drošības talona aizpildīšanu ar atribūtiem, kas ietver vienu no profiliem:

- WsFed, SAML 2.0 profila atribūti, skat. 2.1. nodaļā;
- eIDAS fiziskās personas profila atribūti [5], skat. 0. nodaļā.

2.1. WsFed, SAML 2.0 profila atribūtu apraksts

Pavisam tiek izšķirti dažāda veida drošības taloni, kur katrs satur atribūtu komplektu saskaņā ar 1.tabulu, kur ir aprakstīti šādi lietotāju tipi:

- Izmantojot PFAS AUTH autentifikācijas nodrošinātāju:
  - SU – sistēma (lietotāja vārds un parole);
  - SS – sistēma (sertifikāts);
  - DU – iestādes darbinieks (lietotāja vārds un parole);
  - DS – iestādes darbinieks (sertifikāts).
- PFAS AUTH STS izmantojot no Vienotās pieteikšanas moduli *(IdentitySelector)* saņemtos iedzīvotāja autentifikācijas datus:
  - I – iedzīvotājs;
  - LE – juridiskās personas pilnvarots pārstāvis;
- Vienotās pieteikšanas modulis izmantojot no banku identifikācijas nodrošinātājiem saņemtos iedzīvotāja autentifikācijas datus:
  - I_LVP – iedzīvotāja pilnvarotais pārstāvis;
  - LE_LVP – juridiskās personas pilnvarots pārstāvis;
- banku identifikācijas nodrošinātājs:
  - I_B – iedzīvotājs.

> ***Juridiskās personas pilnvarotais pārstāvis – ir ar banku, eID vai ar citu autentifikācijas līdzekli autentificēts iedzīvotājs, kuram papildus iedzīvotāja autentifikācijas atribūtiem tiek piešķirti juridiskās personas atribūti. Juridiskās personas atribūtu pielasīšana un statusa pārbaude pret UR reģistra datiem notiek Vienotās pieteikšanās modulī.***

2.1.1. Pieprasītāju atribūtu apraksts

Pieprasītāju atribūtu saraksts dots 1.tabulā, kur ar M – atzīmētas *mandatory* (obligātās) tiesības, bet O-atzīmētas – *optional* (izvēles) tiesības.

1.tabula

Pieprasītāju atribūtu apraksts

ATRIBŪTS	VĀRDTELPA	APRAKSTS	SU	SS	DU	DS	I	LE	I_LVP	LE_LVP	I_B
nameidentifier	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/	Iekšējais kods, kas viennozīmīgi identificē autorizāciju drošības talonā (skat. 2.tabulā).	M	M	M	M	M	M	M	M	M

ATRIBŪTS	VĀRDTELPA	APRAKSTS	SU	SS	DU	DS	I	LE	I_LVP	LE_LVP	I_B
authenticationmethod	http://schemas.microsoft.com/ws/2008/06/identity/claims/	IdP URN identifikators	M	M	M	M	M	M	M	M	M
authenticationinstant	http://schemas.microsoft.com/ws/2008/06/identity/claims/	Autentifikācijas apgalvojuma izsniegšanas laiks	M	M	M	M	M	M	M	M	M
privatepersonalidentifier	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/	Personas kods*			O	O	O	O	M	M	M
emailaddress	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/	E-pasta adrese			O	O					
USER_AUTHORITY	http://www.oasis-open.org/RSA2004/attributes/	Lietotāja autentifikācijas iestāde, kur darbojas lietotājs (šobrīd netiek izmantots)			O	O					
sid	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/	Lietotaja sessijas identifikators			O	O	O	O	O	O	O
primarysid	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/	Lietotāja autentifikācijas unikāls identifikators PFAS AUTH	M	M	M	M					
AUTHORITY	http://www.oasis-open.org/RSA2004/attributes/	Lietotāja piešķirtās autentifikācijas iestāde - īpašnieks	O	O	O	O					
name	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/	Vārds	O	O	O	O					
surname	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/	Uzvārds*			O	O	O	O	M	M	M
givenname	http://schemas.xmlsoap.org/ws/20	Vārds*			O	O	O	O	M	M	M

ATRIBŪTS	VĀRDTELPA	APRAKSTS	SU	SS	DU	DS	I	LE	I_LVP	LE_LVP	I_B
	05/05/identity/claims/										
action	http://docs.oasis-open.org/wsfed/authorization/200706/claims/	Lietotājam piešķirtas operācijas PFAS AUTH (aktuālas)	O	O	O	O					
role	http://schemas.microsoft.com/ws/2008/06/identity/claims/	Lietotājam piešķirtas lomas PFAS AUTH (aktuālas)	O	O	O	O					
legalentity	http://ivis.eps.gov.lv/schema/identity/claims/	Juridiskās personas, kuru pārstāv lietotājs, UR 11-zimju kods	O	O	O	O	O			M	
legalentityname	http://ivis.eps.gov.lv/schema/identity/claims/	Juridiskās personas, kuru pārstāv lietotājs, nosaukums (no UR DB)					O			M	
legalentityshortname	http://ivis.eps.gov.lv/schema/identity/claims/	Juridiskās personas, kuru pārstāv lietotājs, saisinātājs nosaukums (no UR DB)					O			O	
legalentityaddress	http://ivis.eps.gov.lv/schema/identity/claims/	Juridiskās personas, kuru pārstāv lietotājs, juridiskā adrese (no UR DB)					O			O	
legalentityposition	http://ivis.eps.gov.lv/schema/identity/claims/	Juridiskās personas, kuru pārstāv lietotājs, amats (no UR DB)								O	
109x34	http://ivis.eps.gov.lv/schema/media/image/	Identifikācijas piegādātāja bildes interneta adrese							O	O	
employeegroup	http://ivis.eps.gov.lv/schema/identity/claims/	Lietotāja grupu identifikatori no PFAS AUTH			O	O					
systemcode	http://ivis.eps.gov.lv/schema/identity/claims/	VISS informācijas sistēmas identifikators, kas veic pieprasījumu	O	O							

ATRIBŪTS	VĀRDTELPA	APRAKSTS	SU	SS	DU	DS	I	LE	I_LVP	LE_LVP	I_B
grantorname	http://ivis.eps.gov.lv/schema/identity/claims/	Pilnvaras izdevēja nosaukums							M	M	
grantor	http://ivis.eps.gov.lv/schema/identity/claims/	Pilnvaras izdevēja identifikators (reģistrācijas numurs – iestādes kods)							M	M	
citizenQAALevel	http://ivis.eps.gov.lv/schema/identity/claims/citizenQAALevel	Autentifikācijas kvalifikācijas līmenis, skat. 3. tabulā.							M	M	
legalentityrepresentation	http://ivis.eps.gov.lv/schema/identity/claims/legalentityrepresentation	Pazīme, ka lietotājam ir tiesības pārstāvēt uzņēmumu kopā vai individuāli ("together" | "alone")						O		O	
ListOfAuthentications	http://ivis.eps.gov.lv/schema/identity/claims/listofauthentications	Pieejamo kontu saraksts							M	M	

*- atribūtiem *givenname, surename, privatepersonalidentifier*, kas tiek nodoti no banku adapteriem, tiek izmantots *OriginalIssuer* atribūtu paplašinājums [6].

Nameidentifier pielaides lietotāju kodu veido viens vai vairāki identifikatori, kopā ar atsauci uz identifikācijas sistēmu, piemēram:

2.tabula

Lietotāja nameidentifier apraksts

PERSONAS TIPS	IDENTIFIKATORA PIEMĒRS	NAMEIDENTIFIER ATRIBŪTS – FORMAT
Iedzīvotājs	*PK:10098610000*	urn:ivis:100001:name.id-viss
Iedzīvotājs ar neapstiprinātu identitāti (pašreiz tikai e-mail)	*jurijs@abcsoftware.lv*	urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
Iestādes darbinieks (no IVIS klasifikatora)	*AU:100001-PK:10098610000*	urn:ivis:100001:name.id-viss
Uzņēmuma paraksttiesīgā persona (persona, kas pēc Uzņēmumu reģistra datiem var vienpersoniski pārstāvēt uzņēmumu)	*PK:07017010000-UR:40003627089*	urn:ivis:100001:name.id-viss
Sistēma	*AU:100001*	urn:ivis:100001:name.id-viss

PERSONAS TIPS	IDENTIFIKATORA PIEMĒRS	NAMEIDENTIFIER ATRIBŪTS – FORMAT
Iedzīvotājs, kuru • pilnvarojis cits iedzīvotājs • pilnvarojis uzņēmums • pilnvarojusi valsts iestāde	*DP:01127612344-PK:07017010000* *DP:40003627089-PK:07017010000* *DP:100001-PK:07017010000*	urn:ivis:100001:name.id-viss

nameidentifier pielaides saņemšanas WIF3.5 koda piemērs:

```
ClaimsIdentity cIdentity = claimsPrincipal.Identity as ClaimsIdentity;
                    if (cIdentity != null)
                        var       nameId       =       GetClaimValue(cIdentity,
ClaimTypes.NameIdentifier);


private string GetClaimValue(IClaimsIdentity identity, string claimType) {
              Claim   claim   =   identity.Claims.FirstOrDefault<Claim>(c   =>
c.ClaimType == claimType);
            if (claim != null && !string.IsNullOrEmpty(claim.Value)) {
                return claim.Value;
            }

            return null;
        }
```

Parasti Autentifikācijas kvalifikācijas līmenis (citizenQAALevel) tiek rēķināts izmantojot 3. tabulā noradīto algoritmu. Atsevišķiem piegādātajiem var norādīt līmeni LVP.STS konfigurācijā:

```
<ivis.lvp.sts>
  <claimsProviderLevels>
    <add realm="https://epak2.abcsoftware.lv/eSign/1.5" level="4"/>
    <add realm="https://epak2.abcsoftware.lv/OpenId" level="1"/>
  </claimsProviderLevels>
</ivis.lvp.sts>
```

3.tabula

Autentifikācijas kvalifikācijas līmenis

Autentifikācijas identifikators	Autentifikācijas kvalifikācijas līmenis
SAKĀS AR URN:IVIS:100001:AM.SIGN	4
SAKĀS AR URN:IVIS:100001:AM.BANK	2
CITI	1

4.tabula

Identitātes piegādātāju autentifikācijas metožu identifikatori

IDENTITĀTES PIEGĀDĀTĀJA NOSAUKUMS	JAUNĀ NOTĀCIJA
E-paraksti	
Autentifikācija ar e-ID kartiņu	URN:IVIS:100001:AM.SIGN-EID
Autentifikācija ar e-paraksta kartiņu	URN:IVIS:100001:AM.SIGN-EME

IDENTITĀTES PIEGĀDĀTĀJA NOSAUKUMS	JAUNĀ NOTĀCIJA
Autentifikācija e-ID, e-Paraksta kartiņu un e-Parakta viedtālruņa lietotni	URN:IVIS:100001:AM.SIGN-LVCRT
Bankas	
Luminor/DNB bankas autentifikācija	URN:IVIS:100001:AM.BANK-DNB
Hansabankas/Swedbank autentifikācija	URN:IVIS:100001:AM.BANK-SWED
Luminor/Nordea bankas autentifikācija	URN:IVIS:100001:AM.BANK-NORDEA
Norvik bankas autentifikācija	URN:IVIS:100001:AM.BANK-NORVIKBANKA
Parex bankas autentifikācija	URN:IVIS:100001:AM.BANK-CITADELE
SEB bankas autentifikācija	URN:IVIS:100001:AM.BANK-SEB
Meridiantrade bankas autentifikācija	URN:IVIS:100001:AM.BANK-MERIDIANTRADE
Privat bankas autentifikācija	URN:IVIS:100001:AM.BANK-PRIVATE
Rietumu bankas autentifikācija	URN:IVIS:100001:AM.BANK-RIETUMU
ABLV bankas autentifikācija	URN:IVIS:100001:AM.BANK-ABLV
BlueOrange bankas autentifikācija	URN:IVIS:100001:AM.BANK-BLUEORANGE

2.2. eIDAS fiziskās personas profila atribūtu apraksts

eIDAS fiziskās personas profila (ārzemnieka) atribūtu saraksts dots 8.tabulā, kur ar M – atzīmētas *mandatory* (obligātās) tiesības, bet O-atzīmētas – *optional* (izvēles) tiesības.

5.tabula

eIDAS fiziskās personas profila atribūtu apraksts

ATRIBŪTS	VĀRDTELPA	APRAKSTS	ĀRZEMNIEKS
nameidentifier	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/	Iekšējais kods, kas viennozīmīgi identificē autorizāciju drošības talonā (skat. 2.tabulā).	M
authenticationmethod	http://schemas.microsoft.com/ws/2008/06/identity/claims/	IdP URN identifikators	M
authenticationinstant	http://schemas.microsoft.com/ws/2008/06/identity/claims/	Autentifikācijas apgalvojuma izsniegšanas laiks	M
PersonIdentifier	http://eidas.europa.eu/attributes/naturalperson/PersonIdentifier	Personas identifikators*	M
FirstName	http://eidas.europa.eu/attributes/naturalperson/CurrentGivenName	Vārds	M
FamilyName	http://eidas.europa.eu/attributes/naturalperson/CurrentFamilyName	Uzvārds	M
DateOfBirth	http://eidas.europa.eu/attributes/naturalperson/DateOfBirth	Dzimšanas datums	M

*- atribūts *PersonIdentifier* ietver PMLP IR ARI izsniegtais tehniskai personas kods, kas ir nemainīgs pret eIDAS personas identifikatoru.

6.tabula

Trasējamība no EIDAS LOA uz VISS autentifikācijas metožu identifikatoriem

EIDAS LEVELS OF ASSURANCE	VISS AUTENTIFIKĀCIJAS IDENTIFIKATOROS

http://eidas.europa.eu/LoA/low	URN:IVIS:100001:AM.EIDAS-LOW
http://eidas.europa.eu/LoA/substantial	URN:IVIS:100001:AM.EIDAS-SUBSTANTIAL
http://eidas.europa.eu/LoA/high	URN:IVIS:100001:AM.EIDAS-HIGH

7.tabula

Trasējamība no VISS autentifikācijas metožu identifikatoriem uz EIDAS LOA

VISS AUTENTIFIKĀCIJAS IDENTIFIKATOROS	**EIDAS LEVELS OF ASSURANCE**
Sākās ar URN:IVIS:100001:AM.SIGN	http://eidas.europa.eu/LoA/high
Sākās ar URN:IVIS:100001:AM.BANK	http://eidas.europa.eu/LoA/substantial

3. Drošā Web Servisa izsaukšana

Drošie Web servisi tiek izstrādāti balstoties uz konsorcija OASIS izstrādātiem protokoliem *WS-Security, WS-Trust, WS-Converation*. Protokoli tika implementēti Windows *Identity Foundation* .NET klašu bibliotēkā 3.5 vai vienkārši .NET 4.0 ietvara versiju.

Drošo web servisu izsaukšana ir detalizēti aprakstīta [4] sadaļā 8.4. WS-Federatoin Active Profile.

3.1. PFAS AUTH STS ieejas punkti

Drošības talonu servisa adrese sastāv no

```
.../Issue.svc/trust/{version}/{name}
```

8.tabula

{version} atribūta iespējamās vērtības un tā atbilstība SAML un WS-* standartiem

VERSION	SOAP	WS-TRUST	WS-ADDRESSING	WS-SECURITY
2005	V1.2	V1.2 (2005)	2005/08	V1.1
13	V1.2	V1.3	2005/08	V1.1
wse3	V1.2	V1.2 (2005)	2004/08	V1.1

9.tabula

{name} atribūta iespējamās vērtības un tā atbilstība WS-* standartiem

NAME	WS-SECURITY
certificate	Certificate profile
certificatemixed	Certificate profile over transport
username	UserName profile
usernamemixed	UserName profile over transport
issuedtokensymmetricbasic256	IssedToken(SAML) profile
issuedtokenmixedsymmetricbasic256	IssedToken(SAML) profile over transport

VISS infrastruktūrā drošie servisi tiek izmantoti tikai ar SSL („*over transport*"), PFAS AUTH STS tiek uzturēti šādas sasaistes (*bindings*) .NET 4.0 ietvaram:

10.tabula

Sasaistes

NAME	BINDING CONFIGURATION

13/certificatemixed	`<ws2007HttpBinding>`
	`  <binding name="certificateMixed">`
	`  <security mode="TransportWithMessageCredential">`
	`  <message clientCredentialType="Certificate" establishSecurityContext="false" />`
	`  </security>`
	`  </binding>`
	`</ws2007HttpBinding>`

13/usernamemixed	`<ws2007HttpBinding>`
	`  <binding name="usernameMixed">`
	`  <security mode="TransportWithMessageCredential">`
	`  <message clientCredentialType="UserName" establishSecurityContext="false" />`

	</security> </binding> </ws2007HttpBinding>
2005/certificatemixed	<wsHttpBinding> <binding name="certificateMixed"> <security mode="TransportWithMessageCredential"> <message clientCredentialType="Certificate" establishSecurityContext="false" /> </security> </binding> </wsHttpBinding>
2005/usernamemixed	<wsHttpBinding> <binding name="usernameMixed"> <security mode="TransportWithMessageCredential"> <message clientCredentialType="UserName" establishSecurityContext="false" /> </security> </binding> </wsHttpBinding>

3.2. ws2007Federation vs ws2007FederationNoSct

No biznesa procesa viedokļa var būt dažādi scenāriji. Piemēram, datu saņemšanas operācija sastāv no vairākiem soļiem: iegūt datus no DIT servisa pa vienam ziņojumam, kopa vairāk nekā ~10 vaicājumi. Šajā gadījuma lietderīgi izmantot WS-SecureConversation, kas nodrošina drošas sesijas uzturēšanu:

> ***Izraksts no wikipedia: WS-SecureConversation is to establish security contexts for multiple SOAP message exchanges, reducing the overhead of key establishment.***

11.tabula

Ziņojumu plūsma atkarībā no *binding* nosaukuma

NAME	MESSAGE FLOW
ws2007FederationNoSct	CallServiceRequest CallServiceResponse
ws2007Federation	IssueSCTRequest IssueSCTResponse CallServiceRequest CallServiceResponse CallServiceRequest CallServiceResponse … CallServiceRequest CallServiceResponse CancelSCTRequest CancelSCTResponse

ws2007Federation izmantošanas rekomendācija:

> ***ws2007Federation gadījumā nedrīkst turēt sesiju atvērtu ilgu laiku, jo sesiju skaits ir ierobežots. Tas nozīmē, ka citiem klientiem varētu būt atteikums, pieprasot sesiju.***

3.3. Droša Web Servisa apraksts

VISS WCF droša servisa adrese sastāv no

```
...Service.svc/{name}
```

12.tabula

{name} atribūta iespējamās vērtības un to atbilstība WS-* standartiem

NAME	SOAP	WS-TRUST	WS-ADDRESSING	WS-SECURITY	WS-SECURECONVERSATION
ws2007Federation	V1.2	V1.3	2005/08	V1.1	V1.3
ws2007FederationNoSct	V1.2	V1.3	2005/08	V1.1	

ws2007Federation vai ws2007FederationNoSct izvēle ir atkarīga no biznesa procesa. Ja klientam jāpieslēdzas pie servisa un jāizsauc vairāk pār ~10 vaicājumiem, tad lietderīgāk izmantot SCT. Ja serviss jāizsauc vienu reizi, tad bez SCT.

13.tabula

Drošiem web servisiem jānodrošina šādas sasaistes (*bindings*) .NET 4.0 ietvaram

NAME	BINDING CONFIGURATION

ws2007Federation	`<ws2007FederationHttpBinding>`
	`  <binding name="ws2007Federation">`
	`  <security mode="TransportWithMessageCredential" />`
	`  </binding>`
	`</ws2007FederationHttpBinding>`

ws2007FederationNoSct	`<ws2007FederationHttpBinding>`
	`  <binding name="ws2007FederationNoSct">`
	`  <security mode="TransportWithMessageCredential">`
	`  <message establishSecurityContext="false" >`
	`  </message>`
	`  </security>`
	`  </binding>`
	`</ws2007FederationHttpBinding>`

14.tabula

Drošiem Web servisiem jānodrošina šādas sasaistes (bindings) .NET 3.5 ietvaram

NAME	BINDING CONFIGURATION

ws2007FederationNoSct	`<customBinding>`
	`  <binding name="ws2007FederationNoSct">`
	`  <security authenticationMode="IssuedTokenOverTransport"`
	`messageSecurityVersion="WSSecurity11WSTrust13WSSecureConversation13WSSecurityPolicy12BasicSecurityProfile10" >`
	`  </security>`
	`  <textMessageEncoding />`
	`  <httpsTransport />`
	`  </binding>`
	`</customBinding>`

3.4. Drošā Web servisa klienta konfigurācija

Klienta konfigurācija atkarīga no biznesa procesa. Ja klientam jāpieslēdzas pie servisa un jāizsauc vairāk pār ~10 vaicājumiem, tad lietderīgāk izmantot SCT. Ja serviss jāizsauc vienu reizi, tad bez SCT.

15.tabula

Drošā servisa tipiskie izsaucēji

CALLER	DESCRIPTION	LIETOTĀJU TIPS
System	Drošo servisu izsauc sistēma. Par sistēmu uzskatam programmatūru, kura darbojās pati pa sevi (Windows Serviss, Web Lietojums, Web Serviss un citi). Sistēmai drošības talonu servisā jāautentificējas ar sertifikāta palīdzību. Sertifikāta reģistrēšana notiek pēc šāda scenārija: sistēma pasūta vai noģenerē sertifikātu ar privāto atslēgu, tad publisko daļu nosūta drošības servisa talonam, drošības servisa konfigurācijā reģistrējas nodotais sertifikāts un tam izdod attiecīgas tiesības.	SU, SS,
User	Drošo servisu izsaukumu iniciē pats lietotājs. To viņš paveic ar attiecīgu programmatūru (Windows Forms). Lietotājam jāautentificējas ar sertifikātu vai lietotāju vārdu, paroli. ***Ja autentificējas ar lietotāja vārdu un paroli, paroli nedrīkst saglabāt.***	DU, DS, I_LVP, LE_LVP, UN_LVP, I_B
System + Delegation	Drošo servisu izsauc sistēma, bet ar lietotāja deleģētām tiesībām.	I, LE, UN,

Atkarībā no izsaucējā mainās klienta konfigurācijas *bindings*, .NET4.0.

16.tabula

Klienta konfigurācijas *bindings*

=IZSAUCĒJS	SYSTEM.SERVICEMODEL CONFIGURATION (PIEMĒRI)
System	<system.serviceModel> <behaviors> <endpointBehaviors> <behavior name="certificatemixed"> <clientCredentials> <clientCertificate findValue="..." storeLocation="LocalMachine" x509FindType="FindByThumbprint" /> </clientCredentials> </behavior> </endpointBehaviors> </behaviors> <bindings> <ws2007HttpBinding> <binding name="certificateMixed"> <security mode="TransportWithMessageCredential"> <message clientCredentialType="Certificate" establishSecurityContext="false" /> </security> </binding> </ws2007HttpBinding> <ws2007FederationHttpBinding> <binding name="ws2007FederationNoSct"> <security mode="TransportWithMessageCredential" > <message establishSecurityContext="false" >

```
            <issuer    address=".../Issue.svc/trust/13/certificatemixed"
binding="ws2007HttpBinding" bindingConfiguration="certificatemixed" />
          </message>
        </security>
      </binding>
    </ws2007FederationHttpBinding>
  </bindings>
  <client>
    <endpoint                  address="...Service.svc/ws2007Federation"
behaviorConfiguration="certificatemixed"
binding="ws2007FederationHttpBinding"
bindingConfiguration="ws2007Federation"              contract="Contract"
name="Name"/>
  </client>
</system.serviceModel>
```

User

```
<system.serviceModel>
  <bindings>
    <ws2007HttpBinding>
      <binding name="usernameMixed">
        <security mode="TransportWithMessageCredential">
          <message                    clientCredentialType="Certificate"
establishSecurityContext="false" />
        </security>
      </binding>
    </ws2007HttpBinding>
    <ws2007FederationHttpBinding>
      <binding name="ws2007FederationNoSct">
        <security mode="TransportWithMessageCredential" >
          <message establishSecurityContext="false" >
            <issuer       address=".../Issue.svc/trust/13/usernamemixed"
binding="ws2007HttpBinding" bindingConfiguration="usernameMixed" />
          </message>
        </security>
      </binding>
    </ws2007FederationHttpBinding>
  </bindings>
  <client>
    <endpoint                  address="...Service.svc/ws2007Federation"
binding="ws2007FederationHttpBinding"
bindingConfiguration="ws2007Federation"              contract="Contract"
name="Name"/>
  </client>
</system.serviceModel>
```

Papildus koda izmaiņas:

```
client.Credentials.UserName.UserName = "User"
client.Credentials.UserName.Password = "password"
```

3.5. PFAS AUTH STS OAuth2 ieejas punkti

Drošības talonu servisa adreses:

```
.../oauth2/token
```

Autorizācija izmantojot lietotaju vardu un paroli https://tools.ietf.org/html/rfc6749#section-1.3.3

Autorizācija izmantojot kodu https://tools.ietf.org/html/rfc6749#section-1.3.1

Autorizācija izmantojot sertifikātu https://tools.ietf.org/html/rfc7523#section-2.1

```
.../oauth2/authorize
```

Authirizācija Web lietojumiem https://tools.ietf.org/html/rfc6749

```
.../oauth2/introspect
```

Ieejas punkts kas izdot informāciju par talonu https://tools.ietf.org/html/rfc7662

```
.../oauth2/revoke
```

Ieejas punkts kas aptur izdoto talonu https://tools.ietf.org/html/rfc7009

4. Bankas identifikācijas piegādātāja izstrādes vadlīnijas

Bankas identifikācijas piegādātāja (adaptera) izstrādei jāizmanto MS *VisualStudio template*, kas pieejams VISS piemēru bibliotēkā.

4.1. Identifikācijas adaptera MS VisualStudio template apraksts

Identifikācijas adapteriem (bankas, e-pasts utt.) var izmantot specializētu VisualStudio šablonu, kas darbojas kā “proxy” no identifikācijas piegādātāja uz vienotās pieteikšanās moduli. Adaptera koda apraksts pieejams 9.2. pielikumā.

4.2. Bankas adaptera specifikācija

VISS infrastruktūrā veidojamajiem bankas adapteriem ir jānodrošina šāda operācija izpilde, izmantojot HTTP komandas:

1. Autentifikācijas pieprasījums

```
GET url?parameters HTTP/1.1
```

CT=string - obligāts parametrs, norada uz autentifikācijas kontekstu, jāatgriež atpakaļ.

2. Autentifikācijas atgriešana

```
POST url?parameters HTTP/1.1
```

PK=string – obligāts parametrs, personas kods;

FN=string – obligāts parametrs, vārds;

LN=string – obligāts parametrs, uzvārds;

CT=string – obligāts parametrs, konteksta informācijā, kas tika sūtīta pieprasījumā.

3. Autentifikācijas pieprasījuma atcelšana

```
POST url?parameters HTTP/1.1
```

Cancel=true

GET izmanto query string, POST izmanto body

4.3. Universālā bankas adaptera specifikācija

Adapteris nodrošina universālu saskarni lietotāju autentifikācijai.

Adaptera specifikācija:

1. Informācijas apmaiņa starp identifikācijas nodrošinātāju un universālo adapteri tiek veikta, izmantojot HTTPS protokolu.
2. Portāls Latvija.lv sagatavo pieprasījumu „4002”, kas nodrošina lietotāja piekļuvi autentifikācijas nodrošinātāja autentifikācijai. Pieprasījums ved uz interneta lapas adresi, kura norādīta konfigurācijas failā.
3. Autentifikācijas nodrošinātājs veic pilnu lietotāja autentifikāciju. Autentifikācijas nodrošinātājs veic tikai fizisko personu autentifikāciju.

4. Autentifikācijas nodrošinātājam jāsaņem lietotāja piekrišana personas datu nosūtīšanai uz portālu Latvija.lv:
    a. Ja lietotājs piekrīt personas datu nosūtīšanai, autentifikācijas nodrošinātājs sagatavo pieprasījumu „3002”, kas nodrošina lietotāja piekļuvi LVP. Pieprasījums tiek nosūtīts uz interneta lapas adresi: https://www.latvija.lv/Default.aspx kuru uztur Valsts digitālās attīstības aģentūra;
    b. Ja lietotājs nepiekrīt personas datu nosūtīšanai, Internetbanka pāradresē lietotāju uz interneta lapas adresi: https://www.latvija.lv/Default.aspx, kuru uztur Valsts digitālās attīstības aģentūra;
    c. Atbilde tiek nodota ar POST metodi.
5. Sūtījumu parakstīšana un parakstu atšifrēšana:
    a. Digitālā paraksta iegūšanai tiek izmantoti šādi lauki: type, version, sender_id, nonce, info, user, date, time. Ja nosūtāmie vai saņemamie dati nesatur kādu no laukiem, tas netiek izmantots. Lauku kārtība ir atbilstoši norādītajai kārtībai;
    b. Kodējumam izmanto UTF-8 kodējumu;
    c. Digitālā paraksta MAC008 vērtības aprēķins tiek veikts, izmantojot publiskās atslēgas RSA algoritmu un SHA-1 "hash" algoritmu. Aprēķinā tiek ņemta vērā pieprasījuma parametru garums, tā saucamie pieprasījuma tukšie lauki:
    $MAC008(x_1,x_2,...,x_n) := RSA(SHA\text{-}1(x_1| \,|\, x_2 \,|\, |... \,|\, |x_n),e,n)$
    kur:
    | | – Simbolu rindas konkatenācija;
    $x_1, x_2, ..., x_n$ – Pieprasījuma parametri;
    e, n – RSA parametri;
    d. Iegūtā simbolu virkne tiek parakstīta ar privāto atslēgu lietojot RSA un SHA-1 funkcijas;
    e. Privātajai atslēgai jābūt 1024 bitus garai;
    f. Iegūtā vērtība ir pieprasījuma elektroniskais paraksts;
    g. Aprēķinātā digitālā paraksta vērtība tiek pārveidota simbolu virknē, izmantojot BASE64 kodu, un nosūtīta darījuma pretējai pusei pieprasījuma parametrā “signature”.
6. Pieprasījuma saņēmējs pārbauda, vai pieprasījuma dati atbilst parakstam:
    a. Ģenerē saņemtajam pieprasījumam atbilstošo simbolu virkni izmantojot algoritmu, kas aprakstīts 5.punktā;
    b. Lietojot Publisko atslēgu, salīdzina iegūto simbolu virkni ar saņemto elektronisko parakstu.
7. Pieprasījumu parakstīšanas un paraksta verifikācijas funkcijas ir OpenSSL bibliotēkas vai analogas bibliotēkas funkcijas.

17.tabula

Nododamie parametri

NOSAUKUMS	TIPS	OBLIGĀTS	APRAKSTS
type	string(4)	jā	Pieprasījuma tips (4002)
version	string(3)	jā	Izmantotais paraksta algoritms. Konstante „008”
sender_id	string(15)	jā	Uzņēmuma identifikators
nonce	string(50)	jā	Unikāla atslēga, kuru ģenerē pieprasījuma nosūtītājs (izmanto, lai garantētu operācijas svaigumu)
returnURL	string(60)	jā	URL, uz kuru jāsūta atbilde

NOSAUKUMS	TIPS	OBLIGĀTS	APRAKSTS
charset	string(10)	nē	Ziņojuma kodējums. Neobligāts parametrs. Pieļaujams ISO-8859-1 (noklusētā vērtība) vai UTF-8
signature	string(300)	jā	Digitālais paraksts

18.tabula

Saņemamie parametri

NOSAUKUMS	TIPS	OBLIGĀTS	APRAKSTS
type	string(4)	jā	Pieprasījuma tips (3002)
version	string(3)	jā	Izmantotais paraksta algoritms. Konstante „008”
user	string(16)	jā	Lietotāja identifikators
date	string(10)	jā	Paketes ģenerācijas datums. Datuma formāts „dd.MM.yyyy”
time	string(8)	jā	Paketes ģenerācijas laiks. Laika formāts „HH:mm:ss”
sender_id	string(15)	jā	Bankas identifikators
info	string(300)	jā	Lauks, kas satur autentificējamās personas personīgos datus – uzvārdu un vārdu, un personas kodu. Ja lauks satur JSON struktūru, tad JSON struktūras elements “lastName” satur uzvārdu, elements “firstName” – vārdu un elements “personCode” – personas kodu. Pretējā gadījumā lauks satur uzvārdu un vārdu, atdalītus ar tukšuma zīmi, un personas kodu, atdalītu ar „;”. Par personas uzvārdu tiek uzskatīts teksts līdz pēdējai tukšuma zīmei pirms „;”. Personas vārds ir pēdējais vārds no pēdējās tukšuma zīmes pirms „;” līdz „;”. „UZVARDS VARDS;PERSONAS_KODS”
charset	string(10)	nē	Ziņojuma kodējums. Neobligāts parametrs. Pieļaujams ISO-8859-1 (noklusētā vērtība) vai UTF-8
signature	string()	jā	Digitālais paraksts

4.4. HTTP plūsmas piemērs autentificējoties ar banku

Piemērs izmantojot Uri

ITEM	URI
Bankas adapters	https://host/bankadapter/
Banka	https://bank/

Solis 1. Pieprasījums uz bankas adapteri

```
GET https://host/bankadapter/?wa=wsignin1.0&wtream=urn:realm HTTP/1.1
```

Solis 2. Pāradresācija uz banku

```
GET http://bank/?CT=0DFA559C-81DD-4D38-9D63-4BD358B128E0
```

Solis 3. Bankas lietotāja interfeiss

[Implementation Specific Traffic]

Solis 4. Atgriešana no bankas

```
HTTP/1.1 200 OK
…
<html xmlns="https://www.w3.org/1999/xhtml">
<head>
<title>Working...</title>
</head>
<body>
<form method="post" action="https://host/bankadapter/SignIn.aspx">
<p>
<input type="hidden" name="PK" value="1111111111" />
<input type="hidden" name="CT" value="0DFA559C-81DD-4D38-9D63-4BD358B128E0" />
<input type="hidden" name="FN" value="Custom" />
<input type="hidden" name="LN" value="Tester" />
<button type="submit">POST</button> <!-- included for requestors that do not
support javascript -->
</p>
</form>
<script type="text/javascript">
setTimeout('document.forms[0].submit()', 0);
</script>
</body>
</html>
```

Solis 5. POST

```
POST https://host/bankadapter/SignIn.aspx HTTP/1.1
…
PK=111111111
CT=0DFA559C-81DD-4D38-9D63-4BD358B128E0
FN=Custom
LN=Tester
```

Solis 6. Testa adapteris atgriež talonu

```
HTTP/1.1 200 OK
…
<html xmlns="https://www.w3.org/1999/xhtml">
<head>
<title>Working...</title>
</head>
<body>
<form method="post" action="https://resource">
<p>
<input type="hidden" name="wa" value="wsignin1.0" />
<input                        type="hidden"                        name="wresult"
value="&lt;RequestSecurityTokenResponse&gt;...&lt;/RequestSecurityTokenRespons
e&gt;" />
<button type="submit">POST</button> <!-- included for requestors that do not
support javascript -->
</p>
</form>
<script type="text/javascript">
setTimeout('document.forms[0].submit()', 0);
</script>
</body>
</html>
```

4.5. Autentifikācijas datu saņemšana no bankas izmantojot tiešo saiti

Dažreiz nepieciešams pārsūtīt autentifikācijas informāciju no bankas uz portālu (noteiktu portāla sadaļu) izmantojot banka internetbankā pieejamo saiti, skat. 4. attēla.

4. attēls. Iespēja izsaukt latvija.lv portāla e-pakalpojumu ar autentifikāciju

Autentifikācijas datu nodošana autentifikācijas adapteram notiek izmantojot saskaņoto protokolu. Izņēmums ir izsaukuma konteksta nodrošināšana (`wctx`). Gadījumā, kad portāls iniciē autentifikācijas datu pieprasījums, adapteris pārsuta kontekstu uz banku un saņem nemainītā veida atpakaļ. Bet gadījumā, kad bankas internetbanka iniciē portāla izsaukums, notiek konteksta datu formēšana atbilstoši šādam kodam:

```
CT=URLencoded(
    pr=wsfederation
    &rm=URLencoded(http://www.latvija.lv/sts)
    [&cx=URLencoded(appContext)], kur

appContext=URLencoded(
    pr=wsfederation
    &rm=URLencoded(http://www.latvija.lv/2.0)
[&cx=URLencoded(/lv/epakalpojumi/EP00)]
```

Rezultātā jābūt šādam kontekstam un notiek autentifikācijas datu nodošana uz 0.e-pakalpojumu:

CT=rm%3Dhttp%253A%252F%252Fwww.latvija.lv%252Fsts%26pr%3Dwsfederation%26cx%3Drm%253Dhttps%25253A%25252F%25252Fwww.latvija.lv%25252F2.0%2526pr%253Dwsfederation%2526cx%253D%25252Flv%25252FEpakalpojumi%25252FEP00

Pielikumā ir HTML lapas teksts kas nodrošina SignIn konteksta izveidošanu.

4.6. HTTP plūsmas piemērs autentificējoties no bankas

Piemērs, izmantojot Uri

ITEM	URI
Bankas adapters	https://host/bankadapter/
Banka	https://bank/

Solis 1. Bankas lietotāja interfeiss

[Implementation Specific Traffic]

Solis 2. Atgriešana no bankas

```
HTTP/1.1 200 OK
…
<html xmlns="https://www.w3.org/1999/xhtml">
<head>
<title>Working...</title>
</head>
<body>
<form method="post" action="https://host/bankadapter/SignIn.aspx">
<p>
<input type="hidden" name="PK" value="1111111111" />
<input type="hidden" name="CT" value="
rm%3Dhttp%253A%252F%252Fwww.latvija.lv%252Fsts%26pr%3Dwsfederation%26cx%3Drm%253Dhttps%25253A%25252F%25252Fwww.latvija.lv%25252F2.0%2526pr%253Dwsfederation%2526cx%253D%25252Flv%25252FEpakalpojumi%25252FEP00" />
<input type="hidden" name="FN" value="Custom" />
<input type="hidden" name="LN" value="Tester" />
<button type="submit">POST</button> <!-- included for requestors that do not support javascript -->
</p>
</form>
<script type="text/javascript">
setTimeout('document.forms[0].submit()', 0);
</script>
</body>
</html>
```

Solis 3. POST

```
POST https://host/bankadapter/SignIn.aspx HTTP/1.1
…
PK=111111111
CT=
rm%3Dhttp%253A%252F%252Fwww.latvija.lv%252Fsts%26pr%3Dwsfederation%26cx%3Drm%253Dhttps%25253A%25252F%25252Fwww.latvija.lv%25252F2.0%2526pr%253Dwsfederation%2526cx%253D%25252Flv%25252FEpakalpojumi%25252FEP00
FN=Custom
LN=Tester
```

Solis 4. Testa adapters atgriež talonu

```
HTTP/1.1 200 OK
…
<html xmlns="https://www.w3.org/1999/xhtml">
<head>
<title>Working...</title>
</head>
<body>
<form method="post" action="https://resource">
<p>
<input type="hidden" name="wa" value="wsignin1.0" />
<input type="hidden" name="wctx" value="
rm%3Dhttps%253A%252F%252Fwww.latvija.lv%252F2.0%26pr%3Dwsfederation%26cx%3D%25
2Flv%252FEpakalpojumi%252FEP00" />
<input type="hidden" name="wresult"
value="&lt;RequestSecurityTokenResponse&gt;...&lt;/RequestSecurityTokenRespons
e&gt;" />
<button type="submit">POST</button> <!-- included for requestors that do not
support javascript -->
</p>
</form>
<script type="text/javascript">
setTimeout('document.forms[0].submit()', 0);
</script>
</body>
</html>
```

5. Vienotā pieteikšanas moduļa izmantošana

Vienoto pieteikšanas moduļa izmantošana detalizēti ir aprakstīta [4] sadaļā 8.5. WS-Federation Passive Profile.

6. Pieejamā informācija no STS metadatiem

Saiti uz metadatiem var iegūt, ieejot STS pirmajā lapā. WS-Federation, SAML1.1 un SAML2 protokoliem metadati tiek veidoti atbilstoši **Error! Reference source not found.** specifikācijai. OAuth2 protokolam metadati tiek veidoti atbilstoši **Error! Reference source not found.** specifikācijai.

6.1.1. STS identifikators

6.1.1.1. WS-Federation, SAML2.0 protokoli:

Elements EntityDescriptor satur entityID atribūtu. Atribūta vērtība norāda uz drošības talona servisa (STS) identifikatoru. Svarīgi pārbaudīt identifikatoru, kad tiek saņemts talons.

```
<EntityDescriptor entityID="https://host/trust" ...>
  ...
</EntityDescriptor>
```

6.1.1.2. OAuth2 protokols:

```
{
  "issuer": "urn:DEV-VRAA:LVP.STS",
  ...
}
```

6.1.2. STS parakstīšanas sertifikāts

6.1.2.1. WS-Federation protokols:

Kad serviss saņem talonu, kas bija izdots ar STS, tad jāpārbauda talona paraksts. Sertifikāta publiskā daļa iekļauta STS metadatu dokumentā. Atrod elementu RoleDescriptor ar atribūtu xsi:type ar vērtību fed:SecurityTokenServiceType. Sertifikāts RAW formāta iekļauts KeyDescriptor elementā. Sertifikāts ir domāts parakstīšanai, ja atribūta use vērtībā ir signing.

```
<RoleDescriptor xsi:type="fed:SecurityTokenServiceType" ...>
  <KeyDescriptor use="signing">
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <X509Data>
        <X509Certificate>MII...</X509Certificate>
      </X509Data>
    </KeyInfo>
  </KeyDescriptor>
  ...
</RoleDescriptor>
```

Atver *notepad* un nokopē elementa X509Certificate saturu tajā. Saglabā datni ar paplašinājumu .cer

6.1.2.2. SAML2.0 protokols:

Atrod elementu IDPSSODescriptor ar atribūta protocolSupportEnumeration vērtību urn:oasis:names:tc:SAML:2.0:protocol

```
<IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
```

```
  <KeyDescriptor use="signing">
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <X509Data>
        <X509Certificate>MII...</X509Certificate>
      </X509Data>
    </KeyInfo>
  </KeyDescriptor>
  ...
</IDPSSODescriptor>
```

Atver *notepad* un nokopē elementa X509Certificate saturu tajā. Saglabā datni ar paplašinājumu .cer.

6.1.2.3. OAuth2 protokols:

Atver saiti uz kuru norāda jwks_uri elements

```
{
  "jwks_uri": "https://host/path/.well-known/openid-configuration?jwks=true",
  ...
}
```

Atrod masīva keys vērtību ar elementa use vērtību sig.

```
{
  "keys": [
    {
      "alg": "RS256",
      "e": "AQAB",
      "kid": "B036...",
      "kty": "RSA",
      "n": "1rgO...",
      "use": "sig",
      "x5c": [
        "MIIG..."
      ],
      "x5t": "sDY..."
    }
  ]
}
```

Atver *notepad* un nokopē elementa x5c saturu tajā. Saglabā datni ar paplašinājumu .cer.

6.1.3. STS piedāvāto pielaižu (claim) saraksts

Atver https://host/LVP.STS/STS/federationmetadata/2007-06/federationmetadata.xml, izmantojot pārlūkprogrammu.

6.1.3.1. WS-Federation protokols:

Atrod elementu RoleDescriptor ar atribūta xsi:type vērtību fed:SecurityTokenServiceType.

```
<RoleDescriptor xsi:type="fed:SecurityTokenServiceType" ...>
    <fed:ClaimTypesOffered>
      <auth:ClaimType  Uri="http://schemas.xmlsoap.org/claims/EmailAddress"  Optional="True"
xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706">
        <auth:DisplayName>Email Address</auth:DisplayName>
      </auth:ClaimType>
```

```
    </fed:ClaimTypesOffered>
</RoleDescriptor>
```

Skat. vēl arī saistīto **Error! Reference source not found.** dokumentu, kur pieejams pilns pielaides apraksts.

6.1.3.2. SAML2.0 protokols:

Atrod elementu IDPSSODescriptor ar atribūta protocolSupportEnumeration vērtību urn:oasis:names:tc:SAML:2.0:protocol.

```
  <IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
      ...
    <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri" FriendlyName="E-Mail Address"
xmlns="urn:oasis:names:tc:SAML:2.0:assertion"/>
  </IDPSSODescriptor>
```

Vairāk informācijas par federācijas metadatiem var atrast standartos SAML2.0, WS-Federation.

6.1.4. STS piedāvāto talonu tipu saraksts

Atver https://host/LVP.STS/STS/federationmetadata/2007-06/federationmetadata.xml, izmantojot pārlūkprogrammu.

6.1.4.1. WS-Federation protokols:

Atrod elementu RoleDescriptor ar atribūta xsi:type vērtību fed:SecurityTokenServiceType.

```
<RoleDescriptor xsi:type="fed:SecurityTokenServiceType" ...>
    <fed:TokenTypesOffered>
      <fed:TokenType Uri="urn:oasis:names:tc:SAML:1.0"/>
    </fed:TokenTypesOffered>
</RoleDescriptor>
```

6.1.4.2. Shibbleth 1.3 and SAML 1.1 protokols

Atgriež tikai SAML1.1 apgalvojumu.

6.1.4.3. SAML 2.0 protokols

Atgriež tikai SAML2.0 apgalvojumu. Vairāk informācijas par federācijas metadatiem var atrast SAML1.1, SAML2.0, WS-Federation standartos.

6.1.5. STS pieejamie ieejas punkti

6.1.5.1. WS-Federation protokols:

Atrod elementu RoleDescriptor ar atribūta xsi:type vērtību fed:SecurityTokenServiceType.

```
  <RoleDescriptor                                    xsi:type="fed:SecurityTokenServiceType"
protocolSupportEnumeration="http://schemas.xmlsoap.org/ws/2005/02/trust    http://docs.oasis-
open.org/wsfed/federation/200706" ...>
    <fed:PassiveRequestorEndpoint>
      <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
        <Address>https://host/STS/wsfed</Address>
      </EndpointReference>
```

```
    </fed:PassiveRequestorEndpoint>
  </RoleDescriptor>
```

6.1.5.2. Shibbleth 1.3

Atrod elementu IDPSSODescriptor/SingleSignOnService ar atribūta Binding vērtību urn:mace:shibboleth:1.0:profiles:AuthnRequest.

```
<IDPSSODescriptor protocolSupportEnumeration="... urn:oasis:names:tc:SAML:1.1:protocol
urn:mace:shibboleth:1.0">
  ...
  <SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"
Location="https://host/STS/saml"/>
  ...
</IDPSSODescriptor>
```

6.1.5.3. SAML 2.0

Atrod elementu IDPSSODescriptor/SingleSignOnService ar atribūta Binding vērtību, kas sākās ar urn:oasis:names:tc:SAML:2.0:bindings.

```
<IDPSSODescriptor protocolSupportEnumeration="... urn:oasis:names:tc:SAML:2.0:protocol">
  ...
  <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://host/STS/saml2"/>
  <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://host/saml2"/>
  ...
</IDPSSODescriptor>
```

6.1.5.4. OAuth2

Elementi, kas beidzās ar “endpoint” norāda uz pieejamiem ieejas punktiem:

```
{
...
  "authorization_endpoint": "https://host/path/oauth2/authorize",
  "introspection_endpoint": "https://host/path/introspect",
  "jwks_uri": "https://host/path/.well-known/openid-configuration?jwks=true",
  "revocation_endpoint": "https://host/path/oauth2/revoke",
  "userinfo_endpoint": "https://host/path/oauth2/userinfo",
...
}
```

6.1.6. STS šifrēšanas sertifikāta publiskā daļa

6.1.6.1. WS-Federation protokols:

Atrod elementu RoleDescriptor ar atribūta xsi:type vērtību fed:SecurityTokenServiceType.

```
<RoleDescriptor xsi:type="fed:SecurityTokenServiceType" ...>
  <KeyDescriptor use="encyption">
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <X509Data>
        <X509Certificate>MII...</X509Certificate>
      </X509Data>
```

```
    </KeyInfo>
  </KeyDescriptor>
  ...
</RoleDescriptor>
```

Atver *notepad* un nokopē elementa X509Certificate saturu tajā. Saglabā datni ar paplašinājumu .cer.

6.1.6.2. SAML2.0 protokols:

Atrod elementu SPSSODescriptor ar atribūta protocolSupportEnumeration vērtību urn:oasis:names:tc:SAML:2.0:protocol.

```
<SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
  <KeyDescriptor use="encryption">
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <X509Data>
        <X509Certificate>MII...</X509Certificate>
      </X509Data>
    </KeyInfo>
  </KeyDescriptor>
  ...
</SPSSODescriptor>
```

Atver *notepad* un nokopē elementa X509Certificate saturu tajā. Saglabā datni ar paplašinājumu .cer.

6.1.7. STS uzturamie algoritmi

6.1.7.1. WS-Federation, SAML2.0 protokoli:

Elementā KeyDescriptor:

```
  <KeyDescriptor use="encryption">
    <KeyInfo
       ...
    </KeyInfo>
    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"/>
  </KeyDescriptor>
```

Elementā Extnesions uzskaitīti uzturamie algoritmi **Error! Reference source not found.**.

```
<Extensions>
  <alg:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <alg:SigningMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"
MinKeySize="1024"/>
  <alg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"
MinKeySize="1024"/>
</Extensions>
```

6.1.7.2. OAuth2 protokols:

```
{
  "token_endpoint_auth_signing_alg_values_supported" : "RS256"
}
```

6.1.8. Uzturētāja apraksts un kontaktinformācija

Elementā Organisation ir pieejams uzturamās organizācijas apraksts:

```
<Organization>
```

```
  <OrganizationName xml:lang="lv">ABC software</OrganizationName>
  <OrganizationDisplayName xml:lang="lv">SIA ABC software</OrganizationDisplayName>
  <OrganizationURL xml:lang="lv">http://www.abcsoftware.lv/</OrganizationURL>
</Organization>
```

Elementā ContactPerson ir pieejama kontaktinformācija:

```
<ContactPerson contactType="technical">
  <Company>ABC software</Company>
  <GivenName>Juris</GivenName>
  <SurName>Gekišs</SurName>
  <EmailAddress>abc@abcsoftware.lv</EmailAddress>
  <TelephoneNumber>+371-67082600</TelephoneNumber>
</ContactPerson>
```

7. Izmantošanas piemēri

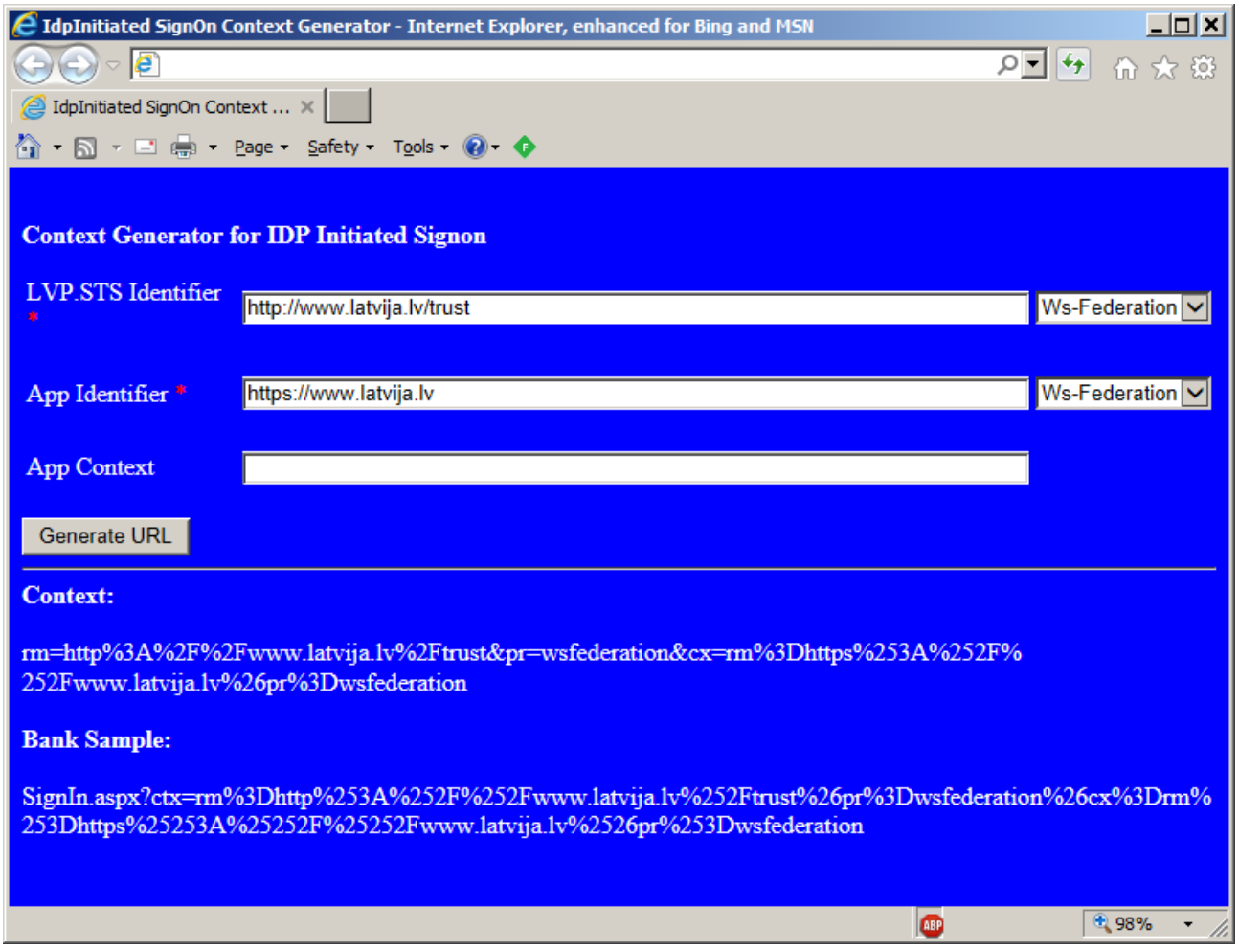
7.1. Idp initiated SignIn

Bank.STS jāieslēdz iespēja autentificēties no bankas (skat. **Error! Reference source not found.**. sadaļu). Sūtot ziņojumu no bankas, jānorāda kontekstu atbilstošā formātā (ACS notācijā), izmantojot šādus parametrus:

- **pr** – protokols;
- **rm** – lietojuma realm;
- **cx** – lietojuma konteksts;
- **ry** – lietojumā atgriešanas URL.

SignIn.aspx?ctx=**URLencoded**(
pr=protocol
&rm=**URLencoded**(https://appRealm)
[&cx=**URLencoded**(appContext)]
[&ry= **URLencoded**(https://appReplyTo)]
)

Ziņojumu kontekstu, kas domāts LVP.STS, var ģenerēt, izmantojot rīku GenerateContext.html.



5.attēls. GenerateContext.html izskats pārlūkprogrammā

7.2. HomeRealmDiscovery

7.2.1. Identitātes piegādātāju saraksta iegūšana

Adresē ir iespēja iegūt identitātes piegādātāju sarakstu: `https://host/STS/IdentityProviders.js`

Izsaukumam tiek izmantoti šādi parametri:

- **version** – obligāts, vērtībai jābūt vienādai ar "*1.0*";
- **protocol** – obligāts, protokols, pēc kura tiks atgriezta atbilde, kur *javascriptnotify* – JavaScript Notify protokols, *wsfederation* – WS-Federation Protokols;
- **realm** – obligāts, klienta lietojuma realms;
- **context** – neobligāts, tā ir papildu informācija, kura tiek atgriezta atpakaļ;
- **reply_to** – neobligāts;
- **callback** – neobligāts, te norāda javascript funkciju, kura tiek izsaukt pēc JSON ielādēs. JSON tiks padota funkcijai kā arguments.

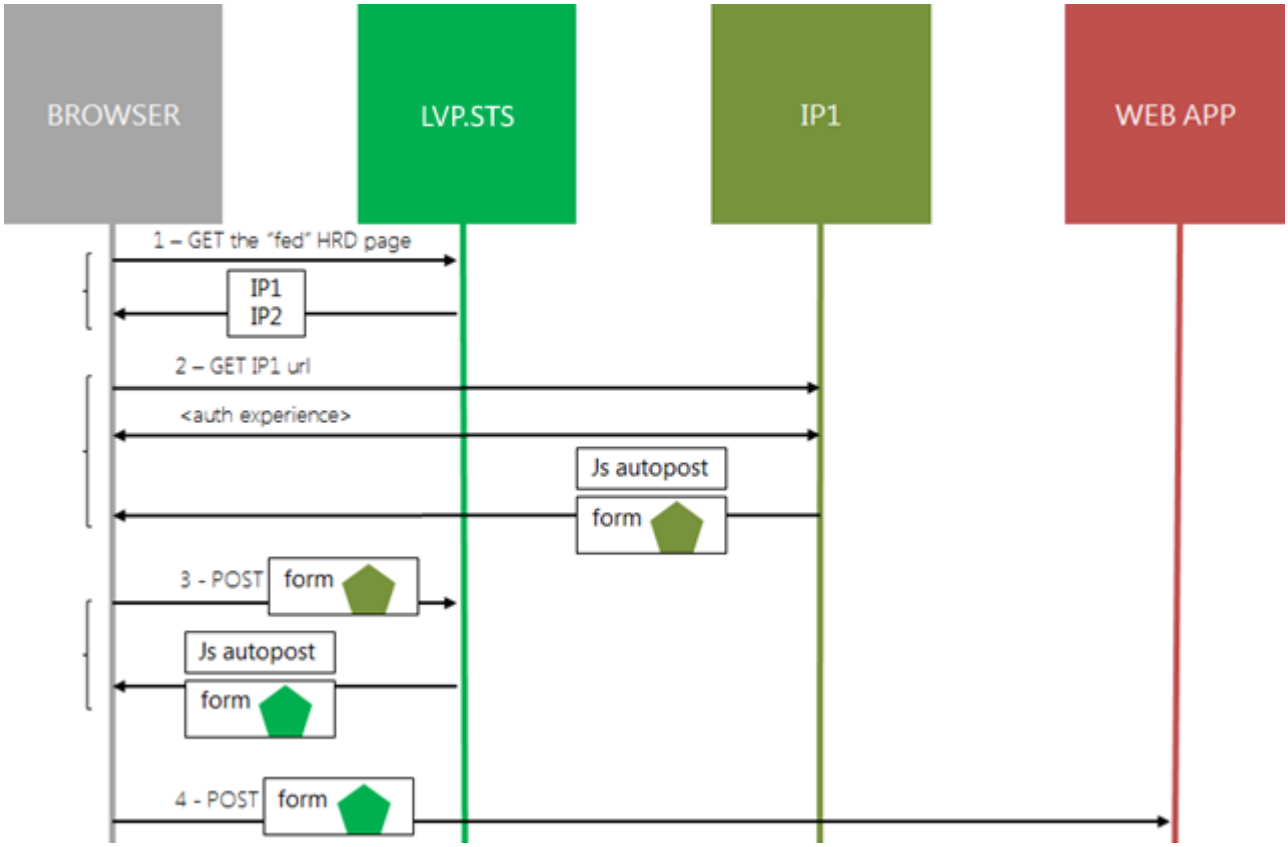
Autentifikācijas nodrošināšanas darbības rezultāts tiek atgriezts JSON formātā. Metadati tiek pieprasīti ar derīgiem parametriem, kā iepriekš aprakstīts. Atbilde ir dokuments, kurā ir JSON masīvs. Masīvam ir šādi parametri:

- **Name** —identitātes piegādātāja nosaukums;
- **LoginUrl** —izveidotais pieteikšanas url;
- **LogoutUrl** — izveidotais atteikšanas url; ja nav uzstādīts, tad nav iespējas atteikties;
- **ImageUrl** —attēls, kas saistīts ar identitātes piegādātāju; ja tukšs, tad nav attēla.

JSON formāta piemērs:

```
[
    {
        "Name": "LVP.STS",
        "LoginUrl":
"https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&wreply=https%3A%2F%
2Fhost%2FIP.STS%2FDefault.aspx&wctx=rm%3Dhttps%253a%252f%252fip.sts%252f1.0%252ftest%26pr%3D
javascriptnotify",
        "LogoutUrl": "",
        "ImageUrl": ""
    }
]
```

7.2.2. Web autentifikācija no Web lapas



6.attēls. Autentifikācijas no Web lapas secības diagramma

H piemērs:

```
<div id="IPDiv"></div>
```

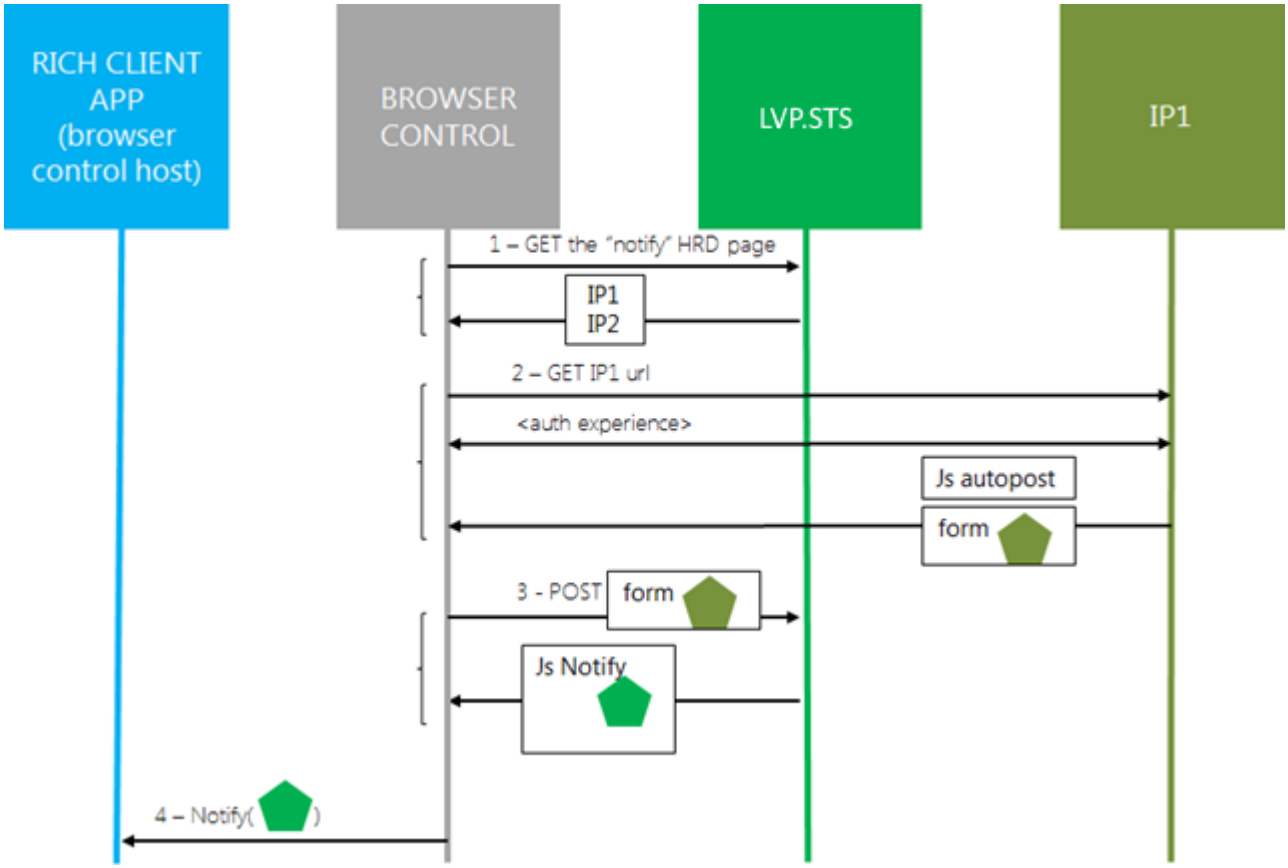
Javascript koda piemērs:

```
    <script type="text/javascript">
        function ShowSigninPage(IPs) {
            $.each(IPs, function (i, ip) {
                $('#login_bank').append('<input type="image" name="' + ip.Name + '" id="' + ip.LoginUrl + '&wfresh=0' + '"  src="' + ip.ImageUrl + '" onclick="ButtonClicked(this);return false;" style="border: 1px solid #4B75A7;display:block;width:67px;height:28px;padding: 0px 20px;"/><br/>');
            });
        }

        function ButtonClicked(input) {
            window.location.href = input.getAttribute("id");
        }
    </script>
    <script src="https://host/LVP.STS/IdentityProviders.js?version=1.0&protocol=wsfederation&realm=https%3A%2F%2Frealm%2F&callback=ShowSigninPage" type="text/javascript"></script>
```

7.2.3. Lietotāju autentifikācija un autorizācija no Rich client lietotnes

Lietotāju autentifikācija Rich client lietojumos notiek, izmantojot pārlūkprogrammas kontroli (BROWSER CONTROL) saskaņā ar *javascript notify flow* protokolu.



7.attēls. Rich Client autentifikācijas secības diagramma

Rich client lietotnē jāpievieno pārlūkprogrammas kontrole un tajā jāieliek H lapas HTML koda piemērs:`<div id="IPDiv"></div>`

un javascript koda piemērs:

```
    <script type="text/javascript">
        function ShowSigninPage(IPs) {
            $.each(IPs, function (i, ip) {
                $('#login_bank').append('<input type="image" name="' + ip.Name + '" id="' + ip.LoginUrl + '&wfresh=0' + '"  src="' + ip.ImageUrl + '" onclick="ButtonClicked(this);return false;" style="border: 1px solid #4B75A7;display:block;width:67px;height:28px;padding: 0px 20px;"/><br/>');
            });
        }

        function ButtonClicked(input) {
            window.location.href = input.getAttribute("id");
        }
    </script>
    <script src="https://host/IP.STS/IdentityProviders.js?version=1.0&protocol=javascriptnotify&realm=https%3A%2F%2Frealm%2F&callback=ShowSigninPage" type="text/javascript"></script>
```

Turpmāk notiek darbības, kas parasti notiek Web lietojumā autentifikācijas gadījumā. Pēdējā solī talons tiek nodots Rich Client lietojumam, kurā tiek pārbaudīts saņemtais drošības talons.

7.3. Interfeisa valodas norādīšana

Sūtot pieprasījumu uz STS, var norādīt vēlamo interfeisa valodu. Pieprasījumam jāpievieno parametrs lang ar vēlamo valodu ISO 639-1 kodā.

Piemērs:

https://host/PFAS.STS/wsfed?...&lang=en

7.4. WS-Federatoin Active Profile

7.4.1. Kā pieprasīt talonu ar nepieciešamo pieprasītāja informāciju (claims)

Pieprasot autentifikāciju, jānorāda vēlamā pieprasītā informācija (claims) elementā Claims. Vairāk informācijas par elementiem var atrast standartos WS-Trust, WS-Identity.

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <trust:Claims xmlns:i="http://schemas.xmlsoap.org/ws/2005/05/identity"
Dialect="http://schemas.xmlsoap.org/ws/2005/05/identity">
    <i:ClaimType Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
Optional="true" />
  </trust:Claims>
</trust:RequestSecurityToken>
```

Izmantojot WCF, jānorāda elements claimTypeRequirements konfigurācijas datnē.

Piemērs:

```
<ws2007FederationHttpBinding>
  <binding name="ws2007FederationNoSct">
    <security mode="TransportWithMessageCredential">
      <message establishSecurityContext="false">
        ...
        <claimTypeRequirements>
          <add claimType="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
isOptional="true"/>
        </claimTypeRequirements>
      </message>
    </security>
  </binding>
</ws2007FederationHttpBinding>
```

7.4.2. Kā pieprasīt noteikta tipa talonu

Pieprasot autentifikāciju, jānorāda vēlamo talona tipu elementā TokenType. Vairāk informācijas par elementiem var atrast standartos WS-Trust:

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <trust:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</trust:TokenType>
</trust:RequestSecurityToken>
```

Izmantojot WCF jānorāda elementā message atribūts issuedTokenType konfigurācijas datnē.

Piemērs:

```
<ws2007FederationHttpBinding>
  <binding name="ws2007FederationNoSct">
    <security mode="TransportWithMessageCredential">
      <message issuedTokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0">
      </message>
    </security>
  </binding>
```

```
</ws2007FederationHttpBinding>
```

7.4.3. Kā pieprasīt talonu uz noteiktu dzīves laiku

Pieprasot autentifikāciju, jānorāda vēlamo talona laiku elementā Lifetime. Vairāk informācijas par elementiem var atrast standartos WS-Trust, WS-Security:

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <trust:Lifetime>
    <wsu:Expires xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">2013-10-15T16:30:10.122Z</wsu:Expires>
  </trust:Lifetime>
</trust:RequestSecurityToken>
```

7.4.4. Kā pieprasīt talonu ar „bearear" atslēgas tipu

Pieprasot autentifikāciju, jānorāda vēlamo talona laiku elementā KeyType. Vairāk informācijas par elementiem var atrast standartā WS-Trust.

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Bearer</trust:KeyType>
</trust:RequestSecurityToken>
```

Izmantojot WCF v4.0, jānorada elementa message atribūts issuedKeyType konfigurācijas datnē.

Piemērs:

```
<ws2007FederationHttpBinding>
  <binding name="ws2007FederationNoSct">
    <security mode="TransportWithMessageCredential">
      <message establishSecurityContext="false" issuedKeyType="BearerKey">
      </message>
    </security>
  </binding>
</ws2007FederationHttpBinding>
```

7.4.5. Kā pieprasīt talonu uz konkrēto lietojuma identifikatoru

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
    <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
      <Address>https://realm</Address>
    </EndpointReference>
  </wsp:AppliesTo>
</trust:RequestSecurityToken>
```

7.4.5.1. Izmantojot WCF

Jāpievieno konfigurācijas datnē elementā tokenRequestParameters *appliesTo* elementu.

Jābūt uzmanīgam - elementam jābūt ar vārdtelpu, kas atbilst *binding* tipam.

```
<binding name="ws2007FederationNoSct">
  <security mode="TransportWithMessageCredential">
    <message establishSecurityContext="false">
      ...
      <tokenRequestParameters>
        <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
          <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
            <Address>https://realm</Address>
```

```
          </EndpointReference>
        </wsp:AppliesTo>
      </tokenRequestParameters>
    </message>
  </security>
</binding>
```

7.4.5.2. Izmantojot WCF un Abc.IdentityModel v1.1 behaviour

```
  <system.serviceModel>
    <extensions>
      <behaviorExtensions>
        <add  name="appliesTo"  type="Abc.IdentityModel.Configuration.ClientAppliesToElement,
Abc.IdentityModel" />
      </behaviorExtensions>
    </extensions>
    <client>
      <endpoint address="https://host/Service/wsFederationNoSct"
          binding="ws2007FederationHttpBinding" bindingConfiguration="ws2007FederationNoSct"
          contract="*" name="Service" behaviorConfiguration="client" />
    </client>
    <behaviors>
      <endpointBehaviors>
        <behavior name="client">
          <appliesTo address="https://realm"/>
        </behavior>
      </endpointBehaviors>
    </behaviors>
  </system.serviceModel>
```

7.4.6. Kā pieprasīt talonu ar nepieciešamo parakstīšanas algoritmu

Veicot autentifikāciju, jānorāda vēlamo talona parakstīšanas algoritmu elementā SignatureAlgorithm. Vairāk informācijas par elementiem var atrast standartā WS-Trust:

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <trust:SignatureAlgorithm>http://www.w3.org/2000/09/xmldsig#rsa-
sha1</trust:SignatureAlgorithm>
</trust:RequestSecurityToken>
```

7.4.7. Kā atjaunot talonu

Tiks definēts vēlāk.

7.4.8. Kā pieprasīt talonu ar nepieciešamo šifrēšanas algoritmu

Veicot autentifikāciju, jānorāda vēlamo talona šifrēšanas algoritmu elementā EncryptionAlgorithm. Vairāk informācijas par elementiem var atrast standartā WS-Trust:

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <trust:EncryptionAlgorithm>http://www.w3.org/2001/04/xmlenc#aes256-cbc</trust:
EncryptionAlgorithm>
</trust:RequestSecurityToken>
```

7.4.9. Kā pieprasīt talonu ar nepieciešamo keyWrap algoritmu

Veicot autentifikāciju, jānorāda vēlamā talona keyWrap algoritmu elementā KeyWrapAlgorithm. Vairāk informācijas par elementiem var atrast standartā WS-Trust:

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <trust:KeyWrapAlgorithm>http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p</trust:
KeyWrapAlgorithm>
</trust:RequestSecurityToken>
```

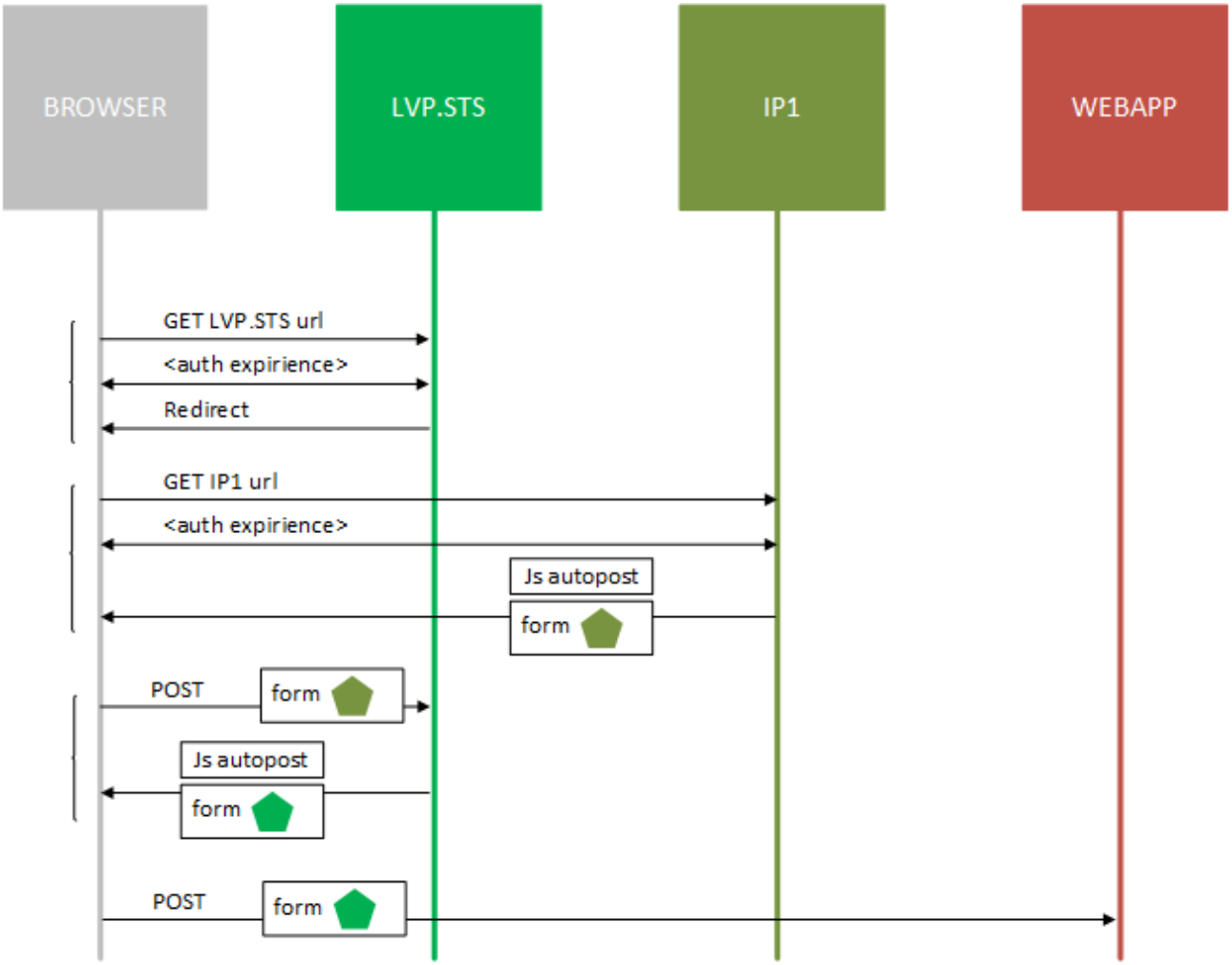
7.4.10.Kā nošifrēt talonu ar nepieciešamo sertifikātu

Veicot autentifikāciju, jānorāda vēlamā talona keyWrap algoritms elementā Encryption. Vairāk informācijas par elementiem var atrast standartā WS-Trust:

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <trust:Encryption>
    <wsse:BinarySecurityToken ValueType="wsse:X509v3" EncodingType="wsse:Base64Binary"
Id="SecurityToken-c7ff1a4e-276d-4af3-8837-1264ab9f69b3">
  MIIBtzCCAWGgAwIBAgIQz4ySuWmd/opBywS0LK...
    </wsse:BinarySecurityToken>
  </trust:Encryption>
</trust:RequestSecurityToken>
```

7.5. WS-Federation Passive Profile

7.5.1. Drošības talona pieprasīšana



8.attēls. Autentifikācijas secības diagramma

Attēlā LVP.STS – vienotais pietiekšanās modulis, IP1 – bankas adapteris, WEBAPP – web lietojums

Piemērs:

https://host/STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F

7.5.2. Kā pieprasīt talonu ar nepieciešamo pieprasīto informāciju (claims)

Jāizveido pieprasījums, skat. 7.4.1. sadaļu. Autentificējoties pieprasījums jānosūta parametrā *wreq*.

Piemērs:

https://host/PFAS.STS/wsfed?wa=wsignin1.0&wtrealm=
https%3A%2F%2Frealm%2F&wreq=%3Ctrust%3ARequestSecurityToken+xmlns%3Atrust%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fws-sx%2Fws-trust%2F200512%22%3E%0D%0A++%3Ctrust%3AClaims+xmlns%3Ai%3D%22http%3A%2F%2Fschemas.xmlsoap.org%2Fws%2F2005%2F05%2Fidentity%22+Dialect%3D%22http%3A%2F%2Fschemas.xmlsoap.org%2Fws%2F2005%2F05%2Fidentity%22%3E%0D%0A++++%3Ci%3AClaimType+Uri%3D%22http%3A%2F%2Fschemas.xmlsoap.org%2Fws%2F2005%2F05%2Fidentity%2Fclaims%2Fname%22+Optional%3D%22true%22+%2F%3E%0D%0A++%3C%2Ftrust%3AClaims%3E%0D%0A%3C%2Ftrust%3ARequestSecurityToken%3E

Izmantojot WIF 3.5 SDK, jāmodificē *web.config* un jādefinē vēlamie claims.

```
<applicationService>
        <claimTypeRequired>
          <claimType type="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name" optional="false"/>
        </claimTypeRequired>
</applicationService>
```

Jāpievieno kods Global.asax.cs datnē, skat 9.5. nodaļu.

7.5.3. Kā pieprasīt noteiktā tipa talonu

Jāizveido pieprasījums, skat. 7.4.2. sadaļu. Autentificējoties pieprasījums jānosūta parametrā *wreq*.

Piemērs:

https://host/PFAS.STS/wsfed?wa=wsignin1.0&wtrealm= https%3A%2F%2Frealm%2F&wreq=%3Ctrust%3ARequestSecurityToken+xmlns%3Atrust%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fws-sx%2Fws-trust%2F200512%22%3E%0D%0A++%3Ctrust%3ATokenType%3Ehttp%3A%2F%2Fdocs.oasis-open.org%2Fwss%2Foasis-wss-saml-token-profile-1.1%23SAMLV2.0%3C%2Ftrust%3ATokenType%3E%0D%0A%3C%2Ftrust%3ARequestSecurityToken%3E

7.5.4. Kā pieprasīt talonu uz noteiktu dzīves laiku

Jāizveido pieprasījums, skat. 7.4.3. sadaļu. Autentificējoties pieprasījums jānosūta parametrā *wreq.*

Piemērs:

https://host/PFAS.STS/wsfed?wa=wsignin1.0&wtrealm= https%3A%2F%2Frealm%2F&wreq=%3Ctrust%3ARequestSecurityToken+xmlns%3Atrust%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fws-sx%2Fws-trust%2F200512%22%3E%0D%0A++%3Ctrust%3ALifetime%3E%0D%0A++++%3Cwsu%3AExpires+xmlns%3Awsu%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fwss%2F2004%2F01%2Foasis-200401-wss-wssecurity-utility-1.0.xsd%22%3E2013-10-15T16%3A30%3A10.122Z%3C%2Fwsu%3AExpires%3E%0D%0A++%3C%2Ftrust%3ALifetime%3E%0D%0A%3C%2Ftrust%3ARequestSecurityToken%3E

7.5.5. LVP.STS vēlamās personas autentifikācijas izvēle

1. Izvēle pirms autentifikācijas.

Juridisko un pilnvaroto kodus iespējams dabūt no claima http://ivis.eps.gov.lv/schema/identity/claims/listofauthentications dekodējot ar Deflate.

Tas izsaucot pieprasījumus 7.5.5.sadaļā vai 7.5.7.sadaļā veiks autentifikāciju.

2. Izvēle pēc autentifikācijas.

Sūtot pieprasījumu zemāk, LVP.STS pēc lietotāja autentifikācijas piedāvās lietotājam izvelēties autentifikācijas veidu.

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
```

```
  <auth:AdditionalContext xmlns:auth="http://docs.oasis-
open.org/wsfed/authorization/200706">
    <auth:ContextItem
Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privatepersonalidentifier">
      <auth:Value></auth:Value>
    </auth:ContextItem>
  </auth:AdditionalContext>
</trust:RequestSecurityToken>
```

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&wreq=%3Ctrust%3ARequestSecurityToken+xmlns%3Atrust%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fws-sx%2Fws-trust%2F200512%22%3E%3Cauth%3AAdditionalContext+xmlns%3Aauth%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fwsfed%2Fauthorization%2F200706%22%3E%3Cauth%3AContextItem+Name%3D%22http%3A%2F%2Fschemas.xmlsoap.org%2Fws%2F2005%2F05%2Fidentity%2Fclaims%2Fprivatepersonalidentifier%22%3E%3Cauth%3AValue+%2F%3E%3C%2Fauth%3AContextItem%3E%3C%2Fauth%3AAdditionalContext%3E%3C%2Ftrust%3ARequestSecurityToken%3E

Sakot no LVP.STS V3 ja nepieciešams izvelēties autentifikāciju to var noradīt parametrā *scope* noradot vērtību *inhabitant inhabitant:prompt*.

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&scope=inhabitant inhabitant:prompt

7.5.6. LVP.STS juridisko personu autentifikācija

Veicot autentifikāciju, jānorāda autentifikācijas konteksts, kurā ievada uzņēmuma reģistrācijas numuru.

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <auth:AdditionalContext xmlns:auth="http://docs.oasis-
open.org/wsfed/authorization/200706">
    <auth:ContextItem Name="http://ivis.eps.gov.lv/schema/identity/claims/legalentity">
      <auth:Value>4000xxxxxxx</auth:Value>
    </auth:ContextItem>
  </auth:AdditionalContext>
</trust:RequestSecurityToken>
```

Ja sūta tukšu numuru, tad LVP.STS pēc lietotajā autentifikācijas piedāvās lietotājam izvelēties juridisko personu no saraksta, ja tādi ir vairāk kā viens.

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <auth:AdditionalContext xmlns:auth="http://docs.oasis-
open.org/wsfed/authorization/200706">
    <auth:ContextItem Name="http://ivis.eps.gov.lv/schema/identity/claims/legalentity">
      <auth:Value></auth:Value>
    </auth:ContextItem>
  </auth:AdditionalContext>
</trust:RequestSecurityToken>
```

Autentificējoties, autentifikācijas konteksts jānosūta parametrā *wreq.*

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&wreq=%3Ctrust%3ARequestSecurityToken+xmlns%3Atrust%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fws-sx%2Fws-trust%2F200512%22%3E%3Cauth%3AAdditionalContext+xmlns%3Aauth%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fwsfed%2Fauthorization%2F200706%22%3E%3Cauth%3AContextItem+Name%3D%22http%3A%2F%2Fivis.eps.gov.lv%2Fschema%2Fidentity%2Fclaims%2Flegalentity%22%3E%3Cauth%3AValue%3E4000xxxxxxx%3C%2Fauth%3AValue%3E%3C%2Fauth%3AContextItem%3E%3C%2Fauth%3AAdditionalContext%3E%3C%2Ftrust%3ARequestSecurityToken%3E

Sakot no LVP.STS V3 uzņēmuma reģistrācijas numuru var norādīt parametrā *scope* norādot vērtību *inhabitant legalentity:4000xxxxxxx*.

Ja nepieciešams izvelēties juridisko personu no saraksta tad jānorada vērtību *inhabitant legalentity:prompt*. Saraksts tiks attēlots ja ir vairāk kā viens uzņēmums.

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&scope=inhabitant legalentity:4000xxxxxxx

7.5.7. LVP.STS valsts iestāžu personu autentifikācija

Veicot autentifikāciju, jānorāda autentifikācijas konteksts, kurā ievada valsts iestādes reģistrācijas numuru no UR.

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <auth:AdditionalContext xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706">
    <auth:ContextItem Name="http://ivis.eps.gov.lv/schema/identity/claims/govlegalentity">
      <auth:Value>9000xxxxxxx</auth:Value>
    </auth:ContextItem>
  </auth:AdditionalContext>
</trust:RequestSecurityToken>
```

Ja sutā tukšu numuru, tad LVP.STS pēc lietotajā autentifikācijas piedāvās lietotājam izvelēties valsts iestādi no saraksta, ja tādas ir vairāk kā viena.

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <auth:AdditionalContext xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706">
    <auth:ContextItem Name="http://ivis.eps.gov.lv/schema/identity/claims/govlegalentity">
      <auth:Value></auth:Value>
    </auth:ContextItem>
  </auth:AdditionalContext>
</trust:RequestSecurityToken>
```

Autentificējoties, autentifikācijas konteksts jānosūta parametrā *wreq.*

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&wreq=%3Ctrust%3ARequestSecurityToken+xmlns%3Atrust%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fws-sx%2Fws-trust%2F200512%22%3E%3Cauth%3AAdditionalContext+xmlns%3Aauth%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fwsfed%2Fauthorization%2F200706%22%3E%3Cauth%3AContextItem+Name%3D%22http%3A%2F%2Fivis.eps.gov.lv%2Fschema%2Fidentity%2Fclaims%2Fgovlegalentity%22%3E%3Cauth%3AValue%3E9000xxxxxxx%3C%2Fauth%3AValue%3E%3C%2Fauth%3AContextItem%3E%3C%2Fauth%3AAdditionalContext%3E%3C%2Ftrust%3ARequestSecurityToken%3E

Sakot no LVP.STS V3 uzņēmuma valsts iestādes reģistrācijas numuru var norādīt parametrā *scope* norādot vērtību *inhabitant legalentity:9000xxxxxxx*.

Ja nepieciešams izvelēties valsts iestādi no saraksta tad jānorada vērtību *inhabitant legalentity:publicbody*. Saraksts tiks attēlots ja ir vairāk kā viens uzņēmums kuram tips ir valsts iestāde.

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&scope=inhabitant legalentity:9000xxxxxxx

7.5.8. LVP.STS pilnvaroto personu autentifikācija

Veicot autentifikāciju, jānorāda autentifikācijas konteksts, kurā ievada pilnvaras devēja reģistrācijas numuru vai personas kodu.

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <auth:AdditionalContext xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706">
    <auth:ContextItem Name="http://ivis.eps.gov.lv/schema/identity/claims/grantor">
      <auth:Value>4000xxxxxxx</auth:Value>
    </auth:ContextItem>
  </auth:AdditionalContext>
</trust:RequestSecurityToken>
```

Ja sūta tukšu numuru, tad LVP.STS pēc lietotajā autentifikācijas piedāvās lietotājam izvelēties kādu no pilnvaru devējiem, ja tādi ir vairāk kā viens.

```
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <auth:AdditionalContext xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706">
    <auth:ContextItem Name="http://ivis.eps.gov.lv/schema/identity/claims/grantor">
      <auth:Value></auth:Value>
    </auth:ContextItem>
  </auth:AdditionalContext>
</trust:RequestSecurityToken>
```

Autentificējoties, autentifikācijas konteksts jānosūta parametrā *wreq.*

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&wreq=%3Ctrust%3ARequestSecurityToken+xmlns%3Atrust%3D%22http%3A%2F%2Fdocs.oasis-open.org%2Fws-sx%2Fws-trust%2F200512%22%3E%3Cauth%3AAdditionalContext+xmlns%3Aauth%3D%22http%3A%2F%2Fdocs.oasis-

open.org%2Fwsfed%2Fauthorization%2F200706%22%3E%3Cauth%3AContextItem+Name%3D%22http%3A%2F%2Fivis.eps.gov.lv%2Fschema%2Fidentity%2Fclaims%2Fgrantor%22%3E%3Cauth%3AValue%3E4000xxxxxxx%3C%2Fauth%3AValue%3E%3C%2Fauth%3AContextItem%3E%3C%2Fauth%3AAdditionalContext%3E%3C%2Ftrust%3ARequestSecurityToken%3E

Sakot no LVP.STS V3 pilnvaras devēja uzņēmuma reģistrācijas numuru vai personas kodu var norādīt parametrā *scope* norādot vērtību *inhabitant grantor:4000xxxxxxx*.

Ja nepieciešams izvelēties pilnvaras devēju no saraksta tad jānorāda vērtību *inhabitant grantor:prompt*. Saraksts tiks attēlots ja ir vairāk kā viens pilnvaras devējs.

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&scope=inhabitant grantor:4000xxxxxxx

7.5.9. PFAS.STS izvēlēties vēlamo autentifikācijas veidu

Izmantojot WIF 3.5 SDK, jāmodificē *web.config* un jādefinē vēlamais autentifikācijas tips.

```
<federatedAuthentication>
  <wsFederation passiveRedirectEnabled="true" issuer="https://host/LVP.STS/wsfed"
realm="https://realm/" requireHttps="true"
authenticationType="urn:oasis:names:tc:SAML:1.0:am:password"
  <cookieHandler requireSsl="true" />
</federatedAuthentication>
```

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&wauth=urn%3Aoasis%3Anames%3Atc%3ASAML%3A1.0%3Aam%3Apassword

WS-Federation Passive Profile ar PFAS.STS uzturamajām autentifikācijas metodēm.

19. tabula

Savietojamības versijas īpašības

AUTHENTICATION METHOD	AUTHENTICATION CONTEXT (WAUTH) URI
Username/Password	urn:oasis:names:tc:SAML:1.0:am:password
Transport Layer Security (TLS) Client	urn:ietf:rfc:2246
Integrated Windows Authentication	urn:federation:authentication:windows

Šajā gadījumā netiks attēlota izvades forma „Izvēlieties autentifikācijas veidu”, bet uzreiz notiks pāradresācija uz formu „Pieteikties VISS portālā”.

7.5.10.LVP.STS/PFAS.STS izvelēties vēlamo autentifikācijas sniedzēju

Izmantojot WIF 3.5 SDK, jāmodificē web.config. Var definēt vēlamo autentifikācijas sniedzēju.

```
<federatedAuthentication>
  <wsFederation passiveRedirectEnabled="true" issuer="https://host/LVP.STS/wsfed"
realm="https://realm/" requireHttps="true" homeRealm="https://idprealm"
  <cookieHandler requireSsl="true" />
</federatedAuthentication>
```

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&whr=https%3A%2F%2Fidprealm

Šajā gadījumā netiks attēlota izvada forma „Izvēlieties autentifikācijas veidu", bet uzreiz notiks pāradresācija uz formu – "Identitātes piegādāja autentifikācijas lapa".

7.5.11. HomeRealmDiscovery protokola izmantošana

Jāizpilda 7.2. nodaļas nosacījumi.

Piemērs:

https://host/LVP.STS/IdentityProviders.js?version=1.0&protocol=wsfederation&realm=https%3A%2F%2Frealm%2F

7.5.12. Pieprasīt autentifikāciju no jauna (Freshness)

Izmantojot WIF 3.5 SDK, modificējot web.config, var definēt freshnes vērtību.

```
<federatedAuthentication>
  <wsFederation passiveRedirectEnabled="true" issuer="https://host/LVP.STS/wsfed"
realm="https://realm/" requireHttps="true" freshness="0"
  <cookieHandler requireSsl="true" />
</federatedAuthentication>
```

Izmantojot .NET 4.5, modificējot web.config, var definēt freshnes vērtību.

```
<federatedConfiguration>
  <wsFederation passiveRedirectEnabled="true" issuer="https://host/LVP.STS/wsfed"
realm="https://realm/" requireHttps="true" freshness="0"
  <cookieHandler requireSsl="true" />
</federatedConfiguration>
```

Piemērs:

https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F &wfesh=0

Šajā gadījumā tiks attēlota izvades forma „Izvēlieties autentifikācijas veidu", lai gan tika ieslēgta STS opcija Web SIngleSignOn.

7.5.13. Kā pieprasīt autentifikāciju ar pieprasījumu citā datnē vai servisā

Jāizveido pieprasījums, skat. 7.4.1, 7.4.2, 7.4.3, 7.4.6, 7.4.8, 7.4.9, 7.4.10. sadaļas un pieprasījums jāsaglabā datnē, vai tam jābūt pieejamam servisā. Datnes vai servisa adresi jānorāda parametrā *wreqptr.*

Izmantojot WIF 3.5 SDK, modificējot web.config var definēt citas datnes adresi.

```
<federatedAuthentication>
  <wsFederation passiveRedirectEnabled="true" issuer="https://host/LVP.STS/wsfed"
realm="https://realm/" requireHttps="true" requestPtr="https://myApphost/Request.xml"
  <cookieHandler requireSsl="true" />
</federatedAuthentication>
```

Izmantojot .NET4.5, modificējot web.config var definēt citas datnes adresi.

```
<federatedConfiguration>
  <wsFederation passiveRedirectEnabled="true" issuer="https://host/LVP.STS/wsfed"
realm="https://realm/" requireHttps="true" requestPtr="https://myApphost/Request.xml"
  <cookieHandler requireSsl="true" />
</federatedConfiguration>
```

Piemērs:

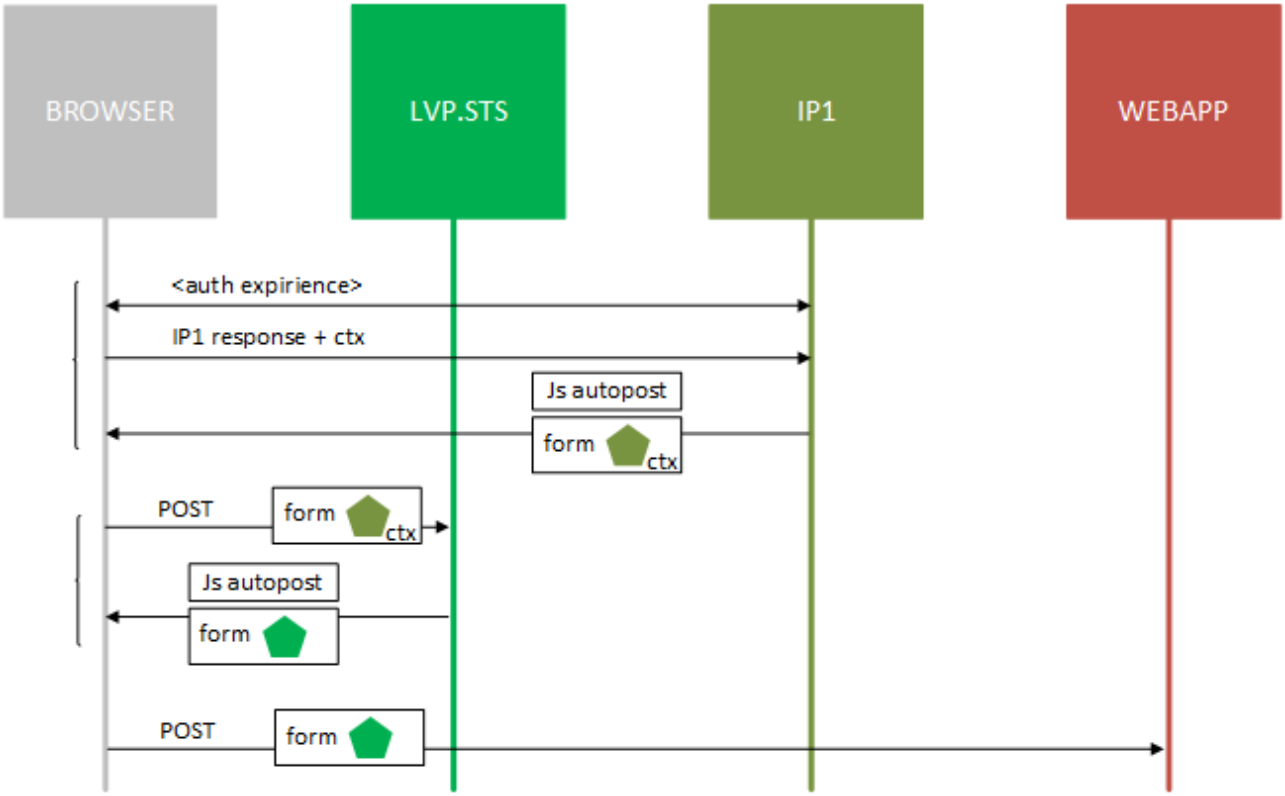
https://host/LVP.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&wreqptr=https%3A%2F%2FmyApphost%2FRequest.xml

Datnes Request.xml satura piemērs:

```
<?xml version="1.0" encoding="utf-8"?>
<trust:RequestSecurityToken xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
  <trust:Claims                    xmlns:i="http://schemas.xmlsoap.org/ws/2005/05/identity"
Dialect="http://schemas.xmlsoap.org/ws/2005/05/identity">
    <i:ClaimType            Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
Optional="true" />
  </trust:Claims>
</trust:RequestSecurityToken>
```

7.5.14.Pieteikšanās no bankām

Jāizpilda 7.1. nodaļas nosacījumi.



9.attēls. Autentifikācija no bankas secības diagramma

kur, LVP.STS – vienotas pietiekšanās modulis, IP1 – bankas adapteris, WEBAPP – web lietojums.

7.5.14.1. ACS notācija

SignIn.aspx?ctx=**URLencoded**(
pr=wsfederation
&rm=**URLencoded**(https://appRealm)
[&cx=**URLencoded**(appContext)]
[&ry=**URLencoded**(https://appReplyTo)]
)

Piemērs:

https://host/BANK.STS/SignIn.aspx?ctx=pr%3Dwsfederation%26rm%3Dhttps%253A%252F%252FappRealm

Konteksta informācijas veidošanai var izmantot **Error! Reference source not found.**. sadaļā piedāvāto kodu.

7.5.14.2. ADFS notācija

SignIn.aspx?ctx=**URLencoded**(
RPID=**URLencoded**(https://appRealm)
&wctx=[**URLencoded**(appContext)]
)

7.5.15.Atteikšanās no STS

Piemērs:

https://host/PFAS.STS/wsfed?wa=wsignout1.0&wtrealm=https%3A%2F%2Frealm%2F&wreply=http://mySite

7.5.16.Lietojuma konteksta informācijas pārsūtīšana caur STS

7.5.16.1. Izmantojot parametru wctx

- Kontekstu var pārsūtīt, izmantojot parametru wctx. Tad konteksts tiks atgriezts atpakaļ pēc autentifikācijas parametrā *wctx*. Piemērs:
  - https://host/PFAS.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&wctx=AppContext
- Izmantojot HomeRealmDiscovery, konteksts jānorāda parametrā **context**. Piemērs:
  - https://host/LVP.STS/IdentityProviders.js?version=1.0&protocol=wsfederation&realm=https%3A%2F%2Frealm%2F&context= AppContext
- Izmantojot pieteikšanos no identitātes piegādātāja, konteksts jānorāda parametrā **cx**. Piemērs:
  - https://host/BANK.STS/SignIn.aspx?ctx=pr%3Dwsfederation%26rm%3Dhttps%253A%252F%252FappRealm%26cx%3DAppContext

7.5.16.2. Izmantojot parametru wreply

- Kontekstu var padot, izmantojot atgriešanās adresi, parametru *wreply.* Izmantojot WIF 3.5 SDK, modificējot web.config var definēt vēlamo autentifikācijas sniedzēju.

```
<federatedAuthentication>
  <wsFederation passiveRedirectEnabled="true" issuer="https://host/LVP.STS/wsfed" realm="https://realm/" requireHttps="true" reply="https://host/App?param=AppContext"
  <cookieHandler requireSsl="true" />
</federatedAuthentication>
```

Piemērs:

https://host/PFAS.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&wreply=https%3A%2F%2Fhost%2FApp%3Fparam%3DAppContext

- Izmantojot HomeRealmDiscovery kontekstu, jānorāda parametrā **reply_to**. Piemērs:

https://host/LVP.STS/IdentityProviders.js?version=1.0&protocol=wsfederation&realm=https%3A%2F%2Frealm%2F&reply_to=https%3A%2F%2Fhost%2FApp%3Fparam%3DAppContext

- Izmantojot pieteikšanos no identitātes piegādātāja kontekstu, jānorāda parametra **ry**. Piemērs:

https://host/BANK.STS/SignIn.aspx?ctx=pr%3Dwsfederation%26rm%3Dhttps%253A%252F%252FappRealm%26ry%3Dhttps%3A%2F%2Fhost%2FApp%3Fparam%3DAppContext

7.5.17.Kā norādīt pieprasījuma kodēšanas algoritmu

Parametrā *wencoding* jānorāda kodēšanas algoritms un pieprasījums jānosūta parametrā *wreq* attiecīgā kodējumā.

Piemērs:

https://host/PFAS.STS/wsfed?wa=wsignin1.0&wtrealm=https%3A%2F%2Frealm%2F&wreq=PHRydXN0OlJlcXVlc3RTZWN1cml0eVRva2VuIHhtbG5zOnRydXN0PSJodHRwOi8vZG9jcy5vYXNpcy1vcGVuLm9yZy93cy1zeC93cy10cnVzdC8yMDA1MTIiPg0KICA8dHJ1c3Q6VG9rZW5UeXBlPmh0dHA6Ly9kb2NzLm9hc2lzLW9wZW4ub3JnL2dzcy9vYXNpcy13c3Mtc2FtbC10b2tlbi1wcm9maWxlLTEuMSNTQU1MVjIuMDwvdHJ1c3Q6VG9rZW5UeXBlPg0KPC90cnVzdDpSZXF1ZXN0U2VjdXJpdHlUb2tlbj4&wencoding=base64url

7.5.18.Lietojuma konteksta informācijas pārsūtīšana caur STS

7.5.18.1. Izmantojot parametru *target*

Kontekstu var pārsūtīt, izmantojot parametru *target*. Tad konteksts tiks atgriezts atpakaļ pēc autentifikācijas parametrā *target.* Piemērs:

https://host/PFAS.STS/wsfed?providerId=https%3A%2F%2Frealm%2F&shire=https%3A%2F%2Fhost%2FApp&target=AppContext

Izmantojot pieteikšanos no identitātes piegādātāja, konteksts jānorāda parametrā **cx**. Piemērs:

https://host/BANK.STS/SignIn.aspx?ctx=pr%3Dsaml11p%26rm%3Dhttps%253A%252F%252FappRealm%26cx%3DAppContext

7.5.18.2. Izmantojot parametru shire

Kontekstu var padot izmantojot atgriešanas adresi, parametrā **shire**. Piemērs:

https://host/PFAS.STS/wsfed?providerId=https%3A%2F%2Frealm%2F&shire=3Dhttps%3A%2F%2Fhost%2FApp%3Fparam%3DAppContext&target=AppContext

Izmantojot pieteikšanu no identitātes piegādātāja, konteksts jānorāda parametrā **ry**. Piemērs:

https://host/BANK.STS/SignIn.aspx?ctx=pr%3Dsaml11p%26rm%3Dhttps%253A%252F%252FappRealm%26ry%3Dhttps%3A%2F%2Fhost%2FApp%3Fparam%3DAppContext

7.6. SAML2.0 Protocol

7.6.1. Drošības talona pieprasīšana

Jāsūta SAML2 autentifikācijas pieprasījums ar GET vai POST.

Piemērs:

```
<samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="aaf23196-1773-2113-474a-fe114412ab72"
    Version="2.0"
    IssueInstant="2004-12-05T09:21:59Z"
```

```
    AssertionConsumerServiceIndex="0"
    AttributeConsumingServiceIndex="0">
  <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
  <samlp:NameIDPolicy
    AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"/>
</samlp:AuthnRequest>
```

7.6.1.1. HTTP Redirect binding

Piemērs:

```
https://host/STS/saml2?SAMLRequest=fZFfa8IwFMXfBb9DyXvaJtZ1BqsURRC2Mabbw95ivc5Am
3TJrXPffmmLY3%2FA15Pzuyf33On8XJXBCaxTRmeEhTEJQBdmr%2FRbRp63K3pL5rPhYOpkVdYib%2FC
on%2BC9AYfDQRB4WDvRvWWksVoY6ZQTWlbgBBZik9%2FfCR7GorYGTWFK8pu6DknnwKL%2FWEetlxmR8
sBHbHJDWZqOKGdsRJM0kfQAjCUJ43KX8s78ctnIz%2Blp5xpYa4dSo1fjOKGM03i8jSeCMzGevHa2%2F
BK5MNo1FdgN2JMqPLmHc0b6WTmiVbsGoTf5qv66Zq2t60x0wXZ2RKydiCJXh3CWVV1CWJgqanfl0%2Bi
n8xutxYOvZL18NKUqPlvZR5el%2BVhYkAgZQdsA6fWVsZXE63W2itrTQ2cVaKV2CjSSqL1v9P%2FAXv4
C
```

7.6.1.2. HTTP Post bindnig

XHTML formas piemērs:

```
<form method="post" action="https://host/STS/saml2">
    <input type="hidden" name="SAMLRequest"
value="PHNhbWxwOkF1dGhuUmVxdWVzdA0KICAgIHhtbG5zOnNhbWxwPSJ1cm46b2FzaXM6bmFtZXM6d
GM6U0FNTDoyLjA6cHJvdG9jb2wiDQogICAgeG1sbnM6c2FtbD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBT
Uw6Mi4wOmFzc2VydGlvbiINCiAgICBJRD0iYWFmMjMxOTYtMTc3My0yMTEzLTQ3NGEtZmUxMTQ0MTJhY
jcyIg0KICAgIFZlcnNpb249IjIuMCINCiAgICBJc3N1ZUluc3RhbnQ9IjIwMDQtMTItMDVUMDk6MjE6N
TlaIg0KICAgIEFzc2VydGlvbkNvbnN1bWVyU2VydmljZUluZGV4PSIwIg0KICAgIEF0dHJpYnV0ZUNvb
nN1bWluZ1NlcnZpY2VJbmRleD0iMCI+DQogICAgPHNhbWw6SXNzdWVyPmh0dHBzOi8vc3AuZXhhbXBsZ
S5jb20vU0FNTDI8L3NhbWw6SXNzdWVyPg0KICAgIDxzYW1scDpOYW1lSURQb2xpY3kNCiAgICAgIEFsb
G93Q3JlYXRlPSJ0cnVlIg0KICAgICAgRm9ybWF0PSJ1cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6b
mFtZWlkLWZvcm1hdDp0cmFuc2llbnQiLz4NCiAgPC9zYW1scDpBdXRoblJlcXVlc3Q+DQo=" />
  </form>
```

7.6.2. Kā pieprasīt talonu ar nepieciešamo pieprasīto informāciju (claims)

Pieprasot autentifikāciju, jānorāda vēlamā pieprasītā informācija (RequestedAttributes) elementā RequestedAttribute. http://docs.oasis-open.org/security/saml-protoc-req-attr-req/v1.0/saml-protoc-req-attr-req-v1.0.html

```
<samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="aaf23196-1773-2113-474a-fe114412ab72"
    Version="2.0"
    IssueInstant="2004-12-05T09:21:59Z"
    AssertionConsumerServiceIndex="0"
    AttributeConsumingServiceIndex="0">
  <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
  <samlp:NameIDPolicy
    AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"/>
  <samlp:Extensions>
    <req-attr:RequestedAttributes xmlns:req-attr="urn:oasis:names:tc:SAML:protocol:ext:req-
attr" xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
      <md:RequestedAttribute isRequired="true"
Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri" />
    </req-attr:RequestedAttributes>
  </samlp:Extensions>
</samlp:AuthnRequest>
```

7.6.3. LVP.STS juridisko personu autentifikācija

Veicot autentifikāciju, jānorāda autentifikācijas konteksts, tad LVP.STS pēc lietotajā autentifikācijas piedāvās lietotajam izvelēties juridisko personu.

```
<samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="aaf23196-1773-2113-474a-fe114412ab72"
    Version="2.0"
    IssueInstant="2004-12-05T09:21:59Z"
    AssertionConsumerServiceIndex="0"
    AttributeConsumingServiceIndex="0">
  <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
  <samlp:NameIDPolicy
    AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"/>
  <samlp:RequestedAuthnContext>
<saml:AuthnContextDeclRef>https://ivis.eps.gov.lv/schema/identity/claims/legalentity</saml:A
uthnContextDeclRef>
  </samlp:RequestedAuthnContext>
</samlp:AuthnRequest
```

7.6.4. LVP.STS valsts iestāžu personu autentifikācija

Veicot autentifikāciju, jānorāda autentifikācijas konteksts, tad LVP.STS pēc lietotajā autentifikācijas piedāvās lietotajam izvelēties juridisko personu.

```
<samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="aaf23196-1773-2113-474a-fe114412ab72"
    Version="2.0"
    IssueInstant="2004-12-05T09:21:59Z"
    AssertionConsumerServiceIndex="0"
    AttributeConsumingServiceIndex="0">
  <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
  <samlp:NameIDPolicy
    AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"/>
  <samlp:RequestedAuthnContext>
<saml:AuthnContextDeclRef>https://ivis.eps.gov.lv/schema/identity/claims/govlegalentity</sam
l:AuthnContextDeclRef>
  </samlp:RequestedAuthnContext>
</samlp:AuthnRequest
```

7.6.5. LVP.STS pilnvaroto personu autentifikācija

Veicot autentifikāciju, jānorāda autentifikācijas konteksts, tad LVP.STS pēc lietotajā autentifikācijas piedāvās lietotājam izvelēties pilnvaras.

```
<samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="aaf23196-1773-2113-474a-fe114412ab72"
    Version="2.0"
    IssueInstant="2004-12-05T09:21:59Z"
    AssertionConsumerServiceIndex="0"
    AttributeConsumingServiceIndex="0">
  <saml:Issuer>https://sp.example.com/SAML2</saml:Issuer>
  <samlp:NameIDPolicy
    AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"/>
```

```
  <samlp:RequestedAuthnContext>
<saml:AuthnContextDeclRef>https://ivis.eps.gov.lv/schema/identity/claims/grantor</saml:Authn
ContextDeclRef>
  </samlp:RequestedAuthnContext>
</samlp:AuthnRequest
```

7.6.6. HomeRealmDiscovery protokola izmantošana

Jāizpilda 7.2. nodaļas nosacījumus.

Piemērs:

https://host/LVP.STS/IdentityProviders.js?version=1.0&protocol=saml2p&realm=https%3A%2F%2Frealm%2F

7.6.7. Pieteikšanās no bankas

Jāizpilda 7.1. nodaļas nosacījumus.

7.6.7.1. ACS notācija

SignIn.aspx?ctx=**URLencoded**(
pr=saml2p
&rm=**URLencoded**(https://appRealm)
[&cx=**URLencoded**(appContext)]
[&ry=**URLencoded**(https://appReplyTo)]
)

Piemērs:

https://host/BANK.STS/SignIn.aspx?ctx=pr%3Dsaml2p%26rm%3Dhttps%253A%252F%252FappRealm

7.6.7.2. ADFS notācija

SignIn.aspx?ctx=**URLencoded**(
RPID=**URLencoded**(https://appRealm)
&RealyState=[**URLencoded**(appContext)]
)

7.6.8. Atteikšana no STS

Piemērs:

```
<samlp:LogoutRequest
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  ID=" _947f5e64-8423-4464-b4b0-fa416dfe62a0"
  Version="2.0"
  IssueInstant="2017-05-31T15:43:18Z"
  Destination="https://host/STS/saml2">
  <saml:Issuer>http://localhost:4568/Default.aspx</saml:Issuer>
  <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">hamilton1@mycompany.test</saml:NameID>
  <samlp:SessionIndex>_6f6daef8-f0b5-4437-8e35-ae44ffc48cfe</samlp:SessionIndex>
</samlp:LogoutRequest>
```

7.6.8.1. HTTP Redirect binding

Piemērs:

```
https://host/STS/saml2?SAMLRequest=fZDNasMwEITvgbyD0F3%2BlR1XJKYFUzCkPdShh16CYq8
agy25XhmSt6%2Fs9BAK7U1azTermS3KvhvE3nyayb7B1wRoyaXvNIrlZUenUQsjsUWhZQ8obC2qp5e9i
LxADKOxpjYdvUP%2BJyQijLY1mpKy2FFyfOAblUDKWcajmHHuTid%2BCpiSPEwbBWkkA0reYUQH7ajzc
CTiBKVGK7V1oyDcsCBhcXgIE8FjEWYflBQuSKulXaiztQMK3z8btH51qPz5oxHN1ytCyHa%2BiMVzzGe
lE3amlt2sFjxJM78AJafOehKHy9a%2F1985vLqwZUGezdhL%2B3cLoRcuk7ZhapGKSeMAdataaGh%2Bl
n3bWaPDx%2F5am36Q%2BupZl%2BVn7W3J3dpBVIBzN6Vu4JIfU5U2ElTGVHBKXJ%2FxhmUQJ0wC50rVP
KsV3Lx%2BkevVNw%3D%3D
```

7.6.8.2. HTTP Post bindnig

XHTML formas piemērs:

```
<form method="post" action="https://host/STS/saml2">
    <input type="hidden" name="SAMLRequest"
value="PHNhbWxwOkxvZ291dFJlcXVlc3QgeG1sbnM6c2FtbHA9InVybjpvYXNpczpuYW1lczp0YzpTQ
U1MOjIuMDpwcm90b2NvbCIgeG1sbnM6c2FtbD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmFzc
2VydGlvbiIgSUQ9IiBfOTQ3ZjVlNjQtODQyMy00NDY0LWI0YjAtZmE0MTZkZmU2MmEwIiBWZXJzaW9uP
SIyLjAiIElzc3VlSW5zdGFudD0iMjAxNy0wNS0zMVQxNTo0MzoxOFoiIERlc3RpbmF0aW9uPSJodHRwc
zovL2hvc3QvU1RTL3NhbWwyIj4NCiAgIDxzYW1sOklzc3Vlcj5odHRwOi8vbG9jYWxob3N0OjQ1NjgvR
GVmYXVsdC5hc3B4PC9zYW1sOklzc3Vlcj4NCiAgIDxzYW1sOk5hbWVJRCBGb3JtYXQ9InVybjpvYXNpc
zpuYW1lczp0YzpTQU1MOjEuMTpuYW1laWQtZm9ybWF0OnVuc3BlY2lmaWVkIj5oYW1pbHRvbjFAbXljb
21wYW55LnRlc3Q8L3NhbWw6TmFtZUlEPg0KICAgPHNhbWxwOlNlc3Npb25JbmRleD5fNmY2ZGFlZjgtZ
jBiNS00NDM3LThlMzUtYWU0NGZmYzQ4Y2ZlPC9zYW1scDpTZXNzaW9uSW5kZXg+DQo=" />
  </form>
```

7.7. OAuth2/OIDC Protocol

PFAS.STS nodrošina OAuth2 protokolu, bet Lvp.Portal.IdentityServer (LVP.IDS) un Lvp.VPM.IdentitySelector (VPM vai LVP.STS) nodrošina OpenID Connect protokolu.

OAuth2 (PFAS STS) nodrošina šādas metodes:

- Token – “/oauth2/token”;
- Introspection – “/oauth2/introspect”;
- Revocation – “/oauth2/revoke”;
- Authorize – “/oauth2/authorize”;
- UserInfo – “/oauth2/userinfo”;
- Discovery – “/.well-known/openid-configuration”.

7.7.1. Discovery endpoint

Atklāšanas galapunktu var izmantot, lai izgūtu metadatus par STS - tas atgriež tādu informāciju kā izsniedzēja identifikatoru, atslēgas, atbalstītās scopes utt. Plašāku informāciju skatiet [8] specifikācijā**.**

Pieprasījuma piemērs:

```
GET /.well-known/openid-configuration
```

7.7.2. Authorize endpoint

Autorizācijas galapunktu var izmantot, lai pārlūkprogrammā pieprasītu marķierus vai autorizācijas kodus. Plašāku informāciju skat. [7] un [10] specifikācijās, Authentication Request nodaļā.

Pieprasījuma piemērs:

```
GET /connect/authorize?
    client_id=client1&
    scope=openid email api1&
```

```
response_type=id_token token&
redirect_uri=https://myapp/callback&
code_challenge=E9Melhoa2OwvFrEMTJguCHaoeK1t8URWbuGJSstw-cM&
code_challenge_method=S256
state=abc&
nonce=xyz
```

20.tabula

PARAMETRS	APRAKSTS	OBLIGĀTS/NEOBLIGĀTS
client_id	Unikālais klienta identifikators	obligāts
scope		obligāts
redirect_uri	precīzi jāatbilst vienam no šī klienta atļautajiem novirzīšanas URI	obligāts
response_type	id_token token id_token token code code is_token code id_token token	
response_mode	form_post	
state	STS atspoguļos stāvokļa vērtību marķiera atbildē, tas ir paredzēts atslēgšanās stāvoklim starp klientu un pakalpojumu sniedzēju, korelējošam pieprasījumam, atbildei un CSRF/replay aizsardzībai.	rekomendējams
nonce	STS atkārtos identitātes marķiera nonce vērtību, tas paredzēts atkārtotas aizsardzības nodrošināšanai	obligāts ja implicit grant
prompt		
code_challenge	PKCE kods	
code_challenge_method	PKCE koda hešošanas metode plain vai S256	
login_hint	var izmantot, lai iepriekš aizpildītu lietotājvārda lauku pieteikšanās lapā	
max_age	ja lietotāja pieteikšanās sesija pārsniedz maksimālo ilgumu (sekundēs), tiks attēlota pieteikšanās lietotāja saskarne	

7.7.2.1. LVP.IDS Konta maiņā

Juridiskai personai:

prompt=none&loginHint=xxxxxxxxxxxx&scope=openid profile epak_legal context_api_legal

Deleģētai personai:

prompt=none&loginHint=xxxxxxxxxxxx&scope=openid profile epak_grantor context_api_grantor

Fiziskai personai:

prompt=none&loginHint=xxxxxxxxxxxx&scope=openid profile epak context_api

7.7.3. Token endpoint

Marķiera galapunktu var izmantot, lai programmatiski pieprasītu marķierus. Plašāku informāciju skat. [6] un [10] specifikācijās Token Request nodaļā.

Pieprasījuma piemērs:

```
POST /connect/token
Content-Type application/x-www-form-urlencoded

    client_id=client1&
    client_secret=secret&
    grant_type=authorization_code&
    code=hdh922&
    redirect_uri=https://myapp.com/callback
```

21.tabula

PARAMETRS	APRAKSTS	OBLIGĀTS/NEOBLIGĀTS
client_id	Unikālais Klinta identifikators	obligāts
client_secret	Klienta slepenais vārds	
grant_type	authorization_code vai password	
scope		
redirect_uri	precīzi jāatbilst vienam no šī klienta atļautajiem novirzīšanas URI	Obligāts, ja grant_type= authorization_code
code	Autorizācijas kods	Obligāts, ja grant_type= authorization_code
code_verifier	PKCE atslēga	
username	Lietotāja vārds	Obligāts, ja grant_type= password
password	Lietotāja parole	Obligāts, ja grant_type= password

7.7.4. UserInfo endpoint

UserInfo galapunktu var izmantot, lai izgūtu lietotāja identitātes informāciju. Plašāku informāciju skat. [10] specifikācijā UserInfo Request nodaļā.

Pieprasījuma piemērs:

```
GET /connect/userinfo
Authorization: Bearer <access_token>
```

Metodes atgriezto datu struktūras piemērs:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "sub": "248289761001",
    "name": "Bob Smith",
    "given_name": "Bob",
    "family_name": "Smith",
    "role": [
        "user",
        "admin"
    ]
}
```

7.7.5. Introspection endpoint

To var izmantot, lai apstiprinātu atsauces (reference) marķierus. Plašāku informāciju skatiet [14] specifikācijā.

Pieprasījuma piemērs:

```
POST /connect/introspect
Authorization: Basic xxxyyy

token=<token>
```

22.tabula

PARAMETRS	APRAKSTS	OBLIGĀTS/NEOBLIGĀTS
token	Atsauces marķieris	obligāts

Metodes atgriezto datu struktūras piemērs:

Veiksmīgas autentifikācijas rezultātā statusa kods 200 vai nu aktīvs, vai neaktīvs marķieris:

```
{
    "active": true,
    "sub": "123"
}
```

Nezināmi vai derīguma termiņa beigu termiņi tiks atzīmēti kā neaktīvi:

```
{
    "active": false,
}
```

7.7.6. Revocation endpoint

Šis galapunkts ļauj atsaukt piekļuves marķierus (tikai atsauces (refrence) marķierus) un refresh marķierus. Plašāku informāciju skatiet [13] specifikācijā.

Pieprasījuma piemērs:

```
POST /connect/revocation
Content-Type: application/x-www-form-urlencoded
Authorization: Basic <clientId:secret>

token=45ghiukldjahdnhzdauz&token_type_hint=refresh_token
```

23.tabula

PARAMETRS	APRAKSTS	OBLIGĀTS/NEOBLIGĀTS
token	Markes atcelšanai	obligāts
token_type_hint	`access_token` vai `refresh_token`	

7.7.7. End session endpoint

Lai izmantotu sesijas beigu punktu, klienta lietojumprogramma lietotāja pārlūkprogrammu novirzīs uz beigu sesijas URL. Izrakstīšanā var piedalīties visas lietojumprogrammas, kurās lietotājs sesijas laikā ir pieteicies, izmantojot pārlūku. Plašāku informāciju skatiet [9] specifikācijā.

Pieprasījuma piemērs:

```
GET
/connect/endsession?id_token_hint=eyJhbGciOiJSUzI1NiIsImtpZCI6IjdlOGFkZmMzMjU1OT
EyNzI0ZDY4NWZmYmIwOThjNDEyIiwidHlwIjoiSldUIn0.eyJuYmYiOjE0OTE3NjUzMjEsImV4cCI6MT
Q5MTc2NTYyMSwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo1MDAwIiwiYXVkIjoianNfb2lkYyIsIm5vbm
NlIjoiYTQwNGFjN2NjYWEwNGFmNzkzNmJjYTkyNTJkYTRhODUiLCJpYXQiOjE0OTE3NjUzMjEsInNpZC
```

```
I6IjI2YTYzNWVmOTQ2ZjRiZGU3ZWUzMzQ2ZjFmMWY1NTZjIiwic3ViIjoiODg0MjExMTMiLCJhdXRoX3
RpbWUiOjE0OTE3NjUzMTksImlkcCI6ImxvY2FsIiwiYW1yIjpbInB3ZCJdfQ.STzOWoeVYMtZdRAeRT9
5cMYEmClixWkmGwVH2Yyiks9BETotbSZiSfgE5kRh72kghN78N3-
RgCTUmM2edB3bZx4H5ut3wWsBnZtQ2JLfhTwJAjaLE9Ykt68ovNJySbm8hjZhHzPWKh55jzshivQvTX0
GdtlbcDoEA1oNONxHkpDIcr3pRoGi6YveEAFsGOeSQwzT76aId-rAALhFPkyKnVc-
uB8IHtGNSyRWLFhwVqAdS3fRNO7iIs5hYRxeFSU7a5ZuUqZ6RRi-bcDhI-
djKO5uAwiyhfpbpYcaY_TxXWoCmq8N8uAw9zqFsQUwcXymfOAi2UF3eFZt02hBu-
shKA&post_logout_redirect_uri=http%3A%2F%2Flocalhost%3A7017%2Findex.html
```

24.tabula

PARAMETRS	APRAKSTS	OBLIGĀTS/NEOBLIGĀTS
id_token_hint	Kad lietotājs tiek novirzīts uz galapunktu, viņam tiks piedāvāts, vai viņš patiešām vēlas izrakstīties.	obligāts
post_logout_redirect_uri	To var izmantot, lai ļautu lietotājam pēc izrakstīšanās novirzīt atpakaļ klientam. Vērtībai jāatbilst vienam no klienta iepriekš konfigurētajiem PostLogoutRedirectUris	
state	Pēc lietotāja novirzīšanas atpakaļ klientam tas tiks atgriezts klientam kā vaicājuma virknes parametrs. Parasti klienti to izmanto, lai novirzītu abos virzienos.	

7.7.8. Check session endpoint

Galapunkts paredzēts, lai nodrošinātu klienta lietojumam iespēju fona režīmā veikt periodisku pārbaudi vai lietotājs joprojām ir autentificējies. Šī adrese jānorāda kā *iframe* avots un jāizmanto javascript window.postMessage pieprasījumi, lai noteiktu vai autentifikācijas sesijas status ir mainījies. Izsaukumu jāveic no tā paša avota (origin) uz kuru tika atgriezts autentifikācijas talons – *redirect_uri* norādītā vērtība *Authorize* izsaukumā. Izsaukumā tiek padota teksta virkne kura tiek veidota apvienojot *client_id* (klienta identifikatoru) un *session_state* vērtības. Atbildē tik atgriezta teksta virkne:

- “unchanged” – autentificētā lietotāja sesijas status nav mainījies.
- “changed” – autentificētā lietotāja sesijas status ir mainījies - tā ir beigusies.
- “error” – pieprasījums ir kļūdais.

Plašāku informāciju skatiet [16] specifikācijā.

iframe tag piemērs:

```
<iframe                                                    id="check-session-iframe"
src="https://vraa.test.lv/Portal.IdentityServer/connect/checksession          "
style="display: none"/>
```

Javascript izsaukuma piemērs:

```
// Izveido ziņojumu
var message = client_id + " " + session_state;

// Nosūta ziņojumu ar POST uz OpenID providera iframe
var idpOrigin = " https://vraa.test.lv/Portal.IdentityServer";
var targetWindow = document.getElementById("op").contentWindow;
targetWindow.postMessage(message, idpOrigin);
```

25.tabula

PARAMETRS	APRAKSTS	OBLIGĀTS/NEOBLIGĀTS
client_id	Klienta identifikators	obligāts
session_state	Parametra *session_state* no Authorize izsaukuma atbildes	obligāts
idpOrigin	Autentifikācijas sniedzēja adrese (*Origin* daļa)	obligāts

7.8. Automatizēto testu palaišana

Autentifikācijas un autorizācijas process ir visai apjomīgs, līdz ar to testējot portāla veiktspēju, jāizņem autentifikācijas soļi. Citādi testa lietotnes veiktspēja var aiziet līdz autentifikācijas un autorizācijas iespējām.

7.8.1. Testēšana ar vienotās pietiekšanās moduli (LVP.STS)

Portāliem, kuri pieslēgti pie vienotās pieteikšanās moduļa (LVP.STS) iespējami divi veiktspējas testēšanas scenāriji.

Testēšana ar autentificētu lietotāju:

3. Veicam autentifikāciju;
   - 3.1. Testa adapteris;
   - 3.2. Vienotās pieteikšanas modulis;
   - 3.3. Testējamais portāls;
4. Testējam lietotnes veiktspēju.

Testēšana ar lietotāju autentifikāciju (autentifikācija ar visu apjomīgo procesu):

1. Veicam autentifikāciju;
   - 1.1. Testa adapteris;
   - 1.2. Vienotās pieteikšanas modulis;
2. Testējamais portāls (izslēdzam token reply detection, lai varētu atbildi izmantot vairākas reizes);
3. Testējam lietotnes veiktspēju.

Test.STS var izmantot Banku autentifikācijas un neapliecinātās identitātes imitācijai. Lai imitētu lietotāju, jānorāda lietotāja vārds un parole, izmantojot HTTP Basic Auth **Error! Reference source not found.** un norādot personas kodu vai e-pastu parametrā pk. Parametrs pk var būt nosūtīts arī POST formā.

Piemērs:

https://host/Test.STS/wsfed?...&pk=1111111111

Header: Basic {TOKEN}, kur {TOKEN} = base64(username + ':' + password)

7.8.2. Testēšana ar PFAS.STS

Portālu, kas pieslēgti pie PFAS.STS testēšanas scenāriji.

Testēšana ar autentificētu lietotāju:

1. Veicam autentifikāciju;
    1.1. PFAS.STS;
    1.2. testējamais portāls;
2. Testējam lietotnes veiktspēju.

Testēšana sakot ar lietotāju autentifikāciju (autentifikācija ar visu apjomīgo procesu):

1. Veicam autentifikāciju;
    1.1. PFAS.STS;
2. Testējamais portāls (izslēdzam token reply detection, lai varētu atbildi izmantot vairākas reizes);
3. Testējam lietotnes veiktspēju.

Lai vienkāršotu autentifikācijas procesu, PFAS.STS jāieslēdz HTTP Basic Auth **Error! Reference source not found.** autentifikāciju.

```
<ivis.pfas.sts>
  <passiveAuthenticationTypes basic="true" ... />
</ivis.pfas.sts>
```

Piemērs:

https://host/PFAS.STS/STS/Basic/BasicSignIn.aspx?...

Header: Basic {TOKEN}, kur {TOKEN} = base64(username + ':' + password)

wresult - vērtību nevar noģenerēt, to var dabūt testiem uz notiktu laiku (pēc noklusējumā 2 stundas).

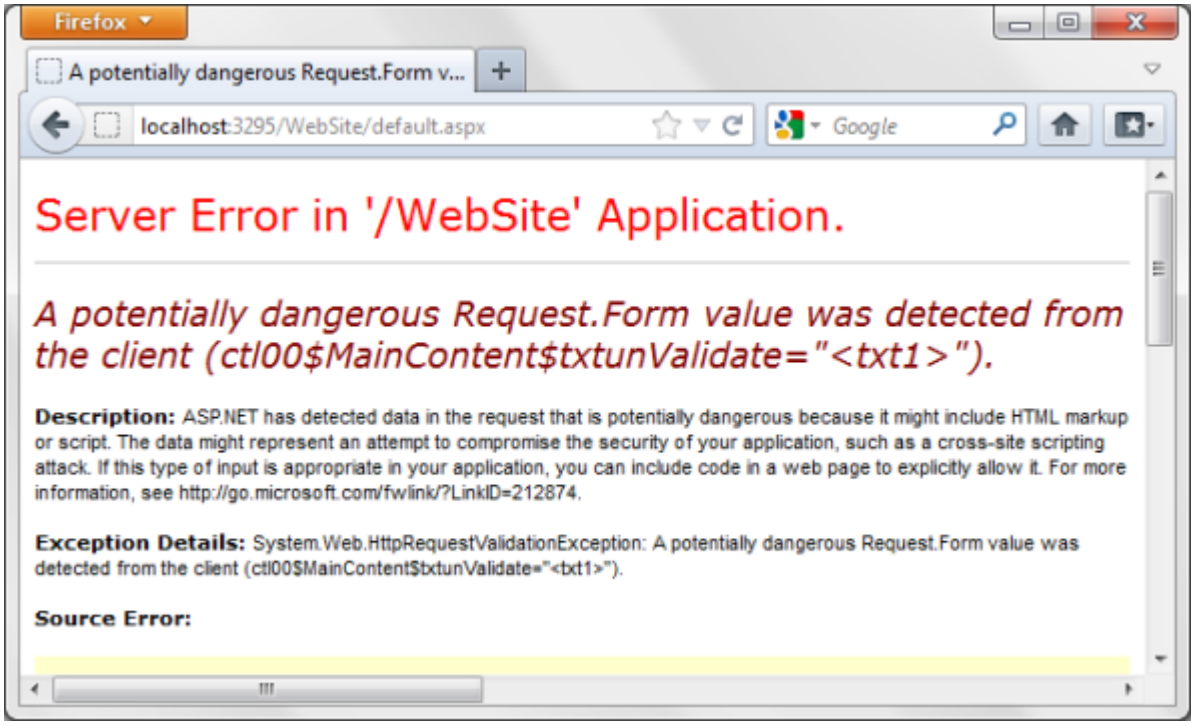
wctx – ir kritiska vērtība.

8. Savietojamība

8.1. Zināmie ierobežojumi

8.1.1. WS-Federation ziņojuma drošības izņēmums

Tā kā WS-Federation atbilde satur XML, ASP.NET 4.0 iebūvētais pieprasījuma pārbaudītājs izraisīs izņēmumu.



10.attēls. Pieprasījuma pārbaudītāja ziņojums

Lai atrisinātu šo problēmu, ir jāizslēdz ziņojuma validācija.

```
<httpRuntime
requestValidationType="Abc.IdentityModel.Protocols.WSFederation.WsFedRequestValidator,
Abc.IdentityModel"/>
```

Skat. arī.9.4. nodaļu.

8.1.2. Apache CXF kļūda

Izmantojot drošos .NET Web servisus no Apache parādās kļūdas paziņojumi:

java.lang.IllegalArgumentException: sp:HttpsToken/wsp:Policy must have a value vai

Failed to build the policy 'PolicyName':sp:HttpsToken/wsp:Policy must have a value

Tas notiek tāpēc, ka .NET drošajā Web servisa, kas attiecās uz standartu WS-SecurityPolicy v1.2 netiek ģenerēts elements wsp:Policy.

WCF servisa ieejas punktiem, kuri konfigurēti ws2007FederationHttpBinding, konfigurācijā pieliek paplašinājums WSSecurityPolicy12BugFixBehavior no bibliotēkas Abc.IdentityModel:

```
<system.serviceModel>
  <extensions>
    <behaviorExtensions>
```

```
        <add                                                    name="wsSecurityPolicy12BugFix"
type="Abc.IdentityModel.Protocols.WSSecurityPolicy.WSSecurityPolicy12BugFixBehavior,
Abc.IdentityModel" />
      </behaviorExtensions>
    </extensions>
    <behaviors>
      <endpointBehaviors>
        <behavior name="wsSecurityPolicy12BugFix.Behavior">
          <wsSecurityPolicy12BugFix />
        </behavior>
      </endpointBehaviors>
    </behaviors>
    <services>
      <service behaviorConfiguration="WcfService.Behavior" name="WcfService">
        <endpoint    address="ws2007FederationNoSct"    binding="ws2007FederationHttpBinding"
bindingConfiguration="ws2007FederationNoSct"                    name="ws2007FederationNoSct"
contract="IWcfServiceContract" behaviorConfiguration="wsSecurityPolicy12BugFix.Behavior" />
    </services>
  </system.serviceModel>
```

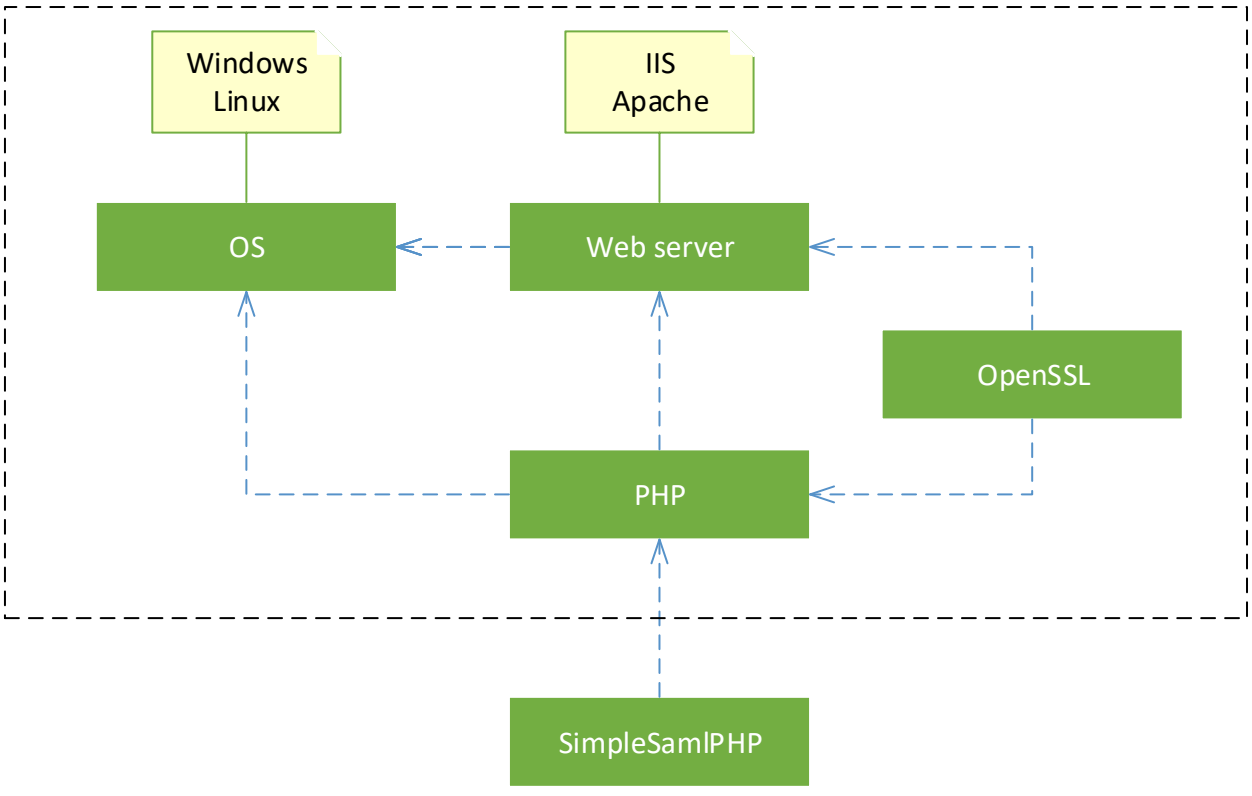
Skat. arī 9.3. nodaļu.

8.1.3. SimpleSamlPHP uzstādīšana

Uzstādot SimpleSamlPHP jābūt instalētam un nokonfigurētam:

- Web serverim ar ieslēgtu SSL;
- PHP.

Operētājsistēmai, Web serverim, PHP, OpenSSL bibliotēkai jābūt savstarpēji savietojamām.

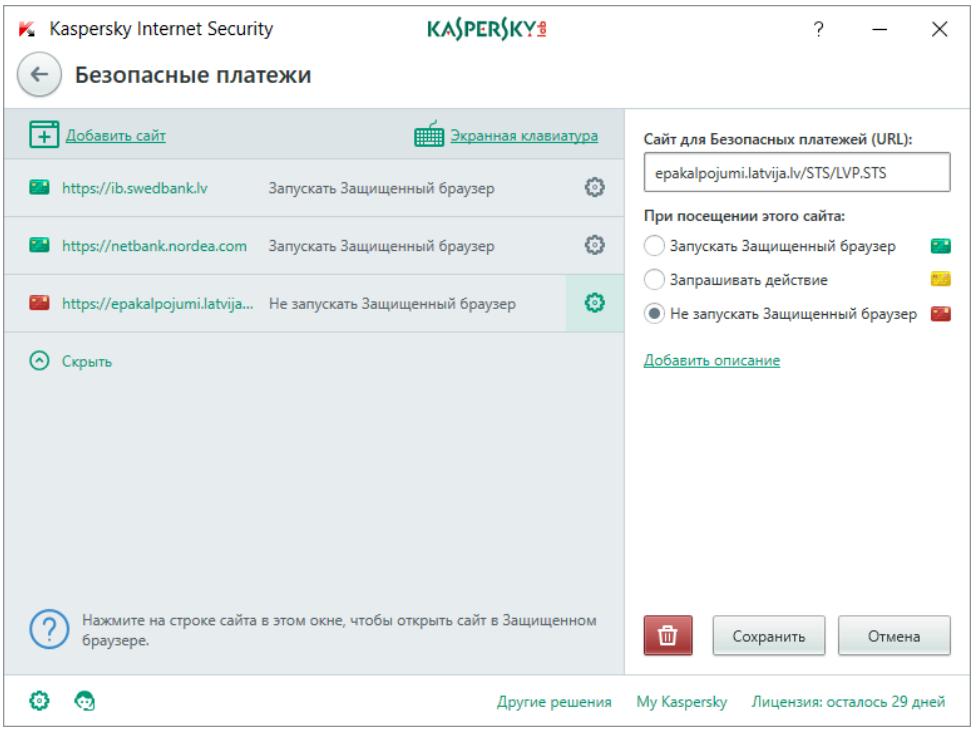


11.attēls. SimpleSamlPHP atkarību diagramma

8.1.4. Kaspersky Internet Security sadarbība ar LVP.STS

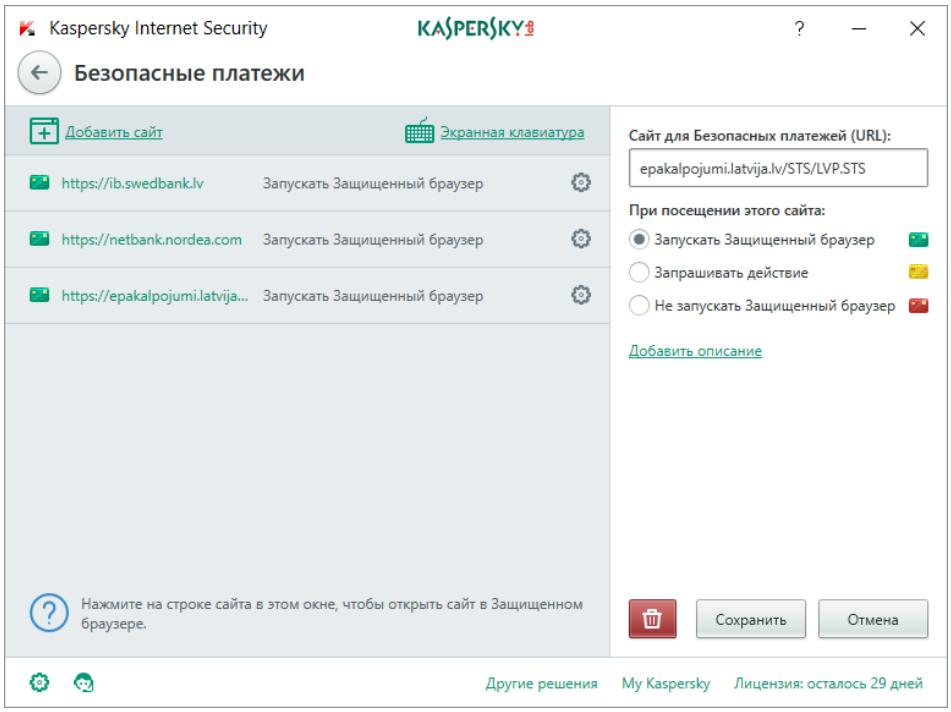
Kasperski Internet Security notiek pāradresācija uz banku (vai tiek atvērta bankas web lapa) drošajā pārlūkā, pēc autentifikācijas bankā tiek attēlots kļūdas paziņojums.

Klientam jāatslēdz drošo pārlūka režīmu vienotajam pieteikšanās modulim. Kā arī nevajag laist drošo režīmu konkrētajam URL: epakalpojumi.latvija.lv/STS/LVP.STS (skat. 12. attēlā).



12.attēls. Kasperski Internet Secrity

Ja tomēr klientam gribas strādāt drošajā režīma, tad vienotajam pieteikšanas modulim jāieslēdz drošais pārlūks, jālaiž drošo režīmu konkrētajam URL: epakalpojumi.latvija.lv/STS/LVP.STS (skat. https://support.kaspersky.ru/12099#block2 un 13. attēlu.)



13.attēls. Kasperski Internet Secrity I

8.2. SAML rīka izmantošanas piemēri

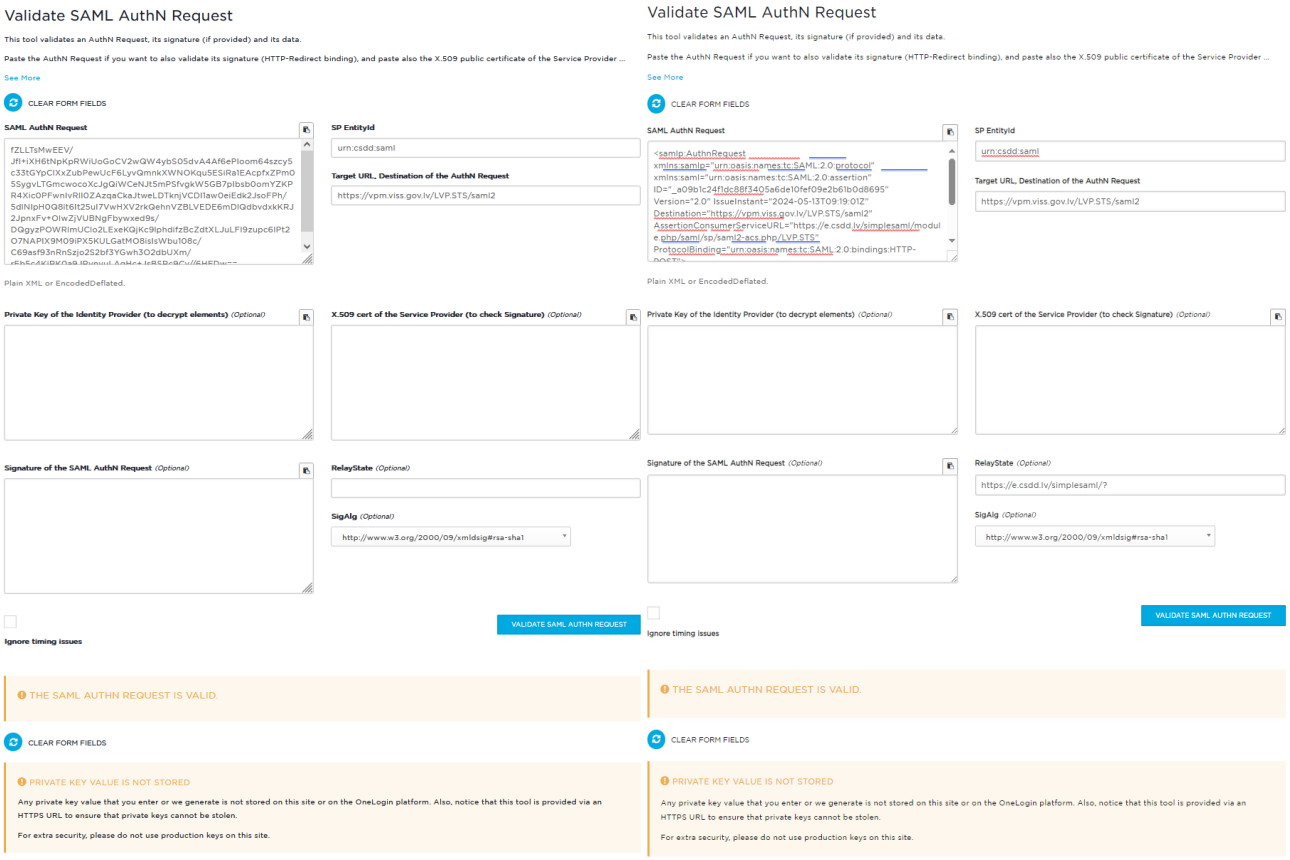
8.2.1. SAML2 pieprasījuma pārbaude

Izmantojās validācijas lapu [Validate SAML AuthN Request Online Tool | SAMLTool.com]

Laukā “SP EntityId” ievadam SAML saņēmējā identifikatoru

Laukā “Target URL, Destination of the AuthN Response” ievadam saņēmēja adresi

Laukā SAML AuthN Request ieliekam SAMLRequest laika vērtību Deflate kodētu Base64 formāta vai dekodētu XML formātā (**Svarīgi XML formatējums bojā parakstu**)



8.2.2. SAML2 atbildes pārbaude

Izmantojās validācijas lapu [SAML Response Validator - Validate SAML Metadata, Signatures & Certificates (samltool.com)]

Laukā “IdP EntityId” ievadam VPM identifikatoru, to var dabūt no metadatiem. Skaties 6.1.1.1

Laukā “X.509 cert of the IdP” ievadam VPM parakstīšanas sertifikātu Base64 formāta, to var dabūt no metadatiem. Skaties 6.1.2.2

Laukā “SP EntityId” ievadam SAML saņēmējā identifikatoru

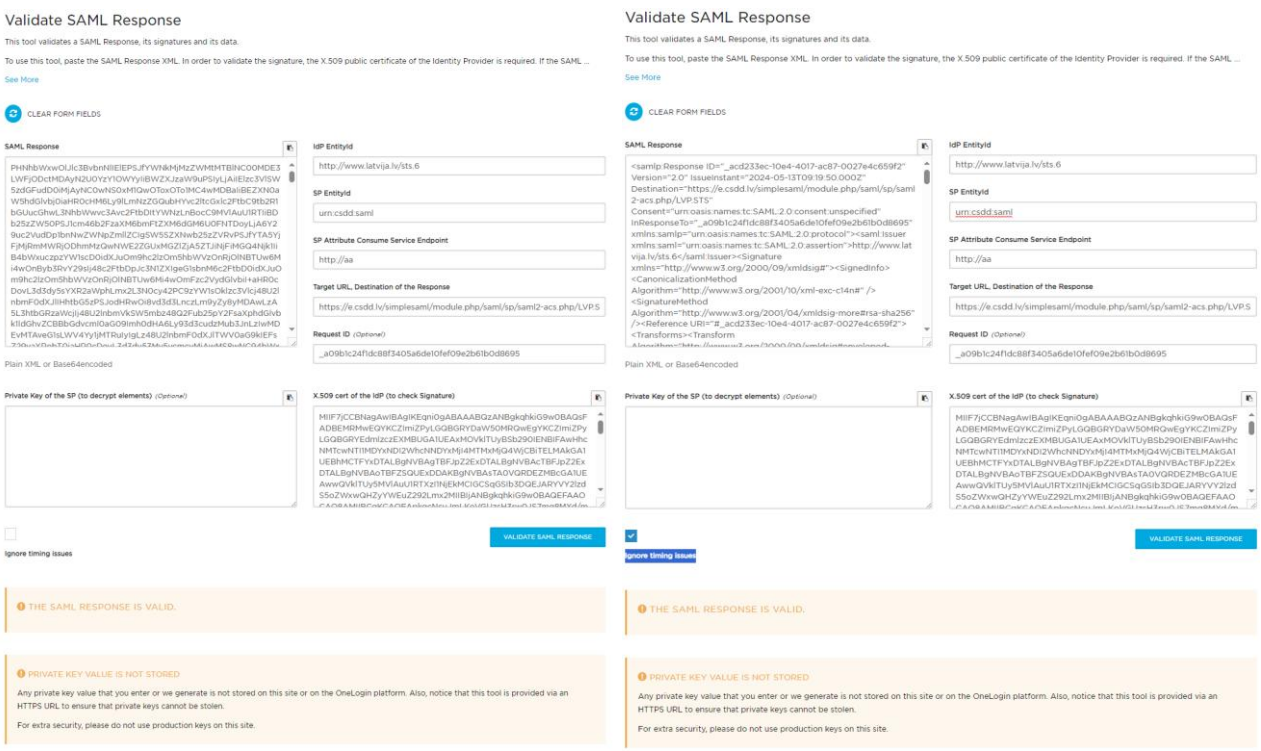
Laukā “Target URL, Destination of the Response” ievadam saņēmēja adresi

Laukā “Reques ID” ievadam pieprasījuma identifikatoru, ja tāds zināms

Laukā “Private Key of the SP” ievadam Base64 kodētu sertifikātu, ja SAML atbilde ir šifrēta

Laukā “SAML response” ieliekam SAMLResponse laika vērtību Base64 formāta vai dekodētu XML formātā (**Svarīgi XML formātejums bojā parakstu**)

Uzlieciet izvēles rūtiņu “Ignore timing issues”.



8.2.3. SAML2 logout atbildes pārbaude

Izmantojās validācijas lapu https://www.samltool.com/validate_logout_req.php

Laukā “EntityId of the source” ievadam SAML saņēmējā identifikātoru

Laukā “Target URL, Destination of the Logout Request” ievadam VPM adresi

Laukā “SAML Logout Request” ieliekam SAMLRequest laika vērtību Deflate kodētu Base64 formāta vai dekodētu XML fromāta (**Svarīgi XML formātejums bojā parakstu**)

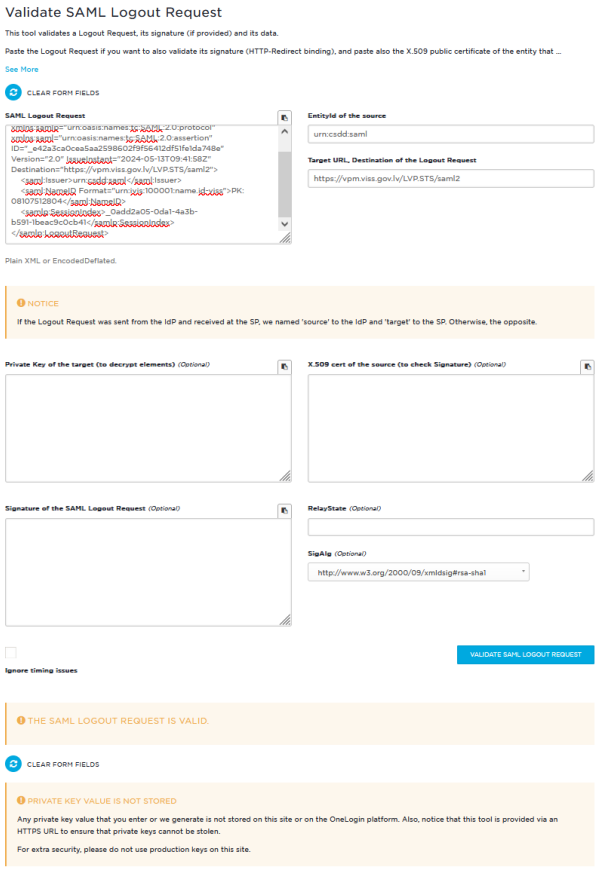
Ja tiek izmantots HTTP-Redirect binding:

Laukā “X.509 cert of the source” ievadam saņēmējā parakstīšanas sertifikātu Base64 formāta

Laukā “RealyState” ieliekam “RelayState” laika vērtību

Laukā “Signature of the SAML Logout Request” ieliekam “Signature” laika vērtību

No izvēlnes “SigAlg” izvelamies atbilstošu algoritmu no “SigAlg” laika vērtības



8.2.4. SAML2 logout pieprasījuma pārbaude

Izmantojās validācijas lapu https://www.samltool.com/validate_logout_res.php

Laukā “EntityId of the source” ievadam VPM identifikatoru, to var dabūt no metadatiem. Skaties 6.1.1.1

Laukā “Target URL, Destination of the Logout Response” ievadam saņēmēja adresi

Laukā “SAML Logout Response” ieliekam SAMLResponse laika vērtību Deflate kodētu Base64 formāta vai dekodētu XML fromāta (**Svarīgi XML formātejums bojā parakstu**)

Laukā “Reques ID” ievadam pieprasījuma identifikatoru, ja tāds zināms

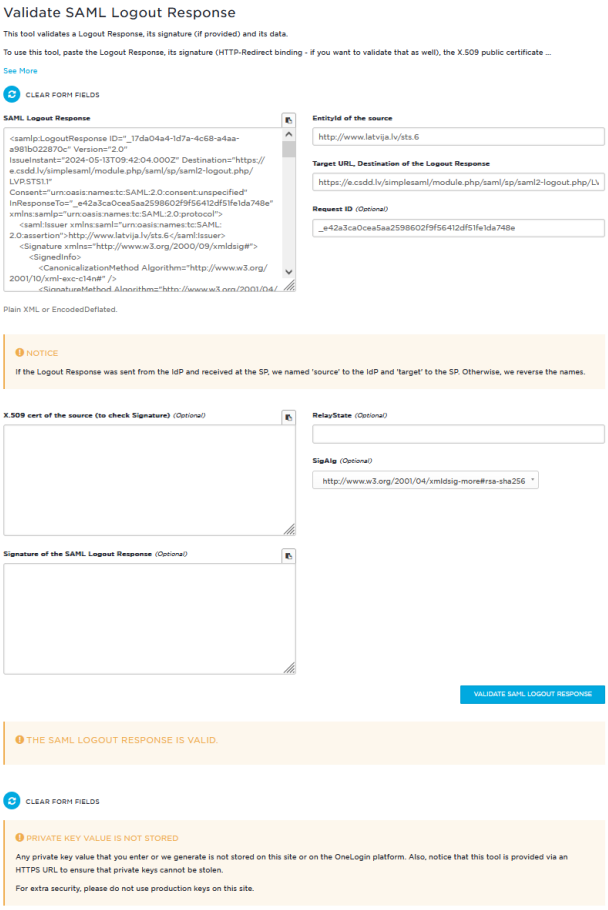
Ja tiek izmantots HTTP-Redirect binding:

Laukā “X.509 cert of the source” ievadam VPM parakstīšanas sertifikātu Base64 formāta, to var dabūt no metadatiem. Skaties 6.1.2.2

Laukā “RealyState” ieliekam RelayState laika vērtību

Laukā “Signature of the SAML Logout Response” ieliekam “Signature” laika vērtību

No izvēlnes “SigAlg” izvēlamies atbilstošu algoritmu no “SigAlg” laika vērtības


Validate SAML Logout Response
This tool validates a Logout Response, its signature (if provided) and its data.
To use this tool, paste the Logout Response, its signature (HTTP-Redirect binding - if you want to validate that as well), the X.509 public certificate ...
See More
CLEAR FORM FIELDS
SAML Logout Response
Entityid of the source
http://www.latvija.lv/sts.6
Target URL, Destination of the Logout Response
Request ID (Optional)
_e42a3ca0cea5aa2598602f9f56412df51fe1da748e
Plain XML or EncodedDeflated.
NOTICE
If the Logout Response was sent from the IdP and received at the SP, we named 'source' to the IdP and 'target' to the SP. Otherwise, we reverse the names.
X.509 cert of the source (to check Signature) (Optional)
RelayState (Optional)
SigAlg (Optional)
http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
Signature of the SAML Logout Response (Optional)
VALIDATE SAML LOGOUT RESPONSE
THE SAML LOGOUT RESPONSE IS VALID.
CLEAR FORM FIELDS
PRIVATE KEY VALUE IS NOT STORED
Any private key value that you enter or we generate is not stored on this site or on the OneLogin platform. Also, notice that this tool is provided via an HTTPS URL to ensure that private keys cannot be stolen.
For extra security, please do not use production keys on this site.

9. Pileikumi

9.1. HTML lapas saites izveidošana autentifikācijas datu saņemšanai no bankas

GenerateContext.html V1.01 pamatkodi

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>IdpInitiated SignOn Context Generator</title>
    <style type="text/css">
        .textbox {width: 500px;}
    </style>
    <script type="text/javascript">
        function SubmitMeACS(stsId, stsProt, appId, appProt, appCtx) {
            if (!stsId || !appId) {
                alert("The textbox should not be empty");
                return;
            }

            var ApplicatoinId = encodeURIComponent(appId);
            var ApplicatoinContext = encodeURIComponent(appCtx);
            var ApplicatoinProtocol = encodeURIComponent(appProt);

            var LvpStsId = encodeURIComponent(stsId);
            var stsCtx = "rm=" + ApplicatoinId + "&pr=" + ApplicatoinProtocol;
            if (ApplicatoinContext) {
                stsCtx = stsCtx + "&cx=" + ApplicatoinContext;
            }

            var LvpStsContext = encodeURIComponent(stsCtx);
            var LvpStsProt = encodeURIComponent(appProt);

            var bankCtx = "rm=" + LvpStsId + "&pr=" + LvpStsProt + "&cx=" + LvpStsContext;
            document.getElementById('lblDisplay').innerHTML = bankCtx;
            document.getElementById('lblBank').innerHTML = "SignIn.aspx?ctx=" +
encodeURIComponent(bankCtx)
        }
    </script>
</head>
<body style="background-color: blue; color: white">
    <p><br />
       <strong>Context Generator for IDP Initiated Signon </strong></p>
    <table>
        <tr>
            <td>
                LVP.STS Identifier
                <label style="color:red">*</label>
            </td>
            <td>
```

```
                <input   type="text"   value="http://www.latvija.lv/trust"   class="textbox"
id="txtLVPSTSRPID" />
            </td>
            <td>
                <select id="optLVPSTSPROT" >
                    <option value="wsfederation">Ws-Federation</option>
                </select>
            </td>
        </tr>
        <tr>
            <td> </td>
            <td> </td>
            <td> </td>
        </tr>
        <tr>
            <td>
                App Identifier
                <label style="color:red">*</label>
            </td>
            <td>
                <input     type="text"     value="http://www.latvija.lv"     class="textbox"
id="txtLVPRPID" />
            </td>
            <td>
                <select id="optLVPPROT" >
                    <option value="wsfederation">Ws-Federation</option>
                </select>
             </td>
        </tr>
        <tr>
            <td> </td>
            <td> </td>
            <td> </td>
        </tr>
        <tr>
            <td>App Context</td>
            <td>
                <input     type="text"     value="/lv/Epakalpojumi/EP00"     class="textbox"
id="txtContext" />
            </td>
            <td></td>
        </tr>
    </table>
    <br />
    <strong>
    <input                                                                    type="button"
onclick="SubmitMeACS(document.getElementById('txtLVPSTSRPID').value,
document.getElementById('optLVPSTSPROT').value, document.getElementById('txtLVPRPID').value,
document.getElementById('optLVPPROT').value,    document.getElementById('txtContext').value)"
value="Generate URL" /></strong>
    <br />
    <hr /><strong>Context:</strong><br /><br />
    <label id="lblDisplay"></label><br /><br />
    <strong>Bank Sample:</strong><br /><br />
```

```
    <label id="lblBank"></label>
</body>
</html>
```

9.2. Testa adaptera pamata kodi

```
// Izveidojam klasi bāzētu uz klases BankAdapterBase
   public class MyBankAdapter : BankAdapterBase {

// Piešķiram autentifikācijas metodi formātā URN:IVIS:100001:AM.BANK-<name>
        protected override string AuthenticationMethod {
            get { return "URN:IVIS:100001:AM.BANK-MYBANKA"; }
        }

// Funkcija kas pārbauda, ka tas ir atbilde no testa bankas
// Mūsu gadījumā atbilde atnāk kā POST
        private static bool IsBankResponse(HttpRequestBase request) {
            return request.HttpMethod == "POST";
        }

// Izveidojam funkciju kas pārbauda atbildi no bankas un atgriež AuthorizationState, ja
lietotājs bija autentificēts, vai CancelAuthorizationStatem ja tika autentifikācijā bija
atcelta
        public override IAuthorizationState ProcessUserAuthorization(HttpRequestBase request
= null) {
            if (request == null) {
                request = this.GetRequestFromContext();
            }

            if (!IsBankResponse(request)) {
                return null;
            }

            if (request.Form["Cancel"] == null) {
                return new AuthorizationState(request.Form["PK"], request.Form["FN"],
request.Form["LN"]) { Context = request["CT"] };
            }
            else {
                return new CancelAuthorizationState() { Context = request["CT"] };
            }
        }

        /// <inheritdoc/>
        public override BankRequestSpecification GetBankRequestSpecification(string context,
Uri returnTo) {
            if (returnTo == null) {
                returnTo = new
Uri(this.GetRequestFromContext().Url.GetLeftPart(UriPartial.Path));
            }

// Izveidojam funkciju kas veido pieprasījumu uz banku
// Klase BankRequestSpecification izmantojas pieprasījuma izveidei un aizpildās no // bankas
pieprasījumā specifikācijas. Mūsu gadījumā metode GET, ar parametru CT

            var myBankUrl = this.GetRequestFromContext().Url.GetLeftPart(UriPartial.Path);
            myBankUrl = myBankUrl.Substring(0, myBankUrl.LastIndexOf('/')) +
"/STS/Abc.Bank.MyBankAdapter.html";
            return new BankRequestSpecification("GET", new Uri(myBankUrl), new
KeyValuePair<string, string>[] { new KeyValuePair<string, string>("CT", context), new
KeyValuePair<string, string>("ret", returnTo.AbsoluteUri) });
        }
    }
```

Datnē SignIn.apsx tiek pielietota testa adaptera klase

```
    private static readonly MyBankAdapter client = new MyBankAdapter();

    protected void Page_Load(object sender, EventArgs e) {
// Pārbaudām vai izsaukums nāk no bankas
        IAuthorizationState authorization = client.ProcessUserAuthorization(null);
        if (authorization == null) {
            // Save current passive context
            if (!this.PassiveContext.IsRequest) {
                throw new
Microsoft.IdentityModel.SecurityTokenService.RequestFailedException((string)this.GetGlobalRe
sourceObject("SR", "ID002"));
            }

// Tiek saglabāts sakuma pieprasījums (kookies) un noģenerēta unikālā konteksta vērtība
            var ctx = Abc.IdentityModel.Web.PassiveAuthentication.SavePassiveContext();


// Ar medoti GetBankRequestSpecification jāizveido pieprasījumu pie bankas.
// Piemēram parametrs ‘nonce’ Swedbankai.
// returnTo parametrs nav obligāts, domāts tam ka banka var atgriezt atbildi uz citu URL.

            // Kick off authorization request
            client.RequestUserAuthorization(ctx, null);
        }
        else {
// No banka atbildes dabūjam to pašu contexta vērtību ko sūtījām uz banku.
// ‘nonce’ Swedbankai
            // Restore passive context
            var ctx = authorization.Context;
            Abc.IdentityModel.Web.PassiveAuthentication.RestorePassiveContext(ctx, true);

            if (!this.PassiveContext.IsRequest) {
                throw new
Microsoft.IdentityModel.SecurityTokenService.RequestFailedException((string)this.GetGlobalRe
sourceObject("SR", "ID002"));
            }

// No banka atbildes (personas kods, vārds, uzvārds) dabūjam Principal un autentificējam pēc
ta.

            if (authorization is AuthorizationState) {
                var principal = client.Authorize(authorization);
                PassiveAuthentication.SignIn(new SessionSecurityToken(principal));
            }
            else {
                PassiveAuthentication.ReturnError(null);
            }
        }
    }
```

9.3. WS-SecurityPolicy V1.2 labojums

Klases WSSecurityPolicy12BugFixBehavior.cs kods

```
// ----------------------------------------------------------------------------
// <copyright file="WSSecurityPolicy12BugFixBehavior.cs" company="ABC software">
//    Copyright © ABC SOFTWARE. All rights reserved.
//    The source code or its parts to use, reproduce, transfer, copy or
```

```
//    keep in an electronic form only from written agreement ABC SOFTWARE.
// </copyright>
// ---------------------------------------------------------------------------

namespace Abc.Samples {
    using System;
    using System.Diagnostics.Contracts;
    using System.ServiceModel.Channels;
    using System.ServiceModel.Configuration;
    using System.ServiceModel.Description;
    using System.ServiceModel.Dispatcher;
    using System.Xml;

    /// <summary>
    /// The WS-SecurityPoloicy v1.2 Bug Fix behaviour.
    /// </summary>
    public class WSSecurityPolicy12BugFixBehavior : BehaviorExtensionElement,
IWsdlExportExtension, IEndpointBehavior {
        /// <summary>
        /// Gets the type of behavior.
        /// </summary>
        /// <returns>A <see cref="T:System.Type" />.</returns>
        public override Type BehaviorType {
            get {
                return typeof(WSSecurityPolicy12BugFixBehavior);
            }
        }

        /// <summary>
        /// Implement to pass data at runtime to bindings to support custom
behavior.
        /// </summary>
        /// <param name="endpoint">The endpoint to modify.</param>
        /// <param name="bindingParameters">The objects that binding elements
require to support the behavior.</param>
        public virtual void AddBindingParameters(ServiceEndpoint endpoint,
BindingParameterCollection bindingParameters) {
        }

        /// <summary>
        /// Implements a modification or extension of the client across an
endpoint.
        /// </summary>
        /// <param name="endpoint">The endpoint that is to be customized.</param>
        /// <param name="clientRuntime">The client runtime to be
customized.</param>
        public virtual void ApplyClientBehavior(ServiceEndpoint endpoint,
ClientRuntime clientRuntime) {
        }

        /// <summary>
        /// Implements a modification or extension of the service across an
endpoint.
        /// </summary>
```

```
        /// <param name="endpoint">The endpoint that exposes the contract.</param>
        /// <param name="endpointDispatcher">The endpoint dispatcher to be modified or extended.</param>
        public virtual void ApplyDispatchBehavior(ServiceEndpoint endpoint, EndpointDispatcher endpointDispatcher) {
        }

        /// <summary>
        /// Writes custom Web Services Description Language (WSDL) elements into the generated WSDL for a contract.
        /// </summary>
        /// <param name="exporter">The <see cref="T:System.ServiceModel.Description.WsdlExporter"/> that exports the contract information.</param>
        /// <param name="context">Provides mappings from exported WSDL elements to the contract description.</param>
        public virtual void ExportContract(WsdlExporter exporter, WsdlContractConversionContext context) {
            throw new NotImplementedException();
        }

        /// <summary>
        /// Writes custom Web Services Description Language (WSDL) elements into the generated WSDL for an endpoint.
        /// </summary>
        /// <param name="exporter">The <see cref="T:System.ServiceModel.Description.WsdlExporter"/> that exports the endpoint information.</param>
        /// <param name="context">Provides mappings from exported WSDL elements to the endpoint description.</param>
        public virtual void ExportEndpoint(WsdlExporter exporter, WsdlEndpointConversionContext context) {
            this.WSSecurityPolicy12BugFix(context);
        }

        /// <summary>
        /// Implement to confirm that the endpoint meets some intended criteria.
        /// </summary>
        /// <param name="endpoint">The endpoint to validate.</param>
        public virtual void Validate(ServiceEndpoint endpoint) {
        }

        /// <summary>
        /// WS-SecurityPolicy12 bug fix.
        /// </summary>
        /// <param name="context">Provides mappings from exported WSDL elements to the endpoint description.</param>
        protected void WSSecurityPolicy12BugFix(WsdlEndpointConversionContext context) {
            Contract.Requires<ArgumentNullException>(context != null);

            var nsmanager = new XmlNamespaceManager(new NameTable());
            nsmanager.AddNamespace("sp", "http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702");
```

```
            nsmanager.AddNamespace("wsp",
"http://schemas.xmlsoap.org/ws/2004/09/policy");

            foreach (object obj in
context.WsdlBinding.ServiceDescription.Extensions) {
                XmlElement element = obj as XmlElement;
                if (element != null && element.LocalName == "Policy" &&
element.NamespaceURI == "http://schemas.xmlsoap.org/ws/2004/09/policy") {
                    var keyValueTokens =
element.SelectNodes("//sp:KeyValueToken", nsmanager);
                    if (keyValueTokens != null) {
                        for (int i = 0; i < keyValueTokens.Count; i++) {
                            var node = keyValueTokens[i];
                            if (node["Policy",
"http://schemas.xmlsoap.org/ws/2004/09/policy"] == null) {

node.AppendChild(element.OwnerDocument.CreateElement("wsp", "Policy",
"http://schemas.xmlsoap.org/ws/2004/09/policy"));
                            }
                        }
                    }

                    var httpsTokens = element.SelectNodes("//sp:HttpsToken",
nsmanager);
                    if (httpsTokens != null) {
                        for (int i = 0; i < httpsTokens.Count; i++) {
                            var node = httpsTokens[i];
                            if (node["Policy",
"http://schemas.xmlsoap.org/ws/2004/09/policy"] == null) {

node.AppendChild(element.OwnerDocument.CreateElement("wsp", "Policy",
"http://schemas.xmlsoap.org/ws/2004/09/policy"));
                            }
                        }
                    }
                }
            }
        }

        /// <summary>
        /// Creates a behavior extension based on the current configuration
settings.
        /// </summary>
        /// <returns>
        /// The behavior extension.
        /// </returns>
        protected override object CreateBehavior() {
            return new WSSecurityPolicy12BugFixBehavior();
        }
    }
}
```

9.4. WS-Federation ziņojuma drošības izņēmuma apstrāde

Klases SampleWsFedRequestValidator.cs kods

```
// ----------------------------------------------------------------------------
// <copyright file="SampleWsFedRequestValidator.cs" company="ABC software">
//    Copyright © ABC SOFTWARE. All rights reserved.
//    The source code or its parts to use, reproduce, transfer, copy or
//    keep in an electronic form only from written agreement ABC SOFTWARE.
// </copyright>
// ----------------------------------------------------------------------------

namespace Abc.STS.Samples
{
    using System;
    using System.Web;
    using System.Web.Util;
    using Microsoft.IdentityModel.Protocols.WSFederation;

    /// <summary>
    /// This IvisRequestValidator validates the wresult parameter of the
    /// WS-Federation passive protocol by checking for a SignInResponse message
    /// in the form post. The SignInResponse message contents are verified later by
    /// the WSFederationPassiveAuthenticationModule or the WIF signin controls.
    /// </summary>
    public class SampleWsFedRequestValidator : RequestValidator
    {
        protected override bool IsValidRequestString(HttpContext context, string value,
RequestValidationSource requestValidationSource, string collectionKey, out int
validationFailureIndex) {
            validationFailureIndex = 0;

            if (requestValidationSource == RequestValidationSource.Form &&
collectionKey.Equals(WSFederationConstants.Parameters.Result, StringComparison.Ordinal)) {
                SignInResponseMessage message =
WSFederationMessage.CreateFromFormPost(context.Request) as SignInResponseMessage;

                if (message != null) {
                    return true;
                }
            }

            return base.IsValidRequestString(context, value, requestValidationSource,
collectionKey, out validationFailureIndex);
        }

    }
}
```

9.5. WS-Federation pieprasīt nepieciešamo informāciju

```
    public class Global : HttpApplication {
        private const int MaxQueryStringLength = 2048;
```

```
        void WSFederationAuthenticationModule_RedirectingToIdentityProvider(object sender,
RedirectingToIdentityProviderEventArgs e) {
            var requeredClaims = ReadRequiredClaimType();
            if (requeredClaims.Count > 0) {
                var rst = new RequestSecurityToken();
                rst.RequestType = "http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue";
                foreach (var item in requeredClaims) {
                    rst.Claims.Add(item);
                }

                WSFederationSerializer serializer = new WSFederationSerializer(new
WSTrust13RequestSerializer(), new WSTrust13ResponseSerializer());
                var request = serializer.GetRequestAsString(rst, new
WSTrustSerializationContext());

                e.SignInRequestMessage.Request = request;
            }

            // FIX: Send as POST if size great than 2k
            if (e.SignInRequestMessage.RequestUrl.Length >= MaxQueryStringLength) {
                e.Cancel = true;

                this.Response.Cache.SetCacheability(System.Web.HttpCacheability.NoCache);
                this.Response.Cache.SetNoStore();
                this.Response.Output.Write(e.SignInRequestMessage.WriteFormPost());
            }
        }

wre
            var serviceElement =
Microsoft.IdentityModel.Configuration.MicrosoftIdentityModelSection.DefaultServiceElement;
            var property = serviceElement.GetType().GetProperty("ApplicationService",
System.Reflection.BindingFlags.NonPublic | System.Reflection.BindingFlags.Instance);
            var applicationservice = property.GetValue(serviceElement, null) as
Microsoft.IdentityModel.Configuration.ConfigurationElementInterceptor;

            List<RequestClaim> collection = new List<RequestClaim>();
            XmlNodeList list = applicationservice.ChildNodes;
            if (list != null && list.Count > 0) {
                foreach (XmlNode node in list[0]) {
                    if (node.LocalName == "claimType") {
                        XmlAttribute namedItem =
(XmlAttribute)node.Attributes.GetNamedItem("type");
                        if (namedItem == null) {
                            throw new ConfigurationErrorsException("Required attribute
missing.", node);
                        }

                        var item = new RequestClaim(namedItem.Value);
                        collection.Add(item);

                        XmlAttribute attribute2 =
(XmlAttribute)node.Attributes.GetNamedItem("optional");
                        if (attribute2 != null) {
```

```
                        item.IsOptional = bool.Parse(attribute2.Value);
                    }
                }
            }
        }

        return collection;
    }
}
```