

|  |  |
| --- | --- |
|  |  |

Par Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana

3.daļa "VISS un Portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju"

IS servisu izstrāde

Standarts

VRAA-6\_15\_11\_58-VISS\_2010-ISs-STD

26.03.2014. versija 1.04

Rīgā 2014

|  |  |  |
| --- | --- | --- |
|  | **IEGULDĪJUMS TAVĀ NĀKOTNĒ** |  |

Dokumenta identifikācija

|  |  |
| --- | --- |
| Dokumenta ID: | VRAA-6\_15\_11\_58-VISS\_2010-ISs-STD-V1.04-26.03.2014. |
| Dokumenta nosaukums: | Par Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana.  3.daļa "VISS un Portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju".  IS servisu izstrāde.  Standarts. |
| Dokumenta kods: | VRAA-6\_15\_11\_58-VISS\_2010-ISs-STD |
| Versija: | Versija 1.04, Laidiens 26.03.2014. (saīsināti V1.04 26.03.2014.) |

Saskaņojumi

|  |  |  |  |
| --- | --- | --- | --- |
| Organizācija | Vārds, uzvārds, amats | Datums | Paraksts |
| Valsts reģionālās attīstības aģentūra | Atbildīgā persona no Pasūtītāja puses |  |  |
| SIA "ABC software" | J.Korņijenko, projekta vadītājs par tehniskiem jautājumiem no Izpildītāja puses | 26.03.2014. |  |
| SIA "ABC software" | M.Pētersons, projekta vadītājs par administratīviem jautājumiem no Izpildītāja puses | 26.03.2014. |  |
| SIA "ABC software" | E.Blumberga, projekta kvalitātes kontroles vadītāja | 26.03.2014. |  |

Izmaiņu vēsture

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versija | Datums | Apraksts | Organizācija | Autors |
| 0.04 | 14.02.2006. | Izveidota dokumenta sākotnējā versija. | SIA „ABC software” | J.Korņijenko |
| 0.09 | 06.03.2006. | Veikti labojumi saskaņā ar ELM.IVIS.CKS.URN\_IVIS\_100001\_DOC\_RCM\_ISS\_2006\_14\_02. | SIA „ABC software” | J.Korņijenko |
| 1.00 | 08.12.2006. | Papildināts IS servisa apraksts, IS servisu izstrādes process un IS servisu standarts. | SIA „ABC software” | J.Korņijenko |
| 1.01 | 30.05.2008. | Papildināts IS servisa apraksts, tai skaitā definēti biznesa (lietišķie) IS servisi, VISS Integrācijas IS servisi un VISS Pieprasījumu serviss. Papildināta sadaļa IS servisu ārējās saskarnes arhitektūras apraksts. | SIA „ABC software” | J.Korņijenko |
| 1.02 | 10.01.2012. | Iestrādātas redakcionālas izmaiņas. | SIA „ABC software” | J.Korņijenko |
| 1.03 | 11.03.2014 | Labota 2.2.2. sadaļa. | SIA „ABC software” | J.Korņijenko |
| 1.04 | 25.03.2014 | Pārstrādātā 2.2. sadaļa. Papildināta 5, sadaļa, veiktas citas redakcionālas izmaiņas | SIA „ABC software” | J.Korņijenko |

**Satura rādītājs**

[Attēlu saraksts 6](#_Toc383698525)

[1. Ievads 7](#_Toc383698526)

[1.1. Dokumenta nolūks 7](#_Toc383698527)

[1.2. Termini un pieņemtie apzīmējumi 7](#_Toc383698528)

[1.2.1. Termini 7](#_Toc383698529)

[1.2.2. Apzīmējumi 8](#_Toc383698530)

[1.3. Saistība ar citiem dokumentiem 9](#_Toc383698531)

[1.4. Dokumenta pārskats 10](#_Toc383698532)

[2. Kas ir IS serviss? 11](#_Toc383698533)

[2.1. Biznesa IS servisi 11](#_Toc383698534)

[2.2. VISS Integrācijas IS serviss 12](#_Toc383698535)

[2.2.1. VISS Integrācijas IS servisu lietošanas arhitektūra 12](#_Toc383698536)

[2.2.2. Integrācijas IS servisu izstrādes prasības 16](#_Toc383698537)

[2.2.3. Pieprasījumu auditācija un kontrole 17](#_Toc383698538)

[2.2.4. Integrācijas IS servisu notikumu apstrāde 17](#_Toc383698539)

[2.3. Klasifikatoru IS serviss 18](#_Toc383698540)

[2.4. VISS Pieprasījumu serviss 18](#_Toc383698541)

[3. IS servisu ārējas saskarnes arhitektūra un standartizācija 21](#_Toc383698542)

[4. IS servisu saskarnes izstrāde 23](#_Toc383698543)

[4.1. Projekta personāls 23](#_Toc383698544)

[4.2. Sākotnējā analīze 23](#_Toc383698545)

[4.3. Prasību specifikācija 24](#_Toc383698546)

[4.3.1. Iestādes apmaiņas infrastruktūra 24](#_Toc383698547)

[4.3.2. IS servisu arhitektūras plānošana 24](#_Toc383698548)

[4.3.2.1. Servisu abstrakcija 24](#_Toc383698549)

[4.3.2.2. Servisu autonomija 24](#_Toc383698550)

[4.3.2.3. Servisu komponējamība 25](#_Toc383698551)

[4.3.2.4. Servisu kontrakti 25](#_Toc383698552)

[4.3.2.5. Servisu sasaiste 25](#_Toc383698553)

[4.3.2.6. Servisu atkārtota izmantojamība 25](#_Toc383698554)

[4.3.2.7. Servisu stāvokļneatkarība 25](#_Toc383698555)

[4.3.2.8. Standartu lietošana 26](#_Toc383698556)

[4.3.2.9. Servisu detalizācijas pakāpe 26](#_Toc383698557)

[4.4. Analīze un dizains 27](#_Toc383698558)

[4.4.1. IS servisu standartizācijas pakāpes izvēle 27](#_Toc383698559)

[4.4.2. Servisa kvalitātes novērtējums 27](#_Toc383698560)

[4.4.3. Sinhronie vai asinhronie servisi 28](#_Toc383698561)

[4.4.4. Sistēmas noslogojuma novērtējums 28](#_Toc383698562)

[4.4.5. Auditāciju plānošana 28](#_Toc383698563)

[4.4.6. IS servisu projektējumu izstrāde 28](#_Toc383698564)

[4.4.7. Testu plānošana 31](#_Toc383698565)

[4.5. Izstrāde 32](#_Toc383698566)

[4.6. Testi 32](#_Toc383698567)

[4.6.1. Testa klientu izmantošana 32](#_Toc383698568)

[4.6.2. Testēšanas servisi 32](#_Toc383698569)

[4.7. Piegāde 32](#_Toc383698570)

[4.7.1. Piegādes nodevumi 32](#_Toc383698571)

[4.7.2. IS servisa reģistrācija VISS IS servisu katalogā 32](#_Toc383698572)

[4.8. Ekspluatācija 33](#_Toc383698573)

[4.8.1. Vadība 33](#_Toc383698574)

[4.8.2. Versijas 33](#_Toc383698575)

[5. IS servisu apraksts 34](#_Toc383698576)

# Attēlu saraksts

[1.attēls. Sinhrons Integrācijas IS serviss WeatherForecast pārlūkprogrammā 12](#_Toc383698577)

[2.attēls. IS servisu implementācijas scenārijs 13](#_Toc383698578)

[3.attēls. Integrācijas IS serviss atrodas iestādes pusē 14](#_Toc383698579)

[4.attēls. Integrācijas IS serviss atrodas VISS pusē, bet biznesa serviss iestādes pusē. 14](#_Toc383698580)

[5.attēls. Asinhrons -> Sinhrons scenārijs 15](#_Toc383698581)

[6.attēls. Sinhrons -> Sinhrons scenārijs 15](#_Toc383698582)

[7.attēls. Asinhrons -> Asinhrons scenārijs 15](#_Toc383698583)

[8.attēls. Sinhrons -> Asinhrons scenārijs 16](#_Toc383698584)

[9.attēls. Integrācijas IS servisa aprakstošā sekcija 17](#_Toc383698585)

[10.attēls. Pieprasījuma servisa detalizēts plānojums 18](#_Toc383698586)

[11.attēls. IS servisu pakāpeniskā standartizācija un darbu sadalījums starp iestādes IS un VISS 21](#_Toc383698587)

[12.attēls. IS servisu pakāpeniskā standartizācija un darbu sadalījums starp iestādes IS un VISS 22](#_Toc383698588)

# Ievads

Ja iepriekšējo desmitgadi nosacīti var uzskatīt par laiku, kad tika veidoti primārie datu reģistri, tad jaunā desmitgade acīmredzami vairāk būs orientēta uz šo reģistru integrāciju un piedāvāto pakalpojumu elektronizāciju un pieejamību iedzīvotājiem. Šo ceļu šobrīd iet visas attīstītās pasaules valstis. Vienota pieeja un atbilstošas tehnoloģijas izvēle integrācijas realizācijā ir nozīmīgākie faktori, kas nosaka valsts pārvaldes elektronizācijas veiksmīgumu. No otras puses - katra valsts iestādes reģistra vai informācijas sistēmas (IS) primārais mērķis ir nodrošināt konkrētas, specifiskas funkcijas, kas jāveic attiecīgajai iestādei. IS un datu bāzes (DB), kuras tiek izmantotas iestādē, ir izvēlētas, izstrādātas un optimizētas tieši šīs konkrētās iestādes vajadzībām, un tās var nebūt orientētas uz iespēju kādam citam saņemt informāciju. Bet, ja tāda iespēja ir paredzēta, tad tā var būt ļoti specifiska un ar dažādiem ierobežojumiem. Pēc situācijas izvērtēšanas (skat. dokumentu [8]) tika pieņemts lēmums turpināt attīstīt valsts reģistru integrāciju uz kopējiem standartiem, kas balstītos uz XML un servisorientētas arhitektūras (SOA) principiem. XML shēmu izstrādes vadlīnijas un loma standartizācija ir aprakstīti XML shēmu izstrādes vadlīniju dokumentā [1]. Informācijas sistēmu integrācijas servisu izstrāde, izmantojot SOA arhitektūras principus, ir šī dokumenta mērķis.

## Dokumenta nolūks

Šī dokumenta mērķis ir mēģināt atklāt problēmas, kuras varētu rasties iestādēs IS servisu izveidē, izmantojot Valsts informācijas sistēmu savietotāju (VISS) kā integrācijas platformu. Aprakstīt IS servisu arhitektūru VISS projekta ietvaros, sniegt IS servisu izstrādes rekomendācijas un standartu definīcijas, aprakstīt IS servisu reģistrācijas procedūras VISS IS servisu katalogā un IS servisu speciālās saskarnes.

Dokumenta auditoriju veido:

* VRAA darbinieki, kuri ir atbildīgi par VISS nodevumu un e-pakalpojumu pieņemšanu un izvērtēšanu;
* Valsts iestādes reģistru IS servisu izstrādātāji;
* Valsts iestādes reģistru e-pakalpojumu izstrādātāji;
* VISS administratori un uzturētāji.

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

## [Termini](#_Toc65487056) un pieņemtie apzīmējumi

### Termini

Dokumentā izmantotie termini ir apkopoti 1.tabulā.

1.tabula

Termini

| Termins | Apraksts |
| --- | --- |
| Informācijas sistēma | Datu bāzes, lietotāja programmatūras, procedūru un funkciju kopums un to savstarpējās attiecības, kas nodrošina sistēmas funkcionēšanu atbilstoši tās pamatmērķiem un pamatuzdevumiem. |
| Integrācijas IS serviss | Standartizēts informācijas sistēmas pieejas veids (serviss), kas pie informācijas resursa nodrošina ārējo saskarni. Integrācijas IS serviss parasti sastāv no divām daļām: servera daļas un klienta daļas. Servera daļa – IS servisa realizācija iestādes pusē (realizācijas veida izvēle ir iestādes kompetencē, saskarnes aprakstīšanas vēlamais standarts ir WSDL); klienta daļa – IS servisa saskarnes realizācija VISS IS servisu kataloga pusē (nodrošina nepieciešamo auditāciju, kontroli). |
| Uz servisiem orientēta arhitektūra | Uz servisiem orientēta arhitektūra sastāv no virknes brīvi savienotu, sadalītu, lietišķu komponenšu, kuras kopā var ietilpt vienā ar uzņēmumu saistītajā procesā. |
| Vārdtelpa (namespace) | XML shēmas vārdtelpa nosaka noslēgtu nosaukumu kopu vai vietu, kur shēma (nosaukumu kopa) tiek glabāta. Vārdtelpas definē ar URI (piemēram, URL) un, tās paredzētas ar nosaukumu piešķiršanu saistītu problēmu risināšanai. Vienas vārdtelpas iekšienē visiem nosaukumiem jābūt unikāliem, lai gan citā vārdtelpā attiecīgajam nosaukumam var būt cita nozīmē. |
| Web Serviss vai IS serviss | Web Servisi ir apzīmējums tādām lietojumprogrammām, kuras ir paredzētas izsaukumiem no citu lietojumprogrammu puses un kuras saņem datus, un sniedz atbildes uz tiem standartizētā formātā (XML), parasti izmantojot SOAP protokolu. Web Servisi ir uz servisiem orientētas arhitektūras (SOA) sastāvdaļu konkrēta realizācija. |

### Apzīmējumi

Dokumentā izmantotie apzīmējumi ir apkopoti 2.tabulā.

2.tabula

Apzīmējumi

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

## Saistība ar citiem dokumentiem

Dokuments ir izstrādāts, balstoties uz šādiem dokumentiem:

1. Par Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana. 3.daļa "VISS un portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju". XML resursu izstrāde. Vadlīnijas (VRAA-6\_15\_11\_58-VISS\_2010-XML-VDL-V1.03-21.02.2012.).
2. Standarts LVS 72:1996, Ieteicamā prakse programmatūras projektējuma aprakstīšanai.
3. Fizisko personu datu aizsardzības likums (http://www.dvi.gov.lv/likumdosana/fpda/).
4. Par Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana. 3.daļa "VISS un portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju". Arhitektūras risinājuma vīzija (VRAA-6\_15\_11\_58-VISS\_2010-ARH-VIZ-V1.02-21.03.2012).
5. Par Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana. 3.daļa "VISS un portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju". Metadatu un e-pakalpojumu identifikācija. Standarts (VRAA-6\_15\_11\_58-VISS\_2010-MET\_EPAK\_ID-STD-V1.03-21.02.2012).

1. [http://www.oasis-open.org/specs/index.php#wssv1.](http://www.oasis-open.org/specs/index.php#wssv1.1)0.
2. <http://www.oasis-open.org/specs/index.php#wssv1.1>.
3. Par Valsts informācijas sistēmu savietotāja, Latvijas valsts portāla www.latvija.lv un elektronisko pakalpojumu izstrāde un uzturēšana. 3.daļa "VISS un Portāla jaunu un esošo moduļu papildinājumu izstrāde, ieviešana, garantijas apkalpošana un uzturēšana saskaņā ar tehnisko specifikāciju". E-pakalpojumu arhitektūras izstrāde. Vadlīnijas. VRAA-6\_15\_11\_58-VISS\_2010-EPAK\_ARH-VDL

## Dokumenta pārskats

Dokuments sastāv no 5 nodalījumiem:

1. nodalījumā – „Ievads” – aprakstīta dokumenta kopējā struktūra, nolūks, izmantotie termini un apzīmējumi, kā arī saistība ar citiem dokumentiem.
2. nodalījumā ir aprakstīta IS servisu arhitektūra.
3. nodalījumā – „IS servisu izstrādes priekšnosacījumi” – ir dotas IS servisu izstrādes rekomendācijas, kas savukārt sadalītas atbilstoši IS servisu izstrādes fāzēm.
4. nodalījumā – „IS servisu izstrāde” – ir sniegts IS servisu tehniskās realizācijas apraksts.
5. nodalījumā – „IS servisu standarts” – ir apkopotas IS servisu standartu definīcijas.

# Kas ir IS serviss?

Ar IS servisiem saprotam visus Web servisus, kas aprakstam lieto WSDL, kā primāro sniegšanas protokolu – SOAP un kā primāro sniegšanas kanālu - HTTP(s), un, kas nodrošina ārējo saskarni pie informācijas sistēmu reģistriem. Tiek izdalīti šādi IS servisu veidi:

* Biznesa (lietišķie) IS servisi – atkal izmantojamas komponentes e-pakalpojumu un integrācijas procesu izstrādei (skat. 2.1.nodaļu).
* Integrācijas IS servisi – integrācijas slānis, kas savieno vienu vai vairāku biznesa IS servisu izsaukumus sinhronu un asinhronu pieprasījumu veidā (skat. 2.1.nodaļu).
* Klasifikatoru IS servisi – e-pakalpojumu darbināšanai nepieciešamie klasifikatori parasti tiek sniegti bez drošības un/vai izmantojot REST protokolu (skat. 2.2.nodaļu).
* VISS Pieprasījumu serviss – centrālais ieejas punkts (*gateway*) VISS infrastruktūrā (skat. 2.3.nodaļu).

Vairāk par katru IS servisa veidu ir rakstīts tālāk šajā dokumentā.

## Biznesa IS servisi

Biznesa IS servisi veic darbības tikai ar biznesa transakcijām, kas ir saistītas ar konkrēto dzīves situāciju vai pārbaudi un ir plaši lietojami e-pakalpojumos vai iestādes darbības nodrošināšanā. Biznesa IS servisus ir iespējams izsaukt tikai VISS infrastruktūras ietvaros un parasti tie ir izmitināti un tos uztur atbildīgā iestāde vai valsts reģistrs. Parasti šādi IS servisi veido iestādes datu apmaiņas ārējo saskarni un papildus biznesa funkcijām varētu veikt datu prasītāja identifikāciju un pieejas tiesību pārbaudi. Biznesa IS servisi tiek izstrādāti iestādē un iestādes vajadzībām, bet, lai tie būtu saskaņoti un savietojami ar VISS, jābūt izpildītiem šādiem nosacījumiem:

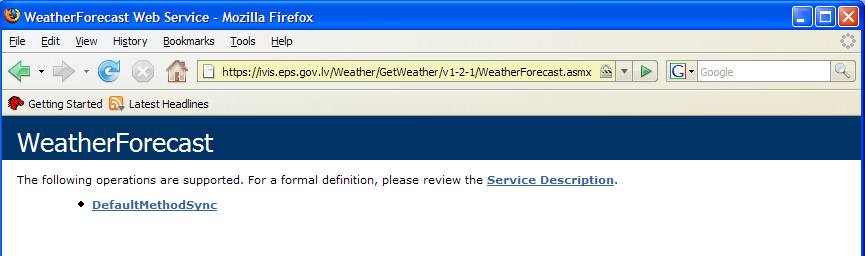
1. IS servisam jāatbilst WS-I rekomendācijai Basic Profile 1.1 (<http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html>).
2. Datu nodošanas un saņemšanas formātam jābūt XML, kas izveidots saskaņā ar VISS XML shēmas katalogā reģistrētu XML shēmu (XML shēmas tipu). XML shēmai jābūt izstrādātai un reģistrētai saskaņā ar XML shēmu izstrādes vadlīniju dokumentu [1].
3. Ir iespējami vairāki arhitektūras risinājumi (skat. 3.nodalījumu) atkarībā no iestādes vajadzībām un drošības politikas.

Biznesa IS servisu piemēri:

1. IS serviss, kas nodrošina dokumenta derīguma pārbaudi Iekšlietu ministrijas Informācijas Centra Nederīgo dokumentu reģistrā (NDR) – saņem XML struktūru ar elementiem dokumenta izdevējvalsts, dokumenta veids, sērija un numurs (URN:IVIS:100001:XSD-IISExchange-NpDR-v1-0-TYPE-NpDRDocumentBasicInfo) un atgriež pārbaudes rezultātu saskaņā ar XML shēmas tipu URN:IVIS:100001:XSD-IISExchange-NpDR-v1-0-TYPE-NpDRDocumentSearchInfo (XML shēmas adrese: <http://ivis.eps.gov.lv/XMLSchemas/100008/IISExchange/v1-0/NpDR.xsd>)
2. VISS notifikāciju serviss – saņem notifikācijas datus un atgriež paziņojuma sūtīšanas rezultātu (URN:IVIS:100001:ISS-EPS-NotificationServiss-v1-5).

## VISS Integrācijas IS serviss

Integrācijas IS servisu kopums tiek veidots kā VISS platforma, uz kuras iespējams izstrādāt e-pakalpojumus vai risināt citus integrācijas uzdevumus. IS servisu izstrādātājiem tiek piedāvāti atkārtotai izmantošanai jau iepriekš izstrādātie IS servisi un integrācijas rīki, kas atvieglo kompozītu IS servisu izstrādi. Tipiska Integrācijas IS servisa WSDL apskates lapa ir parādīta 1.attēlā.



1.attēls. Sinhrons Integrācijas IS serviss WeatherForecast pārlūkprogrammā

Eksistē divi Integrācijas IS servisu tipi:

* Sinhronais (piemēram, https://lvp.viss.gov.lv/CalculationService/CalculationSync.svc);
* Asinhronais (piemēram, https://lvp.viss.gov.lv/CalculationService/CalculationAsync.svc).

### VISS Integrācijas IS servisu lietošanas arhitektūra

VISS Integrācijas IS serviss nodrošina standartizētu ārējo saskarni piekļuvei informācijas resursam un parasti sastāv no divām daļām: servera daļas un klienta daļas. Pēc IS servisa definīcijas servera daļa ir iestādes informācijas sistēmas ārēja saskarne – IS servisa realizācija iestādes pusē. Ārējas saskarnes realizācijas veida izvēle ir iestādes kompetencē, saskarnes aprakstīšanas vēlamais standarts ir WSDL un parasti tie ir biznesa servisi, kas veido iestādes ārējo saskarni (skat. 2.1.nodaļu). Savukārt integrācijas slāni starp VISS Pieprasījumu servisu un IS servisa servera daļu nodrošina IS servisa saskarnes realizācija VISS pusē, kas ir strikti standartizēta un nodrošina nepieciešamo auditāciju un kontroli. Pēc būtības, VISS Integrācijas IS serviss ir neatkarīgs Web lietojums un varētu būt izmitināts gan VISS infrastruktūras ietvaros, gan iestādes datu apmaiņās infrastruktūrā.

Katrai valsts iestādei vai komercuzņēmumam, kas vēlētos piedalīties e-pakalpojumu izstrādē vai VISS pilnveidošanā, tiek piedāvāts izveidot IS servisus, kas nodrošinātu pieslēgšanās saskarni informācijas resursa datiem un ir „melnā kaste”, kurai ir specifisks funkciju klāsts, kā arī - ieejas un izejas parametri (skat. *Web servisu* kopējas izstrādes un arhitektūras prasības 4.3.1.sadaļā).

Realizācijas scenārijs IS servisiem, kas izmanto ārējus sistēmas datus, ir parādīts 2.attēlā.



2.attēls. IS servisu implementācijas scenārijs

Kā redzams 2. attēlā, ir iespējami četri arhitektūras risinājumi:

1. Informācijas sistēma 1 nodrošina Integrācijas IS servisu slāni savā infrastruktūrā – Integrācijas IS servisus A1, A2 un A3. Parasti starp VISS un iestādes infrastruktūru parasti tiek pacelts VPN savienojums, kas garantē datu aizsardzību transporta līmeni. Cita lietotāju identifikāciju un drošības pārbaude nav nepieciešama – viss nepieciešamais tiek nodrošināts VISS infrastruktūrā.
2. Integrācijas IS servisu B1, B2 un B3 servera daļu veido attiecīgi biznesa servisi X1, X2 un X3, kas nodrošina iestādes biznesa datu sniegšanu (skat. 2.1.nodaļu). Parasti šiem servisiem nav nepieciešams realizēt papildus drošības slāni – viss nepieciešamais tiek nodrošināts VISS infrastruktūrā. Šis variants ir piemērotāks lietošanai, jo biznesa servisi X1, X2 un X3 varētu tikt izmantoti citu Integrācijas servisu konstruēšanai atkarība no izstrādātajā vajadzībās.
3. Integrācijas IS servisi C1, C2 un C3 lieto VISS pusē izmitinātos, biznesa servisus, kas savukārt lieto citu formātu un protokolu datu apmaiņai. Šāda pieeja ir nepieciešama, ja paredzēts, ka servisu dati tiek lietoti vairakkārt, bet iestāde, datu - sniedzējs, nevar standartizēt savu saskarni saskaņā ar iepriekšminēto arhitektūru.
4. Integrācijas IS servisu D1, D2 un D3 servera daļā lieto citu formātu un protokolu datu apmaiņai, bet sniegšanas kanāla standartizācija notiek jau VISS pusē. Šāda pieeja ir iespējama, ja iestādes dati ir specifiski un varētu interesēt tikai konkrēta pakalpojuma vai integrācijas projekta ietvaros.

1. un 2. arhitektūras integrācijas risinājumi ir primāri un nodrošina vislabāko iestādes sadarbspēju VISS ietvaros. Pie tam, šie scenāriji ir krietni vienkāršāki realizācijā un uzturēšanā, neprasa nopietnas drošības infrastruktūras izveidi iestādes pusē.

Izmantojot latvija.lv par piemēru, kopumā 1. un 2. scenāriji loģiski dalās trijos slāņos:

* Latvija.lv e-pakalpojums – servisa izmantotājs;
* VISS Pieprasījumu serviss – drošības un personas datu audits;
* Iestāde – biznesa datu sniegšana.

|  |  |
| --- | --- |
|  | .attēls. Integrācijas IS serviss atrodas iestādes pusē |
|  | .attēls. Integrācijas IS serviss atrodas VISS pusē, bet biznesa serviss iestādes pusē. |

Integrācijas serviss nodrošinās IS servisu klienta daļu, bet informācijas resursu turētāji nodrošinās sinhronus vai asinhronus servisu pieprasījumus. Sinhronas komunikācijas gadījumā, servisa lietotājam, lai saņemtu atbildes ziņojumu, ir jāgaida, kamēr izsauktā servisa operācija tiek izpildīta. Asinhronas komunikācijas gadījumā, servisa lietotājam nav jāgaida (lai arī tas sinhronā režīmā var saņemt tūlītēju atbildes ziņojumu, ka servisa operācija veiksmīgi iedarbināta), kamēr izsauktā servisa operācija tiek izpildīta, un tas var turpināt veikt citas darbības („nosūta un aizmirst” princips). Šajā gadījumā, ja servisa lietotājam ir nepieciešama atbilde pēc operācijas pabeigšanas, tiek izmantota t.s. atzvanīšana (*callback*), t.i., servisa lietotāja izsauktais serviss pēc operācijas izpildes veic izsaukumu uz servisa lietotāja norādīto vietni.

VISS integrācijas IS servisi nodrošina piekļuvi jebkuram informācijas resursam gan sinhroni, gan asinhroni. Sevišķi svarīgi ir piedāvāt asinhronu pieeju servisiem, kuriem *back-end* sistēma spēj nodrošināt tikai sinhrono piekļuvi. Tomēr, ja nav iespējams garantēt iestādes datu - sniedzēja IS pieejamību, tad vēlamais protokols e-pakalpojumu un IS servisu implementācijā arī ir asinhrons.

Iespējamie integrācijas scenāriji VISS infrastruktūrā ir parādīti 5.-8.attēlos, kur:

* A – Pieprasījumu serviss vai Orķestrēšanas serviss;
* B – IS servisa klienta daļa (Integrācijas serviss);
* C – IS servisa servera daļa (*back-end* sistēma).

Tā kā „back-end” sistēma var būt sinhrona vai asinhrona un VISS Integrācijas servisi arī var būt sinhroni vai asinhroni, veidojas četri dažādi scenāriji (skat. 5.-8.attēlu).

1. Asinhrons -> Sinhrons (*simulated async*).



5.attēls. Asinhrons -> Sinhrons scenārijs

1. Sinhrons -> Sinhrons.



6.attēls. Sinhrons -> Sinhrons scenārijs

1. Asinhrons -> Asinhrons.



7.attēls. Asinhrons -> Asinhrons scenārijs

1. Sinhrons -> Asinhrons.



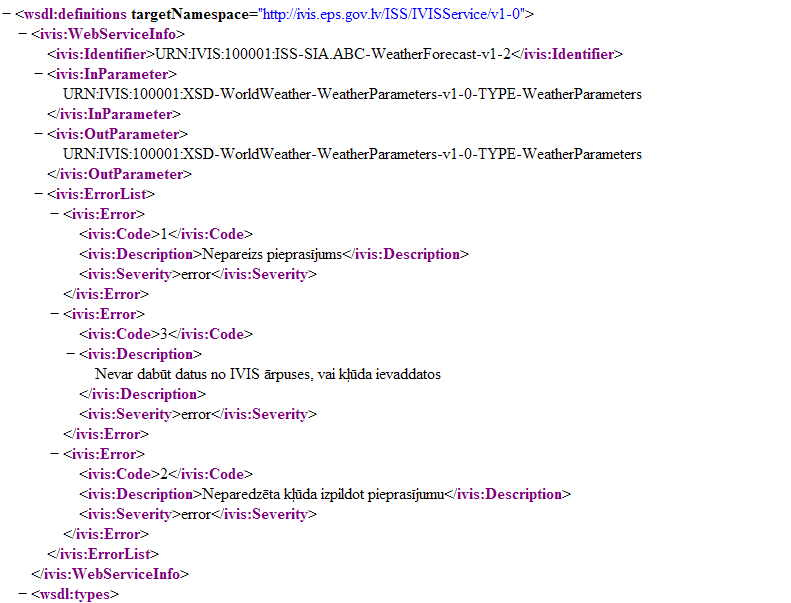
8.attēls. Sinhrons -> Asinhrons scenārijs

Atkarībā no biznesa scenārija un prasībām, implementējot IS servisus un e-pakalpojumus, var rasties nepieciešamība pēc jebkuras no šīm kombinācijām. Priekšrocība tomēr dodama asinhronajiem scenārijiem. Scenāriju „Sinhrons -> Asinhrons” iespējams izmantot tikai un vienīgi gadījumā, ja *back-end* sistēma piedāvā vienīgi asinhronu izsaukumu, vai biznesa scenārijs prasa izmantot sinhrono variantu.

### Integrācijas IS servisu izstrādes prasības

VISS nodrošina iespēju darboties ar sinhroniem un asinhroniem integrācijas IS servisiem. Neatkarīgi no tehnoloģijas vai realizācijas visiem Integrācijas IS servisiem ir izvirzītas šādas prasības:

1. IS servisam jāatbilst WS-I rekomendācijai Basic Profile 1.1 (<http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html>).
2. IS servisa vārdtelpai (namespace) jābūt http://ivis.eps.gov.lv/ISS/IVISService/v1-0
3. Datu nodošanas un saņemšanas formāts ir XML. XML ziņojumam, kas satur IS servisa biznesa datus, jābūt noformētam atbilstoši iepriekš definētai XML shēmai (XML shēmas tipam). XML shēmai jābūt reģistrētai VISS XML shēmu katalogā saskaņā ar XML shēmu izstrādes vadlīniju dokumentu [1].
4. WSDL deklarācijai jāsatur speciāla XML sekcija (skat. 9.attēlu), kas ietver IS servisa identifikatoru, atgriežamo kļūdu un citu notikumu kopu, ienākošo un izejošo XML shēmas datu tipu identifikatorus.



9.attēls. Integrācijas IS servisa aprakstošā sekcija

1. Sinhrons Integrācijas IS serviss satur obligāto metodi (SOAPAction) DefaultMethodSync, kas saņem tās darbināšanai nepieciešamos datus aploksnē IVISRequest (<http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0/IVISRequest.xsd>) un datus, kā arī citu saistīto informāciju par izpildi atgriež aploksnē IVISResponse (<http://ivis.eps.gov.lv/XMLSchemas/100001/IVIS/v1-0/IVISResponse.xsd>). Detalizēti minētie XML shēmas tipi un to lietošana ir aprakstīta dokumentos [5] un [8].
2. Asinhrons Integrācijas IS serviss satur obligāto metodi (SOAPAction) DefaultMethod, kas saņem tās darbināšanai nepieciešamos datus aploksnē IVISRequest un datus atgriež IVISResponse struktūrā, saglabājot tos VISS Pieprasījumu servisā Asinhrono datu krātuvē [4].

Konkrēta IS servisa realizācija ir atkarīga no platformas. Tai skaitā, ir izstrādātas konkrētas rekomendācijas Windows platformai [5],[8].

### Pieprasījumu auditācija un kontrole

Pieprasījumu auditācija un kontrole notiek VISS pieprasījumu servisā automātiski, pamatojoties uz informāciju, kas ir nodrošināta IS servisa izsaukumā.

### Integrācijas IS servisu notikumu apstrāde

Integrācijas IS servisiem jāatgriež kļūdas paziņojumi lietojot speciālo sekciju IVISResponse.ErrorList. Visiem iespējamiem notikumiem jābūt deklarētiem WSDL WebServiceInfo sekcijā. Tiek definēti 4 notikumu veidi: 0 – kļūda, 1 – brīdinājums, 2 – informācija, 3 – servisa pārbaudes (*debug*) informācija.

Kļūdu kategorija (0 – kļūda) tiek lietota tikai un vienīgi, ja IS servisā notika problēma, kuru nav iespējams atrisināt un kuras dēļ nav iespējams turpināt IS servisa izpildi.

## Klasifikatoru IS serviss

Klasifikatoru IS servisi parasti nodrošina e-pakalpojumiem un citiem integrācijas uzdevumiem pieeju pie nepieciešamajiem klasifikatoriem. Visi kopējie klasifikatori (ATVK, iestāžu klasifikators) tiek izmitināti un uzturēti VISS, bet klasifikatori, kas ir specifiski konkrētai iestādei, parasti tiek izmitināti un uzturēti iestādē. Pieeja klasifikatoru IS servisiem parasti nav ierobežota.

Piemēram, šādi klasifikatoru servisi:

1. Iekšlietu ministrijas Informācijas centra klasifikatoru izgūšanas serviss, kurā pēc klasifikatora identifikatora var saņemt pilnu klasifikatoru XML veidā (URN:IVIS:100001:ISS-IEM.IC-IeMClassificator-v1-1).
2. Centrālas Statistikas Biroja ATVK klasifikators, kurā pēc reģiona koda var saņemt reģiona nosaukumu vai citu reģionu sarakstu, kas ir ietverts ievadītajā reģionā.

VISS infrastruktūrā tiek nodrošināts kopējais klasifikatoru katalogs, visi klasifikatori ir pieejami publiski, izmantojot REST publicēto saskarni.

## VISS Pieprasījumu serviss

VISS Pieprasījumu serviss ir svarīga infrastruktūras sastāvdaļa (URN:IVIS:100001:ISS-EPS-RequestWebService-v1-9). Pieprasījumu IS serviss nodrošina pieeju Integrācijas IS servisiem, izpildot pieprasījumu auditācijas un kontroles uzdevumus.

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

Sīkāk par Pieprasījumu IS servisu skatīt Arhitektūras risinājuma vīzijā [4] un Metadatu un e-pakalpojumu identifikācijas standarta dokumentā [5] un . E-pakalpojumu arhitektūras izstrādes vadlīniju dokumentā [8].

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



10.attēls. Pieprasījuma servisa detalizēts plānojums

Pieprasījumu serviss esošā realizācija atbilst gan Web Services Security v1.0[[1]](#footnote-1) [6], gan Web Services Security v1.1[[2]](#footnote-2) [7] standartiem. Pieprasījumu servisa saskarni veido metodes (Microsoft .NET C# sintaksē), kas ir aprakstītas 3.tabulā.

3.tabula

Pieprasījuma servisa metodes

| Pieprasījuma servisa metode | Apraksts |
| --- | --- |
| IVISResponse SubmitSync (IVISRequest request) | Ar metodi SubmitSync tiek nodrošināts sinhrons izsaukums. Sinhrona izsaukuma gadījumā par parametru nepieciešams izmantot IVISRequest struktūru (skat. 2.2.2.sadaļā). Metode atgriezīs arī IVISResponse struktūru, kas attiecīgā IS servisa darbības rezultātā parasti tiek papildināta ar kādiem biznesa datiem <Body> elementā. |
| IVISResponse Submit (IVISRequest request) | Asinhrons izsaukums tiks uzsākts, izsaucot metodi Submit. Kā metodes parametru nepieciešams izmantot IVISRequest struktūru (skat. 2.2.2.sadaļā). Veiksmīga izsaukuma gadījumā, izsaucējam tiks atgriezta IVISResponse struktūra ar aizpildītu CorrelationID – konkrētā pieprasījuma identifikatoru. Šis elements tiks lietots, lai sasaistītu oriģinālo pieprasījumu ar „back-end” sistēmas sniegto atbildi. Tas var tikt lietots kā parametrs Pieprasījumu servisa metodēm Poll, GetStatus un Dispose. Asinhronajā scenārijā <Body> elements netiks atgriezts. |
| int GetStatus (CorrelationID requestID) | GetStatus metode atgriezīs sagatavotās atbildes statusu. Tā ir jāizsauc pēc Submit metodes izsaukuma, lai uzzinātu, vai atbilde jau ir sagatavota. Kā parametrs tai jānodod CorrelationID.  Pieprasījuma servisa GetStatus metodes implementācijai tiks izmantota Asinhrono pieprasījumu rezultātu krātuve (speciāla datu bāze asinhrono rezultātu glabāšanai). |
| IVISResponse Poll (CorrelationID requestID) | Poll metode atgriezīs sagatavoto atbildes ziņojumu. To nepieciešams izsaukt pēc Submit metodes izsaukuma, lai iegūtu atbildes ziņojumu. Vispirms ar GetStatus metodi ir jānoskaidro, vai atbilde jau ir sagatavota. Ja atbilde jau ir sagatavota, tad to var iegūt ar Poll metodi. Poll metodi var izsaukt vairākas reizes. Kamēr atbilde ar Dispose metodi nebūs dzēsta, to vēl joprojām varēs saņemt. Kā parametrs tai jānodod CorrelationID. Poll metode atgriež IVISResponse struktūru. |
| int Dispose (CorrelationID requestID) | Dispose metode tiek lietota asinhrona scenārija gadījumā. Šī metode obligāti jāizsauc, lai Pieprasījumu servisam paziņotu, ka atbilde ir veiksmīgi saņemta un turpmāk to var neuzglabāt, pretējā gadījumā - atbilde tiks automātiski izdzēsta pēc noteikta laika perioda. Pieprasījuma servisa Dispose metode izmantos Asinhrono pieprasījumu rezultātu datu bāzi atbildes meklēšanai. |
| URNType StartTransaction(URNType eServiceID) | Metode StartTransaction tiek izsaukta, kad nepieciešams uzsākt konkrētu e-pakalpojuma instanci. Parametrs eServiceID ir e-pakalpojuma identifikators. Metode atgriež e-pakalpojuma instances (transakcijas) URN identifikatoru. |
| void EndTransaction(URNType transactionID) | Metode EndTransaction tiek izsaukta, kad nepieciešams pabeigt konkrētu e-pakalpojuma instanci. Parametrs TransactionID ir e-pakalpojuma instances (transakcijas) identifikators. |

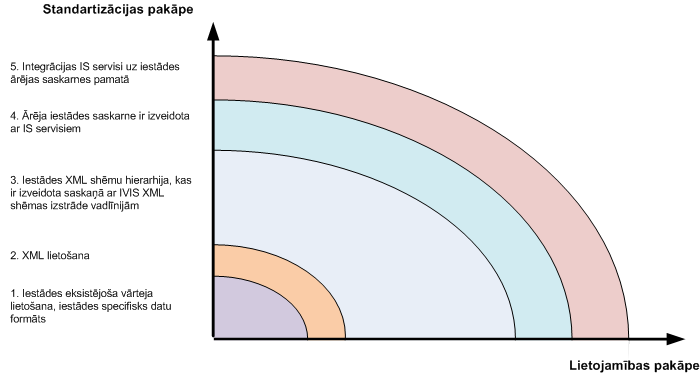
Pieprasījumu un vaicājumu protokolā tiek izmantoti divi ziņojumu veidi – IVISRequest un IVISResponse. Kļūdu gadījumā izsaucējs saņems <Errors> struktūru atbilstoši 2.2.4.sadaļā Pieprasījumu serviss nodrošinās šādas kopīgas funkcijas:

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

# IS servisu ārējas saskarnes arhitektūra un standartizācija

Iestādes standartizācijas procesu vienmēr ir iespējams sadalīt posmos (skat. 11.attēlu) tādā veidā, lai darbu varētu sākt zemākā standartizācijas līmenī un tas tiktu turpināts pēc iespējas pakāpeniski. Izmantojot 11.attēlā parādīto diagrammu, aprakstīsim IS servisu pakāpeniskās standartizācijas scenāriju sīkāk:

1. Pirmajā IS servisu izstrādes posmā tiek izmantota kāda eksistējoša vārteja (*gateway*) uz iestādes IS. Piemēram, asinhronai apmaiņai tas varētu būt FTP, sinhronai - HTTP. Iestādes IS realizē datu eksportu uz citām sistēmām, izmantojot kādu speciālu formātu (piemēram, .csv),, bet veidot eksportu kā XML ar WS palīdzību nav iespējams resursu trūkumu dēļ.
2. Bieži vien esošās sistēmas datus ir relatīvi vienkārši pārveidot XML formātā, lai izveidotu XML saskarni, kuru varētu lietot citā sakarā un kas nodrošinātu atkārtotas izmantošanas iespēju. Otrajā IS servisu pilnveidošanas fāzē iestādes ietvaros tiks pieņemts lēmums par datu eksportēšanu klientiem tikai XML formātā. Rezultātā –.csv tiek aizstāts ar XML.
3. Ar laiku iestāde nolems visiem saviem klientiem eksportēt datus, izmantojot standartizēto XML hierarhiju, kas apraksta visas iestādei nepieciešamās datu struktūras, tas būtiski vienkāršo datu eksporta procedūru. Trešajā IS servisu pilnveidošanas fāzē tiek pielietots XML izveidotājs atbilstoši VISS katalogā reģistrētajai XML hierarhijai [1]. Šī fāze ir svarīgākā un tiek sasniegts servisa vislielākais standartizācijas un lietojamības pieaugums, jo sintaksiski un semantiski servisa pieprasījumi un atbildes ir saskaņoti ar vienoto standartu.
4. Ceturtajā fāzē HTTP protokols tiks aizstāts ar Web Servisu saskarni.
5. Piektajā fāzē tiks izpildītas citas VISS prasības, kas nodrošina Integrācijas IS servisu (skat. 2.1.nodaļu) slāņu izstrādi e-pakalpojumu un integrācijas procesiem.



11.attēls. IS servisu pakāpeniskā standartizācija un darbu sadalījums starp iestādes IS un VISS

IS servisu standartizācijas, piemēram, ir ilustratīva nozīme, un tas apraksta tikai vienu no iespējamiem scenārijiem iestādes integrācijai VISS. Katrā konkrētā situācijā sākotnējās analīzes fāzē ir jānosaka vēlamākā IS servisu integrācijas pakāpe (skat. 4.4.1.nodaļu).

Veidojot iestādes ārējo apmaiņas infrastruktūru svarīgi lietot iepriekš saskaņotas un publicētas XML shēmas (hierarhijas) saskaņā ar [1]. Tiklīdz tiek lietoti vienādi tipi (skat. 12.attēlu), ir garantēta datu kvalitāte un precizitāte, un tiks izslēgti gadījumi, kad pēc dažām konvertācijām un pārveidojumiem „kravu transportlīdzeklis” pēkšņi kļūs par „lidmašīnu” utt. Nav tik svarīgs transporta protokola līmenis (SOAP, HTTP, FTP utt.), bet daudz svarīgāk ir nodrošināt formātu sintaksisko un semantisko saskaņošanu.



12.attēls. IS servisu pakāpeniskā standartizācija un darbu sadalījums starp iestādes IS un VISS

# IS servisu saskarnes izstrāde

Šajā nodalījumā ir aplūkotas problēmas, kuras varētu rasties e-pakalpojumu izstrādē vai integrācijas projekta realizācijas gaitā un kuras ir attiecināmas uz IS servisu izstrādi, darbināšanu un uzturēšanu. IS servisu izstrādes process ir iedalīts šādās loģiskās fāzēs:

1. Projekta personāls
2. Sākotnējā analīze.
3. Prasību specifikācija.
4. Analīze.
5. Projektēšana.
6. Testēšana.
7. Piegāde.
8. Ekspluatācija.

Tālāk dokumentā ir aprakstīts IS servisu izstrādes process un tiek sniegti vispārēji tā izstrādes ieteikumi. Pēc nepieciešamības, katra sadaļa ir paplašināta ar specifiskām rekomendācijām, kas ir attiecināmas uz konkrētiem IS servisa veidiem.

## Projekta personāls

Projekta grupā ir jāiekļauj gan priekšmetiskā apgabala eksperti, gan IT speciālisti, kam ir plašas zināšanas par esošo sistēmu, lai nodrošinātu jaunā slāņa savietojamību kā biznesa loģikas ziņā, tā arī tehniskā ziņā.

Projekta sākuma stadijā visiem projekta dalībniekiem ir jāiepazīstas ar VISS prasībām un vadlīnijām, lai izveidotos vienots priekšstats par jēdzieniem un izmantotajiem tehnoloģiskajiem risinājumiem. Viens no iespējamiem variantiem ir noorganizēt nelielu semināru ar konsultantu līdzdalību, kuriem ir pieredze e-pakalpojumu vai integrācijas izstrādes projektos. Šādā seminārā var gūt priekšstatu par īstenojamo projektu un apzināt varbūtējās izglītošanas nepieciešamību, kas veidotu pamatu konkrētam apmācības plānam.

Iesaistītajiem arhitektiem un izstrādātājiem ir jābūt lietpratējiem (nepieciešamības gadījumā, izejot attiecīgu apmācību) ar SOAP, XML un XML shēmām saistītos jautājumos, kā arī tiem vajadzētu pārzināt dažādās, klientu sistēmās izmantotās platformas tādā līmenī, lai sistēmas servisu testēšanu varētu veikt pēc iespējas pilnīgāk.

Turklāt, par priekšrocību būtu uzskatāma situācija, ka projekta dalībnieki, it īpaši tehniski orientētie, labi pārzinātu un būtu guvuši pieredzi dalītajos risinājumos, lielu sistēmu izstrādāšanā un tīmekļa risinājumos.

## Sākotnējā analīze

Kad virs kādas jau esošas sistēmas tiek veidots IS servisu ”slānis”, ir svarīgi noskaidrot, konkrēti kādu informāciju būtu vēlams nodot VISS lietotāju un e-pakalpojumu izstrādātāju rīcībā ar jauno servisu starpniecību.

Nevajadzētu izvirzīt par mērķi visa datu modeļa vai visu pamatsistēmas funkciju nodošanu atklātībā, tā vietā vajadzētu noteikt, kādi dati un/vai funkcijas varētu interesēt plašu mērķauditoriju.

Kā labs starta punkts var noderēt attiecīgās iestādes pakalpojumu elektronizācijas plāni vai veidojamo e-pakalpojumu specifikācijas [8]. Šajos dokumentos jau ir apzināti nepieciešamie dati, identificētas attiecīgo lietotāju vajadzības un iespējamās operācijas.

Veicot analīzi, vienmēr nepieciešams atcerēties, kādas potenciālas izmantošanas iespējas VISS lietotāji un e-pakalpojumu izstrādātāji vēl varētu pieprasīt no konkrētā IS servisa.

## Prasību specifikācija

Kad tiek noteikta prasību specifikācija kādam integrācijas risinājumam, ir lietderīgi iestādei izvirzāmās prasības sadalīt divās daļās:

* iestādes apmaiņas infrastruktūra;
* iestādes biznesa loģika un procesi.

Svarīgi ir nodalīt funkcionālās prasības, kuras ir kopējas visiem servisiem (drošība, pieteikumu apstrāde, veiktspējas monitorings) un specifiskās funkcionālas prasības, kuras tiek izstrādātas katram servisam atsevišķi.

### Iestādes apmaiņas infrastruktūra

IS servisu „slāņa” izveide virs jau eksistējošas sistēmas nosaka, ka īpaša uzmanība ir jāpievērš tam, kādā veidā tīri tehniski šis slānis tiks izveidots un kādas būs saskarnes starp šo slāni un jau esošo sistēmu, kā arī starp šo slāni un citām – ārējām – sistēmām.

Izšķiroša nozīme ir tam, lai IS servisu ”slānis” neiejauktos esošās sistēmas infrastruktūrā, tāpēc ir jānodrošina, lai savienojums starp šīm divām sistēmām pēc iespējas netiktu apgrūtināts ar īpaši sarežģītām pārbaudēm un papildu pieprasījumiem no reģistra datubāzes (DB) puses. To, iespējams, panāk tādā veidā, ka IS servisu ”slāni” pēc iespējas ”piekļauj” esošajai sistēmai ar iepriekš izveidoto protokolu un metožu palīdzību. Tāpēc šo jauno ”slāni” zināmā mērā var uzskatīt par analogu tīmekļa saskarnei ar kādu sistēmu. Gan tīmekli, gan IS servisu slāņus var uzskatīt par lietotāja saskarnes slāņiem, kuri nodrošina piekļuvi informācijai kādā sistēmā. Šie slāņi izmanto pamatsistēmas datus, lai nodotu lietotāju rīcībā jaunas funkcijas, nemainot pamatsistēmas darbības veidu.

### IS servisu arhitektūras plānošana

Plānojot IS servisu arhitektūru, jāievēro šādas kopējas izstrādes prasības:

#### Servisu abstrakcija

Nebūtiskā informācija par servisiem ir jāabstrahē. Servisu tehniskajiem kontraktiem ir jāsatur tikai būtiskā informācija. Informācija par servisiem ir ierobežota ar to, kas publicēta servisa tehniskajā kontraktā. Servisu abstrakcijas principa primārais nolūks ir novērst nevajadzīgas (liekas) informācijas izplatīšanu par servisu.

Servisa lietotājiem par servisu nepieciešams publicēt tikai tā nolūku un iespējas, kā arī detaļas par to, kā servisu programmiski izsaukt un lietot. Nepublicējot specifisku informāciju par servisu (biznesa loģika, realizācija, implementācija), tiek nodrošināts, ka serviss var nākotnē attīstīties (piemēram, optimizēta servisa iekšējā biznesa loģika), vienlaikus nodrošinot izpildi saistībām, kādas serviss uzņēmies ar sākotnēji publicēto informāciju (servisa nolūks un iespējas).

#### Servisu autonomija

Servisiem ir jābūt pēc iespējas autonomiem. Servisu autonomija raksturo servisa realizācijas (implementācijas) neatkarību. Ja programmatūra eksistē autonomā izpildes režīmā, tā spēj izpildīt savas funkcijas neatkarīgi no ārējām ietekmēm. Jo vairāk serviss ir autonoms, jo mazāka ir tā atkarība no dažādām ārējām ietekmēm.

Servisu autonomijas principa nolūks ir veicināt divus primāros ieguvumus: ceļot servisu autonomijas līmeni, ceļas to uzticamība (*reliability*) un uzvedības pareģojamība.

#### Servisu komponējamība

Servisiem ir jābūt komponējamiem (izmantojamiem kompozīcijā). Neatkarīgi no kompozīcijas lieluma un sarežģītības, servisiem ir jābūt efektīviem kompozīcijas dalībniekiem. Servisiem ir jābūt spējīgiem piedalīties vairākās dažādās kompozīcijās, lai risinātu vairākas dažādas lielākas problēmas.

Servisu komponējamības princips ir savstarpēji saistīts ar servisu atkārtotas izmantojamības principu. Servisu komponējamības principā uzsvars ir uz to, lai servisi tiktu projektēti un realizēti tā, lai tie varētu efektīvi piedalīties vairākās servisu kompozīcijās pat tad, ja sākotnēji neeksistē prasības servisa dalībai kompozīcijā.

#### Servisu kontrakti

Servisu kontraktu princips paredz, ka projektējot servisus, servisu aprakstīšanai ir jāizmanto standartizēti kontrakti. Servisa kontrakts var sastāvēt no servisu aprakstošu dokumentu grupas, kurā katrs dokuments apraksta noteiktu daļu no servisa. Dokumenti var būt tehniski (standartizēti, strukturēti, mašīnlasāmi; piemēram, XML dokuments) un netehniski (piemēram, juridisks līgums). Servisa kontrakts vienmēr ietvers vismaz vienu vai vairākus tehniskos servisa aprakstus, bet var gadīties arī gadījumi, kad būs nepieciešamība pēc netehniskiem dokumentiem, kas papildina tehniskās nianses.

Piemēram, servisa, kas realizēts kā tīkla serviss, kontrakts var sastāvēt no šādiem servisu aprakstošiem dokumentiem:

* WSDL definīcija (tehnisks dokuments);
* XML shēmas definīcija (tehnisks dokuments);
* Servisa projektējuma apraksts, kas ietver servisa līmeņa vienošanās (netehnisks dokuments).

#### Servisu sasaiste

Servisu sasaistes princips paredz, ka projektējot servisus, tie jāprojektē tā, lai tie būtu vāji saistīti (*loosely coupled*). Konsekventi piekopjot sasaistes samazināšanu gan pašos servisos, gan saskarnēs starp tiem, notiek virzība uz tādu stāvokli, kurā servisu kontrakti kļūst neatkarīgāki no servisa realizācijas, un servisi kļūst aizvien neatkarīgāki viens no otra. Tā tiek radīta vide, kurā servisi un to izmantotāji var pēc nepieciešamības attīstīties neatkarīgi viens no otra.

#### Servisu atkārtota izmantojamība

Projektējot servisus, tie jāprojektē tā, lai tie būtu atkārtoti izmantojami. Servisiem ir jāsatur un jāizsaka agnostiska loģika. Servisi ir jāvar pozicionēt kā atkārtoti izmantojamus resursus. Servisu atkārtotas izmantojamības principā ar terminu „atkārtota izmantojamība” (*reusability*) tiek norādīts uz kādas komponentes (servisa) potenciālu tikt atkārtoti izmantotam, turpretī „atkārtota izmantošana” (*reuse*) ir kādas komponentes (servisa) atkārtotas izmantošanas fakts. Šis princips ir vērsts uz to, lai vairotu servisu potenciālu tikt izmantotiem atkārtoti.

#### Servisu stāvokļneatkarība

Projektējot servisus, tie jāprojektē tā, lai tiktu minimizēta nepieciešamība pēc stāvokļa informācijas.

Pārmērīga stāvokļa informācijas pārvaldība var kompromitēt servisa pieejamību, kā arī mazināt tā mērogojamības potenciālu. Līdz ar to, servisi jāprojektē tā, lai tie būtu atkarīgi no stāvokļa informācijas (*stateful*) tikai tad, kad tas tiešām ir nepieciešams.

#### Standartu lietošana

SOA kā arhitektoniskais modelis ir agnostisks attiecībā uz tehnoloģiskajām platformām. Servisu orientēts risinājums var būt veidots no servisiem, kas realizēti kā komponentes, tīkla servisi vai to kombinācijas. Šobrīd nozarē, kā vispiemērotākā tehnoloģiskā platforma SOA realizācijai tiek uzskatīta tīkla servisu platforma, kura tiek definēta, izmantojot dažādus nozares standartus un specifikācijas, kuras plaši atbalsta dažādi piegādātāji.

IS servisu darbības principi ir jābalsta uz šādām atvērtām tehnoloģijām un specifikācijām:

* XML Schema Definition Language (konkrētā realizācija saskaņā ar [1]);
* Web Services Description Language (WSDL);
* Simple Object Access Protocol (SOAP);
* Universal Description, Discovery, and Integration (UDDI);
* WS-I Basic Profile (versija 1.1 vai jaunāka).

Minētie profili sastāv no brīvi pieejamiem un atvērtiem tīkla servisu standartiem un specifikācijām (daudzas no tām minētas citās šajā dokumentā aprakstītajās prasībās), kā arī to skaidrojumiem, uzlabojumiem, interpretācijām un paplašinājumiem ar mērķi vairot tīkla servisu sadarbspēju.

#### Servisu detalizācijas pakāpe

Izstrādājot konkrēto IS servisu specifikāciju, nepieciešams analizēt, kādi dati ir pieejami un kādā veidā var tikt izmantoti citu sistēmu darba procesos. Bieži vien šie jaunie servisi sastāvēs no jaunu datu sagatavošanas vai esošo datu meklēšanas, taču ir iespējams definēt servisus, kuri var ietilpt sarežģītākos iestādes procesos.

Kad ir definēti attiecīgie servisi, balstoties uz lietošanas veidu citās sistēmās, ir jānosaka šo servisu forma.

Nepieciešams izvairīties no pārāk detalizētiem servisiem, kuru gaitā viena izsaukuma rezultātā tiek sniegta tikai viena noteikta informācijas vienība. Tā vietā ir jācenšas nodrošināt lielāka datu daudzuma sniegšanu katra izsaukuma gadījumā vai arī datu apkopošanu, lai līdz minimumam samazinātu viena servisa izsaukumu skaitu, jo katrs izsaukums aizņem laiku un noslogo sistēmas resursus.

Jāparedz arī tāda sadarbības iespēja, kad no iestādes puses tiek nodrošināts IS serviss, kas sniedz vairāk informācijas, nekā tas klientam ir nepieciešams, un VISS pusē notiek informācijas ierobežošana. Ar šādu pieeju vienu IS servisu būs iespējams izmantot vairāku klientu apkalpošanai. Jāraugās arī, lai netiktu veidoti pārāk ģeneratīvi servisi, t.i., servisi, kuriem klienta sistēmai pastāv daudz iespēju saņemtās atbildes grupēt pēc vajadzības. Šādus servisus ir grūti lietot un uzturēt.

Ja nav zināmas klienta sistēmas vajadzības un, līdz ar to, nav arī īsti zināms, kādi var būt pieprasījumi, lai šīs vajadzības tiktu apmierinātas, tad savus servisus var ieviest šādi:

1. Piemēram, ja tiek sniegta informācija saistībā ar personas kodu, tad var definēt virkni iekšēji pieejamu detalizētu servisu, kuri dod iespēju atrast nelielu skaitu datu elementu:

* Serviss A – pieņem personas kodu un atbildē sniedz personas vārdu, uzvārdu, dzimšanas datumu utt.
* Serviss B – pieņem personas kodu un atbildē sniedz personas adresi*.*

1. Pēc tam tiek definēts viens vai vairāki ārējām sistēmām pieejami mazāk detalizēti servisi, kuri savāc informāciju no detalizētajiem servisiem un nosūta to atpakaļ, piemēram:

* Serviss AB –pieņem personas kodu un atbildē sniedz personas datus un adresi*,*
* Serviss MultiAB – pieņem personu kodus noteiktā daudzumā un atbildē sniedz meklējamo personu datus un adreses*.*

Ar šīs metodes palīdzību iespējams nodrošināt minimālu nepieciešamo uzturēšanu pēc tam, kad kļūst zināmas klientu sistēmu vajadzības, jo, balstoties uz detalizētajiem servisiem, var individuāli veidot jaunus, mazāk detalizētus servisus.

## Analīze un dizains

### IS servisu standartizācijas pakāpes izvēle

Kā jau tika minēts 3.nodalījumā, standartizācijas pasākumus kāda projekta ietvaros ir iespējams regulēt atkarībā no laika un resursu pietiekamības.

Bieži vien lietotajos rīkos ir atrodama speciāla bibliotēka vai funkciju kopums, ar kuru vairāk vai mazāk automātiski var pārveidot esošo datu struktūru XML formātā. Taču, ja šīs XML nav saskaņotas ar VISS XML shēmu katalogā reģistrētām XML shēmām, tad netiek radīta iespēja to atkārtotai lietošanai vienlaicīgi ar citām sistēmām (skat. 3.nodalījumā). Tāpēc šo konvertācijas procesu alternatīvi iespējams īstenot jaunajā IS servisu slānī, piemēram, vispārēja mehānisma veidā, lai individuāli tiktu nodrošināta šo procesu atbilstība standartiem. Cita iespēja ir izpildīt to IS servisa klienta pusē (VISS serveru daļā).

Veicot projektējamo IS servisu analīzi, nepieciešams izvērtēt iepriekš minētos jautājumus un, pamatojoties uz šo vērtējumu, izvēlēties kādu noteiktu standartizācijas pakāpi.

### Servisa kvalitātes novērtējums

Veidojot e-pakalpojumus vai citus integrācijas risinājumus pēc SOA, tiek īstenota komunikācija starp dažādām lietojumprogrammām, kuras, līdz ar to, kļūst savstarpēji atkarīgas. Sistēmas kvalitāti (QoS) nosaka šo savstarpēji saistīto komponenšu veiktspēja, stabilitāte, darbspējas laiks u.c. parametri.

Tas rada jaunas problēmas, jo sistēmas QoS vairs nav iespējams noteikt, vadoties tikai pēc tās iekšējiem parametriem, bet nepieciešams ņemt vērā arī citu sistēmu QoS parametrus.

Sakarā ar to, ka uz servisiem orientēto risinājumu galvenais uzdevums ir integrācija ar ārējām sistēmām, īpaša uzmanība jāpievērš šādiem QoS rādītājiem:

* atbildes laika garantijas (parasti, sinhroniem servisiem tas nevar būt lielāks par 3-5 sekundēm),
* darbspējas laiks,
* remonta laiks – cik ilgs laiks var paiet no kļūmes rašanās līdz tās novēršanai,
* transakciju mehānismi,
* kas reglamentē šo servisu sniegšanu.

Atbildes laika garantijas ir jādefinē katram atsevišķam sistēmas servisam, jo katra servisa biznesa process apmaiņu nodrošinošās sistēmas iekšienē var būt ļoti atšķirīgs. Atbildes laiki, protams, vienmēr ir atkarīgi no transporta formas, t.i., ja par transporta kanālu tiek izmantots HTTP, tad atbildes laiku variācijas būs lielas.

Vienam servisam, piemēram, pietiek ar datu paņemšanu no vienas iekšējās sistēmas, turpretim citam servisam nāksies apkopot datus no vairākām atšķirīgām iekšējām sistēmām.

### Sinhronie vai asinhronie servisi

Apmaināmais datu daudzums ir nozīmīgs arī servisu dizainam. Jāapsver, pie kāda datu daudzuma var novest atsevišķie pieprasījumi, un kā šie pieprasījumi ietekmēs esošo sistēmu. Ja atgriežamo datu daudzums ir neliels, un atbildes laiks ir pietiekams, vieglākais ceļš ir sinhronu IS servisu izstrāde. Ja sagaidāms, ka atgriežamo datu daudzums varētu būt apjomīgs, vai pieprasījumus nevar izpildīt tiešsaistes režīmā, vai arī eksistē kāds kanāla ierobežojums, jāapspriež asinhrono servisu dizains, kas ir sarežģītāks no izstrādes viedokļa, bet tai pašā laikā ir daudz drošāks (skat. 4.4.4.sadaļu).

### Sistēmas noslogojuma novērtējums

Projektēšanas fāzē ir jānovērtē iespējamais sistēmas noslogojums. Ļoti būtiski ir, lai jaunie servisi nenoslogotu esošo sistēmu, tāpēc IS servisu slānis ir jākonstruē tā, lai nodrošinātos pret pamatsistēmas pārslodzēm. Pirmkārt, jānoskaidro, cik lielu noslodzi esošā sistēma var izturēt, neizejot no ierindas, un, vadoties pēc tā, ir jānosaka pieļaujamās slodzes augšējā robeža IS servisu slānī.

Šos robežlielums var izmantot, lai definētu noteikumus attiecībā uz klienta sistēmai sūtāmo pieprasījumu lielumu, piemēram, pieprasījums par personu kodiem nedrīkst pārsniegt 10 personas vienā reizē.

Ja izrādās, ka paredzamie pieprasījumi varētu būt tādi, ka tie radītu pārāk lielu noslodzi esošajai sistēmai, piemēram, pieļaujot personas datu nosūtīšanu par vairāk nekā 10 000 iedzīvotajiem vienā reizē, ir jāizveido alternatīvs mehānisms.

Viens no iespējamiem risinājumiem ir reģistra kopiju izmantošana atsevišķā datu bāzes sistēmā, tas nodrošinātu, ka sistēmas servisi strādās ar precīzu esošās sistēmas datu dublikātu (replicēto kopiju) un nevis ar oriģinālo reģistru.

Reizēm varētu apspriest gadījumus, kad lietderīgi būtu izmantot asinhronu komunikāciju starp IS servisu slāni un pamatsistēmu. Tā rezultātā jaunie pieprasījumi nostājas ”rindā” un tiek apkalpoti tad, kad rodas iespēja, līdz ar to, IS servisu pieprasījumi netraucē procesiem, kurus pamatsistēma veic attiecīgajā brīdī.

### Auditāciju plānošana

Analīzes fāzē ir jāizvērtē, cik kritiski ir saņemamie un pārsūtamie dati, un, vadoties pēc šī vērtējuma, jādefinē auditējamo datu apstrādes stratēģija. Ja IS serviss kā izpildes rezultātu saņem vai atdot personas datus (personas kods, vārds, uzvārds, dzimšanas datums), jāparedz speciāli auditācijas žurnāli ar meklēšanas saskarni, kas atbilstu Personas datu aizsardzības likumam [3].

Integrācijas IS servisu audits notiek VISS Pieprasījumu servisā. Kā arī VISS piedāvā speciālu lietotāju saskarni auditējamo datu meklēšanai un analīzei.

### IS servisu projektējumu izstrāde

Viss, ko nevar definēt IS servisa aprakstā un izmantotajās XML shēmās (saskaņā ar [1]), ir jādefinē IS servisa projektējumā, lai klientu un servera sistēmas: no vienas puses – vienādi interpretētu jēdzienus, bet no otras puses – vienādi interpretētu pienākumu sadalījumu.

IS servisa projektējums varētu būt daļa no cita sistēmas PPA, kā arī iespējams definēt vairākus IS servisus (ja tie ir loģiski saistīti) vienā projektējumā. Tomēr Integrācijas servisiem, īpaši gadījumos, kad tie potenciāli tiks lietoti vairākos e-pakalpojumos vai integrācijas uzdevumos, rekomendēts veidot atsevišķu projektējumu, kas ietver tikai nepieciešamo informāciju par IS servisu. IS servisa projektējums tiek veidots atbilstoši LVS [2], papildinot to ar aprakstošām sadaļām, kas ļautu sasniegt dokumenta pabeigtību. Aptuvens projektējuma saturs ir dots zemāk:

1. IEVADS

1.1. Dokumenta nolūks

1.2. Darbības sfēra

1.3. Definīcijas un saīsinājumi

1.3.1. Definīcijas

1.3.2. Saīsinājumi

1.4. Saistītie dokumenti

1.5. Dokumenta pārskats

2. IS SERVISA DEFINĪCIJA

2.1. Vispārīgs apraksts

2.2. Konceptuālā uzbūve

2.3. Izpildes scenārijs

2.4. Iestādes IS apraksts

3. ATKARĪBU APRAKSTS

3.1. IS servisa servera un klienta daļas atkarības

3.1.1. Datu drošības pasākumi

3.1.2. Autentifikācija un šifrēšana

4. DETALIZĒTS PROJEKTĒJUMS

4.1. Detalizēts moduļu projektējums

4.1.1. Modulis 1

4.1.2. Modulis 2

4.1.3. Modulis 3

4.2. Datu detalizēts projektējums

4.2.1. XML shēmas tips 1

4.2.2. XML shēmas tips 2

4.2.3. XML shēmas tips 3

4.3. Vienlaicīgo procesu dekompozīcija

5. IS SERVISA KVALITĀTES RĀDĪTĀJI

IS servisa vispārīgais apraksts (skat. 2.1.nodaļu) precizē prasību specificēšanas laikā izstrādāto IS servisa definīciju, tai skaitā:

* IS servisa tips.
* Nosaukums.
* Īss apraksts.
* Lietotāja instrukcija.
* Ieejas dati.
* Apstrāde.
* Rezultātā saņemtie dati.
* IS servisa veids (sinhrons, asinhrons).
* Ierobežojumi.
* Reģioni, kuros IS serviss tiek sniegts.
* Iestāde, kura ir atbildīga par IS servisa sniegšanu.
* Reģistrs, kas nodrošina IS servisa sniegšanu.
* Maksājumu informācija.
* Normatīvie akti, kas reglamentē IS servisa sniegšanu.
* Minimālais autentifikācijas līmenis.

Papildus minētajiem IS servisa atribūtiem Integrācijas servisiem obligāti, bet biznesa un klasifikatoru IS servisiem – vēlams, jādefinē IS servisa VISS URN identifikators pēc šāda standarta: URN:IVIS:100001:ISS-<iestādes VISS ident>-<IS servisa angļu nosaukums>-v<versija, *major*-*minor*>, kur

* <iestādes VISS ident> – IS servisa izstrādātājas iestādes (īpašnieka) identifikators, kļūst pieejams pēc reģistrācijas VISS, piemēram, TM.VVDZ, TM.VZD, IeM.PMLP utt.
* <IS servisa angļu nosaukums> – parasti tas ir IS servisa WSDL elementa *service* atribūts *name,* piemēram, *GetPersonEstateList*.
* <versija, *major*-*minor*> – IS servisa versija, tas major un minor komponente, piemēram, 1-0, 1-1 utt.

IS servisa, kas pēc norādītā personas koda atgriež personas īpašumu sarakstu, identifikatora piemērs: URN:IVIS:100001:ISS-TM.VVDZ-GetPersonEstateList-v1-0.

PPA 2.2. un 2.3.nodaļā - Konceptuālā uzbūve un Izpildes scenārijs – tiek aprakstīta IS servisa konceptuālā arhitektūra, tai skaitā, IS servisa klienta un servera daļa (ja tās eksistē), kā arī neformālā veidā tiek aprakstīts IS servisa darbības scenārijs.

PPA 3.nodalījumā - ATKARĪBU APRAKSTS – ir dots IS servisa servera un klienta daļas atkarību apraksts (ja nepieciešams), datu drošības pasākumi, ja starp IS servisa servera un klienta daļu tiek veikta datu šifrēšana vai kāda speciāla protokola lietošana. Jāpievērš uzmanība šādiem drošības faktoriem:

* autorizācijas mehānisms,
* digitālu sertifikātu lietošana,
* transporta protokola šifrēšana,
* datu šifrēšana,
* iestāžu Virtual Private Network (VPN) infrastruktūras izmantošana,
* aizsardzība pret „hakeru” uzbrukumiem, piemēram, ar IP bloķēšanu un nodrošinājumu pret Denial of service attacks (*DoS attack*),
* servisu iedalījums drošības grupās.

PPA 4.nodalījumā ir dots IS servisa detalizēts projektējums. Katra moduļa aprakstam ir jāsatur vismaz šādas sadaļas: identificējums, apraksts, ievade, apstrāde, izvade un kļūdas. Sadaļu saturs ir definēts LV Standartā [2], nepieciešams pievērst uzmanību tikai tiem punktiem, kas ir svarīgi IS servisā. Moduļu aprakstam jāsatur informācija par ievades un izvades datu validēšanu, piemēram:

1. Serviss ievadē saņem kādas personas kodu un atbildē sniedz tās personas vārdu, kurai attiecīgais personas kods ir piešķirts. Rodas jautājumi:

Vai klienta sistēmai ir jānodrošina, ka persona ar attiecīgo personas kodu patiešām eksistē vēl pirms pieprasījuma izdarīšanas servisu sniedzēja sistēmai?

Ja tiek nolemts, ka par šādu validēšanu atbild klienta sistēma, tad jāņem vērā, ka tas apgrūtina uzņēmuma loģikas uzturēšanu un uz tās pamata veidoto dokumentāciju.

1. Ir arī svarīgi dokumentēt pušu rīcību kļūdas rašanās gadījumā:

Kā rīkojas servisu sniedzējs kļūdas situācijā, ja, piemēram, klienta sistēma savā pieprasījumā ir nosūtījusi nepareizi formatētu personas kodu?

Vai atpakaļ ir jānosūta tukša atbilde vai kāds konkrēts biznesa kļūdas kods, ja, piemēram, pēc saņemtā personas koda nekas netiek atrasts?

Kļūdas sekcijā jādefinē visas iespējamās kļūdu situācijas (servisu sniedzēja sistēmas iziešana no ierindas, kļūda servera transakcijā utt.) starp sistēmām, lai būtu skaidrs, kā ir jārīkojas attiecīgā situācijā.

Integrācijas IS servisos tiek definēta speciāla XML sekcija (skat. 2.1.nodaļu), kur IS servisa izstrādātājs definē visas iespējamā kļūdas un citus notikumu tipus, tai skaitā, brīdinājumus (warning), informāciju (information) un atkļūdošanas informāciju (verbose).

Datu detalizēts projektējums (4.2.nodaļa) parasti paredz lietoto XML shēmu tipu aprakstu, kas varētu ietvert XML shēmas tipa ER diagrammu un XML ziņojuma piemēru.

IS servisa kvalitātes rādītāju (5.nodalījums) tabula sniedz šādu kvalitātes atribūtu uzskaitījumu saskaņā ar 4.4.2.sadaļu: atbildes, remonta un darbspējas laiks, maksimālais pieprasījumu skaits laika intervālā u.c.

### Testu plānošana

IS servisu slānim ir jāiziet virkne pamatīgu veiktspējas testu, un ir svarīgi, lai šie testi tiktu izpildīti saskaņā ar vairākiem atšķirīgiem scenārijiem.

Ja, piemēram, ir izveidots IS servisu slānis, kurš kā transporta formātu starp servera un klienta daļu izmanto HTTP, varētu būt lietderīgi izmantot šādus testa scenārijus:

* tests, lai pārbaudītu IS servisa spēju nosūtīt datus caur sistēmu:  
  tiek izveidota servisa metode, kura atgriež ievades datus, tos neapstrādājot un neveicot pamatsistēmas izsaukšanu (uz „aizbāžņiem”), pēc tam jāizmēra tās atkārtotas izsaukšanas veiktspēja,
* vienkārša pieprasījuma testēšana dažādos veidos:
* nešifrēts transports (piemēram, HTTP),
* šifrēts transports (piemēram, HTTPS),
* šifrēts XML pieprasījums un/vai atbilde pēc WS-Security standarta, ja tāds tiek pielietots,
* vairāku atšķirīgu vienlaicīgu pieprasījumu testēšana, lai noteiktu mērogojamību.

## Izstrāde

Konkrētas izstrādes rekomendācijas, kodēšanas vadlīnijās utt. ir atkarīgas no lietojamās platformas un uzdevuma. Viena no esošām vadlīnijām aptver IS servisu un e-pakalpojumu izstrādi MS Windows .NET platformā [8] un ir paredzēta IS servisiem, kas tiks izmitināti VISS vidē.

## Testi

### Testa klientu izmantošana

Lai nodrošinātu maksimālu sadarbību starp dažādām platformām, testēšanas fāzē ir jāizstrādā testa klienti attiecīgajām platformām/ tehnoloģijām.

Šie testa klienti demonstrē, kādā veidā iespējams izsaukt piedāvātos servisus un kā var apstrādāt saņemtās atbildes. Kopā ar dokumentāciju tie definē izmantotos jēdzienus un to savstarpējo sakarību sistēmā.

IS servisu slāņu testēšanai arī iespējams lietot specializētus klientus: WSStudio un WSTest lietojumus. Tos ir iespējams lejupielādēt VISS portālā.

### Testēšanas servisi

Dažos gadījumos jāizstrādā īpaša testēšanas sistēma vai virkne testēšanas servisu, kurus klientu sistēmas varētu izmantot to testēšanas fāzēs. Šīs sistēmas bieži tiek izmantotas testēšanas fāzē tādu kļūdu situāciju provocēšanai, kam nevajag izmantot darba sistēmas. Kā piemēru var minēt darbspējas testus: to laikā mēģina noskaidrot maksimālo noslogojuma pakāpi, šajā nolūkā pārkāpjot maksimāli pieļaujamās robežas.

Var izstrādāt īpašu testēšanas servisu, kurā tiek reģistrētas atsevišķas, ar veikto pieprasījumu saistītas tehniskās detaļas, tādā veidā noskaidrojot, vai klienta sistēma ievēro starp sistēmām noslēgto līgumu un pareizi izsauc attiecīgo servisu.

## Piegāde

### Piegādes nodevumi

Servisu sniedzēja piegāde VISS IS servisu katalogam ietver IS servisu aprakstu saskaņā ar 5.nodalījumu – IS SERVISA KVALITĀTES RĀDĪTĀJI. IS servisiem, kas tiks izmitināti VISS infrastruktūrā, jānodrošina šādu nodevumu saraksts:

* IS servisa izejas teksti.
* IS servisa uzstādīšanas programmatūra.
* Administratora uzstādīšanas un uzturēšanas rokasgrāmata.

### IS servisa reģistrācija VISS IS servisu katalogā

Katrs IS serviss tiek reģistrēts VISS IS servisu katalogā. Reģistrācija ir nepieciešama, lai:

* uzturētu IS servisu aprakstošos datus,
* dotu iespēju veikt IS servisu testēšanu un pārraudzību.

## Ekspluatācija

### Vadība

Uz servisiem orientēta risinājuma gadījumā, kad dažādi lietotāji izmanto dažādu servisu kombinācijas, ir svarīgi nodrošināt to klientu pārvaldību, kam ir piekļuve tiem vai citiem konkrētiem servisiem, un kontrolēt, kas izmanto tos vai citus servisus.

Tāpēc VISS IS servisu katalogs nodrošina iespēju atvērt un aizvērt piekļuvi servisiem un saņemt statistikas datus par izmantošanu pēc klientiem, e-pakalpojumiem vai integrācijas uzdevumiem, servisu veidiem un klientu autentifikācijas pakāpes.

Jāpieļauj situācija, kad dažādi klienti var lietot viena servisa dažādas versijas.

### Versijas

Ja tiek gatavotas piedāvāto servisu jaunas versijas, ir jāizlemj, kādā veidā notiks sadalīšana versijās. Versiju sadalīšana notiek gan iestādes pusē, gan VISS IS servisu kataloga pusē.

Pirmkārt, nepieciešams pieņemt lēmumu par to, kā jaunie servisi tiks saukti. Ilustrācijai izmantosim sistēmu, kura piedāvā URL bāzes servisus, t.i., klientu sistēmas var piekļūt servisiem, izmantojot Interneta adresi <http://pmlp.iem.gov.lv/IVIS/V1-0/IVISService.aspx>. Ja tiek izlaista jauna *IVISService* versija, tas tiek izdarīts šādi: <http://pmlp.iem.gov.lv/IVIS/V2-0/IVISService.aspx>.

Ja tiek izlaists jauns serviss, tad nepieciešams saglabāt arī tā veco versiju, ja ne pastāvīgi, tad vismaz uz laiku, kamēr visi IS servisa lietotāji sāks izmantot jauno servisu. Dažādās versijas var vienlaicīgi pastāvēt servisu sniedzēja sistēmā, un klientu sistēmas var vienkārši pāriet uz jauno versiju tad, kad tas atbilst klientu plāniem.

# IS servisu apraksts

VISS IS servisu katalogs tiek lietots kā IS servisu telefongrāmata (https://lvp.viss.gov.lv/VISS.ISSK), kur iespējams meklēt un apskatīt informāciju par visiem reģistrētiem IS servisiem. IS servisa standartelementu vārdnīcas apraksts ir sniegts 4.tabulā.

4.tabula

IS servisa standarta XML elementu vārdnīca

| Elementi, 1., 2. un 3. līmenis | Min/Max | Paskaidrojums |
| --- | --- | --- |
| Nosaukums | 0..1 | IS servisa īsais nosaukums, parasti angļu valoda |
| Virsraksts | 0..n | Alternatīvs IS servisa nosaukums latviski |
| Status | 1..1 | IS servisa statuss (DRAFT, PUBLISHED) |
| Versija | 0..1 | IS servisa versija |
| Identifikators VISS (IVIS) | 1..1 | IS servisa unikālais identifikators saskaņā ar Metadatu un IS servisu standartu |
| Īpašnieks | 1..1 | Entītija (iestāde, komercuzņēmums vai cilvēks), kas ir atbildīga par IS servisu izveidi |
| Izveidošanas datums | 0..1 | IS servisa (vai konkrētas versijas) izveides datums |
| Modificēšanas datums | 1..1 | Datums, kad IS servisa apraksts tika modificēts |
| Apraksts | 0..1 | IS servisa īss apraksts |
| Maksa | 0..1 | Informācija par IS servisa sniegšanas izmaksām |
| Autentifikācijas līmenis | 0..1 | IS servisa autentifikācijas līmenis:   * 0. līmenis – nedeklarētā (anonīmā) identitāte. * 1. līmenis – deklarētā identitāte, kuras ietvaros klients deklarē savu identitāti (faktisko vai elektronisko), taču tā netiek apliecināta, jeb pārbaudīta. * 2. līmenis – apliecinātā identitāte, kuras ietvaros klients noteiktā veidā apliecina savu identitāti, piemēram, uzrādot pasi klātienē. * 3. līmenis – kvalificēti apliecinātā identitāte. |
| Wsdl URN | 0..1 | IS servisa wsdl, atsauce uz VISS Resursu katalogā ievietoto wsdl XML datni |
| Maksa | 0..1 | Maksa par IS servisa lietošanu |
| Instrukcija | 0..n | Instrukcija, kā lietot IS servisu |
| Sinhronitāte | 0..1 | IS servisa veids: sinhrons vai asinhrons |
| Prasības | 0..n | Papildus nosacījumi, kad IS serviss var būt pieejams saņēmējam |
| Kontaktinformācija | 0..1 | IS servisa izstrādātāja kontaktinformācija |
| Ieejas parametri | 0..1 | Ienākošie parametri.  Integrācijas IS servisiem parametrs tiek aizpildīts, lietojot IS servisa WSDL WebServiceInfo sekciju, kas parasti satur XML shēmas tipa URN identifikatoru (skat. 2.2.nodaļu). |
| Izejas parametri | 0..1 | Izejošie parametri.  Integrācijas IS servisiem parametrs tiek aizpildīts, lietojot IS servisa WSDL WebServiceInfo sekciju, kas parasti satur XML shēmas tipa URN identifikatoru (skat. 2.2.nodaļu). |
| Servisa URL | 0..1 | IS servisa produkcijas versijas URL adrese |
| Servisa testa URL | 0..1 | IS servisa testa versijas URL adrese |
| Atslēgvārdi | 0..n | Vārdi vai nosacījumi, kuri tiek lietoti, lai IS servisa priekšmetisko iemeslu aprakstītu tik specifiski, cik tas ir iespējams. Tos vajadzētu ņemt no kontrolējamas vārdnīcas vai saraksta. |
| Tips | 0..n | IS servisa tips:   * Integrācijas IS serviss – VISS integrācijas IS serviss; * Pieprasījumu serviss – konkrēta servisa tips; * Klasifikators – palicis no VISS iepriekšējās versijas, dažreiz klasifikatori izvietoja izmantojot Web servisu; lai uzsvērtu, ka tas ir klasifikators tika izveidots atsevišķs veids; * Ģeotelpiskie servisi – ĢDS un Ģeoportāla projekta ietvaros; * E-veselības servisi – E-veselūibas projekta ietvaros; * Citi Web servisi – gan droši, gan nedroši biznesa servisi. |
| ErrorList | 0..1 | Integrācijas IS servisa kļūdu (notikumu) saraksts tiek aizpildīts, lietojot IS servisa WSDL WebServiceInfo sekciju (skat. 2.2.nodaļu) |
| Error | 0..n | IS servisa kļūdas (notikumu) apraksts |
| Code | 1..1 | Atribūts. Kļūdu (notikumu) kods |
| Severity | 1..1 | Atribūts. Notikuma svarīgums: kļūda (*error*), brīdinājums (*warning*), informācija (*information*) un atkļūdošanas informācija (*verbose*). |

1. OASIS standarts "Web Services Security v1.0" (WS-Security 2004), kas ietver SOAP Message Security 1.0 (WS-Security 2004), Web Services Security UsernameToken Profile 1.0 un Web Services Security X.509 Certificate Token Profile [↑](#footnote-ref-1)
2. OASIS standarts "Web Services Security v1.1", kas ietver WS-Security Core Specification 1.1, Username Token Profile 1.1, SAML Token Profile 1.1, X.509 Token Profile 1.1, Kerberos Token Profile 1.1, Rights Expression Language (REL) Token Profile 1.1 un SOAP ar Attachments (SWA) Profile 1.1 [↑](#footnote-ref-2)