

ПРИЛОЖЕНИЕ № 12
к протоколу заседания подкомиссии
по использованию информационных технологий
при предоставлении государственных
и муниципальных услуг Правительственной комиссии
по использованию информационных технологий
для улучшения качества жизни и условий ведения
предпринимательской деятельности
от 3 марта 2017 г. № ____

ОДОБРЕНО
подкомиссией по использованию
информационных технологий при предоставлении
государственных и муниципальных услуг
Правительственной комиссии по использованию
информационных технологий для улучшения качества жизни
и условий ведения предпринимательской деятельности
(протокол от 3 марта 2017 г. № ____)

ИНФРАСТРУКТУРА ЭЛЕКТРОННОГО ПРАВИТЕЛЬСТВА

**Методические рекомендации по работе
с Единой системой
межведомственного электронного взаимодействия
версия 3.3.0.0**

Москва 2016

История документа

Версия	Дата	Автор	Комментарии
3.0.9.8	16.06.2015	Чернявский В.Е.	Обновлён глоссарий Обновлены вложенные файлы схем СМЭВ. Дополнен п.2.3.1. Добавлен п. 2.3.6. Дополнен п.3.2.1.
	17.06.2015	Анна Миронова	Исправлена нумерация таблиц, рисунков, а также ссылки на рисунки и таблицы по всему документу. Исправлен абзац 2 п.2.1. Добавлен раздел «История документа».
3.0.9.8.1	23.06.2015	Чернявский В.Е.	Обновлена wsdl-схема СМЭВ (мелкое изменение)
3.0.9.8.2	19.10.2015	Чернявский В.Е.	Описание работы с Файловым хранилищем
3.1.0.0	19.10.2015	Чернявский В.Е.	Описание схемы 1.1* Описание тега передачи кода ФРГУ
3.1.0.1	29.08.2016	Чернявский В.Е.	Мелкие исправления в названиях схем (номера 1.2 на 1.1), более подробно описан элемент SmevFault, описано новое значение элемента RequestRejected (FAILURE), исправлены ошибки в диаграммах
3.2.0.0	15.09.2016	Чернявский В.Е.	Дополнения, описывающие новый функционал СМЭВ, внедренный в 2016 году: а) Добавлен раздел №8 б) Откорректирован раздел 2.2 (перенесен в 3.6) в) Обновлена схема smev-message-exchange-types-1.1 г) Добавлены разделы 9 и 10, а также приложение 8.

3.3.0.0	30.11.2016	Чернявский В.Е.	Доработан раздел 2.6.3; Значительно доработан раздел 3 (доработаны схемы и описания к ним, добавлен раздел 3.6); Добавлен раздел 2.2.5 (Обмен справочными данными в составе сообщений по видам сведений в СМЭВ). Утратили актуальность и удалены разделы 2.5.3 и 2.5.4.
3.3.0.0	10.02.2017	Ковалев Р.С.	В п. 2.2.1 скорректировано толкование термина «вида сведений»; в п. 2.3.3 скорректировано толкование «широковещательных рассылок»; добавлен п. 2.3.4 «Реестровый вид сведений»; добавлен п. 3.8 «Структура сообщений с вложениями»
3.3.0.0	01.03.2017	Чернявский В.Е.	Уточнены формулировки п. 2.2.5

Содержание

1.	Введение.....	8
1.1.	Назначение документа.....	8
1.2.	Цели и требования	9
1.3.	Термины, определения, соглашения	10
2.	Основы взаимодействия	12
2.1.	Основные понятия и правила обмена информацией	12
2.1.1	Понятия и правила обмена информацией	12
2.1.2	Схемы единого электронного сервиса	13
2.2.	Концепция «Виды сведений»	14
2.2.1	Толкование термина	14
2.2.2	Маршрутизация запросов на основании передаваемых сведений.....	14
2.2.3	Маршрутизация запросов по коду маршрутизации	15
2.2.4	Требования к описанию форматов сведений	16
2.2.5	Обмен справочными данными в составе сообщений по видам сведений в СМЭВ 17	
2.2.6	Версионность форматов сведений	17
2.2.7	Структура вида сведений в СМЭВ.....	18
2.3.	Типы сообщений	18
2.3.1	Сообщения типа «Запрос»	18
2.3.2	Сообщения типа «Ответ»	19
2.3.3	Широковещательные рассылки.....	19
2.3.4	Приоритетная доставка	22
2.3.5	Сообщения об отказах в ответе на уровне схемы СМЭВ	22
2.3.6	Возврат статусов запросов на уровне схемы СМЭВ	22
2.3.7	Указание мнемоники сервера отправки сообщения.....	23
2.4.	Жизненный цикл сообщений	24
2.4.1	Жизненный цикл сообщения типа «Запрос».....	24
2.4.2	Жизненный цикл сообщения типа «Ответ»	25
2.4.3	Жизненный цикл бизнес-взаимодействия	26
2.5.	Организация очередей	26
2.5.1	Получение сообщения с фильтрацией по виду сведений	27
2.5.2	Подтверждение приёма сообщения	28
2.6.	Организация очередей статусов	29
2.6.1	Получение уведомления из очереди статусов	29
2.6.2	Структура сообщения с запросом статусного сообщения из очереди статусов 31	

2.6.3	Структура статусного сообщения из очереди статусов	31
3.	Требования к структуре сообщений	34
3.1.	Общие положения	34
3.2.	Структура сообщения с запросом сведений, которое ИС потребителя передает в СМЭВ	35
3.2.1	Блок данных запроса	37
3.2.2	Блок содержимого вложений	38
3.2.3	Электронная подпись органа власти	39
3.3.	Структура сообщения с запросом сведений, которое ИС поставщика получает из СМЭВ	39
3.3.1	Блок данных СМЭВ-конверта	41
3.3.2	Блок содержимого вложений	42
3.3.3	Электронная подпись СМЭВ	42
3.4.	Структура сообщения с ответом, которое ИС поставщика передает в СМЭВ	42
3.4.1	Блок данных ответа	44
3.4.2	Блок содержимого вложений	45
3.4.3	Электронная подпись органа власти	45
3.5.	Структура сообщения с ответом, которое ИС потребителя получает из СМЭВ	46
3.5.1	Блок данных СМЭВ-конверта	47
3.5.2	Блок содержимого вложений	49
3.5.3	Электронная подпись СМЭВ	49
3.6.	Структура сообщения с ответом о статусе ранее отправленного в СМЭВ сообщения, которое ИС потребителя получает из СМЭВ	50
3.6.1	Блок данных СМЭВ-конверта	51
3.6.2	Электронная подпись СМЭВ	52
3.7.	Особенности схемы сервиса СМЭВ 1.1* (1.2)	52
4.	Электронные подписи	58
4.1.	Виды электронных подписей	58
4.2.	Порядок использования электронных подписей	58
4.2.1	Использование электронных подписей при передаче запроса	58
4.2.2	Использование электронных подписей при передаче ответа	61
4.3.	Правила формирования ЭП	62
4.3.1	Подписи в формате PKCS#7	62
4.4.	Электронные подписи субъектов взаимодействия – физических лиц	63
4.4.1	Общие требования к электронной подписи, формируемой от имени должностных лиц органов власти при межведомственном информационном обмене	

4.4.2	Электронная подпись при межведомственном взаимодействии	63
4.5.	Электронные подписи субъектов взаимодействия – информационных систем.....	65
4.5.1	Общие требования электронной подписи, формируемой от имени органа власти при межведомственном информационном обмене	65
4.5.2	Общие требования к электронной подписи, формируемой узлами СМЭВ	65
4.5.3	Правила формирования электронной подписи информационной системы.....	66
4.5.4	Подписание вложений электронной подписью информационной системы.....	67
5.	Пересылка вложений с использованием Файлового хранилища	68
6.	Сценарии асинхронного взаимодействия	72
6.1.	Межведомственный запрос.....	72
6.2.	Пакет запросов – пакет ответов.....	76
7.	Взаимодействие через системы-агрегаторы	79
8.	Использование системы генерации кодов транзакций.....	80
9.	Использование push-нотификаций	95
9.1.	Описание сервиса приема push-уведомлений	95
10.	Использование клиентской библиотеки и набора «toolset»	101
11.	Приложения	102
11.1.	Приложение 1: Алгоритм нормализации XML	102
11.2.	Приложение 2: Результат трансформации urn://smev-gov-ru/xmldsig/transform	104
11.3.	Приложение 3: Профиль формата PKCS#7, которому должны удовлетворять подписи вложенных файлов.....	105
11.4.	Приложение 4: Образцовая реализация трансформации urn://smev-gov-ru/xmldsig/transform.....	106
11.5.	Приложение 5: Сценарии тестирования алгоритма нормализации XML.....	113
11.5.1	Сценарий 1: тестирование правил 1, 2, 6 (здесь и далее под правилами понимаются подпункты алгоритма нормализации, описанного в Приложении 1).....	113
11.5.2	Сценарий 2: тестирование правил 4, 5.....	113
11.5.3	Сценарий 3: тестирование правил 3, 7, 8.....	114
11.6.	Приложение 6: Перечень ошибок, возвращаемых транспортной подсистемой СМЭВ	115
11.7.	Приложение 7: Формирование видов сведений с включением справочников	126
11.8.	Приложение 8: Описание набора «toolset»	128
11.8.1	Состав набора	128
11.8.2	Требования к ПО для запуска набора	129
11.8.3	Порядок запуска утилит набора	129

11.8.4	Установочная конфигурация	130
11.8.5	Описание входных и выходных параметров.....	134

1. ВВЕДЕНИЕ

1.1. НАЗНАЧЕНИЕ ДОКУМЕНТА

Требования, указанные в документе, следует рассматривать в дополнение к требованиям, содержащимся в приказе Министерства связи и массовых коммуникаций Российской Федерации от 27 декабря 2010 г. № 190 «Об утверждении технических требований к взаимодействию информационных систем в единой системе межведомственного электронного взаимодействия».

В рамках документа рассматриваются следующие вопросы:

- Структура электронного сообщения, служебные блоки данных в передаваемых в СМЭВ сообщениях.
- Правила применения и форматы электронной подписи, формируемой от имени должностных лиц органов власти при межведомственном информационном обмене.
- Правила применения и форматы электронной подписи, формируемой от имени органа власти при межведомственном информационном обмене.
- Правила применения и форматы электронной подписи, формируемой системой межведомственного электронного взаимодействия при обработке электронных сообщений, передаваемых через нее.
- Правила заполнения служебных элементов электронных сообщений СМЭВ, определяемые необходимостью формирования целостных отчетов об истории обмена электронными сообщениями через СМЭВ в рамках оказания государственных услуг или выполнения государственных функций, а также формирования аналитических отчетов по межведомственному взаимодействию.

Описываемые в документе правила являются обязательными к применению участниками информационного обмена с использованием системы межведомственного электронного взаимодействия.

Документ содержит описание форматов сообщений и алгоритмов формирования различных типов электронной подписи, применяемой в электронных сообщениях, передаваемых в СМЭВ.

В данный момент номер Методических рекомендаций формируется по шаблону А.Х.У.З, где:

А – номер поколения единой системы межведомственного электронного взаимодействия.

Х – номер поколения документа. Изменение данного номера означает значительные изменения в структуре и содержании документа.

У – Номер основного релиза документа в рамках поколения. Документ может содержать освещение новых и/или незначительную переработку содержащихся в предыдущей версии документа тем. Плановая подготовка основного релиза документа

осуществляется раз в квартал. Основные релизы утверждаются Подкомиссией по использованию информационных технологий при предоставлении государственных и муниципальных услуг Правительственной комиссии по внедрению информационных технологий в деятельность государственных органов и органов местного самоуправления.

Z – номер технологического релиза в рамках основного релиза. Может содержать в себе стилистические, редакционные, незначительные технические изменения. Данные тип релизов выпускается по необходимости и не проходит специализированной процедуры утверждения Подкомиссией по использованию информационных технологий при предоставлении государственных и муниципальных услуг Правительственной комиссии по внедрению информационных технологий в деятельность государственных органов и органов местного самоуправления.

1.2. ЦЕЛИ И ТРЕБОВАНИЯ

Данный документ разработан в целях реализации и во исполнение:

- Федерального закона от 27 июля 2010 г. № 210-ФЗ «Об организации предоставления государственных и муниципальных услуг».
- Федерального закона от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи».
- Постановления Правительства Российской Федерации от 9 февраля 2012 года № 111 «Об электронной подписи, используемой органами исполнительной власти и органами местного самоуправления при организации электронного взаимодействия между собой, о порядке ее использования, а также об установлении требований к обеспечению совместимости средств электронной подписи».
- Постановления Правительства Российской Федерации от 8 сентября 2010 г. № 697 «О единой системе межведомственного электронного взаимодействия» (далее Постановление № 697).
- а также в рамках реализации:
- Соглашений о взаимном признании электронных подписей, заключенных между Минкомсвязью РФ и федеральными органами исполнительной власти.
- Соглашений о взаимодействии при обеспечении оказания (исполнения) государственных (муниципальных) услуг (функций) федеральными органами исполнительной власти, заключенных между Минкомсвязью РФ и федеральными органами исполнительной власти.

1.3. ТЕРМИНЫ, ОПРЕДЕЛЕНИЯ, СОГЛАШЕНИЯ

В документе используются следующие термины и определения:

Бизнес-транзакция	Набор транзакций (пар запрос-ответ) между одним Потребителем и Поставщиком сведений, связанных единой бизнес-логикой получения сведения.
ГРУ	Геораспределенный узел СМЭВ
ГУЦ	Информационная система головного удостоверяющего центра
ЕПГУ	Единый портал государственных и муниципальных услуг (функций)
ЕСНСИ	Единая система справочников и классификаторов, используемых в государственных и муниципальных информационных системах (Единая система нормативно-справочной информации)
ИС	Информационная система
НИК	Набор инструментов клиента СМЭВ
СГКТ	Система генерации кодов транзакций
СПКТ	Сервис предоставления кодов транзакций СГКТ
СМЭВ	Система межведомственного электронного взаимодействия
ОВ	Орган власти
ОИВ	Органы исполнительной власти
Оператор СМЭВ	Министерство связи и массовых коммуникаций Российской Федерации (в соответствии с постановлением Правительства РФ N 697 от 08.09.2010).
Органы и организации	Федеральные органы исполнительной власти, государственные внебюджетные фонды, исполнительные органы государственной власти субъектов Российской Федерации, органы местного самоуправления, государственные и муниципальные учреждения, многофункциональных центров, иные органы и организации.
Отправитель сообщения	Информационная система, отправляющая сообщение через СМЭВ
Очередь доставки запросов поставщиков	Очередь СМЭВ хранящая пул запросов для поставщика сведений.
Очередь доставки ответов потребителей	Очередь СМЭВ хранящая ответы на запросы в адрес потребителя.
Получатель сообщения	Информационная система, получающая сообщение из СМЭВ.
Потребитель сведений	Участник взаимодействия, запрашивающий сведения у поставщика сведений.
Поставщик сведений	Участник взаимодействия, поставляющий сведения в ответ на запрос сведений.
Прикладная схема (поставщика)	XML-схема, описывающая состав структурированных сведений, передаваемых в рамках запросов и ответов, в соответствии с требованиями поставщика.
РФ	Российская Федерация
УВ	Участник взаимодействия – орган или организация, участвующая в электронном обмене через СМЭВ.
ФРГУ	Федеральный реестр государственных услуг
ФХ	Файловое хранилище СМЭВ
ЭП	Электронная подпись

ЭП-ОВ	Электронная подпись органа власти
ЭП-СМЭВ	Электронная подпись СМЭВ
ЭП-СП	Электронная подпись для служебного пользования (должностного лица)
ЦНСИ	Центральный модуль ЕСНСИ.
Namespace, пространство имен	Логическая группировка уникальных идентификаторов (имён), подробнее см. http://www.w3.org/TR/2009/REC-xml-names-20091208/
MTOM	Message Transmission Optimization Mechanism, механизм передачи вложений в двоичном формате с сообщениями протокола SOAP как необработанных байтов. См. также http://www.w3.org/TR/soap12-mtom/
Target namespace, пространство имен XML-схемы	Логическая группировка уникальных идентификаторов (имён) элементов и атрибутов в схеме XML-документа, подробнее см. http://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/
Qualified name, полное имя (XML элемента)	Пара, состоящая из префикса в виде <u>пространства имен</u> и <u>локального имени</u> XML-элемента. См. http://www.w3.org/TR/2009/REC-xml-names-20091208/#ns-qualnames
URI	Unified Resource Identifier, уникальный идентификатор ресурса. Может быть либо URL – уникальным локатором ресурса, либо URN – уникальным именем ресурса. См. также http://tools.ietf.org/html/rfc3986

Для описания требований к участникам взаимодействия используются следующие соглашения, выделенные жирным шрифтом:

может – разрешено, но необязательно,

не может – запрещено, **должен** – обязательно.

2. ОСНОВЫ ВЗАИМОДЕЙСТВИЯ

2.1. ОСНОВНЫЕ ПОНЯТИЯ И ПРАВИЛА ОБМЕНА ИНФОРМАЦИЕЙ

2.1.1 Понятия и правила обмена информацией

СМЭВ обеспечивает взаимодействие информационных систем (далее – **ИС**) федеральных органов исполнительной власти, государственных внебюджетных фондов, исполнительных органов государственной власти субъектов Российской Федерации, органов местного самоуправления, государственных и муниципальных учреждений, многофункциональных центров, иных органов и организаций (далее – **органы и организации**), используемых при предоставлении государственных и муниципальных услуг и исполнении государственных и муниципальных функций в электронной форме. Органы и организации выступают в качестве **участников взаимодействия**.

Участники взаимодействия могут запрашивать сведения или предоставлять их другим участникам, в зависимости от этого они являются **потребителями** или **поставщиками** сведений, путем обмена данными между ИС участников с использованием единого электронного сервиса СМЭВ (далее – **единый электронный сервис**). Единый электронный сервис реализован в виде веб-сервиса, предоставляемого СМЭВ. Схемы единого электронного сервиса приведены в п. 2.1.2.

В рамках информационного взаимодействия ИС поставщика и потребителя обмениваются **сообщениями**. ИС, отправляющая сообщение через СМЭВ, является **отправителем сообщения**, а ИС, получающая сообщение из СМЭВ, – **получателем**. Участник взаимодействия, запрашивающий сведения у поставщика сведений именуется **потребителем сведений**. Участник взаимодействия, поставляющий сведения в ответ на запрос сведений именуется – **поставщиком сведений**. В ходе обмена в СМЭВ организуются **очереди доставки запросов поставщиков**, где накапливаются запросы, приходящие в адрес поставщиков сведений и **очереди доставки ответов потребителей**, где накапливаются ответы на запросы в адрес потребителей.

Упрощенно, процесс обмена сообщениями между ИС потребителя и ИС поставщика через СМЭВ включает последовательность следующих шагов (Рисунок 1):

- передача запроса от ИС потребителя в СМЭВ;
- размещение запроса в СМЭВ в очереди запросов поставщика;
- получение запроса ИС поставщика из СМЭВ;
- подготовка поставщиком ответа на запрос;
- передача подготовленного ответа из ИС поставщика в СМЭВ;
- размещение ответа в СМЭВ в очереди ответов потребителя;
- получение ответа ИС потребителя из СМЭВ.

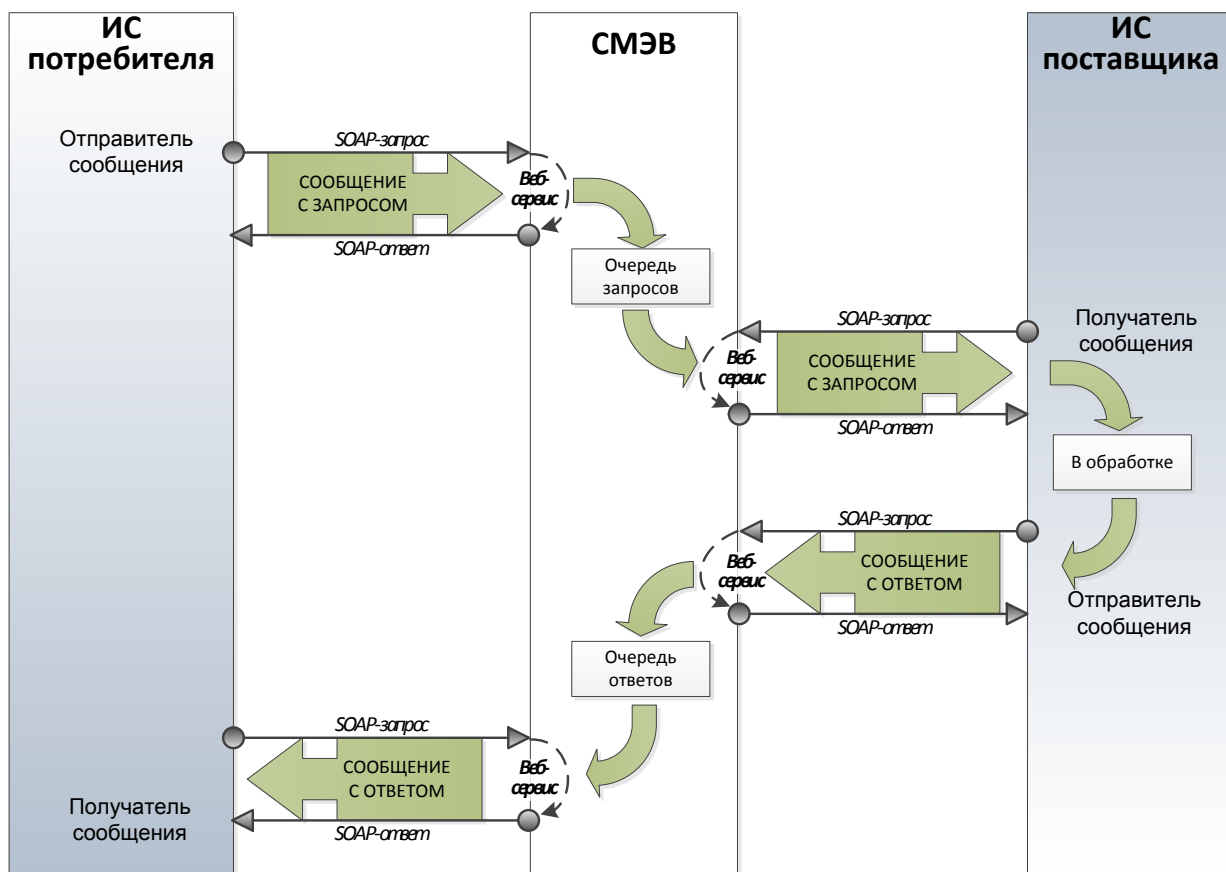






Рисунок 1 – Взаимодействие ИС потребителя и ИС поставщика через СМЭВ

Взаимодействие в СМЭВ осуществляется в асинхронном режиме, в стиле электронной почты. Каждая из операций передачи/получения сообщения (с запросом или ответом) реализуется путем вызова соответствующего метода единого электронного сервиса. Передача сообщений через СМЭВ реализована с использованием механизма очередей.

2.1.2 Схемы единого электронного сервиса

Текущие схемы единого электронного сервиса представлены во вложенных файлах.

 smev-message-exchange-basic-1.2.xsd	 smev-message-exchange-types-1.2.xsd
 smev-message-exchange-faults-1.2.xsd	 smev-message-exchange-service-1.2.wsdl

2.2. КОНЦЕПЦИЯ «ВИДЫ СВЕДЕНИЙ»

2.2.1 Толкование термина

В контексте настоящего документа, под «**видом сведений**» понимается протокол передачи сведений определенного *вида* между информационной системой поставщика и информационной системой потребителя.

Вид сведений представляет собой машиночитаемое описание, включающее в себя структурные и семантические правила, которым должны соответствовать сообщения, передаваемые через СМЭВ. Указанные правила описания вида сведений разделяются на общеобязательные и специфичные.

Общеобязательные правила задают параметры сообщения в целом, и не зависят от специфики передаваемых сведений.

В частности, общеобязательные правила определяют:

- структуру всего сообщения;
- требования к формированию блоков, содержащих электронную подпись;
- требования к формированию элементов, используемых для маршрутизации сообщения;
- и т.д.

Общеобязательные правила устанавливаются оператором СМЭВ, и описаны далее в настоящем документе.

Специфичные правила определяют параметры, которым должно удовлетворять СМЭВ-сообщение при обмене между конкретным поставщиком и потребителем.

Вводимые этими правилами ограничения задают требования к форматам тех блоков СМЭВ-сообщения, в которых содержатся:

- описания допустимых вариантов обращений за сведениями;
- описания вариантов ответов на обращения за сведениями.

Специфичные правила для вида сведений определяются одной из сторон обмена – либо поставщиком сведений, либо их потребителем.

2.2.2 Маршрутизация запросов на основании передаваемых сведений

Вместо концепции точки доступа (endpoint) для адресации запросов используется концепция видов сведений. Поскольку запрос представляет собой XML-документ, вид сведений может быть однозначно определен по [полному имени](#) корневого элемента этого XML-документа. Это возможно в связи с тем, что любая схема XML-документа должна иметь глобально уникальное [пространство имен схемы](#) и внутри одной схемы имена элементов корневого уровня должны быть уникальны. Одной из функций СМЭВ является технический контроль уникальности [пространств имен схем](#). На основании вида сведений СМЭВ может определить, какому поставщику должен быть направлен запрос.

2.2.3 Маршрутизация запросов по коду маршрутизации

Распоряжение Правительства Российской Федерации от 29 июня 2012 г. № 1123-р определяет перечень сведений, находящихся в распоряжении государственных органов субъектов Российской Федерации, органов местного самоуправления, территориальных государственных внебюджетных фондов либо подведомственных государственным органам субъектов Российской Федерации или органам местного самоуправления организаций, участвующих в предоставлении государственных или муниципальных услуг, и необходимых для предоставления государственных услуг федеральными органами исполнительной власти и органами государственных внебюджетных фондов Российской Федерации.

Для выполнения данного Распоряжения Правительства в СМЭВ поддерживается возможность маршрутизации запросов по коду маршрутизации. СМЭВ поддерживает использование следующих кодов маршрутизации:

- код ОКТМО;
- произвольный код маршрутизации.

Код маршрутизации в сообщении заполняется и передается внутри передаваемых сведений в соответствии с требованиями вида сведений с табличной маршрутизацией. Код маршрутизации не должен превышать 200 символов.

СМЭВ в сообщении определяет код маршрутизации по xpath-выражению. Xpath-выражение передается для регистрации в СМЭВ в заявке на регистрацию вида сведений с табличной маршрутизацией.

СМЭВ содержит информацию о том, необходимо ли маршрутизировать запросы данного типа в регионы, и если необходимо, то СМЭВ хранит xpath-выражение, указывающее, где именно в структуре передаваемого запроса вида сведения содержится информация о коде маршрутизации.

Пример схемы вида сведений с кодом маршрутизации «RegionCode»:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="urn://geo/tabl/1.0.0"
  targetNamespace="urn://geo/tabl/1.0.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
>
  <xs:element name="TestRegionalRoutingRequest">
    <xs:annotation>
      <xs:documentation>Элемент запроса</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="RegionCode" type="xs:string">
          <xs:annotation>
            <xs:documentation>Код региона</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="RequestContent" type="xs:string">
          <xs:annotation>
```

```

        <xs:documentation>Текст запроса</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="TestRegionalRoutingResponse">
  <xs:annotation>
    <xs:documentation>Элемент ответа</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ResponseContent" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

Пример запроса по приведенной схеме вида сведений:

```

<?xml version="1.0" encoding="UTF-8"?>
<ns1:TestRegionalRoutingRequest xmlns:ns1="urn://geo/tabl/1.0.0">
  <ns1:RegionCode>71000000</ns1:RegionCode>
  <ns1:RequestContent>Запрос</ns1:RequestContent>
</ns1:TestRegionalRoutingRequest>

```

Пример xpath-выражения, указывающего место в структуре передаваемого запроса «TestRegionalRoutingRequest» где содержится код маршрутизации «RegionCode»: /*[local-name()='TestRegionalRoutingRequest']/*[local-name()='RegionCode']/text()

2.2.4 Требования к описанию форматов сведений

Форматы сведений разрабатываются поставщиком с использованием языка описания схем данных XML Schema Definition (XSD) и должны соответствовать следующим правилам:

- Для каждого вида сведений один из элементов, описанных на корневом уровне схемы, должен представлять собой "корневой элемент запроса".
- Для каждого вида сведений, кроме передаваемых с использованием широко-вещательных рассылок, один из элементов, описанных на корневом уровне схемы, должен представлять собой "корневой элемент ответа".
- Если для определенного вида сведений возможны различные варианты ответов – их необходимо описать с использованием одного структурированного типа (конструкция xs:choice).

Для каждого вида сведений корневой элемент запроса, и корневой элемент ответа должны быть описаны в одной схеме (иметь одно и то же [пространство имен схемы](#)). При этом схема может быть разбита на несколько XML-документов (конструкция xs:include), а также ссылаться на другие XML-схемы (конструкция xs:import).

Поскольку СМЭВ перед доставкой запроса контролирует соответствие XML-документов зарегистрированным схемам, отказы в обслуживании со стороны поставщиков по причине несоответствия полученного XML-документа схеме запрещены.

XML схемы видов сведений, регистрируемые в СМЭВ, должны удовлетворять требованиям документа «Требования к XML-схемам, регистрируемым в СМЭВ».

2.2.5 Обмен справочными данными в составе сообщений по видам сведений в СМЭВ

Разрабатываемые схемы видов сведений в полях запросов и ответов могут содержать справочные значения (конечные ограниченные наборы не изменяемых значений, либо значений с циклом изменений не чаще 1-го раза в сутки). Для ведения подобных справочных значений необходимо использовать полный цикл создания и ведения справочников, который предоставляет ЦНСИ – как посредством пользовательского web-интерфейса, так и через API (web-сервис, файлы, либо обмен через СМЭВ3.x). Т.о. данные справочные поля схем видов сведений в обязательном порядке должны использовать типы, определенные в схемах справочников ЦНСИ. Так же в схемах подобных видов сведений должны быть использованы Namespace xsd-схем справочников ЦНСИ. Получить xsd-схемы справочников необходимо посредством пользовательского интерфейса ЦНСИ после их публикации в системе.

Полученные xsd-схемы справочников включаются в основную схему вида сведений с помощью инструкции import. Помимо этого namespace, идентификатор ЦНСИ и название каждого используемого справочника обязательно должны быть описаны в руководстве пользователя вида сведений.

Включение схем справочников в схему видов сведений позволяет СМЭВ проверять правильность заполнения справочных полей запросов и ответов. Для передачи схемы справочника в СМЭВ необходимо для данного справочника в интерфейсе ЦНСИ установить метку об его использовании при обмене в СМЭВ. Справочники с такими метками будут автоматически опубликованы в СМЭВ.

Строки, публикуемых в ЦНСИ справочников, должны содержать в своем составе уникальный идентификатор строки справочника, Справочный код - единственный атрибут (первичный ключ или код), значение которого однозначно идентифицирует конкретный экземпляр из всего набора возможных экземпляров справочных данных (справочников). Выгрузке и обмену подлежат именно Справочные коды, а не сами значения строки справочника, что позволяет как поставщику, так и потребителю сведений идентифицировать значение строки справочника при возможных вариациях в формате ее содержимого.

Данное требование обязательно к выполнению для справочников размером не более 2000 записей.

Описание порядка формирования схем с включением справочников приведено в Приложении 7.

2.2.6 Версионность форматов сведений

При необходимости внесения изменений в формат сведений, в СМЭВ необходимо зарегистрировать новую версию XML-схемы. Чтобы обеспечить корректную маршрутизацию сообщений, соответствующих устаревшим версиям форматов сведений, в СМЭВ сохраняется полная история всех изменений, включая все предыдущие версии

XML-схем. Для каждой новой версии формата сведений XML-схема должна иметь отличающийся от предыдущих версий форматов Target namespace.

2.2.7 Структура вида сведений в СМЭВ

В СМЭВ вид сведений представляет собой одну или несколько версий форматов сведений. Каждая версия формата состоит из одной или нескольких XML-схем: одна описывает сведения, передаваемые в запросе и ответе, а остальные (при необходимости) импортируются в неё оператором `xs:import`.

2.3. ТИПЫ СООБЩЕНИЙ

Сообщения, передаваемые в СМЭВ, типизируются на **запрос и ответ**. С точки зрения СМЭВ все сообщения не отличаются и обрабатываются одинаковым образом.

2.3.1 Сообщения типа «Запрос»

К сообщениям типа «Запрос» (далее – **запрос**) относятся сообщения, исходящие от инициатора взаимодействия: межведомственные запросы, запросы на оказание государственных или муниципальных услуг, широковещательные (в т.ч. реестровые) рассылки.

Сообщения типа «запрос» проходят контроль корректности данных в два этапа – **синхронная** и **асинхронная** (необязательная) проверка.

Первый этап – синхронная проверка. После выполнения всех синхронных проверок, запрос помещается в очередь на асинхронную проверку. Если проверка прошла успешно, то в ответе возвращается сообщение об успешной проверке, при наличии ошибок метод `{urn://x-artefacts-smev-gov-ru/services/message-exchange/1.1:SendRequest}` возвращает `fault`.

Асинхронная проверка осуществляется после успешного завершения этапа синхронной проверки. При передаче сообщения в асинхронную проверку СМЭВ в ответ на запрос возвращает в синхронном режиме сообщение, где в блоке `MessageMetadata` содержится следующий тег: `<ns2:Status>requestIsQueued</ns2:Status>`.

Если какая-либо асинхронная проверка показала ошибку, СМЭВ помещает во входящую очередь ответов отправителя запроса сообщение об ошибке. Сообщение об ошибке будет получено отправителем запроса при очередном запросе `GetResponse`.

Отличить ответы поставщика данных от сообщений СМЭВ об ошибках асинхронного контроля можно по содержимому элемента `{urn://x-artefacts-smev-gov-ru/services/message-exchange/types/1.1:GetResponseResponse}`: если его дочерний элемент `SenderProvidedResponseData` содержит элемент `MessagePrimaryContent`, то это ответ Поставщика, а если элемент `AsyncProcessingStatus` – ответ об ошибке асинхронной обработки СМЭВ.

Допустимо направление несколько запросов в рамках одной бизнес-транзакции (запрос сведений, дозапрос сведений, сообщение статуса Потребителя о получении сведений и т.д.). В случае множественных запросов идентификатор первичного запроса помещается в поле `ReferenceMessageID` всех последующих запросов в рамках бизнес-транзакции.

2.3.2 Сообщения типа «Ответ»

Сообщения типа «Ответ» (далее – **ответ**) могут содержать либо запрошенные данные, либо мотивированный отказ в приеме запроса к исполнению. Запросы, представляющие собой широковещательные рассылки, не требуют ответов.

Сообщения типа «Ответ» проходят контроль корректности данных аналогично сообщениями типа «Запрос».

2.3.3 Широковещательные рассылки

Встречаются ситуации, при которых система-отправитель сообщения-запроса должна довести до других участников взаимодействия вид сведений по схеме «опубликование», то есть конечный перечень получателей данного вида сведений поставщику заранее не известен или не имеет для него значения. Потребители инициативно подписываются на получение данного вида сведений. Такое взаимодействие называется широковещательной рассылкой.

Широковещательная рассылка – предполагает, что сообщение-запрос передается ИС-подписчиком в опубликованном ИС-поставщиком виде. Ответ на данное сообщение-запрос не поддерживается (это контролируется СМЭВ).

2.3.4 Реестровый вид сведений

Информационный обмен, при котором в каждом СМЭВ-сообщении содержится единичный экземпляр сведений указанной категории, предъявляет к поставщику высокие требования к производительности. В этом случае следует использовать реестровый вид сведений. Характерной особенностью данной категории видов сведений является предопределенная структура бизнес-блока сообщения.

Бизнес-блок сообщения-запроса реестрового вида сведений состоит из следующих логических частей:

Заголовок – содержит элементы, значения которых применимы ко всему реестру.

Реестр – состоит из записей. *Запись* – один передаваемый экземпляр сведений. Все записи внутри одного реестра имеют одинаковую структуру.

Запись может включать в себя следующие элементы:

- Идентификатор записи реестра. В одном реестре не должно быть двух и более записей, содержащих одно и то же значение данного элемента.
- Элемент описания вложений к записи реестра – содержит элементы, описывающие все вложения, относящиеся к данной записи. Подробнее см. раздел 3.8

Пример бизнес-блока реестрового вида сведений:

```
<?xml version='1.0' encoding='UTF-8' ?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="urn://x-artefacts-smev-gov-ru/mr3000/example/1.0.0"
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="urn://x-artefacts-smev-gov-ru/mr3000/example/1.0.0">
```

```

<!--
В сообщении-запросе необходимо передать реестр документов.
Каждый документ состоит из заголовка и нескольких вложений.
Атрибутивный состав заголовка одинаков для всех документов.
Форматы вложений, прилагаемых к документам различны, но заранее
предопределены.
-->

<xsd:element name="RegistryExampleRequest">
  <xsd:complexType>
    <xsd:sequence>
      <!-- В запросе может быть реестровый блок, для передачи
однотипных сведений с вложениями и ЭП-->
      <xsd:element name="RegistryBlock"
type="tns:_RegistryBlockType" minOccurs="0" />
      <!-- Объявление блока вложений ко всему сообщению-->
      <xsd:element name="AttachmentsBlock"
type="tns:_AttachmentsBlockType" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="_RegistryBlockType">
  <xsd:sequence>
    <xsd:element name="RegistryRecord"
type="tns:_RegistryRecordType" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="_RegistryRecordType">
  <xsd:sequence>
    <!-- Объявление обязательного идентификатора реестровой записи-->
    <xsd:element name="RegistryRecordID" type="xsd:string" />
    <!-- Объявление блока вложений для реестровой записи-->
    <xsd:element name="AttachmentsBlock"
type="tns:_AttachmentsBlockType" minOccurs="0" />
    <!-- Описание элементов, описывающих реквизиты заголовков
передаваемых документов - отражается любая ведомственная специфика-->
    <xsd:element name="DocumentNumber" type="xsd:string" />
    <xsd:element name="DocumentDate" type="xsd:dateTime" />
    <xsd:element name="DocumentOriginator" type="xsd:string" />
    <xsd:element name="DocumentAttachmentsAmount" type="xsd:int" />
  </xsd:sequence>
</xsd:complexType>

<!-- Блок описания вложений внутри бизнес-блока сообщения СМЭВ3
Может быть включен также в реестровую запись. Описание требований к
представлению блока описания вложений приведено в Разделе 3.8
-->
<xsd:complexType name="_AttachmentsBlockType">
  <xsd:sequence>
    <xsd:element name="AttachmentDescription"
type="tns:_AttachmentDescriptionType" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="_AttachmentDescriptionType">
  <xsd:sequence>
    <!-- Указание способа передачи вложения: через ФХ или методом
МТОМ-передачи -->
    <xsd:choice>
      <!-- Ссылка на папку ФХ, в котором содержится передаваемое
вложение-->

```

```

        <xsd:element name="AttachmentFSLink" type="xsd:string"/>
        <!-- Признак передачи вложения методом MTOM -->
        <xsd:element name="IsMTOMAttachmentContent"
type="xsd:boolean"/>
    </xsd:choice>

    <!-- Описание возможных форматов вложения -->
    <xsd:choice>
        <xsd:element name="AttachmentFormat1"
type="tns:_StructuredAttachmentFormatType1" />
        <xsd:element name="AttachmentFormat2"
type="tns:_StructuredAttachmentFormatType2" />
    </xsd:choice>

    <!-- Блок передачи файла отсоединенной электронной подписи
вложения -->
    <xsd:element name="AttachmentSignatureFSLink" type="xsd:string"
minOccurs="0" />

</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="_StructuredAttachmentFormatType1">
    <xsd:sequence>
        <!-- Обязательное и явное указание: является ли формат
структурированным или не является -->
        <xsd:element name="IsUnstructuredFormat" type="xsd:boolean" />
        <!-- Обязательное и явное указание: будет ли содержимое вложения
упаковываться Zip-алгоритмом -->
        <xsd:element name="IsZippedPacket" type="xsd:boolean" />
        <!-- Если формат структурированный, то описываем его структуру --
>
        <xsd:element name="DocumentRecordNumber" type="xsd:int" />
        <xsd:element name="DebtorName" type="xsd:string" />
        <xsd:element name="DebtorIdDoc" type="xsd:string" />
        <xsd:element name="PenaltyAmount" type="xsd:double" />
        <xsd:element name="ExecutionDate" type="xsd:date" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="_StructuredAttachmentFormatType2">
    <xsd:sequence>
        <!-- Явно указываем является ли формат структурированным или нет
-->
        <xsd:element name="IsUnstructuredFormat" type="xsd:boolean" />
        <!-- Обязательное и явное указание: будет ли содержимое вложения
упаковываться Zip-алгоритмом -->
        <xsd:element name="IsZippedPacket" type="xsd:boolean" />
        <!-- Если формат структурированный, то описываем его структуру --
>
        <xsd:element name="DocumentRecordNumber" type="xsd:int" />
        <xsd:element name="FilialCode" type="xsd:string" />
        <xsd:element name="ProcessingStatus" type="xsd:double" />
    </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```

2.3.5 Приоритетная доставка

В СМЭВ поддерживается два уровня приоритета для запросов: обычные и приоритетные. При регистрации информационной системы в СМЭВ ей может быть присвоен статус «Особо важная» (VIP). В этом случае получателю сообщений СМЭВ первыми будет отдавать запросы от ИС с признаком VIP.

Все ответы доставляются с одинаковым приоритетом. Приоритеты доставки также не применяются к широковещательным рассылкам.

СМЭВ не предоставляет других возможностей влиять на приоритетность отправляемых сообщений.

2.3.6 Сообщения об отказах в ответе на уровне схемы СМЭВ

Поставщик сведений может отказать в предоставлении запрашиваемых сведений. Все возможные отказы в предоставлении сведений делятся на четыре типа:

1. Отказ в предоставлении сведений. Отсутствуют права на получение информации (например, в случае, если поставщик проверяет ЭП-СП).
2. Отказ в предоставлении сведений. Невозможно определить объект запроса информации. Например, если не хватает сведений в запросе и нужен дозапрос информации.
3. Уведомление об отсутствии сведений.
4. Ошибка при предоставлении сведений.

Данные сообщения об отказах выносятся на уровень схемы СМЭВ и не включаются в схемы видов сведений. **Включение сообщений об отказах непосредственно в схему видов сведений – запрещено.**

Сообщение об ошибке помещается в элемент RequestRejected и может принимать четыре значения (элемент - RejectionReasonCode): ACCESS_DENIED, UNKNOWN_REQUEST_DESCRIPTION, NO_DATA, FAILURE, соответствующие описанным выше типам отказов.

Также предусмотрено поле для текстового комментария к отказу (элемент - RejectionReasonDescription).

2.3.7 Возврат статусов запросов на уровне схемы СМЭВ

Поставщик сведений может в ответ на запрос возвращать неограниченное количество статусных сообщений.

Данные сообщения выносятся на уровень схемы СМЭВ и не включаются в схемы видов сведений. **Включение сообщений о статусах непосредственно в схему видов сведений – запрещено.**

Сообщение о статусе помещается в элемент RequestStatus. В элемент StatusCode помещается код статуса, значение которого описывается в паспорте Вида сведения. Статус может сопровождаться неограниченным количеством параметров (элемент StatusParameter), которые описываются парами «ключ»-«значение» (Key-Value). В поле StatusDescription можно поместить расширенное описание статуса.

2.3.8 Указание мнемоники сервера отправки сообщения

В случае, если информационная система отправителя запросов представляет собой многосерверную (многонодную) архитектуру, в которой отправка запросов осуществляется с разных серверов, а обработка ответов должна происходить на сервере-отправителе, возникает проблема маршрутизации ответа на запрос на сервер-отправитель.

Для решения данной проблемы в схеме запросов СМЭВ введен специальный элемент указания мнемоники сервера-отправителя – «NodeID». При получении запроса с данным идентификатором СМЭВ3.x создает для данной информационной системы и данного сервера отдельную очередь доставки ответов. Чтобы получить сообщение из данной очереди в запросе на чтение входящей очереди необходимо указать мнемонику сервера-получателя (элемент «NodeID»).

2.4. ЖИЗНЕННЫЙ ЦИКЛ СООБЩЕНИЙ

2.4.1 Жизненный цикл сообщения типа «Запрос»

Жизненный цикл сообщения типа «Запрос» в СМЭВ представлен на рисунке 2.

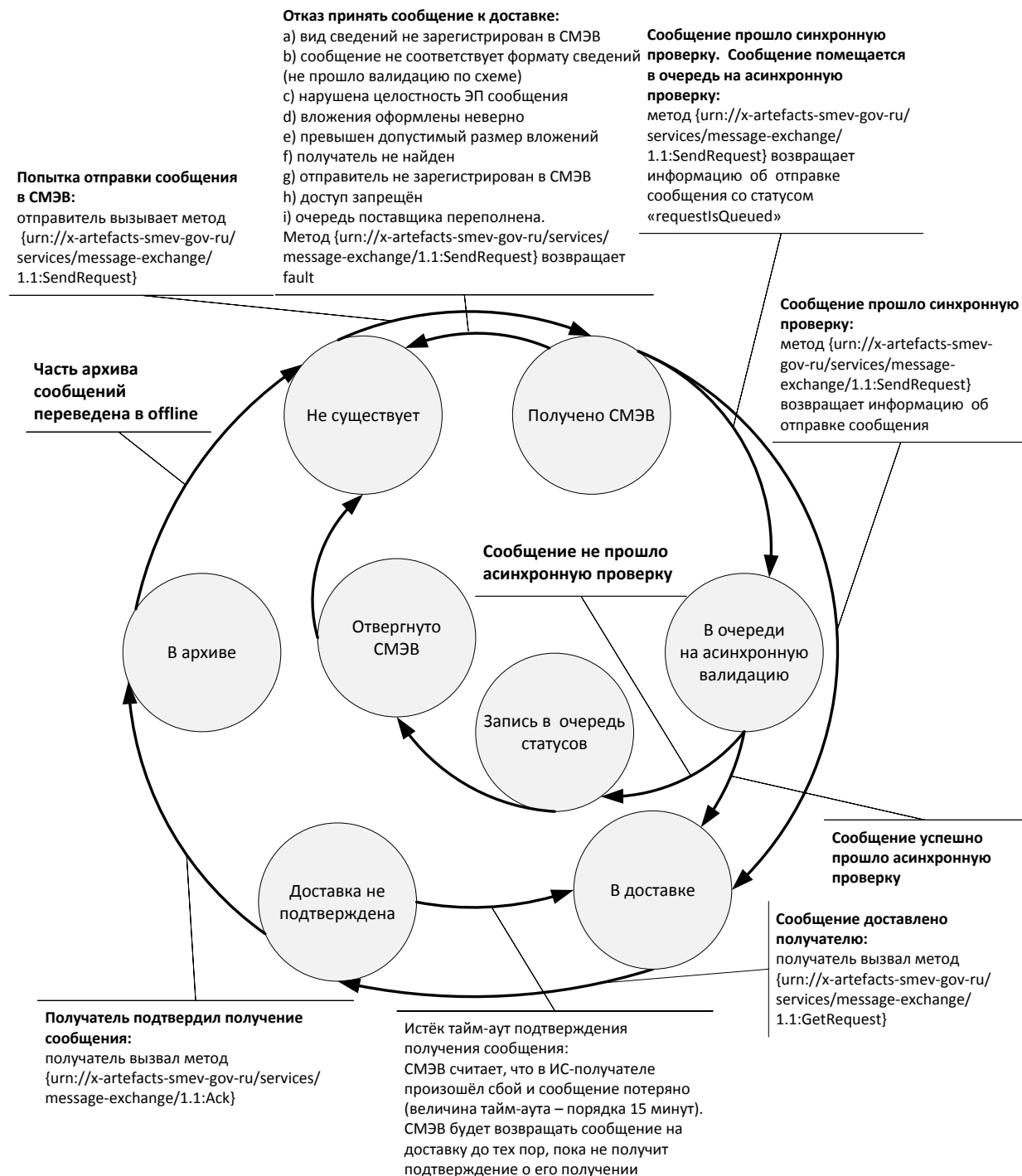


Рисунок 2 – Жизненный цикл сообщений типа «Запрос»

2.4.2 Жизненный цикл сообщения типа «Ответ»

Жизненный цикл сообщения типа «Ответ» в СМЭВ представлен на рисунке 3.

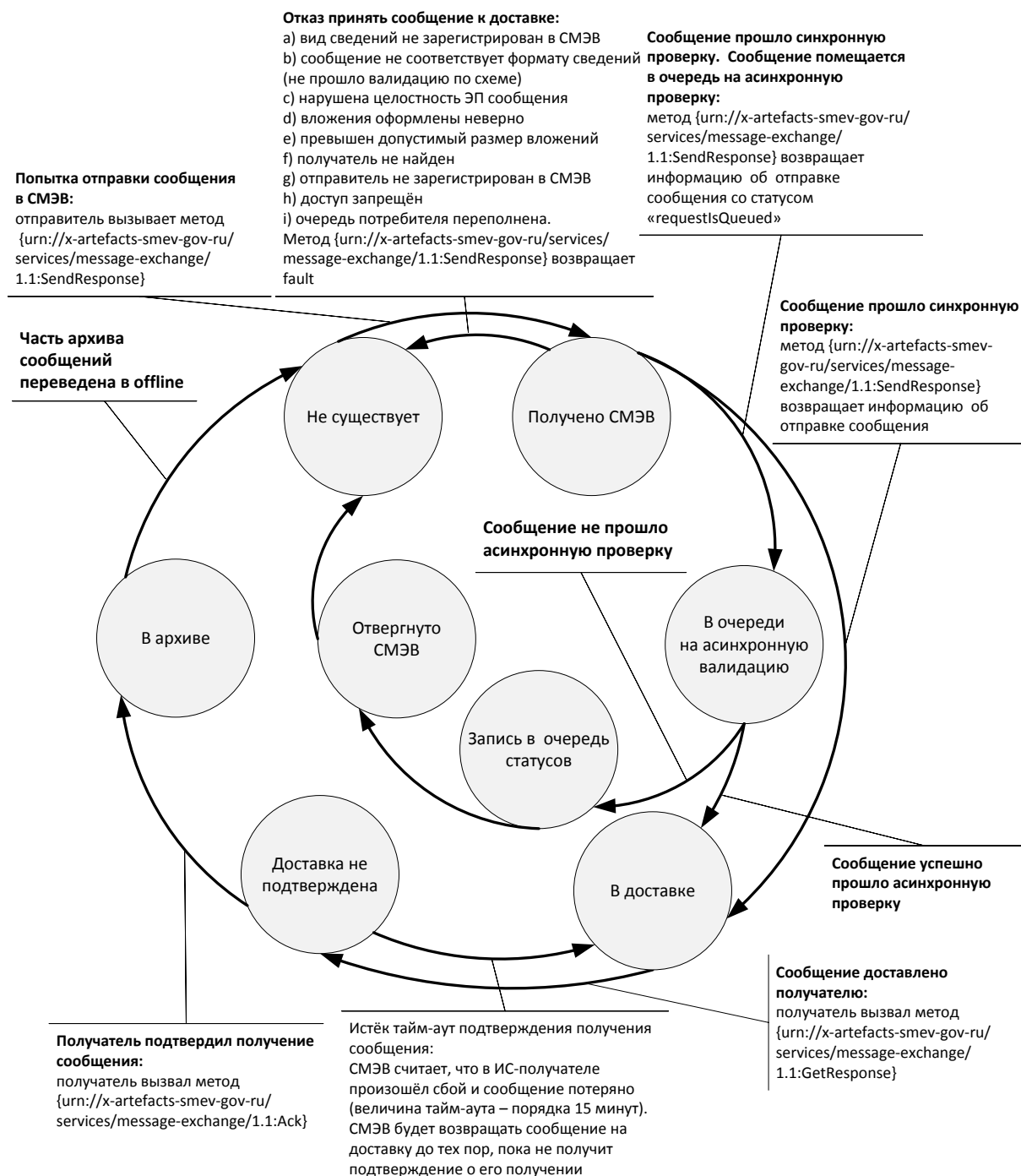


Рисунок 3 – Жизненный цикл сообщений типа «Ответ»

2.4.3 Жизненный цикл бизнес-взаимодействия

СМЭВ осуществляет бизнес-взаимодействия, состоящие из пересылки сообщения типа «Запрос» и, возможно, пересылки сообщений типа «Ответ». СМЭВ не предоставляет участникам взаимодействия средств для получения информации о статусах. Основным механизмом доставки статусных сообщений об ошибках асинхронной обработки становятся системные ответы СМЭВ, получаемые методом `getResponse`.

2.5. ОРГАНИЗАЦИЯ ОЧЕРЕДЕЙ

Все очереди сообщений в СМЭВ являются входящими и закреплены за получателями сообщений. Все сообщения, которые отправляют информационные системы, сразу попадают в очереди входящих сообщений, закреплённые за получателями (информационными системами). Очереди входящих сообщений делятся на очереди входящих запросов и очереди входящих ответов на запросы.

В очередь входящих запросов попадают запросы по всем видам сведений (далее – виды запросов), а в очередь входящих ответов на запросы – ответы по всем видам сведений (далее – виды ответов).

Возможно два сценария выборки сообщения из очереди входящих сообщений: с фильтрацией и без фильтрации по виду сведений.

При приёме без фильтрации по виду сведений (Рисунок 4), получатель выберет первое сообщение, имеющееся в очереди входящих сообщений, независимо от того, к какому виду сведений оно относится. Для этого необходимо вызвать метод `getRequest` (или `getResponse`) единого электронного сервиса СМЭВ, без указания параметров `MessageTypeSelector/NamespaceURI` и `MessageTypeSelector/RootElementLocalName`.

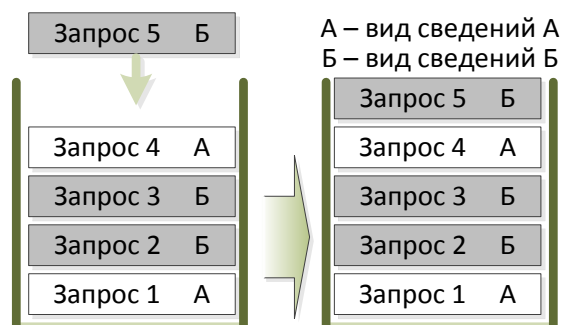


Рисунок 4 – Очередной запрос ИС потребителя помещается в очередь входящих запросов (ИС поставщика работает в режиме общих очередей)

При приёме сообщения с фильтрацией по виду сведений (Рисунок 5) СМЭВ будет искать сообщения в очереди входящих сообщений, относящиеся к запрошенному виду сведений, и вернёт первое из них. Если сообщений запрошенного вида сведений в очереди входящих сообщений нет, СМЭВ не вернёт ничего, даже если в очереди входящих сообщений есть сообщения других видов сведений. Чтобы использовать этот сценарий, необходимо при вызове метода `getRequest` (`getResponse`) заполнить параметры `MessageTypeSelector/NamespaceURI` и `MessageTypeSelector/RootElementLocalName`.

Правила заполнения этих параметров описаны в разделе 2.5.1 «Получение сообщения с фильтрацией по виду сведений».

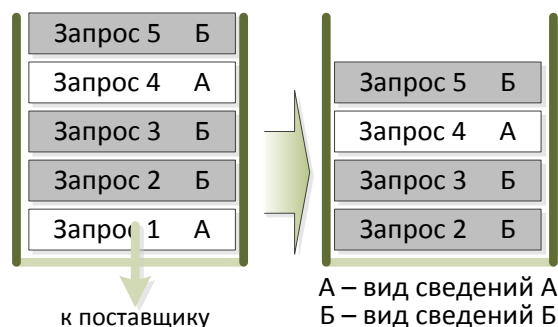


Рисунок 5 – ИС поставщика «забирает» из очереди входящих запросов очередной запрос без указания вида сведений, в режиме общих очередей

Если поставщиком был указан вид сведений «Б», в этом случае (Рисунок 6) ИС поставщика получит не запрос № 1, который находится в начале очереди входящих сообщений, а запрос № 2, так как среди запросов сведений вида «Б» в очереди первым был размещен запрос № 2.

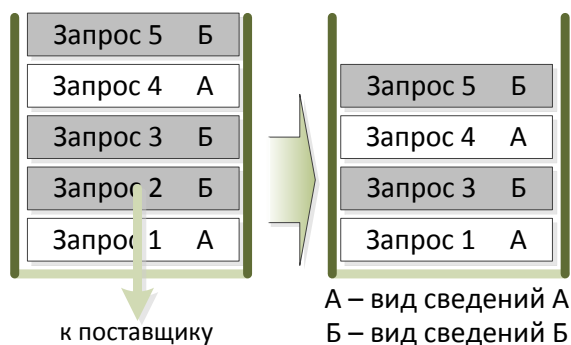


Рисунок 6 – ИС поставщика «забирает» из очереди входящих запросов очередной запрос с указанием вида сведений «Б», в режиме общих очередей

2.5.1 Получение сообщения с фильтрацией по виду сведений

В соответствии с вышеизложенным, для получения сообщения с фильтрацией по виду сведений, нужно в параметрах запроса `getRequest` (`getResponse`) заполнить элементы данных `MessageTypeSelector/NamespaceURI` и `MessageTypeSelector/RootElementLocalName`.

В соответствии с разделом 2.2 «Концепция «Виды сведений» описание формата запроса и формата ответа для вида сведений представляет собой объявление XML-элемента. Полное имя (qualified name) этого элемента используется участниками взаимодействия для задания вида сведений в методах `getRequest` и `getResponse`. В качестве аргумента `MessageTypeSelector/NamespaceURI` передаётся target namespace схемы, в которой описан элемент, а в качестве аргумента `MessageTypeSelector/RootElementLocalName` – имя (local name) элемента.

Если описание формата вида сведений имеет несколько версий, то можно указать qualified name элемента-запроса из любой версии описания. При этом будут выбираться все сообщения, соответствующие данному виду сведений, независимо от того, какой версии формата они соответствуют.

В методе `getResponse` для задания вида сведений можно использовать как qualified name элемента – запроса, так и qualified name элемента – ответа. Это же относится и к методу `getRequest`.

2.5.2 Подтверждение приёма сообщения

Особенностью организации очередей входящих сообщений в СМЭВ является необходимость подтверждения ИС участника взаимодействия получения сообщения из СМЭВ. Если в течение 15 минут этого не происходит, то сообщение считается недоставленным и возвращается в очередь входящих сообщений. Получение в ИС поставщика очередного запроса (запрос № 1) и помещение этого запроса в очередь на подтверждение получения запроса показано на рисунке (Рисунок 7).

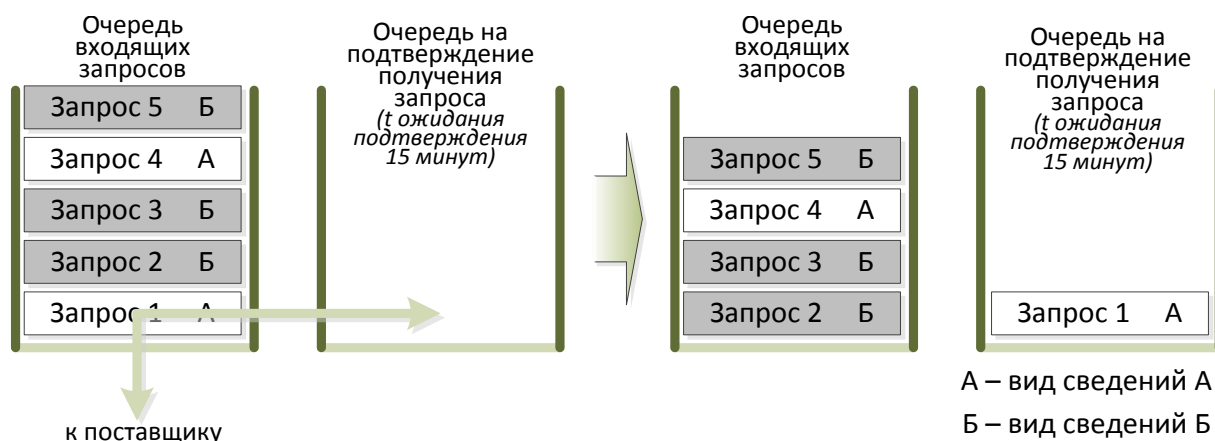


Рисунок 7 – Получение в ИС поставщика очередного запроса и помещение этого запроса в очередь на подтверждение получения запроса

Состояние очередей, изображенное справа (Рисунок 7), соответствует ситуации, когда ИС поставщика получила запрос № 1 (в очереди входящих запросов этого запроса уже нет), но еще не подтвердила его получение (поэтому запрос № 1 находится в очереди на подтверждение получения запроса). Так как время, отведенное на подтверждение получения, еще не истекло (15 минут с момента получения запроса поставщиком сервиса), то никаких действий с запросом № 1, находящимся в очереди на подтверждение получения запроса, не предпринимается.

Если в это же время, то есть до истечения 15 минут с момента получения поставщиком запроса № 1, ИС поставщика обратится в СМЭВ за получением следующего очередного запроса (теперь это будет запрос № 2), то ИС поставщика получит этот запрос № 2, а в очередь на подтверждение получения запроса будет помещен еще один запрос – запрос № 2 (Рисунок 8).

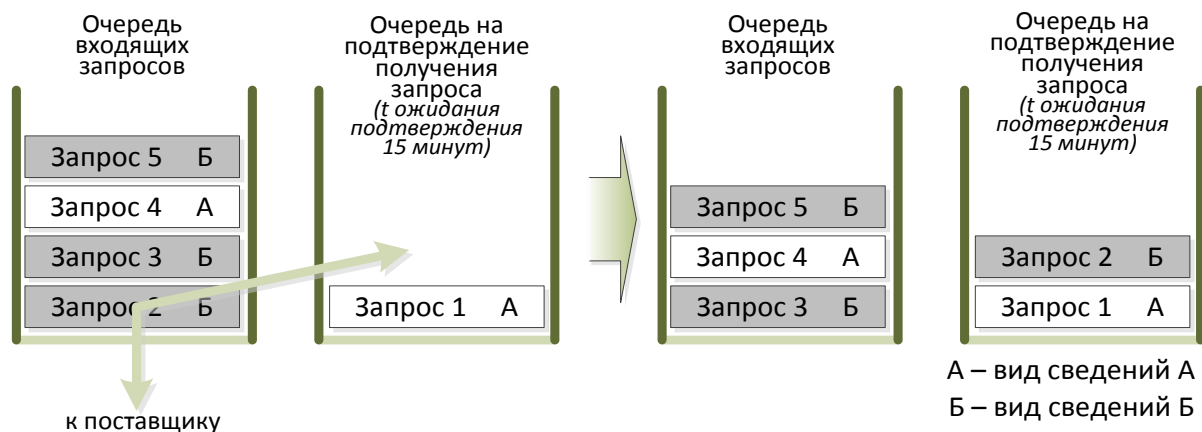


Рисунок 8 – Получение ИС поставщика следующего очередного запроса и помещение этого запроса в очередь на подтверждение получения запроса

Удаление запросов из очереди на подтверждение получения запроса может происходить в двух случаях: если ИС поставщика прислала подтверждение получения запроса или истекло время ожидания подтверждения. На следующем рисунке (Рисунок 9) приведена ситуация, когда ИС поставщика прислала в СМЭВ подтверждение получения запроса № 2, а по запросу № 1 истекло время ожидания подтверждения.

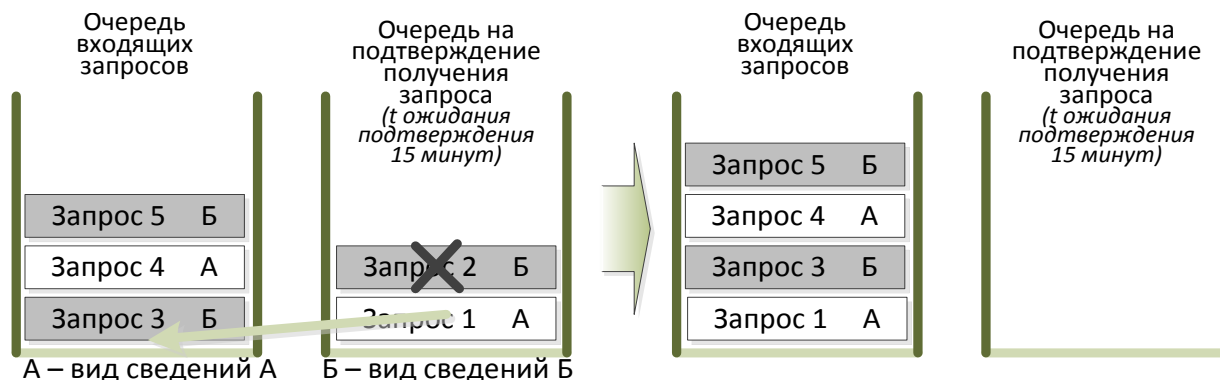


Рисунок 9 – ИС поставщика сервиса прислала подтверждение получения запроса № 2, а по запросу № 1 истекло время ожидания подтверждения

В результате, из очереди на подтверждение получения запроса будет удален запрос № 2, а запрос № 1 будет возвращен в начало очереди запросов.

2.6. ОРГАНИЗАЦИЯ ОЧЕРЕДЕЙ СТАТУСОВ

2.6.1 Получение уведомления из очереди статусов

Очереди статусов в СМЭВ закреплены за отправителями сообщений. В очередь статусов попадают статусные сообщения. Статусные сообщения – это сообщения, включающие сведения об ошибках асинхронной обработки сообщения и push-уведомления о наличии сообщений. При использовании схем версии 1.1* основным механизмом доставки статусных сообщений об ошибках асинхронной обработки становятся системные ответы СМЭВ, получаемые методом `getResponse`. Очередь статусов в этом случае становится источником вспомогательной информации, например для получения сообщений об ошибках push-нотификаций. При получении статусного сообщения из очереди статусов ИС отправителя, получатель выберет первое статусное

сообщение, имеющееся в очереди статусов данной ИС отправителя. Для получения статусного сообщения из очереди статусов ИС отправителя необходимо вызвать метод getStatus. Перечень уведомлений, которые могут содержаться в статусных сообщениях очереди статусов ИС отправителя сообщения, приведены в таблице 1.

Таблица 1 – Перечень возможных уведомлений, которые могут содержаться в статусных сообщениях очереди статусов ИС отправителя сообщения

№	Уведомление	Причины возникновения
1	Ошибка проверки ftp файлов вложения. Файлы повреждены либо данные о файлах, переданные в сообщении, некорректные.	<ul style="list-style-type: none"> – Не равен суммарный размер файлов вложения сообщения, находящегося в области долговременного хранения ФХ, суммарному размеру файлов вложения сообщения, находившегося в директории для записи ФХ. – Не равны хэши файлов вложения сообщения, находящихся в области долговременного хранения ФХ, хэшам файлов вложения сообщения, находившихся в директории для записи ФХ.
2	Сертификат ЭП-ОВ не действительный. Верификация в ГУЦ не пройдена.	ИС ГУЦ в ответ на запрос вернул результат о том, что сертификат ЭП-ОВ не действительный.
3	Ошибка асинхронного процессинга СМЭВ. Данные сообщения некорректные либо отсутствуют.	<ul style="list-style-type: none"> – Некорректные данные (например, СМЭВ не может определить маршрут сообщения) о сообщении в БД сообщений, находящихся в очереди асинхронной обработки. – Отсутствует обратный адрес для сообщения, находящегося в очереди асинхронных процессов. – Отсутствует запись о сообщении в БД сообщений, находящихся в очереди асинхронной обработки. – В записи о сообщении в БД сообщений, находящихся в очереди асинхронной обработки, присутствуют противоречивые данные.
4	Push-уведомления о наличии сообщений.	Если СМЭВ вследствие проблем на стороне информационной системы не может осуществить отправку сервису приема push-уведомлений push-нотификаций, то эти push-нотификации СМЭВ помещает в статусную очередь

№	Уведомление	Причины возникновения
		информационной системы.

В случае отсутствия уведомлений в очереди статусов получатель получит пустое уведомление.

2.6.2 Структура сообщения с запросом статусного сообщения из очереди статусов

Структура сообщения, соответствующая передаче в СМЭВ запроса от ИС отправителя на получение статусного сообщения из очереди статусов, приведена на рисунке ниже (Рисунок 10).

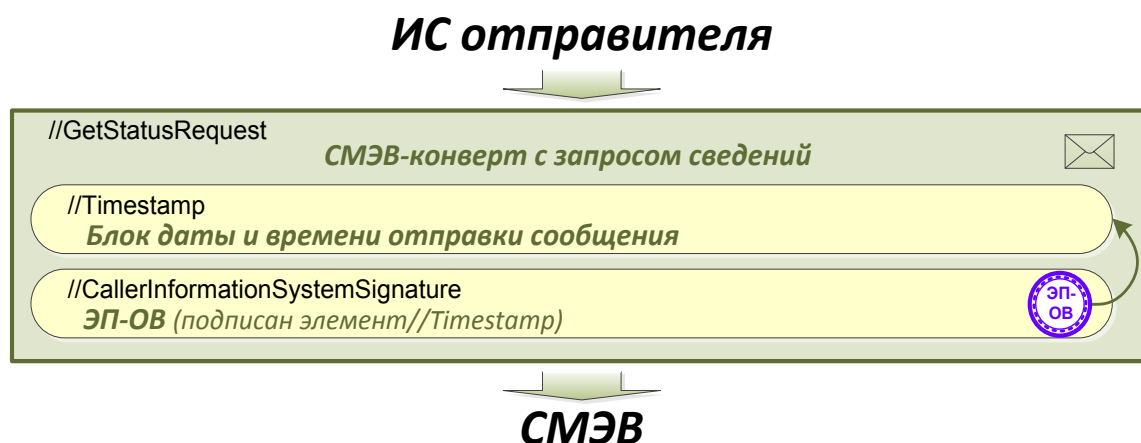


Рисунок 10 – Структура сообщения с запросом статусного сообщения, которое ИС отправителя сообщения передает в СМЭВ

СМЭВ-конверт с запросом сведений (//GetStatusRequest), направляемый ИС отправителя в СМЭВ для получения статусного сообщения из очереди статусов, включает следующие элементы:

- **блок даты и времени отправки сообщения** (//Timestamp), который включает сведения о дате и времени отправки сообщения для получения статусного сообщения из очереди статусов;
- **электронная подпись органа власти** (ЭП-ОВ) (//CallerInformationSystemSignature).

2.6.3 Структура статусного сообщения из очереди статусов

Структура статусного сообщения, соответствующая передаче из очереди статусов СМЭВ уведомления ИС отправителя, приведена на рисунке ниже (Рисунок 11).

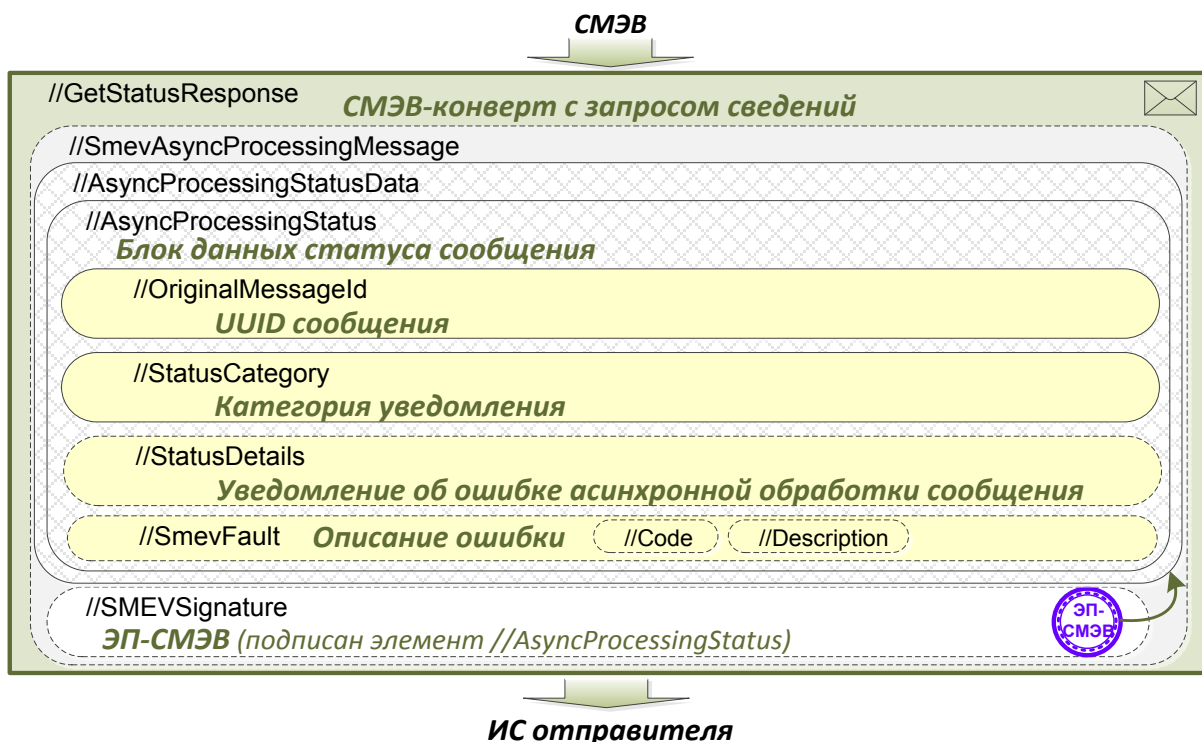


Рисунок 11– Структура статусного сообщения, которое ИС отправителя сообщения получает из СМЭВ

СМЭВ-конверт со сведениями (//GetStatusResponse), получаемый ИС отправителя из СМЭВ, включающий уведомление из очереди статусов, содержит следующие элементы:

- **блок данных СМЭВ-конверта** //AsyncProcessingStatusData, содержащий в себе блок статуса сообщения //AsyncProcessingStatus и заполняемый СМЭВ;
- **электронная подпись СМЭВ** (далее - ЭП-СМЭВ) (//SMEVSignature), заполняемый СМЭВ.

Блок //AsyncProcessingStatus содержит элементы:

- идентификатор сообщения (//OriginalMessageId), сформированный отправителем сообщения;
- категория статуса (//StatusCategory), содержащий одно из следующих возможных значений:
 - doesNotExist (Запрос с таким идентификатором не найден в БД СМЭВ);
 - requestIsQueued (Запрос находится в очереди асинхронной обработки);
 - requestIsAcceptedBySmev (Запрос доставляется поставщику);
 - requestIsRejectedBySmev (Запрос не прошёл асинхронную обработку);
 - underProcessing (Обрабатывается поставщиком вида сведений);
 - responseIsAcceptedBySmev (Запрос выполнен или отвергнут поставщиком сервиса (ответ находится в очереди ответов СМЭВ));
 - responseIsRejectedBySmev (Ответ не прошёл асинхронную обработку);

- messageIsArchived (Сообщение переведено в архив);
 - messageIsDelivered (Сообщение получено получателем).
- уведомление об описании статуса сообщения (//StatusDetails), содержащий описание статуса сообщения;
 - элемент типа //SmevFault, содержащий в себе элемент //Code, определяющий код ошибки, и элемент //Description, содержащий описание ошибки.

3. ТРЕБОВАНИЯ К СТРУКТУРЕ СООБЩЕНИЙ

3.1. ОБЩИЕ ПОЛОЖЕНИЯ

Электронные сообщения в системе межведомственного электронного взаимодействия передаются в формате XML в кодировке UTF-8 с указанием кодировки в заголовке сообщения. Соответствующие им WSDL и XSD файлы также должны использовать кодировку UTF-8 с указанием кодировки в заголовке сообщения.

ИС участников взаимодействия в теле электронных сообщений должны поддерживать применение блоков и элементов данных, а также электронных подписей, описанных в данном документе. Использование других блоков и элементов, отличных от описанных в данном документе, **не допускается**.

Для именования пространств имен элементов в сообщениях зарезервированы два источника со схемой URN (базовые URI):

- urn://x-artefacts-smev-gov-ru/;
- urn://smev-gov-ru/.

Процесс отправки ИС потребителя запроса и получения ответа на запрос от ИС поставщика представляет собой последовательность вызовов единого электронного сервиса СМЭВ информационными системами потребителя и поставщика:

- передача в СМЭВ запроса из ИС потребителя (//SendRequestRequest);
- получение из СМЭВ запроса в ИС поставщика (//GetRequestResponse);
- подтверждение поставщиком получения запроса из СМЭВ (//AckRequest);
- передача в СМЭВ ответа из ИС поставщика (//SendResponseRequest);
- получение из СМЭВ ответа в ИС потребителя (//GetResponseResponse)
- подтверждение потребителем получения ответа из СМЭВ (//AckRequest).

Перечисленные в скобках элементы являются, по своему назначению, **конвертами сообщений** (далее – **СМЭВ-конверты**), так как представляют собой «оболочку» для передачи сообщений в СМЭВ, включающих блоки и элементы служебных и бизнес данных, а также электронные подписи.

Наименования перечисленных элементов образуются из слов Send/Get и Request/Response, соответствующих назначению элемента. Первый слог в имени элемента образуется словом «Send» или «Get», которое соответствует выполняемому действию с точки зрения ИС участника взаимодействия. Например, с точки зрения потребителя, он посылает (Send) запрос, а с точки зрения поставщика, он получает (Get) этот же запрос. Вторым слогом образуется словом «Request» или «Response» и определяет назначение сообщения с точки зрения бизнес-логики: слово «Request» означает запрос от потребителя к поставщику, а слово «Response» означает ответ от поставщика к потребителю. Третий слог образуется также словом «Request» или «Response», но несет другой смысл: слово «Request» соответствует SOAP-запросу, а слово «Response» SOAP-ответу.

Элемент AckRequest (от acknowledgement request) является запросом на подтверждение и содержит ссылку на сообщение (идентификатор сообщения), получение которого подтверждается методом Ack.

3.2. СТРУКТУРА СООБЩЕНИЯ С ЗАПРОСОМ СВЕДЕНИЙ, КОТОРОЕ ИС ПОТРЕБИТЕЛЯ ПЕРЕДАЕТ В СМЭВ

Структура сообщения, соответствующая передаче в СМЭВ запроса от ИС потребителя к ИС поставщика, приведена на рисунке ниже (**Рисунок 12**).

ИС потребителя

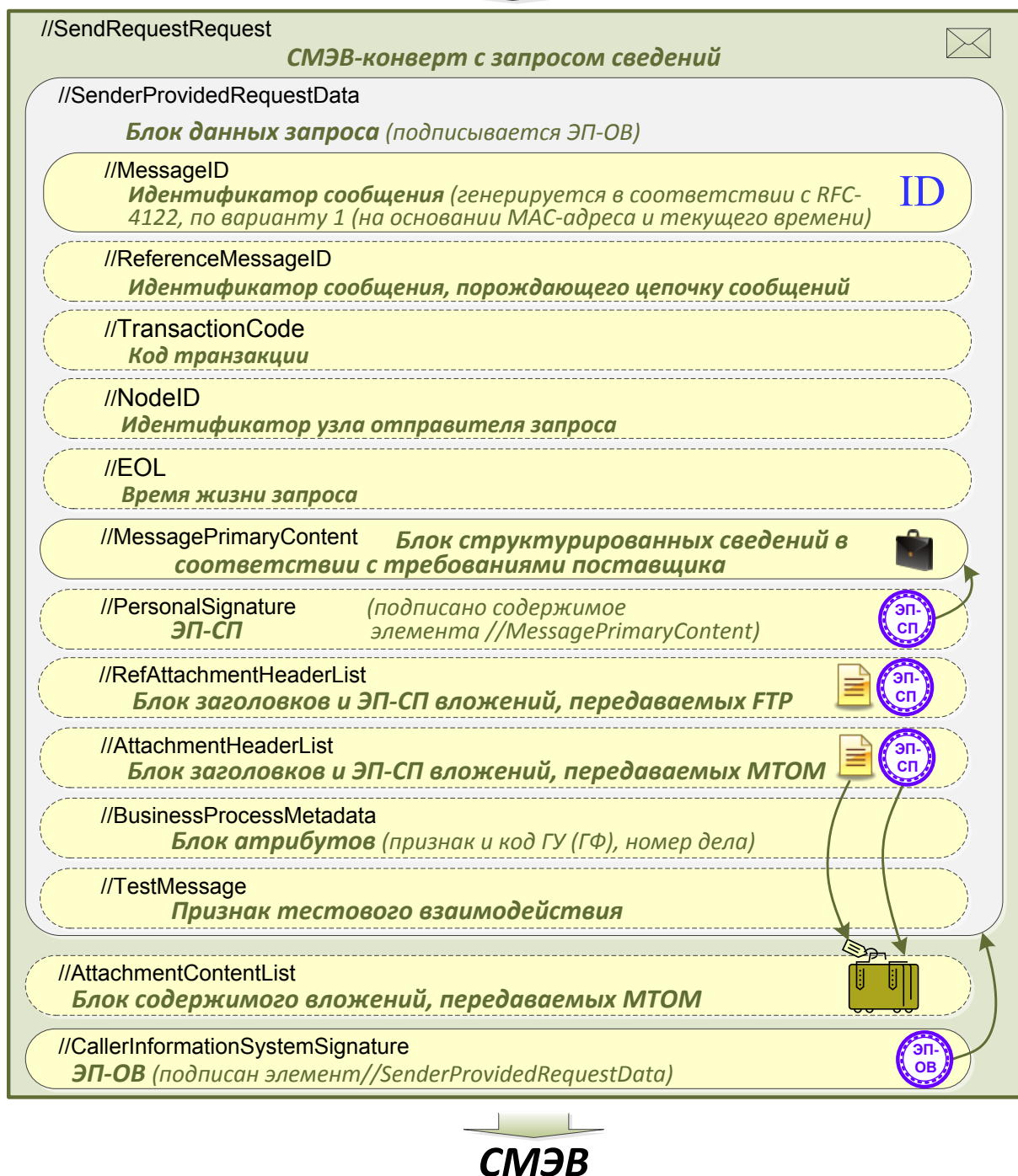


Рисунок 12 – Структура сообщения с запросом сведений, которое ИС потребителя передает в СМЭВ

Элементы XML-структуры на рисунке изображены в виде прямоугольников со скругленными (за исключением СМЭВ-конверта) краями, с привязкой к элементам (имена соответствующих элементов XML-структуры приведены в верхнем левом углу прямоугольников). Обязательные элементы изображены непрерывной линией, а необязательные – пунктирной. Элементы, соответствующие СМЭВ-конвертам, имеют в верхних правых углах изображения конвертов, а также дополнительно выделены темно-зеленой утолщенной линией и заливкой.

СМЭВ-конверт с запросом сведений (//SendRequestRequest), направляемый ИС потребителя в СМЭВ (для последующей передачи запроса из СМЭВ в ИС поставщика), включает следующие элементы:

- **блок данных запроса** (//SenderProvidedRequestData), который включает структурированные сведения в соответствии с требованиями поставщика, а также служебные данные, заполняемые потребителем сведений;
- **блок содержимого вложений** (//AttachmentContentList);
- **электронная подпись органа власти** (ЭП-ОВ) (//CallerInformationSystemSignature).

3.2.1 Блок данных запроса

Блок данных запроса может включать от двух до одиннадцати элементов, которые заполняются в ИС потребителя:

- идентификатор сообщения (//MessageID), обязательный элемент, идентификатор сообщения в виде UUID, основанного на времени, сгенерированный отправителем. UUID необходимо генерировать по версии 1 (см. п. 4.2 «Algorithms for Creating a Time-Based UUID» RFC 4122 <http://rfc.askapache.com/rfc4122/rfc4122.html#section-4.2>). СМЭВ использует метку времени, содержащуюся в UUID, для проверки срока годности сообщения, к которому относится данный UUID. Для СМЭВ срок годности одного сообщения составляет 24 часа;
- идентификатор первичного сообщения (//ReferenceMessageID), опциональный элемент, указывающий на первичный MessageID в цепочке запросов одной бизнес-транзакции. При отправке первичного запроса ReferenceMessageID и MessageID совпадают.
- код транзакции (//TransactionCode), опциональный элемент, указывающий на транзакцию оказания государственной услуги или выполнения государственной функции, в рамках которой посылается запрос. Если в транзакции запрос является первым, то данный элемент следует заполнять значением, полученным в СГКТ. Если в транзакции запрос является промежуточным, то данный элемент следует заполнять значением, полученным в запросе, на основании которого посылается данный запрос. Правила получения и использования кода транзакции приведены в разделе 8.
- идентификатор узла (сервера) отправителя запроса (//NodeID), опциональный элемент для маршрутизации ответа на запрос на сервер-отправитель, если информационная система отправителя запросов представляет собой многосерверную (многонодную) архитектуру.
- время жизни запроса (//EOL), опциональный элемент, определяющий время, до истечения которого запрос является для ИС потребителя актуальным.
- блок структурированных сведений в соответствии с требованиями поставщика (//MessagePrimaryContent), обязательный элемент, представляющий собой XML

документ, заполненный по формату, разработанному поставщиком сведений. Поставщик, для которого предназначен запрос, определяется в СМЭВ по полному имени корневого элемента в этом блоке. Этот блок не предназначен для передачи вложений, при возникновении такой необходимости следует использовать блоки содержимого вложений, заголовков и ЭП-СП вложений;

- электронная подпись должностного лица (далее - ЭП-СП), (`//PersonalSignature`). По требованиям поставщика сведений эта подпись может быть обязательной для подписи блока сведений по форматам поставщика. С помощью ЭП-СП подписывается элемент, находящийся в `//MessagePrimaryContent` (между открывающим и закрывающим тегами);
- блок заголовков и ЭП-СП вложений, передаваемых FTP (`//RefAttachmentHeaderList`), который содержит идентификаторы вложений, хэш коды вложений, MIME-типы вложений, а также ЭП-СП этих вложений в формате PKCS#7 detached (подробнее о порядке формирования электронных подписей см. раздел 4. «Электронные подписи»). Перед отправкой сообщения вложения должны быть загружены в файловое хранилище СМЭВ средствами FTP;
- блок заголовков и ЭП-СП вложений, передаваемых МТОМ (`//AttachmentHeaderList`), который содержит ссылки на идентификаторы вложений в блоке содержимого вложений, MIME-типы вложений, а также ЭП-СП этих вложений в формате PKCS#7 detached (подробнее о порядке формирования электронных подписей см. раздел 4. «Электронные подписи»);
- блок атрибутов бизнес-процесса (`//BusinessProcessMetadata`). Состав данных этого блока расширяемый и описывается отдельной XML-схемой `urn://x-artefacts-smev-gov-ru/services/message-exchange/business-process-metadata/1.0`. В настоящее время в него входят код государственной услуги или государственной функции согласно реестру государственных услуг, признак услуги или функции, код ФРГУ информационной системы-отправителя запроса, а также номер дела, в рамках которых сформирован запрос. Эта информация не требуется для работы СМЭВ и в настоящее время не обязательна для заполнения, однако она может быть полезна для разрешения вопросов участников взаимодействия по взаимодействию с СМЭВ;
- признак тестового взаимодействия (`//TestMessage`). Если этот элемент присутствует, то это означает, что запрос – тестовый. Данный признак используется для тестирования видов сведений.

Блок данных запроса подписывается ЭП-ОВ.

3.2.2 Блок содержимого вложений

Блок содержимого вложений может быть добавлен, если потребителю необходимо передать в ИС поставщика информацию (в том числе неструктурированную), которая не входит в блок структурированных сведений в соответствии с требованиями поставщика. Вложенные файлы и идентификаторы вложений располагаются вне подписанного с

помощью ЭП-ОВ блока данных запроса для корректной реализации кодирования вложений с помощью механизма оптимизации передачи сообщений МТОМ.

Суммарный объем вложенных файлов не должен превышать 5Мб. В противном случае при пересылке файлов необходимо использовать механизм Файлового хранилища (см. раздел 5).

Обращаем внимание, что значение элемента //Id блока содержимого вложений должно содержать в качестве первого (начального) символа латинскую букву или нижнее подчеркивание.

3.2.3 Электронная подпись органа власти

Электронная подпись ЭП-ОВ, формируемая от имени органа власти, участвующего в межведомственном взаимодействии и выступающего в роли потребителя сведений, подписывает блок данных запроса. С помощью ЭП-ОВ обеспечивается целостность запроса и идентификация ИС отправителя.

3.3. СТРУКТУРА СООБЩЕНИЯ С ЗАПРОСОМ СВЕДЕНИЙ, КОТОРОЕ ИС ПОСТАВЩИКА ПОЛУЧАЕТ ИЗ СМЭВ

Структура сообщения, соответствующего получению в ИС поставщика из СМЭВ запроса от ИС потребителя, приведена на рисунке ниже (Рисунок 13).

СМЭВ

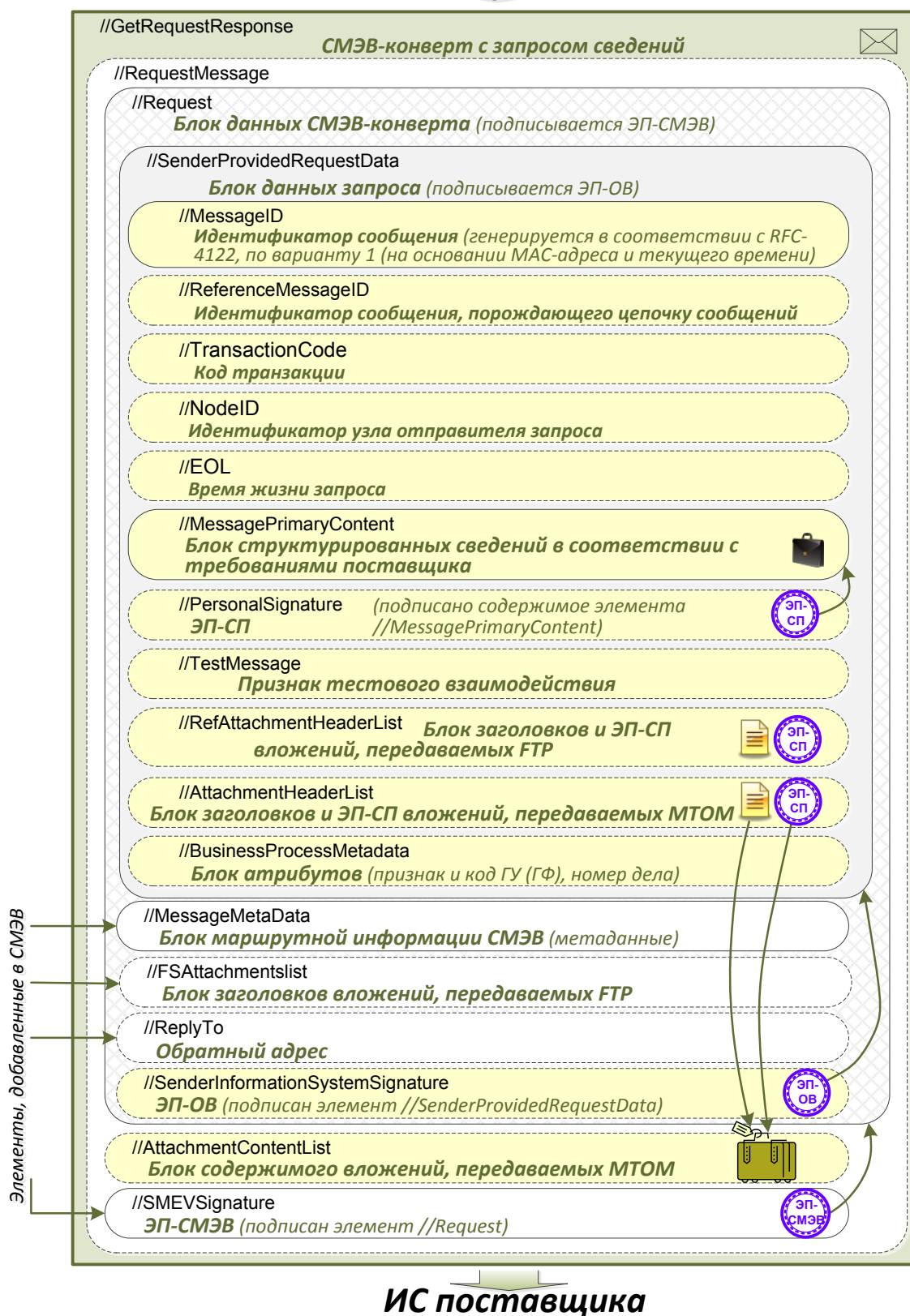


Рисунок 13 – Структура сообщения с запросом сведений, которое ИС поставщика получает из СМЭВ

При получении из СМЭВ SOAP-ответа, ИС поставщика проверяет в СМЭВ-конверте наличие элемента //RequestMessage (присутствует, если очередь запросов не пуста). Элемент //RequestMessage включает три элемента:

- **блок данных СМЭВ-конверта** (//Request);
- **блок содержимого вложений, передаваемых МТОМ** (//AttachmentContentList);
- **электронная подпись СМЭВ** (далее - ЭП-СМЭВ), (//SMEVSignature).

3.3.1 Блок данных СМЭВ-конверта

Блок данных СМЭВ-конверта //Request содержит элементы:

- блок данных запроса //SenderProvidedRequestData, сформированный отправителем запроса (см. п. 3.2.1);
- ЭП-ОВ, которой ИС отправителя подписан блок данных запроса.

а также три дополнительных элемента, добавленных СМЭВ (на рисунке выделены заливкой белым цветом):

- обратный адрес (//ReplyTo), необходимый для доставки ответа потребителю (обратный адрес не является мнемоникой отправителя сообщения или именем его очереди, его формат не специфицируется). При отправке ответа на запрос ИС поставщика копирует это значение в элемент //SenderProvidedResponseData/To, прочитав который, СМЭВ, в свою очередь, определяет, кому доставить ответ на запрос. Следует также иметь в виду, что в разных запросах, пришедших от одной и той же ИС отправителя, содержимое элемента //ReplyTo может отличаться.
- блок маршрутной информации СМЭВ (//MessageMetaData) с метаданными, включающими элементы:
 - тип сообщения (запрос «REQUEST», ответ «RESPONSE», рассылка «BROADCAST») (//MessageType);
 - информация об отправителе сообщения (//Sender), включающая вычисляемую на основе анализа сертификата ЭП-ОВ мнемонику отправителя, предназначенную для машинной обработки (Mnemonic);
 - метка времени получения в СМЭВ сообщения от ИС отправителя (//SendingTimeStamp). Содержит дату и время, начиная с которого отсчитывается срок исполнения запроса;
 - информация о получателе сообщения (//Recipient), определенная маршрутизатором и включающая мнемонику получателя, предназначенную для машинной обработки (//Mnemonic);
 - дата и время доставки сообщения получателю (//DeliveryTimeStamp).
- блок заголовков вложений, передаваемых FTP (//FSAttachmentsList) с данными, включающими элементы:

- идентификатор вложения (//uuid);
- имя пользователя (в совокупности с паролем необходимо для получения доступа к файловому хранилищу СМЭВ, чтобы скачать файл вложения) (//UserName);
- пароль (в совокупности с именем пользователя необходим для получения доступа к файловому хранилищу СМЭВ, чтобы скачать файл вложения) (//Password);
- имя файла вложения (//FileName).

3.3.2 Блок содержимого вложений

Блок содержимого вложений, передаваемых МТОМ, не изменяется при прохождении через СМЭВ и соответствует блоку содержимого вложений сообщения с запросом, которое ИС потребителя передала в СМЭВ.

3.3.3 Электронная подпись СМЭВ

С помощью ЭП-СМЭВ (//SMEVSignature) подписываются блок данных запроса (вместе с ЭП-ОВ), а также добавленные в СМЭВ блок маршрутной информации СМЭВ и обратный адрес.

С помощью ЭП-СМЭВ обеспечивается целостность сообщения с запросом на всем пути от отправителя до получателя, подтверждение поступления запроса из СМЭВ во время, указанное в метке времени, и право ИС потребителя на направление запроса в ИС поставщика.

3.4. СТРУКТУРА СООБЩЕНИЯ С ОТВЕТОМ, КОТОРОЕ ИС ПОСТАВЩИКА ПЕРЕДАЕТ В СМЭВ

Структура сообщения, соответствующего передаче в СМЭВ ответа от ИС поставщика, приведена на рисунке ниже (**Рисунок 14**).

ИС поставщика

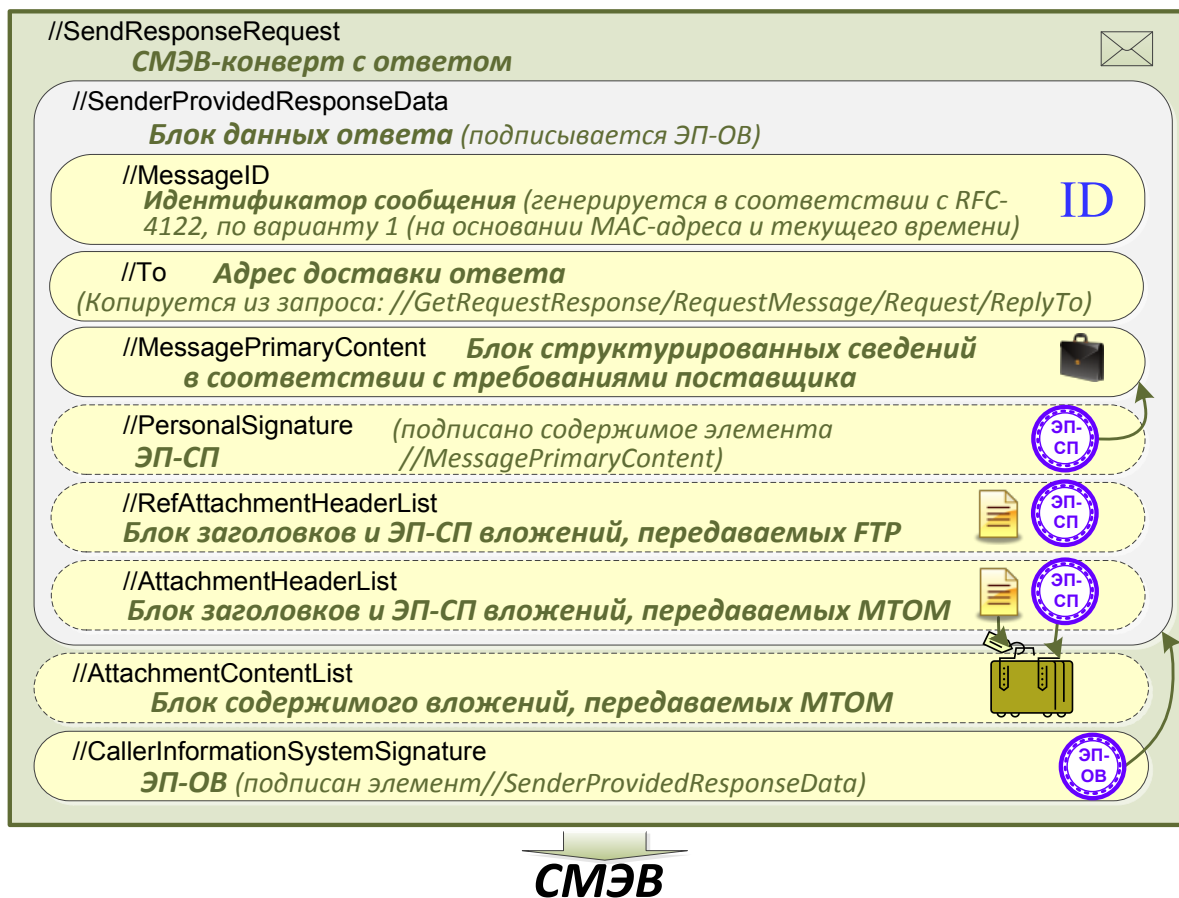


Рисунок 14 – Структура сообщения с ответом, которое ИС поставщика передает в СМЭВ

Назначение элементов сообщения, с помощью которого передается ответ от ИС поставщика в СМЭВ (для последующей передачи в ИС потребителя), в основном соответствует назначению элементов сообщений, с помощью которых был передан запрос от ИС потребителя к ИС поставщика. Отличие состоит в появлении нескольких новых элементов, а также в изменении названий некоторых элементов.

СМЭВ-конверт с ответом (//SendResponseRequest), направляемый ИС поставщика в СМЭВ (для последующей передачи ответа из СМЭВ в ИС потребителя), включает элементы:

- **блок данных ответа** (//SenderProvidedResponseData), который включает структурированные сведения в соответствии с требованиями поставщика, а также служебные данные, заполняемые поставщиком сведений;
- **блок содержимого вложений, передаваемых МТОМ** (//AttachmentContentList);
- **электронная подпись органа власти** (ЭП-ОВ) (//CallerInformationSystemSignature).

3.4.1 Блок данных ответа

Блок данных ответа может включать от трех до шести элементов, которые заполняются в ИС поставщика:

- идентификатор сообщения (`//MessageID`), обязательный элемент, идентификатор сообщения в виде UUID, основанного на времени, сгенерированный отправителем. UUID необходимо генерировать по версии 1 (см. п. 4.2 «Algorithms for Creating a Time-Based UUID» RFC 4122 <http://rfc.askapache.com/rfc4122/rfc4122.html#section-4.2>). СМЭВ использует метку времени, содержащуюся в UUID, для проверки срока годности сообщения, к которому относится данный UUID. Для СМЭВ срок годности одного сообщения составляет 24 часа;
- адрес доставки ответа (`//To`), обязательный элемент, в который должно быть скопировано содержимое элемента `//GetRequestResponse/RequestMessage/Request/ReplyTo` запроса, на который отправляется ответ;
- блок структурированных сведений в соответствии с требованиями поставщика (`//MessagePrimaryContent`), обязательный элемент, представляющий собой XML документ, заполненный по формату, разработанному поставщиком сведений. Этот блок не предназначен для передачи вложений, при возникновении такой необходимости следует использовать блоки содержимого вложений, заголовков и ЭП-СП вложений;
- электронная подпись должностного лица (далее - ЭП-СП), (`//PersonalSignature`). По требованиям поставщика сведений эта подпись может быть обязательной для подписи блока сведений по форматам поставщика. С помощью ЭП-СП подписывается элемент, находящийся в `//MessagePrimaryContent` (между открывающим и закрывающим тегами);
- блок заголовков и ЭП-СП вложений, передаваемых FTP (`//RefAttachmentHeaderList`), который содержит идентификаторы вложений, хэш коды вложений, MIME-типы вложений, а также ЭП-СП этих вложений в формате PKCS#7 detached (подробнее о порядке формирования электронных подписей см. раздел 4. «Электронные подписи»). Перед отправкой сообщения вложения должны быть загружены в файловое хранилище СМЭВ средствами FTP;
- блок заголовков и ЭП-СП вложений, передаваемых МТОМ (`//AttachmentHeaderList`), который содержит ссылки на идентификаторы вложений в блоке содержимого вложений, MIME-типы вложений, а также ЭП-СП этих вложений в формате PKCS#7 detached (подробнее о порядке формирования электронных подписей см. раздел 4. «Электронные подписи»).

А также дополнительные (`//choice`) элементы:

- Отклонение запроса (`//RequestRejected`), который может быть использован ИС поставщика для информирования ИС потребителя об отклонении его запроса. Содержит код причины отклонения запроса (`//RejectionReasonCode`) (см. п.

2.3.6) и описание причины отклонения запроса, в человекочитаемом виде (`//RejectionReasonDescription`), которое заполняется ИС поставщика.

- Статус запроса (`//RequestStatus`), который может быть использован ИС поставщика для информирования ИС потребителя о статусе обработки его запроса. Содержит код бизнес-статуса запроса (`//StatusCode`), который заполняется ИС поставщика, и описание бизнес-статуса запроса, в человекочитаемом виде (`//StatusDescription`), которое также заполняется ИС поставщика.
- Статус обработки сообщения (запроса либо ответа на запрос) в СМЭВ (`//AsyncProcessingStatus`) (см. п. 2.6.3). Данный элемент предназначен для использования только СМЭВ для информирования ИС о статусе обработки их сообщений в СМЭВ. В то же время данный элемент не предназначен для использования ИС поставщиков для информирования ИС потребителей о статусах обработки их сообщений. В случае использования ИС поставщика данного элемента в отправляемом ею ответе СМЭВ вернет данной ИС ошибку.

3.4.2 Блок содержимого вложений

Блок содержимого вложений может быть добавлен, если поставщику необходимо передать информацию (в том числе неструктурированную), которая не входит в блок данных ответа.

Суммарный объем вложенных файлов не должен превышать 5Мб. В противном случае при пересылке файлов необходимо использовать механизм Файлового хранилища (см. раздел 5).

Обращаем внимание, что значение элемента `//Id` блока содержимого вложений должно содержать в качестве первого (начального) символа латинскую букву или нижнее подчеркивание.

3.4.3 Электронная подпись органа власти

Электронная подпись ЭП-ОВ, формируемая от имени органа власти, участвующего в межведомственном взаимодействии и выступающего в роли поставщика сведений, подписывает блок данных ответа. С помощью ЭП-ОВ обеспечивается целостность ответа и идентификация ИС отправителя.

3.5. СТРУКТУРА СООБЩЕНИЯ С ОТВЕТОМ, КОТОРОЕ ИС ПОТРЕБИТЕЛЯ ПОЛУЧАЕТ ИЗ СМЭВ

При получении из СМЭВ SOAP-ответа, ИС потребителя проверяет в СМЭВ-конверте наличие элемента //ResponseMessage (присутствует, если очередь ответов не пуста). Элемент //ResponseMessage включает три элемента (**Рисунок 15**):

- **блок данных СМЭВ-конверта** (//Response);
- **блок содержимого вложений, передаваемых МТОМ** (//AttachmentContentList);
- **электронная подпись СМЭВ** (//SMEVSignature).

СМЭВ

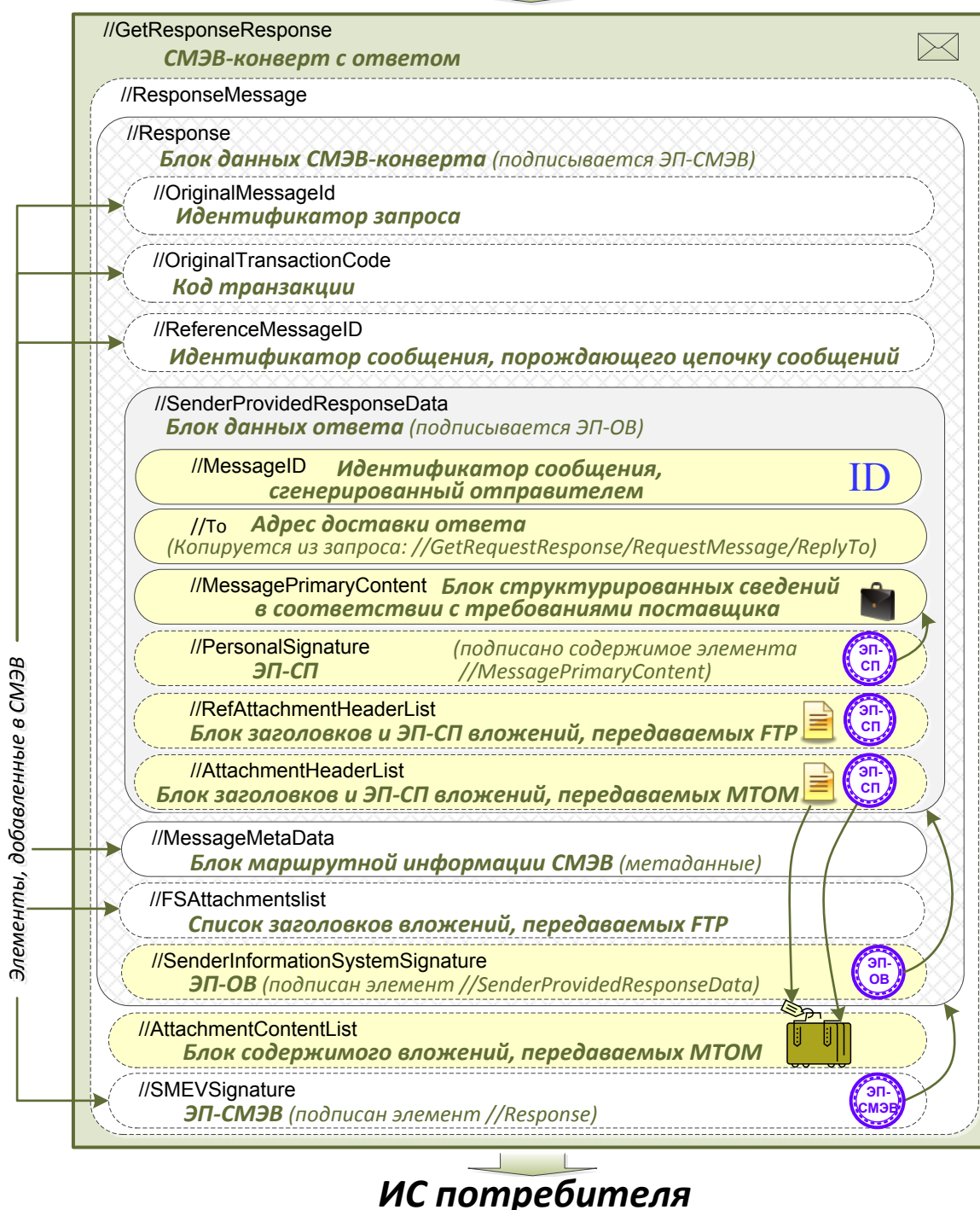


Рисунок 15 – Общая структура сообщения с ответом, которое ИС потребителя получает из СМЭВ (без указания элементов RequestRejected, RequestStatus или AsyncProcessingStatus)

3.5.1 Блок данных СМЭВ-конверта

Блок данных СМЭВ-конверта //Response содержит элементы:

- блок данных ответа //SenderProvidedResponseData, сформированный отправителем ответа (см. п. 3.4.1);

- ЭП-ОВ, которой ИС отправителя подписан блок данных ответа, а также пять дополнительных элементов, добавленными СМЭВ (на рисунке выделены заливкой белым цветом):
- идентификатор запроса (`//OriginalMessageId`), заполняемый СМЭВ значением идентификатора запроса, на который пришел ответ.
- код транзакции (`//OriginalTransactionCode`), заполняемый СМЭВ значением кода транзакции, в рамках которой пришел ответ.
- идентификатор первичного запроса (`//ReferenceMessageId`), заполняемый СМЭВ значением идентификатора запроса, являющегося источником цепочки запросов. Если в целочке запросов всего один запрос, то этот элемент заполняется значением элемента `//OriginalMessageId`.
- блок маршрутной информации СМЭВ (`//MessageMetaData`) с метаданными, включающими элементы:
 - тип сообщения (запрос «REQUEST», ответ «RESPONSE», рассылка «BROADCAST») (`//MessageType`);
 - информация об отправителе сообщения (`//Sender`), включающая вычисляемую на основе анализа сертификата ЭП-ОВ мнемонику отправителя, предназначенную для машинной обработки (`Mnemonic`);
 - метка времени получения в СМЭВ сообщения от ИС отправителя (`//SendingTimeStamp`). Содержит дату и время, начиная с которого отсчитывается срок исполнения запроса;
 - информация об отправителе сообщения (`//Recipient`), определенная маршрутизатором и включающая мнемонику получателя, предназначенную для машинной обработки (`//Mnemonic`);
 - дата и время доставки сообщения получателю (`//DeliveryTimeStamp`);
- блок заголовков вложений, передаваемых FTP (`//FSAttachmentsList`), с данными, включающими элементы:
 - идентификатор вложения (`//uuid`);
 - имя пользователя (в совокупности с паролем необходимо для получения доступа к файловому хранилищу СМЭВ, чтобы скачать файл вложения) (`//UserName`);
 - пароль (в совокупности с именем пользователя необходим для получения доступа к файловому хранилищу СМЭВ, чтобы скачать файл вложения) (`//Password`);
 - имя файла вложения (`//FileName`).

3.5.2 Блок содержимого вложений

Структура блока содержимого вложений, передаваемых МТОМ //AttachmentContentList, аналогична одноименному элементу в сообщении с ответом, направленном из ИС поставщика в СМЭВ.

3.5.3 Электронная подпись СМЭВ

Структура ЭП-СМЭВ //SMEVSignature аналогична одноименному элементу в //RequestMessage запроса (см. п. 3.3.3).

С помощью ЭП-СМЭВ обеспечивается целостность сообщения с ответом на всем пути от отправителя до получателя, подтверждение поступления ответа из СМЭВ во время, указанное в метке времени, и право на обращение ИС потребителя за ответом.

3.6. СТРУКТУРА СООБЩЕНИЯ С ОТВЕТОМ О СТАТУСЕ РАНЕЕ ОТПРАВЛЕННОГО В СМЭВ СООБЩЕНИЯ, КОТОРОЕ ИС ПОТРЕБИТЕЛЯ ПОЛУЧАЕТ ИЗ СМЭВ

При получении из СМЭВ SOAP-ответа, ИС потребителя проверяет в СМЭВ-конверте наличие элемента //ResponseMessage (присутствует, если очередь ответов не пуста). Элемент //ResponseMessage включает два элемента (Рисунок 16):

- **блок данных СМЭВ-конверта** (//Response);
- **электронная подпись СМЭВ** (//SMEVSignature).

СМЭВ

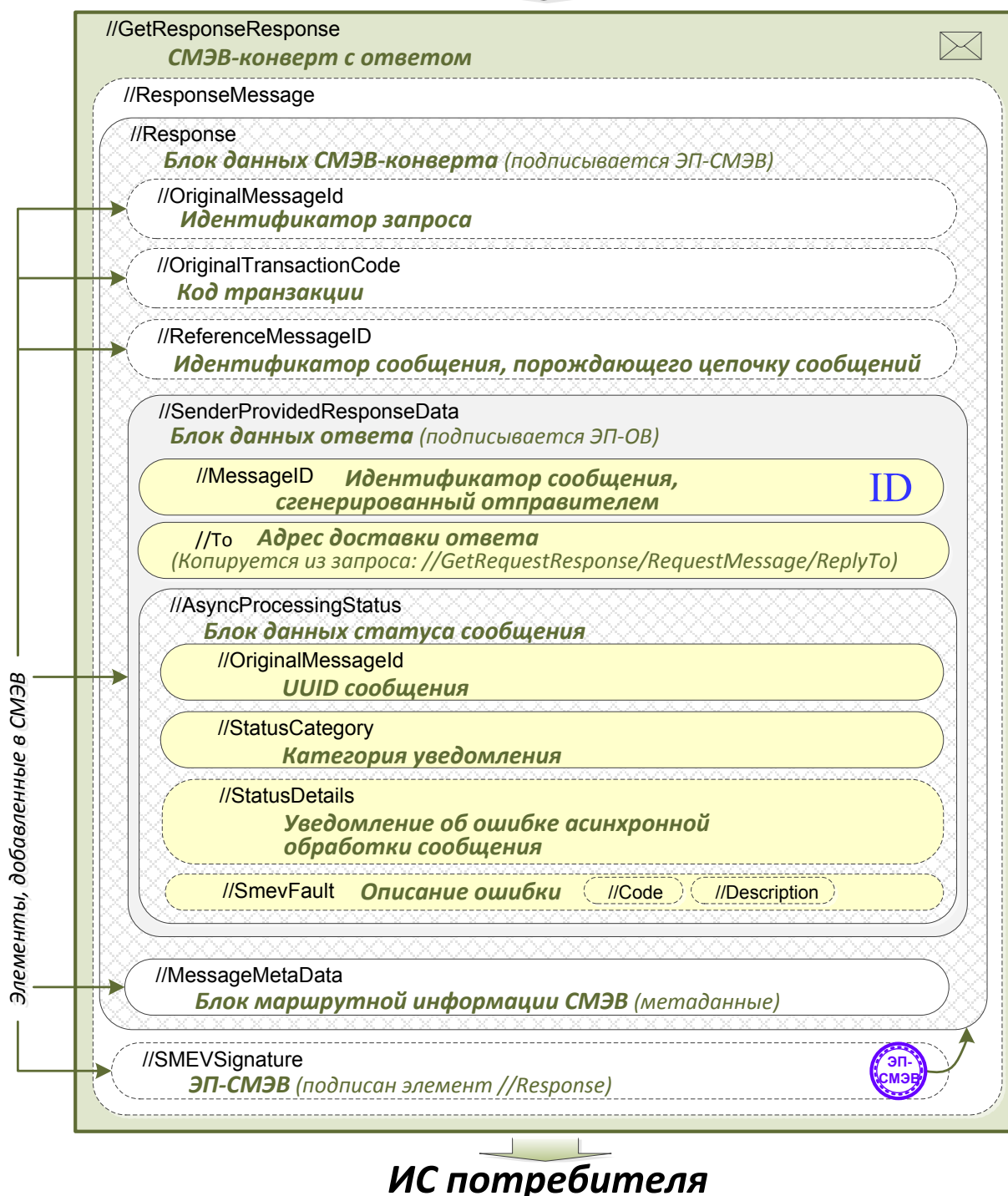


Рисунок 16 – Общая структура сообщения с блоком AsyncProcessingStatus, которое ИС потребителя получает из СМЭВ

3.6.1 Блок данных СМЭВ-конверта

Блок данных СМЭВ-конверта //Response содержит элементы:

- идентификатор запроса (//OriginalMessageId), заполняемый СМЭВ значением идентификатора запроса, на который высылается ответ.

- код транзакции (`//OriginalTransactionCode`), заполняемый СМЭВ значением кода транзакции, в рамках которой высылается ответ.
- идентификатор первичного запроса (`//ReferenceMessageID`), заполняемый СМЭВ значение идентификатора запроса, являющегося источником цепочки запросов. Если в целочке запросов всего один запрос, то этот элемент заполняется значением элемента `//OriginalMessageId`.
- блок данных ответа `//SenderProvidedResponseData`, сформированный отправителем ответа;
- блок маршрутной информации СМЭВ (`//MessageMetaData`) с метаданными (см. п. 3.5.1).

Блок данных ответа включает три элемента, которые заполняются в СМЭВ:

- идентификатор сообщения (`//MessageID`), обязательный элемент, идентификатор сообщения в виде UUID, основанного на времени, сгенерированный отправителем. UUID необходимо генерировать по версии 1 (см. п. 4.2 «Algorithms for Creating a Time-Based UUID» RFC 4122 <http://rfc.askapache.com/rfc4122/rfc4122.html#section-4.2>);
- адрес доставки ответа (`//To`), обязательный элемент, в который копируется содержимое элемента `//GetRequestResponse/RequestMessage/ Request/ReplyTo` запроса, на который отправляется ответ;
- блок данных статуса сообщения (`//AsyncProcessingStatus`) (см. п. 2.6.3).

3.6.2 Электронная подпись СМЭВ

Структура ЭП-СМЭВ `//SMEVSignature` аналогична одноименному элементу в `//RequestMessage` запроса (см. п. 3.3.3).

С помощью ЭП-СМЭВ обеспечивается целостность сообщения с ответом на всем пути от отправителя до получателя, подтверждение поступления ответа из СМЭВ во время, указанное в метке времени, и право на обращение ИС потребителя за ответом.

3.7. ОСОБЕННОСТИ СХЕМЫ СЕРВИСА СМЭВ 1.1* (1.2)

Для плавного перехода от схемы 1.1. к новым возможностям схемы 1.2 (опубликованной в методических рекомендациях 0.9.8.2) произведены следующие изменения:

- Осуществлено обновление схемы сервиса СМЭВ. Новая схема включает в себя как особенности схемы 1.1, так и новые элементы схемы версии 1.2
- Номер новой схемы будет понижен до версии 1.1 (условное название новой схемы 1.1*).

Все участники взаимодействия, желающие остаться на версии схемы 1.1, смогут отправлять сообщения и получать сообщения из своих очередей доставки. При этом не доступна возможность получения сообщений из статусной очереди (`GetStatus`) и возможность получать сообщения со статусами в ответах.

Все участники взаимодействия, желающие перейти на версию схемы 1.1* (1.2), смогут отправлять сообщения, получать сообщения из своих очередей доставки и статусных очередей. При этом для осуществления информационного взаимодействия по какому-либо виду сведений с применением новых полей схемы сервиса версии 1.1* необходимо, чтобы на указанную версию схемы перешли Потребитель и Поставщик по этому виду сведений.

Схема сервиса версии 1.1*, как и ранее распространенная схема версии 1.2, включает в себя ряд новых элементов, обеспечивающих расширенные возможности процесса обмена сообщениями. Перечень новых элементов схемы 1.1* приведен в таблице 2.

Таблица 2 – Перечень новых элементов схемы 1.1* (1.2)

№	Элемент	Описание изменения	Комментарий
1	Новые элементы схемы 1.1* (1.2)		
1.1	ReferenceMessageID	Идентификатор сообщения, порождающего цепочку сообщений. Включен в содержательную часть: <ul style="list-style-type: none"> – запроса SenderProvidedRequestData (см. п. 3.2); – ответа с сообщением из очереди доставки ответов Response (см. п. 3.5). 	Является опциональным элементом, но может использоваться для формирования целочки запросов в рамках одной бизнес-транзакции, путем помещения в данное поле ID первого сообщения в цепочке запросов (см. п. 3.2.1).
1.2	TransactionCode	Идентификатор кода транзакции запроса. Включен в содержательную часть запроса SenderProvidedRequestData.	Описание использования приведено в разделе 8
1.3	OriginalTransactionCode	Идентификатор кода транзакции ответа с сообщением из очереди доставки ответов Response (см. п. 3.5).	Заполняется автоматически СМЭВ на основании кода транзакции запроса
1.4	RequestStatus	Элемент, определяющий структуру бизнес-статуса обработки ответа на запрос. Включен в содержательную часть ответа на запрос SenderProvidedResponseData как <choice> элемент (см. п. 3.4.1).	Элемент включает следующий набор параметров: <ul style="list-style-type: none"> – Код бизнес-статуса запроса (обязательный параметр). – Пару параметров «ключ»-«значение» (опциональный параметр). – Расширенное описание бизнес-статуса запроса (обязательный параметр).
1.5	AsyncProcessingStatus	Элемент, определяющий структуру ошибки асинхронной обработки запроса. Включен в содержательную часть	Используется как элемент выбора в конверте SenderProvidedRequestData (см. п. 3.2).

№	Элемент	Описание изменения	Комментарий
		ответа на запрос SenderProvidedResponseData как <choice> элемент (см. п. 3.4.1).	
1.6	SmevFault	Элемент, определяющий структуру пары параметров «код»-«описание» ошибки. Включен в содержательную часть AsyncProcessingStatus (см. п. 3.4.1).	Заполняется кодом ошибки. Элемент конверта AsyncProcessingStatus. Входит в содержательную часть ответа на запрос сообщения из статусной очереди SmevAsyncProcessingMessage и содержится в элементе AsyncProcessingStatusData. Также элемент AsyncProcessingStatus включен в содержательную часть ответа на запрос SenderProvidedResponseData как элемент типа <choice>(см. п. 3.4.1) . Является опциональным элементом
1.7	EOL	Элемент, определяющий время актуальности сообщения. Включен в содержательную часть запроса SenderProvidedRequestData (см. п. 3.2.1).	Если отправляемое сообщение должно иметь срок актуальности, то в элемент EOL следует добавить метку времени истечения срока актуальности сообщения с указанием временной зоны (см. п. 3.2.1). Является опциональным элементом
1.8	NodeID	Элемент, определяющий мнемонический код сервера отправителя сообщения. Включен в содержательную часть запроса SenderProvidedRequestData (см. п. 3.2.1)..	Элемент введен для маршрутизации ответа на запрос на сервер-отправитель, если информационная система отправителя запросов представляет собой многосерверную (многонодную) архитектуру (см. п. 3.2.1).. Является опциональным элементом
1.9	AsyncProcessingStatusData	Конверт для AsyncProcessingStatus	Используется только для ошибок push-нотификации (см. п. 2.6). Статусы обработки сообщений возвращаются непосредственно в ответах СМЭВ.
2.0	RejectionReasonCode	Подэлемент – RejectionReasonCode элемента RequestRejected может принимать новое значение FAILURE	Код ошибок запроса может возвращать значение FAILURE (уведомление об отсутствии сведений).

3.8. СТРУКТУРА СООБЩЕНИЯ С ВЛОЖЕНИЯМИ

Сведения могут передаваться как в теле сообщения, так и во вложении. Вложения целесообразно использовать для передачи неструктурированной информации (файлы бинарного формата) или структурированной информации, размер которой превышает технологическое ограничение СМЭВ (5 Мб).

Использование вложений включает в себя два этапа:

- описание форматов вложений, которые предполагается передавать;

- непосредственная передача вложений.

Описание форматов вложений выполняется на этапе проектирования XSD-описания вида сведений.

В блок структурированных сведений (/MessagePrimaryContent) необходимо включить XSD-описание формата вложения. Указанное описание включает в себя следующие обязательные сведения:

1. признак того, является или нет вложение неструктурированным;
2. признак того, будет ли вложение помещено в zip-архив;
3. способ передачи вложения (ссылка в ФХ или элемент бинарного содержимого);
4. ссылка ФХ на файл отсоединенной электронной подписи вложения;
5. список объявлений возможных форматов вложений;
6. блок описания конкретных форматов структурированных вложений.

Данное описание может быть включено либо непосредственно в основной XSD-файл, описывающий формат вида сведений, либо быть импортированным из отдельного XSD-файла. В последнем случае блок описания конкретных форматов структурированных вложений должен быть вынесен в отдельный XSD-файл. Ниже представлены примеры описания форматов вложений.

Пример описания блока вложений размещенного в основном XSD-описании вида сведений:

```
<!-- Блок описания вложений внутри бизнес-блока сообщения СМЭВ3
      Может быть включен также в реестровую запись.
-->
<xsd:complexType name="_AttachmentsBlockType">
  <xsd:sequence>
    <xsd:element name="AttachmentDescription" type="tns:_AttachmentDescriptionType"
minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="_AttachmentDescriptionType">
  <xsd:sequence>
    <!-- Указание способа передачи вложения: через ФХ или методом MTOM-передачи -->
    <xsd:choice>
      <!-- Ссылка на папку ФХ, в котором содержится передаваемое вложение-->
      <xsd:element name="AttachmentFSLink" type="xsd:string"/>
      <!-- Признак передачи вложения методом MTOM-->
      <xsd:element name="IsMTOMAttachmentContent" type="xsd:boolean"/>
    </xsd:choice>

    <!-- Описание возможных форматов вложения -->
    <xsd:choice>
      <xsd:element name="AttachmentFormat1" type="tns:_StructuredAttachmentFormatType1" />
      <xsd:element name="AttachmentFormat2" type="tns:_StructuredAttachmentFormatType2" />
    </xsd:choice>

    <!-- Блок передачи файла отсоединенной электронной подписи вложения -->
    <xsd:element name="AttachmentSignatureFSLink" type="xsd:string" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="_StructuredAttachmentFormatType1">
  <xsd:sequence>
    <!-- Обязательное и явное указание: является ли формат структурированным или не является
-->
    <xsd:element name="IsUnstructuredFormat" type="xsd:boolean" />
    <!-- Обязательное и явное указание: будет ли содержимое вложения упаковываться Zip-
```

```

алгоритмом -->
    <xsd:element name="IsZippedPacket" type="xsd:boolean" />
    <!-- Если формат структурированный, то описываем его структуру -->
    <xsd:element name="DocumentRecordNumber" type="xsd:int" />
    <xsd:element name="DebtorName" type="xsd:string" />
    <xsd:element name="DebtorIdDoc" type="xsd:string" />
    <xsd:element name="PenaltyAmount" type="xsd:double" />
    <xsd:element name="ExecutionDate" type="xsd:date" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="_StructuredAttachmentFormatType2">
  <xsd:sequence>
    <!-- Явно указываем является ли формат структурированным или нет -->
    <xsd:element name="IsUnstructuredFormat" type="xsd:boolean" />
    <!-- Обязательное и явное указание: будет ли содержимое вложения упаковываться Zip-
алгоритмом -->
    <xsd:element name="IsZippedPacket" type="xsd:boolean" />
    <!-- Если формат структурированный, то описываем его структуру -->
    <xsd:element name="DocumentRecordNumber" type="xsd:int" />
    <xsd:element name="FilialCode" type="xsd:string" />
    <xsd:element name="ProcessingStatus" type="xsd:double" />
  </xsd:sequence>
</xsd:complexType>

```

В данном примере приведены объявления и описания двух форматов вложений: `_StructuredAttachmentFormatType1`, `_StructuredAttachmentFormatType2`. Объявление данных форматов представлено в конструкции

```

<xsd:choice>
  <xsd:element name="AttachmentFormat1" type="tns:_StructuredAttachmentFormatType1" />
  <xsd:element name="AttachmentFormat2" type="tns:_StructuredAttachmentFormatType2" />
</xsd:choice>

```

При использовании отсоединенного способа описания форматов вложений. В основном файле описания вида сведений должны быть определены блоки `AttachmentsBlockType`, `AttachmentDescriptionType`. А в отсоединенный файл описания вложений вынесены `_StructuredAttachmentFormatTypeXX`. Допускается не более 30 объявлений форматов.

Пример описания блока вложений, формат которых вынесен в отсоединенный XSD-файл:

Содержание размещается в основном XSD-файле описания вида сведений:

```

<xsd:complexType name="_AttachmentsBlockType">
  <xsd:sequence>
    <xsd:element name="AttachmentDescription" type="tns:_AttachmentDescriptionType"
minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="_AttachmentDescriptionType">
  <xsd:sequence>
    <!-- Указание способа передачи вложения: через ФХ или методом МТОМ-передачи -->
    <xsd:choice>
      <!-- Ссылка на папку ФХ, в котором содержится передаваемое вложение-->
      <xsd:element name="AttachmentFSLink" type="xsd:string"/>
      <!-- Признак передачи вложения методом МТОМ-->
      <xsd:element name="IsMTOMAttachmentContent" type="xsd:boolean"/>
    </xsd:choice>

    <!-- Описание возможных форматов вложения -->
    <xsd:choice>
      <xsd:element name="AttachmentFormat1" type="tns:_StructuredAttachmentFormatType1" />
      <xsd:element name="AttachmentFormat2" type="tns:_StructuredAttachmentFormatType2" />
    </xsd:choice>

    <!-- Блок передачи файла отсоединенной электронной подписи вложения -->
    <xsd:element name="AttachmentSignatureFSLink" type="xsd:string" minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>

```



```
</xsd:sequence>
</xsd:complexType>
```

Содержание размещается в отсоединенном XSD-файле описания форматов вложений:

```
<xsd:complexType name="_StructuredAttachmentFormatType1">
  <xsd:sequence>
    <!-- Обязательное и явное указание: является ли формат структурированным или не является -->
    <xsd:element name="IsUnstructuredFormat" type="xsd:boolean" />
    <!-- Обязательное и явное указание: будет ли содержимое вложения упаковываться Zip-
    алгоритмом -->
    <xsd:element name="IsZippedPacket" type="xsd:boolean" />
    <!-- Если формат структурированный, то описываем его структуру -->
    <xsd:element name="DocumentRecordNumber" type="xsd:int" />
    <xsd:element name="DebtorName" type="xsd:string" />
    <xsd:element name="DebtorIdDoc" type="xsd:string" />
    <xsd:element name="PenaltyAmount" type="xsd:double" />
    <xsd:element name="ExecutionDate" type="xsd:date" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="_StructuredAttachmentFormatType2">
  <xsd:sequence>
    <!-- Явно указываем является ли формат структурированным или нет -->
    <xsd:element name="IsUnstructuredFormat" type="xsd:boolean" />
    <!-- Обязательное и явное указание: будет ли содержимое вложения упаковываться Zip-
    алгоритмом -->
    <xsd:element name="IsZippedPacket" type="xsd:boolean" />
    <!-- Если формат структурированный, то описываем его структуру -->
    <xsd:element name="DocumentRecordNumber" type="xsd:int" />
    <xsd:element name="FilialCode" type="xsd:string" />
    <xsd:element name="ProcessingStatus" type="xsd:double" />
  </xsd:sequence>
</xsd:complexType>
```

Непосредственная передача вложений осуществляется путем заполнения блоков:

- Блок заголовков и ЭП-СП вложений, передаваемых FTP (//RefAttachmentHeaderList);
- Блок заголовков и ЭП-СП вложений, передаваемых MTOM (//AttachmentHeaderList);
- Блок содержимого вложений, передаваемых MTOM (//AttachmentContentList).

4. ЭЛЕКТРОННЫЕ ПОДПИСИ

4.1. ВИДЫ ЭЛЕКТРОННЫХ ПОДПИСЕЙ

В электронных сообщениях, передаваемых через СМЭВ, применяются следующие усиленные квалифицированные электронные подписи:

- электронная подпись, формируемая от имени должностного лица органа власти, участвующего в межведомственном взаимодействии (далее - **ЭП-СП**);
- электронная подпись, формируемая от имени органа власти, участвующего в межведомственном взаимодействии (далее - **ЭП-ОВ**);
- электронная подпись, формируемая в СМЭВ при обработке электронных сообщений, передаваемых через СМЭВ (далее - **ЭП-СМЭВ**).

Формирование ЭП-ОВ аналогично простановке печати организации на подписанном должностным лицом документе. ЭП-СМЭВ в этом случае можно считать аналогом печати почтовой организации на конверте, в котором передается документ.

Электронная подпись ЭП-СП является **необязательной**, а ее включение в состав сообщения может быть обусловлено наличием соответствующего нормативно закреплённого требования, в котором поставщик устанавливает необходимость подписания запроса уполномоченным лицом. Соответствующее требование должно быть отражено в Описании поставляемого вида сведений.

Электронные подписи ЭП-ОВ и ЭП-СМЭВ являются **обязательными**.

4.2. ПОРЯДОК ИСПОЛЬЗОВАНИЯ ЭЛЕКТРОННЫХ ПОДПИСЕЙ

4.2.1 Использование электронных подписей при передаче запроса

Передача запроса от потребителя к поставщику сервиса сопровождается операциями по формированию и проверке электронных подписей (Рисунок 17).

Перед отправкой сообщения с запросом, должностное лицо ОВ может подписать (при необходимости) с помощью ЭП-СП два элемента в сообщении:

- блок структурированных сведений в соответствии с требованиями поставщика (подписывается содержимое элемента //MessagePrimaryContent, заключенное между открывающим и закрывающим тегами элемента). ЭП-СП хранится в элементе //PersonalSignature;
- блок содержимого вложений (файлы, размещенные в элементе //AttachmentContentList). Каждый из файлов, размещенных в элементе //AttachmentContentList, подписывается отдельной ЭП-СП. Соответствующие ЭП-СП передаются в блоке заголовков и ЭП-СП вложений (элемент //AttachmentHeaderList).

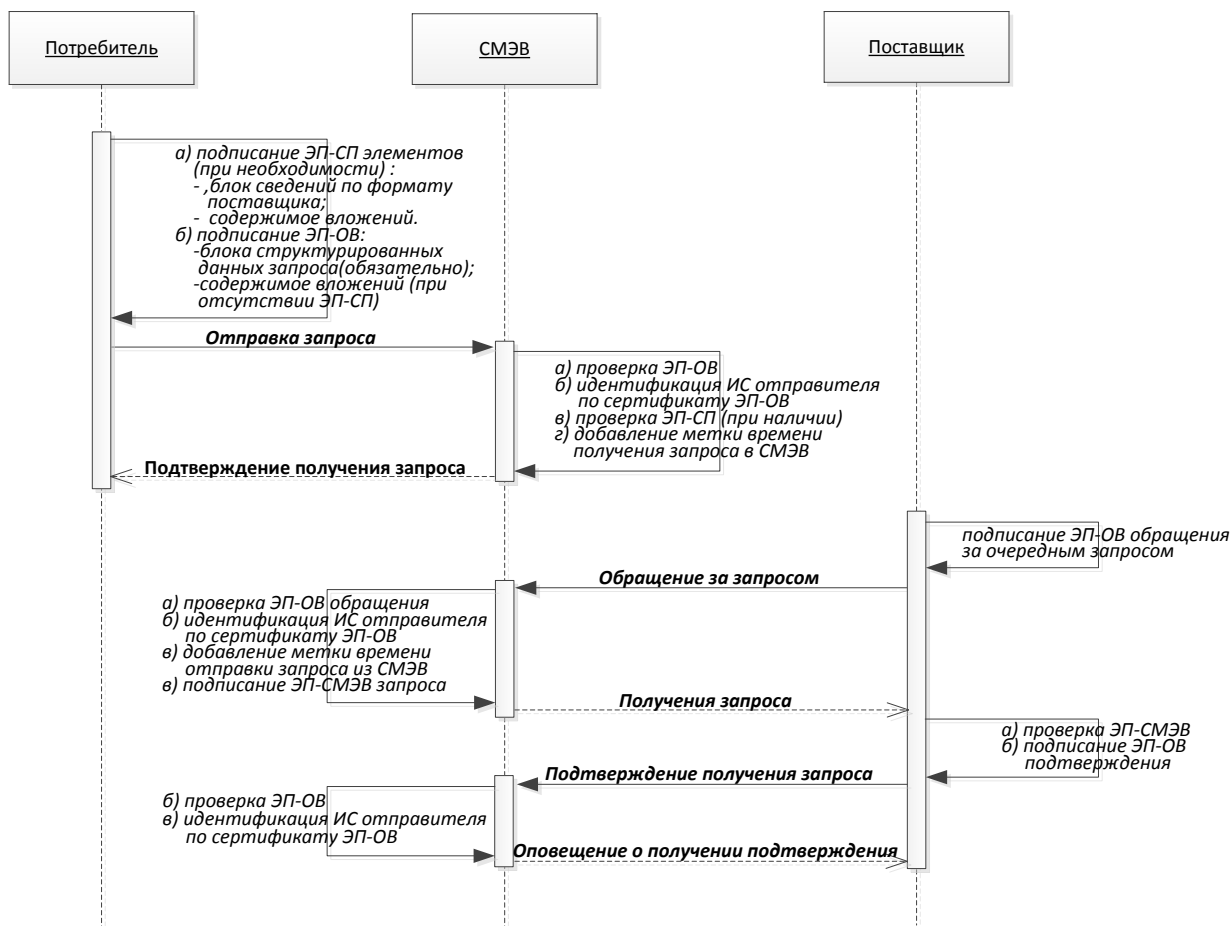


Рисунок 17 – Использование электронных подписей при передаче запроса от потребителя к поставщику сервиса

Если содержимое вложений (файлы, размещенные в элементе //AttachmentContentList) с помощью ЭП-СП должностным лицом не подписываются, то содержимое вложений вместо ЭП-СП должно быть подписано с помощью ЭП-ОВ, которая, в свою очередь, помещается в блок заголовков и ЭП-СП вложений, вместо ЭП-СП вложений.

Перед подписанием запроса с помощью ЭП-СП должна осуществляться проверка наличия и действительности у должностного лица ОВ его сертификата. Ответственным за легитимность использования ЭП-СП является участник взаимодействия, отправляющий электронное сообщение.

Сформированные и подписанные, при необходимости, электронной подписью ЭП-СП сведения, заполненные в соответствии с требованиями поставщика, дополняются служебной информацией и вместе образуют блок данных запроса (элемент //SenderProvidedRequestData). Этот блок данных запроса подписывается ЭП-ОВ (элемент //CallerInformationSystemSignature).

На этом формирование электронных подписей запроса на стороне ИС потребителя завершается. Запрос, подписанный ЭП-ОВ и, при необходимости, ЭП-СП, поступает в СМЭВ.

СМЭВ автоматически осуществляет:

- проверку ЭП-ОВ, в том числе входящего в состав ЭП-ОВ сертификата;
- идентификацию ИС отправителя запроса по сертификату ЭП-ОВ;
- проверку по реестру прав доступа СМЭВ (далее – матрица доступа) возможности обращения ИС отправителя к ИС получателя электронного сообщения;
- добавление блока маршрутной информации (в том числе метки времени получения запроса в СМЭВ).

Для получения из СМЭВ запроса поставщик готовит обращение за очередным запросом и подписывает его ЭП-ОВ.

Получив от поставщика такое обращение, СМЭВ автоматически осуществляет:

- проверку ЭП-ОВ, в том числе входящего в состав ЭП-ОВ сертификата;
- идентификацию ИС, обратившейся за получением запроса, по сертификату ЭП-ОВ;
- проверку по матрице доступа возможности получения этой ИС электронного сообщения;
- добавление метки времени отправки запроса из СМЭВ и подписание запроса с помощью ЭП-СМЭВ.

Получив из СМЭВ сообщение с запросом, ИС поставщика проверяет сертификат и корректность формирования ЭП-СМЭВ. Успешность проверки гарантирует:

- поступление запроса из СМЭВ, а не из другого источника;
- поступление запроса в СМЭВ от ИС отправителя и из СМЭВ в ИС получателя во время, указанное в метках времени;
- право на обращение ИС отправителя к ИС получателя запроса;
- целостность запроса на всем маршруте от ИС отправителя до ИС получателя.

ИС поставщика может также проверить сертификат и корректность формирования ЭП-ОВ в запросе. Такая проверка избыточна, но в случае разбора инцидентов может быть полезна.

ИС поставщика может также проверить сертификат и корректность формирования ЭП-СП должностного лица ОВ - отправителя.

Получив запрос и выполнив необходимые проверки, поставщик должен подтвердить получение запроса. Для этого ИС поставщика готовит подтверждение получения запроса и подписывает его ЭП-ОВ. СМЭВ, получив подтверждение, проверяет ЭП-ОВ, которой подписано подтверждение, и по сертификату ЭП-ОВ идентифицирует ИС-отправителя сообщения. В случае успешной идентификации, СМЭВ по идентификатору сообщения определяет запрос, получение которого подтверждено, и выводит его из обработки.

4.2.2 Использование электронных подписей при передаче ответа

Формирование и подписание с помощью ЭП ответов на запросы (Рисунок 18) выполняется подобно формированию и подписанию с помощью ЭП запросов.

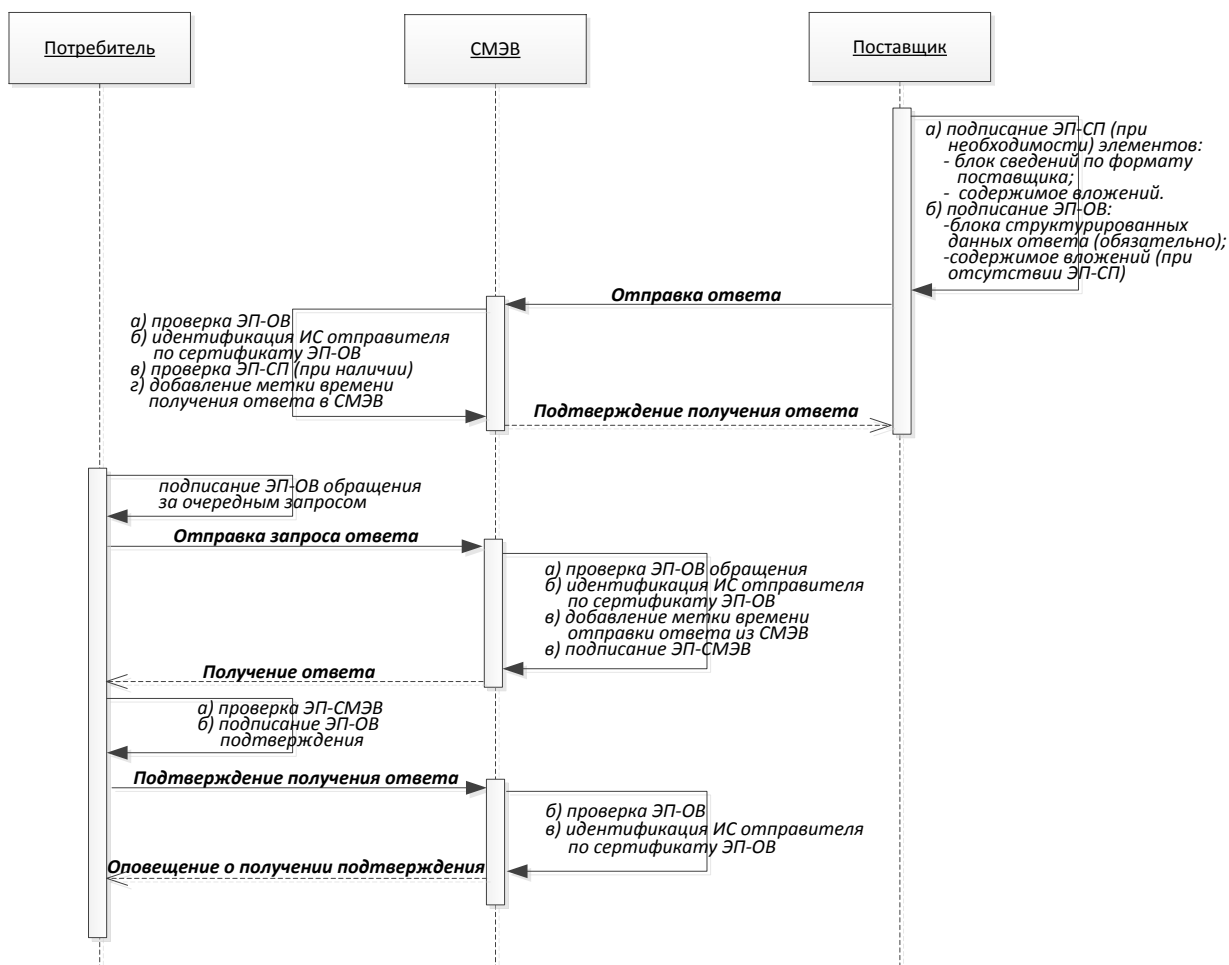


Рисунок 18 – Использование электронных подписей при передаче ответа от поставщика к потребителю

В отличие от формирования запроса, при подготовке и отправке ответа инициатором выступает уже не потребитель, а поставщик. Порядок подписания с помощью ЭП-СП сведений должностным лицом ОВ-поставщика такой же, как и в случае подписания ЭП-СП сведений в запросе. Подписание с помощью ЭП-ОВ блока структурированных данных ответа поставщиком отличается только структурой подписываемого блока структурированных данных ответа (рис. 14). Структура данных, которые добавляются к ответу в СМЭВ и, затем вместе с подписанным с помощью ЭП-ОВ блоком данных, подписываются в СМЭВ ЭП-СМЭВ, также имеет отличия от соответствующей структуры данных, которые добавляются в СМЭВ к запросу. К запросу СМЭВ добавляет элемент //ReplyTo, выполняющий функции обратного адреса, а к ответу добавляет элемент //OriginalMessageId, в который записывает идентификатор запроса, в ответ на который сформирован данный ответ.

Порядок подготовки потребителем подтверждения получения ответа, подписания его ЭП-ОВ и отправки подписанного подтверждения в СМЭВ аналогичен соответствующим действиям при подтверждении получения запроса поставщиком.

4.3. ПРАВИЛА ФОРМИРОВАНИЯ ЭП

При формировании ЭП всех видов должны использоваться алгоритмы, представленные в таблице ниже (Таблица 3).

Таблица 3 – Алгоритмы

	Наименование	URI
Расчет хеш-суммы	ГОСТ Р 34.11-94	http://www.w3.org/2001/04/xmldsig-more#gostr3411
Формирование подписи	ГОСТ Р 34.10-2001	http://www.w3.org/2001/04/xmldsig-more#gostr34102001-gostr3411
Канонизация (для XMLDSig)	Exclusive XML Canonicalization от 18 июля 2002	http://www.w3.org/2001/10/xml-exc-c14n#
Дополнительная трансформация (для XMLDSig)	Нормализация СМЭВ	urn://smev-gov-ru/xmldsig/transform

Далее по тексту этого раздела, если имя элемента указано без пространства имен, подразумевается пространство имен <urn://x-artefacts-smev-gov-ru/services/message-exchange/types/1.1>.

4.3.1 Подписи в формате PKCS#7

Формат PKCS#7 используется для подписания файлов, вложенных в сообщения.

Используется версия 1.5 спецификации PKCS#7 (RFC-2315).

На формат подписи накладываются следующие ограничения:

- Для корневого элемента ContentInfo единственный допустимый contentType - SignedData.
- Подпись должна быть detached (т.е. для элемента SignedData/contentInfo/contentType единственное допустимое значение - 1.2.840.113549.1.7.1, а элемент SignedData/contentInfo/content должен отсутствовать).
- Для вычисления message digest разрешён только алгоритм ГОСТ 34.11-94.
- Для генерации ЭП разрешён только алгоритм ГОСТ 34.10-2001.
- Разрешено применять только X-509 сертификаты. Сертификаты PKCS#6 запрещены.
- Запрещено размещать более одной ЭП в PKCS#7-криптосообщении.
- В элементе SignerInfo должны присутствовать следующие authenticated attributes:
 - contentType (1.2.840.113549.1.9.3), всегда имеет значение 1.2.840.113549.1.7.1.

- messageDigest (1.2.840.113549.1.9.4), содержит ГОСТ-digest подписываемого файла.

Более формально большая часть данных ограничений описана в профиле формата PKCS#7, приложение 3. В профиле также отражён тот факт, что в данном контексте формат PKCS#7 используется только для передачи ЭП и не используется для передачи зашифрованных данных и CRL. Профиль использует типы, определённые в стандарте PKCS#9 (RFC-2985).

4.4. ЭЛЕКТРОННЫЕ ПОДПИСИ СУБЪЕКТОВ ВЗАИМОДЕЙСТВИЯ – ФИЗИЧЕСКИХ ЛИЦ

4.4.1 Общие требования к электронной подписи, формируемой от имени должностных лиц органов власти при межведомственном информационном обмене

Сертификаты и ключи электронной подписи (п. 3 ст. 14 Федерального закона № 63-ФЗ «Об электронной подписи») должностного лица выдаются на имя физического лица представителя органа власти и применяются в информационных системах при оказании государственных и муниципальных услуг/исполнении государственных и муниципальных функций с использованием системы межведомственного электронного взаимодействия для формирования и (или) проверки электронных подписей.

Данные подписи аналогичны собственноручным подписям этих сотрудников и подтверждают, в том числе, факт формирования электронного документа конкретным сотрудником ОВ в ИС ОВ.

Ответственность за хранение и использование ключа подписи ЭП-СП несет должностное лицо. Порядок хранения и использования ключа подписи ЭП-СП контролируется представителями органов власти.

Перевыпуск существующих сертификатов ключей ЭП-СП должностных лиц ОВ для использования при межведомственном взаимодействии не является обязательным: возможно использовать ранее выданные и действительные сертификаты ключей подписи должностных лиц при условии, что они выданы одним из аккредитованных удостоверяющих центров, входящих в единое пространство доверия ЭП, формируемое Минкомсвязью РФ.

4.4.2 Электронная подпись при межведомственном взаимодействии

ЭП-СП подписывает бизнес-данные сообщения, представленные в XML, а также приложенные файлы. Поскольку вложения передаются отдельно от бизнес-данных, ЭП-СП ставится отдельно на бизнес-данные, отдельно на каждый приложенный файл.

4.4.2.1 Правила формирования электронной подписи сообщений

Правила формирования электронной подписи сообщений представлены в таблице ниже (Таблица 4).

Таблица 4 – Правила формирования электронной подписи сообщений

Формат подписи	XMLDSig detached (https://www.w3.org/TR/xmlsig-core/)
Трансформация, дополнительно к канонизации	urn://smev-gov-ru/xmlsig/transform
Требования к форматированию	В XML-структуре подписи между элементами не допускается наличие текстовых узлов, в том числе переводов строки.
Подписываемый элемент	Для запросов и ответов - корневой элемент XML-документа, представляющего бизнес-данные запроса или ответа.
Размещение в сообщении	//SenderProvidedRequestData/ PersonalSignature/dsig:Signature (для запросов), //SenderProvidedResponseData/PersonalSignature/dsig:Signature (для ответов),
Способ помещения подписи в сообщение	Передается клиентом веб-сервиса в структуре параметров методов SendRequest, SendResponse.
Способ извлечения подписи для проверки	ЭП извлекается и проверяется клиентом веб-сервиса .

4.4.2.2 Правила формирования электронной подписи вложений

Правила формирования электронной подписи вложений представлены в таблице ниже (Таблица 5).

Таблица 5 – Правила формирования электронной подписи вложений

Формат подписи	PKCS#7
Ограничения на использование формата	Описаны в разделе « Подписи в формате PKCS#7 »
Способ помещения подписи в сообщение	Передается клиентом веб-сервиса в структуре параметров методов SendRequest, SendResponse.
Способ извлечения	Подписи находятся в элементах

подписи для проверки	//AttachmentHeaderList/AttachmentHeader/SignaturePKCS7 входящих сообщений.
----------------------	---

4.5. ЭЛЕКТРОННЫЕ ПОДПИСИ СУБЪЕКТОВ ВЗАИМОДЕЙСТВИЯ – ИНФОРМАЦИОННЫХ СИСТЕМ

4.5.1 Общие требования электронной подписи, формируемой от имени органа власти при межведомственном информационном обмене

Сертификаты и ключи электронной подписи (п. 3 ст. 14 Федерального закона от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи»), используемые для формирования электронных подписей органа власти выдаются на имя органа власти и применяются в информационных системах при оказании государственных и муниципальных услуг/исполнении государственных и муниципальных функций с использованием СМЭВ для формирования ЭП.

ЭП-ОВ аналогичны гербовой печати организации и подтверждают:

- факт формирования межведомственного запроса в информационной системе ОВ, подписавшего межведомственный запрос (далее – запрос);
- факт наличия у лица, сформировавшего в ИС ОВ электронный документ (запрос либо ответ), соответствующих полномочий по подписанию/проверке ЭП на момент формирования электронного документа.

Орган власти, отправляющий электронный документ с использованием СМЭВ другому участнику взаимодействия, гарантирует наличие соответствующих полномочий у своего должностного лица на обращение к информационному ресурсу другого ведомства либо на подготовку ответа на поступивший запрос (в случае если ответ формируется не автоматически в ИС).

Количество формируемых на ОВ сертификатов ЭП-ОВ не может быть меньше количества информационных систем данного ОВ, непосредственно подключенных к СМЭВ.

Ответственность за хранение и использование ключа подписи ЭП-ОВ несет должностное лицо. Порядок хранения и использования ключа подписи ЭП-ОВ контролируется представителями органов власти.

4.5.2 Общие требования к электронной подписи, формируемой узлами СМЭВ

Сертификаты и ключи электронной подписи (п. 3 ст. 14 Федерального закона от 6 апреля 2011 г. № 63-ФЗ «Об электронной подписи»), используемые для формирования электронных подписей в сообщениях, проходящих через федеральный и региональные узлы СМЭВ, выдаются на имя оператора соответствующей системы межведомственного электронного взаимодействия и применяются для формирования ЭП.

Общие требования к электронной подписи, формируемой узлами СМЭВ, представлены в таблице ниже (Таблица 6).

ЭП-СМЭВ подтверждает:

- факт прохождения электронного сообщения через СМЭВ;
- факт аутентификации и авторизации в соответствии с правилами, указанными в реестре прав доступа к электронным сервисам (матрице доступа);
- неизменность сведений, внесенных в электронное сообщение СМЭВ.

Ответственность за хранение и использование ключа подписи ЭП-СМЭВ обеспечивается организационно-техническими мероприятиями оператора СМЭВ.

Таблица 6 – Общие требования к электронной подписи, формируемой узлами СМЭВ

Формат подписи	XMLDSig detached
Трансформация, дополнительно к канонизации	urn://smev-gov-ru/xmldsig/transform
Требования к форматированию	В XML-структуре подписи, между элементами не допускается наличие текстовых узлов, в том числе переводов строки.
Подписываемый элемент	<ul style="list-style-type: none"> - Для запросов – элемент //SendRequestResponse - Для ответов – элемент //MessageMetadata - При выборке сообщения из очереди – элемент //Request - При подтверждении получения сообщения – ЭП СМЭВ отсутствует.
Размещение во входящем сообщении	Тело SOAP конверта, элемент //CallerInformationSystemSignature

4.5.3 Правила формирования электронной подписи информационной системы

Общие требования к электронной подписи, формируемой узлами СМЭВ представлены в таблице ниже (Таблица 7).

Таблица 7 – Правила формирования электронной подписи информационной системы

Формат подписи	XMLDSig detached
Трансформация, дополнительно к канонизации	urn://smev-gov-ru/xmldsig/transform
Требования к форматированию	В XML-структуре подписи, между элементами не допускается наличие текстовых узлов, в том числе переводов строки.

Подписываемый элемент	<ul style="list-style-type: none"> - Для запросов – элемент //SenderProvidedRequestData - Для ответов – элемент //SenderProvidedResponseData - При выборке сообщения из очереди – элемент //MessageTypeSelector - При подтверждении получения сообщения – элемент //AckTargetMessage
Размещение в исходящем сообщении	Элемент //CallerInformationSystemSignature, см. схему sme-v-message-exchange-types-1.1.0.xsd.
Размещение во входящем сообщении	ЭП-ОВ отправителя попадает к получателю только при вызове методов GetRequest, GetResponse (выборка сообщения из очереди). Она находится в теле SOAP-конверта, элемент //SenderInformationSystemSignature.

4.5.4 Подписание вложений электронной подписью информационной системы

В случае если сообщение содержит вложения и какие-либо из них не подписаны ЭП-СП, информационная система должна перед отправкой сообщения подписать такие вложения ЭП-ОВ. Это необходимо для защиты от подмены вложений.

Подпись формируется по тем же правилам, что и ЭП-СП (Таблица 8).

Таблица 8 – Правила формирования ЭП-ОВ

Формат подписи	PKCS#7
Ограничения на использование формата	Описаны в разделе « Подписи в формате PKCS#7 »
Способ помещения подписи в сообщение	Передается клиентом веб-сервиса в структуре параметров методов SendRequest, SendResponse.
Способ извлечения подписи для проверки	Подписи находятся в элементах //AttachmentHeaderList/AttachmentHeader/SignaturePKCS7 входящих сообщений.

5. ПЕРЕСЫЛКА ВЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ФАЙЛОВОГО ХРАНИЛИЩА

В СМЭВ имеется возможность передачи файлов вложений отдельно от сообщения. Для этого используется Файловое хранилище СМЭВ. Использование Файлового хранилища обязательно, если суммарный объем вложений сообщения превышает 5 Мб. При этом суммарный объем файлов сообщения не должен превышать 1 Гб.

Загрузка файлов в Файловое хранилище осуществляется по протоколу FTP. Каждый участник взаимодействия получает доступ к отдельной директории FTP-сервера Файлового хранилища для загрузки файлов вложений. Для каждого файла ИС отправителя должна создать отдельную директорию, в качестве названия которой должен быть использован UUID, сгенерированный по алгоритму, аналогичному генерации UUID сообщения (см. п. 4.2 «Algorithms for Creating a Time-Based UUID» RFC 4122 <http://rfc.askapache.com/rfc4122/rfc4122.html#section-4.2>).

Общий процесс передачи файлов посредством Файлового хранилища представлен на рисунке 19.

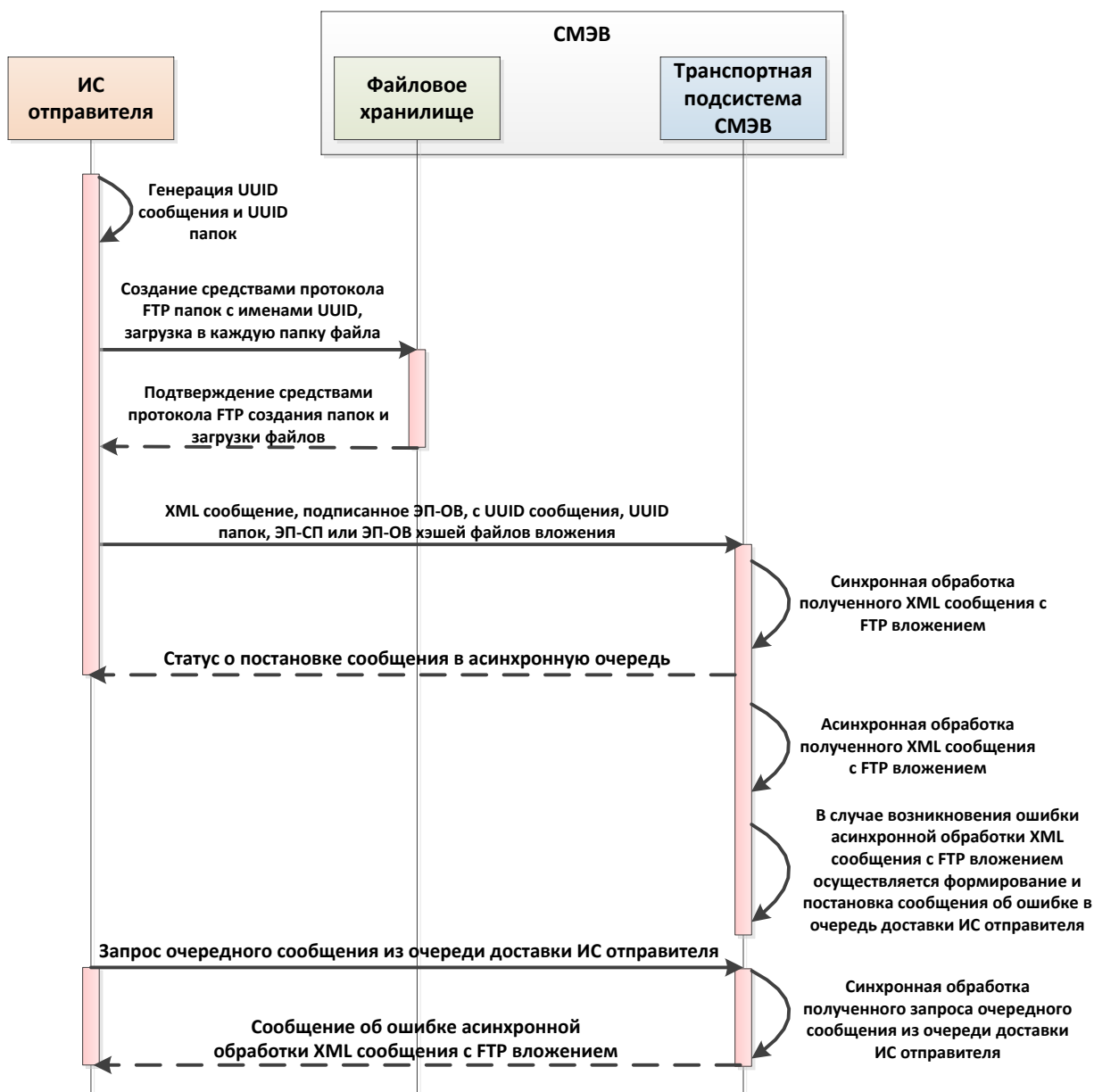


Рисунок 19 – Диаграмма последовательности отправки файлов посредством Файлового хранилища

При отправке сообщения, которому принадлежат загруженные файлы, UUID созданных папок с файлами указываются в сообщении в соответствующих тегах SenderProvidedRequestData (для запроса) и SenderProvidedResponseData (для ответа). Данные теги включают элемент RefAttachmentHeaderList, который описывается как лист значений.

Очистка выделенной для информационной системы отдельной директории FTP-сервера Файлового хранилища производится автоматически в ходе обработки отправленного сообщения с файлами.

Общий процесс получения сообщения с файлами в Файловом хранилище представлен на рисунке 20.

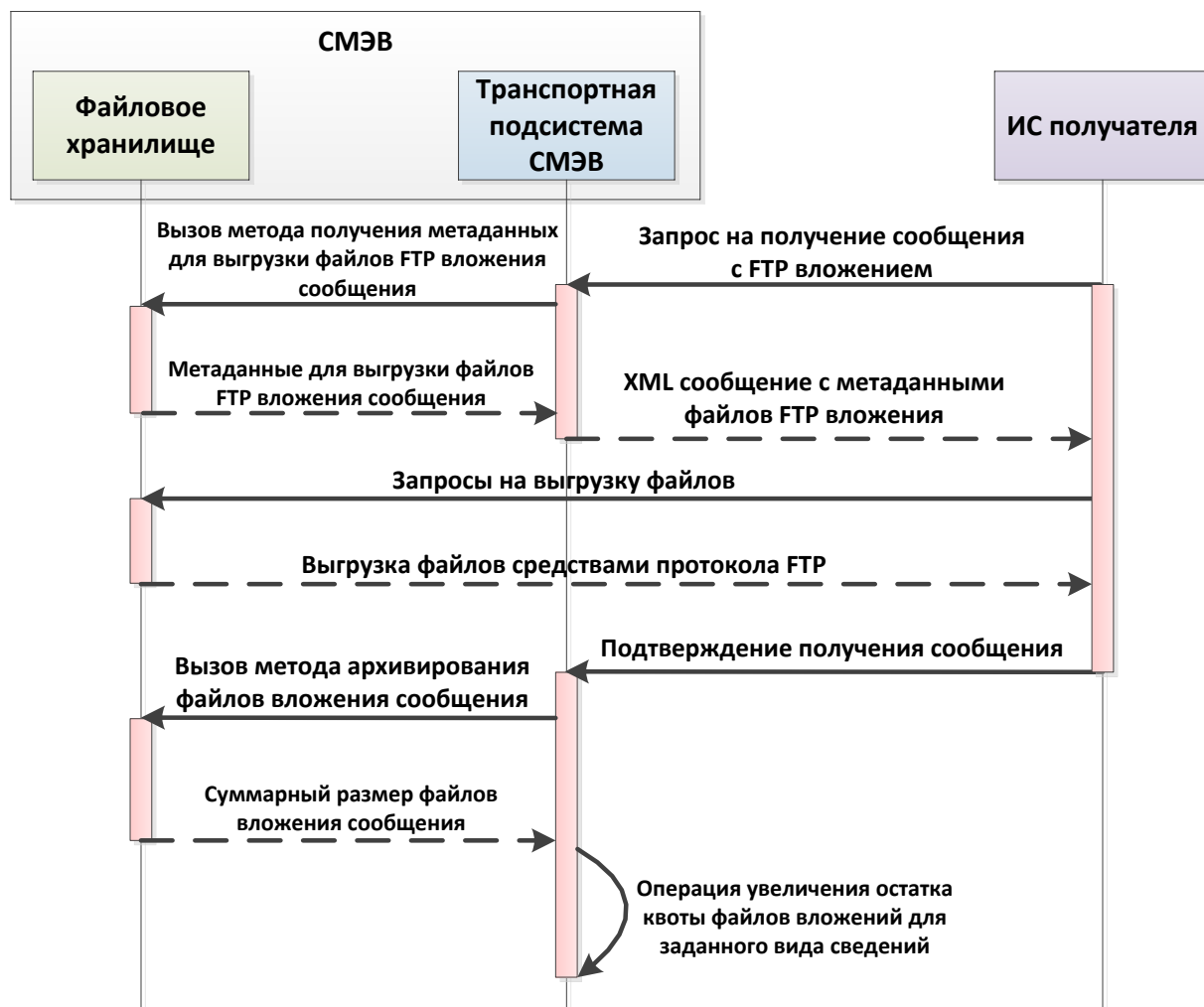


Рисунок 20 – Диаграмма последовательности получения файлов посредством Файлового хранилища

В составе входящего сообщения содержатся ссылки на пришедшие файлы - тег FSAttachmentsLis, представляющий собой лист элементов FSAttachment значений типа FSAuthInfo, содержащих ссылку на файл (uuid), логин (UserName), пароль (Password), имя файла (FileName). Для выгрузки файла на стороне информационной системы получателя необходимо сформировать запрос вида: ftp://логин:пароль@ip-адрес:порт/UUID_файла/имя_файла, где «ip-адрес:порт» - адрес Файлового хранилища СМЭВ3.x

После доставки сообщения и при получении от информационной системы получателя подтверждения о получении сообщения СМЭВ3.x очищает область доставки Файлового хранилища, доступ к доставленным файлам закрывается.

Структурная схема взаимодействия Файлового хранилища со СМЭВ3.x в рамках передачи вложения средствами протокола FTP приведена на рисунке 21.

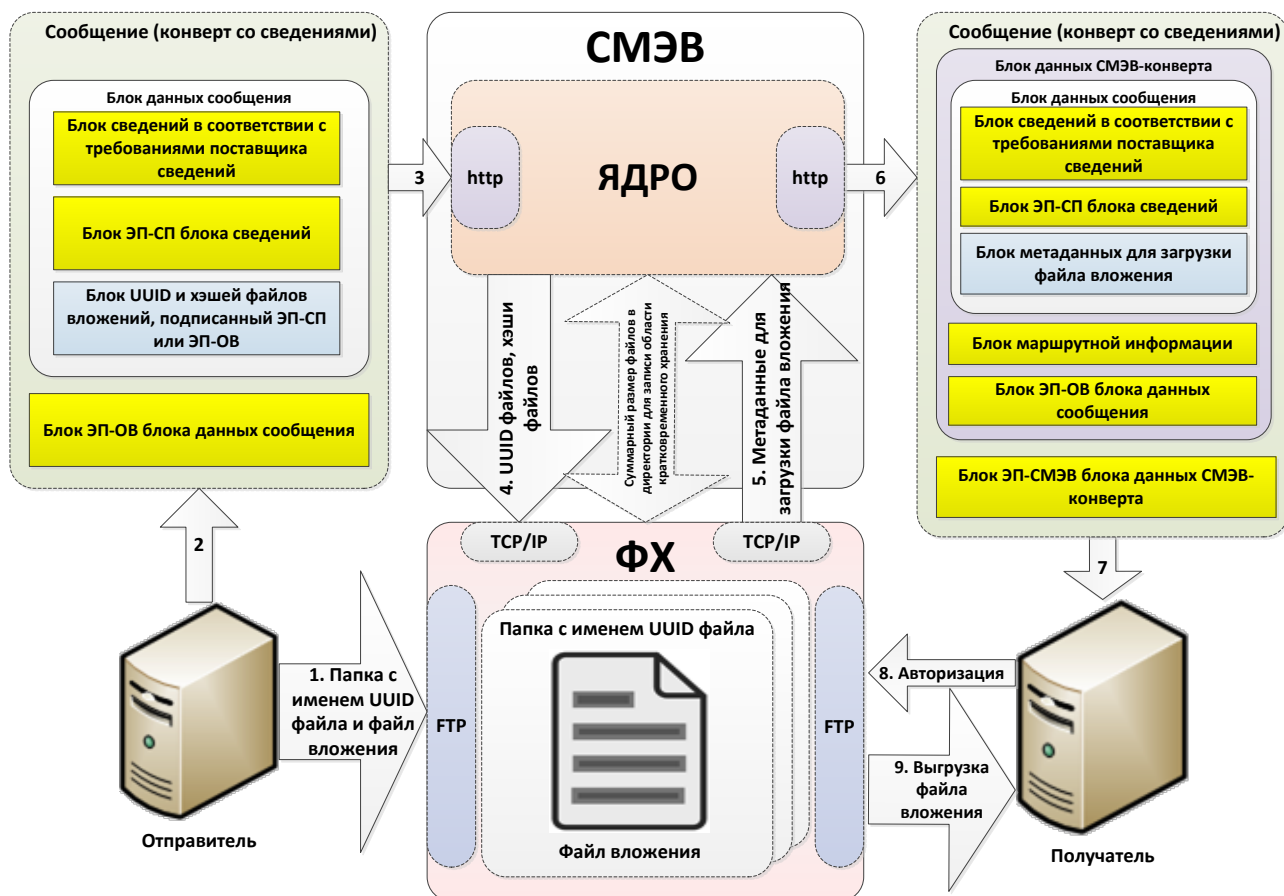


Рисунок 21 – Структурная схема взаимодействия Файлового хранилища со СМЭВ в рамках передачи вложения средствами протокола FTP

6. СЦЕНАРИИ АСИНХРОННОГО ВЗАИМОДЕЙСТВИЯ

6.1. МЕЖВЕДОМСТВЕННЫЙ ЗАПРОС

Упрощенно типовой сценарий межведомственного взаимодействия включает одно сообщение – запрос и одно сообщение – ответ (Рисунок 22).

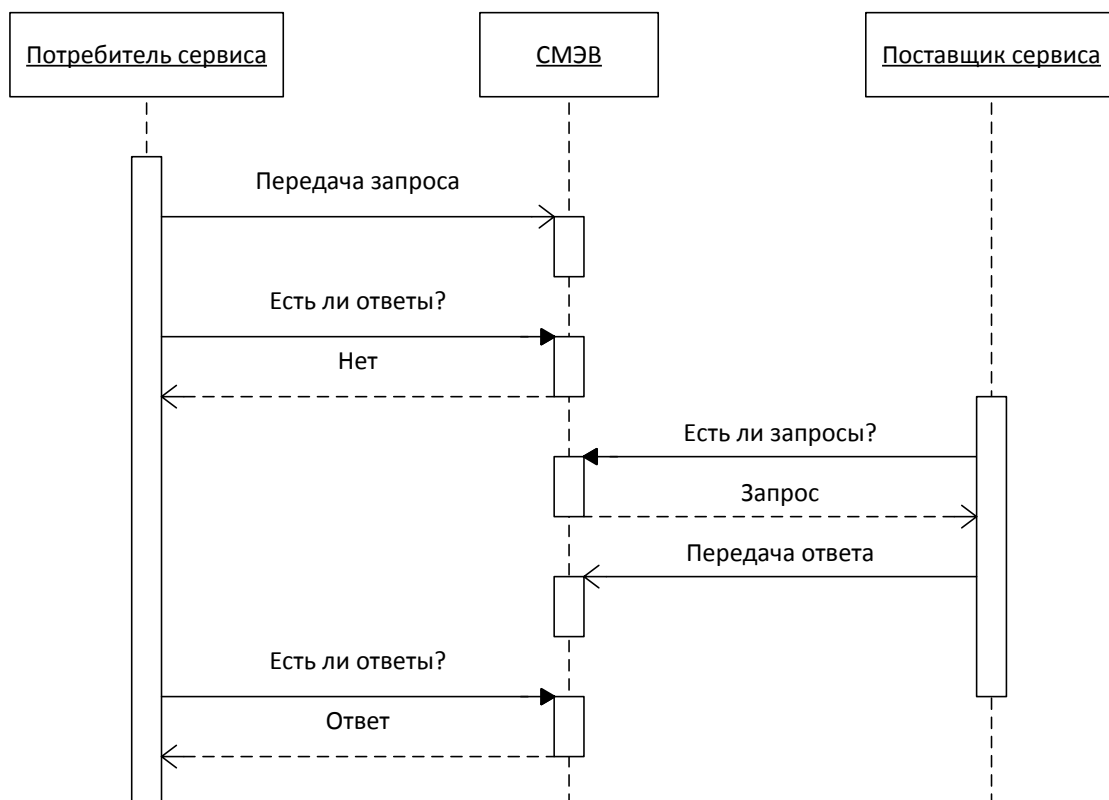


Рисунок 22 – Типовой сценарий межведомственного взаимодействия (упрощенно)

Обмен сообщениями между ИС потребителя и ИС поставщика, реализуемый в СМЭВ, осуществляется путем вызова соответствующих методов веб-сервиса **SMEVMessageExchangeService**, предоставляемого СМЭВ. Веб-сервис **SMEVMessageExchangeService** предоставляет восемь методов.

Пять методов используются для передачи запроса от ИС потребителя к ИС поставщика и ответа от ИС поставщика к ИС потребителя:

- **SendRequest** (послать запрос), служит для передачи запроса от ИС потребителя в СМЭВ;
- **GetRequest** (получить запрос), служит для получения запроса ИС поставщика из СМЭВ;
- **Ack** (подтвердить получение), служит для подтверждения получения сообщения из очереди, должен вызываться после получения сообщения методами **GetRequest** или **GetResponse**;

- **SendResponse** (послать ответ), служит для передачи ответа на запрос от ИС поставщика в СМЭВ;
- **GetResponse** (получить ответ), служит для получения из СМЭВ ответа на запрос от ИС потребителя.

На протяжении жизненного цикла запрос (ответ на запрос) проходит ряд состояний (статусов).

Далее на диаграмме (Рисунок 23) представлена последовательность обращений к веб-сервису СМЭВ `urn://x-artefacts-smev-gov-ru/services/message-exchange/1.1` (обращения к веб-сервису выделены полужирным шрифтом) при передаче запроса от ИС потребителя к ИС поставщика и ответа от ИС поставщика к ИС потребителя. На диаграмме также показаны наиболее важные действия, которые выполняются СМЭВ, ИС поставщика и ИС потребителя в промежутках между обращениями к веб-сервису СМЭВ.

Перед отправкой в СМЭВ запроса сведений ИС потребителя должна подготовить этот запрос. Подготовка запроса включает корректное заполнение блока структурированных данных запроса `//SenderProvidedRequestData`, в том числе блока сведений по форматам поставщика `//MessagePrimaryContent` (правильность заполнения элемента `//MessagePrimaryContent` будет потом проверяться в СМЭВ на соответствие схеме XSD и, при наличии, Schematron, разработанными поставщиком), добавление ЭП-ОВ для элемента `//SenderProvidedRequestData` и, при необходимости, добавление вложений (`//AttachmentContentList` и `//AttachmentHeaderList`).

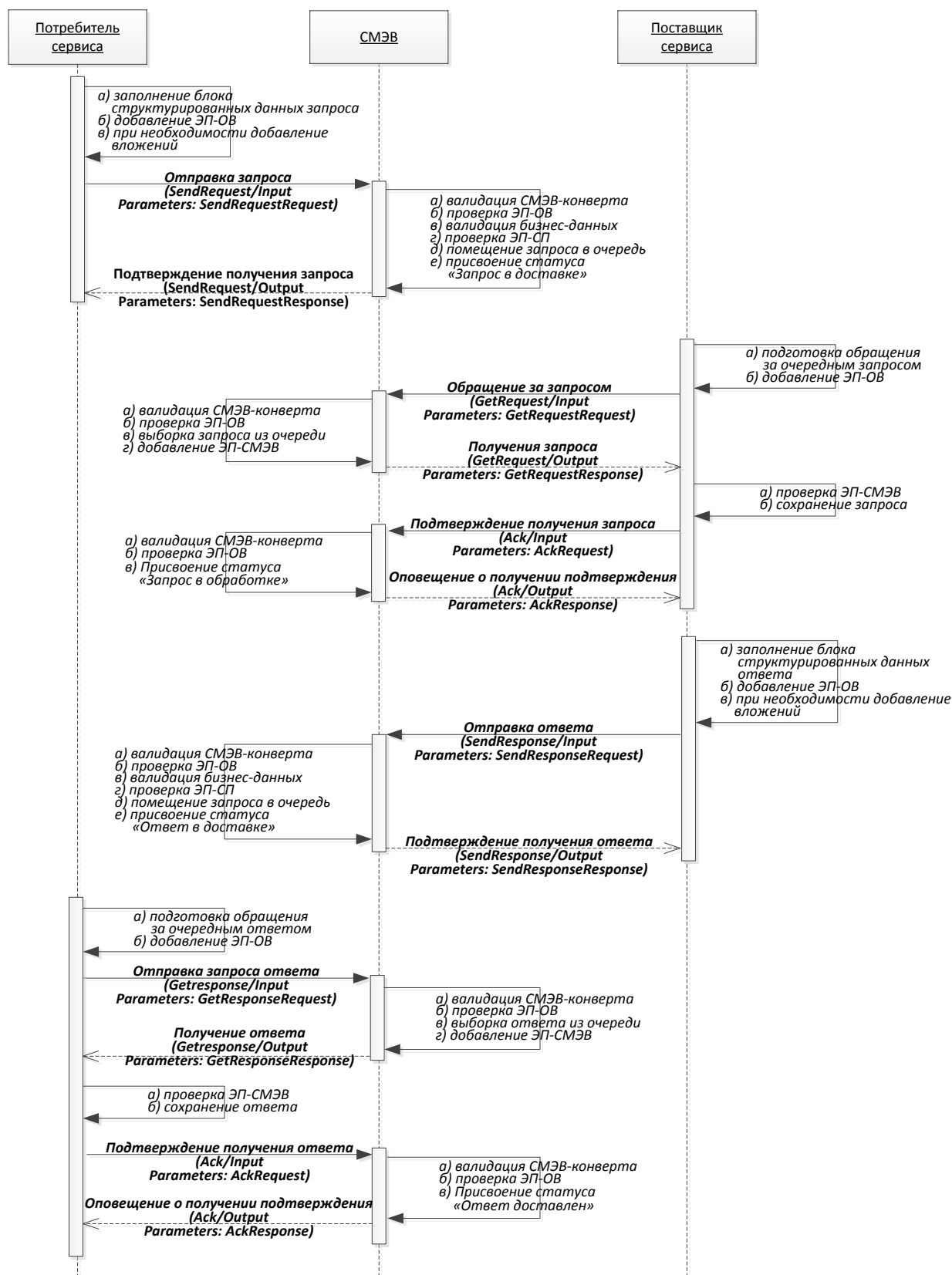


Рисунок 23 – Последовательность обращений к веб-сервису СМЭВ при передаче сообщений с запросами и ответами

Затем запрос сведений передается в СМЭВ с помощью метода `SendRequest`, в СМЭВ последовательно выполняются следующие операции:

- **форматно-логический контроль (далее - ФЛК) СМЭВ-конверта** по схеме XSD. Под ФЛК понимается проверка формата данных, а также контроль логики заполнения данных, осуществляемые путем проверки соответствия этих данных документам на языке XSD и, при необходимости, Schematron (пример проверки: срок лишения специального права не может быть менее одного месяца и более трех лет). Как синоним ФЛК, в указанном значении, далее используется также термин **валидация**;
- **проверка ЭП-ОВ** на предмет корректности и на предмет действительности соответствующих сертификатов ключей подписи. ЭП-ОВ также используется для идентификации потребителя сервиса, приславшего запрос;
- **валидация бизнес-данных** по схеме XSD и, при наличии, Schematron, разработанными поставщиком сервиса. Также проверяется полное имя корневого элемента блока структурированных сведений `//MessagePrimaryContent` для идентификации ИС поставщика - получателя запроса;
- **проверка ЭП-СП** (в элементе `//PersonalSignature` и в блоке заголовков вложений `//AttachmentHeaderList`);
- **помещение запроса в очередь запросов.**

Запрос будет находиться в очереди запросов до тех пор, пока при очередном обращении в СМЭВ его не получит ИС поставщика. Для получения запроса ИС поставщика подготавливает и подписывает ЭП-ОВ обращение за запросом, а затем, вызвав метод `GetRequest`, передает это обращение в СМЭВ. СМЭВ по ЭП-ОВ идентифицирует ИС поставщика и, при наличии недоставленных запросов, возвращает в ИС поставщика очередной запрос, предварительно подписав его ЭП-СМЭВ.

Получив из СМЭВ запрос, ИС поставщика проверяет ЭП-СМЭВ и, в случае успешной проверки, сохраняет у себя этот запрос, а в СМЭВ передает подтверждение получения запроса путем вызова метода `Ask`. СМЭВ, получив от ИС поставщика подтверждение получения запроса, снимает его с обработки, устанавливая ему внутренний признак «Обработан».

ИС поставщика, в свою очередь, готовит ответ на полученный запрос и, подписав его ЭП-ОВ, отправляет в СМЭВ путем вызова метода `SendResponse`. СМЭВ, получив ответ от ИС поставщика, выполняет с сообщением действия, аналогичные действиям при получении запроса от ИС потребителя, и помещает ответ в очередь ответов.

Затем ИС потребителя вызывает метод `GetResponse` и передает в СМЭВ подготовленный и подписанный ЭП-ОВ запрос очередного ответа. СМЭВ по ЭП-ОВ идентифицирует ИС потребителя и, при наличии недоставленных ответов, возвращает в ИС потребителя очередной ответ, предварительно подписав его ЭП-СМЭВ. Так же как и ИС поставщика при получении запроса, ИС потребителя при получении ответа проверяет ЭП-СМЭВ, сохраняет у себя этот ответ и подтверждает получение ответа вызовом метода

Ask. СМЭВ, получив от ИС потребителя подтверждение получения ответа, присваивает ответу внутренний признак «Обработан».

Следует также заметить, что все значимые события при обращении потребителя или поставщика в СМЭВ, от получения SOAP-запроса до отправки SOAP-ответа, фиксируются в журнале СМЭВ.

6.2. ПАКЕТ ЗАПРОСОВ – ПАКЕТ ОТВЕТОВ

Если поставщик вида сведений желает получать запросы в пакетах, он описывает это в своей прикладной схеме. Например, бизнес-данные запроса описываются бизнес-сущностью `ЗапросВыпискиИзКакогоТоРеестра`. Поставщик вида сведений выпускает новую версию XML-схемы своего вида сведений, в которой вводится бизнес-сущность `ПакетЗапросовВыписокИзКакогоТоРеестра`, которая содержит список сущностей `ЗапросВыпискиИзКакогоТоРеестра`. Последовательность действий при обмене данными:

1. Потребитель вида сведений накапливает запросы в течение N-го промежутка времени.
2. Потребитель вида сведений из имеющихся запросов формирует пакет запросов, согласно прикладной схеме поставщика вида сведений.
3. Потребитель вида сведений посылает пакет запросов поставщику одним сообщением СМЭВ.
4. Поставщик получает пакет запросов, генерирует пакет ответов и отправляет потребителю одним сообщением СМЭВ в соответствии со схемой своего вида сведений.
5. Потребитель вида сведений получает пакет ответов, разбивает его на одиночные ответы.

Пример схемы:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="urn://x-artefacts-smev-ru-examples/split-response/1.0"
  targetNamespace="urn://x-artefacts-smev-ru-examples/split-response/1.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:annotation>
    <xs:documentation>
      Пример прикладной схемы, позволяющей отправлять запросы в пачках,
      и таким же образом принимать ответы.
    </xs:documentation>
  </xs:annotation>

  <xs:complexType name="GetPersonNameBySNILSRequestType">
    <xs:annotation>
      <xs:documentation>Запрос на получение ФИО человека по СНИЛС.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="SNILS" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
```

```

<xs:complexType name="GetPersonNameBySNILSResponseType">
  <xs:annotation>
    <xs:documentation>
      Ответ на запрос на получение ФИО человека по СНИЛС.
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element name="NotFound" type="tns:Void"/>
    <xs:sequence>
      <xs:element name="FamilyName" type="xs:string"/>
      <xs:element name="FirstName" type="xs:string"/>
      <xs:element name="Patronymic" type="xs:string"/>
    </xs:sequence>
  </xs:choice>
</xs:complexType>

<xs:element name="GetPersonNameBySNILSBatchRequest">
  <xs:annotation>
    <xs:documentation>
      Пачка запросов на получение ФИО человека по СНИЛС.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="GetPersonNameBySNILSRequest" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="tns:GetPersonNameBySNILSRequestType">
              <xs:attribute name="numberInBatch" type="xs:string">
                <xs:annotation>
                  <xs:documentation>
                    По значению этого атрибута, элементы пачки ответов
                    можно будет связать
                    с соответствующими элементами пачки запросов.
                  </xs:documentation>
                </xs:annotation>
              </xs:attribute>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="GetPersonNameBySNILSBatchResponse">
  <xs:annotation>
    <xs:documentation>
      Пачка ответов на запрос на получение ФИО человека по СНИЛС.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="GetPersonNameBySNILSResponse" maxOccurs="unbounded">
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="tns:GetPersonNameBySNILSResponseType">
              <xs:attribute name="numberInBatch" type="xs:string">
                <xs:annotation>
                  <xs:documentation>
                    По значению этого атрибута, элементы пачки ответов

```

```
        можно будет связать  
        с соответствующими элементами пачки запросов.  
    </xs:documentation>  
  </xs:annotation>  
  </xs:attribute>  
  </xs:extension>  
  </xs:complexContent>  
  </xs:complexType>  
  </xs:element>  
  </xs:sequence>  
  </xs:complexType>  
  </xs:element>  
  
  <xs:complexType name="Void"/>  
  
</xs:schema>
```

7. ВЗАИМОДЕЙСТВИЕ ЧЕРЕЗ СИСТЕМЫ-АГРЕГАТОРЫ

В ряде случаев информационные системы участников взаимодействия могут обмениваться сообщениями через информационные системы-интеграторы, которые предоставляют системам внутренний интерфейс интеграции либо терминальный доступ для ввода запросов сведений.

Для системы-интегратора в настройках СМЭВ выставляется специальный признак. СМЭВ, получая запросы от системы-интегратора, осуществляет проверку доступа информационной системы к запрашиваемому сведению не по мнемонике системы-интегратора, а по коду ФРГУ реальной системы, которая взаимодействует через систему-интегратор.

Код ФРГУ передается в блоке атрибутов бизнес-процесса (`//BusinessProcessMetadata`, внутри тега `«frgu»`) (см. п.3.2.1) и является обязательным для запросов для систем-интеграторов. При этом запросы подписываются ЭП-ОВ системы-интегратора.

Пример вызова:

```
<ns:BusinessProcessMetadata xmlns:tns="urn://x-artefacts-smev-gov-ru/services/message-  
exchange/business-process-metadata/1.0">  
  <tns:frgu>123</tns:frgu>  
</ns:BusinessProcessMetadata>
```

Таким образом, система-интегратор должна взять на себя:

1. Формирование запросов в СМЭВ.
2. Подписание запросов собственной ЭП-ОВ.
3. Простановку в запросе кода ФРГУ реально взаимодействующей системы.
4. Отправку запросов и получение ответов.
5. Распределение ответом между реально взаимодействующими системами.

8. ИСПОЛЬЗОВАНИЕ СИСТЕМЫ ГЕНЕРАЦИИ КОДОВ ТРАНЗАКЦИЙ

Для получения расширенной аналитической отчетности по предоставляемым ведомствами услугам и выполняемым функциям для каждой транзакции предоставления услуги или выполнения функции вводится уникальный код (код транзакции). Генерация данных кодов производится системой генерации кодов транзакций СМЭВ (далее – СГКТ).

Процесс выдачи кодов транзакций осуществляется посредством отдельного Сервиса предоставления кодов транзакций СГКТ (далее – СПКТ) и включает следующие сценарии:

Схема и описание основного сценария приведены на рисунке 24 и таблице 9.

Схемы и описания альтернативных сценариев приведены на рисунках 25, 26 и таблицах 10, 11.

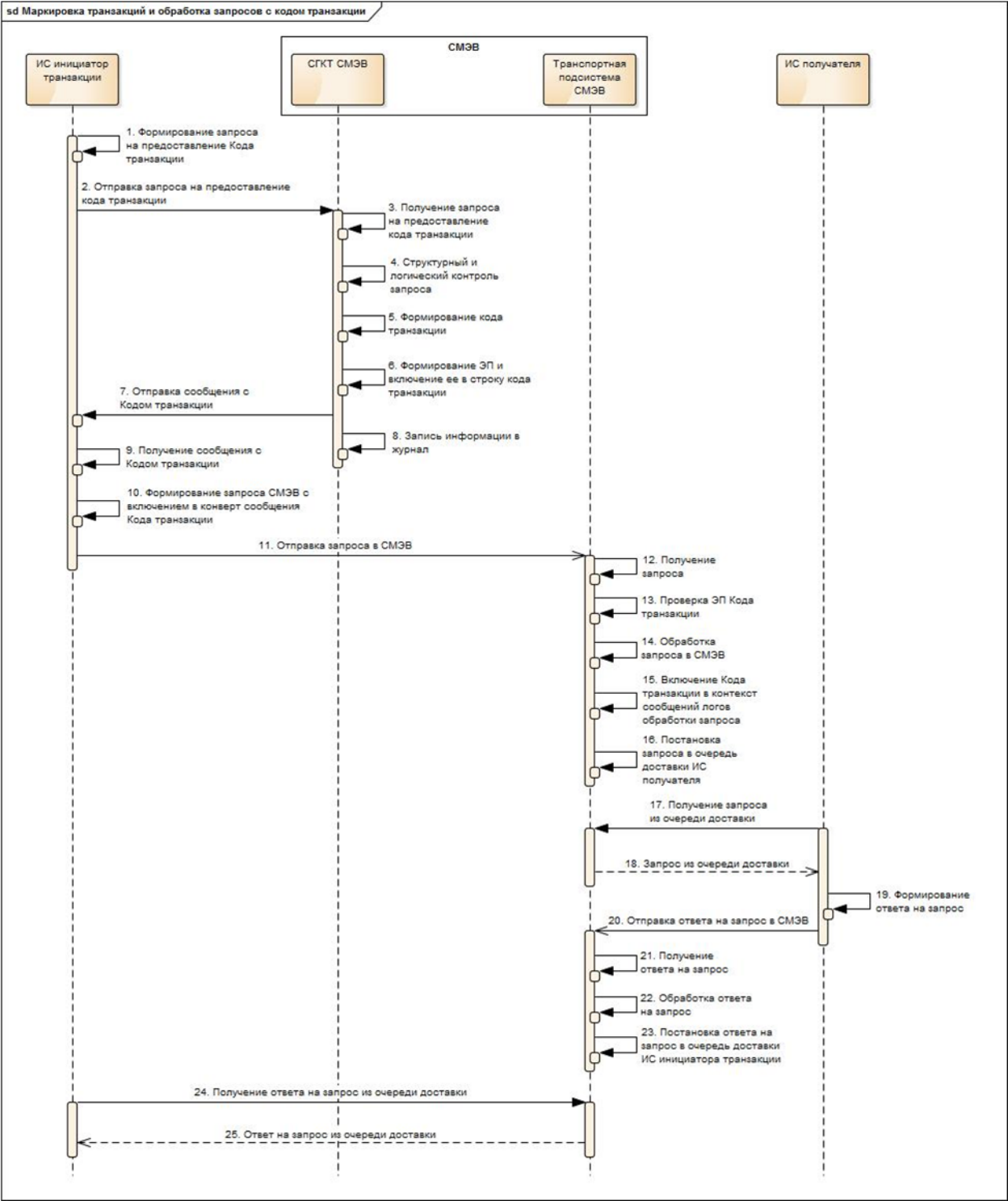


Рисунок 24 – Диаграмма основного сценария процесса транзакции предоставления услуги или выполнения функции

Предусловие	Средствами ИС инициировано предоставление услуги или выполнение функции. ИС, средствами которой инициировано предоставление услуги или выполнение функции, инициирует
-------------	---

		транзакцию предоставления услуги или выполнения функции. ИС, инициирующая транзакцию предоставления услуги или выполнения функции, является ИС инициатором транзакции. ИС инициатор транзакции имеет следующие данные: <ul style="list-style-type: none"> – Мнемоника ИС инициатора транзакции. – Код ФРГУ услуги либо признака функции. 	
Постусловие		Транзакция предоставления услуги или выполнения функции завершена.	
№ шага	Участник	Наименование шага	Описание шага
1	ИС инициатор транзакции	Формирование запроса на предоставление Кода транзакции	Формируется запрос на предоставление кода транзакции, в который включается следующая информация: <ul style="list-style-type: none"> — Мнемоника информационной системы, отправившей запрос (обязательно). — Код ФРГУ услуги, в рамках предоставления которой запрошен Код транзакции, или признак функции (обязательно). — Строка расширенной информации о системе, отправившей запрос (опционально). — Строка расширенной информации о потребителе услуги (опционально).
2	ИС инициатор транзакции	Отправка запроса на предоставление Кода транзакции	Выполняется отправка запроса на предоставление Кода транзакции в систему генерации кодов транзакции
3	СГКТ СМЭВ	Получение запроса на предоставление Кода транзакции	Получение запроса на предоставление Кода транзакции
4	СГКТ СМЭВ	Структурный и логический контроль запроса	Выполняется структурный контроль запроса, а также контроль на наличие и корректность обязательных параметров
5	СГКТ СМЭВ	Формирование Кода транзакции	Выполняется формирование и включение в Код транзакции: <ul style="list-style-type: none"> — универсального уникального идентификатора транзакции; — кода ФРГУ услуги или признака функции, указанного в запросе.
6	СГКТ СМЭВ	Формирование ЭП и включение ее в строку Кода транзакции	Формирование ЭП и включение ЭП в строку Кода транзакции.

7	СГКТ СМЭВ	Отправка сообщения с Кодом транзакции	Отправка сообщения со сформированным кодом транзакции в ИС инициатора транзакции
8	СГКТ СМЭВ	Запись информации в журнал	Запись информации в журнал успешных операций по выдаче транзакций. Записываемая информация включает: <ul style="list-style-type: none"> — универсальный уникальный идентификатор транзакции; — метку времени формирования Кода; — мнемоника информационной системы, запросившей Код транзакции; — код ФРГУ услуги, для выполнения которой запрошен Код, или признак функции; — строку расширенных сведений об услуге или функции (при наличии); — строку расширенных сведений о потребителе услуги (при наличии).
9	ИС инициатор транзакции	Получение сообщения с Кодом транзакции	Получение сообщения со сформированным Кодом транзакции
10	ИС инициатор транзакции	Формирование запроса с включением в конверт сообщения Кода транзакции	Формирование запроса к поставщику информации и включение в конверт сообщения Кода транзакции, полученного от системы генерации кодов транзакций
11	ИС инициатор транзакции	Отправка запроса в СМЭВ	Отправка в СМЭВ сообщения с запросом, конверт которого содержит сформированный Код транзакции
12	Транспортная подсистема СМЭВ	Получение запроса	Получение сообщения с запросом, конверт которого содержит Код транзакции
13	Транспортная подсистема СМЭВ	Проверка ЭП Кода транзакции	Проверка электронной подписи Кода транзакции. Результат – проверка выполнена успешно
14	Транспортная подсистема СМЭВ	Обработка запроса в СМЭВ	Обработка в СМЭВ сообщения с запросом. В рамках обработки запроса осуществляется формирование обратного адреса, который включает следующие сведения:

			<ul style="list-style-type: none"> — Идентификатор ИС инициатора транзакции. — Идентификатор вида сведений, по которому отправлен запрос. — Уникальный идентификатор запроса. — Код транзакции.
15	Транспортная подсистема СМЭВ	Включение кода транзакции в контекст сообщений логов обработки запроса	Осуществляется включение кода транзакции в контекст сообщений логов обработки запроса. Сообщения логов обработки запроса записываются в подсистему централизованного логирования.
16	Транспортная подсистема СМЭВ	Постановка запроса в очередь доставки ИС получателя	Осуществляется постановка запроса, содержащего Код транзакции и обратный адрес, в очередь доставки ИС получателя.
17	ИС получателя	Получение запроса из очереди доставки	Осуществляется запрос очередного сообщения из своей очереди доставки.
18	Транспортная подсистема СМЭВ	Запрос из очереди доставки	Осуществляется возврат запроса, включающего Код транзакции и обратный адрес.
19	ИС получателя	Формирование ответа на запрос	<p>Далее если ИС получателя для формирования ответа необходимо сделать еще запрос, то осуществляется формирование данного запроса, в который помещается Код транзакции, полученный из запроса ИС инициатора транзакции.</p> <p>В случае отсутствия необходимости делать еще запрос либо после получения ответа на свой запрос осуществляется формирование ответа на запрос ИС инициатора транзакции, в который включается обратный адрес.</p>
20	ИС получателя	Отправка ответа на запрос в СМЭВ	Отправка в СМЭВ ответа на запрос, конверт которого содержит обратный адрес с Кодом транзакции.
21	Транспортная подсистема СМЭВ	Получение ответа на запрос	Получение ответа на запрос, конверт которого содержит обратный адрес с Кодом транзакции.
22	Транспортная подсистема	Обработка ответа на запрос	Обработка в СМЭВ ответа на запрос. В рамках обработки ответа на запрос

	СМЭВ		<p>осуществляется извлечение из обратного адреса Кода транзакции и включение его в конверт ответа на запрос.</p> <p>А также осуществляется включение извлеченного Кода транзакции в контекст сообщений логов обработки ответа на запрос.</p> <p>Сообщения логов обработки ответа на запрос записываются в подсистему централизованного логирования.</p>
23	Транспортная подсистема СМЭВ	Постановка ответа на запрос в очередь доставки ИС инициатора транзакции	Осуществляется постановка ответа на запрос, содержащего Код транзакции, в очередь доставки ИС инициатора транзакции.
24	ИС инициатор транзакции	Получение ответа на запрос из очереди доставки	Осуществляется запрос очередного сообщения из своей очереди доставки.
25	Транспортная подсистема СМЭВ	Ответ на запрос из очереди доставки	Осуществляется возврат ответа на запрос, включающего Код транзакции.

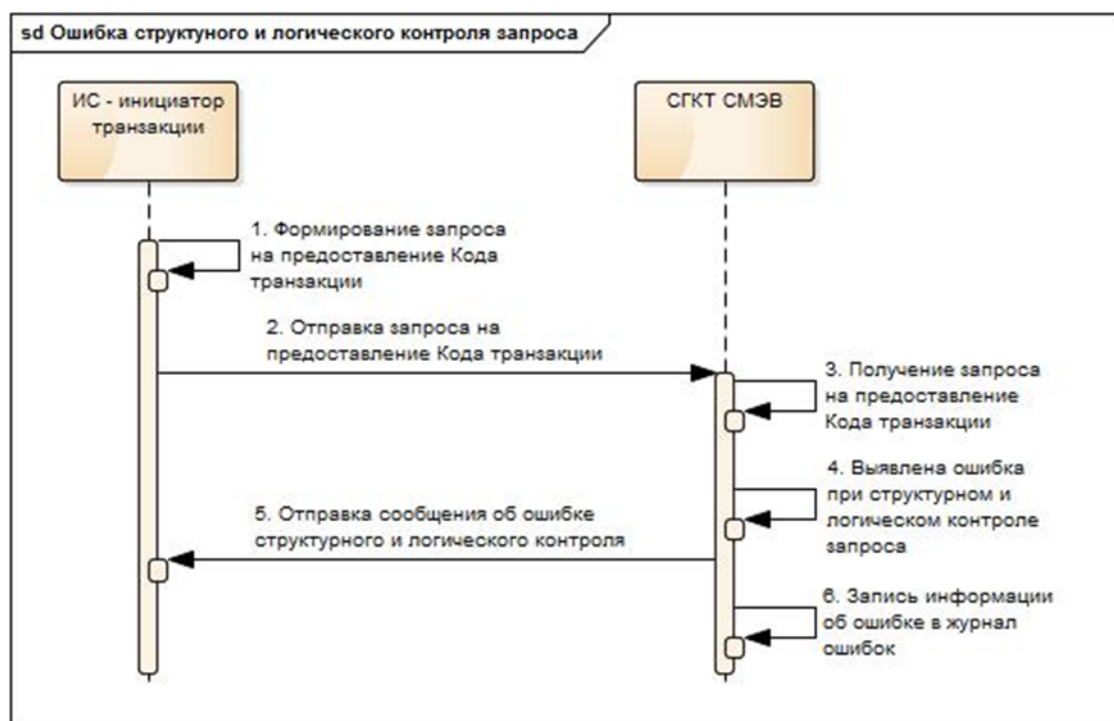


Рисунок 25 – Диаграмма альтернативного сценария процесса ошибки запросов на выдачу кодов транзакций

Таблица 10 – Описание альтернативного сценария процесса ошибки запросов на выдачу кодов транзакций

Предусловие		<p>Средствами ИС инициировано предоставление услуги или выполнение функции. ИС, средствами которой инициировано предоставление услуги или выполнение функции, инициирует транзакцию предоставления услуги или выполнения функции. ИС, инициирующая транзакцию предоставления услуги или выполнения функции, является ИС инициатором транзакции.</p> <p>ИС инициатор транзакции имеет следующие данные:</p> <ul style="list-style-type: none"> – Мнемоника ИС инициатора транзакции. – Код ФРГУ услуги либо признака функции. 	
Постусловие		В ИС получено сообщение об ошибке структурного и логического контроля запроса	
№ шага	Участник	Наименование шага	Описание шага
1	ИС инициатор транзакции	Формирование запроса на предоставление Кода транзакции	<p>Формируется запрос на предоставление кода транзакции, в который включается следующая информация:</p> <ul style="list-style-type: none"> — Мнемоника информационной системы, отправившей запрос (обязательно). — Код ФРГУ услуги, в рамках предоставления которой запрошен Код транзакции, или признак функции (обязательно). — Строка расширенной информации о системе, отправившей запрос (опционально). — Строка расширенной информации о потребителе услуги (опционально).
2	ИС инициатор транзакции	Отправка запроса на предоставление Кода транзакции	Выполняется отправка запроса на предоставление Кода транзакции в систему генерации Кодов транзакций
3	СГКТ СМЭВ	Получение запроса на предоставление Кода транзакции	Получение запроса на предоставление Кода транзакций
4	СГКТ СМЭВ	Выявлена ошибка при структурном и логическом контроле запроса	При выполнении структурного и логического контроля запроса выявлена ошибка
5	СГКТ СМЭВ	Отправка сообщения об ошибке	В ответ на запрос системы инициатора транзакции направляется сообщение об ошибке структурного и логического

		структурного и логического контроля	контроля
6	СГКТ СМЭВ	Запись информации об ошибке в журнал ошибок системы генерации Кодов транзакций	Выполняется запись информации об ошибке структурного и логического контроля в журнал ошибок системы генерации Кодов транзакций. Информация журнала ошибок системы генерации Кодов транзакций предназначена для возможных расследований ошибок при выдаче кодов транзакций

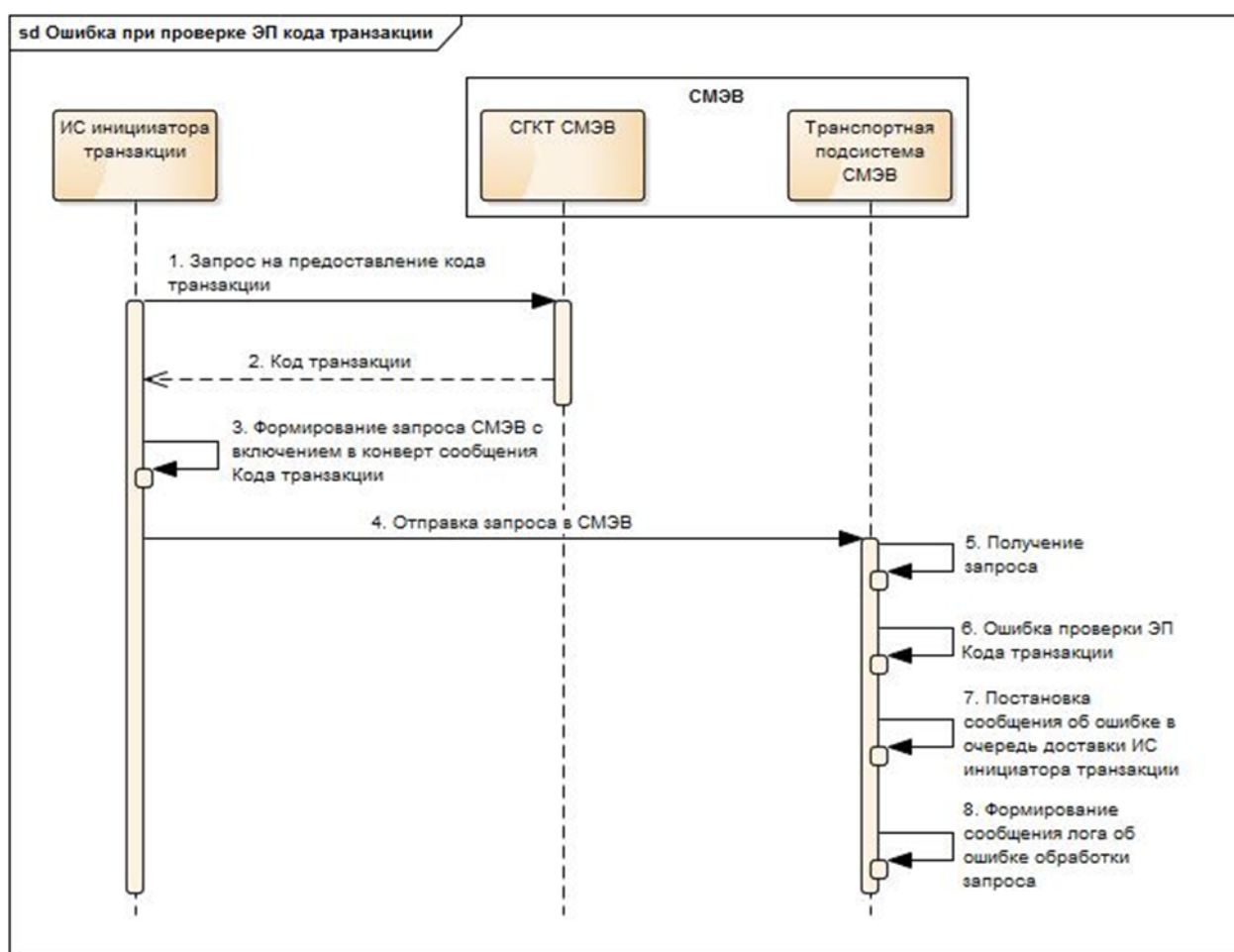


Рисунок 26 – Диаграмма альтернативного сценария процесса ошибки проверки электронной подписи кодов транзакций

Таблица 11 – Описание альтернативного сценария процесса ошибки проверки электронной подписи кодов транзакций

Предусловие	Средствами ИС инициировано предоставление услуги или выполнение функции. ИС, средствами которой инициировано
-------------	--

		<p>предоставление услуги или выполнение функции, инициирует транзакцию предоставления услуги или выполнения функции. ИС, инициирующая транзакцию предоставления услуги или выполнения функции, является ИС инициатором транзакции.</p> <p>ИС инициатор транзакции имеет следующие данные:</p> <ul style="list-style-type: none"> – Мнемоника ИС инициатора транзакции. – Код ФРГУ услуги либо признака функции. 	
Постусловие		Постановка в статусную очередь доставки либо в очередь доставки ответов ИС инициатора транзакции сообщения об ошибке обработки запроса	
№ шага	Участник	Наименование шага	Описание шага
1	ИС инициатор транзакции	Запрос на предоставление Кода транзакции	<p>Выполняется отправка запроса на предоставление кода транзакции, в который включается следующая информация:</p> <ul style="list-style-type: none"> — Мнемоника информационной системы, отправившей запрос (обязательно). — Код ФРГУ услуги, в рамках предоставления которой запрошен Код транзакции, или признак функции (обязательно). — Строка расширенной информации о системе, отправившей запрос (опционально). — Строка расширенной информации о потребителе услуги (опционально).
2	СГКТ СМЭВ	Код транзакции	В случае успешной обработки запроса на предоставление кода транзакции осуществляется отправка сообщения со сформированным кодом транзакции ИС инициатору транзакции.
3	ИС инициатор транзакции	Формирование запроса с включением в конверт сообщения Кода транзакции	<p>Формирование запроса и включение в конверт сообщения поврежденного Кода транзакции.</p> <p>Повреждение кода транзакции, полученного от СГКТ, возможно на стороне ИС инициатора транзакции в процессе формирования запроса, включающего Код транзакции.</p>
4	ИС инициатор транзакции	Отправка запроса в СМЭВ	Отправка в СМЭВ сообщения с запросом, конверт которого содержит

			поврежденный ЭП Кода транзакции
5	Транспортная подсистема СМЭВ	Получение запроса	Получение сообщения с запросом, конверт которого содержит поврежденный ЭП кода транзакции
6	Транспортная подсистема СМЭВ	Ошибка проверки ЭП Кода транзакции	Проверка ЭП Кода транзакции. Результат – Ошибка при проверке ЭП
7	Транспортная подсистема СМЭВ	Постановка сообщения об ошибке в очередь доставки ИС инициатора транзакции	Осуществляется формирование и постановка сообщения об ошибке в статусную очередь доставки либо в очередь доставки ответов ИС инициатора транзакции.
8	Транспортная подсистема СМЭВ	Формирование сообщения лога об ошибке обработки запроса	Осуществляется формирование сообщения лога об ошибке обработки запроса по причине некорректной ЭП.

Сервис предоставления кодов транзакций

Сервис предоставления кодов транзакций (СПКТ) реализован для обеспечения информационного взаимодействия по SOAP протоколу.

СПКТ осуществляет прием входных и отправку выходных данных, приведенных в таблице 12.

Таблица 12 – Перечень входных и выходных данных СПКТ

№	Название	Тип	Обозначение	Описание
1	Входные данные			
1.1	Данные об услуге	Контейнер	FRGUInformation	Данные ФРГУ передаются средствами метода TransactionCode web-сервиса и включает следующие данные: <ul style="list-style-type: none"> – Мнемоника информационной системы. – Код ФРГУ услуги либо признак функции. – Расширенные сведения об услуге или функции. – Расширенные сведения

				о потребителе услуги или функции. Предлагаемый пример SOAP-запроса приведен на рисунке 27.
1.1.1	Мнемоника информационной системы	String	FRGUInteractionParticipantCode	Мнемоника информационной системы длиной не более 20 символов
1.1.2	Код ФРГУ услуги либо признак функции	String	FRGUServiceCode	Код ФРГУ услуги длиной не более 20 символов либо признак функции длиной не более 20 символов.
1.1.3	Расширенные сведения об услуге или функции	String	FRGUServiceDescription	Расширенные сведения об услуге или функции не более 1000 символов.
1.1.4	Расширенные сведения о потребителе услуги или функции	String	FRGUServiceRecipientDescription	Расширенные сведения о потребителе услуги или функции не более 1000 символов.
2	Выходные данные			
2.1	Ответное сообщение на отправленные данные об услуге	Контейнер	Marker	Ответное сообщение на отправленные данные ФРГУ. Ответное сообщение передается средствами метода TransactionCode web-сервиса и включает следующие данные: – Идентификатор кода транзакции. – Код транзакции.
2.1.1	Идентификатор Кода транзакции	String	TransactionUUID	Идентификатор Кода транзакции, генерируемый в соответствии с RFC-4122, по варианту 1 (на основании MAC-адреса и текущего времени).

2.1.2	Код транзакции	String	TransactionCode	<p>Код транзакции в виде строки: TransactionUUID FRGUServiceCode SignatureDetached, где:</p> <ul style="list-style-type: none"> – TransactionUUID - идентификатор Кода транзакции. – FRGUServiceCode - код ФРГУ услуги либо признак функции. – SignatureDetached – это не квалифицированная электронная подпись строки «TransactionUUID FRGUServiceCode» не содержащая сертификата в формате base64. <p>Не квалифицированная электронная подпись генерируется по алгоритму MD5 с RSA с использованием стандартной библиотеки «keytool» из базовой поставки jdk.</p>
-------	----------------	--------	-----------------	--

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <m:generateMark xmlns:m="http://endpoints.transactionmarker.voskhod.ru/">
      <FRGUInformation>

        <FRGUInteractionPartisipantCode>a</FRGUInteractionPartisipantCode>
        <FRGUServiceCode>a</FRGUServiceCode>
        <FRGUServiceDescription>a</FRGUServiceDescription>

        <FRGUServiceRecipientDescription>a</FRGUServiceRecipientDescription>
      </FRGUInformation>
    </m:generateMark>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Рисунок 27 –SOAP-структура запроса кода транзакции

```

<?xml version="1.0"?>
<wsdl:definitions xmlns:ns1="http://voskhod.ru/transaction-marker"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" name="MarkerService"
targetNamespace="http://endpoints.transactionmarker.voskhod.ru/"
xmlns:tns="http://endpoints.transactionmarker.voskhod.ru/">
  <wsdl:types>
    <xs:schema targetNamespace="http://voskhod.ru/transaction-marker"
version="1.0">
      <xs:complexType name="FRGUInformation">
        <xs:sequence>
          <xs:element name="FRGUInteractionPartisipantCode">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:minLength value="1"/>
                <xs:maxLength value="20"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="FRGUServiceCode">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:minLength value="1"/>
                <xs:maxLength value="20"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element minOccurs="0" name="FRGUServiceDescription"
nillable="true">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:minLength value="1"/>
                <xs:maxLength value="1000"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element minOccurs="0"
name="FRGUServiceRecipientDescription" nillable="true">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:minLength value="1"/>
                <xs:maxLength value="1000"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="marker">
        <xs:sequence>
          <xs:element name="TransactionUUID" type="xs:string"/>
          <xs:element name="TransactionCode" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
    <xs:schema attributeFormDefault="unqualified"
elementFormDefault="unqualified"
targetNamespace="http://endpoints.transactionmarker.voskhod.ru/">
      <xs:import namespace="http://voskhod.ru/transaction-marker"/>
      <xs:element name="frguInformation" type="ns1:FRGUInformation"/>
      <xs:element name="generateMark" type="tns:generateMark"/>
      <xs:element name="generateMarkResponse"
type="tns:generateMarkResponse"/>
      <xs:complexType name="generateMark">
        <xs:sequence>
          <xs:element name="FRGUInformation"
type="ns1:FRGUInformation"/>
        </xs:sequence>
      </xs:complexType>
      <xs:complexType name="generateMarkResponse">

```

```

        <xs:sequence>
            <xs:element minOccurs="0" name="return"
type="ns1:marker"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
</wsdl:types>
<wsdl:message name="generateMarkResponse">
    <wsdl:part element="tns:generateMarkResponse" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:message name="generateMark">
    <wsdl:part element="tns:generateMark" name="parameters">
    </wsdl:part>
</wsdl:message>
<wsdl:portType name="IMarker">
    <wsdl:operation name="generateMark">
        <wsdl:input message="tns:generateMark" name="generateMark">
        </wsdl:input>
        <wsdl:output message="tns:generateMarkResponse"
name="generateMarkResponse">
        </wsdl:output>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="MarkerServiceSoapBinding" type="tns:IMarker">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="generateMark">
        <soap:operation soapAction="" style="document"/>
        <wsdl:input name="generateMark">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="generateMarkResponse">
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="MarkerService">
    <wsdl:port binding="tns:MarkerServiceSoapBinding" name="MarkerPort">
        <soap:address location="http://localhost:8080/tm/Marker"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Рисунок 28 – wsdl-описание web-сервиса СГКТ

В описание «SenderProvidedRequestData» схемы «smev-message-exchange-types-1.1.xsd» добавлен следующий обязательный элемент:

```

<xs:element name="TransactionCode" type="basic:string-1500">
<xs:annotation>
    <xs:documentation>
        Код транзакции, в рамках которой осуществляется отправка запроса отправителем.
        Генерируется системой генерации кодов транзакций.
    </xs:documentation>
</xs:annotation>
</xs:element>

```

В описание «Response» схемы «smev-message-exchange-types-1.1.xsd» добавлен следующий обязательный элемент:

```

<xs:element name="OriginalTransactionCode" type="basic:string-1500">
<xs:annotation>
    <xs:documentation>
        Код транзакции, в рамках которой осуществляется отправка ответа отправителем.
        Подставляется код транзакции, полученный в запросе, на который отправлен ответ.
    </xs:documentation>
</xs:annotation>

```

```
</xs:element>
```

Элементы «TransactionCode» и «OriginalTransactionCode» предназначены для передачи строки кода транзакции. Строка кода транзакции передается в виде: Transaction_UUID|FRGU_Code|Signature_Detached, где:

- a) Transaction_UUID – блок UUID кода транзакции, сгенерированный СГКТ в соответствии с RFC-4122, по варианту 1 (на основании MAC-адреса и текущего времени);
- b) FRGU_Code – блок кода ФРГУ услуги либо признака функции;
- c) Signature_Detached – блок неквалифицированной электронной подписи хэш-суммы (по md5) строки «Transaction_UUID|FRGU_Code» в формате base64, сформированной СГКТ по алгоритму RSA и не содержащая сертификата;
- d) | - разделитель блоков кода транзакции.

В схему «smev-message-exchange-faults-1.1.xsd» добавлено следующее описание ошибки обработки кодов транзакций:

```
<xs:element name="TransactionCodeInvalid" type="basic:SmevFault">
  <xs:annotation>
    <xs:documentation>
      Ошибка возникает в случае нарушения целостности кода транзакции.
      Возвращается в процессе обработки кодов транзакций запросов (SendRequest)
    </xs:documentation>
  </xs:annotation>
</xs:element>
```

9. ИСПОЛЬЗОВАНИЕ PUSH-НОТИФИКАЦИЙ

В СМЭВ предусмотрена возможность предоставления информационным системам участников взаимодействия сведений о количестве сообщений в своих очередях доставки. Для этого ИС УВ со своей стороны должны развернуть сервис приема push-уведомлений и подать заявку на его регистрацию данного в СМЭВ.

Участником взаимодействия может быть разработан сервис приема push-уведомлений самостоятельно в соответствии с описанием, приведенным в п. 9.1, либо использовано готовое решение, которое входит в состав набора шаблонов электронных сервисов.

9.1. ОПИСАНИЕ СЕРВИСА ПРИЕМА PUSH-УВЕДОМЛЕНИЙ

Сервис приема push-уведомлений осуществляет прием входных и отправку выходных данных, приведенных в таблице Таблица 13.

Таблица 13 – Перечень входных и выходных данных сервиса приема push-уведомлений

№	Название	Тип	Обозначение	Описание
1	Входные данные			
1.1	Push-уведомление	Контейнер	PushNotification	<p>Push-уведомление передается средствами метода PushNotification web-сервиса и включает следующие данные (см. Рисунок 29):</p> <ul style="list-style-type: none"> – Время опроса очередей доставки ИС УВ; – Наименования очередей доставки; – Количество сообщений в каждой очереди доставки. <p>Предлагаемый пример SOAP-запроса приведен на Рисунок 30.</p>
1.1.1	Время опроса очередей доставки ИС	dateTime	InformationTimestamp	Время опроса в формате ГГГГ-ММ-ДДТЧЧ:ММ:ССZ

	УВ			
1.1.2	Данные состояния очереди	Контейнер	QueueInformation	Включает следующие данные: <ul style="list-style-type: none"> – Наименование очереди доставки; – Количество сообщений в очереди доставки.
1.1.2.1	Наименование очереди доставки	String	QueueName	Наименование одной из трех очередей доставки (не более 500 символов).
1.1.2.2	Количество сообщений в очереди доставки	Int	QueueSize	Количество сообщений в очереди доставки, которая соответствует queueName.
2	Выходные данные			
2.1	Ответное сообщение на отправленное push-уведомление	Контейнер	PushNotificationResponse	Ответное сообщение на отправленное push-уведомление является подтверждением успешного получения ИС УВ push-уведомления. Ответное сообщение передается средствами метода PushNotification web-сервиса и не содержит никаких бизнес данных (см. Рисунок 31).
2.2	Ответное сообщение с ошибкой на отправленное push-	Контейнер	PushNotificationException	Ответное сообщение с ошибкой на отправленное push-уведомление

	уведомление			является подтверждением успешного получения ИС УВ push-уведомления. Ответное сообщение с ошибкой передается средствами метода PushNotification web-сервиса и может содержать описание ошибки (см. Рисунок 32).
2.3	Метод передачи данных push-уведомления	Метод SOAP-сервера	PushNotificationData	Метод передает следующие данные: <ul style="list-style-type: none"> – Время опроса очередей доставки ИС УВ; – Наименования очередей доставки; – Количество сообщений в каждой очереди доставки.

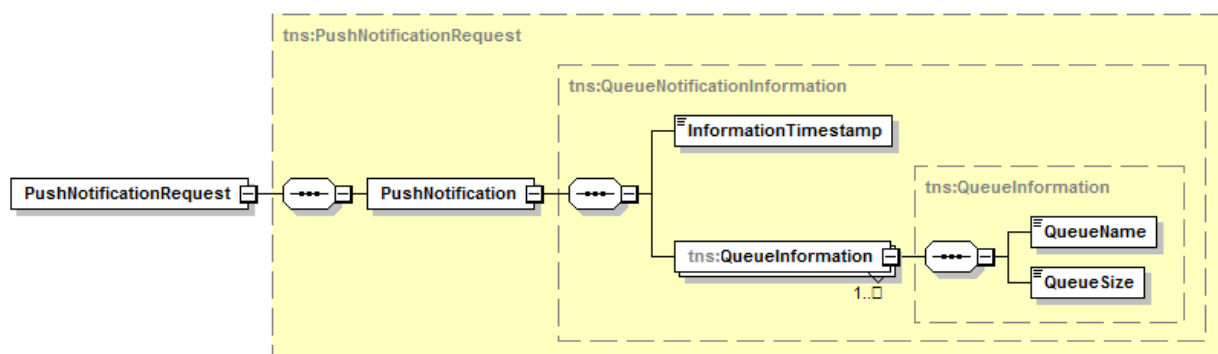


Рисунок 29 – Предлагаемая схема структуры данных push-уведомления

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <PushNotificationRequest xmlns:m="urn://x-artefacts-smev-gov-ru/smev/1.0">
      <PushNotification>
        <InformationTimestamp>2015-02-06T15:14:20Z</InformationTimestamp>
        <QueueInformation>
          <QueueName>name1</QueueName>
```

```

        <QueueSize>2</QueueSize>
      </QueueInformation>
    </QueueInformation>
    <QueueName>name2</QueueName>
    <QueueSize>25</QueueSize>
  </QueueInformation>
  <QueueInformation>
    <QueueName>name3</QueueName>
    <QueueSize>12</QueueSize>
  </QueueInformation>
</PushNotification>
</PushNotificationRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Рисунок 30 – Предлагаемая SOAP-структура push-уведомления

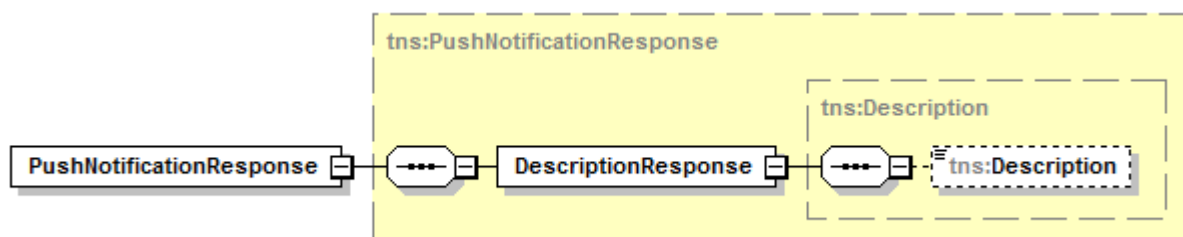


Рисунок 31 – Предлагаемая схема структуры данных ответного сообщения на отправленное push-уведомление

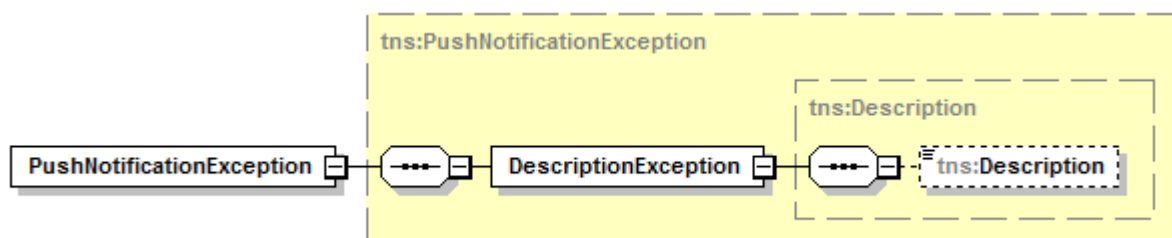


Рисунок 32 – Предлагаемая схема структуры данных ответного сообщения с ошибкой на отправленное push-уведомление

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn://x-artefacts-
smev-gov-ru/smev/1.0" targetNamespace="urn://x-artefacts-smev-gov-ru/smev/1.0"
elementFormDefault="unqualified">
  <xs:element name="PushNotificationRequest" type="tns:PushNotificationRequest"/>
  <xs:complexType name="PushNotificationRequest">
    <xs:sequence>
      <xs:element name="PushNotification"
type="tns:QueueNotificationInformation"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="QueueNotificationInformation">
    <xs:sequence>
      <xs:element name="InformationTimestamp" type="xs:dateTime"/>
      <xs:element name="QueueInformation" type="tns:QueueInformation"
form="qualified" nillable="true" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="QueueInformation">
    <xs:sequence>
      <xs:element name="QueueName">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
            <xs:maxLength value="500"/>
          </xs:restriction>
        </xs:simpleType>

```

```

        </xs:element>
        <xs:element name="QueueSize" type="xs:int"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="PushNotificationResponse"
type="tns:PushNotificationResponse"/>
<xs:complexType name="PushNotificationResponse">
    <xs:sequence>
        <xs:element name="DescriptionResponse" type="tns:Description"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="PushNotificationException"
type="tns:PushNotificationException"/>
<xs:complexType name="PushNotificationException">
    <xs:sequence>
        <xs:element name="DescriptionException" type="tns:Description"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Description">
    <xs:sequence>
        <xs:element name="Description" type="xs:string" form="qualified"
nillable="true" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
</xs:schema>

```

**Рисунок 33 – Предлагаемая xsd-схема сервиса приема push-уведомлений
«PushNotificationSchema.xsd»**

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:tns="urn://x-artefacts-
smev-gov-ru/smev/1.0" targetNamespace="urn://x-artefacts-smev-gov-ru/smev/1.0">
    <wsdl:types>
        <xs:schema>
            <xs:import namespace="urn://x-artefacts-smev-gov-ru/smev/1.0"
schemaLocation="PushMessageSchema.xsd"/>
        </xs:schema>
    </wsdl:types>
    <wsdl:message name="PushNotificationRequest">
        <wsdl:part name="parameter" element="tns:PushNotificationRequest"/>
    </wsdl:message>
    <wsdl:message name="PushNotificationResponse">
        <wsdl:part name="parameter" element="tns:PushNotificationResponse"/>
    </wsdl:message>
    <wsdl:message name="PushNotificationException">
        <wsdl:part name="parameter" element="tns:PushNotificationException"/>
    </wsdl:message>
    <wsdl:portType name="PushNotificationType">
        <wsdl:operation name="PushNotification">
            <wsdl:input name="PushNotificationRequest"
message="tns:PushNotificationRequest"/>
            <wsdl:output name="PushNotificationResponse"
message="tns:PushNotificationResponse"/>
            <wsdl:fault name="PushNotificationException"
message="tns:PushNotificationException"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="PushNotificationBinding" type="tns:PushNotificationType">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="PushNotification">
            <soap:operation soapAction=" PushNotification "/>
            <wsdl:input name="PushNotificationRequest">
                <soap:body use="literal"/>
            </wsdl:input>
        </wsdl:operation>
    </wsdl:binding>

```

```

        </wsdl:input>
        <wsdl:output name="PushNotificationResponse">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="PushNotificationException">
            <soap:fault name="PushNotificationException" use="literal"/>
        </wsdl:fault>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="PushNotificationService">
    <wsdl:port name="PushNotificationPort"
binding="tns:PushNotificationBinding">
        <soap:address location="ServiceAdress"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Рисунок 34 – Предлагаемое wsdl-описание web-сервиса приема push-уведомлений

Если СМЭВ вследствие проблем на стороне информационной системы не может осуществить отправку сервису приема push-уведомлений push-нотификаций, то эти push-нотификации СМЭВ помещает в статусную очередь информационной системы. В этом случае для получения push-нотификаций из статусной очереди информационной системе следует вызвать метод `getStatus` (см. п. 2.6.1).

10. ИСПОЛЬЗОВАНИЕ КЛИЕНТСКОЙ БИБЛИОТЕКИ И НАБОРА «TOOLSET»

Для помощи участникам взаимодействия в разработке информационного взаимодействия ИС со СМЭВ предоставляется клиентская библиотека и набор «toolset», которые входят в состав набора шаблонов электронных сервисов.

Клиентская библиотека подключается к проекту ИС в среде разработки и предоставляет набор методов, с помощью которых осуществляется разработка программной логики процесса информационного взаимодействия со СМЭВ. При этом дорабатываемая ИС должна быть разработана средствами программной платформы jdk. Описание методов клиентской библиотеки предоставляется в виде «javadocs», который входит в состав набора шаблонов электронных сервисов.

Набор «toolset» представляет собой набор отдельных приложений, которые предоставляют платформонезависимые интерфейсы взаимодействия с ИС и запускаются на платформе jre. Каждое приложение набора предоставляет готовое решение, инкапсулирующее часть логики процесса информационного взаимодействия со СМЭВ.х, которое может быть использовано при доработке ИС УВ. При этом дорабатываемая ИС УВ может быть разработана средствами любой программной платформы. Состав набора, требования к необходимому ПО для функционирования набора и порядок его запуска приведены в приложении 8.

11. ПРИЛОЖЕНИЯ

11.1. ПРИЛОЖЕНИЕ 1: АЛГОРИТМ НОРМАЛИЗАЦИИ XML

При подписании XML-фрагментов ЭП в формате XMLDSig, обязательно использование трансформации `urn://smev-gov.ru/xmlsig/transform`. Ее алгоритм:

1. XML declaration и processing instructions, если есть, вырезаются:

ВХОД:

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet type="text/css" href="style.css"?>
<qwe xmlns="http://t.e.s.t">
  <myns:rty xmlns:myns="http://y.e.s">yes!</myns:rty>
  <iop value="yes, yes!"/>
</qwe>
```

ВЫХОД:

```
<qwe xmlns="http://t.e.s.t">
  <myns:rty xmlns:myns="http://y.e.s">yes!</myns:rty>
  <iop value="yes, yes!"/>
</qwe>
```

2. Если текстовый узел содержит только пробельные символы (код символа меньше или равен '\u0020'), этот текстовый узел вырезается.

ВХОД:

```
<qwe xmlns="http://t.e.s.t">
  <myns:rty xmlns:myns="http://y.e.s">yes!</myns:rty>
  <iop value="yes, yes!"/>
</qwe>
```

ВЫХОД:

```
<qwe xmlns="http://t.e.s.t"><myns:rty xmlns:myns="http://y.e.s">yes!</myns:rty><iop
value="yes, yes!"/>
</qwe>
```

3. После применения правил 1 и 2, если даже у элемента нет дочерних узлов, элемент не может быть представлен в виде **empty element tag** (<http://www.w3.org/TR/2008/REC-xml-20081126/#sec-starttags>, правило [44]), а должен быть преобразован в пару **start-tag + end-tag**.

ВХОД:

```
<qwe xmlns="http://t.e.s.t">
  <myns:rty xmlns:myns="http://y.e.s">yes!</myns:rty>
  <iop value="yes, yes!"/>
</qwe>
```

ВЫХОД:

```
<qwe xmlns="http://t.e.s.t">
  <myns:rty xmlns:myns="http://y.e.s">yes!</myns:rty>
  <iop value="yes, yes!"></iop>
</qwe>
```

4. Удалить namespace prefix, которые на текущем уровне объявляются, но не используются.
5. Проверить, что namespace текущего элемента объявлен либо выше по дереву, либо в текущем элементе. Если не объявлен, объявить в текущем элементе.
6. Namespace prefix элементов и атрибутов должны быть заменены на автоматически сгенерированные. Сгенерированный префикс состоит из литерала «ns», и порядкового номера сгенерированного префикса в рамках обрабатываемого XML-фрагмента, начиная с единицы. При генерации префиксов должно устраняться их дублирование.

ВХОД:

```
<qwe xmlns="http://t.e.s.t">
  <myns:rty xmlns:myns="http://y.e.s">yes!</myns:rty>
  <iop value="yes, yes!"/>
</qwe>
```

ВЫХОД:

```
<ns1:qwe xmlns:ns1="http://t.e.s.t">
  <ns2:rty xmlns:ns2="http://y.e.s">yes!</ns2:rty>
  <ns1:iop value="yes, yes!"></ns1:iop>
</ns1:qwe>
```

ВХОД:

```
<nns:x xmlns:nns="http://a" attrB="value1" attrA="value2">
  <y xmlns="http://a">yes!</y>
</nns:x>
```

ВЫХОД:

```
<ns1:x xmlns:ns1="http://a" attrA="value2" attrB="value1">
  <ns1:y>yes!</ns1:y>
</ns1:x>
```

7. Атрибуты должны быть отсортированы в алфавитном порядке: сначала по **namespace URI** (если атрибут - в qualified form), затем – по **local name**. Атрибуты в unqualified form после сортировки идут после атрибутов в qualified form.
8. Объявления namespace prefix должны находиться перед атрибутами. Объявления префиксов должны быть отсортированы в порядке объявления, а именно:
 - a. Первым объявляется префикс пространства имен элемента, если он не был объявлен выше по дереву.
 - b. Далее объявляются префиксы пространств имен атрибутов, если они требуются. Порядок этих объявлений соответствует порядку атрибутов, отсортированных в алфавитном порядке (см. п.5).

Развернутый пример результата трансформации `urn://smev-gov.ru/xmlsig/transform` представлен в приложении [2](#). Образцовая реализация алгоритма на Java для Apache Santuario представлена в приложении [4](#).

Сценарии тестирования алгоритма приведены в приложении 5. Для использования сценариев их необходимо сохранить в файлах в кодировке UTF-8.

11.2. ПРИЛОЖЕНИЕ 2: РЕЗУЛЬТАТ ТРАНСФОРМАЦИИ URN://SMEV-GOV-RU/XMLDSIG/TRANSFORM

Вход:

```
<ns2:SenderProvidedRequestData xmlns:ns2="urn://x-artefacts-smev-gov-ru/services/message-exchange/types/1.0" Id="SIGNED_BY_CONSUMER">
  <MessagePrimaryContent xmlns="urn://x-artefacts-smev-gov-ru/services/message-exchange/types/basic/1.0">
    <SomeRequest:SomeRequest xmlns:SomeRequest="urn://x-artifacts-it-ru/vs/smev/test/test-business-data/1.0">
      <x xmlns="urn://x-artifacts-it-ru/vs/smev/test/test-business-data/1.0">qweqwe</x>
    </SomeRequest:SomeRequest>
  </MessagePrimaryContent>
</ns2:SenderProvidedRequestData>
```

Выход:

```
<ns1:SenderProvidedRequestData xmlns:ns1="urn://x-artefacts-smev-gov-ru/services/message-exchange/types/1.0"
Id="SIGNED_BY_CONSUMER"><ns2:MessagePrimaryContent xmlns:ns2="urn://x-artefacts-smev-gov-ru/services/message-exchange/types/basic/1.0"><ns3:SomeRequest xmlns:ns3="urn://x-artifacts-it-ru/vs/smev/test/test-business-data/1.0"><ns3:x>qweqwe</ns3:x></ns3:SomeRequest></ns2:MessagePrimaryContent></ns1:SenderProvidedRequestData>
```


11.3. ПРИЛОЖЕНИЕ 3: ПРОФИЛЬ ФОРМАТА PKCS#7, КОТОРОМУ ДОЛЖНЫ УДОВЛЕТВОРЯТЬ ПОДПИСИ ВЛОЖЕННЫХ ФАЙЛОВ

```
pkcs-7 OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) 7}
pkcs-9 OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) 9}
gost-r OBJECT IDENTIFIER ::= {iso(1) member-body(2) rus(643) khz(2) 2}

SignatureContentType OBJECT IDENTIFIER ::= {pkcs-7 2} -- PKCS#7 SignedData
SignedFileContentType OBJECT IDENTIFIER ::= {pkcs-7 1} -- PKCS#7 data
DigestAlgorithmIdentifier OBJECT IDENTIFIER ::= {gost-r 9} -- GOST R 34.11-94
DigestEncryptionAlgorithmIdentifier OBJECT IDENTIFIER ::= {gost-r 19} -- GOST R 34.10-2001

Version INTEGER ::= 1 -- PKCS#7 standard version. Refers to version 1.5.

ContentInfo ::= SEQUENCE {
    contentType SignatureContentType,
    content SignedData
}

SignedData ::= SEQUENCE {
    version Version,
    digestAlgorithms DigestAlgorithmIdentifiers,
    contentInfo ExternalContentInfo,
    certificates ExtendedCertificatesAndCertificates,
    signerInfos SignerInfos
}

DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier

ExternalContentInfo ::= SEQUENCE {
    contentType SignedFileContentType
}

ExtendedCertificatesAndCertificates ::= SET OF ExtendedCertificateOrCertificate

ExtendedCertificateOrCertificate ::= CHOICE {
    certificate Certificate -- X.509
}

SignerInfos ::= SET OF SignerInfo

SignerInfo ::= SEQUENCE {
    version Version,
    issuerAndSerialNumber IssuerAndSerialNumber,
    digestAlgorithm DigestAlgorithmIdentifier,
    authenticatedAttributes [0] IMPLICIT Attributes,
    digestEncryptionAlgorithm DigestEncryptionAlgorithmIdentifier,
    encryptedDigest EncryptedDigest
    unauthenticatedAttributes [1] IMPLICIT Attributes OPTIONAL }

EncryptedDigest ::= OCTET STRING
```

11.4. ПРИЛОЖЕНИЕ 4: ОБРАЗЦОВАЯ РЕАЛИЗАЦИЯ ТРАНСФОРМАЦИИ URN://SMEV-GOV-RU/XMLDSIG/TRANSFORM

```
package ru.it.dob.commons.crypto.dsig.impl;

import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.UnsupportedEncodingException;
import java.util.Collections;
import java.util.Comparator;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;
import java.util.Stack;

import javax.xml.parsers.ParserConfigurationException;
import javax.xml.stream.XMLEventFactory;
import javax.xml.stream.XMLEventReader;
import javax.xml.stream.XMLEventWriter;
import javax.xml.stream.XMLInputFactory;
import javax.xml.stream.XMLOutputFactory;
import javax.xml.stream.XMLStreamException;
import javax.xml.stream.events.Attribute;
import javax.xml.stream.events.EndElement;
import javax.xml.stream.events.Namespace;
import javax.xml.stream.events.StartElement;
import javax.xml.stream.events.XMLEvent;

import org.apache.xml.security.c14n.CanonicalizationException;
import org.apache.xml.security.c14n.InvalidCanonicalizerException;
import org.apache.xml.security.signature.XMLSignatureInput;
import org.apache.xml.security.transforms.Transform;
import org.apache.xml.security.transforms.TransformSpi;
import org.apache.xml.security.transforms.TransformationException;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.xml.sax.SAXException;

/**
 * Класс, реализующий алгоритм трансформации "urn://smev-gov-ru/xmldsig/transform"
 * для Apache Santuario.
 * @author dpryakhin
 */
public class SmevTransformSpi extends TransformSpi {
    public static final String ALGORITHM_URN = "urn://smev-gov-ru/xmldsig/transform";
    private static final String ENCODING_UTF_8 = "UTF-8";

    private static Logger logger = LoggerFactory.getLogger(SmevTransformSpi.class);
    private static AttributeSortingComparator attributeSortingComparator =
        new AttributeSortingComparator();

    private static ThreadLocal<XMLInputFactory> inputFactory =
        new ThreadLocal<XMLInputFactory>() {
            @Override
            protected XMLInputFactory initialValue() {
                return XMLInputFactory.newInstance();
            }
        };
};
```

```

private static ThreadLocal<XMLOutputFactory> outputFactory =
    new ThreadLocal<XMLOutputFactory>() {
        @Override
        protected XMLOutputFactory initialValue() {
            return XMLOutputFactory.newInstance();
        }
    };

private static ThreadLocal<XMLEventFactory> eventFactory =
    new ThreadLocal<XMLEventFactory>() {
        @Override
        protected XMLEventFactory initialValue() {
            return XMLEventFactory.newInstance();
        }
    };

@Override
protected String engineGetURI() {
    return ALGORITHM_URN;
}

@Override
protected XMLSignatureInput enginePerformTransform(XMLSignatureInput argInput,
    OutputStream argOutput, Transform argTransform) throws IOException,
    CanonicalizationException, InvalidCanonicalizerException,
    TransformationException, ParserConfigurationException, SAXException {

    process(argInput.getOctetStream(), argOutput);
    XMLSignatureInput result = new XMLSignatureInput((byte[]) null);
    result.setOutputStream(argOutput);
    return result;
}

@Override
protected XMLSignatureInput enginePerformTransform(XMLSignatureInput argInput,
    Transform argTransform) throws IOException, CanonicalizationException,
    InvalidCanonicalizerException, TransformationException,
    ParserConfigurationException, SAXException {

    return enginePerformTransform(argInput);
}

@Override
protected XMLSignatureInput enginePerformTransform(XMLSignatureInput argInput)
    throws IOException, CanonicalizationException,
    InvalidCanonicalizerException, TransformationException,
    ParserConfigurationException, SAXException {

    ByteArrayOutputStream result = new ByteArrayOutputStream();
    process(argInput.getOctetStream(), result);
    byte[] postTransformData = result.toByteArray();

    return new XMLSignatureInput(postTransformData);
}

public void process(InputStream argSrc, OutputStream argDst) throws
TransformationException {

    DebugOutputStream debugStream = null;

```

```

Stack<List<Namespace>> prefixMappingStack = new Stack<List<Namespace>>();
int prefixCnt = 1;
XMLEventReader src = null;
XMLEventWriter dst = null;
try {
    src = inputFactory.get().createXMLEventReader(argSrc, ENCODING_UTF_8);
    if (logger.isDebugEnabled()) {
        debugStream = new DebugOutputStream(argDst);
        dst = outputFactory.get().createXMLEventWriter(debugStream, ENCODING_UTF_8);
    } else {
        dst = outputFactory.get().createXMLEventWriter(argDst, ENCODING_UTF_8);
    }
    XMLEventFactory factory = eventFactory.get();

    while(src.hasNext()) {
        XMLEvent event = src.nextEvent();

        if (event.isCharacters()) {
            String data = event.asCharacters().getData();
            // Отсекаем возвраты каретки и пробельные строки.
            if (!data.trim().isEmpty()) {
                dst.add(event);
            }
            continue;
        } else if (event.isStartElement()) {
            List<Namespace> myPrefixMappings = new LinkedList<Namespace>();
            prefixMappingStack.push(myPrefixMappings);

            // Обработка элемента: NS prefix rewriting.
            // N.B. Элементы в unqualified form не поддерживаются.
            StartElement srcEvent = (StartElement)event;
            String nsURI = srcEvent.getName().getNamespaceURI();
            String prefix = findPrefix(nsURI, prefixMappingStack);

            if (prefix == null) {
                prefix = "ns" + String.valueOf(prefixCnt++);
                myPrefixMappings.add(factory.createNamespace(prefix, nsURI));
            }
            StartElement dstEvent = factory.createStartElement(
                prefix, nsURI, srcEvent.getName().getLocalPart());
            dst.add(dstEvent);

            // == Обработка атрибутов. Два шага: отсортировать, проэпить namespace URI.

            Iterator<Attribute> srcAttributeIterator = srcEvent.getAttributes();
            // Положим атрибуты в list, чтобы их можно было отсортировать.
            List<Attribute> srcAttributeList = new LinkedList<Attribute>();
            while(srcAttributeIterator.hasNext()) {
                srcAttributeList.add(srcAttributeIterator.next());
            }
            // Сортировка атрибутов по алфавиту.
            Collections.sort(srcAttributeList, attributeSortingComparator);

            // Обработка префиксов. Аналогична обработке префиксов элементов,
            // за исключением того, что у атрибут может не иметь namespace.
            List<Attribute> dstAttributeList = new LinkedList<Attribute>();
            for (Attribute srcAttribute : srcAttributeList) {
                String attributeNsURI = srcAttribute.getName().getNamespaceURI();
                String attributeLocalName = srcAttribute.getName().getLocalPart();
                String value = srcAttribute.getValue();

```

```

String attributePrefix = null;
Attribute dstAttribute = null;
if (attributeNsURI != null && !"".equals(attributeNsURI)) {
    attributePrefix = findPrefix(attributeNsURI, prefixMappingStack);
    if (attributePrefix == null) {
        attributePrefix = "ns" + String.valueOf(prefixCnt++);
        myPrefixMappings.add(factory.createNamespace(
            attributePrefix, attributeNsURI));
    }
    dstAttribute = factory.createAttribute(
        attributePrefix, attributeNsURI, attributeLocalName, value);
} else {
    dstAttribute = factory.createAttribute(attributeLocalName, value);
}
dstAttributeList.add(dstAttribute);
}

// Вынести namespace prefix mappings для текущего элемента.
// Их порядок детерминирован, т.к. перед мэппингом атрибуты
// были отсортированы.
// Поэтому дополнительной сортировки здесь не нужно.
for (Namespace mapping : myPrefixMappings) {
    dst.add(mapping);
}

// Вывести атрибуты.
// N.B. Мы не выводим атрибуты сразу вместе с элементом, используя метод
// XMLEventFactory.createStartElement(prefix, nsURI, localName,
// List<Namespace>, List<Attribute>),
// потому что при использовании этого метода порядок атрибутов
// в выходном документе меняется произвольным образом.
for (Attribute attr : dstAttributeList) {
    dst.add(attr);
}

continue;
} else if (event.isEndElement()) {
    // Гарантируем, что empty tags запишутся в форме <a></a>, а не в форме <a/>.
    dst.add(eventFactory.get().createSpace(""));

    // NS prefix rewriting
    EndElement srcEvent = (EndElement)event;
    String nsURI = srcEvent.getName().getNamespaceURI();
    String prefix = findPrefix(nsURI, prefixMappingStack);
    if (prefix == null) {
        throw new TransformationException(
            "EndElement: prefix mapping is not found for namespace " + nsURI);
    }

    EndElement dstEvent = eventFactory.get().
        createEndElement(prefix, nsURI, srcEvent.getName().getLocalPart());
    dst.add(dstEvent);
    continue;
} else if (event.isAttribute()) {
    // Атрибуты обрабатываются в событии startElement.
    continue;
}

// Остальные события (processing instructions, start document, etc.)
// нас не интересуют.
}
} catch (XMLStreamException e) {

```

```

Object exArgs[] = { e.getMessage() };
throw new TransformationException(
    "Can not perform transformation " + ALGORITHM_URN, exArgs, e
);
} finally {
    if (src != null) {
        try {
            src.close();
        } catch (XMLStreamException e) {
            logger.warn("Can not close XMLEventReader", e);
        }
    }
    if (dst != null) {
        try {
            dst.close();
        } catch (XMLStreamException e) {
            logger.warn("Can not close XMLEventWriter", e);
        }
    }
    try {
        argSrc.close();
    } catch (IOException e) {
        logger.warn("Can not close input stream.", e);
    }
    try {
        argDst.close();
    } catch (IOException e) {
        logger.warn("Can not close output stream.", e);
    }

    if (logger.isDebugEnabled()) {
        try {
            String contentAfterCanonizationAndTransforms =
                new String(debugStream.getCollectedData(), "UTF-8");
            logger.debug("Content after canonization: " +
                contentAfterCanonizationAndTransforms);
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
}
}

private String findPrefix(String argNamespaceURI, Stack<List<Namespace>>
argMappingStack) {
    if (argNamespaceURI == null) {
        throw new IllegalArgumentException("No namespace элементы не поддерживаются.");
    }

    for (List<Namespace> elementMappingList : argMappingStack) {
        for (Namespace mapping : elementMappingList) {
            if (argNamespaceURI.equals(mapping.getNamespaceURI())) {
                return mapping.getPrefix();
            }
        }
    }
    return null;
}

private static class AttributeSortingComparator implements Comparator<Attribute> {
    @Override

```

```

public int compare(Attribute x, Attribute y) {
    String xNS = x.getName().getNamespaceURI();
    String xLocal = x.getName().getLocalPart();
    String yNS = y.getName().getNamespaceURI();
    String yLocal = y.getName().getLocalPart();

    // Сначала сравниваем namespaces.
    if (xNS == null || xNS.equals("")) {
        if (yNS != null && !"".equals(xNS)) {
            return 1;
        }
    } else {
        if (yNS == null || "".equals(yNS)) {
            return -1;
        } else {
            int nsComparisonResult = xNS.compareTo(yNS);
            if (nsComparisonResult != 0) {
                return nsComparisonResult;
            }
        }
    }

    // Если namespaces признаны эквивалентными, сравниваем local names.
    return xLocal.compareTo(yLocal);
}

private static class DebugOutputStream extends OutputStream {
    private ByteArrayOutputStream collector = new ByteArrayOutputStream();
    private OutputStream wrappedStream;

    public DebugOutputStream(OutputStream arg) {
        wrappedStream = arg;
    }

    public byte[] getCollectedData() {
        try {
            collector.flush();
        } catch (IOException e) {
        }
        return collector.toByteArray();
    }

    @Override
    public void write(int b) throws IOException {
        collector.write(b);
        wrappedStream.write(b);
    }

    @Override
    public void close() throws IOException {
        collector.close();
        wrappedStream.close();
        super.close();
    }
}

```

```
@Override
public void flush() throws IOException {
    collector.flush();
    wrappedStream.flush();
}

}
```


11.5. ПРИЛОЖЕНИЕ 5: СЦЕНАРИИ ТЕСТИРОВАНИЯ АЛГОРИТМА НОРМАЛИЗАЦИИ XML

11.5.1 Сценарий 1: тестирование правил 1, 2, 6 (здесь и далее под правилами понимаются подпункты алгоритма нормализации, описанного в Приложении 1).

Вход

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Тестирование правил 1, 2, 6:
      - XML declaration выше, этот комментарий, и следующая за ним processing
instruction должны быть вырезаны;
      - Переводы строки должны быть удалены;
      - Namespace prefixes заменяются на автоматически сгенерированные.
-->
<?xml-stylesheet type="text/xsl" href="style.xsl"?>

<elementOne xmlns="http://test/1">
    <qwe:elementTwo xmlns:qwe="http://test/2">asd</qwe:elementTwo>
</elementOne>
```

Выход

```
<ns1:elementOne xmlns:ns1="http://test/1"><ns2:elementTwo
xmlns:ns2="http://test/2">asd</ns2:elementTwo></ns1:elementOne>
```

11.5.2 Сценарий 2: тестирование правил 4, 5

Вход

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
      Всё то же, что в test case 1, плюс правила 4 и 5:
      - Удалить namespace prefix, которые на текущем уровне объявляются, но не
используются.
      - Проверить, что namespace текущего элемента объявлен либо выше по дереву, либо
в текущем элементе. Если не объявлен, объявить в текущем элементе
-->
<?xml-stylesheet type="text/xsl" href="style.xsl"?>

<elementOne xmlns="http://test/1" xmlns:qwe="http://test/2" xmlns:asd="http://test/3">
    <qwe:elementTwo>
        <asd:elementThree>
            <!-- Проверка обработки default namespace. -->
            <elementFour> z x c </elementFour>
            <!-- Тестирование ситуации, когда для одного namespace объявляется
несколько префиксов во вложенных элементах. -->
            <qqq:elementFive xmlns:qqq="http://test/2"> w w w
</qqq:elementFive>
        </asd:elementThree>
        <!-- Ситуация, когда prefix был объявлен выше, чем должно быть в
нормальной форме,
при нормализации переносится ниже, и это приводит к генерации нескольких
префиксов
для одного namespace в sibling элементах. -->
        <asd:elementSix>eee</asd:elementSix>
    </qwe:elementTwo>
</elementOne>
```

Выход

```
<ns1:elementOne xmlns:ns1="http://test/1"><ns2:elementTwo
xmlns:ns2="http://test/2"><ns3:elementThree
xmlns:ns3="http://test/3"><ns1:elementFour> z x c </ns1:elementFour><ns2:elementFive>
w w w </ns2:elementFive></ns3:elementThree><ns4:elementSix
xmlns:ns4="http://test/3">eee</ns4:elementSix></ns2:elementTwo></ns1:elementOne>
```

11.5.3 Сценарий 3: тестирование правил 3, 7, 8

Вход

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Всё то же, что в test case 1, плюс правила 3, 7 и 8:
    - Атрибуты должны быть отсортированы в алфавитном порядке: сначала по namespace
    URI (если атрибут - в qualified form), затем - по local name.
    Атрибуты в unqualified form после сортировки идут после атрибутов в
    qualified form.
    - Объявления namespace prefix должны находиться перед атрибутами. Объявления
    префиксов должны быть отсортированы в порядке объявления, а именно:
    а. Первым объявляется префикс пространства имен элемента, если он не
    был объявлен выше по дереву.
    б. Далее объявляются префиксы пространств имен атрибутов, если они
    требуются.
    Порядок этих объявлений соответствует порядку атрибутов,
    отсортированных в алфавитном порядке (см. п.5).
-->
<?xml-stylesheet type="text/xsl" href="style.xsl"?>

<elementOne xmlns="http://test/1" xmlns:qwe="http://test/2" xmlns:asd="http://test/3">
  <qwe:elementTwo>
    <asd:elementThree xmlns:wer="http://test/a" xmlns:zxc="http://test/0"
wer:attZ="zzz" attB="bbb" attA="aaa" zxc:attC="ccc" asd:attD="ddd" asd:attE="eee"
qwe:attF="fff"/>
  </qwe:elementTwo>
</elementOne>
```

Выход

```
<ns1:elementOne xmlns:ns1="http://test/1"><ns2:elementTwo
xmlns:ns2="http://test/2"><ns3:elementThree xmlns:ns3="http://test/3"
xmlns:ns4="http://test/0" xmlns:ns5="http://test/a" ns4:attC="ccc" ns2:attF="fff"
ns3:attD="ddd" ns3:attE="eee" ns5:attZ="zzz" attA="aaa"
attB="bbb"></ns3:elementThree></ns2:elementTwo></ns1:elementOne>
```

11.6. ПРИЛОЖЕНИЕ 6: ПЕРЕЧЕНЬ ОШИБОК, ВОЗВРАЩАЕМЫХ ТРАНСПОРТНОЙ ПОДСИСТЕМОЙ СМЭВ

Таблица 14 – Перечень ошибок, возвращаемых транспортной подсистемой СМЭВ после отправки сообщения методом `sendRequest`

Исключение	Текст ошибки
<code>AccessDeniedException</code>	Доступ запрещён
<code>AttachmentContentMiscoordinationException</code>	<p>"Количество вложений - " + @количество_вложений + ", нет ни одного заголовка."</p> <p>"Количество вложений - " + @количество_вложений + ", количество заголовков - " + @количество_заголовков</p> <p>"Вложение [Id=\" + @id_вложения + "\"] не имеет заголовка."</p> <p>"Некорректная информация о ftp вложениях; message id = " + @id_сообщения</p> <p>"Вложения не имеют заполненных требуемых полей."</p>
<code>AttachmentSizeLimitExceededException</code>	<p>Превышен максимально допустимый суммарный размер присоединённых файлов.</p> <p>Превышен максимально допустимый суммарный размер ftp файлов.</p>
<code>QuoteLimitExceededException</code>	Квота на файловое хранилище для получателя превышена!
<code>BusinessDataTypesNotSupportedException</code>	<p>Неподдерживаемый тип запроса.</p> <p>Попытка послать сообщение {" + @requestNamespaceURI + "} + @requestRootElementLocalName + " через метод <code>sendRequest</code>, в то время как этот тип сообщений зарегистрирован как " + @recipientSMEVAddress.getMessageCategory()</p>
<code>InvalidContentException</code>	"Нарушен формат бизнес-конверта."

Исключение	Текст ошибки
	"Бизнес-данные сообщения не соответствуют схеме, зарегистрированной в СМЭВ. MessageId = " + @Message_Id
RecipientIsNotFoundException	<p>Не удалось найти получателя по причине неполноты входных данных: " + @error</p> <p>"Невозможно определить получателя для сообщения. Полное имя корневого элемента: {" + @requestNamespaceURI + "}" + @requestRootElementLocalName</p> <p>"Не удалось найти получателя по причине неполноты входных данных: " + @error"</p> <p>"Невозможно определить получателя для сообщения. Полное имя корневого элемента: {" + @requestNamespaceURI + "}" + @requestRootElementLocalName + "; Ошибка ОКТМО:" + @error</p> <p>"Найдено несколько получателей для сообщения. Полное имя корневого элемента: {" + @requestNamespaceURI + "}" + @requestRootElementLocalName</p> <p>"Не удалось найти получателя по причине неполноты входных данных: " + @error</p>
SMEVFailureException	Ошибка СМЭВ. Обратитесь в службу технической поддержки.
SenderIsNotRegisteredException	<p>"Информационная система не зарегистрирована в СМЭВ."</p> <p>"Сертификат сотрудника не зарегистрирован в СМЭВ."</p>
SignatureVerificationFaultException	<p>"Отсутствует ЭП-ОВ"</p> <p>"Срок действия сертификата истёк. Сертификат действителен до " + @validUntil</p> <p>"Срок действия сертификата не начался. Сертификат</p>

Исключение	Текст ошибки
	<p>действителен с " + @validSince</p> <p>"Сертификат сотрудника не действителен."</p> <p>"Проверка подписи на вложении " + @id_вложения + ": срок действия сертификата истёк."</p> <p>"Проверка подписи на вложении " + @id_вложения + ": " + @error</p> <p>"Срок действия сертификата " + @signatureTypeAsString + " истёк."</p> <p>@signatureTypeAsString + " не соответствует подписанным данным: "</p> <p>@signatureTypeAsString + " отсутствует в сообщении " + @MessageId</p> <p>"Сертификат отозван. Код ответа в ГУЦ:" + @code</p>
DestinationOverflowException	"Очередь, в которую должно быть отправлено сообщение, переполнена."
MessageIsAlreadySentException	"Сообщение с идентификатором " + @messageId + " было послано ранее."
InvalidMessageIdFormatException	"Недопустимый формат идентификатора сообщения. См. RFC-4122."
StaleMessageIdException	"Timestamp идентификатора сообщения слишком давний."

Таблица 15 – Перечень ошибок, возвращаемых транспортной подсистемой СМЭВ после отправки сообщения методом sendResponse

Исключение	Текст ошибки
AccessDeniedException	Доступ запрещён

Исключение	Текст ошибки
AttachmentContentMiscoordinationException	<p>"Количество вложений - " + @количество_вложений + ", нет ни одного заголовка."</p> <p>"Количество вложений - " + @количество_вложений + ", количество заголовков - " + @количество_заголовков</p> <p>"Вложение [Id=\" + @id_вложения + \"] не имеет заголовка."</p> <p>"Некорректная информация о ftp вложениях; message id = " + @id_сообщения</p> <p>"Вложения не имеют заполненных требуемых полей."</p>
AttachmentSizeLimitExceededException	<p>Превышен максимально допустимый суммарный размер присоединённых файлов.</p> <p>Превышен максимально допустимый суммарный размер ftp файлов.</p>
QuoteLimitExceededException	Квота на файловое хранилище для получателя превышена!
BusinessDataTypesNotSupportedException	<p>"Неподдерживаемый тип запроса."</p> <p>"Попытка послать сообщение {" + @businessDataNamespaceURI + "} + @businessDataRootElementLocalName + " через метод sendResponse, в то время как этот тип сообщений зарегистрирован как " + @messageType</p>
InvalidContentException	<p>"Нарушен формат бизнес-конверта."</p> <p>"Попытка послать сообщение {" + @businessDataNamespaceURI + "} + @businessDataRootElementLocalName + " через метод sendResponse, в то время как этот тип сообщений не</p>

Исключение	Текст ошибки
	<p>зарегистрирован в СМЭВ."</p> <p>"Бизнес-данные сообщения не соответствуют схеме, зарегистрированной в СМЭВ. MessageId = " + @MessageId</p>
RecipientIsNotFoundException	<p>"Невозможно определить получателя для ответа на запрос. Адресная информация: " + @SenderProvidedResponseData().getTo()</p> <p>"Не удалось найти получателя по причине неполноты входных данных: " @error</p> <p>"Невозможно определить получателя для ответа на запрос. Адресная информация: " + @SenderProvidedResponseData().getTo()</p> <p>"Не удалось найти получателя по причине неполноты входных данных: " + @error</p> <p>"Не удалось найти получателя по причине неполноты входных данных: " + @error</p>
SMEVFailureException	Ошибка СМЭВ. Обратитесь в службу технической поддержки.
SenderIsNotRegisteredException	<p>"Информационная система не зарегистрирована в СМЭВ."</p> <p>"Сертификат, которым подписано вложение, не зарегистрирован в СМЭВ."</p>
SignatureVerificationFaultException	<p>"Отсутствует ЭП-ОВ"</p> <p>"Срок действия сертификата истёк. Сертификат действителен до " + @validUntil</p> <p>"Срок действия сертификата не начался. Сертификат действителен с " + @validSince</p> <p>"Сертификат, которым подписано вложение, не</p>

Исключение	Текст ошибки
	<p>действителен."</p> <p>"Проверка подписи на вложении " + @id_вложения + ": срок действия сертификата истёк."</p> <p>"Проверка подписи на вложении " + @id_вложения + ": " + @error</p> <p>"Срок действия сертификата " + @signatureTypeAsString + " истёк."</p> <p>@signatureTypeAsString + " не соответствует подписанным данным: "</p> <p>@signatureTypeAsString + " отсутствует в сообщении " + @MessageId</p> <p>"Сертификат отозван. Код ответа в ГУЦ:" + @code</p>
DestinationOverflowException	"Очередь, в которую должно быть отправлено сообщение, переполнена."
MessageIsAlreadySentException	"Сообщение с идентификатором " + @messageId + " было послано ранее."
InvalidMessageIdFormatException	"Недопустимый формат идентификатора сообщения. См. RFC-4122."
StaleMessageIdException	"Timestamp идентификатора сообщения слишком давний."

Таблица 16 – Перечень ошибок, возвращаемых транспортной подсистемой СМЭВ после запроса на получение сообщения методом getRequest

Исключение	Текст ошибки
InvalidContentException	<p>"Нарушен формат бизнес-конверта."</p> <p>"Бизнес-данные сообщения не соответствуют схеме, зарегистрированной в СМЭВ. MessageId</p>

Исключение	Текст ошибки
	= " + @MessageId
SMEVFailureException	Ошибка СМЭВ. Обратитесь в службу технической поддержки.
SenderIsNotRegisteredException	"Отправитель не зарегистрирован в СМЭВ" "Предъявленный сертификат пользователя " + @CallerCertificate.getSubjectX500Principal().getName(X500Principal.RFC1779) + " не зарегистрирован в СМЭВ"
SignatureVerificationFaultException	"Отсутствует ЭП-ОВ" "Срок действия сертификата истёк. Сертификат действителен до " + @validUntil "Срок действия сертификата не начался. Сертификат действителен с " + @validSince "Срок действия сертификата " + @signatureTypeAsString + " истёк." @signatureTypeAsString + " не соответствует подписанным данным: " @signatureTypeAsString + " отсутствует в сообщении " + @MessageId "Сертификат отозван. Код ответа в ГУЦ:" + @code
UnknownMessageTypeException	"Входящая очередь запрошенного типа сообщений, принадлежащая пользователю " + @CallerCertificate.getSubjectX500Principal().getName(X500Principal.RFC1779) + " не зарегистрирована в СМЭВ"

Таблица 17 – Перечень ошибок, возвращаемых транспортной подсистемой СМЭВ после запроса на получение сообщения методом `getResponse`

Исключение	Текст ошибки
------------	--------------

Исключение	Текст ошибки
InvalidContentException	<p>"Нарушен формат бизнес-конверта."</p> <p>"Бизнес-данные сообщения не соответствуют схеме, зарегистрированной в СМЭВ. MessageId = " + @MessageId</p>
SMEVFailureException	Ошибка СМЭВ. Обратитесь в службу технической поддержки.
SenderIsNotRegisteredException	<p>"Отправитель не зарегистрирован в СМЭВ"</p> <p>"Предъявленный сертификат пользователя " + @CallerCertificate.getSubjectX500Principal().getName(X500Principal.RFC1779) + " не зарегистрирован в СМЭВ"</p>
SignatureVerificationFaultException	<p>"<getResponse> Отсутствует ЭП-ОВ"</p> <p>"Срок действия сертификата истёк. Сертификат действителен до " + @validUntil</p> <p>"Срок действия сертификата не начался. Сертификат действителен с " + @validSince</p> <p>"Срок действия сертификата " + @signatureTypeAsString + " истёк."</p> <p>@signatureTypeAsString + " не соответствует подписанным данным: "</p> <p>@signatureTypeAsString + " отсутствует в сообщении " + @MessageId</p> <p>"Сертификат отозван. Код ответа в ГУЦ:" + @code</p>
UnknownMessageTypeException	<p>"Входящая очередь запрошенного типа сообщений, принадлежащая пользователю " + @CallerCertificate.getSubjectX500Principal().getName(X500Principal.RFC1779) + " не зарегистрирована в СМЭВ"</p>

Таблица 18 – Перечень ошибок, возвращаемых транспортной подсистемой СМЭВ после отправки подтверждения получения сообщения методом ask

Исключение	Текст ошибки
InvalidContentException	"Нарушен формат бизнес-конверта." "Бизнес-данные сообщения не соответствуют схеме, зарегистрированной в СМЭВ. MessageId = " + @MessageId
SMEVFailureException	Ошибка СМЭВ. Обратитесь в службу технической поддержки.
SenderIsNotRegisteredException	"Информационная система не зарегистрирована в СМЭВ." "Предъявленный сертификат пользователя " + @CallerCertificate.getSubjectX500Principal().getName(X500Principal.RFC1779) + " не зарегистрирован в СМЭВ"
SignatureVerificationFaultException	" Отсутствует ЭП-ОВ" "Срок действия сертификата истёк. Сертификат действителен до " + @validUntil "Срок действия сертификата не начался. Сертификат действителен с " + @validSince "Срок действия сертификата " + @signatureTypeAsString + " истёк." @signatureTypeAsString + " не соответствует подписанным данным: " @signatureTypeAsString + " отсутствует в сообщении " + @MessageId "Сертификат отозван. Код ответа в ГУЦ:" + @code
TargetMessageIsNotFoundException	"Сообщение " + @AckTargetMessage " не найдено среди неподтверждённых."

Таблица 19 – Перечень ошибок, возвращаемых транспортной подсистемой СМЭВ после обращения к методу getStatus

Исключение	Текст ошибки
InvalidContentException	<p>"Нарушен формат бизнес-конверта."</p> <p>"Бизнес-данные сообщения не соответствуют схеме, зарегистрированной в СМЭВ. MessageId = " + @MessageId</p>
SMEVFailureException	<p>Ошибка СМЭВ. Обратитесь в службу технической поддержки.</p>
SenderIsNotRegisteredException	<p>"Информационная система не зарегистрирована в СМЭВ."</p> <p>"Предъявленный сертификат пользователя " + @CallerCertificate.getSubjectX500Principal().getName(X500Principal.RFC1779) + " не зарегистрирован в СМЭВ"</p>
SignatureVerificationFaultException	<p>" Отсутствует ЭП-ОВ"</p> <p>"Срок действия сертификата истёк. Сертификат действителен до " + @validUntil</p> <p>"Срок действия сертификата не начался. Сертификат действителен с " + @validSince</p> <p>"Срок действия сертификата " + @signatureTypeAsString + " истёк."</p> <p>@signatureTypeAsString + " не соответствует подписанным данным: "</p> <p>@signatureTypeAsString + " отсутствует в сообщении " + @MessageId</p> <p>"Сертификат отозван. Код ответа в ГУЦ:" + @code</p>

Исключение	Текст ошибки
UnknownMessageTypeException	"Входящая очередь запрошенного типа сообщений, принадлежащая пользователю " + @CallerCertificate.getSubjectX500Principal().getName(X500Principal.RFC1779) + " не зарегистрирована в СМЭВ"

11.7. ПРИЛОЖЕНИЕ 7: ФОРМИРОВАНИЕ ВИДОВ СВЕДЕНИЙ С ВКЛЮЧЕНИЕМ СПРАВОЧНИКОВ

Поставщик должен импортировать в XSD-схему вида сведений пространство имен XSD-схем, содержащих объявления типов данных для всех использованных справочников.

Импорт пространства имен осуществляется с использованием инструкции `import`.

Пример вида сведений, импортирующего пространство имен с объявлениями типов данных справочника «gender», приведен на рисунке 35. В приведенном примере импорт пространства имен обеспечивается инструкцией вида:

```
<xs:import namespace="urn://cnsi-dictionary/types/gender/3.0.0"
schemaLocation="types/gender.xsd" />.
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="urn://simple_test/1.0"
  xmlns:gender="urn://cnsi-dictionary/types/gender/3.0.0"
  targetNamespace="urn://simple_test/1.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:import namespace="urn://cnsi-dictionary/types/gender/3.0.0"
  schemaLocation="types/gender.xsd"/>

  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Person" type="tns:Person" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="Person">
    <xs:sequence>
      <xs:element name="FIO" type="xs:string"/>
      <xs:element name="Gender" type="gender:ID"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>
```

Рисунок 35 – Пример XSD-схемы вида сведений, импортирующего схему с объявлением типов справочников

Импортируемая XSD-схема с объявлениями типов справочника «gender» приведена на рисунке 36.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="urn://cnsi-dictionary/types/gender/3.0.0"
  targetNamespace="urn://cnsi-dictionary/types/gender/3.0.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:simpleType name="Name">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="10"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ID">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="10"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="Gender">
    <xs:sequence>
      <xs:element name="Name" type="tns:Name" minOccurs="0"/>
      <xs:element name="ID" type="tns:ID" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>

```

Рисунок 36 – Пример XSD-схемы с объявлениями типов справочника «gender»

Пример запроса, соответствующего схеме на рисунке 35, приведен на рисунке 37.

```

<ns:root xmlns:ns="urn://simple_test/1.0">
  <ns:Person>
    <ns:FIO>Иванов И.И.</ns:FIO>
    <ns:Gender>0001</ns:Gender>
  </ns:Person>
  <ns:Person>
    <ns:FIO>Петрова А.А.</ns:FIO>
    <ns:Gender>0002</ns:Gender>
  </ns:Person>
</ns:root>

```

Рисунок 37 – Пример запроса с использованием значений справочника «gender»

11.8. ПРИЛОЖЕНИЕ 8: ОПИСАНИЕ НАБОРА «TOOLSET»

11.8.1 Состав набора

Набор «toolset» поставляется в следующей комплектации:

- Набор утилит с расширением *.bat для запуска в среде Windows:
 - transactiontool (осуществляет запуск файла transactiontool-3.4.14.0.jar из директории lib, который предоставляет код транзакции из системы генерации кодов транзакций);
 - identitytool (осуществляет запуск файла identitytool-3.4.14.0.jar из директории lib, который предоставляет сгенерированные идентификаторы сообщений либо файлов вложений);
 - signertool (осуществляет запуск файла signertool-3.4.14.0.jar из директории lib, который предоставляет ЭП-СП содержательной части сообщения или файлов вложений);
 - messagetool (осуществляет запуск файла messagetool-3.4.14.0.jar из директории lib, который формирует и отправляет сообщения в СМЭВ3.x);
 - adminservtool (осуществляет запуск файла adminservtool-3.4.14.0.jar из директории lib, который формирует и отправляет сообщения в административный сервис СМЭВ3.x для переключения на резервный ГРУ).
- Набор утилит с расширением *.sh для запуска в среде Linux:
 - transactiontool (осуществляет запуск файла transactiontool-3.4.14.0.jar из директории lib, который предоставляет код транзакции из системы генерации кодов транзакций);
 - identitytool (осуществляет запуск файла identitytool-3.4.14.0.jar из директории lib, который предоставляет сгенерированные идентификаторы сообщений либо файлов вложений);
 - signertool (осуществляет запуск файла signertool-3.4.14.0.jar из директории lib, который предоставляет ЭП-СП содержательной части сообщения или файлов вложений);
 - messagetool (осуществляет запуск файла messagetool-3.4.14.0.jar из директории lib, который формирует и отправляет сообщения в СМЭВ3.x);
 - adminservtool (осуществляет запуск файла adminservtool-3.4.14.0.jar из директории lib, который формирует и отправляет сообщения в административный сервис СМЭВ3.x для переключения на резервный ГРУ).
- Набор конфигурационных файлов с расширением *. properties:
 - config (содержит настройки приведенных выше файлов);
 - log4j (содержит настройки встроенного сервиса логирования функционирования НИК).

- Директория lib (содержит файлы расширения *.jar, обеспечивающих функционирование НИК).

11.8.2 Требования к ПО для запуска набора

Для обеспечения функционирования набора «toolset» предварительно должно быть развернуто ПО, приведенное в таблице 20.

Таблица 20 – ПО необходимое для функционирования набора «toolset»

№	Категории ПО	Наименование	Версия	Характеристики
1	Виртуальная машина java	Java JRE	1.7.0.5 – 1.7.0.21	ПО исполнения файлов расширения *.jar, находящихся в директории lib.
2	Средство криптозащиты информации	КриптоПро CSP либо КриптоПро JCP	3.6.1 1.0.54	ПО, необходимое для функционирования signertool, adminservtool и messagetool.
3	Библиотека криптоалгоритмов	Trusted Java	2.0	ПО, необходимое для функционирования signertool, adminservtool и messagetool, только в случае использования КриптоПро CSP.

11.8.3 Порядок запуска утилит набора

При запуске любой утилиты, кроме adminservtool, без входных данных либо с неверными входными данными в консоли выведется описание требований к входным данным для запущенной утилиты.

При запуске утилит в среде Windows либо Linux входные данные можно указать:

- либо в командной строке в соответствии с шаблоном: <наименование утилиты> -<параметр 1> -<параметр 2> ... -<параметр n>, например: transactiontool.bat -serviceCode 11111111111111111111 -participantCode 11111111111111111111 -of output/trans.out;
- либо в файле, например: создать файл trans.args (возможно создание текстового файла с любым расширением: например, *.args, *.txt, *.in и т.д.) с содержанием, приведенным на рисунке 38, и запустить следующим образом: transactiontool.bat -argsFile input/trans.args.

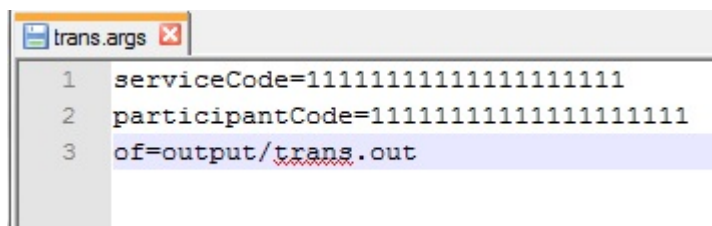


Рисунок 38 – Содержимое файла trans.args с входными данными

В приведенном выше примере необязательный параметр «of output/trans.out» указывает на место, название и расширение создаваемого файла с выходными данными. Без указания данного параметра выходные данные будут выведены в консоль.

11.8.4 Установочная конфигурация

Установочная конфигурация набора «toolset» определяется обязательными настройками, приведенными в таблице 21.

Таблица 21 – Настройки утилит набора «toolset»

№	Наименование настройки	Усл. обозн.	Ед. изм.	Пример
1	Настройки файла config.props			
1.1	Настройки transactiontool			
1.1.1	Адрес web-сервиса системы генерации кодов транзакций	transaction.service.url	-	http://10.215.0.77:8580/tm/Marker?wsdl
1.1.2	Таймаут соединения с web-сервисом системы генерации кодов транзакций	transaction.service.timeout	Миллисекунда	10000
1.2	Настройки messagetool			
1.2.1	Перечень идентификаторов ГРУ СМЭВ3.х	transport.list	-	main, reserve, где: – main – это идентификатор основного ГРУ СМЭВ3.х – reserve – это идентификатор резервного ГРУ СМЭВ3.х
1.2.2	Адрес web-сервиса основного ГРУ СМЭВ3.х	transport.main.url, где main – это идентификатор основного ГРУ СМЭВ3.х	-	http://10.215.0.62:7500/ws?wsdl

№	Наименование настройки	Усл. обозн.	Ед. изм.	Пример
1.2.3	Время, через которое произойдет переключение на резервный ГРУ при условии разрыва установленного соединения с основным ГРУ	transport.main.timeout, где main – это идентификатор основного ГРУ СМЭВ3.х	Миллисекунда	100000
1.2.4	Адрес web-сервиса резервного ГРУ СМЭВ3.х	transport.reserve.url, где reserve – это идентификатор резервного ГРУ СМЭВ3.х	-	http://10.215.0.63:7500/ws?wsdl
1.2.5	Время, через которое произойдет переключение на резервный ГРУ при условии разрыва установленного соединения с основным ГРУ.	transport.reserve.timeout, где reserve – это идентификатор резервного ГРУ СМЭВ3.х	Миллисекунда	100000
1.2.6	Адрес FTP-сервера ФХ СМЭВ3.х	large.attachment.transport.address	-	10.215.0.136
1.2.7	Логин доступа в FTP-сервер ФХ СМЭВ3.х	large.attachment.transport.login	-	anonymous
1.2.8	Пароль доступа в FTP-сервер ФХ СМЭВ3.х	large.attachment.transport.password	-	smev
1.2.9	Количество попыток переотправки файла вложения сообщения	large.attachment.transport.retries	Шт.	3
1.2.10	Таймаут соединения с FTP-сервером ФХ СМЭВ3.х	large.attachment.transport.timeout	Миллисекунда	10000
1.2.11	Контрольная проверка сообщения, отправляемого в СМЭВ3.х (если =0, то проверка отключена; если =1, то проверка включена)	template.validate.input	-	1

№	Наименование настройки	Усл. обозн.	Ед. изм.	Пример
1.2.12	Контрольная проверка сообщения, принятого из СМЭВ3.х (если =0, то проверка отключена; если =1, то проверка включена)	template.validate.output	-	1
1.2.13	Переключатель логирования событий контрольной проверки сообщения, отправляемого в СМЭВ3.х (если =0, то логирование отключено; если =1, то логирование включено)	template.log.input	-	1
1.2.14	Переключатель логирования событий контрольной проверки сообщения, принятого из СМЭВ3.х (если =0, то логирование отключено; если =1, то логирование включено)	template.log.output	-	1
1.2.15	Контрольная проверка ЭП-СМЭВ (если =0, то проверка отключена; если =1, то проверка включена)	template.validate.smev.si gnature	-	1
1.2.16	Параметр сохранения принимаемых вложений сообщений (путь	template.local.storage	-	output/client

№	Наименование настройки	Усл. обозн.	Ед. изм.	Пример
	сохранения файлов вложений)			
1.3	Настройки signertool			
1.3.1	Имя криптопровайдера	signer.provider	-	JCP
1.3.2	Алиас сертификата	signer.certificate.alias	-	dev2
1.3.3	Алиас ключа	signer.private.key.alias	-	dev2
1.3.4	Пароль	signer.private.key.passw ord	-	123456
1.3.5	Место нахождения сертификата	signer.smev.certificate.st ore	-	C:/docs/smev/client_201 5/testData/cert
1.4	Настройки adminservtool			
1.4.1	Адрес административного сервиса	adminservice.url	-	http://localhost:7070/adm serv/AdministrativeServi ce?wsdl
1.4.2	Таймаут соединения с административным сервисом	adminservice.timeout	Миллис екунда	100000
2	Настройки файла log4j.props			
2.1	Обозначение логгеров	log4j.rootLogger	-	STDOUT, FILE
2.2	Имплементация логгера в файл	log4j.appender.FILE	-	org.apache.log4j.Rolling FileAppender
2.3	Наименование и место создания файла логгера	log4j.appender.FILE.File	-	toolset.log
2.4	Максимальный размер файла логгера, при достижении которого создается новый файл логгера	log4j.appender.FILE.Ma xFileSize	Мегабай т	16MB
2.5	Максимальное количество файлов логгера, при достижении которого старые файлы заменяются новыми файлами логгера	log4j.appender.FILE.Ma xBackupIndex	Шт.	256
2.6	Имплементация формата сообщения	log4j.appender.FILE.layo ut	-	org.apache.log4j.Pattern Layout

№	Наименование настройки	Усл. обозн.	Ед. изм.	Пример
	лога, записывающегося в файл логгера			
2.7	Регулярное выражение, описывающее формат сообщения лога, записывающегося в файл логгера	log4j.appender.FILE.layout.conversionPattern	-	[%d{dd.MM.YYYY HH:mm:ss.SSS}] %5p [%t] - %m%n
2.8	Имплементация логгера в консоль	log4j.appender.STDOUT	-	org.apache.log4j.Console Appender
2.9	Имплементация формата сообщения лога, выводящего в консоль	log4j.appender.STDOUT.layout	-	org.apache.log4j.Pattern Layout
2.10	Регулярное выражение, описывающее формат сообщения лога, выводящего в консоль	log4j.appender.STDOUT.layout.conversionPattern	-	[%d{dd.MM.YYYY HH:mm:ss.SSS}] %5p [%t] - %m%n
2.11	Кодировка сообщения лога, выводящего в консоль	log4j.appender.STDOUT.Encoding	-	windows-1251

11.8.5 Описание входных и выходных параметров

Описание входных и выходных данных набора «toolset» приведено в таблице 22.

Таблица 22 – Перечень входных и выходных данных набора «toolset»

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
1	Утилита transactiontool			
1.1	Входные данные			
1.1.1	Код ФРГУ госуслуги/признак госфункции	serviceCode	Обязательный параметр, не более 20 символов	Например: 11111111111111111111
1.1.2	Код ФРГУ информационной системы	participantCode	Обязательный параметр, не более 20 символов	Например: 11111111111111111111

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
1.1.3	Сведения о получателе услуги либо функции	participantDescr	Опциональный параметр, не более 1000 символов	-
1.1.4	Сведения об услуге либо функции	serviceDescr	Опциональный параметр, не более 1000 символов	-
1.2	Выходные данные			
1.2.1	Код транзакции	-	-	Например: d02b9d45-cbe0-11e5-a4f5-86958c02569a11111111111111111111 W94wQVXgoIrcYgReRDzZun+zv62MUTqJG2v7tinVUapvI7cSnXf0jCFcYPEwKGXINzXKQMn0PhEqFwcojnop8OFVBQ9LFfjGGhZko3xJ8Hnl7Ypn/CJgo5MeXyLTSUt6iCHEhSynz764gzorCsXYv4QBql7LwKEx65a7WVT1BB0+nt/Bioy//JHR728nyfFP0ELXvrg41b0Kdc0REtwVfnw2Seg0ELbgaEtrKKN9hQtVDC9x0flotlcZlY4I84IBxGM2SWff68Yy63dp0XZAINqUx+FgHh04vuXaLfmFrIB73kB32iYexmXwjTBiCEKYgEvo+jlJ9/3+cwVPI n24cQ==
2	Утилита messagetool			
2.1	Входные данные			
2.1.1	Вид сообщения	cmd	Обязательный параметр, принимает одно из следующих значений: – sendReq – sendResp – getReq – getResp – sendRejResp	cmd sendReq означает сообщение, передаваемое через метод sendRequest сервиса СМЭВ. cmd sendResp означает сообщение, передаваемое через метод sendResponse сервиса СМЭВ. cmd getReq означает сообщение,

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
			<ul style="list-style-type: none"> – sendStatusResp – ack – getStatus 	<p>передаваемое через метод getRequest сервиса СМЭВ.</p> <p>cmd getResp означает сообщение, передаваемое через метод getResponse сервиса СМЭВ.</p> <p>cmd sendRejResp означает сообщение об отклонении запроса, передаваемое через метод sendResponse сервиса СМЭВ.</p> <p>cmd sendStatusResp означает статусное сообщение, передаваемое через метод sendResponse сервиса СМЭВ.</p> <p>cmd ack означает сообщение, передаваемое через метод ack сервиса СМЭВ.</p> <p>cmd getStatus означает сообщение, передаваемое через метод getStatus сервиса СМЭВ.</p>
2.1.2	Блок бизнес-данных сообщения	if	Обязательный параметр для cmd sendReq и cmd sendResp. Принимает путь к файлу формата *.xml с блоком бизнес-данных сообщения.	Например: if=testdata/request.xml
2.1.3	Код транзакции	tid	Обязательный параметр для cmd sendReq.	Например: tid=1bbe12f1-ded5-11e5-a9db-86958c02569a 1234567890

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
			Принимает код транзакции, полученный из системы генерации кодов транзакций средствами утилиты transactiontool	1234567890jOW0eLrHN7j yiQ6bsfbuA/NK3R0CEdD 7Rqm2smNNgflMgsb8xdO YzExucVQuA3ZhFZ2T4kL UJQGfGgsZZZzYcHR/Zw/ ZVQd4sUoifa9xKBtrVOL BR997TLea43rjn1PrGvw8 136spSMON9kjpzhkgSAB Lanww+Aeb/TsHg+gdmbi 2q8joWg9gKbx5AP2jCNV F4yaUSFUutMKvN3rAg+ km7p88zR1xepzq0+f3jr8j WhsbUnwoNDXcqVY0h0 gS1ZayeRL5nndodzGTNe U0WKvt7pCtHltj4zCppG3 olArgirk3gkJ7C4di6jnxgST uHokdoh4g0C9qRU4fFO2 mpmjQ==
2.1.4	Файлы вложения, передаваемые вместе с сообщением	mtomAtts	Опциональный параметр для cmd sendReq и cmd sendResp. Принимает пути к файлам, передаваемым вместе с сообщением. Пути к файлам приводятся через «;».	Например: mtomAtts=testdata/mtom1.txt;testdata/mtom2.txt
2.1.5	Файлы вложения, передаваемые отдельно от сообщения	largeAtts	Опциональный параметр для cmd sendReq и cmd sendResp. Принимает пути к файлам, передаваемым отдельно от сообщения. Пути к файлам приводятся через «;».	Например: largeAtts=testdata/large_file1.jpg;testdata/large_file2.jpg
2.1.6	Информация о бизнес-процессе, в рамках	bpm	Опциональный параметр для	Например: bpm=testdata/bpm1.xml

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
	которого пересылается данное сообщение.		cmd sendReq. Принимает путь к файлу формата *.xml с описанием бизнес-процесса.	
2.1.7	Время жизни сообщения	eol	Опциональный параметр для cmd sendReq. Принимает значение даты в формате ДД.ММ.ГГГГ ЧЧ:ММ:СС.	Например: eol=07.10.2015 18:43:00
2.1.8	ЭП-СП файлов вложения или содержательной части сообщения	ps	Опциональный параметр для cmd sendReq и cmd sendResp. Принимает путь к файлу формата *.xml с ЭП-СП, полученного средствами утилиты signertool	Например: ps=testdata/personal_signature.xml
2.1.9	Обратный адрес	replyTo	Обязательный параметр для cmd sendResp, cmd sendRejResp и cmd sendStatusResp. Принимает значение обратного адреса.	Например: replyTo=eyJzaWQiOiJmM3L CJtaWQiOiJmMWNhOGNj NC02Y2RjLTExZTUtOTU xMi1kNGM5ZWZmM2I1 OTQiLCJ0Y2QiOiJmNTky Nzc5Ni1lMWFILTQ4ZmE tYjE0MS1kNzkwYzlhMD ZiMTd8MjAxNS0wNC0y NFQwOTozNzo1MS44OT h8ZnJndTJ8VHZjWjF4T2 VnTEgrTVpyVXRIdWpBd CtRcHlxWmh6RHVTSHN xbkJPYkkraWJzN3k0MGZ jODhISkxyQy9RSW1hTV YvYmNYcHBsUkh4NWpz S294UVpEREV6UINmMD

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
				NRK1ROa1lCRms0L1FtZ W1NNFVHYmpZY2FQO U16YzBXT0QzblzYXdzZ jhzWU96VWpQclV5cTZP enhiZk8rK01Fd0psYmpjd WtXSDZQNUFFWTF4UU 0wbHRPdGhZQ2h6Ul4S WQ0RGFGanBkWEZZMit ITzFxOWhsazlFOVZBZnd tTHIXMGJ4MTBmTGM2 bVBUSUF2ajJ6QjVPV010 cW5ydjJobUF1Qm9WQjJ WWXI5TE4wT3VEcWZ5 MkRpb3NzSjdYVWc1cHh RTnphdWtpb1Nxc2dZSW VNSkhrZ1U2MXBPZlZ5c mdRWXBJZEo3UHM0dXl leFh3SXXJME5WdDFGYX g3YmhDbLhSRFZpZ3RxN EYwaVA0Tk9vSTd0L1pp WIFWVDkzQ2FKL2hON2 lpcXVFb3piUkRRZ3Ayd1 pRbktMai8zRjZuaUlxSFJo ZFFTMDFmcEJoV28xbEZ kNDhySk1hMkplL2ovemQ zeHJDa2RIT2p3NGJySnBl bXZIMzFWODI5SFM5OF ZTMk93YVRuL2dTRjY5 Yi91S2QzbFICN3pjMGVt NDQ5OUt3Ym1DcUIVT1 BvU1JVWldiL2RCcEczV DBEb2c3MCtIS3JoT3ljY2 hCUDdUZU10Q09SZU9n YUFGa0FHdHVicGNBL1 VJUEkrdkZDUkpQV1lyK 0IMUDRXU3NpektzVi9E QzdZSmdpcmE0dzJ1RHZ XOTIUWIY3dm5sOTltUE JiQnloTm9tL3gyN2E2U0t oK3MrSitLaXQ0SGV6a1V GMGxUeUdwWkZzU0w0 dW43RmVxMTg9liwicmlk IjoiYzFjYThjYzQtNmNkY y0xMWU1LTk1MTItZDRj OWVmZjNiNTk0liwiZW9 sIjoxNDQ0MjE0MTEwLC JuaWQiOiIxliwic2xjIjoieC

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
				1hcnRlZmFjdHMtc2lldiInb3YtcnVfdGVzdGluZ19sb2FkIn0=
2.1.10	Идентификатор узла инфраструктуры ИС	nodeId	Опциональный параметр для cmd= getReq и cmd= getResp. Принимает значение идентификатора узла инфраструктуры ИС.	Например: nodeId=someNode
2.1.11	Идентификатор вида сведений	rts	Опциональный параметр для cmd= getReq и cmd= getResp. Принимает значение идентификатора вида сведений.	Например: rts={urn://x-artefacts-smev-gov-ru/services/message-exchange/1.1}localName Request
2.1.12	Код статуса	code	Обязательный параметр для cmd=sendRejResp и cmd=sendStatusResp. Принимает значение кода статуса.	Например: code=1
2.1.13	Описание статуса	desc	Обязательный параметр для cmd=sendRejResp и cmd=sendStatusResp. Принимает описание статуса.	Например: desc=Some description
2.1.14	Идентификатор сообщения (запроса либо ответа)	msgId	Обязательный параметр для cmd=ack.	Например: msgId=7d2799a4-6cd1-11e5-8561-

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
			Принимает значение идентификатора сообщения.	2c44fd105812
2.2	Выходные данные			
2.2.1	Синхронный ответ на запрос, включающий его метаданные и данные сообщений из очередей доставки либо статистики входящих очередей.	-	-	<p>Для cmd sendReq, cmd sendResp, sendRejResp и cmd sendStatusResp возвращается либо ошибка обработки сообщения, либо уведомление о постановке сообщения в асинхронную обработку.</p> <p>Для cmd getReq и cmd getResp возвращает сообщение из очереди доставки либо ошибку обработки запроса.</p> <p>Для cmd ack возвращает ответное сообщение либо ошибку обработки запроса.</p> <p>Для cmd getStatus возвращает сообщение из статусной очереди либо ошибку обработки запроса.</p>
2.2.1.1	Данные сообщений из очередей доставки либо статистики входящих очередей.	-	-	<p>Возвращается «NO_MESSAGE» в результате:</p> <ul style="list-style-type: none"> – ошибки обработки запроса – отсутствия сообщений в очередях доставки – запрос отправлен sendReq,

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
				sendResp, ack и sendRejResp В остальных случаях возвращается xml-строка сообщения.
2.2.1.2	Метаданные в составе:			
	путь к файлу с дополнительными свойствами	map <имя файла>	-	
	идентификатор сообщения, в ответ на которое отсылается текущее	origMsgId <идентификатор>	В соответствии с RFC-4122	Например: 29f2c3e6-ce71-11e5-bca4-08606ee4e99b
	идентификатор исходящего сообщения	refMsgId <идентификатор>	В соответствии с RFC-4122	Например: 29f2c3e6-ce71-11e5-bca4-08606ee4e99b
3	Утилита identitytool			
3.1	Входные данные			
3.1.1	Вид запроса	cmd	Обязательный параметр, принимает одно из следующих значений: <ul style="list-style-type: none"> – msgUuid – attachUuid – attachId 	Если необходимо запросить идентификатор сообщения, то запрос делается с параметром msgUuid . Например: identitytool.bat -cmd msgUuid Если необходимо запросить идентификатор файла вложения сообщения, то запрос делается с параметром attachUuid . Например: identitytool.bat -cmd attachUuid. Если необходимо запросить наименование файла вложения сообщения, то запрос делается с параметрами attachId ,

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
				mime (mime-тип файла, определяющее его расширение) и num (порядковый номер). Например: identitytool.bat -cmd attachId -mime "text/plain" -num 1.
3.2	Выходные данные			
3.2.1	Идентификатор сообщения	-	В соответствии с RFC-4122	Например: 29f2c3e6-ce71-11e5-bca4-08606ee4e99b
3.2.2	Идентификатор файла вложения сообщения	-	В соответствии с RFC-4122	Например: 7caf8c1c-c485-4265-b6f0-4680d7aca4f0
3.2.3	Наименование файла	-	-	Например: __ATT_ID_SMEV_C_AU TOGEN__1.txt
4	Утилита signertool			
4.1	Входные данные			
4.1.1	Вид запроса	cmd	Обязательный параметр, принимает одно из следующих значений: – signXml – signHash – digest – validate	cmd signXml означает запрос на получение ЭП-СП блока бизнес данных сообщения. cmd signHash означает запрос на получение ЭП-СП хэша файла. cmd digest означает получение хэша файла. cmd validate означает получение данных результатов проверки целостности блока бизнес-данных сообщения.
4.1.2	Файл ресурса для обработки	if	Обязательный параметр.	Например: для cmd signXml: signertool.bat -cmd signXml -if testdata/request.xml для cmd signHash: signertool.bat -cmd

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
				<p>signHash -if testdata/hash.file</p> <p>для cmd digest: signertool.bat -cmd digest -if testdata/binary.file</p> <p>для cmd validate: signertool.bat -cmd validate -if testdata/request.xml -sig testdata/signature.xml</p>
4.1.3	ЭП-СП блока бизнес данных сообщения	sig	Обязательный параметр для validate	Например: signertool.bat -cmd validate -if testdata/request.xml -sig testdata/signature.xml
4.2	Выходные данные			
4.2.1	ЭП-СП блока бизнес данных сообщения	-	-	
4.2.2	ЭП-СП хэша файла	-	-	
4.2.3	Хэш файла	-	-	
4.2.4	Результат проверки целостности блока бизнес данных сообщения	-	-	
5	Утилита adminservtool			
5.1	Входные данные			
5.1.1	Отсутствуют	-	-	-
5.2	Выходные данные			
5.2.1	Ответ о приеме запроса на переключение	SwitchResponse	-	
5.2.2	ЭП-ОВ не прошла проверку	SignatureVerificationFault	-	
5.2.3	ИС отправителя не зарегистрирована в СМЭВ	SenderIsNotRegisteredFault	-	
5.2.4	ИС отправителя установлена блокировка на переключение	SenderIsSwitchBlockedFault	-	

№	Наименование данных	Усл. обозн.	Ограничения	Комментарий
5.2.5	Ошибка СМЭВ	SMEVFailure	-	