

«Программа для токенизации персональных данных»

Аннотация

Настоящий документ является общим описанием Автоматизированной информационной системы «Программа для токенизации персональных данных».

Документ содержит общее описание, сведения о назначении и функциональных подсистем Системы.

Перечень сокращений

Термин	Определение
А/к	Авиакомпания
Бронирование (резервирование)	Предварительное закрепление за пассажиром места на воздушном судне на определенный рейс и дату или предварительное закрепление объема и тоннажа для перевозки багажа, груза на воздушном судне
ВВЛ	Внутренние воздушные линии
МВЛ	Международные воздушные линии
ПД	Персональные данные
Токен	Комбинация символов, которая используется для шифрования информации
ДПД	Детокенизация ПД
Инвенторная система (Inventory System)	База данных с информацией о пассажирах, рейсах, наличии мест, тарифах и допуслугах перевозчика, а также данные о взаиморасчетах между всеми участниками процесса продажи билетов
Хост, Хост а/к	Раздел а/к в инвенторной системе бронирования
DCS	Departure Control System, система регистрации пассажиров/багажа и управления отправлениями аэропорта
PNR	Код бронирования в инвенторной системе. Представляет собой шестизначный буквенный код.
VCR	Запись об электронном билете в системе бронирования

1. Назначение системы

1.1 Вид деятельности, для автоматизации которой предназначена Система

Вид деятельности, для автоматизации которой было разработано программное обеспечение (Система) – услуги по обеспечению сохранности персональных данных граждан РФ, приобретающих авиаперевозку авиакомпаний РФ, при использовании авиакомпаниями РФ зарубежных систем резервирования авиакомпаний (инвенторных систем).

Система создана в целях построения современной инфраструктуры, обеспечивающей возможность продажи и оформления авиаперевозок с поддержкой кириллицы авиакомпаниями и агентами РФ (независимо от поддержки кириллицы инвенторной системой), а также сохранность персональных данных граждан РФ на территории РФ, в соответствии с требованиями ФЗ № 955 от 24.07.2019.

Система представляет собой полное, масштабируемое и современное решение, которое обеспечивает обработку трафика, направляемого системами авиакомпаний, аэропортов и агентов в инвенторные системы и направляемого инвенторными системами в авиакомпании, аэропорты и агентства.

Система предоставляет следующие возможности:

- контроль на наличие ПД пассажиров онлайн транзакций бронирования и оформления перевозок, направляемых от систем авиакомпаний в инвенторные системы (зарубежные);
- генерация токенов для ПД пассажиров для следующих данных: имя, фамилия, отчество, номер паспорта, адрес электронной почты, номер телефона;
- замена открытых ПД на токены в транзакциях бронирования и оформления перевозок, направляемых от систем авиакомпаний в инвенторные системы (зарубежные);
- контроль на наличие токенов онлайн транзакций при бронировании и оформлении перевозок, направляемых от инвенторных систем (зарубежных) в системы авиакомпаний;
- замена токенов на открытые ПД в транзакциях бронирования и оформления перевозок, направляемых от инвенторных систем (зарубежных) в системы авиакомпаний;
- контроль на наличие токенов в пакетных и потоковых файлах, направляемых от инвенторных систем (зарубежных) в системы авиакомпаний;
- замена токенов на открытые ПД в пакетных и потоковых файлах, направляемых от инвенторных систем (зарубежных) в системы авиакомпаний;
- хранение токенов и открытых ПД на территории РФ.

1.2 Перечень объектов автоматизации, на которых используется Система

Объектом автоматизации является процесс деперсонализации ПД пассажиров, граждан РФ, при продаже авиаперевозок авиакомпаний РФ, использующих зарубежные инвенторные системы.

1.3 Перечень функций, реализуемых Системой

- анализ типа перевозки (МВЛ, ВВЛ)
- анализ типа гражданства пассажира (РФ, не РФ)
- анализ типа запроса в хост а/к
- анализ типа ответа хоста а/к
- поиск ПД пассажира в запросе в хост а/к
- поиск ПД пассажиров в ответе хоста а/к
- формирование уникальных токенов
- замена ПД на токены и наоборот
- обработка процедур изменения маршрута в бронировании (МВЛ-ВВЛ и наоборот)
- обработка процедур разделения PNR
- обработка запросов на получение ПД пассажиров.

2. Описание системы

Система создаётся в целях выполнения требований ФЗ № 955 в части сохранности ПД граждан РФ при бронировании и выполнении перевозок.

Система представляет собой полное, масштабируемое и современное решение, которое обеспечивает безопасное использование авиакомпаниями РФ зарубежных инвенторных систем без передачи ПД пассажиров за территорию РФ.

2.1 Структура Системы и назначение её частей

По виду автоматизированной деятельности Система представляет собой многопользовательскую территориально распределенную информационно-телекоммуникационную систему обработки и передачи информации с различным уровнем доступа пользователей к обрабатываемой информации, содержащей персональные данные пассажиров.

Система состоит из следующих функциональных частей:

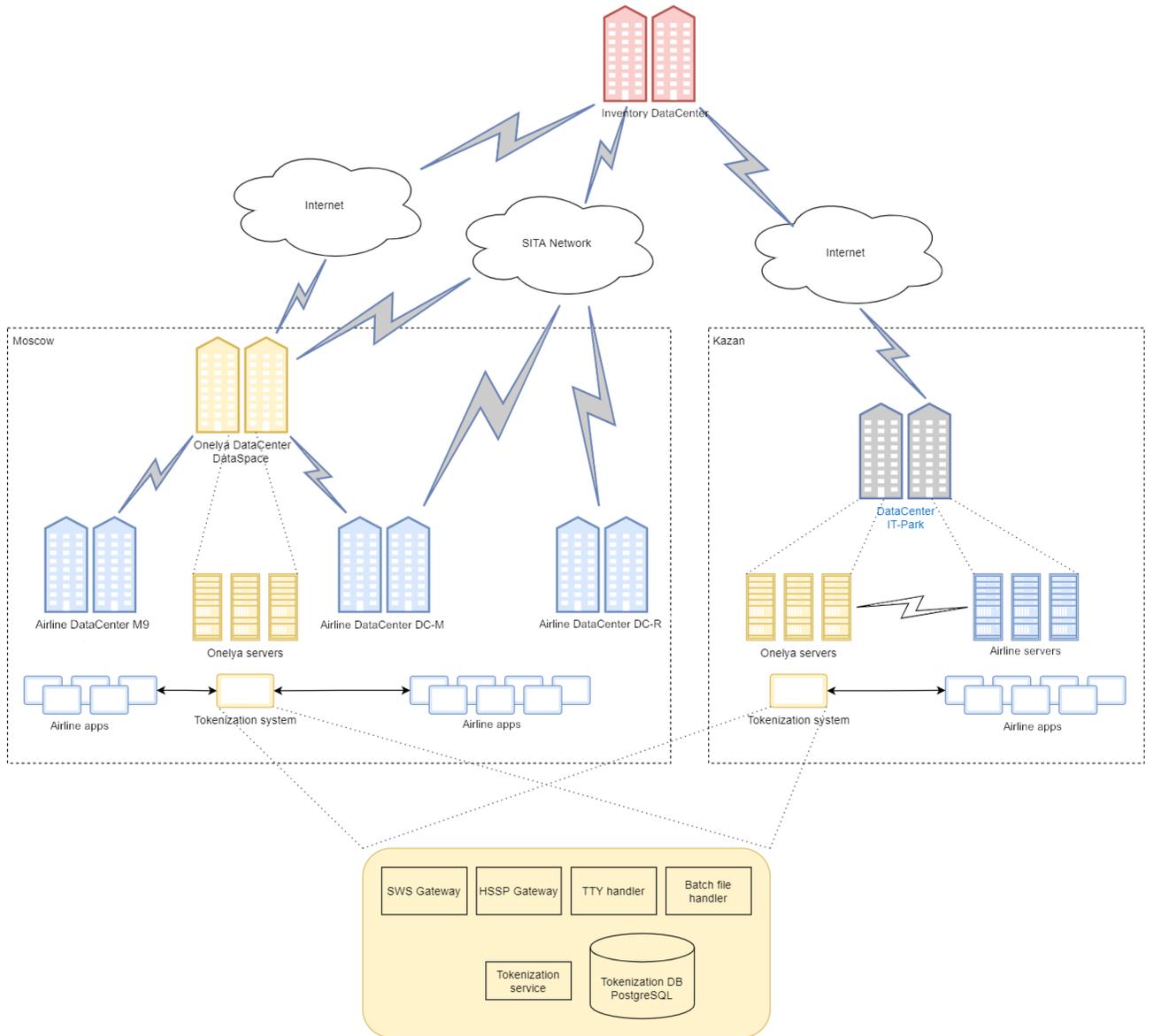
- **модуль обработки онлайн трафика** – выполняется анализ запросов от фронтальных систем а/к в хосты а/к на предмет наличия в запросах открытых ПД, замена таких данных на токены, направление запросов с токенами в хосты а/к; выполняется анализ ответов от хостов а/к во фронтальные системы а/к на предмет наличия токенов, замена токенов на открытые ПД и направление запросов с открытыми ПД во фронтальные системы а/к;
- **модуль обработки трафика бэк-офиса** – выполняется анализ пакетных файлов и потоковых выгрузок из хостов а/к в бэк-офисные системы а/к, замена токенов на открытые ПД для дальнейшей загрузки в бэк-офисные системы а/к ;
- в хосты а/к, замена открытых ПД на токены для дальнейшей отправки в хосты а/к;
- **модуль обработки терминального трафика** – выполняется анализ запросов от терминалов а/к (офисы продаж, колл-центры, технологические службы) в хосты а/к на предмет наличия ПД, выполняется замена ПД на токены и направление запросов в хосты а/к; выполняется анализ ответов от хостов а/к в терминалы а/к на предмет наличия токенов, выполняется замена токенов на открытые ПД и направление ответов в терминалы а/к;
- **модуль формирования токенов** – выполняется генерация уникальных наборов латинских или кириллических символов (токенов) для фамилии/имени/отчества пассажира каждого пассажира в бронирования (заказа), выполняется генерация токенов для адреса электронной почты заказа, выполняется генерация токенов для номера телефона и номера паспорта;
- **модуль обработки запросов от внешних сервисов** – rest-сервис, принимает запрос на получение открытых ПД от внешней системы, формирует ответ с открытыми ПД в соответствии с признаками, указанными в запросе;
- **база данных токенов и открытых ПД** – хранилище записей о токенах и соответствующих ПД пассажиров, маршрутах, датах и номерах рейсов, указанных в

PNR.

2.2 Обобщенная схема решения



2.3 Схема размещения и интеграции



2.4 Сведения о технологическом стеке.

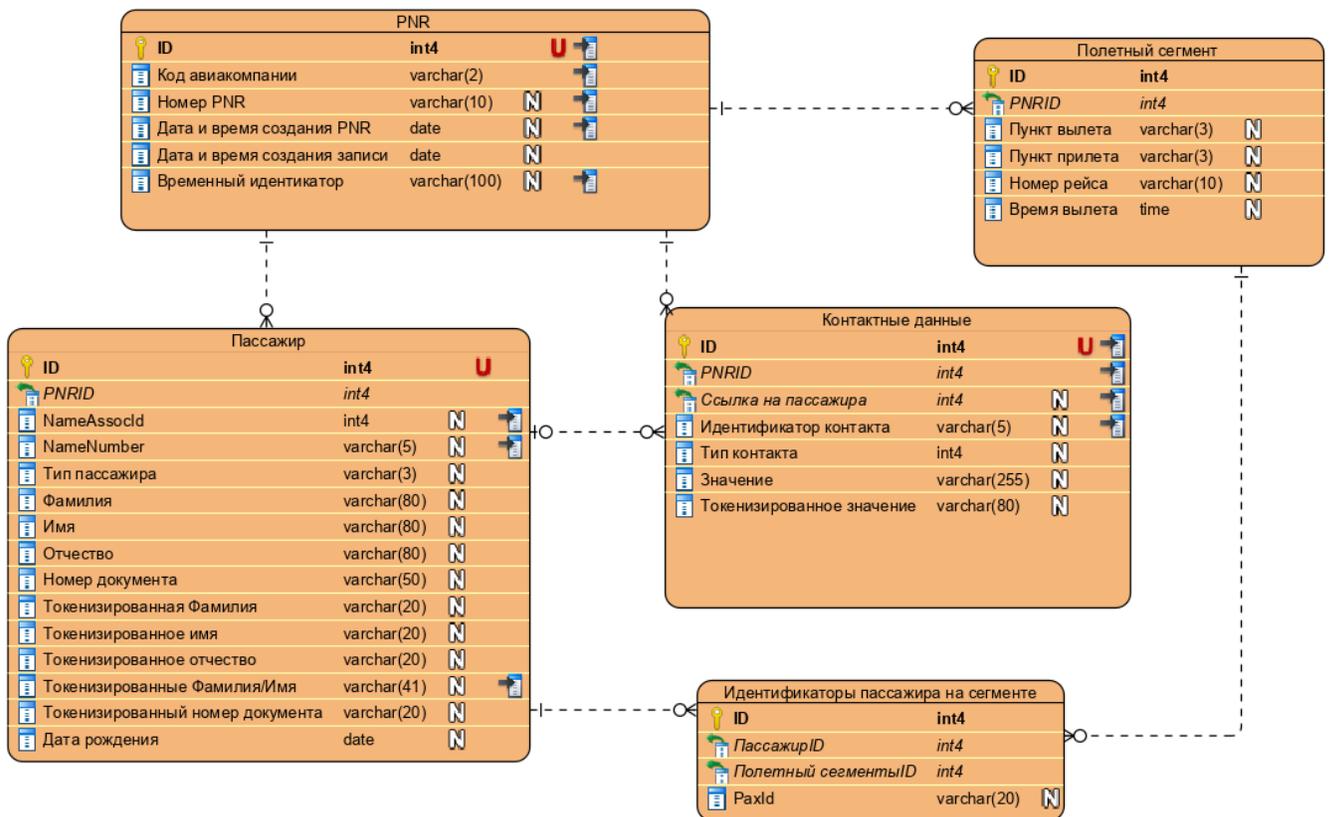
При разработке решения использованы следующие средства:

- платформа решения: .net Core
- язык разработки: C#
- база данных для хранения токенов и открытых ПД: PostgreSQL
- система кэширования данных: Redis
- операционная система: Lunix
- контейнер для приложения: Docker
- управление работой приложений (оркестровка): Kubernetes

3. Сведения о функционале решения

Представлена детальная информация о процедуре токенизации ПД, шлюзе обработки веб-сервисов и перечне бизнес-процессов которые обрабатываются процедурой токенизации.

3.1 Структура базы данных токенов и ПД.



3.2 Алгоритмы токенизации

Фамилия Имя Отчество

ФИО пассажира - единственное поле в бронировании, для которого необходимо иметь гарантировано глобально уникальный токен, поскольку при определенных обработках это поле может быть единственным идентификатором, по которому возможно будет идентифицировать пассажира и выполнить детокенизацию.

1. Сгенерировать UUID v1.
2. Получить строковое значение в виде **9d17b524-75e7-4fa2-819c-1b0b6ceb4cd9**
3. Удалить все символы, кроме цифр и букв: **9d17b52475e74fa2819c1b0b6ceb4cd9**
4. Привести все буквы в верхней регистр: **9D17B52475E74FA2819C1B0B6CEB4CD9**
5. Заменить все цифры на буквы в соответствии с таблицей

Цифра	Буква
0	Z
1	Y
2	X
3	W
4	V
5	U
6	T
7	S
8	R
9	Q

6. Или, что то же самое, выполнить ASCII-преобразование:
7. (char) (90-digit) **QDYSBUXVSUESVFAXRYQCYBZBTCEBVCDQ**
8. Полученную строку разбить на части в зависимости от состава исходных данных:
 - если в исходных данных есть отчество, то разбивается на три части:
 - 8 символов - имя: **QDYSBUXV**
 - 10 символов - отчество: **SUESVFAXRY**
 - 14 символов - фамилия: **QCYBZBTCEBVCDQ**

- если в исходных данных отчество отсутствует, то разбивается на две части:
 - 15 символов - имя: **QDYSBUXVSUESVFA**
 - 17 символов - фамилия: **XRYQCYBZBTCEBVCDQ**
9. К токену фамилии добавляется префикс **ZZZ**. Это необходимо для того, чтобы сторонние системы могли определить, что ФИО было токенизировано:
- фамилия: **ZZZQCYBZBTCEBVCDQ**
или
 - фамилия: **ZZZXRYQCYBZBTCEBVCDQ**
10. При наличии нескольких пассажиров в брони необходимо обеспечить уникальность каждой из частей токена ФИО в брони. Т.е. токен фамилии, токен имени и токен отчества не могут повторяться в пределах одной брони.

Дата рождения

Реальную дату рождения пассажира нельзя модифицировать при сохранении в системе бронирования, т.к. дата рождения определяет тип пассажира (ADT, CHD, INF). При этом тип пассажира определяется с учетом дат перелета. Например, младенцем (INF) может считаться только тот пассажир, которому будет не более 2 полных лет на дату первого вылета. В случае со взрослым пассажиром, в случае сопровождения ребенка, возраст взрослого пассажира должен быть не менее 18 лет на дату первого вылета. Возможны и более сложные ограничения на дату рождения. С учетом большого количества подобных нюансов и ограничений заменять дату рождения пассажира не представляется возможным. Таким образом токенизация даты рождения не выполняется.

Номер документа

1. Получить текущее время в формате UNIX Timestamp, т.е. количество секунд с 01.01.1970
2. Преобразовать в строку: **1588166284**

Дата окончания действия документа

При токенизированном номере документа нет необходимости заменять реальную дату окончания документа на реальную.

Номер телефона

1. Сгенерировать случайный набор из 7 цифр: **9382648**
2. Добавить фиксированный префикс **000** (необходим для идентификации того, что номер телефона токенизирован): **0009382648**

Адрес электронной почты

1. Заменить имя почтового ящика на строку, состоящую из префикса ZZZ и случайного набора из 10 символов. Заменить исходное доменное имя сервера на oneliya.ru: user@gmail.com => ZZZFAYAAJAFDL@ONELIYA.RU.

3.3 Идентификация пассажиров

Общее описание

В различных интерфейсах взаимодействия с инвенторной системой используются разные способы идентификации пассажиров. При этом некоторые идентификаторы становятся известны не сразу, а только в процессе обработки определенных запросов Sabre Web Services. В связи с этим возникает необходимость хранения для каждого пассажира нескольких идентификаторов, а также метода для их актуализации. При выполнении операций токенизации/детокенизации может использоваться один из вариантов идентификации, в зависимости от их наличия в исходном запросе/ответе. Далее перечислены возможные виды идентификаторов в порядке убывания приоритета их использования в случае наличие в исходном обрабатываемом документе нескольких идентификаторов.

Виды идентификаторов

nameAssocId

Целочисленный идентификатор, как правило, соответствует порядковому номеру пассажира в брони. Пример:

```
<stl19:Passenger id="3" nameType="S" referenceNumber="ADT" nameAssocId="1" nameId="01.01" elementId="pnr-3.1">
```

nameId

Строковый идентификатор вида "01.01", "02.01", "02.02" и т.д. Пример:

```
<stl19:Passenger id="3" nameType="S" referenceNumber="ADT" nameAssocId="1" nameId="01.01" elementId="pnr-3.1">
```

raxId

Строковый уникальный идентификатор пассажира на рейсе, присваивается на этапе регистрации. Если в бронировании несколько сегментов, то на каждом из них одному и тому же пассажиру будет присвоен отдельный идентификатор. Пример:

```
<PassengerID>F120CE320001</PassengerID>
```

raxNameTkn

Искусственный строковый идентификатор, который используется в тех случаях, когда в теле исходного документа отсутствуют какие-либо из вышеперечисленных идентификаторов. Формируется путем конкатенации токенизированных Фамилии и Имени пассажира, разделенных символом "/". Пример:

```
<ns2:LastName>NIKORAGUA</ns2:LastName> + '/' + <ns2:FirstName>NIKOLAY IVANOVICH SR</ns2:FirstName> = NIKORAGUA/NIKOLAY IVANOVICH SR
```

3.4 Шлюз обработки веб-сервисов.

Краткое описание

Шлюз осуществляет проксирование сервисов инвенторной системы (Sabre Web Services, SWS) для токенизации/детокенизации персональных данных, которые могут передаваться в запросах/ответах сервисов. При этом как правило, для запросов выполняется процедура токенизации, для ответов - детокенизация. Токенизации/детокенизации подлежат только запросы/ответы определенных сервисов. Если запрос/ответ сервиса не содержит персональных данных, он не обрабатывается, т.е. передается в неизменном виде. Также необходимо иметь в виду, что токенизации подлежат только персональные данные граждан РФ в случае авиаперевозки только по территории РФ (ВВЛ). Персональные данные пассажиров, не являющихся гражданами РФ, не токенизируются. Также не токенизируются ПД граждан РФ при международных перелетах (МВЛ). Если перелет содержит хотя бы один международный сегмент, он считается МВЛ, соответственно, данные пассажиров не токенизируются.

Шлюз должен иметь внешний endpoint (без авторизации), на который системы а/к будут направлять запросы, вместо SWS. Все запросы/ответы передаются в формате XML. Все заголовки, полученные вместе с запросом/ответом должны передаваться в SWS/вызывающей системе в неизменном виде (за исключением заголовка PAX-NATIONALITY (см. Бронирование, заголовок передается при вызове TravelItineraryAddInfoLLSRQ/RS), который должен быть удален при передаче в SWS).

Процесс обработки сервисов SWS можно разбить на два блока, отличающихся алгоритмом работы:

- Создание бронирования. При этом добавление в бронирование маршрута и пассажиров выполняется разными сервисами, поэтому требуется дополнительная логика обработки.
- Действия с уже созданным бронированием (чтение, изменение, обмен, возврат, регистрация и т.д.). При выполнении этих действий сначала вызывается сервис чтения брони (GetReservationRQ), а потом уже сервисы действий.

Общий алгоритм работы с сервисами SWS со стороны вызывающих систем выглядит следующим образом:

1. Вызывается сервис SessionCreateRQ, который открывает сессию и возвращает ее токен.
2. Последовательно вызываются необходимые сервисы, при этом в заголовке каждого запроса передается токен, полученный при вызове SessionCreateRQ. SWS считает,

что все эти сервисы вызываются в рамках одной сессии, т.е. являются взаимосвязанными. Если первым после открытия сессии вызывается GetReservationRQ, то считается, что все последующие сервисы вызываются в отношении того PNR, который был указан в GetReservationRQ. Все вызовы в рамках одной сессии выполняются с одним и тем же токеном.

3. После вызова всех значимых сервисов, отправляется запрос SessionCloseRQ, который закрывает текущую сессию. Также сессия закрывается автоматически, если прошло более 30 минут с момента отправки последнего запроса с токеном сессии.

Создание бронирования

Алгоритм описан в разделе 3.5.

Различные действия с бронированиями

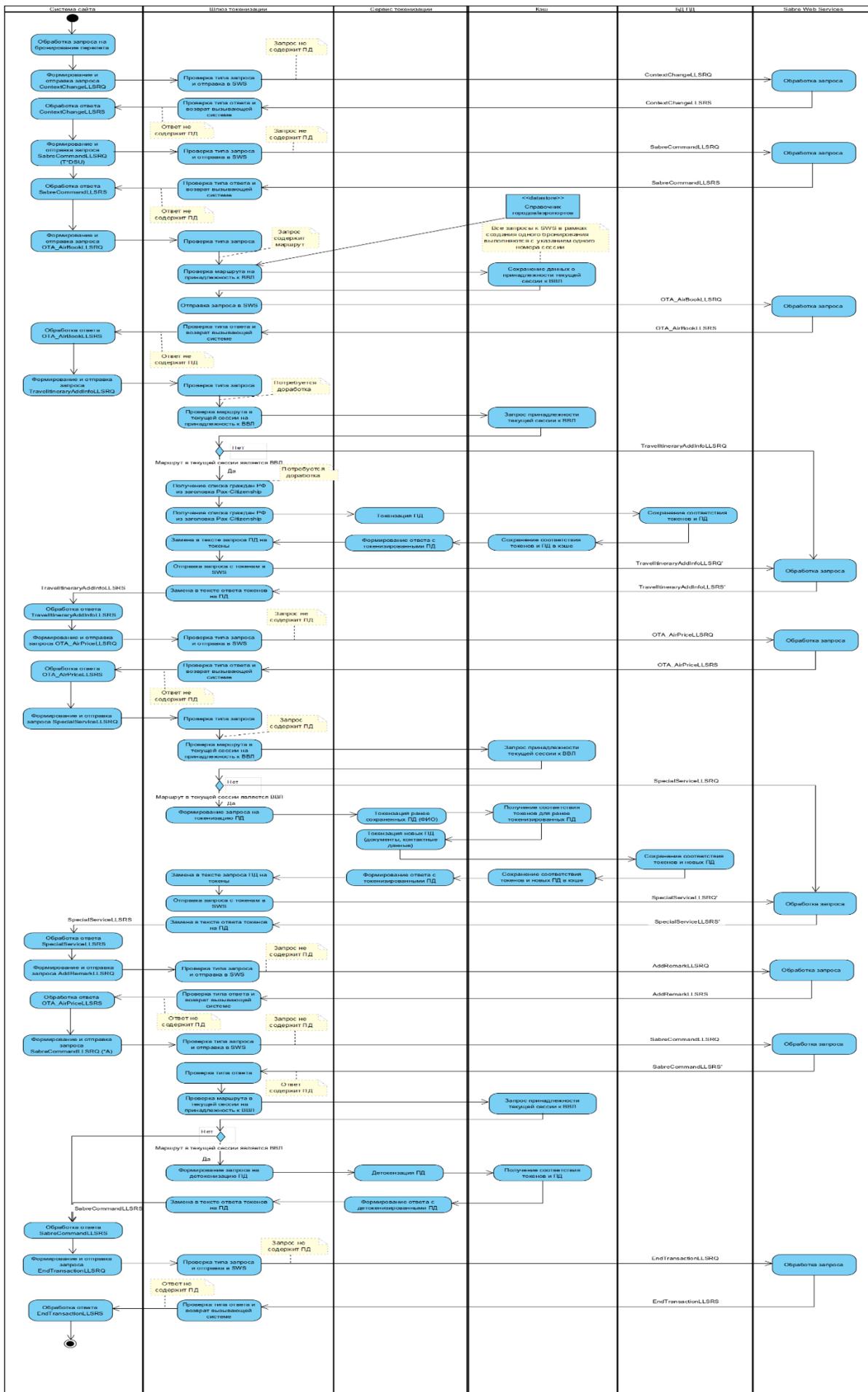
1. Система а/к отправляет SWS-запрос на endpoint шлюза.
2. Шлюз проверяет тип запроса на необходимость выполнения токенизации.
3. Если тип запроса не предполагает наличие персональных данных, запрос отправляется в SWS без изменений.
4. Если тип запроса предполагает наличие персональных данных, шлюз выполняет их замену:
 - a. В теле запроса выбираются все поля, содержащие персональные данные
 - b. Для всех наборов полей, относящихся к пассажиру или контактным данным, выполняется токенизация путем вызова метода /reservation/Mask Веб-сервиса токенизации ПД.
 - c. В теле исходного запроса выполняется замена открытых ПД на токенизированные, полученные в ответе метода /reservation/Mask.
5. Если в качестве запроса выступает GetReservationRQ, необходимо сохранить в кэше пару "токен сессии" - "номер PNR". Токен сессии получить из элемента /Envelope/Header/Security/BinarySecurityToken запроса. Номер PNR - из элемента /Envelope/Body/GetReservationRQ/Locator. Это необходимо для корректной токенизации последующих запросов, вызываемых в рамках данной сессии, т.к. они не всегда содержат номер PNR в явном виде, поэтому его нужно извлекать из кэша по ключу - токenu сессии.
6. Шлюз отправляет модифицированный запрос в SWS.
7. Шлюз получает ответ от SWS.
8. Шлюз проверяет тип ответа на необходимость выполнения детокенизации.
9. Если тип ответа не предполагает наличие персональных данных, ответ возвращается вызывающей системе без изменений.

10. Если тип ответа предполагает наличие персональных данных, шлюз выполняет их замену:

- a. В теле ответа выбираются все поля, содержащие токенизированные персональные данные
- b. Для всех наборов полей, относящихся к пассажиру или контактному данным, выполняется детокенизация путем вызова метода /reservation/Unmask Веб-сервиса токенизации ПД.
- c. В теле исходного ответа выполняется замена токенизированных ПД на детокенизированные (т.е. открытые), полученные в ответе метода /reservation/Unmask.

11. Шлюз возвращает модифицированный ответ вызывающей системе.

3.5 Создание бронирования



Последовательность вызова сервисов SWS при выполнении бронирования на сайте авиакомпании:

№ п/п	Сервис	Описание	Наличие ПД в за-просе	Наличие ПД в от-вете
1	SessionCreateRQ/RS	Создание сес-сии	-	-
2	SabreCommandLLSRQ/RS	Выполнение ко-манды T*DSU	-	-
3	OTA_AirBookLLSRQ/RS	Добавление сег-ментов в бронь	-	-
4	<u>TravelItineraryAddInfoLLSRQ/RS</u>	Добавление пас-сажиров в бронь	ФИО Email Телефон	ФИО Email Телефон
5	OTA_AirPriceLLSRQ/RS	Расчет прай-синга по брони	-	-
6	<u>SpecialServiceLLSRQ/RS</u>	Добавление эле-ментов SSR в бронь	ФИО Номер доку-мента Телефон	ФИО Номер до-кумента Телефон
7	AddRemarkLLSRQ/RS	Добавление ре-марок в бронь	-	-
8	<u>SabreCommandLLSRQ/RS</u>	Выполнение ко-манды *A	-	ФИО Номер до-кумента Телефон
9	EndTransactionLLSRQ/RS	Завершение со-здания брони	-	-
10	SessionCloseRQ/RS	Закрытие сес-сии	-	-

Определение принадлежности маршрута к ВВЛ

Полетные сегменты добавляются в маршрут сервисом `OTA_AirBookLLSRQ/RS`. Необходимо построить список всех пунктов на маршруте, извлекая их из элементов:

- `/Envelope/Body/OTA_AirBookRQ/OriginDestinationInformation/FlightSegment/OriginLocation` - пункты вылета (IATA-код города или аэропорта)
- `/Envelope/Body/OTA_AirBookRQ/OriginDestinationInformation/FlightSegment/DestinationLocation` - пункты прилета (IATA-код города или аэропорта)

Далее необходимо проверить, все ли эти пункты находятся на территории РФ (требуется сделать отдельный справочник для этого или признак принадлежности к РФ в общем справочнике). Если хотя бы один из пунктов не относится к РФ, то считается, что это маршрут МВЛ и токенизация не требуется.

Определение гражданства пассажиров

Пассажиры добавляются в бронирование сервисом `TravelItineraryAddInfoLLSRQ/RS`. Однако в самом запросе не предусмотрено полей для указания гражданства пассажиров. Гражданство указывается при последующем вызове `SpecialServiceLLSRQ/RS`, но токенизация ФИО должна быть выполнена именно при вызове `TravelItineraryAddInfoLLSRQ/RS`. Поэтому решено, что вместе с запросом `TravelItineraryAddInfoLLSRQ/RS` будет передан HTTP-заголовок `PAX-NATIONALITY`, в котором будет передаваться список гражданств для всех пассажиров в бронировании в формате: `<номер пассажира в брони>-<2-х буквенный ISO-код страны гражданства>`. Записи разделяются точкой с запятой. Например:

`PAX-NATIONALITY: 01.01-RU;02.01-CN;03.01-DE;04.01-US;05.01-RU`

Токенизировать необходимо ФИО только тех пассажиров, для которых указано гражданство RU.

Формирование и сохранение токенов

В процессе бронирования при соблюдении необходимых условий на маршрут и гражданство выполняется первичная токенизация ПД пассажиров. Т.е. должны быть сгенерированы новые токены в соответствии с алгоритмом токенизации, которые должны быть сохранены в БД ПД. Поскольку в процессе бронирования номер PNR еще не известен, сохранять токены в БД необходимо с временным идентификатором (`TemporaryId`), см. ниже. Структура хранения токенов в БД должна выглядеть примерно следующим образом:

Данные о бронировании

- Код авиакомпании
- Номер PNR
- Дата/время создания PNR
- Дата/время создания записи

- Временный идентификатор (TemporaryId, используется до присвоения номера PNR)

Полетные сегменты (для каждого бронирования)

- Ссылка на бронирование
- Пункт вылета - 3-х буквенный код IATA
(/Envelope/Body/OTA_AirBookRQ/OriginDestinationInformation/FlightSegment/OriginLocation@LocationCode)
- Пункт прилета - 3-х буквенный код IATA
(/Envelope/Body/OTA_AirBookRQ/OriginDestinationInformation/FlightSegment/DestinationLocation@LocationCode)
- Номер рейса
(/Envelope/Body/OTA_AirBookRQ/OriginDestinationInformation/FlightSegment@FlightNumber)
- Время вылета
(/Envelope/Body/OTA_AirBookRQ/OriginDestinationInformation/FlightSegment@DepartureDateTime)

Данные о пассажирах (для каждого бронирования)

- Ссылка на бронирование
- nameAssocId
- nameNumber
- Имя
- Имя - токен
- Фамилия
- Фамилия - токен
- Отчество
- Отчество - токен
- Дата рождения
- Номер документа
- Номер документа - токен
- Тип пассажира (ADT, CHD, INF)

Идентификаторы пассажира на сегменте (для каждого пассажира в бронировании)

- Ссылка на пассажира в бронировании
- Ссылка на полетный сегмент
- `рахId`

Контактные данные (для каждого бронирования)

- Ссылка на бронирование
- Тип контактных данных (Email/Phone)
- Открытое значение
- Токенизированное значение
- Ссылка на пассажира (в случае, если контактные данные относятся к конкретному пассажиру)

Ведение сессий

Взаимодействие вызывающей системы и инвенторной системы ведется с использованием сессий. Все сервисы создания бронирования выполняются в рамках одной сессии. Сессия идентифицируется токеном, который указывается в каждом запросе в поле */Envelope/Header/Security/BinarySecurityToken*. Все запросы, вызывающиеся в рамках одной сессии имеют одно и то же значение токена.

Поскольку номер PNR присваивается бронированию в самом конце, после вызова всех основных сервисов (добавление маршрута, добавление пассажиров, указание документов и т.д.), требуется сохранять в БД ПД соответствие сгенерированных токенов и открытых ПД в привязке к временному идентификатору (`TemporaryId`). В качестве этого временного идентификатора необходимо использовать токен сессии. После того, как бронирование создано, ему присваивается номер PNR (`PNR locator`). Это происходит при вызове сервиса `EndTransactionLLSRQ/RS`. Ответ этого сервиса содержит номер PNR в атрибуте */soap-env:Envelope/soap-env:Body/EndTransactionRS/ItineraryRef@ID*. При получении этого ответа необходимо выполнять замену временного идентификатора `TemporaryId` на постоянный номер PNR (`ReservationNumber`). Также необходимо сохранять дату/время создания бронирования из атрибута */soap-env:Envelope/soap-env:Body/EndTransactionRS/ItineraryRef/Source@CreateDateTime*.

3.6 Выпуск билетов

Последовательность вызова сервисов SWS при оплате и выпуске билетов на сайте авиакомпании:

№ п/п	Сервис	Описание	Наличие ПД в запросе	Наличие ПД в ответе
1	SessionCreateRQ/RS	Создание сессии	-	-
2	<u>GetReservationRQ/RS</u>	Получение данных о брони	-	ФИО Номер документа Email
3	DesignatePrinterLLSRQ/RS	Назначение принтера	-	-
4	AirTicketRQ	Выпуск билетов	-	-
5	EndTransactionLLSRQ/RS	Завершение транзакции	-	-
6	SessionCloseRQ/RS	Закрытие сессии	-	-

3.7 Просмотр брони

Последовательность вызова сервисов SWS при просмотре оплаченного бронирования на сайте авиакомпании:

№ п/п	Сервис	Описание	Наличие ПД в запросе	Наличие ПД в ответе
1	SessionCreateRQ/RS	Создание сессии	-	-
2	<u>GetReservationRQ/RS</u>	Получение данных о брони	-	ФИО Номер документа Email
3	<u>TicketingDocumentServicesRQ/RS</u>	Получение данных о билетах	-	ФИО
4	IgnoreTransactionLLSRQ/RS	Отмена транзакции	-	-
5	SessionCloseRQ/RS	Закрытие сессии	-	-

3.8 Заказ дополнительных услуг

Последовательность вызова сервисов SWS при добавлении дополнительных услуг в бронь на сайте авиакомпании:

№ п/п	Сервис	Описание	Наличие ПД в запросе	Наличие ПД в ответе
1	SessionCreateRQ/RS	Создание сессии	-	-
2	<u>GetReservationRQ/RS</u>	Получение данных о брони	-	ФИО Номер доку-мента
3	SabreCommandLLSRQ/RS	Выполнение команды T*DSU	-	-
4	GetAncillaryOffersRQ/RS	Получение списка доступных доп.услуг	-	-
5	<u>UpdateReservationRQ/RS</u>	Внесение изменений в бронирование	ФИО	-
6	EndTransactionLLSRQ/RS	Завершение транзакции	-	-
7	SessionCloseRQ/RS	Закрытие сессии	-	-

3.9 Изменение данных пассажира

Последовательность вызова сервисов SWS при изменении данных пассажира в оплаченном бронировании на сайте авиакомпании:

№ п/п	Сервис	Описание	Наличие ПД в запросе	Наличие ПД в ответе
1	SessionCreateRQ/RS	Создание сессии	-	-
2	<u>GetReservationRQ/RS</u>	Получение данных о брони	-	ФИО Номер документа Email
3	<u>TicketingDocumentServicesRQ/RS</u>	Получение данных о билетах	-	ФИО
4	<u>UpdateReservationRQ/RS</u>	Внесение изменений в бронирование	ФИО	-
5	<u>SpecialServiceLLSRQ/RS</u>	Добавление элементов SSR в бронь	ФИО Номер документа Телефон	ФИО Номер документа Телефон
6	EndTransactionLLSRQ/RS	Завершение создания брони	-	-
7	SessionCloseRQ/RS	Закрытие сессии	-	-

3.10 Изменение контактных данных

Последовательность вызова сервисов SWS при изменении контактных данных (email, телефон) оплаченном бронировании на сайте авиакомпании:

№ п/п	Сервис	Описание	Наличие ПД в за-просе	Наличие ПД в от-вете
1	SessionCreateRQ/RS	Создание сес-сии	-	-
2	<u>GetReservationRQ/RS</u>	Получение данных о брони	-	ФИО Номер до-кумента Email
3	<u>TravelItineraryModifyInfoLLSRQ/RS</u>	Изменение контактных данных	Email Телефон	Email Телефон
4	<u>SpecialServiceLLSRQ/RS</u>	Изменение элементов SSR в брони	Телефон	Телефон
5	EndTransactionLLSRQ/RS	Завершение создания брони	-	-
6	SessionCloseRQ/RS	Закрытие сес-сии	-	-

3.11 Регистрация на рейс

Последовательность вызова сервисов SWS при выполнении регистрации на рейс на сайте авиакомпании:

№ п/п	Сервис	Описание	Наличие ПД в запросе	Наличие ПД в ответе
1	SessionCreateRQ/RS	Создание сессии	-	-
2	<u>GetReservationRQ/RS</u>	Возвращает общую информацию по бронированию	-	ФИО Номер документа Email Телефон
3	<u>GetPassengerDataRQ/RS</u>	Возвращает данные пассажира из бронирования	ФИО	ФИО Номер документа
4	<u>TicketingDocumentServicesRQ/RS</u>	Возвращает данные о билетах	-	ФИО
5	<u>EnhancedSeatMapRQ/RS</u>	Возвращает карту мест в салоне воздушного судна	ФИО	ФИО
6	<u>SelectSeatRQ/RS</u>	Осуществляет выбор места в салоне для пассажира	ФИО	ФИО
7	<u>UpdatePassengerDetailsRQ/RS</u>	Обновляет данные о пассажире в бронировании	ФИО	ФИО
8	<u>ACS_CheckInPassengerRQ/RS</u>	Выполняет регистрацию пассажира	ФИО	ФИО
9	<u>ACS_ReprintBPRQ/RS</u>	Возвращает набор данных, необходимых для печати посадочного талона	ФИО	ФИО
10	<u>ACS_SeatChangeRQ/RS</u>	Изменяет ранее выбранное место пассажира в салоне	-	ФИО

№ п/п	Сервис	Описание	Наличие ПД в за- просе	Наличие ПД в от- вете
11	SessionCloseRQ/RS	Закрытие сессии	-	-

3.12 Возврат и обмен билетов

Последовательность вызова сервисов SWS при изменении маршрута в ранее созданной и оплаченной брони на сайте авиакомпании:

№ п/п	Сервис	Описание	Наличие ПД в запросе	Наличие ПД в ответе
1	SessionCreateRQ/RS	Создание сессии	-	-
2	<u>GetReservationRQ/RS</u>	Получение данных о брони	-	ФИО Номер документа Email
3	<u>TicketingDocumentServicesRQ/RS</u>	Получение данных о билетах	-	ФИО
4	<u>TKT_ExchangeShoppingRQ/RS</u>	Поиск альтернативных предложений	ФИО	ФИО
5	OTA_CancelLSRQ/RS	Отмена сегментов маршрута	-	-
6	OTA_AirBookLLSRQ/RS	Бронирование новых сегментов	-	-
7	<u>AERRQ/RS</u>	Расчет суммы за обмен	-	ФИО
8	OTA_AirPriceLLSRQ	Прайсинг измененного бронирования	-	-
9	EndTransactionLLSRQ/RS	Завершение транзакции	-	-
10	SessionCloseRQ/RS	Закрытие сессии	-	-

3.13 Разделение бронирования

Последовательность вызова сервисов SWS при разделении бронирования, например для обмена или возврата билетов по части пассажиров из брони:

№ п/п	Сервис	Описание	Наличие ПД в запросе	Наличие ПД в ответе
1	SessionCreateRQ/RS	Создание сессии	-	-
2	<u>GetReservationRQ/RS</u>	Получение данных о брони	-	ФИО Номер документа Email Телефон
3	TravelItineraryDivideLLSRQ/RS	Разделение брони	-	-
4	<u>GetReservationRQ/RS</u>	Получение данных о брони (вызывается без указания номера PNR, возвращает данные по новой брони, получившейся в результате разделения)	-	ФИО Номер документа Email Телефон
5	<u>SpecialServiceLLSRQ/RS</u>	Добавление элемента SSR в бронь (SU SMSINFO <номер телефона>)	Телефон	Телефон
6	<u>GetReservationRQ/RS</u>	Получение данных о брони (вызывается без указания номера PNR, возвращает данные по новой брони, получившейся в результате разделения)	-	ФИО Номер документа Email Телефон
7	<u>UpdateReservationRQ/RS</u>	Обновление бронирования	-	-

№ п/п	Сервис	Описание	Наличие ПД в запросе	Наличие ПД в ответе
8	<u>SpecialServiceLLSRQ/RS</u>	Добавление элемента SSR в бронь (SU SMSINFO <номер телефона>)	Телефон	Телефон
9	TravelItineraryDivideLLSRQ/RS	Еще один вызов сервиса разделения брони, но без указания пассажиров. Переключает контекст на исходную бронь		
10	<u>GetReservationRQ/RS</u>	Получение данных о брони (вызывается без указания номера PNR, возвращает данные по первоначальной брони)		ФИО Номер документа Email Телефон
11	EndTransactionLLSRQ/RS	Завершение транзакции	-	-
12	SessionCloseRQ/RS	Закрытие сессии	-	-

Действия при разделении бронирования

В ряде случаев (возврат, обмен билетов для части пассажиров) выполняется разделение бронирования на два. При этом часть пассажиров остается в старой брони, часть переносится в новую. Все остальные данные, не связанные с пассажирами (маршрут, контакты т.д.), копируются в новую бронь без изменения. Разделение брони осуществляется сервисом `TravelItineraryDivideLLSRQ`, который обрабатывает текущую бронь, ранее полученную при вызове сервиса `GetReservationRQ`. В запросе `TravelItineraryDivideLLSRQ` указывается список номеров пассажиров (`nameNumber`), которых необходимо перенести в новую бронь (в элементе `/Envelope/Body/TravelItineraryDivideRQ/Divide/CustomerInfo/PersonName@NameNumber`). При получении запроса `TravelItineraryDivideLLSRQ` шлюз обработки SWS должен выполнять следующие действия:

1. Проверить, является ли разделяемая бронь (текущая в рамках сессии) токенизируемой. Если нет, то не выполнять дополнительных действий.
2. Если бронь токенизируемая, то при получении ответа `TravelItineraryDivideLLSRQ` необходимо выполнить деление брони в базе токенизации. Нужно создать новую запись PNR путем копирования существующей со следующими изменениями:
 - Номер нового бронирования получить из `/Envelope/Body/TravelItineraryDivideRS/ItineraryRef@ID`
 - Дату/врем создания нового бронирования получить из `/Envelope/Body/TravelItineraryDivideRS/ItineraryRef/Source@CreateDateTime`
 - Перенести в новое бронирование пассажиров, номера которых указаны в `/Envelope/Body/TravelItineraryDivideRQ/Divide/CustomerInfo/PersonName@NameNumber`. Переносятся только токенизированные пассажиры
 - Остальные записи, относящиеся к разделяемому бронированию (данные о маршруте, контактные данные) должны быть скопированы в новую бронь без изменений
3. Из старого бронирования удалить записи перенесенных пассажиров.
4. Желательно дополнительно сохранять ссылку на исходное бронирование в новом. Это может помочь при разборе сложных ситуаций.

При разделении брони возможна ситуация, когда в новой (или старой) брони отсутствуют токенизируемые пассажиры. Однако такая бронь все равно считается токенизированной, т.к. содержит токенизированные контактные данные, поэтому запросы/ответы, относящиеся к такой брони также должны проходить общий процесс обработки шлюзом SWS.

Важно иметь в виду, что строковый идентификатор (`nameNumber`, он же `nameId`) пассажиров в старой и новой брони меняется. Т.е. происходит перенумерация пассажиров. Однако целочисленный идентификатор (`nameAssocId`) остается неизменным. Например, было бронирование АААААА с 4-мя пассажирами:

Пассажир	NameNumber (nameId)	NameAssocId
PAX1	01.01	1
PAX2	02.01	2
PAX3	03.01	3
PAX4	04.01	4

Было выполнено разделение бронирования, при этом в новый PNR ВВВВВВ были перенесены пассажиры PAX2 и PAX3. В этом случае идентификаторы станут следующими:

PNR	Пассажир	NameNumber (nameId)	NameAssocId
AAAAAA	PAX1	01.01	1
AAAAAA	PAX4	02.01	4
ВВВВВВ	PAX2	01.01	2
ВВВВВВ	PAX3	02.01	3

Поэтому необходимо при разделении бронирования соответствующим образом обновлять идентификаторы пассажиров в БД ПД.

3.14 Поиск бронирований

Последовательность вызова сервисов SWS при поиске бронирований:

№ п/п	Сервис	Описание	Наличие ПД в запросе	Наличие ПД в ответе
1	SessionCreateRQ/RS	Создание сессии	-	-
2	<u>Trip_SearchRQ/RS</u>	Поиск бронирований по различным критериям	ФИО Номер документа Email Телефон	ФИО Номер документа Email
3	SessionCloseRQ/RS	Закрытие сессии	-	-

Алгоритм обработки запроса Trip_SearchRQ

Поскольку поиск бронирований может осуществляться по Фамилии/Email/Телефону/Номеру документа, при этом одному из этих критериев может соответствовать несколько различных токенов, для корректной обработки недостаточно просто заменить в запросе ПД на токен. Необходимо выполнить следующий алгоритм:

1. Необходимо выбрать в БД токенизации все записи, соответствующие указанным в запросе Фамилии/Email/Телефону/Номеру документа, для которых дата последнего сегмента больше или равна текущей дате - 7 дней.
2. В случае наличия в запросе блока FlightCriteria (/Envelope/Body/Trip_SearchRQ/ReadRequests/ReservationReadRequest/FlightCriteria) необходимо выбирать из БД только те PNR, полетные сегменты которых соответствуют критериям поиска по:
 - Номеру рейса (/Envelope/Body/Trip_SearchRQ/ReadRequests/ReservationReadRequest/FlightCriteria/Flight/FlightNumber)
 - Аэропорту вылета (/Envelope/Body/Trip_SearchRQ/ReadRequests/ReservationReadRequest/FlightCriteria/Flight/DepartureAirport)
 - Аэропорту прилета (/Envelope/Body/Trip_SearchRQ/ReadRequests/ReservationReadRequest/FlightCriteria/Flight/ArrivalAirport)
 - Диапазону времени вылета (/Envelope/Body/Trip_SearchRQ/ReadRequests/ReservationReadRequest/FlightCriteria/Flight/DepartureDateTime)

3. Для всех найденных записей требуется сформировать свой запрос Trip_SearchRQ, подставляя соответствующие токены.
4. Каждый запрос отправить в SWS (желательно распараллелить их, чтобы минимизировать общее время обработки).
5. Полученные ответы анализируются на наличие бронирований (/soap-env:Envelope/soap-env:Body/Trip_SearchRS/ReservationsList/Reservations).
6. Каждый из полученных ответов проходит процедуру детокенизации.
7. Из всех ответов, содержащих бронирования, должны извлекаться сами бронирования и "склеиваться" в один общий ответ. При этом необходимо соблюдать ограничение на количество результатов, которое указано в запросе в параметре /Envelope/Body/Trip_SearchRQ/ReadRequests/ReservationReadRequest/Return Options@MaxItemsReturned. Если количество полученных бронирований превышает значение этого параметра, то игнорировать все "лишние" бронирования.
8. В общем ответе фактическое количество результатов должно сохраняться в атрибуте /soap-env:Envelope/soap-env:Body/Trip_SearchRS/ReservationsList@TotalResults.
9. Итоговый сформированный ответ возвращается вызывающей системе.
10. Если исходный запрос не содержит критериев NameCriteria, EmailCriteria, TelephoneCriteria, DocumentCriteria, то он отправляется в SWS без изменений. В этом случае выполняется только детокенизация ответа.

При обработке запроса необходимо учитывать возможность нечеткого поиска, который определяется атрибутом MatchMode. Возможные варианты:

- EXACT, EXACT_NO_COMPRESS - точное совпадение
- START - значение поля начинается с указанного текста
- END - значение поля заканчивается на указанный текст
- ANYWHERE - значение поля содержит указанный текст
- SIMILAR - нечеткий поиск по "похожести". Точный алгоритм определения "похожести" неизвестен, поэтому считаем его равным ANYWHERE

3.15 Прочие процессы

Сервис	Описание	Наличие ПД в за-просе	Наличие ПД в ответе
<u>ACS PassengerListRQ/RS</u>	Возвращает список пассажиров на рейсе Сервис устарел и заменен на GetPassengerListRQ/RS, но часть систем продолжают его использование	-	ФИО
<u>ACS PassengerSecurityRQ/RS</u>	Возвращает статус прохождения пассажиром контроля безопасности	ФИО	-
<u>GetPassengerListRQ/RS</u>	Возвращает список пассажиров на рейсе	-	ФИО
<u>IssueBagTagRQ/RS</u>	Формирует данные для багажной бирки	ФИО	ФИО
<u>MISCServicesRQ/RS</u>	Выпускает EMD и MSR	ФИО	ФИО
<u>PriceQuoteServicesRQ/RS</u>	Возвращает данные прайс-квоты (price quote, PQ)	-	ФИО
<u>TravelItineraryModifyInfoLLSRQ/RS</u>	Изменяет бронь: номер телефона и/или email и/или ФИО пассажиров	ФИО Телефон Email	ФИО Телефон Email
<u>Trip SearchRQ/RS</u>	Осуществляет поиск бронирований по различным критериям	-	ФИО Телефон Email Документ
<u>UpdateSecurityInfoRQ/RS</u>	Обновляет данные документа	ФИО	ФИО

Сервис	Описание	Наличие ПД в запросе	Наличие ПД в ответе
		Документ	
<u>VCR_DisplayLLSRQ/RS</u>	Возвращает информацию о билете (VCR)	ФИО	ФИО

3.16 Обновление данных в базе данных ПД.

В процессе управления бронированием возможно внесение изменений в данные, которые хранятся в БД ПД. При этом могут меняться как персональные данные, так и справочные (данные о маршруте). Необходимо отслеживать такие изменения и соответствующим образом модифицировать данные в БД. Изменяться могут следующие данные:

- Номер документа
- Контактные данные (email, номер телефона)
- Сведения о маршруте (не токенизируются)

При изменении персональных данных необходимо сгенерировать новые токены для обновляемых полей и заменить ими открытые ПД в запросе.

При изменении всех данных желательно сохранять историю изменений (когда и что поменялось).

Изменение номера документа

Происходит в ходе выполнения процесса Изменение данных пассажира. Новый номер документа указывается в запросе SpecialServiceLLSRQ в поле /Envelope/Body/SpecialServiceRQ/SpecialServiceInfo/AdvancePassenger/Document@Number.

Изменение контактных данных

Происходит в ходе выполнения процесса Изменение контактных данных в брони. Новые email и телефон указываются в запросе TravelItineraryModifyInfoLLSRQ в полях:

- Email - /Envelope/Body/TravelItineraryModifyInfoRQ/CustomerInfo/Email@Address
- Телефон
- /Envelope/Body/TravelItineraryModifyInfoRQ/CustomerInfo/ContactNumbers/ContactNumber@Phone

Также новый номер телефона указывается в запросе SpecialServiceLLSRQ, однако, поскольку он вызывается после TravelItineraryModifyInfoLLSRQ, в БД будут уже измененные данные, поэтому нет необходимости в обновлении БД.

Важно иметь в виду, что изменяться могут или только email или только телефон или оба поля одновременно. При этом в TravelItineraryModifyInfoLLSRQ всегда передаются оба поля. Поэтому необходимо сравнивать email и телефон в запросе с теми, что хранятся в БД, и выполнять формирование нового токена только для тех полей, которые отличаются от сохраненных данных.

Изменение маршрута

Изменение маршрута может происходить не только по запросу пассажира (т.н. voluntary changes), так и по не зависящим от него причинам (involuntary changes), например, при изменении расписания или отмене рейса. Поэтому отследить такие изменения на уровне запросов не представляется возможным. Наиболее правильным будет получать актуальную информацию о маршруте из ответа GetReservationRQ/RS. Поскольку этот сервис вызывается очень часто, то желательно обработку маршрута, получаемого из его ответа, выполнять в асинхронном режиме (например, складывая ответы GetReservationRS в очередь, которая обрабатывается отдельным модулем). Маршрут необходимо получать из следующих полей:

- Пункт вылета - */soap-env:Envelope/soap-env:Body/stl19:GetReservationRS/stl19:Reservation/stl19:PassengerReservation/stl19:Segments/stl19:Segment/stl19:Air/stl19:DepartureAirport*
- Пункт прилета - */soap-env:Envelope/soap-env:Body/stl19:GetReservationRS/stl19:Reservation/stl19:PassengerReservation/stl19:Segments/stl19:Segment/stl19:Air/stl19:ArrivalAirport*
- Номер рейса - */soap-env:Envelope/soap-env:Body/stl19:GetReservationRS/stl19:Reservation/stl19:PassengerReservation/stl19:Segments/stl19:Segment/stl19:Air/stl19:OperatingFlightNumber*
- Время вылета - */Envelope/Body/GetReservationRS/Reservation/PassengerReservation/Segments/Segment/Air/DepartureDateTime*

Особенности работы с транзакциями

Все изменения, вносимые в бронирование, выполняются в рамках транзакций. Транзакция начинается в рамках сессии в момент вызова GetReservationRQ и завершается вызовом EndTransactionLLSRQ (подтверждение изменений) или IgnoreTransactionLLSRQ (отмена изменений). До момента подтверждения изменений все вызовы сервисов в рамках текущей сессии должны осуществляться с учетом тех изменений, которые были сделаны в рамках этой сессии. Но если при этом в рамках другой сессии одновременно осуществляется работа с этим же бронированием, то эта другая сессия должна работать со своей копией исходных данных.

Сохранять изменения, сделанные в рамках сессии, в БД ПД, необходимо в случае успешного вызова EndTransactionLLSRQ. Успешным вызов считается в том случае, если в ответе EndTransactionLLSRQ присутствует элемент */soap-env:Envelope/soap-*

env:Body/EndTransactionRS/stl:ApplicationResults/stl:Success. Если данный элемент отсутствует, то при завершении транзакции произошла ошибка на стороне инвенторной системы, и все изменения, сделанные в рамках данной сессии не должны сохраниться в БД ПД.

В случае, если после сделанных изменений в той же сессии осуществляется вызов *IgnoreTransactionLLSRQ*, то все изменения отменяются и в рамках текущей сессии работа выполняется с исходными данными.

3.15 Обработка трафика бэк-офисных систем

Система производственной аналитической платформы используется в авиакомпании в качестве системы пост-букинга. Она получает информацию обо всех бронированиях, билетах, EMD, а также обо всех изменениях в них, непосредственно из Sabre PSS в виде так называемых фидов (feeds). Фиды содержат сообщения в формате XML. Фиды поступают от Sabre PSS в очередь сообщений авиакомпании (используется IBM Integration Bus). Далее система производственной аналитической платформы вычитывает фиды из очереди и сохраняет данные в своей БД (Oracle). С т.з. токенизации ПД в данном процессе необходимо выполнение детокенизации ПД.

Поскольку персональные данные содержатся только в 3-х типах фидов, детокенизация должна осуществляться только для них:

- PNR - данные о бронированиях
- VCR - данные о билетах
- DCS - данные о прохождении пассажиром процесса регистрации и посадки на рейс.

PNR

Детокенизируемые ПД

Путь	Описание	Пример
/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:EmailAddress/stl13:EmailAddress/stl13:Address	Email	<stl13:Address> <u>M190469@RAMBLER.RU</u> </stl13:Address>
/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Passengers/stl13:Passenger/stl13:FirstName	Имя Отчество	<stl13:FirstName> IVAN </stl13:FirstName>
/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Passengers/stl13:Passenger/stl13:LastName	Фамилия	<stl13:LastName> IVANOV </stl13:LastName>
/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Passengers/stl13:Passenger/stl13:SpecialRequests/stl13:APISRequest/stl13:DOCSEntry/stl13:DocumentNumber	Номер документа	<stl13:DocumentNumber> 258741369 </stl13:DocumentNumber>
/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Passengers/stl13:Passenger/stl13:SpecialRequests/stl13:APISRequest/stl13:DOCSEntry/stl13:Forename	Имя Отчество	<stl13:Forename> IVAN </stl13:Forename>

Путь	Описание	Пример
/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Passengers/stl13:Passenger/stl13:SpecialRequests/stl13:APISRequest/stl13:DOCSEntry/stl13:Surname	Фамилия	<stl13:Surname> IVANOV </stl13:Surname>
/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Passengers/stl13:Passenger/stl13:SpecialRequests/stl13:GenericSpecialRequest/stl13:FreeText	ФИО	<stl13:FreeText> IVANOV/ANDREY /01JAN19</stl13:FreeText>
/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Passengers/stl13:Passenger/stl13:SpecialRequests/stl13:GenericSpecialRequest/stl13:FullText	ФИО	<stl13:FullText>INFT SU 1763Z10JUN/ IVANOV/ANDREY /01JAN19 NN1</stl13:FullText>
/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Segments/stl13:Air/stl13:SegmentSpecialRequests/stl13:GenericSpecialRequest/stl13:FreeText	ФИО	<stl13:FreeText> IVANOV/ANDREY /01JAN19</stl13:FreeText>
/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Segments/stl13:Air/stl13:SegmentSpecialRequests/stl13:GenericSpecialRequest/stl13:FullText	ФИО	<stl13:FullText>INFT SU 1763Z10JUN/ IVANOV/ANDREY /01JAN19 NN1</stl13:FullText>

Идентификация пассажиров

Тип идентификации	Путь	Пример
nameAssocId	/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Passengers/stl13:Passenger/@nameAssocId	<stl13:Passenger id="3" nameType="S" referenceNumber="ADT" nameId="01.01" nameAssocId="1" elementId="pnr-3.1">
nameId	/asds3_6:SabreASDS/stl13:GetReservationRS/stl13:Reservation/stl13:PassengerReservation/stl13:Passengers/stl13:Passenger/@nameId	<stl13:Passenger id="3" nameType="S" referenceNumber="ADT" nameId="01.01" nameAssocId="1" elementId="pnr-3.1">

VCR

Детокенизируемые ПД

Путь	Описание	Пример
/ns2:TicketingDocument.Notification/ns2:Ticketing/ns2:Ticket/ns2:Customer/ns2:Traveler/ns2:Name	ФИО	<ns2:Name> TEST/ADAM </ns2:Name>
/ns2:TicketingDocument.Notification/ns2:Ticketing/ns2:Ticket/ns2:Customer/ns2:Traveler/ns2:FirstName	Имя Отчество	<ns2:FirstName> ADAM </ns2:FirstName>
/ns2:TicketingDocument.Notification/ns2:Ticketing/ns2:Ticket/ns2:Customer/ns2:Traveler/ns2:LastName	Фамилия	<ns2:LastName> TEST </ns2:LastName>

Идентификация пассажиров

Тип идентификации	Путь	Пример
nameAssocId	/ns2:TicketingDocument.Notification/ns2:Ticketing/ns2:Ticket/ns2:Customer/ns2:Traveler/@nameId	<ns2:Traveler nameId="1" nameNumber="1.1">

DCS

Детокенизируемые ПД

Путь	Описание	Пример
/ACS:ACS_PDC_Flt_Notif/PAX_Record/Flight/Identification/Name_Last	Фамилия	<Name_Last> QRDNIJ </Name_Last>
/ACS:ACS_PDC_Flt_Notif/PAX_Record/Flight/Identification/Name_First	Имя Отчество	<Name_First> GIEL </Name_First>
/ACS:ACS_PDC_Flt_Notif/PAX_Record/DocX/TravelDoc/Number	Номер документа	<Number> 123456789 </Number>
/ACS:ACS_PDC_Flt_Notif/PAX_Record/DocX/TravelDoc/Pax/@first	Имя	<Pax first="GIEL" dob="01JAN2011" countryA="USA" gender="F"> QRDNIJ </Pax>
/ACS:ACS_PDC_Flt_Notif/PAX_Record/DocX/TravelDoc/Pax	Фамилия	<Pax first="GIEL" dob="01JAN2011" countryA="USA" gender="F"> QRDNIJ </Pax>

Идентификация пассажиров

Тип идентификации	Путь	Пример
paxId	/ACS:ACS_PDC_Flt_Notif/PAX_Record/PAX_ID/@uniqueID	<PAX_ID pnrCreate="2020-06-02T02:00:00-05:00" resID="1" itinID="0" uniqueID="F512E5DA0001" >UKEQRF</PAX_ID>