Бастион-3 – Face. Руководство программиста

Версия 2024.3

(25.10.2024)

Самара, 2024

Оглавление

1 Общие сведения

API модуля «Бастион-3 – Face» (далее – Face API) предназначен для интеграции ПК «Бастион-3» с системами биометрической идентификации (далее – внешняя система) с целью реализации в ПК «Бастион-3» различных режимов доступа, включающих распознавание лиц.

Взаимодействие с внешними системами производится с использованием протокола на основе стандарта ONVIF Profile A, C.

Face API доступен при установке модуля **«Бастион-3 – Face»**, который описан в документе «Бастион-3 – Face. Руководство администратора». Там приведена вся необходимая информация об условиях применения, установке и настройке модуля.

Face API позволяет обеспечивать доступ в трёх режимах, это:

- Режим идентификации;
- Режим двухфакторной аутентификации;
- Режим отслеживания прохода на виртуальных точках прохода.

Подробнее о режимах прохода см. п. 2.

Face API позволяет решить следующие задачи:

1. Синхронизация информации о пропусках ПК «Бастион-3» с внешней системой;
2. Синхронизация информации о точках прохода ПК «Бастион-3» с внешней системой;
3. Передача в ПК «Бастион-3» событий идентификации внешней системой;
4. Передача в ПК «Бастион-3» команд подтверждения доступа на точках прохода, работающих в режиме двухфакторной аутентификации по запросу;
5. Передача в ПК «Бастион-3» событий нарушения режима доступа;

Подробнее о реализации взаимодействий см. п. 3.

Задачи по управлению камерами видеонаблюдения и связи камер с точками прохода СКУД полностью лежат на стороне внешней системы.

Для производителей внешних систем, выполняющих интеграцию с ПК «Бастион-3» своими силами через Face API, доступно приложение-эмулятор внешней системы, которое можно получить вместе с исходным кодом, обратившись в отдел технической поддержки ГК «ТвинПро».

2 Сценарии использования

2.1 Режим идентификации

В этом режиме решение об идентификации персоны принимается внешней системой исключительно на основе распознавания лица (Рис. 1). То есть, внешняя система будет работать в режиме поиска лиц 1: N, где N – общее число лиц в системе.

Для получения доступа посетителю достаточно появиться в поле зрения камеры видеонаблюдения, связанной с определенным направлением прохода (вход или выход) заданной точки доступа.

Для предоставления доступа внешняя система должна сгенерировать событие типа **AccessGranted/Credential** (подробнее см. в п. 3.5). ПК «Бастион-3» получит событие и примет окончательное решение о предоставлении доступа на основе уровня доступа пропуска.

Внешняя система может также может передавать события отказа в доступе **Denied/Credential** в случае, если обнаружено неизвестное лицо, либо обнаруженная персона нарушает какие-либо дополнительные правила, проверяемые внешней системой (например, превышена температура, нет маски и т. п.).

Во всех случаях к генерируемому внешней системой событию может быть привязана фотография события.

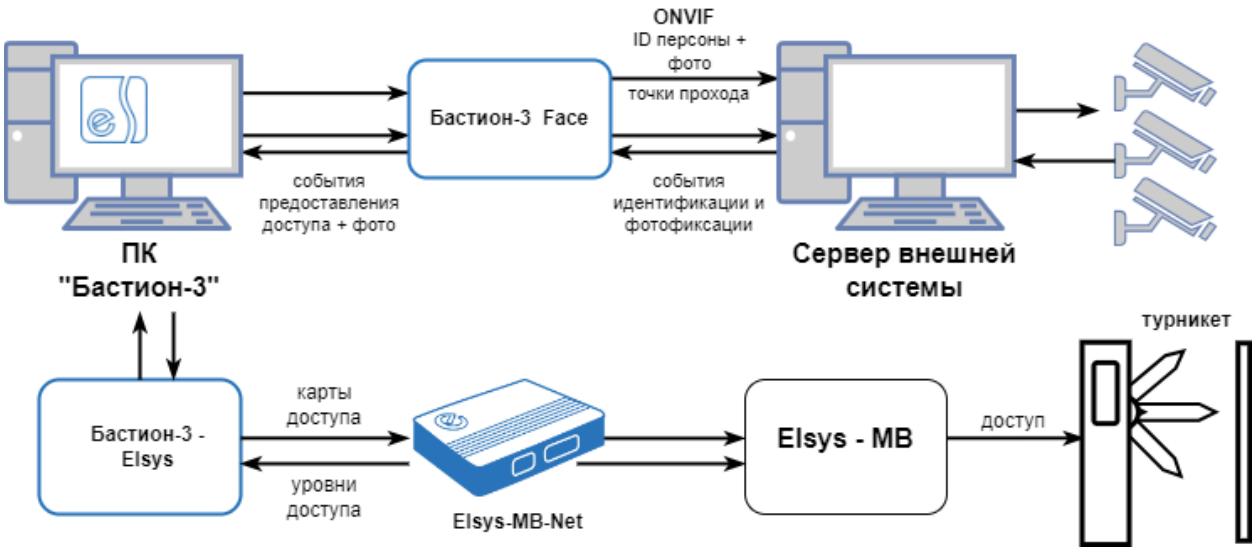

Рис. 1. Режим идентификации

2.2 Режим двухфакторной аутентификации

В режиме двухфакторной аутентификации идентификация персоны выполняется в ПК «Бастион-3» по карте доступа или какому-либо другому идентификатору, а внешняя система должна анализировать события **Request/Credential** для подтверждения или отказа в доступе. То есть, в этом режиме внешняя система будет работать в режиме сравнения лиц 1: 1.

Обычно в этом режиме для получения доступа посетитель сначала должен приложить пропуск к считывателю на точке прохода. При этом его лицо должно быть в зоне обзора камеры видеонаблюдения, которая ассоциирована с соответствующим направлением (вход или выход) в точке прохода. Если пропуск посетителя разрешает проход на этой точке прохода, то ПК «Бастион-3» сгенерирует событие типа **Request/Credential** (Рис. 2).

Внешняя система, получив событие от ПК «Бастион-3», принимает решение о соответствии лица на фотографии, полученной ранее от ПК «Бастион-3», биометрическим данным, получаемым с

камеры видеонаблюдения. Результат аутентификации передается от внешней системы в ПК «Бастион-3» путём вызова метода **ExternalAuthorization** (подробнее в п. 3.6).

В метод **ExternalAuthorization** должно также быть передано событие типа **AccessGranted/Credential** в случае положительного решения, или событие типа **Denied/Credential** в случае принятия внешней системой отрицательного решения.

Во всех случаях к передаваемому событию может быть прикреплена фотография, полученная с камеры видеонаблюдения.

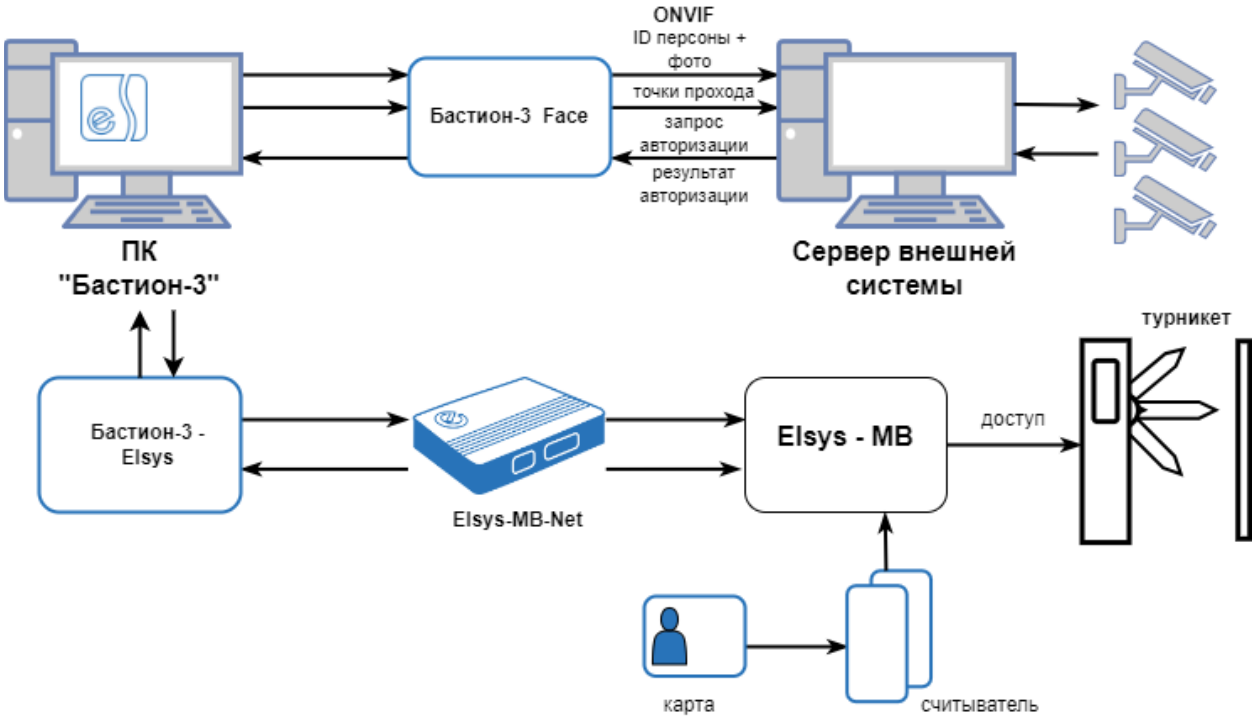

Рис. 2. Режим двухфакторной аутентификации

2.3 Виртуальные точки прохода

Виртуальная точка прохода в ПК «Бастион-3» не связана с каким-либо преграждающим устройством и предназначена только для контроля полномочий нахождения пропусков в заданной области.

С точки зрения интеграции этот режим ничем не отличается от режима идентификации. Виртуальные точки прохода при синхронизации ничем не отличаются от реальных точек.

В этом режиме решение об идентификации персоны принимается внешней системой исключительно на основе распознавания лица (Рис. 1). То есть, внешняя система будет работать в режиме поиска лиц 1: N, где N – общее число лиц в системе.

Для подтверждения своих полномочий посетителю достаточно появиться в поле зрения камеры видеонаблюдения, связанной с определенной виртуальной точкой прохода.

Для подтверждения полномочий внешняя система должна сгенерировать событие типа **AccessGranted/Credential** (подробнее см. в п. 3.5). ПК «Бастион-3» получит событие и примет окончательное решение о правомерности доступа на основе уровня доступа пропуска.

Внешняя система может также может передавать события отказа в доступе **Denied/Credential** в случае, если обнаружено неизвестное лицо, либо обнаруженная персона нарушает какие-либо дополнительные правила, проверяемые внешней системой (например, превышена температура, нет маски и т. п.).

Во всех случаях к генерируемому внешней системой событию может быть привязана фотография события.

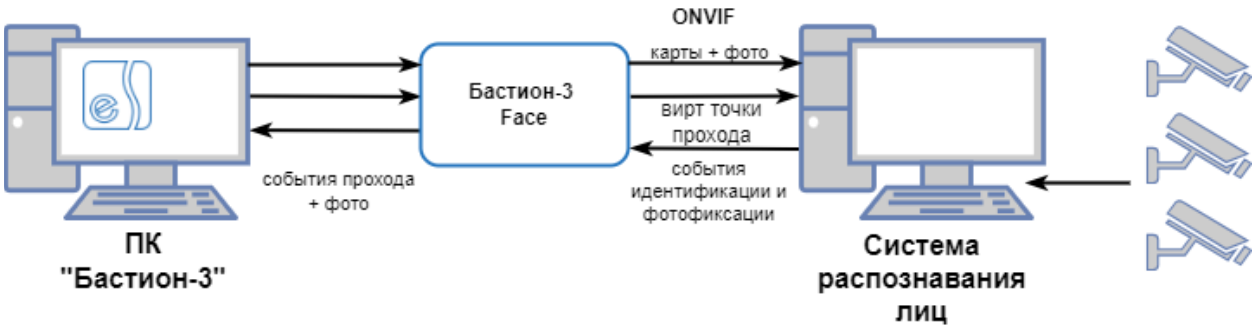

Рис. 3. Работа системы с виртуальными точками прохода

3 Протокол интеграции

3.1 Общие положения

В основе протокола интеграции лежат профили ONVIF A и C. Профиль A используется для синхронизации информации о пропусках (Credential Service). Профиль C используется для обмена событиями (Event Service) и для синхронизации информации о точках прохода (Access Control Service).

Подробная информация о профилях и службах, включая спецификации сетевых интерфейсов, приведена на сайте ONVIF по адресу: https://www.onvif.org/profiles/specifications/

Сетевой интерфейс **Device** реализуется на обеих сторонах взаимодействия в качестве точки входа, как этого требуют спецификации ONVIF в документе ONVIF Core Specification (https://www.onvif.org/specs/core/ONVIF-Core-Specification.pdf).

Механизм обмена событиями с использованием модели Real-time Pull Point должен быть также реализован на обеих сторонах взаимодействия, в том числе и на стороне внешней системы.

3.2 Механизм обмена событиями

Для обмена событиями на обеих сторонах взаимодействия необходима реализация сетевого интерфейса **EventPortType** (https://www.onvif.org/ver10/events/wsdl/event.wsdl).

Список методов, которые должны поддерживаться реализацией **EventPortType:**

Метод	Информация
CreatePullPointSubscription	Создание новой подписки на события
GetEventProperties	Получение набора поддерживаемых параметров для фильтрации событий

В результате вызова метода CreatePullPointSubscription должна возвращаться информация о новом экземпляре сетевой службы, реализующей интерфейс **PullPointSubscription** (https://www.onvif.org/ver10/events/wsdl/event.wsdl), в котором должны быть реализованы следующие методы:

Метод	Информация
PullMessage	Получить события
Unsubscribe	Отписаться от получения событий

Подробнее модель механизма обмена событиями описана в документе ONVIF Core Specification (https://www.onvif.org/specs/core/ONVIF-Core-Specification.pdf) в разделе 9.1. Диаграмма последовательности при работе с сетевыми интерфейсами механизма обмена событиями изображена на Рис. 4.

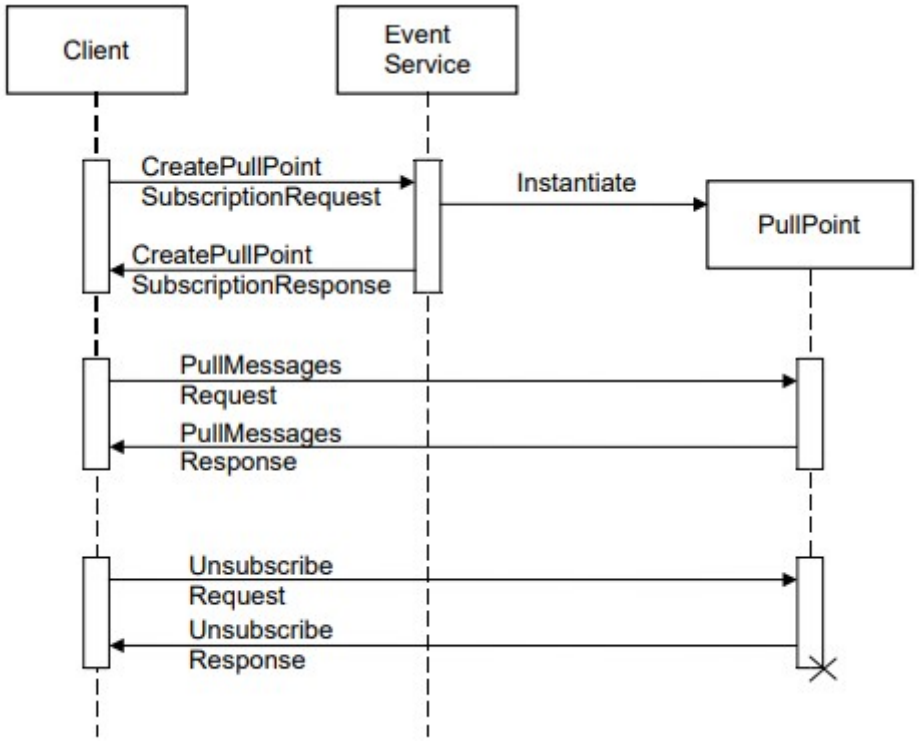

Рис. 4. Диаграмма последовательности при работе с событиями

3.3 Синхронизация пропусков

3.3.1 Общие сведения

Для синхронизации информации о персонах Face API предлагает два варианта, которые отличаются тем, на чьей стороне (на стороне Face API или на стороне внешней системы) реализуется сетевой интерфейс **CredentialPort** (https://www.onvif.org/ver10/credential/wsdl/credential.wsdl), который подробнее описан в документе ONVIF Credential Service Specification (https://www.onvif.org/specs/srv/access/ONVIF-Credential-Service-Spec.pdf).

В качестве идентификаторов пропусков используются полные номера карт доступа из ПК «Бастион-3».

> ***Внимание!*** *При реализации интеграции на стороне внешней системы следует учитывать, что одной персоне в ПК «Бастион-3» может соответствовать несколько пропусков. Поэтому при синхронизации данных о пропусках одна и та же персона, с одной и той же фотографией, может встречаться несколько раз в списке пропусков.*

3.3.2 Реализация службы пропусков на стороне Face API

Этот вариант использует реализацию интерфейса **CredentialPort** на стороне Face API, которая включает следующие методы:

Метод	Описание
GetCredentialList	Получить список всех персон с возможностью указать лимит и стартовый идентификатор
GetCredentials	Получить список персон, соответствующий переданному набору идентификаторов

Остальные методы в реализации **CredentialPort** на стороне Face API не реализованы, поскольку служба предназначена только для получения информации о пропусках.

Оба метода в ответе содержат объекты типа **Credential** (https://www.onvif.org/ver10/credential/wsdl/credential.wsdl), который содержит данные пропуска в следующем виде:

- Credential – пропуск
  - token – строка с идентификатором пропуска (номером карты доступа)
  - CredentialHolderReference – строка с ФИО обладателя пропуска
  - CredentialIdentifier
    - Type
      - Name - строка со значением «pt:Face»
      - FormatType – строка со значением «JPEG File»
    - Value – Фотография обладателя карты доступа в формате hexBinary
  - Attributes[] - массив атрибутов, содержащих дополнительную информацию о персоне (телефон, доп. Телефон, e-mail, а также значения дополнительных полей)

- Name – название поля
- Value – значение поля

Остальные поля пропуска в Face API не задействованы.

Кроме того, Face API генерирует события, возникающие при изменении, добавлении или удалении пропуска. Получение событий выполняется при помощи реализованного на стороне Face API механизма обмена событиями.

Формат события типа **Credential/Changed**, которое генерируется при добавлении или изменении пропуска:

```
<tns1: Configuration
  xmlns:wstop="http://docs.oasisopen.org/wsn/t-1"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">
  <Credential wstop:topic="true">
    <Changed>
      <tt:MessageDescription IsProperty="false">
        <tt:Source>
          <tt:SimpleItemDescription Name="CredentialToken" Type="pt:ReferenceToken"/>
        </tt:Source>
      </tt:MessageDescription>
    </Changed>
  </Credential>
</tns1:Configuration>
```

В событии используются следующие атрибуты:

Атрибут	Описание
CredentialToken	Идентификатор пропуска (номер карты доступа)

При получении этого события внешней системе следует вызвать метод **GetCredentials** для получения актуальной информации о пропуске, CredentialToken которого передан в событии.

При удалении пропуска Face API генерирует событие типа **Credential/Removed** в следующем формате:

```
<tns1:Configuration
  xmlns:wstop="http://docs.oasisopen.org/wsn/t-1"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">
  <Credential wstop:topic="true">
    <Removed>
      <tt:MessageDescription IsProperty="false">
        <tt:Source>
          <tt:SimpleItemDescription Name="CredentialToken" Type="pt:ReferenceToken"/>
        </tt:Source>
      </tt:MessageDescription>
    </Removed>
  </Credential>
</tns1:Configuration>
```

В событии представлены используются атрибуты:

Атрибут	Описание
CredentialToken	Идентификатор пропуска (номер карты доступа)

При получении этого события внешней системе следует удалить из набора пропуск, CredentialToken которой передан в событии.

Таким образом задача по поддержанию набора пропусков в актуальном состоянии частично лежит на стороне внешней системы. ПК «Бастион-3», в свою очередь, информирует о необходимости актуализации набора пропусков.

3.3.3 Реализация службы персон на стороне внешней системы

Этот вариант предполагает реализацию сетевого интерфейса **CredentialPort** (https://www.onvif.org/ver10/credential/wsdl/credential.wsdl) на стороне внешней системы и позволяет обойтись без использования событий о добавлении/изменении/удалении пропусков. В этом случае ответственность за поддержание актуальности набора пропусков во внешней системе полностью лежит на стороне ПК «Бастион-3».

Этот вариант поддерживается Face API автоматически в том случае, когда ПК «Бастион-3» обнаруживает, что на стороне внешней системы есть реализация интерфейса **CredentialPort.** Для этого необходимо, чтобы метод **GetServices** в реализации интерфейса **Device** на стороне внешней системы возвращал информацию о реализации интерфейса **CredentialPort.**

В реализации сетевого интерфейса **CredentialPort** должен поддерживаться следующий набор методов:

Метод	Описание
GetCredentialInfoList	Получить список всех пропусков с возможностью указать лимит и стартовый идентификатор; этот метод отличается от GetCredentialList форматом вызова и способом возврата данных в ответе
GetCredentialInfo	Получить список пропусков, соответствующий переданному набору уникальных идентификаторов; этот метод отличается от GetCredentials форматом вызова и способом возврата данных в ответе
DeleteCredential	Удалить пропуск
CreateCredential	Создать пропуск
ModifyCredential	Изменить данные пропуска

3.4 Синхронизация точек прохода

Для получении внешней системой информации о точках прохода из ПК «Бастион-3» в Face API предусмотрена реализация сетевого интерфейса **PACSPort** (https://www.onvif.org/ver10/pacs/accesscontrol.wsdl), включая следующие методы:

Метод	Описание
GetAccessPointInfo	Получить список всех точек прохода с возможностью указать лимит и стартовый идентификатор

GetAccessPointInfoList	Получить список точек прохода, соответствующий переданному набору идентификаторов
ExternalAuthorization	*Передать подтверждение двухфакторной аутентификации; этот метод используется в режиме двухфакторной аутентификации, подробнее в п. 3.6*

Остальные методы в Face API не задействованы.

«Точка прохода» или AccessPoint в ONVIF соответствует направлению прохода или виртуальной точке доступа в ПК «Бастион-3». В качестве идентификаторов точек прохода используются внутренние идентификаторы устройств ПК «Бастион-3».

Методы получения информации о точках прохода возвращают в ответе объекты типа **AccessPointInfo** (https://www.onvif.org/ver10/pacs/accesscontrol.wsdl) в следующем виде:

- AccessPointInfo – точка прохода
  - token – строка с идентификатором точки прохода
  - Name – имя точки прохода
  - Description – описание точки прохода
  - Entity – строка со значением «AccessPoint»
  - Capabilities
    - ExternalAuthorization – true при работе в режиме двухфакторной аутентификации, false – в остальных случаях

Другие поля в Face API не задействованы.

Кроме того, Face API генерирует события, возникающие при изменении, добавлении и удалении точки прохода. Получение событий выполняется при помощи реализованного на стороне Face API механизма обмена событиями.

Формат события типа **AccessPoint/Changed**, которое генерируется при добавлении или изменении точки прохода:

```
<tns1:Configuration
  xmlns:wstop="http://docs.oasisopen.org/wsn/t-1"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">
  <AccessPoint>
    <Changed>
      <tt:MessageDescription IsProperty="false">
        <tt:Source>
          <tt:SimpleItemDescription Name="AccessPointToken" Type="pt:ReferenceToken"/>
        </tt:Source>
      </tt:MessageDescription>
    </Changed>
  </AccessPoint>
</tns1:Configuration>
```

В событии представлены следующие атрибуты:

Атрибут	Описание
AccessPointToken	Идентификатор точки прохода

При получении этого события внешней системе следует вызвать метод **GetAccessPointInfoList** для получения актуальной информации о точке прохода, AccessPointToken которой передан в событии.

При удалении точки прохода Face API генерирует событие типа **AccessPoint/Removed** в следующем формате:

```
<tns1:Configuration
  xmlns:wstop="http://docs.oasisopen.org/wsn/t-1"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">
  <AccessPoint>
    <Removed>
      <tt:MessageDescription IsProperty="false">
        <tt:Source>
          <tt:SimpleItemDescription Name="AccessPointToken" Type="pt:ReferenceToken"/>
        </tt:Source>
      </tt:MessageDescription>
    </Removed>
  </AccessPoint>
</tns1:Configuration>
```

В событии используются атрибуты:

Атрибут	Описание
AccessPointToken	Идентификатор точки прохода

При получении события этого типа внешней системе следует удалить из набора точку прохода, AccessPointToken которой передан в событии.

Таким образом задача по поддержанию набора точек прохода в актуальном состоянии частично лежит на стороне внешней системы. ПК «Бастион-3», в свою очередь, информирует о необходимости актуализации набора точек прохода.

3.5 Передача событий в режиме идентификации

Для точек прохода, которые работают в режиме идентификации, а также для виртуальных точек прохода, ПК «Бастион-3» ожидает событий идентификации от внешней системы. Режим идентификации считается включенным если свойство **ExternalAuthorization** в **Capabilities** соответствующего экземпляра типа **AccessPointInfo** установлено в **false**. Настройка режимов работы точек прохода производится в ПК «Бастион-3».

Событие типа **AccessGranted/Credential** передается при успешной идентификации пропуска из внешней системы в ПК «Бастион-3» в следующем формате:

```
<tns1:Configuration
  xmlns:wstop="http://docs.oasisopen.org/wsn/t-1"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">
  <tns1:AccessControl wstop:topic="true">
    <AccessGranted wstop:topic="true">
      <Credential wstop:topic="true">
        <tt:MessageDescription IsProperty="false">
          <tt:Source>
            <tt:SimpleItemDescription Name="AccessPointToken" Type="pt:ReferenceToken"/>
          </tt:Source>
          <tt:Data>
```

```
            <tt:SimpleItemDescription Name="CredentialToken" Type="pt:ReferenceToken"/>
            <tt:SimpleItemDescription Name="Reason" Type="xs:string"/>
          </tt:Data>
        </tt:MessageDescription>
      </Credential>
    </AccessGranted>
  </tns1:AccessControl>
</tns1:Configuration>
```

В событии указаны атрибуты:

Атрибут	Описание
AccessPointToken	Идентификатор точки прохода, где возникло событие
CredentialToken	Идентификатор пропуска
Reason	Фотография, полученная с камер видеонаблюдения, в формате JPEG, преобразованном в base64

Событие типа **Denied/Credential** передается если обнаружено неизвестное лицо, либо обнаруженная персона нарушает какие-либо дополнительные правила, проверяемые внешней системой (например, превышена температура, нет маски и т. п.). Событие передаётся из внешней системы в ПК «Бастион-3» в следующем формате:

```
<tns1:Configuration
  xmlns:wstop="http://docs.oasisopen.org/wsn/t-1"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">
  <tns1:AccessControl wstop:topic="true">
    <Denied wstop:topic="true">
      <Credential wstop:topic="true">
        <tt:MessageDescription IsProperty="false">
          <tt:Source>
            <tt:SimpleItemDescription Name="AccessPointToken" Type="pt:ReferenceToken"/>
          </tt:Source>
          <tt:Data>
            <tt:SimpleItemDescription Name="CredentialToken" Type="pt:ReferenceToken"/>
            <tt:SimpleItemDescription Name="Reason" Type="xs:string"/>
            <tt:SimpleItemDescription Name="Temperature" Type="xs:decimal"/>
            <tt:SimpleItemDescription Name="TemperatureViolation" Type="xs:boolean"/>
            <tt:SimpleItemDescription Name="HasMask" Type="xs:boolean"/>
            <tt:SimpleItemDescription Name="MaskViolation" Type="xs:boolean"/>
          </tt:Data>
        </tt:MessageDescription>
      </Credential>
    </Denied>
  </tns1:AccessControl>
</tns1:Configuration>
```

В событии используются атрибуты:

Атрибут	Описание
AccessPointToken	Идентификатор точки прохода, где возникло событие
CredentialToken	Идентификатор пропуска (если персона идентифицирована)

Reason	Фотография, полученная с камер видеонаблюдения, в формате JPEG, преобразованном в base64
Temperature	Температура тела человека в градусах Цельсия (при наличии соответствующей функции внешней системы)
TemperatureViolation	Признак превышения максимально допустимой температуры тела (при наличии функции и при настроенном во внешней системе запрета на доступ при превышении определённой температуры)
HasMask	Признак наличия лицевой маски
MaskViolation	Признак нарушения доступа по признаку наличия лицевой маски (в зависимости от того, настроен ли во внешней системе запрет на доступ по признаку отсутствия или наличия маски)

Значение атрибута **Temperature** игнорируется ПК «Бастион-3» в случае, если атрибут **TemperatureViolation** отсутствует или его значение – false.

Значение атрибута **HasMask** игнорируется ПК «Бастион-3» в случае, если атрибут **MaskViolation** отсутствует или его значение – false.

Порог максимально допустимой температуры, наличие средств измерения температуры и необходимость использования маски должны настраиваться во внешней системе, если требуется их использовать.

3.6 Передача подтверждений двухфакторной аутентификации

На точках прохода, которые работают в режиме двухфакторной аутентификации, ПК «Бастион-3», после идентификации пропуска ожидает подтверждения доступа от внешней системы. Режим двухфакторной аутентификации считается включенным если свойство **ExternalAuthorization** в **Capabilities** соответствующего экземпляра типа **AccessPointInfo** установлено в **true**. Настройка режимов работы точек прохода производится в ПК «Бастион-3».

Сигналом о начале ожидания подтверждения на определённой точке прохода является генерация ПК «Бастион-3» события типа **Request/Credential** в следующем формате:

```
<tns1:Configuration
  xmlns:wstop="http://docs.oasisopen.org/wsn/t-1"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">
  <tns1:AccessControl wstop:topic="true">
    <Request wstop:topic="true">
      <Credential wstop:topic="true">
        <tt:MessageDescription IsProperty="false">
          <tt:Source>
            <tt:SimpleItemDescription Name="AccessPointToken" Type="pt:ReferenceToken"/>
          </tt:Source>
          <tt:Data>
            <tt:SimpleItemDescription Name="CredentialToken" Type="pt:ReferenceToken"/>
            <tt:SimpleItemDescription Name="CredentialHolderName" Type="xs:string"/>
          </tt:Data>
        </tt:MessageDescription>
```

```
      </Credential>
    </Request>
  </tns1:AccessControl>
</tns1:Configuration>
```

Событие содержит следующие атрибуты:

Атрибут	Описание
AccessPointToken	Идентификатор точки прохода, которая ожидает подтверждения
CredentialToken	Идентификатор пропуска, предположительный владелец которого ожидает подтверждения доступа на соответствующей точке прохода
CredentialHolderName	ФИО владельца пропуска, который ожидает подтверждения доступа на точке прохода

Получив такое событие, внешняя система должна выполнить проверку по собственным алгоритмам и оповестить ПК «Бастион-3» о своём решении путём вызова метода **ExternalAuthorization**. Метод реализован на стороне ПК «Бастион-3» в сетевом интерфейса **PACSPort** (п. 3.4) и принимает следующие входные аргументы:

Аргумент	Описание
AccessPointToken	Идентификатор точки прохода, на которой требуется подтверждение
CredentialToken	Идентификатор пропуска
Reason	Строка с сообщением типа AccessGranted/Credential или Denied/Credential, в зависимости от принятого решения
Decision	Решение внешней системы: Granted (0) – положительное или Denied (1) – отрицательное

В значении аргумента **Reason** должна содержаться строка, в которой должно быть представлено в формате xml сообщение типа AccessGranted/Credential или Denied/Credential, в зависимости от принятого решения (подробные описания форматов событий типа AccessGranted/Credential и Denied/Credential приведены в п. 3.5).

3.7 Передача событий нарушения доступа

В случае фиксации внешней системой какого-либо нарушения доступа (например, перепрыгивание посетителем турникета, или иное событие такого рода) внешняя система может оповестить об этом ПК «Бастион-3» путём генерации события типа **AccessViolation/Credential** в следующем формате:

```
<tns1:Configuration
  xmlns:wstop="http://docs.oasisopen.org/wsn/t-1"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">
  <tns1:AccessControl wstop:topic="true">
    <AccessViolation wstop:topic="true">
      <Credential wstop:topic="true">
```

```
        <tt:MessageDescription IsProperty="false">
          <tt:Source>
            <tt:SimpleItemDescription Name="AccessPointToken" Type="pt:ReferenceToken"/>
          </tt:Source>
          <tt:Data>
            <tt:SimpleItemDescription Name="CredentialToken" Type="pt:ReferenceToken"/>
            <tt:SimpleItemDescription Name="Reason" Type="xs:string"/>
            <tt:SimpleItemDescription Name="Description" Type="xs:string"/>
          </tt:Data>
          </tt:MessageDescription>
        </Credential>
    </AccessViolation>
  </tns1:AccessControl>
</tns1:Configuration>
```

Событие может содержать следующие атрибуты:

Атрибут	Описание
AccessPointToken	Идентификатор точки прохода, которая ожидает подтверждения
CredentialToken	Идентификатор пропуска, если в событии идентифицирован его владелец
Reason	Фотография, полученная с камер видеонаблюдения, в формате JPEG, преобразованном в base64
Description	Произвольный текст с описанием нарушения

Передача событий подтверждения доступа СКУД

Для направлений прохода, настроенных в режиме идентификации, ПК «Бастион-3» генерирует событие типа **Authorized/Credential** в формате:

```
<tns1:Configuration
  xmlns:wstop="http://docs.oasisopen.org/wsn/t-1"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">
  <tns1:AccessControl wstop:topic="true">
    <Authorized wstop:topic="true">
      <Credential wstop:topic="true">
        <tt:MessageDescription IsProperty="false">
          <tt:Source>
            <tt:SimpleItemDescription Name="AccessPointToken" Type="pt:ReferenceToken"/>
          </tt:Source>
          <tt:Data>
            <tt:SimpleItemDescription Name="CredentialToken" Type="pt:ReferenceToken"/>
          </tt:Data>
        </tt:MessageDescription>
      </Credential>
    </Authorized>
  </tns1:AccessControl>
</tns1:Configuration>
```

События данного типа генерируются в тот момент, когда СКУД предоставляет доступ на направлениии прохода после прикладывания карты или после обработки события типа **AccessGranted/Credential** от внешней системы, то есть тогда, когда разблокируется

преграждающее устройство. Внешняя система имеет возможность использовать данное события для индикации о том, что посетитель может осуществить проход.

В случае, когда СКУД не предоставляет доступ на направлении прохода (в случае, если пропуск посетителя заблокирован, не имеет прав на проход в данной точке доступа, или нарушает временные зоны), генерируется событие типа **Refused/Credential** в формате:

```
<tns1:Configuration
  xmlns:wstop="http://docs.oasisopen.org/wsn/t-1"
  xmlns:tns1="http://www.onvif.org/ver10/topics"
  xmlns:tt="http://www.onvif.org/ver10/schema">
  <tns1:AccessControl wstop:topic="true">
    <Refused wstop:topic="true">
      <Credential wstop:topic="true">
        <tt:MessageDescription IsProperty="false">
          <tt:Source>
            <tt:SimpleItemDescription Name="AccessPointToken" Type="pt:ReferenceToken"/>
          </tt:Source>
          <tt:Data>
            <tt:SimpleItemDescription Name="CredentialToken" Type="pt:ReferenceToken"/>
          </tt:Data>
        </tt:MessageDescription>
      </Credential>
    </Refused>
  </tns1:AccessControl>
</tns1:Configuration>
```

Приложение 1. История изменений документа

2.0.2 – 25.11.2022

Добавлена информация о событиях Authorized и Refused.

2.0.1 – 25.11.2021

Первая версия документа.