

Making the railway system  
work better for society.

# EVR specifications for the implementation of the interfaces

Document ID: 013PPS1131-03

## *Document History*

<i>Version</i>	<i>Date</i>	<i>Comments</i>
1.0	25.11.2019	Published version
2.0	09.12.2020	Updated version (enhanced VehicleInfo data object for details & export, referring to Data Objects in annexed WSDL definition files)
3.0	15.12.2020	Updated UserDataService

## Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	Description of the update .....	3
<b>2</b>	<b>D-ARS provisioned services.....</b>	<b>4</b>
2.1	VehicleQueryService.....	4
2.1.1	IVehicleQueryService interface description .....	5
2.1.2	XSD schema.....	5
2.2	VehicleHistoryQueryService.....	6
2.2.1	IVehicleHistoryQueryservice interface description (C#).....	6
2.2.2	XSD schema.....	6
2.3	ApplicationQueryService .....	6
2.3.1	IApplicationQueryService interface description (C#).....	7
2.3.2	XSD schema.....	7
2.4	StatusService .....	7
2.4.1	IStatusService interface description (C#).....	7
2.4.2	XSD schema StatusInformation.xsd .....	8
2.5	UserDataService .....	8
2.5.1	IExportUserData interface description (C#) .....	8
2.5.2	XSD schema UserDataTransferObject.....	9
<b>3</b>	<b>EVR provided services .....</b>	<b>9</b>
3.1	ReferenceDataService .....	9
3.1.1	IGetReferenceDataService interface description (C#).....	10
3.1.2	XSD schema ReferenceDataTransferObject.....	10
3.2	VehicleQueryCallbackService .....	11
3.2.1	IAynchronousVehicleQueryCallbackService interface description (C#).....	11
3.2.2	XDS schema QueryAsynchronouslyWithCompressionCallback .....	11
<b>4</b>	<b>Organisation's codes.....</b>	<b>12</b>
<b>5</b>	<b>Security .....</b>	<b>12</b>
<b>6</b>	<b>Encoding.....</b>	<b>12</b>
<b>7</b>	<b>Annexes.....</b>	<b>12</b>
<b>8</b>	<b>VehicleInfo data object mapping version 1.0 to 2.0.....</b>	<b>13</b>

## List of Figures

Figure 1 - D-ARS provisioned services .....	4
Figure 2 - VehicleQueryService.....	4
Figure 3 - VehicleHistoryQueryService .....	6
Figure 4 - ApplicationQueryService.....	7
Figure 5 - StatusService .....	7
Figure 6 - UserDataService .....	8
Figure 7 - EVR provided services.....	9
Figure 8 - ReferenceDataService .....	10
Figure 9 - VehicleQueryCallbackService.....	11

## 1 Introduction

The Commission Implementing Decision (EU) 2018/1614 allows MSs that want to keep their Vehicle Register decentralized to do it until 2024.

The Agency has the obligation of defining the interfaces that will describe the interaction between the EVR and the D-ARS.

This guide deals with the integration of existing IT systems into the European Vehicle Register (EVR). This guide provides the technical information for IT professionals to configure their system to operate with the EVR.

There are four services which must be provided by the D-ARS systems in order to facilitate EVR access to the vehicles stored in each system. The following section provides an overview of these services.

### 1.1 Description of the update

This updated version of the specifications simplifies them in order to make them easier to be implemented, maintained and extended for future proof changes, in order to help to reduce ongoing and future development and maintenance costs.

A more detailed description of the changes is described in the following paragraphs and in chapter 8.

After version 1.0 of the original information schemas was introduced and with services implementation it was deemed necessary to release an updated version to cover missing model information, both enhance and simplify the referencing of Data Objects defined in annexed WSDL service definition files and ease their usage.

Version 2.0 of the present document revises the annexed VehicleQuery WSDL definition so that VehicleInfo Data Object schema definitions are reused to cover both the functionalities of displaying a vehicle's details and of exporting to XML. This enforces consistency between exported XML documents of Vehicle Information data and Vehicle Information details document views.

VehicleInfo data object was enhanced for its model to cover both Decision (EU) 2018/1614 and Decision 2007/756/EC based Vehicle Information data, which simplifies the consumption and promotes the consistent handling of respective data by client systems, while also easing the transition of Vehicle Information data models to the latest specifications.

Version 2.0 of the document also corrects a truncation in the original text of ApplicationInfo data object definition (now included in the annexed WSDL file for ApplicationQueryService).

Please note that section 2.3 on ApplicationQueryService will be updated in a later version.

## 2 D-ARS provisioned services

In order to facilitate MSs mission of keeping their Vehicle Register connected, or develop new systems to be connected to EVR, the Agency will reuse the current services definitions of the services exposed by the current Vehicle Register.

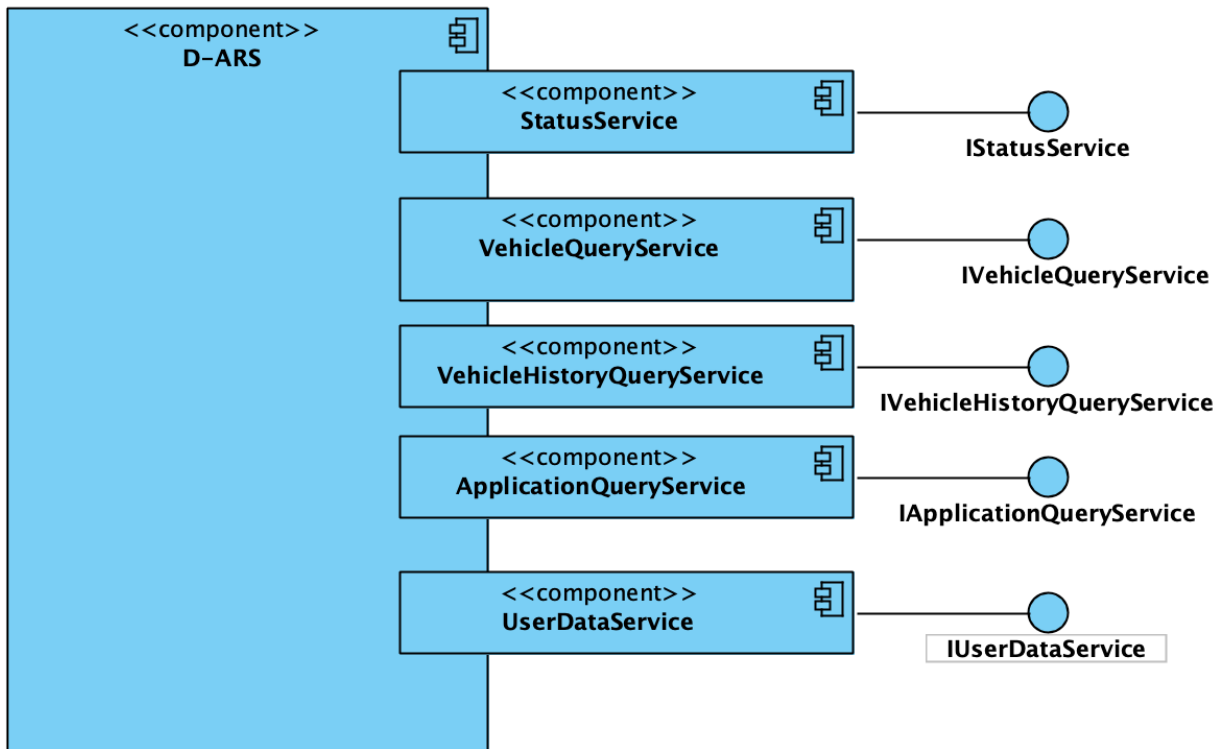


Figure 1 - D-ARS provisioned services

### 2.1 VehicleQueryService

This is the service which is called to query information of Registered vehicles in an asynchronous or synchronous manner.

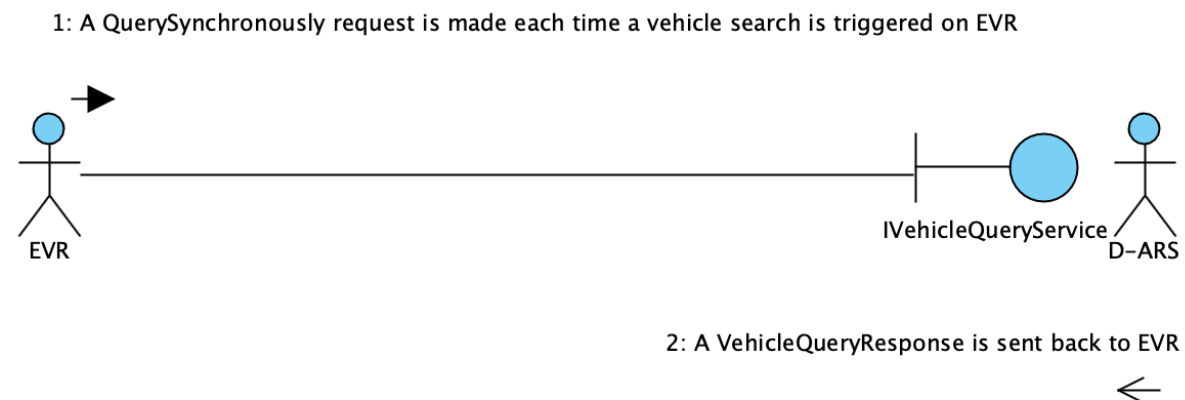


Figure 2 - VehicleQueryService

### 2.1.1 *IVehicleQueryService interface description*

```
[ServiceContract(Namespace = "http://www.era.europa.eu/EC-
VVR/Vehicle/Query/1.0", SessionMode = SessionMode.Allowed)]
public interface IVehicleQueryService
{
    [OperationContract]
    VehicleQueryResponse QuerySynchronously(string query, bool indicationIfToGet
TotalCount);

    [OperationContract]
    byte[] QuerySynchronouslyWithCompression(string query, bool indicationIfToGe
tTotalCount);

    [OperationContract(IsOneWay = true)]
    void QueryAsynchronouslyWithCompression(string query, bool indicationIfToGet
TotalCount);

    [OperationContract]
    VehicleInfo GetVehicleInformation(string evn);

    [OperationContract]
    VehicleInfo GetVehicleHistoryInformation(long? vehicleId);
}
```

In order to implement the *QueryAsynchronouslyWithCompression(string query, bool indicationIfToGetTotalCount)* it is necessary to take the following into account:

- This method is called in an asynchronous manner: the EVR will invoke the service to send the request; your system must call the VehicleQueryCallback Service on the EVR once the result is ready from your system.

- In order to do a correct call-back it is mandatory to read out the header information of the incoming message. The message id must be read out, stored and then set in the response message header (In the *RelatesTo* property). Without that the EVR cannot provide the response to the message sent.

### 2.1.2 *XSD schema*

The VehicleQueryResponse object describes the response information for searching/reporting requests and searching the history of a vehicle. Most of the times this data will be transformed into a byte array using the standard compression algorithm GZip. Its definition is included in the WSDL service definition file for VehicleQueryService (VehicleQuery.xml).

VehicleInfo data object is used by EVR to display a vehicle's details. Its definition is included in the WSDL service definition file for VehicleQueryService (VehicleQuery.xml).

XSD Schema types from VehicleInfo data object are reused in VehicleInfo.xsd for vehicle information XML export purpose.

## 2.2 VehicleHistoryQueryService

This service is called to retrieve the complete history of a registered vehicle, as it is recorded in the D-ARS system.

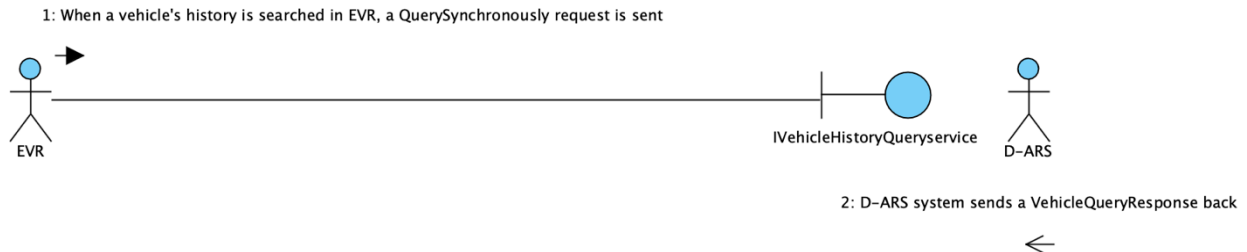


Figure 3 - VehicleHistoryQueryService

### 2.2.1 IVehicleHistoryQueryservice interface description (C#)

```

[ServiceContract(Namespace = "http://www.era.europa.eu/EC-
VVR/Vehicle/Query/1.0", SessionMode = SessionMode.Allowed)]
public interface IVehicleHistoryQueryService
{
    [OperationContract]
    VehicleQueryResponse QuerySynchronously(string query, bool indicationIfToGetTotalCount);

    [OperationContract]
    byte[] QuerySynchronouslyWithCompression(string query, bool indicationIfToGetTotalCount);
}
  
```

### 2.2.2 XSD schema

This service uses the VehicleInfo data transfer object to send the historical vehicle information from D-ARS to EVR system. It is exactly the same object specified in section 2.1.2 .

## 2.3 ApplicationQueryService

This service is called to retrieve the Application of a registered vehicle.

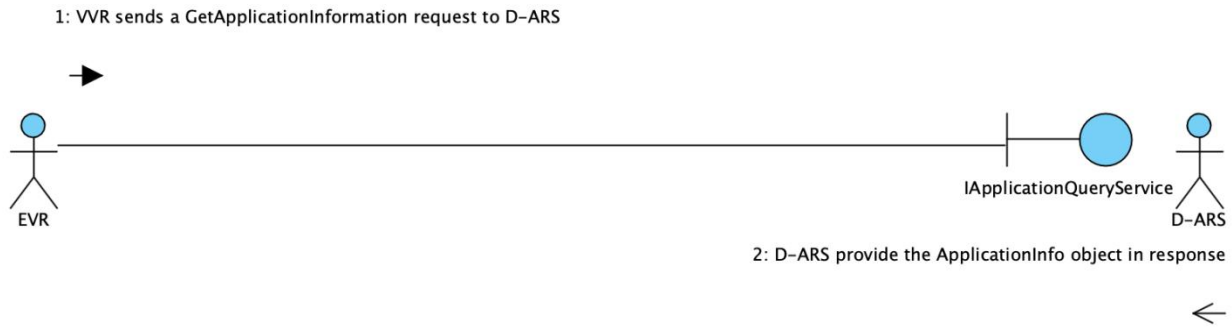


Figure 4 - ApplicationQueryService

2.3.1 IApplicationQueryService interface description (C#)

```

[ServiceContract(Namespace = "http://www.era.europa.eu/EC-VVR/Application/Query/1.0", SessionMode = SessionMode.Allowed)]
public interface IApplicationQueryService
{
    [OperationContract]
    ApplicationInfo GetApplicationInformation(string applicationId);
}
    
```

2.3.2 XSD schema

ApplicationInfo data object is used to transfer application information to the EVR. Its definition is included in the WSDL service definition file for ApplicationQueryService (ApplicationQuery.xml).

2.4 StatusService

This service is called to retrieve information about the state of the system.

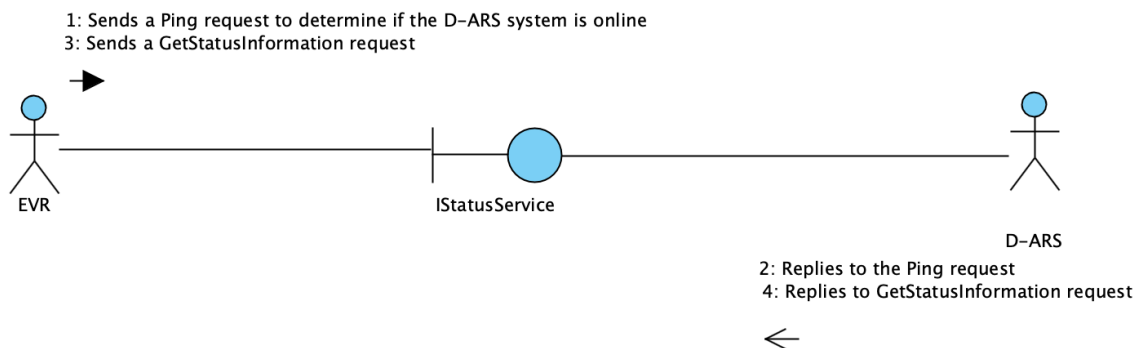


Figure 5 - StatusService

2.4.1 IStatusService interface description (C#)

```
[ServiceContract(Namespace = "http://www.era.europa.eu/EC-
VVR/Status/1.0", SessionMode = SessionMode.Allowed)]
public interface IStatusService
{

[OperationContract]
void Ping();

[OperationContract]
StatusInformation GetStatusInformation();

}
```

#### 2.4.2 XSD schema StatusInformation.xsd

StatusInformation data object is used to transfer application information to the EVR. Its definition is included in the WSDL service definition file for StatusService (Status.xml).

## 2.5 UserDataService

This service will be exposed by DARS systems and will allow EVR system to push validated users to D-ARS system.

The GetExportedUsers web service in the D-ARS returns the users that were exported to the D-ARS from the EVR calling the web service. The UserDataTransferObject is used for transferring users from EVR to D-ARS while ExportedUsersTransferObject is used to get information about the users transferred from EVR to the D-ARS.

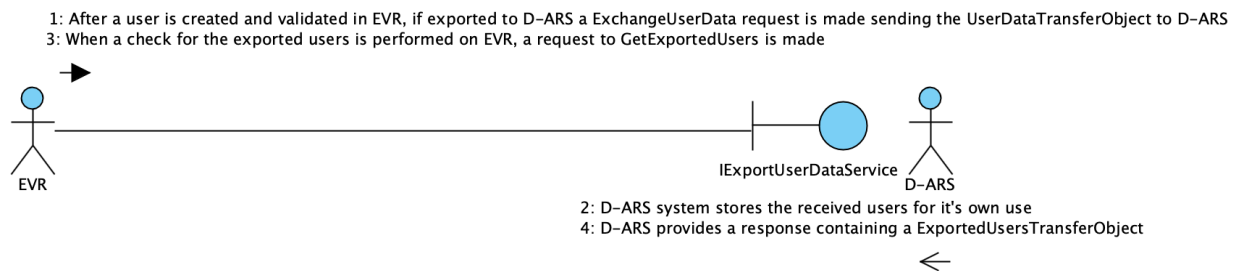


Figure 6 - UserDataService

#### 2.5.1 IExportUserData interface description (C#)

```
[ServiceContract(Namespace = "http://www.era.europa.eu/EC-VVR/UserData",
SessionMode = SessionMode.Allowed)]
public interface IExportUserDataService
{

[OperationContract]
string ExchangeUserData(UserdataTransferObject userdata);

[OperationContract]
ExportedUsersTransferObject GetExportedUsers();

}
```



```

    [OperationContract]
    ExportedUsersTransferObject GetExportedUsersForUsers (params string[]
    usernames);
    }
    
```

2.5.2 XSD schema UserDataTransferObject

UserdataTransferObject data object is used to is used to export validated EVR users to D-ARS. Its definition is included in the WSDL service definition file for UserDataService (UserData.xml).

3 EVR provided services

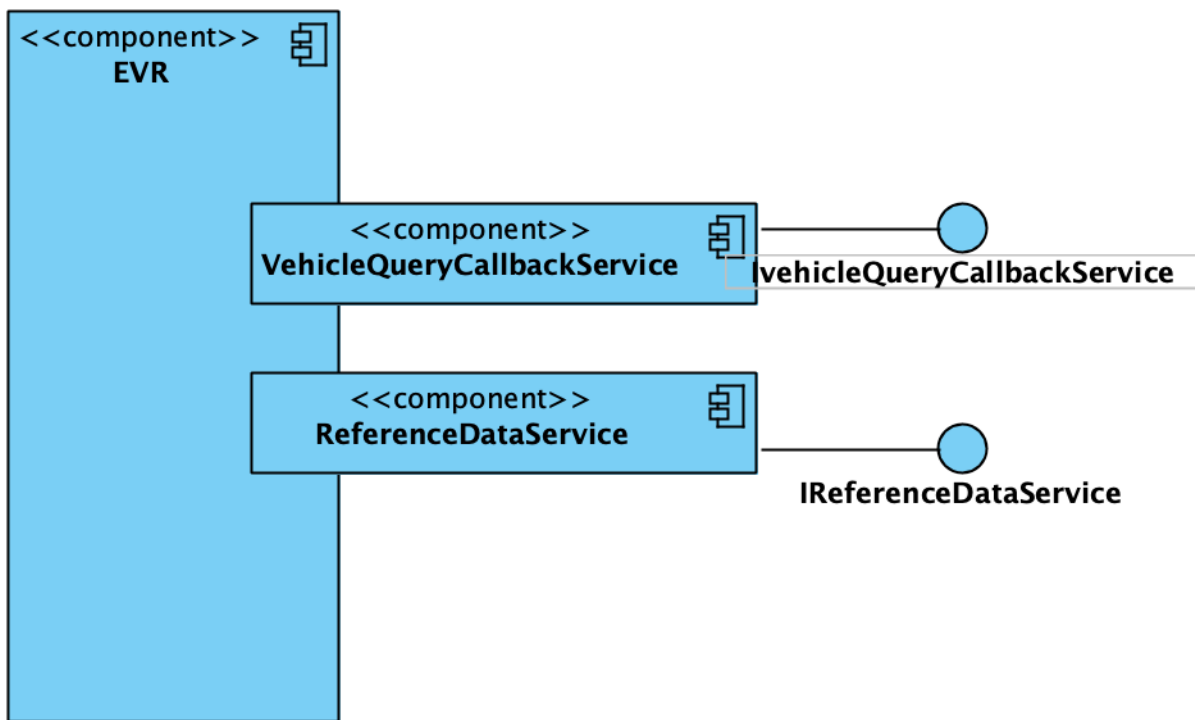


Figure 7 - EVR provided services

3.1 ReferenceDataService

This service is exposed by EVR and it allows D-ARS to retrieve the following reference data:

- Countries
- Zones
- Document types
- Safety authorities
- Coded restrictions
- EVN wizzard reference data

These data come in XML files for most of them, exception made by the EVN wizzard files which come in HTM files. All the files are zipped and send as binary content.

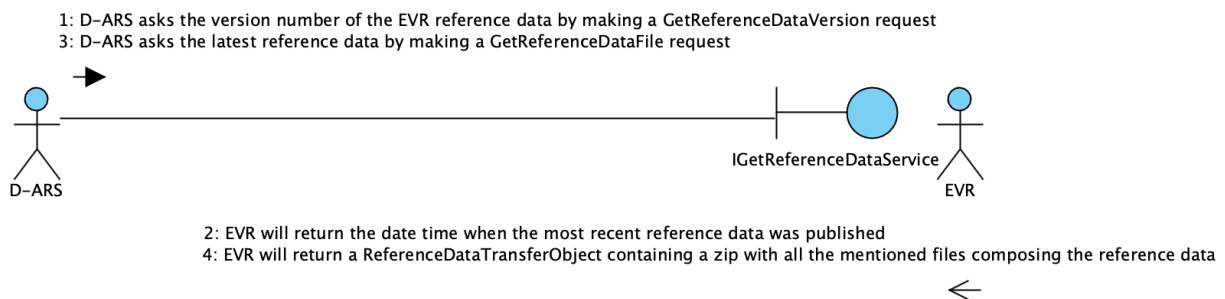


Figure 8 - ReferenceDataService

### 3.1.1 IGetReferenceDataService interface description (C#)

```

[ServiceContract(Namespace = "http://www.era.europa.eu/EC-VVR/ReferenceData",
SessionMode = SessionMode.Allowed)]
public interface IGetReferenceDataService
{
    [OperationContract]
    ReferenceDataTransferObject GetReferenceDataFile(bool compression);

    [OperationContract]
    DateTime GetReferenceDataVersion();
}
  
```

### 3.1.2 XSD schema ReferenceDataTransferObject

ReferenceDataTransferObject data object is used to send reference data from EVR to D-ARS systems. Its definition is included in the WSDL service definition file for ReferenceDataService (ReferenceData.xml).

## 3.2 VehicleQueryCallbackService

This service is used to allow D-ARS systems to send the results of a report request back to the EVR system.

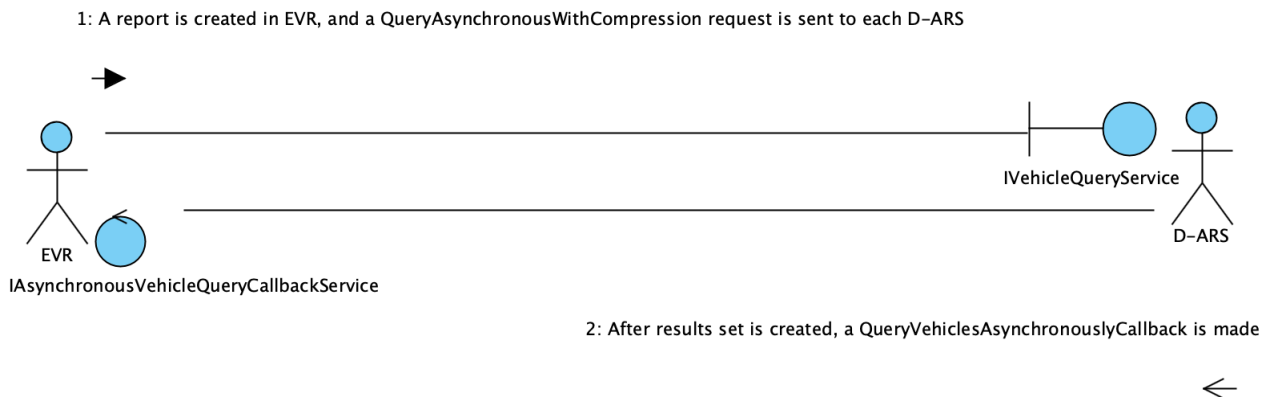


Figure 9 - VehicleQueryCallbackService

### 3.2.1 IAsynchronousVehicleQueryCallbackService interface description (C#)

```

[ServiceContract (Namespace = "http://www.era.europa.eu/EC-VVR/AsynchronousVehicleQueryCallback", SessionMode = SessionMode.Allowed)]
public interface IAsynchronousVehicleQueryCallbackService
{
    [OperationContract (IsOneWay=true)]
    void QueryVehiclesAsynchronouslyCallback(byte[] response);
}
  
```

### 3.2.2 XDS schema QueryAsynchronouslyWithCompressionCallback

`QueryAsynchronouslyWithCompressionCallback` object definition is included in the WSDL service definition file for VehicleQueryCallbackService (VehicleQueryCallback.xml).

## 4 Organisation's codes

The Agency will implement and make available inside EVR a module allowing users to search for organisations based on the following attributes:

- *Organisation Code*
- *Organisation name*
- *Registered business number*
- *Address*
- *Town*
- *Country code*
- *Post code*
- *Email address*

The primary functionality will be the search based on organisation code.

The search will return all the above specified fields plus the Vehicle Keeper Marking number in case of a Keeper organisation. This information can further be used in the process of registering the vehicle by the Keeper ( applicant for vehicle registration ) using the D-ARS system.

## 5 Security

The security used to authenticate and to encrypt the data exchange is based on X.509 certificates. The details can be viewed in the WSDL in the attachments. The certificates used should be provided by both communicating parties (one for the EVR and one for the D-ARS).

Security certificates to secure the EVR will be issued by the Agency, the same can be done for the D-ARS in case the organization hosting them don't have a security certificate that could be used.

## 6 Encoding

All xml messages passed use UTF-8 data encoding. This can be seen in header of the schema files.

## 7 Annexes

To the current document we annex the following below mentioned files:

- *ApplicationQuery.xml*
- *ReferenceData.xml*
- *Status.xml*
- *UserData.xml*
- *VehicleHistoryQuery.xml*
- *VehicleQuery.xml*
- *VehicleQueryCallback.xml*

All the above are the WSDL files for the respective services. Please be aware that the sample WSDL files contain data specific to where the services are currently hosted and used X509 security certificates.

Additionally we have attached ReferenceData.zip file containing the reference data composed of:

- Countries
- Zones
- Document types
- Safety authorities
- Coded restrictions
- EVN wizzard reference data.

## 8 VehicleInfo data object mapping version 1.0 to 2.0

Changes from version 1.0 to version 2.0 of VehicleInfo data object model are described in detail below.

### Schema header

	version="2.0"
targetNamespace="http://schemas.datacontract.org/2004/07/"	targetNamespace="http://www.era.europa.eu/EC-VVR/Vehicle/Query/1.0"
xmlns:tns="http://schemas.datacontract.org/2004/07/"	xmlns="http://www.era.europa.eu/EC-VVR/Vehicle/Query/1.0"

### VehicleInfo type

VehicleInfo type requires specific order of fields (xs:sequence)	VehicleInfo type allows any order of fields (xs:all)
	<b>ADDED</b> <b>DateTime of XML vehicle details generation/exporting</b> <xs:element name="TimestampUTC" type="xs:dateTime" nillable="true"/>
	<b>ADDED</b> <xs:element name="Regime" type="Regime"/>
<xs:element name="_prevVehicleState" nillable="true" type="tns:VehicleInfo" />	<xs:element name="PreviousVehicleState" type="VehicleInfo" minOccurs="0"/>
<xs:element name="additionalAuthorisationsField" nillable="true" type="tns:ArrayOfAdditionalAuthorisation" />	<xs:element name="AdditionalAuthorisations" type="AdditionalAuthorisations" minOccurs="0"/>
<xs:element name="authorisationDateField" type="xs:dateTime" />	In Authorisation type: <xs:element name="Date" type="xs:dateTime" nillable="true"/>
<xs:element name="authorisationExpireDateField" type="xs:dateTime" />	In Authorisation type: <xs:element name="ExpirationDate" type="xs:dateTime" nillable="true"/>
<xs:element name="authorisedCountriesField" nillable="true" type="xs:string" />	In Authorisation type: <xs:element name="AuthorisedCountries" type="xs:string"/>

<xs:element name="authorizingNSAField" nillable="true" type="xs:string" />	In Authorisation type: <xs:element name="AuthorisingEntityName" type="xs:string" minOccurs="0"/>
<xs:element name="codedRestrictionsField" nillable="true" type="xs:string" />	In Authorisation type: <xs:element name="CodedRestrictions" type="xs:string" minOccurs="0"/>
<xs:element name="eCIssuingBodyInformationField" nillable="true" type="tns:ECIssuingBodyInformation" />	<xs:element name="ECIssuingBodyInformation" type="ECDeclarationInformation" minOccurs="0"/>
<xs:element name="eINField" nillable="true" type="xs:string" />	In Authorisation type: <xs:element name="EIN" type="xs:string"/>
<xs:element name="eVNField" nillable="true" type="xs:string" />	<xs:element name="EVN" type="xs:string"/>
<xs:element name="ecDeclarationDateField" type="xs:dateTime" />	In ECDeclarationInformation type of ECIssuingBodyInformation field: <xs:element name="ECDeclarationDate" type="xs:dateTime" nillable="true"/>
<xs:element name="ecDeclarationReferenceField" nillable="true" type="xs:string" />	In ECDeclarationInformation type of ECIssuingBodyInformation field: <xs:element name="ECDeclarationReference" type="xs:string"/>
<xs:element name="eraTvReferenceField" nillable="true" type="xs:string" />	<xs:element name="EraTvReference" type="xs:string"/>
<xs:element name="eraTvWebAddressField" nillable="true" type="xs:string" />	<xs:element name="EraTvWebAddress" type="xs:string" minOccurs="0"/>
<xs:element name="isSuspendedField" type="xs:boolean" />	In Authorisation type: <xs:element name="IsSuspended" type="xs:boolean" nillable="true"/>
<xs:element name="keeperInformationField" nillable="true" type="tns:KeeperInformation" />	<xs:element name="KeeperInformation" type="KeeperInformation"/>
<xs:element name="maintenanceEntityInformationField" nillable="true" type="tns:MaintenanceEntityInformation" />	<xs:element name="ECMInformation" type="OrganisationInformation"/>
<xs:element name="manufacturingYearField" nillable="true" type="xs:string" />	<xs:element name="ManufacturingYear" type="xs:integer"/>
<xs:element name="nonCodedRestrictionsField" nillable="true" type="xs:string" />	In Authorisation type: <xs:element name="NonCodedRestrictions" type="NonCodedRestrictions" minOccurs="0"/>
<xs:element name="nsaCountryField" nillable="true" type="xs:string" />	<b>ONLY FOR Decision 2007/756/EC</b> <xs:element name="NSACountry" type="xs:string" minOccurs="0"/>
	<b>ADDED, ONLY FOR Decision (EU) 2018/1614</b> <xs:element name="MemberStateOfRegistration" type="xs:string" minOccurs="0"/>
<xs:element name="ownerInformationField" nillable="true" type="tns:OwnerInformation" />	<xs:element name="OwnerInformation" type="OrganisationInformation"/>
<xs:element name="previousEvnField" nillable="true" type="xs:string" />	<xs:element name="PreviousEVN" type="xs:string"/>
<xs:element name="rRSOwnerInformationField" nillable="true" type="tns:RRSOwnerInformation" />	In ECDeclarationInformation type of RollingStock field of ECDeclarationOfVerification type/field: <xs:element name="RollingStock" type="ECDeclarationInformation" minOccurs="0"/>
	<b>ADDED, ONLY FOR Decision (EU) 2018/1614</b> In ECDeclarationInformation type of RollingStock field of ECDeclarationOfVerification type/field:

	<xs:element name="OnBoardCCS" type="ECDeclarationInformation" minOccurs="0"/>
<xs:element name="registrationTypeField" type="tns:RegistrationType" />	<b>REMOVED (obsolete)</b>
<xs:element name="rrsReferenceField" nillable="true" type="xs:string" />	In ECDeclarationInformation type of RollingStock field of ECDeclarationOfVerification type/field: <xs:element name="RollingStock" type="ECDeclarationInformation" minOccurs="0"/>
<xs:element name="seriesField" nillable="true" type="xs:string" />	<xs:element name="Series" type="xs:string"/>
<xs:element name="withdrawalDateField" type="xs:dateTime" />	<b>ONLY FOR Decision 2007/756/EC</b> In WithdrawalInformation type/field: <xs:element name="WithdrawalDate" type="xs:dateTime" nillable="true"/>
<xs:element name="withdrawalModeField" nillable="true" type="xs:string" />	<b>ONLY FOR Decision 2007/756/EC</b> In WithdrawalInformation type/field: <xs:element name="WithdrawalMode" type="xs:string"/>
	<b>ADDED, ONLY FOR Decision (EU) 2018/1614</b> <xs:element name="RegistrationStatusInformation" type="RegistrationStatusInformation" minOccurs="0"/>
	<b>ADDED</b> <b>2.2 in Decision 2007/756/EC</b> <b>11.1 in Decision (EU) 2018/1614</b> In Authorisation type/field: <xs:element name="AuthorisingEntityName" type="xs:string" minOccurs="0"/>
	<b>ADDED</b> In Authorisation type/field: <xs:element name="AuthorisingEntityCountryCode" type="xs:string"/>

#### AdditionalAuthorisation type (renamed/merged with Authorisation type)

AdditionalAuthorisation	Authorisation (in AdditionalAuthorisations parent)
<xs:element name="authorisationDateField" nillable="true" type="xs:dateTime" />	<xs:element name="Date" type="xs:dateTime" nillable="true"/>
<xs:element name="authorisationExpirationDateField" nillable="true" type="xs:dateTime" />	<xs:element name="ExpirationDate" type="xs:dateTime" nillable="true"/>
<xs:element name="authorisedCountriesField" nillable="true" type="xs:string" />	<xs:element name="AuthorisedCountries" type="xs:string"/>
<xs:element name="authorizingNSAField" nillable="true" type="xs:string" />	<xs:element name="AuthorisingEntityName" type="xs:string" minOccurs="0"/>
<xs:element name="codedRestrictionsField" nillable="true" type="xs:string" />	<xs:element name="CodedRestrictions" type="xs:string" minOccurs="0"/>
<xs:element name="einField" nillable="true" type="xs:string" />	<xs:element name="EIN" type="xs:string"/>
<xs:element name="evnField" nillable="true" type="xs:string" />	<b>REMOVED (always equal to VehicleInfo.EVN)</b>
<xs:element name="isSuspendedField" type="xs:boolean" />	<xs:element name="IsSuspended" type="xs:boolean" nillable="true"/>

<xs:element name="nonCodedRestrictionsField" nillable="true" type="xs:string" />	<xs:element name="NonCodedRestrictions" type="NonCodedRestrictions" minOccurs="0"/>
<xs:element name="typeField" nillable="true" type="xs:string" />	<b>REMOVED (obsolete)</b>

### ECIssuingBodyInformation type (reusing ECDeclarationInformation type instead)

ECIssuingBodyInformation	Reusing ECDeclarationInformation
<xs:element name="countryCodeField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="CountryCode" type="xs:string"/>
<xs:element name="emailField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="Email" type="xs:string"/>
<xs:element name="nameField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="OrganisationName" type="xs:string"/>
<xs:element name="registeredBusinessNumberField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="RegisteredBusinessNumber" type="xs:string"/>
<xs:element name="streetAndNumberField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="StreetAndNumber" type="xs:string"/>
<xs:element name="townField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="Town" type="xs:string"/>
<xs:element name="zipCodeField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="ZipCode" type="xs:string"/>
	<b>MOVED</b> from VehicleInfo <xs:element name="ECDeclarationDate" type="xs:dateTime" nillable="true"/>
	<b>MOVED</b> from VehicleInfo <xs:element name="ECDeclarationReference" type="xs:string"/>

### KeeperInformation type

	Extends OrganisationInformation
<xs:element name="countryCodeField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="CountryCode" type="xs:string"/>
<xs:element name="emailField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="Email" type="xs:string"/>
<xs:element name="nameField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="OrganisationName" type="xs:string"/>
<xs:element name="registeredBusinessNumberField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="RegisteredBusinessNumber" type="xs:string"/>
<xs:element name="streetAndNumberField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="StreetAndNumber" type="xs:string"/>
<xs:element name="townField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="Town" type="xs:string"/>
<xs:element name="vKMField" nillable="true" type="xs:string" />	<xs:element name="VKM" type="xs:string"/>
<xs:element name="zipCodeField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="ZipCode" type="xs:string"/>



**MaintenanceEntityInformation type (reusing OrganisationInformation instead)**

MaintenanceEntityInformation	Reusing OrganisationInformation
<xs:element name="countryCodeField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="CountryCode" type="xs:string"/>
<xs:element name="emailField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="Email" type="xs:string"/>
<xs:element name="nameField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="OrganisationName" type="xs:string"/>
<xs:element name="registeredBusinessNumberField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="RegisteredBusinessNumber" type="xs:string"/>
<xs:element name="streetAndNumberField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="StreetAndNumber" type="xs:string"/>
<xs:element name="townField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="Town" type="xs:string"/>
<xs:element name="zipCodeField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="ZipCode" type="xs:string"/>

**OwnerInformation type (reusing OrganisationInformation instead)**

OwnerInformation	Reusing OrganisationInformation
<xs:element name="countryCodeField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="CountryCode" type="xs:string"/>
<xs:element name="emailField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="Email" type="xs:string"/>
<xs:element name="nameField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="OrganisationName" type="xs:string"/>
<xs:element name="registeredBusinessNumberField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="RegisteredBusinessNumber" type="xs:string"/>
<xs:element name="streetAndNumberField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="StreetAndNumber" type="xs:string"/>
<xs:element name="townField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="Town" type="xs:string"/>
<xs:element name="zipCodeField" nillable="true" type="xs:string" />	In OrganisationInformation ancestor type: <xs:element name="ZipCode" type="xs:string"/>

**RRSOwnerInformation type**

Obsolete, removed and using ECDeclarationOfVerification type instead, containing RollingStock and OnBoardCCS items of type ECDeclarationInformation. Note that ECDeclarationInformation is reused for ECIssuingBodyInformation field of VehicleInfo too.

RRSOwnerInformation	ECDeclarationOfVerification
	<b>Replacing RRSOwnerInformation ONLY FOR Decision (EU) 2018/1614</b>

	<xs:element name="RollingStock" type="ECDeclarationInformation" minOccurs="0"/>
	<b>ADDED, ONLY FOR Decision (EU) 2018/1614</b> <xs:element name="OnBoardCCS" type="ECDeclarationInformation" minOccurs="0"/>

## Registration type

Obsolete, removed

## Added types

VehiclesInfo	helper type, not actively used
Regime	xs:string restriction allowing values "Decision 2007/756/EC" and "Decision (EU) 2018/1614"
ECDeclarationOfVerification <b>ONLY FOR Decision (EU) 2018/1614</b>	contains "RollingStock", "OnBoardCCS", both as ECDeclarationInformation
OrganisationInformation	base type for ECDeclarationInformation and KeeperInformation, also used for ECIssuingBodyInformation, OwnerInformation and ECMinformation fields of VehicleInfo
WithdrawalInformation <b>ONLY FOR Decision 2007/756/EC</b>	Encapsulates WithdrawalMode and WithdrawalDate
RegistrationStatusInformation <b>ONLY FOR Decision (EU) 2018/1614</b>	Encapsulates RegistrationStatus (corresponds to WithdrawalMode), RegistrationStatusDate (corresponds to WithdrawalDate), RegistrationStatusReason
Authorisation	Encapsulating authorisation-related fields of VehicleInfo and reused at child nodes of AdditionalAuthorisations, replacing AdditionalAuthorisation
NonCodedRestrictions	Contains xs:string NonCodedRestriction children
AdditionalFields <b>ONLY FOR Decision (EU) 2018/1614</b>	Contains AdditionalField children
AdditionalField <b>ONLY FOR Decision (EU) 2018/1614</b>	Contains Name, Value xs:string pair