



# **Badge Lead Management System**

## **ESC Delegate Web Service “Lead Management Real Time”**

Technical User Guide

Version 1.8 – May 2022

ESC Projects and Systems Division

# Contents

1. Introduction.....	3
1.1. Overview .....	3
1.2. Glossary.....	3
1.3. Terminology .....	3
2. System Overview .....	4
2.1. Architecture .....	4
2.2. Technology.....	4
2.3. Network Architecture .....	4
2.4. Error handling and logging.....	5
2.5. Disaster Recovery.....	5
3. Accessing the ESC Delegate Web Service .....	6
3.1. Querying the Web Service .....	6
3.1.1. Badge Identifier.....	6
3.1.2. Web Service Request .....	8
3.2. Delegate Data.....	8
3.3. Processing Web Service Response.....	9
3.3.1. Delegate XML format .....	9
3.3.2. Web Service Response .....	9
4. Security .....	11
4.1. Authentication .....	11
4.2. Authorization .....	11
4.3. Data Protection .....	11
5. Appendix A – Error Codes.....	12
5.1. System Configuration Errors .....	12
5.2. Application Errors .....	12
5.3. Client Errors .....	13
6. Appendix B – Badge.....	14
7. Appendix C – Web Service Request / Response samples.....	15
7.1. Request sample.....	15
7.2. Response sample .....	15
7.3. Null Response sample.....	15
8. Appendix D – Test Web Service.....	17

## 1. Introduction

### 1.1. Overview

This document describes the ESC Delegate Web Service interface that will allow ESC industry partners to read and access ESC delegates' information from badges in real time, enabling your exhibit or Satellite Symposium staff to store, edit and amend delegate data using your own hardware and software.

Intended users of this interface are companies who owns lead management technologies, and who can implement a programme interfacing the ESC interface and the lead management system owned.

This document is a technical user guide, aimed for the person(s) who will implement this programme.

### 1.2. Glossary

Define the different actors / systems mentioned in the documentation.

- **ESC Delegate Web Service:** the interface described in this document. The interface may also be referenced in the document as simply "the interface".
- **Partner system:** the lead management system that is owned by the industry partner.

### 1.3. Terminology

Define the different terms used in the documentation.

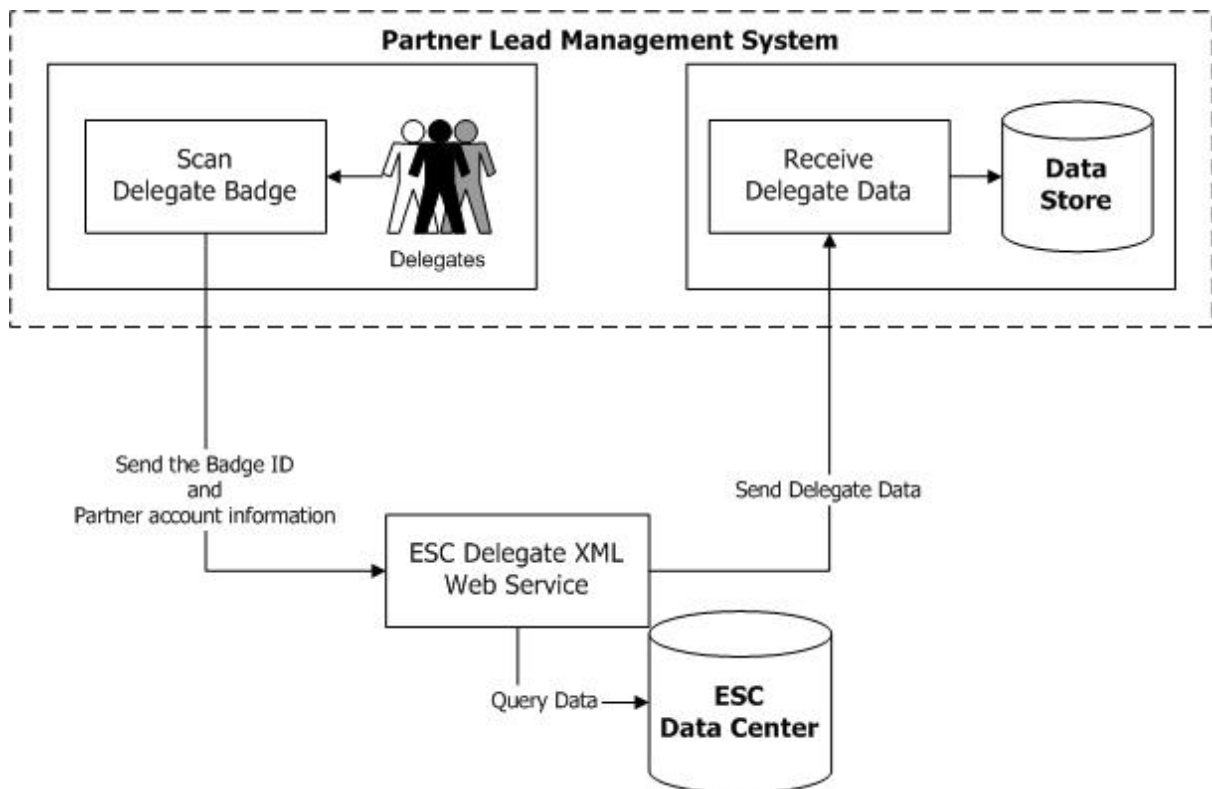
Term	Description
XML	Extensible Markup Language. Its primary purpose is to facilitate the sharing of data across different information systems, particularly systems connected.
Web Service	Software system designed to support interoperable Machine to Machine interaction over a network. Web services are frequently just Web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.
WSDL	Web Services Description Language – XML based language that provides a model for describing Web services.
SOAP	Simple Object Access Protocol. Protocol for exchanging XML-based messages over computer networks, normally using HTTP. SOAP forms the foundation layer of the Web services stack, providing a basic messaging framework.
HTTP	Hypertext Transfer Protocol will be used to query the Web Services and retrieve the XML data.
SSL	Secure Socket Layer is one of the cryptographic protocols which provide secure communications on the Internet and other data transfers.
HTTPS	Secure protocol based on the combination of HTTP and SSL.

## 2. System Overview

### 2.1. Architecture

The diagram outlines the overall system architecture.

1. The partner system is responsible for scanning the badge identifier, calling the ESC Delegate Web Service, and processing the results.
2. The ESC Delegate Web Service is responsible for returning the delegate data for the badge identifier that was sent.



### 2.2. Technology

The technologies used are the HTTP, SOAP and XML standards limiting the integration issues with partner systems.

### 2.3. Network Architecture

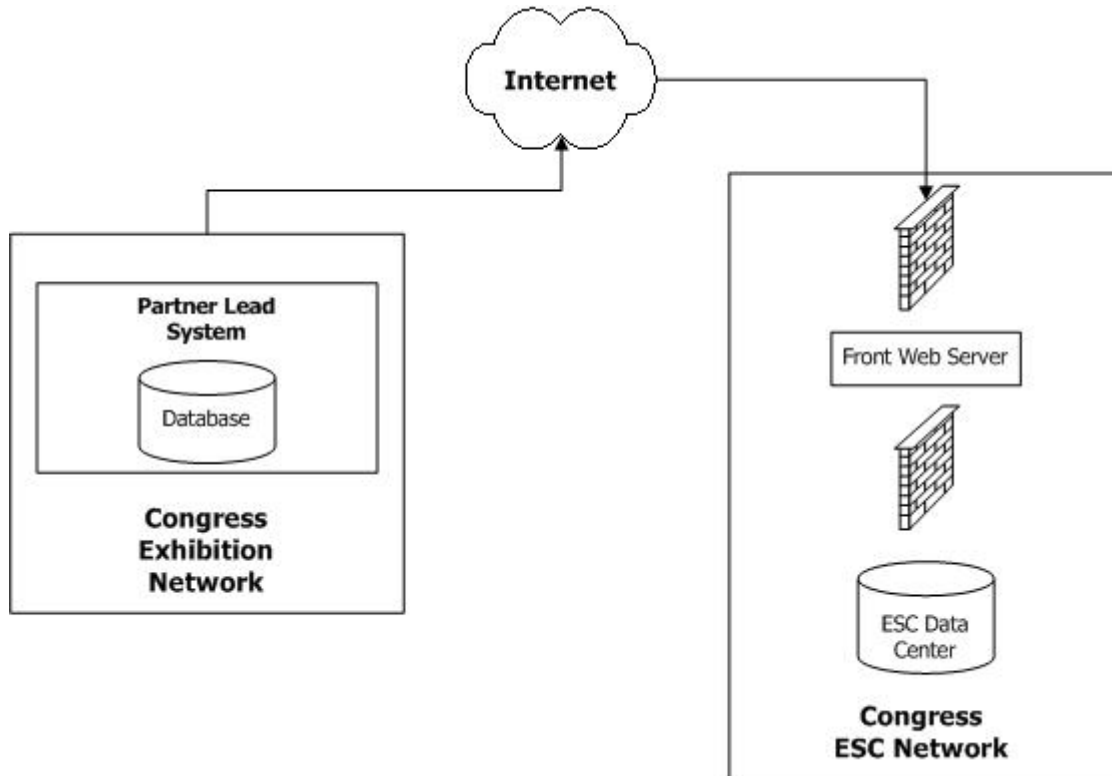
The diagram below shows the network infrastructure of the system.

The ESC Delegate Web Service will be accessible via the internet with a URL (e.g., <https://partners.escardio.org/delegateWS/DelegateInformation.aspx>)

A test environment is provided by the ESC in order to implement the service prior to the service. The details are available in [Appendix D](#).

The WSDL will be available by appending? WSDL to the URL (e.g., <https://partners.escardio.org/delegateWS/DelegateInformation.aspx?WSDL>)

**NOTE:** The ESC Delegate Web Service URL is subject to change. The URL should be easily configurable by the partner if a change occurs.



#### **2.4. Error handling and logging**

The ESC Delegate Web Service handles errors gracefully by returning SOAP exceptions when an error occurs. These exceptions must be handled properly by the programme of the partner integrating the systems.

A list of possible errors is listed in [Appendix A – Error Codes](#).

All requests sent to the system are logged in the ESC Data Centre. The data is logged for disaster recovery as detailed further in this document.

#### **2.5. Disaster Recovery**

The system is designed so that ESC administrators could provide delegate information after the fact through an XML file for a given list of badge identifiers.

In order for the disaster recovery plan to work, it is assumed that the partner systems also keep track of the badge identifiers that are scanned, even if the ESC Delegate Web Service did not respond.

### **3. Accessing the ESC Delegate Web Service**

The data exchange occurring between the systems is based on XML, encapsulated by the SOAP protocol. A SOAP request is issued to the Web Service that sends back a SOAP response.

#### **3.1. Querying the Web Service**

In order to query the Delegate Web Service, the caller will need to pass the following parameters:

- Partner Account and Key
- Badge Identifier
- Additional Information

The Partner account and key will uniquely identify and validate the partner, granting him privileges. The account format will be GUID. The account is only valid for a certain time only.

A GUID should contain 32 digits with 4 dashes (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx)

**Partner account sample:** << given by email >>

The additional information is currently not used. It expects a text string.

#### **3.1.1. Badge Identifier**

The badge identifier uniquely identifies a delegate and can be extracted by reading the QR Code on the delegate badge. The badge identifier is of type integer.

# Christophe Graziani

FRANCE



**DELEGATE**

## Figure 1 - Badge Sample

The badge identifier is 0202233S000560985 for the above sample.

### 3.1.2. Web Service Request

To query the Web Service, the programme will send a SOAP request as shown below.

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RetrieveDelegateInformation
xmlns="http://www.escardio.org/webservices/DelegateInformation">
      <account>string</account>
      <key>string</key>
      <badgeID>int</badgeID>
      <additionalInformation>string</additionalInformation>
    </RetrieveDelegateInformation>
  </soap:Body>
</soap:Envelope>
```

A request sample is available in [Appendix C – Web Service Request Sample](#).

### 3.2. Delegate Data

The delegate data returned by the ESC Delegate Web Service is summarised in the table below. All data is returned as text. Some data may not be available depending on the information provided by the delegate and is outlined as null-able.

Name	Null-able	Length	Comments
<b>General</b>			
Title		80	Professor, Doctor, Mister, etc...
Family Name		100	
Initials		8	
First Name		40	
Email		80	
<b>Activity</b>			
Profession	X	80	
Sub-specialties	X		0 to n separated by semi-colon
<b>Address</b>			
Address Line 1	X	45	
Address Line 2	X	45	
Address Line 3	X	45	
Address Line 4	X	45	
Postal Code	X	15	
City		80	



Country		50	
CountryCode		2	Country ISO Code
Phone	X	80	
Fax	X	80	

### 3.3. Processing Web Service Response

The Web Service sends back the response as XML embedded in a SOAP envelope.

#### 3.3.1. Delegate XML format

The Delegate Information is sent as XML as shown below:

```
<Delegate xmlns="http://www.escardio.org/webservices/DelegateInformation/Delegate">
  <Title>string</Title>
  <FamilyName>string</FamilyName>
  <FirstName>string</FirstName>
  <Initials>string</Initials>
  <Profession>string</Profession>
  <Specialties>string</Specialties>
  <Email>string</Email>
  <AddressLine1>string</AddressLine1>
  <AddressLine2>string</AddressLine2>
  <AddressLine3>string</AddressLine3>
  <AddressLine4>string</AddressLine4>
  <PostalCode>string</PostalCode>
  <City>string</City>
  <Country>string</Country>
  <CountryCode>string</CountryCode>
  <Phone>string</Phone>
  <Fax>string</Fax>
</Delegate>
```

#### 3.3.2. Web Service Response

The Web Service response that is sent back is a SOAP response embedding the XML information:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <RetrieveDelegateInformationResponse
  xmlns="http://www.escardio.org/webservices/DelegateInformation">
      <Delegate xmlns="http://www.escardio.org/webservices/DelegateInformation/Delegate">
        <Title>string</Title>
        <FamilyName>string</FamilyName>
        <FirstName>string</FirstName>
        <Initials>string</Initials>
        <Profession>string</Profession>
        <Specialties>string</Specialties>
        <Email>string</Email>
        <AddressLine1>string</AddressLine1>
        <AddressLine2>string</AddressLine2>
```

```
<AddressLine3>string</AddressLine3>  
  <AddressLine4>string</AddressLine4>  
<PostalCode>string</PostalCode>  
<City>string</City>  
<Country>string</Country>  
  <CountryCode>string</CountryCode>  
<Phone>string</Phone>  
<Fax>string</Fax>  
</Delegate>  
</RetrieveDelegateInformationResponse>  
</soap:Body>  
</soap:Envelope>
```

A response sample is available in [Appendix C – Web Service Response Sample](#).

In case no delegate information is available, the response contains a null value. A null response sample is available in [Appendix C – Web Service Null Response Sample](#).

**Note:** Delegate information is available up to 15 minutes after the delegate registers.

## **4. Security**

Security encompasses different but related things: authentication, authorization, and data protection. It is really important to the ESC in order to protect the delegate data that will be sent back.

### ***4.1. Authentication***

Authentication is the process of verifying the digital identity of the sender of a communication.

The account and key provided to each partner will be used to authenticate the sender. Both the account and key must be sent with the badge identifier.

Authentication will occur for each request.

### ***4.2. Authorization***

Authorization is the process of checking that the consumer has been granted permission to use that resource. The system will enforce who can access the system and when.

### ***4.3. Data Protection***

The data transferred is encrypted using SSL over HTTP. All data exchanged between the partner system and the ESC Delegate Web Service is fully encrypted.

This approach secures both the partner account information, preventing abuse of that account and the delegate data that is sent back.

## 5. Appendix A – Error Codes

### 5.1. System Configuration Errors

- **Error 001 - Application Configuration Error**

Raised if the ESC Delegate Web Service is not properly configured.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Error 001 - Application Configuration Error - Please contact the
administrator!</faultstring>
      <detail/>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

- **Error 002 - Database Configuration Error**

Raised if the ESC Delegate Web Service is not properly configured.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Error 002 - Database Configuration Error - Please contact the
administrator!</faultstring>
      <detail/>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

### 5.2. Application Errors

- **Error 101 - Parameter not set properly**

Raised if the ESC Delegate Web Service did not receive all mandatory parameters (Partner account and key, Badge Identifier).

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Error 101 - Parameter not set properly!</faultstring>
      <detail/>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

```
</soap:Fault>
</soap:Body>
</soap:Envelope>
```

- **Error 102 - Invalid account**

Raised if the caller did not pass a valid partner account.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Error 102 - Invalid account!</faultstring>
      <detail/>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

### 5.3. Client Errors

Some errors may also be raised if the SOAP request issued by the partner system is not correct. It could be the case if a string is passed as the badge identifier instead of an integer.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Client</faultcode>
      <faultstring>Server was unable to read request. ---> There is an error in XML document (6,
35). ---> Input string was not in a correct format.</faultstring>
      <detail/>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

**6. Appendix B – Badge**

Badge Identifier = 0202233S000560985

**Christophe  
Graziani**

**FRANCE**



**DELEGATE**

## 7. Appendix C – Web Service Request / Response samples

### 7.1. Request sample

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:del="http://www.escardio.org/webservices/DelegateInformation">
  <soap:Header/>
  <soap:Body>
    <del:RetrieveDelegateInformation>
      <del:account>b8df01d9-1aa6-4277-af54-4ac33722cdf0</del:account>
      <del:key>123</del:key>
      <del:badgeID>0202233EEH2200103</del:badgeID>
      <del:additionalInformation>Information</del:additionalInformation>
    </del:RetrieveDelegateInformation>
  </soap:Body>
</soap:Envelope>
```

### 7.2. Response sample

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <TestDelegateInformationResponse
xmlns="http://www.escardio.org/webservices/DelegateInformation">
      <Delegate xmlns="http://www.escardio.org/webservices/DelegateInformation/Delegate">
        <Title>Title</Title>
        <FamilyName>Last name</FamilyName>
        <FirstName>First name</FirstName>
        <Initials>Initials</Initials>
        <Profession>Profession</Profession>
        <Specialty>Specialty</Specialty>
        <Email>email address</Email>
        <AddressLine1>2035 route des colles</AddressLine1>
        <AddressLine2/>
        <AddressLine3/>
        <AddressLine4/>
        <PostalCode>06903</PostalCode>
        <City>Sophia-Antipolis</City>
        <Country>France</Country>
        <CountryCode>FR</CountryCode>
        <Phone/>
        <Fax/>
      </Delegate>
    </TestDelegateInformationResponse>
  </soap:Body>
</soap:Envelope>
```

### 7.3. Null Response sample

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<soap:Body>
  <RetrieveDelegateInformationResponse
xmlns="http://www.escardio.org/webservices/DelegateInformation">
  <Delegate
xmlns="http://www.escardio.org/webservices/DelegateInformation/Delegate"/>
  </RetrieveDelegateInformationResponse>
</soap:Body>
</soap:Envelope>
```

xsi:nil="true"



## 8. Appendix D – Test Web Service

A test account is available at the following address (URL):

<https://partners.escardio.org/delegateWS/DelegateInformation.aspx>

**Note that the test version is using a self-signed certificate, which may result in some warnings or errors depending on your implementation.**

Below are the details for the test account and key:

- Account: <<given by email>>
- Key: <<given by email>>

**Note:** the key is case-sensitive.

Some badge bar codes are given in JPEG format to perform the tests. The value for each barcode is displayed below the barcode. This is the value of the delegate ID that needs to be sent to the Web Service.

The test data are simulated data, with no actual names.