eMIP Protocol

Implementation Guide

GIREVE

29/07/2019

Summaries

eMIP Protocol Version Reference

This eMIP Protocol Implementation Guide document is based on eMIP v0.7.4, as defined on document:

- eMIP Protocol Description : Gireve_Tech_eMIP-V0.7.4_ProtocolDescription_1.0.11_en.pdf

Related documents

Doc reference	Content	Doc name
Cnx_Project_Mgt	Connection Project	Gireve_Tech_Cnx_ConnectionProjectManagement_1.1.5_en.pdf
eMIP_Protocol_Descr	eMIP Protocol description	Gireve_Tech_eMIP-V0.7.4_ProtocolDescription_1.0.11_en.pdf

1 License of use of the eMIP protocol

> Any use of the eMIP protocol by the User involves acceptance of the terms of the present license.
>
> If you do not wish to accept these conditions, you shall refrain from using the eMIP protocol.

1.1 Preamble

The eMIP protocol (**eM**obility **I**nter-operation **P**rotocol) is an open protocol which describes the rules of communication allowing notably the transfer of data and the consumption of services between the software platform of an operator and the platform of GIREVE. This protocol is the fruit of hard work and investments by GIREVE and is part of its know-how and its intellectual property.

1.2 Intellectual Property

The eMIP protocol is the exclusive property of GIREVE in accordance with the provisions of the Code of the intellectual property.

It is the same for the documentation of any nature, technical or commercial, which accompanies eventually the eMIP protocol.

The present license of use granted by GIREVE does not imply any transfer of intellectual property rights to the User.

1.3 Scope of the license

GIREVE concedes to the User a non-exclusive, non-transferable license of use of the eMIP protocol, including its documentation, worldwide in the course of its usual professional activities and notably in order to develop software products based on the eMIP protocol.

Any non-authorized use is strictly prohibited, notably any modification, communication, distribution and commercialization of the eMIP protocol as such by the User in any form whatsoever.

The present license is conceded free of any charge and without any time limit. However, the present license could be terminated automatically ipso jure and without any formality or prior notice in the event of infringement of the terms of present license by the User, notably in case of an act that infringes the intellectual property rights of GIREVE.

1.4 Confidentiality

The User shall respect a strict confidentiality of the eMIP protocol and shall not disclose it to any third person in any form whatsoever, even after the expiry or the termination of the present license.

Moreover, the User shall take all necessary measures ensuring that its employees are bound by the same obligations of confidentiality.

In case a third party asks to access to the eMIP protocol, the User may indicate to him how he may access the eMIP protocol.

1.5 Evolutions

Only GIREVE has the right to modify and make evolutions of the eMIP protocol.

The new versions of the eMIP protocol could be communicated by GIREVE to the User under the terms of the present license or of any new terms defined by GIREVE.

1.6 Disclaimer

The eMIP protocol and its documentation are provided “as is” without any warranty, explicit or implicit, of any kind.

It shall be used under the sole responsibility of the User.

GIREVE could not be held liable for the use made by the User for any reason whatsoever.

1.7 Governing law – juridiction

The present license shall be governed by French law. French law shall apply to both rules and form and the merits, notwithstanding the place of performance of the essential or auxiliary obligations.

IN THE EVENT OF A DISPUTE ARISING OUT OF THE PRESENT LICENCE, IT IS EXPRESSLY AGREED THAT THE COMPETENT COURTS OF PARIS SHALL HAVE EXCLUSIVE JURISDICTION, EVEN IN THE EVENT THERE IS MORE THAN ONE DEFENDANT OR IN CASE OF RECOURSE IN WARRANTY.

2 Introduction

2.1 Aims

The **eMobility Interoperation Protocol**, called **eMIP**, is provided by **GIREVE** (www.gireve.com) as part of his main objective: “open access to vehicle charging stations”. In the context, eMIP targets two goals:

- Enabling roaming of charging services by providing a charge authorisation and a data clearing house API.
- Providing access to a comprehensive charging point database.

This document describes **guidelines to perform a proper link with the GIREVE’s platform using the eMIP protocol**. After an overview of the eMobility Interoperation environment and the GIREVE’s platform, this document will focus on describing technical procedures required to enable the communication with the GIREVE’s platform. Then, it will provide guidelines to implement the applicative eMIP communication. The last sections are dedicated to Charge Point Operator (CPO) and eMobility Service Provider (eMSP) use cases.

2.2 Intended Audience

This document is dedicated to technical teams (system administrators, developers, etc.) of systems connected or to be connected to the GIREVE’s platform through eMIP. Both Charge Point Operator (CPO) and eMobility Service Provider (eMSP) use cases are covered.

Notice that use-cases related to Data Aggregator, described in the eMIP Protocol Description document (cf doc [eMIP_Protocol_Descr]), are not covered by this document.

2.3 Definitions and Abbreviations

Word	Meaning
Charging Connector	A Charging Connector is an interface to deliver electricity (a plug, a socket ….)
Charging Point	A Charging Point, synonym of EVSE, is the energy supply element. An electrical vehicle can be connected to a Charging Point on one of its connectors. A Charging Point may have several Charging Connectors but only one vehicle could be charged at a time on a Charging Point.
Charging Pool	A Charging Pool is a location where we can find charging infrastructure elements. A Charging Pool may contain several Charging Stations.
Charging Station	A Charging Station is a physical element, visible for the users, on which we find Charging Points. The Charging Station is also the “Human-Machine Interface” (HMI) between the charge infrastructure and the end users.
CPO	Charge Point Operator See 3.2.2 Charge Point Operator (CPO)
Data Aggregator	See 3.2.4 Data Aggregator
eMIP	eMobility Interoperation Protocol. The access to GIREVE’s eMobility Services Platform is based on eMIP protocol.
eMI^3	eMobility ITC Interoperability Innovation Group The eMI^3 is an open group of significant actors from the global Electric Vehicles market who joined forces to harmonize the ICT data definitions, formats, interfaces, and exchange mechanisms in order to enable a common language among all ICT platforms for Electric Vehicles.

eMSP	eMobility Services Provider See 3.2.3 eMobility Services Provider (eMSP)
EV	Electric Vehicle
EVCI	Electric Vehicle Charge Infrastructure Or in French: *Infrastucture de Recharge de Véhicule Électrique* (IRVE) According to eMI^{3}definition, the charge infrastructures are supposed to be organized in 4 hierarchical levels: Pool, Station, Point and Connector. • A Charging Pool (“*Zone*” in French) is a location where we can find charging infrastructure elements. The main attributes of a Pool describe “location” information (address, geo-coordinates …) and operators’ information (owner, technical operator …). A Pool may contain several Stations. • A Charging Station (“*Borne*” in French) is a physical element, visible for the users, on which we find Points. The Station is also the “Human-Machine Interface” (HMI) between the charge infrastructure and the end users. Its main attributes are related to HMI (badge reader, languages …). A Station may contain several Points. • A Charging Point, synonym of EVSE, is the energy supply element. One vehicle can be connected to a Point. Its main attributes are related to energy supply (Voltage, AC/DC, mode, maximum power …). A Point may have several Connectors. Only one is active at a time. • A Charging Connector is an interface to deliver electricity.
EVSE	Electric Vehicle Supply Equipment EVSE is a synonym of Charging Point.
GIREVE	*Groupement pour l’Itinérance des Recharges Électriques de Véhicules* Or in English: Grouping to promote Roaming when Recharging Electric Vehicles See 3.2.1 GIREVE
GIREVE’s eMobility Services Platform	The GIREVE’s eMobility Services Platform is said GIREVE’s Platform or Inter-operation Platform. Its acronym is IoP.
HTTP	Hypertext Transfer Protocol HTTP is an application protocol for information systems.
IoP	Inter-operation Platform. IOP is the acronym of the GIREVE’s eMobility Services Platform.
Platform Access Contract	The Platform Access Contract is a B2B contact between GIREVE and an operator (eMSP or CPO). It indicates the technical, legal and financial terms that apply to access to all or part of the GIREVE’s services. Especially, the Communication Partner(s) designation and the CPO-eMSP Service Access Table Mapping.
RPC	*Référentiel des Points de Charge* Or in English: Charge Points Repository, Reference database of charging Points for electric Cars The RPC is a system, built around a database that contains Electric Vehicles Charge Infrastructure (EVCI) description. It is accessible and manageable through eMIP via a web service API.

Service Access Table Mapping	The Service Access Table Mapping is a table, maintained by GIREVE, indicating the contractual relationship between CPOs and eMSPs on terms of GIREVE's Platform services. Typically, for a CPO, this table permits to know which eMSP can have access to Authorisation service via the IoP, and in which conditions (location, service type …).
SOAP	Simple Object Access protocol SOAP is a communication protocol for exchanging structured information as web services in computer networks. It uses the XML format.
Transaction	The word "Transaction" is used for a Client-Server unique exchange, go and back. In eMIP, based on SOAP web services, a "Transaction" represents the request call and its response reception.
W3C	World Wide Web Consortium The W3C is the main international standards organization for the World Wide Web.
WSDL	Web Service Definition Language Technical description of functionality that is offered by a SOAP web service. It uses the XML format.
XML	eXtensible Markup Language XML is a textual data format. This format is used to describe web services as WSDL, and to exchange web services.
XSD	An XML Schema, or XSD, describes the structure of an XML document. The format of XSD document is XML.

3 eMobility Interoperation Environment

3.1 Global Overview

The eMIP protocol defines SOAP web service interfaces between several e-mobility system actors. To simplify the understanding, these actors have been divided into several roles: GIREVE's Platform, Data Aggregator, CPO and eMSP, even if one actor may perform several roles.

All these roles are depicted in the *Figure 1* below and detailed in the paragraph *3.2 Roles and Actors*.

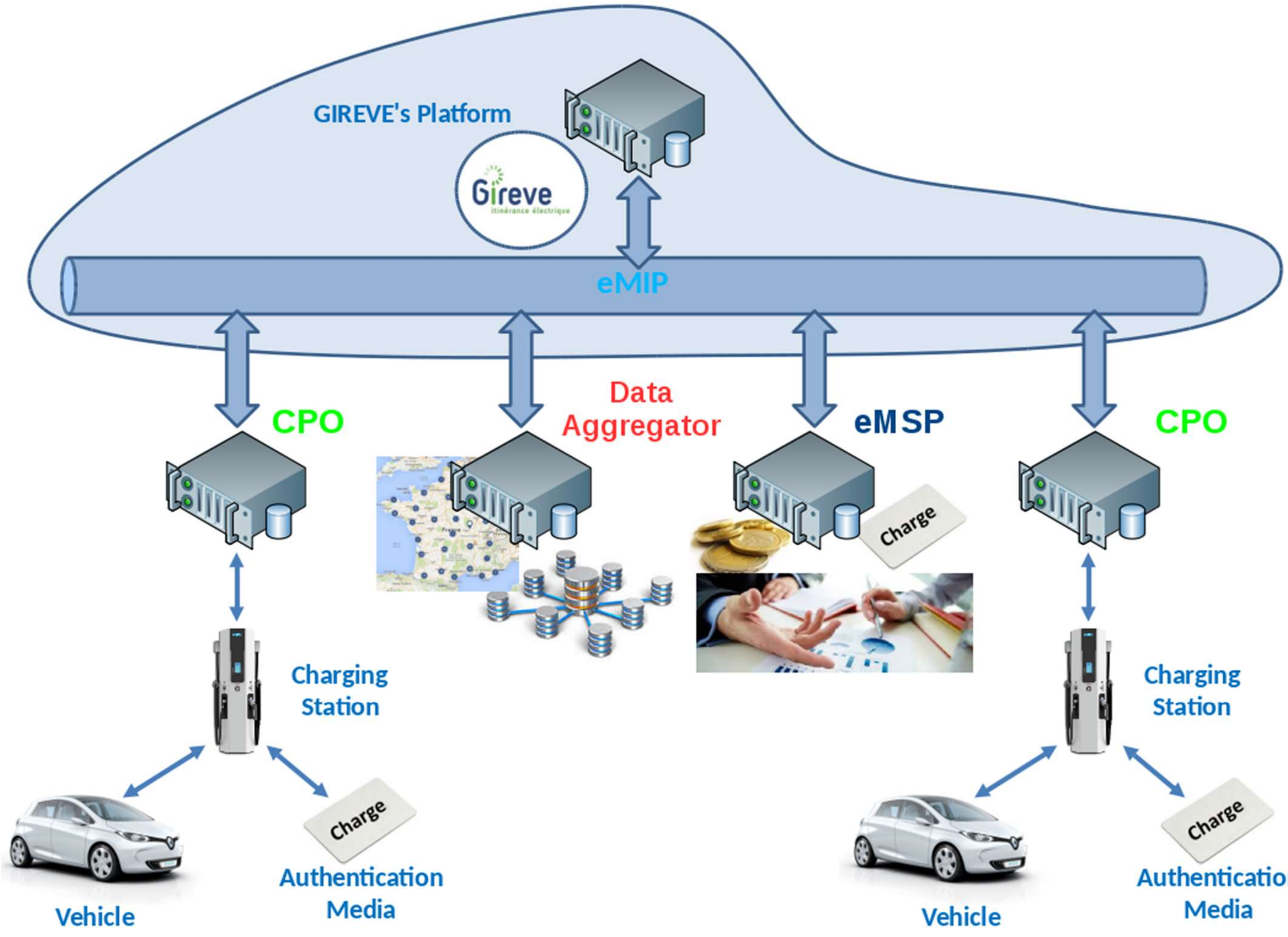


Figure 1: Global Overview of the eMIP Environment

All eMIP interfaces are defined as SOAP 1.2 web services. Therefore, their technical description is represented by WSDL files as explained in the eMIP Protocol Description document (cf doc [eMIP_Protocol_Descr]).

The GIREVE's Platform, as an Inter-operation Platform (IoP), implements a HTTP server providing the eMIP SOAP interfaces for CPOs and eMSPs. CPO and eMSP systems are clients of this server. The related services are prefixed by "eMIP_ToIOP_".

The GIREVE's Platform also implements a HTTP client for the rest of eMIP SOAP interfaces. The server part is implemented, partly or completely following the use case and the assumed role, by the CPO and eMSP partners systems. This HTTP client is used for some use cases in eMIP. The related services are prefixed by "eMIP_FromIOP_".

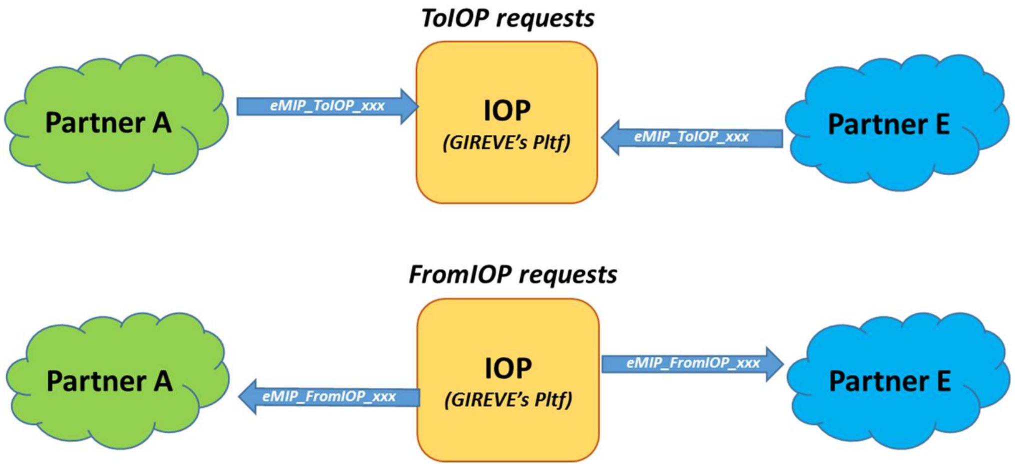


Figure 2: FromIOP and ToIOP requests

For all the information exchanges, as depicted in *Figure 1*, the communication is performed through the GIREVE's Platform using the eMIP protocol. For example, a CPO can verify the authorisation of an end-user by contacting the GIREVE's Platform, which will check its database or contact the relevant eMSP to retrieve the authorisation result. The same behaviour applies for updating or finding information about an EVCI or finding a Charging Point. Hence, even if an actor communicates only with the GIREVE's Platform, it can still interact indirectly with other actors.

3.2 Roles and Actors

3.2.1 GIREVE

In eMIP, GIREVE is an eMobility services "inter-operator". It manages the GIREVE's eMobility Services Platform (named IOP), and the RPC Data Aggregator (named RPC).

The GIREVE's eMobility Services Platform, the Inter-operation Platform (IoP), provides technical and functional means to intermediate services between different actors of the system.

3.2.2 Charge Point Operator (CPO)

A Charge Point Operator (CPO) provides electric vehicle users, who have a contract with an eMobility Services Provider, a charge infrastructure: charging pools, charging stations, charging points, charging connectors and their management system.

A CPO shall have a valid "Platform Access Contract" with GIREVE to be authorised to access to the GIREVE's Platform services.

Notice: CPO is a role in eMIP. An actor may have several roles.

3.2.3 eMobility Services Provider (eMSP)

An eMobility Services Provider (eMSP) provides electric vehicle users with various services useful for the eMobility. This can be EV recharge services, electric vehicle rental, Car-Sharing services, navigation services, etc. An eMSP has a B2C relationship with these "final customer".

An eMSP shall have a valid "Platform Access Contract" with GIREVE to be authorised to access to the GIREVE's Platform services.

To provide services like charge services, charge point search and localization, an eMSP shall be in a contractual B2B relationship with CPOs. The contract can apply on all services of the eMSP or in only a bunch of them.

Notice: eMSP is a role in eMIP. An actor may have several roles.

3.2.4 Data Aggregator

A Data Aggregator is a system that manages EVCI data in a given area and which can be requested by an IoP to get this data.

Example: the RPC provided by GIREVE is a Data Aggregator which covers at least the French territory.

3.2.5 Communication Partner

A Communication Partner is a system actor which is technically connected to the GIREVE's Platform and exchanges messages with it. A Communication Partner is a client in the client/server communication, and may be also a server depending of the use cases.

To be authorised to be connected to the GIREVE's platform, a Communication Partner shall obtain the necessary security credentials from GIREVE (certificates, passwords …) and communicate the IP addresses of its physical machines. Only known IP addresses will be authorised to connect.

A Communication Partner may host one or many Operators (CPO and/or eMSP).

Example: A CPO system or an eMSP system, connected to the GIREVE's Platform, is a Communication Partner.

3.2.6 Operator

An Operator is the system actor which is functionally connected to the GIREVE's Platform and exchanges messages with it. This is a business role and its related company owns a "Platform Access Contract" with GIREVE.

An Operator manages at least one Communication Partner, and eventually several, which is technically connected to the GIREVE's Platform.

The separation between the two roles "Operator" and "Communication Partner" is mainly important when a Communication Partner hosts several Operators. This notion also allows distinguishing between all the Communication Partners of an Operator.

Example: a CPO or an eMSP is an Operator. A CPO system is a Communication Partner of the CPO.

3.2.7 Operators, Communication Partners Relationships

The illustrations below helps understanding the difference between Operator and Communication Partner.

The Communication Partner is defined by its IP-addresses, has a dedicated client Certificate, and hosts several Operators (Operations).

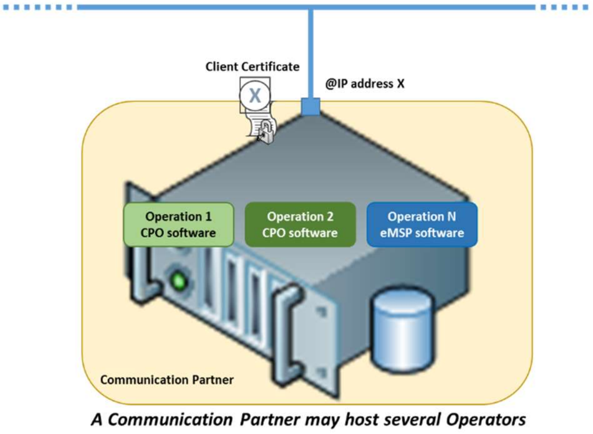


Figure 3: Communication Partner and Operator

An Operator may be hosted by several Communication Partners. In the classical situation where the partner has two servers, a “Normal Server” and a “Backup Server”, it may define two Partners with two different IP addresses … In this situation an Operation will be hosted by two Communication Partners.

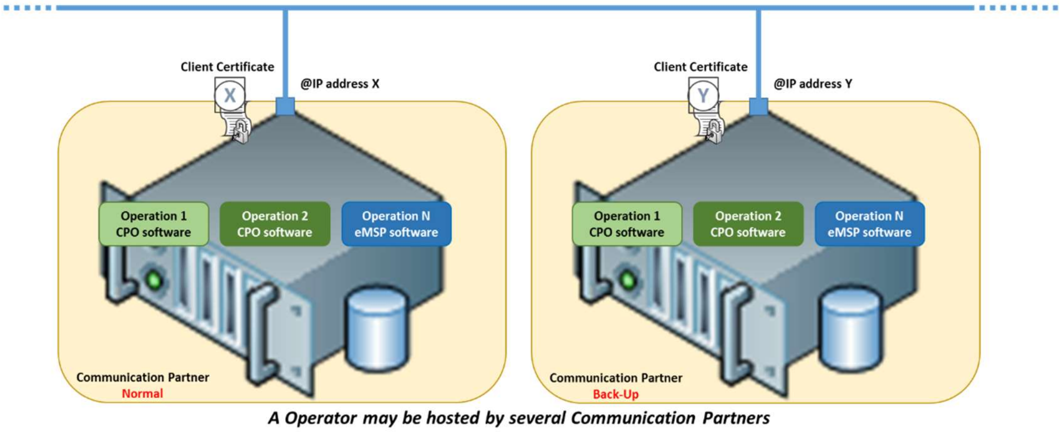


Figure 4: Partners Operators : [n,n] relation

3.3 eMI3 standards and declination

3.3.1 Charging Operator Id

The Operator-Id MUST match the following structure (the notation corresponds to the augmented Backus-Naur Form (ABNF) as defined in RFC5234):

<Operator ID> = <Country Code> <S> <Spot Operator ID>

With

<Country Code> = 2 ALPHA

; Two character country code according to ISO-3166-1 (Alpha-2-Code)

<Spot Operator ID> = 3 (ALPHA / DIGIT)

; Three alphanumeric characters, referring to the Operator

<S> = *1 (“*”)

; optional separator

3.3.2 Charging Pool Id

The Charging Pool Id MUST match the following structure (the notation corresponds to the augmented Backus-Naur Form (ABNF) as defined in RFC5234):

<Charging Pool Id> = <Country Code> <S> <Spot Operator ID> <S> <ID Type> <Pool ID>

With

<Country Code> = 2 ALPHA

; Two character country code according to ISO-3166-1 (Alpha-2-Code)

<Spot Operator ID> = 3 (ALPHA / DIGIT)

; Three alphanumeric characters, referring to the Operator

<ID Type> = “**P**”

; One character “P” indicating that this ID represents a “Pool”

<Pool Instance> = (ALPHA / DIGIT) 1 * 30 (1*(ALPHA / DIGIT) [/ <S>])

; between 1 and 31sequence of alphanumeric characters, including additional optional or separators, start with alphanumeric character, referring to a specific Pool in EVSE Operator data system.

ALPHA = %x41-5A / %x61-7A

; according to RFC 5234 (7-Bit ASCII)

DIGIT = %x30-39

; according to RFC 5234 (7-Bit ASCII)

<S> = *1 (“*”)

; optional separator

An example for a valid EVSE Pool ID is “IT*123*P456*AB789” with “IT” indicating Italy, “123” representing a particular Spot Operator, “P” indicating that this ID represents a “Pool” and “456*AB789” representing one of its Pool.

NOTE: In contrast to the eMA ID, no check digit is specified for the EVSE Pool ID in this document.

Alpha characters SHALL be interpreted case insensitively.

eMI3 strongly recommends that implementations SHOULD ,

- use the separator between country code and spot operator Id
- use the separator between spot operator id and ID type

3.3.3 Charging Station Id

There is no eMI3 standard for identifying Charging Stations. But, in order to keep a unique and relevant way of identifying EVCI elements, IOP will use "eMI3 like" way of identifying Charging Stations.

The Charging Station Id MUST match the following structure (the notation corresponds to the augmented Backus-Naur Form (ABNF) as defined in RFC5234):

<Charging Station Id> = <Country Code> <S> <Spot Operator ID> <S> <ID Type> <Station ID>

With

<Country Code> = 2 ALPHA

; Two character country code according to ISO-3166-1 (Alpha-2-Code)

<Spot Operator ID> = 3 (ALPHA / DIGIT)

; Three alphanumeric characters, referring to the Operator

<ID Type> = "**S**"

; One character "S" indicating that this ID represents a "Station"

<Pool Instance> = (ALPHA / DIGIT) 1 * 30 (1*(ALPHA / DIGIT) [/ <S>])

; between 1 and 31sequence of alphanumeric characters, including additional optional or separators, start with alphanumeric character, referring to a specific Charging Station in EVSE Operator data system.

ALPHA = %x41-5A / %x61-7A

; according to RFC 5234 (7-Bit ASCII)

DIGIT = %x30-39

; according to RFC 5234 (7-Bit ASCII)

<S> = *1 ("*")

; optional separator

An example for a valid EVSE Pool ID is "IT*123*S456*AB789" with "IT" indicating Italy, "123" representing a particular Spot Operator, "S" indicating that this ID represents a "Station" and "456*AB789" representing one of its Charging Station.

NOTE: In contrast to the eMA ID, no check digit is specified for the Charging Station ID in this document.

Alpha characters SHALL be interpreted case insensitively.

We strongly recommend that implementations SHOULD ,

- use the separator between country code and spot operator Id
- use the separator between spot operator id and ID type

3.3.4 Charging Point Id (EVSE-Id)

The EVSE-Id MUST match the following structure (the notation corresponds to the augmented Backus-Naur Form (ABNF) as defined in RFC5234):

<EVSE ID> = <Country Code> <S> <Spot Operator ID> <S> <ID Type> <EVSE ID>

With

<Country Code> = 2 ALPHA

; Two character country code according to ISO-3166-1 (Alpha-2-Code)

<Spot Operator ID> = 3 (ALPHA / DIGIT)

; Three alphanumeric characters, referring to the Operator

<ID Type> = “**E**”

; One character “E” indicating that this ID represents an “EVSE”

<Power Outlet ID> = (ALPHA / DIGIT)1 * *30 (1*(ALPHA / DIGIT) / [<S>])

; between 1 and 31 sequence of alphanumeric characters or separators, including additional optional separators start with alphanumeric character, internal number allowing the EVSE Operator to identify one specific EVSE

ALPHA = %x41-5A / %x61-7A

; according to RFC 5234 (7-Bit ASCII)

DIGIT = %x30-39

; according to RFC 5234 (7-Bit ASCII)

<S> = *1 (“*”)

; optional separator

An example for, a valid EVSE ID is “FR*A23*E45B*78C” with “FR” indicating France, “A23” representing a particular EVSE Operator, “E” indicating that it is of type “EVSE” and “45B*78C” representing the power outlet ID, that is to say one of its EVSEs. NOTE: In contrast to the eMA ID, no check digit is specified for the EVSE ID in this document.

Alpha characters SHALL be interpreted **case insensitively**.

Even though all valid formats of EVSE Id SHALL be readable to ensure compatibility with ISO/IEC 15118, eMI3 strongly recommends that implementations SHOULD ,

- use the separator between country code and Spot Operator ID
- use the separator between Spot Operator ID and ID type

3.3.5 Charging Connector Id

There is no eMI3 standard for identifying Charing Stations. But, in order to keep a unique and relevant way of identifying EVCI elements, IOP will use "eMI3 like" way of identifying Charging Connectors.

The Charging Connector-Id MUST match the following structure (the notation corresponds to the augmented Backus-Naur Form (ABNF) as defined in RFC5234):

<Charging Connector ID> = <Country Code> <S> <Spot Operator ID> <S> <ID Type> <Connector ID>

With

<Country Code> = 2 ALPHA

; Two character country code according to ISO-3166-1 (Alpha-2-Code)

<Spot Operator ID> = 3 (ALPHA / DIGIT)

; Three alphanumeric characters, referring to the Operator

<ID Type> = "**X**"

; One character "X" indicating that this ID represents a "Connector"

<Connector ID> = (ALPHA / DIGIT)1 * *30 (1*(ALPHA / DIGIT) / [<S>])

; between 1 and 31 sequence of alphanumeric characters or separators, including additional optional separators start with alphanumeric character, internal number allowing the EVSE Operator to identify one specific Connector

ALPHA = %x41-5A / %x61-7A

; according to RFC 5234 (7-Bit ASCII)

DIGIT = %x30-39

; according to RFC 5234 (7-Bit ASCII)

<S> = *1 ("*")

; optional separator

An example for, a valid Charging Connector ID is "FR*A23*X45B*78C" with "FR" indicating France, "A23" representing a particular EVSE Operator, "X" indicating that it is of type "Connector" and "45B*78C" representing the Connector ID. NOTE: In contrast to the eMA ID, no check digit is specified for the Charging Connector ID in this document.

Alpha characters SHALL be interpreted **case insensitively**.

We strongly recommend that implementations SHOULD ,

- use the separator between country code and Spot Operator ID
- use the separator between Spot Operator ID and ID type

4 Integration with the GIREVE's Platform

The integration with the GIREVE's Platform involves several tasks and actions, which will be managed as a project named "Connection project". This project, its phases, tasks and milestones, and the way to manage it, are described in the document [Cnx_Project_Mgt]. Please refer to this document.

5 Integration Guidelines

5.1 Identifier Requirements

During the “Identifier Sharing” phase, GIREVE and the Operators define and share the necessary credentials and technical information to make possible the communication session setup between the IoP and the Communication Partner(s).

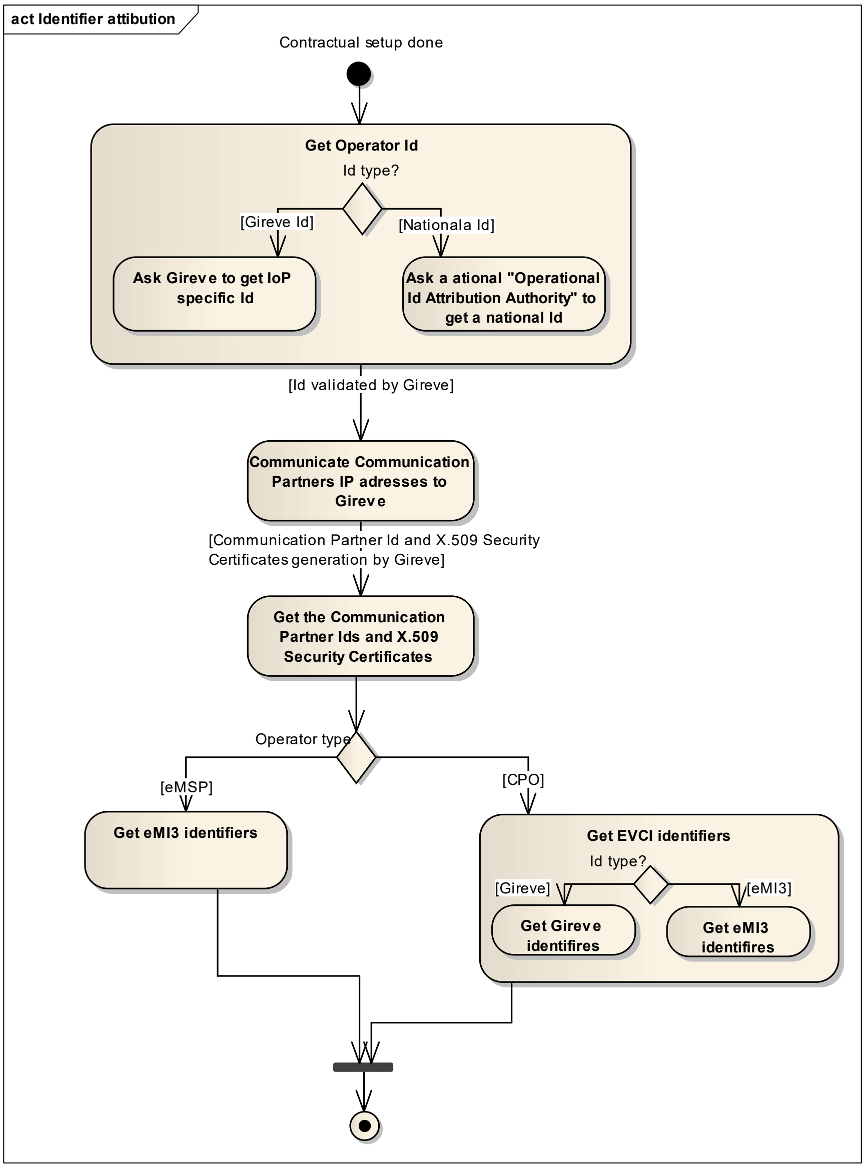

act Identifier attibution
Contractual setup done
Get Operator Id
Id type?
[Gireve Id]
[Nationala Id]
Ask Gireve to get IoP specific Id
Ask a ational "Operational Id Attribution Authority" to get a national Id
[Id validated by Gireve]
Communicate Communication Partners IP adresses to Gireve
[Communication Partner Id and X.509 Security Certificates generation by Gireve]
Get the Communication Partner Ids and X.509 Security Certificates
Operator type
[eMSP]
[CPO]
Get eMI3 identifiers
Get EVCI identifiers
Id type?
[Gireve]
[eMI3]
Get Gireve identifires
Get eMI3 identifires

Figure 5: Operator and Communication Partner(s) identifiers generation and attribution

5.1.1 Definition of Identifiers

Each Operation shall have a unique identifier named “operatorId”.

Each Communication Partner shall have a unique identifier named “partnerId”.

These identifiers are generated or provided by/to the IoP team during the “Identifier Sharing” phase. This involves three steps in the process:

1. First, the Operator may contact a national Operational Id Attribution Authority in order to get a unique Operator Identifier. *In France, this authority is the AFIREV. The Operator might avoid this step and request directly to GIREVE to get an identifier specific to the GIREVE's Platform.*
2. Then, the Operator has to provide to GIREVE its Operator Id and the list of its Communication Partners, with related IP address(s) as depicted in *Figure 6*. *There is typically one IP address per Communication Partner, but more are authorised depending of the deployment configuration (e.g. several physical servers deployed with different IP addresses).*
3. Finally, GIREVE will generate identifiers for the Communication Partners of the Operator, as depicted in *Figure* 7

Parameter	Value
Operator Id	Identifier generated by AFIREV (for France) or by GIREVE
Communication Partner 1	
IP address 1	109.58.46.01
Communication Partner 2	
IP address 1	109.58.46.02
IP address 2	109.58.46.03

Figure 6: Example of Operator's List of Communication Partners

Parameter	Value
Operator Id	Identifier generated by AFIREV (for France) or by GIREVE
Communication Partner 1	
Communication Partner 1 Id	partnerId1 (generated by GIREVE)
IP address 1	109.58.46.01
Communication Partner 2	
Communication Partner 2 Id	partnerId2 (generated by GIREVE)
IP address 1	109.58.46.02
IP address 2	109.58.46.03

Figure 7: Example of Operator's Completed List of Communication Partners

GIREVE provides the document templates to fill-in and sign by the partners. Retrieve these documents by contacting GIREVE.

The Operators shall submit any changes on the identification information as soon as a change happens.

5.1.2 Certificate Issuing

To insure authentication of the communicating systems on the “IoP-Communication Partner” link, all the Communication Partners shall have valid X.509 Security Certificates to be able to setup an eMIP communication session.

These certificates are provided and updated by the IoP technical team. The transmission of certificates, their passwords and updates to the Operators mechanism is to be discussed with the IoP technical team during the “Identifier Sharing” phase.

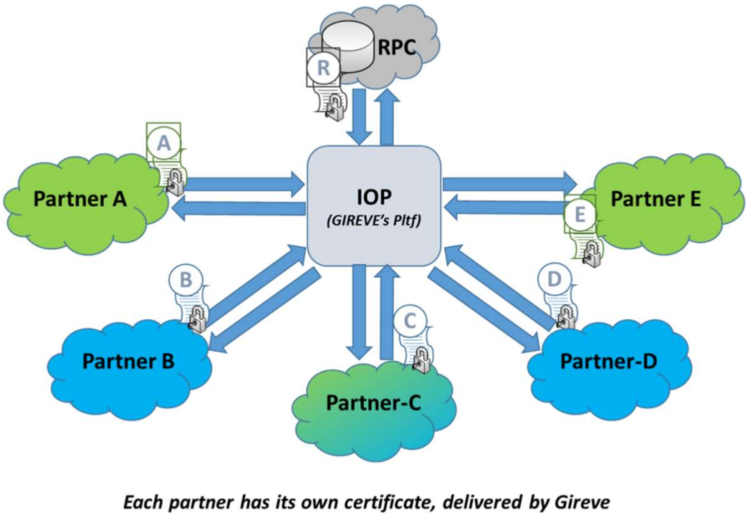


Figure 8: Certificates provided by the IoP technical team

5.1.3 Definition of Identifiers for CPO

In order to perform the first upload in the GIREVE’s RPC of its EVCI description, a CPO shall either send the EVCI description to the GIREVE technical team following a specific template provided by GIREVE; or it can use the eMIP protocol to upload these EVCI description data to the IoP. This can be discussed with GIREVE technical team during the “Technical Setup” phase.

This process implies providing of an identifier to each EVCI element. There are three main identifiers:

- The CPO back-end system identifiers (which is seen by IOP as an **external** identifier).
- The eMI^3 identifiers (following ***eMI^3*** group requirements).
- The **GIREVE** identifiers

GIREVE provides the document template to fill-in with the EVCI elements description. Retrieve this document by contacting GIREVE. The update of this information is described in section 6.

GIREVE promotes the use of eMI3 identifiers

5.1.4 Definition of Identifiers for eMSP

The user authorisation list (called in eMIP “Authentication Data”) must be uploaded by eMSPs to IOP, for two purposes:

- Contractual reasons: The contract between eMSP-Operator and GIREVE is based on the number of authorised users managed by the eMSP. So sharing this user authorisation list is necessary to manage the contract.
- Operational features: Some CPO would be interested by getting this list, for implementing asynchronous authorisation method as the main solution or as a downgraded mode.

In order to perform the first upload or the update of its user authorisation list to the IoP, the eMSP Operator shall use the eMIP protocol.

This process implies providing an identifier to each eMSP end user. There are two possible cases:

- The eMSP Operator may already have eMI^3 identifiers and use them directly,
- The eMSP Operator generates eMI^3 identifiers, following the eMI3 group requirements, and uses these new identifiers.

The update of these identifiers is described in section 7.

5.2 Technical basic Requirements: https, security

This section provides guidelines for the "Technical Setup" phase and the discussion with the IoP technical team.

5.2.1 Technical communication setup

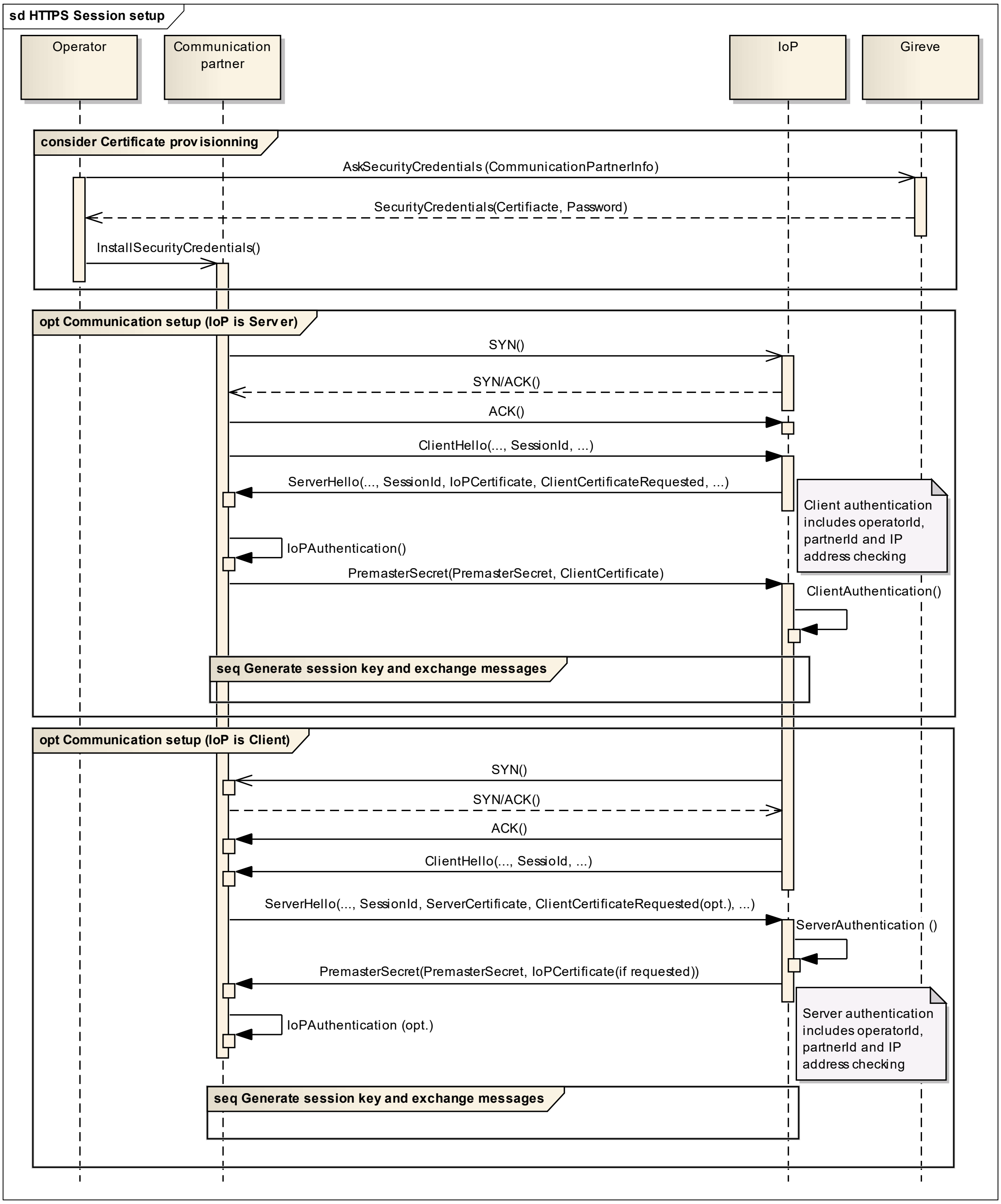

sd HTTPS Session setup
Operator
Communication partner
IoP
Gireve
consider Certificate provisionning
AskSecurityCredentials (CommunicationPartnerInfo)
SecurityCredentials(Certifiacte, Password)
InstallSecurityCredentials()
opt Communication setup (IoP is Server)
SYN()
SYN/ACK()
ACK()
ClientHello(..., SessionId, ...)
ServerHello(..., SessionId, IoPCertificate, ClientCertificateRequested, ...)
Client authentication includes operatorId, partnerId and IP address checking
IoPAuthentication()
PremasterSecret(PremasterSecret, ClientCertificate)
ClientAuthentication()
seq Generate session key and exchange messages
opt Communication setup (IoP is Client)
SYN()
SYN/ACK()
ACK()
ClientHello(..., Sessiold, ...)
ServerHello(..., SessionId, ServerCertificate, ClientCertificateRequested(opt.), ...)
ServerAuthentication ()
PremasterSecret(PremasterSecret, IoPCertificate(if requested))
IoPAuthentication (opt.)
Server authentication includes operatorId, partnerId and IP address checking
seq Generate session key and exchange messages

Figure 9: HTTPs communication session setup

HTTPs Security

All web services shall be exchanged only on a secured HTTPs connection.

A bi-directional authentication, using the Mutual SSL Authentication mode for HTTPs, is even required prior to each communication setup from a Communication Partner to the IoP (services prefixed by “eMIP_ToIOP_”). This authentication is made possible using the certificate exchanged during the “Identifier sharing” phase. This step can be quickly summarized: the Communication Partner can authenticate the IoP using the certificate provided in the HTTPs initialisation, signed by a well-known Trusted Authority; the IoP will authenticate the Communication Partner by requesting its certificate using the HTTPs field “CertificateRequest”.

The Operator is recommended to implement the same bi-directional authentication for communication from the IoP to a Communication Partner (services prefixed by “eMIP_FromIOP_”), but a uni-directional authentication HTTPs connection is also acceptable depending of the constraints of the Partner. This shall be specified with the IoP technical team during the “Technical Setup” phase.

This process is depicted in the figure below:

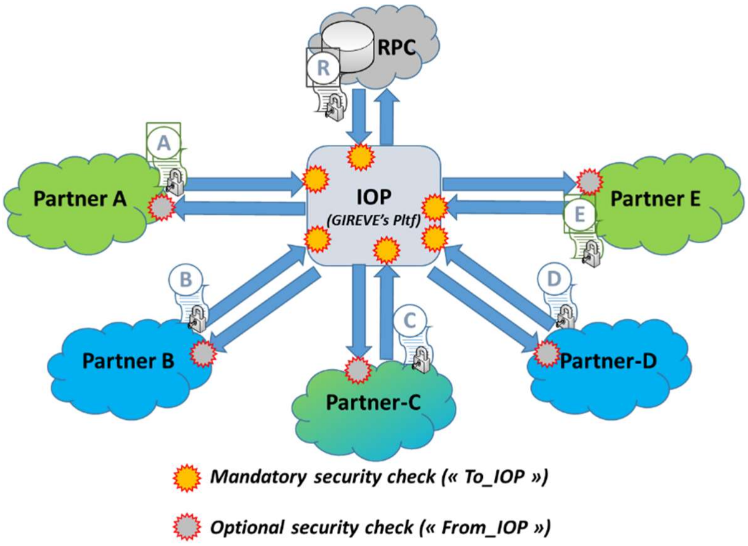

RPC
R
A
Partner A
IOP
(GIREVE's Pltf)
E
Partner E
B
Partner B
C
Partner-C
D
Partner-D
Mandatory security check (« To_IOP »)
Optional security check (« From_IOP »)

Figure 10: Connexion schema over the GIREVE’s IoP

Identifier Verification

The Operator and Communication Partner identifiers provided in the eMIP applicative messages, using the fields “operatorId" and “partnerId”, are authenticated using the certificate and the IP address used for the communication.

Before authorising the setup of a new communication session asked by a Communication Partner, the following points are checked by the IoP:

1. Retrieve the “partnerId” from the request message,
   - Check the availability of the “partnerId” in the IoP database,
   - Check if the corresponding Communication Partner is activated,
2. Check that the IP address used for the communication matches one of the IP addresses registered for this Communication Partner,
3. Check that the “CN” field of the Communication Partner certificate matches the “CertificatePartnerId” registered for this Communication Partner,
4. Retrieve the “operatorId” from the request message,
   - Check the availability of the “operatorId” in the IoP database,
   - Check if the corresponding Operator is activated,

5. Check that the Communication Partner and the Operator are associated in the IoP database.

If a step in this verification process fails, the communication setup is aborted. The IoP may send back an error response.

5.3 Web Services implementation Guidelines

5.3.1 Web Services Usage Requirements

Version 1.0 of eMIP specifies operations between the IoP, the CPOs and the eMSPs. Some operations are defined to be invoked on the IoP whereas others are considered for invocation at the CPOs or eMSPs systems. In order to allow such bidirectional service invocation, three servers are defined: IoP, CPO server and eMSP server.

The eMIP only allows Request-Response Message Exchange Patterns (MEP) between both services. Other MEPs like OneWay, Notification or Solicit-Response are not defined.

All eMIP services are based on Web Services paradigms adopting the SOAP 1.2 message framework. The interface description for these services is based on a WSDL which is provided by GIREVE.

According to the eMIP specification the communication channel should be secured by HTTPs over SSL/TLS (as defined in section 5.1).

All eMIP messages exchanged shall use UTF-8 character encoding.

5.3.2 Manage SOAP and WSDL

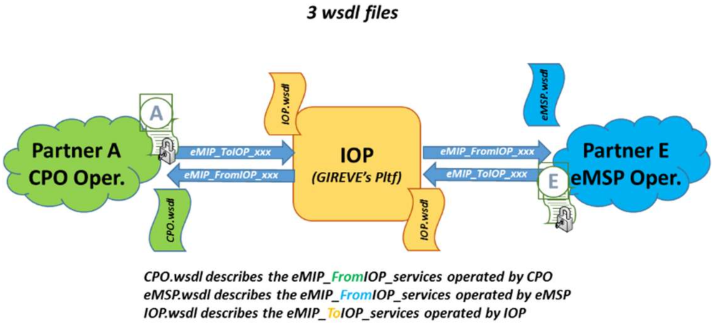
Several tools allow generating code in many languages from WSDL, but also to visualise and test quickly a SOAP communication.

There is no requirement to use such tools, but it is really discourage to manage WSDL manually. Without any guaranty, some examples of tools are listed below:

- SoapUi (www.soapui.org)
    - Navigate through WSDL files
    - Simulate a server or client communication
- Eclipse Web Tools Platform (https://eclipse.org/webtools/ws/)
    - Navigate through WSDL files
    - Manage SOAP communication for Java applications

The WSDL are described in three files:

- IOP.wsdl which describes the WebServices for which IOP is server
- eMSP.wsdl which describes the WebServices for which the eMSP is server
- CPO.wsdl which describes the WebServices for which the CPO is server



CPO.wsdl describes the eMIP_FromIOP_services operated by CPO
eMSP.wsdl describes the eMIP_FromIOP_services operated by eMSP
IOP.wsdl describes the eMIP_ToIOP_services operated by IOP

Figure 11: Structure of the WSDL files

5.3.3 XML Schema Data Types

The W3C organism has defined several generic data types that can be used to describe XML messages. It is possible to define other data types by describing them in a XSD document.

A data type is defined by a name (e.g. string, int, date, time, dateTime, etc.) and a namespace (e.g. xsd, iopfind, etc.). The namespace is always defined in the XML file and point to a link where the definition of each data type name can be found. For W3C generic data types, the link is an URL to the XSD schema defining them, which is "http://www.w3.org/2001/XMLSchema". For a custom data type, the link is the location of the related XSD document.

These data types are used in the WSDL documents that describe the web service messages. Therefore, at the beginning of a WSDL file, all namespaces are defined and point to generic and custom XSD documents.

In the eMIP Protocol Description document (cf doc [eMIP_Protocol_Descr]), W3C generic data types are prefixed with the namespace "xsd". Therefore "xsd:string" means "This value is a string as defined by the W3C".

Custom data types are also used in eMIP as it is generally done for SOAP APIs. All requests and responses are defined as data types, as well as complex data of the communication API. These custom data types are using different namespaces that all starts with "iop". E.g. "iopfind" for data types related to EVSE Searching. The location of XSD documents describing these custom data types are always mentioned in the WSDL file.

5.3.4 SOAP Structure

A SOAP message is divided into a SOAP header and a SOAP body as described by the WSDL files.

For a request, the SOAP Body contains the request data.

For a response, the SOAP Body either contains the response data, or a SOAP Fault element in case of error. This exception mechanism through SOAP Fault element allows notifying the requestor about application level issues.

5.3.5 Application Header

In eMIP, each SOAP body of a request or a response is starting with the same attributes. These attributes can be grouped as an XML custom data type respectively called "Request Application Header" and "Response Application Header". All request data types inherit from "Request Application Header" and all response data types inherit from the "Response Application Header" data type.

5.4 IOP interface and eMIP protocol implementation Guidelines

5.4.1 Transaction Management

A “Transaction” represents a Client-Server unique exchange, go and back. In eMIP, based on SOAP web services, a “Transaction” represents the request call and its response reception. Each Transaction is identified by a “transactionId” available, as a message field, in all requests and responses of eMIP web services.

The concept of “Transaction” is used for traceability purpose: the IoP stores it with all exchange traces, and the Communication Partner shall do the same.

The “transactionId” is either generated by the Communication Partner or by the IoP.

There are two main use cases related to this transaction id:

- The transaction id is computed by the client (within the meaning of client in “client-server communication”),
- The transaction id is computed by the IoP system (when it plays the server role and the client does not generate a transactionId)

The first one is the main use case, always followed by the IoP and that all Communication Partners should support.

The second use case only applies when a Communication Partner can’t generate a transaction id.

Transaction id computed by the client

Before to sends a request, the client (within the meaning of client in “client-server communication”) computes a unique identifier for the transaction. This transaction id shall be unique for him.

The client fills the “transactionId” field of the request with this id and sends it to the server. Once the server receives the request, it shall store the relevant information and link them to this transaction id, and then send back a response with the same transaction id.

When the IoP is the client, it will always proceed as explained above.

When the IoP is the server, it will always send back the “transactionId” field if present.

Transaction id computed by the IoP server

If ever a client didn’t provide a “transactionId” field in a request, the IoP will always generate a unique identifier, store relevant information and link them to this transaction id, and then send back a response with this new transaction id. This client shall then store the relevant transaction data indexing them by the received “transactionId”.

We recommend for a Client, to implement transaction id computed by the client.

Important Notice:

For traceability and log, all the messages exchanged with IoP **shall be** stored by the Communication partner. These messages are indexed by the transaction id.

5.4.2 Response Status Handling

The HTTP and SOAP layers provide information about the status of a transaction:

- First, the HTTP layer notifies about the success or the failure of the communication exchange via a HTTP Status Code.
- Then, the SOAP layer notifies about the success or the failure of the service processing operation.
    - In case of failure, a SOAP Fault exception is returned with a SOAP Fault Code instead of the expected Response element.
    - The SOAP Fault Code indicates the reason of the failure. E.g. *INTERNAL_ERROR → MSG_SUIVI_WS_CANNOT_CREATE_TRANSACTION_EXCEPTION means "IoP did not succeed to create a valid transactionId"*.
    - The list of SOAP Fault Codes is available in the eMIP Protocol Description document (cf doc [eMIP_Protocol_Descr]).

To refine the service processing termination status, a dedicated field, "requestStatus", is included in all eMIP response messages. The "requestStatus" field indicates the applicative success or failure of the requested operation; furthermore, it may provide additional informative details in case of success. E.g. *201 →Ok - Warning: the CPO of the EVSE cannot be identified*.

- A requestsStatus < 10000 is a "success", "information" or "warning" status. It indicates that the treatment has been done, fully or partially.
- A requesteStatus >=10000 is an "error" or "fatal status. It indicates that the treatment has not been done.

The list of "requestStatus" values is available in the eMIP Protocol Description document (cf doc [eMIP_Protocol_Descr]).

5.4.3 Services and data exchange: Partner as a client, IOP as a server

All WebServices solicited by a partner as a client (and to IOP as a server) are named "eMIP_ToIOP_xxx".

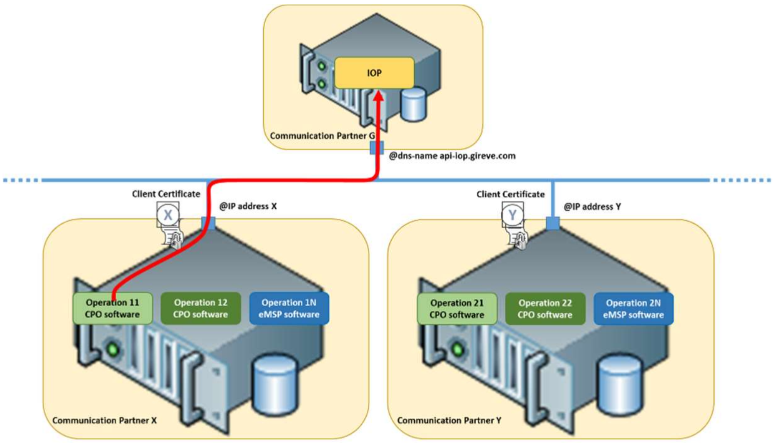

IOP
Communication Partner G
@dns-name api-iop.gireve.com
Client Certificate
X
@IP address X
Client Certificate
Y
@IP address Y
Operation 11 CPO software
Operation 12 CPO software
Operation 1N eMSP software
Communication Partner X
Operation 21 CPO software
Operation 22 CPO software
Operation 2N eMSP software
Communication Partner Y

Figure 12: Partner as a client, IOP as a server

The client will have to install its client Certificate and activate the mutual authentication method

The client will have to describe in the eMIP header,

- The transactionId
- The Id of the Communication Partner which sends the request (partner "X" in the illustration)
- The Id of the "source" Operator (operator "11" in the illustration)

The client will have to fill all the necessary data fields, depending on each WebService.

The client will have to request the WebService on the URL based on the dns-name,

- api-iop.gireve.com to reach the productive environment
- api-pp-iop.gireve.com to reach the pre-prod environment
- api-r7-iop.gireve.com to reach the test environment

5.4.4 Services and data exchange: Partner as a server, IOP as a client

All WebServices solicited by IOP as a client (to a partner as a server) are named “eMIP_FromIOP_xxx”.

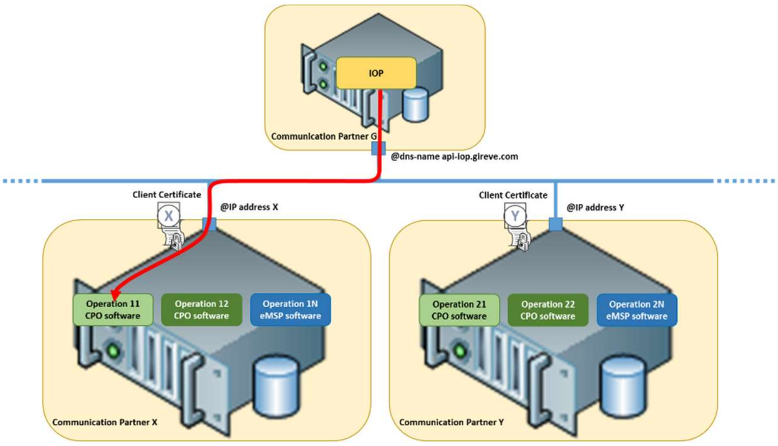

IOP
Communication Partner G
@dns-name api-iop.gireve.com
Client Certificate
@IP address X
X
Client Certificate
@IP address Y
Y
Operation 11 CPO software
Operation 12 CPO software
Operation 1N eMSP software
Operation 21 CPO software
Operation 22 CPO software
Operation 2N eMSP software
Communication Partner X
Communication Partner Y

Figure 13: Partner as a server, IOP as a client

IOP (the client) will describe in the eMIP header,

- The transactionId
- The Id of the Communication Partner (its own parnerId, see partner “G” in the illustration).

On reception of such a request, the partner will **have to route this request to the relevant Operator, based on the field “targetOperatorId”** which contains the Id of this relevant operator: “operator 11” in the illustration.

Please note that all the “FromIOP” Webservices contain the field “targetOperatorId” among their request parameters.

5.4.5 Trace Requirements

The Communication Partner shall store (log) all exchanges for traceability purpose.

Each exchange trace shall contain at least: the “transactionId” of the exchange (it will be required by GIREVE in case of analysis), all the “Session” identifiers for Authorisation and ChargeDetailRecord messages, and the messages timestamps (it will be required by the other Operator in case of analysis).

Each trace shall be kept stored at least one year. After this period, depending of the applicable local, traces may be completely deleted.

5.4.6 Timing Requirements

Heartbeat

As defined in the eMIP Protocol Description document (cf doc [eMIP_Protocol_Descr]), the Communication Partner shall contact the IoP system at least every “Heartbeat Period”, either with the “Heartbeat” request or with any other service request or response.

The default value of the “Heartbeat Period” is set to 300 seconds.

If the IoP does not receive any message from the Communication Partner during a “Heartbeat Period”, this Partner will be declared offline and unreachable.

Timeout and retries

All “FromIoP” web service calls shall be answered within 5 seconds, typically in less than 800ms. Any call from the IoP to a Communication Partner that will not be answered within 5 seconds, will be considered as lost.

A retry method is implemented on IoP side, for messages that may accept differed response. Typically:

- eMIP_FromIOP_SetChargeDetailRecord

Because the IoP may send retry messages to other actors, the Operator should wait 15 seconds without response (instead of 5 seconds) before to discard a message. Any call to the IoP (from a Communication Partner) that will not be answered within 15 seconds, should be considered as lost.

The Operator has to implement a “Store & Forward” mechanism (and, by the way, a retry mechanism) for “ToIoP” web service calls that may accept differed response to avoid loss of data. Typically:

- eMIP_ToIOP_SetxxxAvailability/busyStatus
- eMIP_ToIOP_SetxxxStaticData
- eMIP_ToIOP_SetChargeDetailRecord

5.5 IOP interface and functional implementation Guidelines

5.5.1 Service Session Management

A Service Session represents the execution of a charge service. This Service Session may include, following the scenario, the authorisation to access to the service and the charge session records management.

The Service Session is identified by each system, the identifiers may be different: for the GIREVE’s Platform (“serviceSessionId”), the CPO Operator (“partnerServiceSessionId” or “execPartnerSessionId”) and the eMSP Operator (“partnerServiceSessionId”, “salePartnerSessionId”).

The Service Session identifiers are used for traceability in the service delivery purpose.

The lifecycle of a Service Session can be described as below:

- A Service Session starts by an authorisation message from a CPO (ToIOP_GetServiceAuthorisation) or from an eMSP (ToIOP_SetServiceAuthorisation). The Operator may define its own session identifier and mentions it in the “partnerServiceSessionId” field.
- The IoP defines a unique identifier for this Service Session and sets it in the response field “serviceSessionId”.
- Apart from the authorisation messages, other messages can be exchanged during the session, about Charge Detail Records for example. In this case, the IoP and the Operators remind the same session identifiers, “serviceSessionid” and “partnerServiceSessionid” respectively, for each exchange.
- During the session or at the end of the session (depending of the use case), the CPO establishes a “Charge Detail Record” (CDR) embedding the “serviceSessionId” and optionally its own service session identifier, named “execPartnerSessionId”. The CDRs are sent to the IoP as described in section 6.
- When receiving a CDR, the IoP checks if the eMSP sharing this session has communicated a “partnerServiceSessionId” during the related message exchanges. In this case, the IoP adds the “partnerServiceSessionId” to the related CDR “salePartnerSessionId” field. The CDRs are retrieved by eMSPs as described in section 7.
- The Service Session is closed when the charge roaming service is finished.

The following illustration shows how sessionIds are exchanged between actors during Roaming Authorisation transactions.

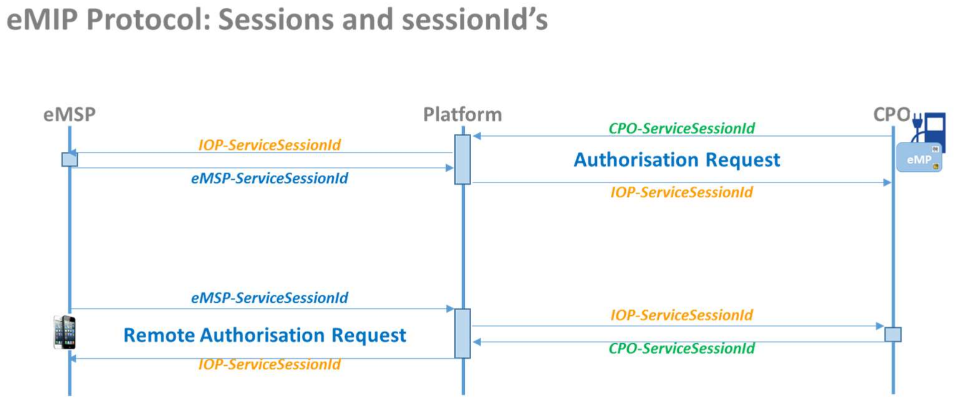


Figure 14: CPO-sesionId, eMSP-sessionId and IOP-sessionId

5.5.2 Meters, and Meter Limits

Meters

In eMIP protocol, we name “Meters”, the data which describes quantitatively (mainly) and qualitatively the delivered service. These meters are feedbacks about the delivered service session. The current classical meters for electro-mobility are “duration” (the duration of service session) and “energy” (the energy quantity, delivered to the vehicle).

The meters are ones of the main inputs for Tariff models and for price calculation (billing).

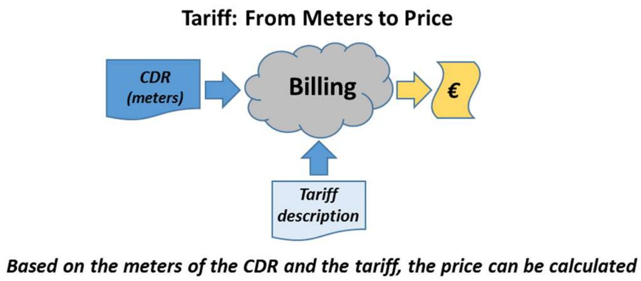

Tariff: From Meters to Price
CDR
(meters)
Billing
€
Tariff
description
Based on the meters of the CDR and the tariff, the price can be calculated

Figure 15: CDR meters are involved in price calculation

The eMSP needs to have some meter values to invoice its customers (B2C).

The CPO needs to have some meter values to invoice the eMSP (B2B).

In both cases, the billing mechanism is based on meter values, even if tariffs and pricing calculation algorithm could be different.

The CPO (its Charging Station and/or its backend-system) is the source of these Meters. It shall send these meters to the eMSP at the end of the charging session (final CDR), and optionally during the charging session (intermediate Charge Detail Record).

Absolute and repetitive meters

An absolute meter is used to manage a quantitative information related to a charge session, and measured **from the beginning of the session**.

A repetitive meter is used to manage a quantitative information related to a charge session, **and measured per step of X minutes**, from the beginning of the session. So, a repetitive meter is a repetition of values, sorted by instance number. By this way, the CDR receiver would be able to rebuild the evolution in time. The typical value foor X is 5 minutes

The following illustration shows an Absolute meter and a Repetitive meter (X is 5 minutes).

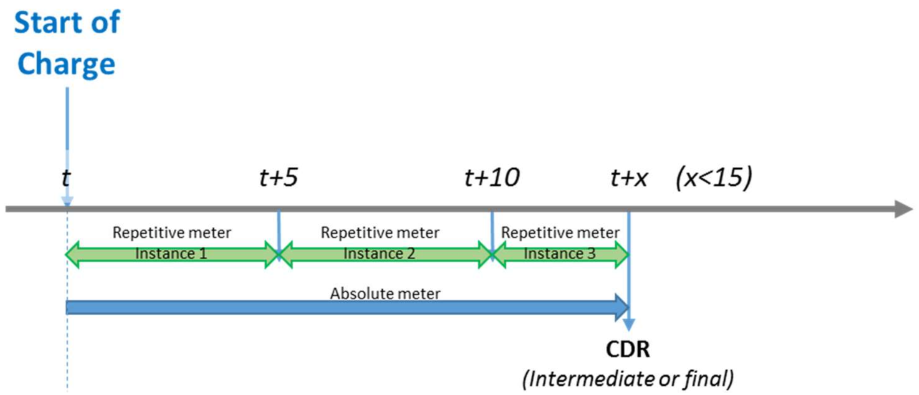

Start of Charge
t
t+5
t+10
t+x (x<15)
Repetitive meter Instance 1
Repetitive meter Instance 2
Repetitive meter Instance 3
Absolute meter
CDR
(Intermediate or final)

Figure 16: Absolute and repetitive meters

eMIP implementation of Meters

In eMIP, a meter data is structured in a set of three fields:

- The meterIdType which describes the kind of meter (“duration”, “energy” for example)
- The meterValue which contains the quantity
- The meterUnit which contains the unit

For example. The following triplet means that the service session had a “duration” of “145” “minutes”.

- 1 (“duration”)
- 145
- Minutes

The eMIP protocol and the GIREVE’s IOP Platform are fully open for new meters (new “meterIdType”). Nevertheless eMSP and CPO shall share the same “Meters catalogue”. The current content of this catalogue, and thus the possible values list for the meterIdType is described in the following table:

MeteridType			
Code	M/O	Signification	Unit
1	M	Total Duration (total duration of the time interval during which the service is available on the ChargingPoint)	"min" (for Minutes)
2	M	Total Energy (total energy for the charge session)	"Wh" (for Watt x hours)
3	O	B2B Service Cost	Local Currency ("EUR" for Euro)
4	O	*reserved for future use*	
5	O	Total veh-charge-duration (total duration of the time interval during which the vehicle consumed electricity)	"min" (for Minutes)
10xx	O	Charging duration at a given level of power ("x" value) Example: 1012 means "Charging duration at 12 kW"	"min" (for Minutes)
21	O	Average power, per step of 1 minutes (Repetitive meter)	"W" (for Watts)
25	O	Average power, per step of 5 minutes (Repetitive meter)	"W" (for Watts)
31	O	Maximum consumed power reached, per step of 1 minutes (Repetitive meter)	"W" (for Watts)
35	O	Maximum consumed power reached, per step of 5 minutes (Repetitive meter)	"W" (for Watts)
41	O	Delivered energy, per step of 1 minutes (Repetitive meter)	"Wh" (for Watt x hours)
45	O	Delivered energy, per step of 5 minutes (Repetitive meter)	"Wh" (for Watt x hours)

Meter value is a string coded decimal (XML). The decimal separator is "."

The meters managed by the CPO will be sent to the relevant eMSP, i.e. the eMSP that manages the user.

Any eMSPs in contract with a given CPO may receive, from this CPO, CDR that contains the meters managed by it. This means that the eMSP may receive different set of meters, coming from different CPOs.

Meter Limits management

This feature is mainly relevant in case of pre-payment B2C contract. In this situation, the eMSP would want to limit the service delivery to what the customer is able to pay, i.e. the amount of its own pre-paid account. We name this amount the “Price Limit”.

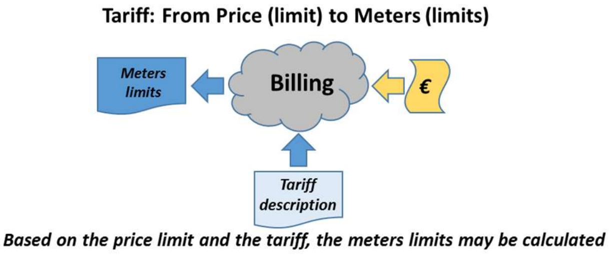

Tariff: From Price (limit) to Meters (limits)
Meters limits
Billing
€
Tariff description
Based on the price limit and the tariff, the meters limits may be calculated

Figure 17: Price limit gives Meters limits

The main principles about Meters limits are:

- The CPO doesn’t know the eMSP B2C tariff, and thus is not able to manage a “Price Limit”
- The eMSP is the B2C tariff owner and thus is able to convert the “Price Limit” to “Meters Limits”
- The CPO has to manage these “Meters Limits” and stop the service when at least one of these limits is reached.

Thus, the eMSP has to define and calculate these “Meters Limits”. Nevertheless, the billing calculation should be non-bijective: it should not work vice versa, as well. In some situations, it could be impossible to calculate a single and exact “Meters Limits” set, based on a “Price Limit”. Two solutions for dealing with this problem:

- Define an approximated “Meters Limits” set and use the Intermediate CDR to follow up the amount and send a Stop request to the CPO when reaching the Price Limit”
- Define a “severe” “Meters Limits” that ensures the final price will not be greater than the “Price Limit”, even if it doesn’t engage the full available money.

5.5.3 RequestedServiceId

IoP is able to manage a large diversity of end-user-services: Standard Charge, Rapid Charge …

The authorisation process is activated for a given service. This service is identified by the data filed "requestedServiceId".

The current value list is:

RequestedServiceId	
Code	Signification
0	reserved value
1	Generic Charge Service

The default value to be used is "1", "Generic Charge Service".

The value "0" is reserved and shall never be used by a CPO.

5.5.4 Message length limitation

For technical issues, some ToIOP_ Services are limited in size. The following table lists the related services and the concerned parameter with the limitation.

ToIOP_ message	Parameter	limitation
eMIP_ToIOP_SearchEVSERequest	maxCount	200 (maximum of EVSEDescrip the IoP can retrieve per request)
eMIP_ToIOP_SetAuthenticationData		The number of authenticationData entries is limited to 500

5.5.5 Privacy Policy Enforcement

The Communication Partner shall respect data retention time requirements according to the local regulation.

5.5.6 Integration – Main points of attention

- **We recommend to implement transaction id computed by the client.**
- **Logs must refer to transactionId**
- **After each call of an eMIP WebService, the requestor shall test the value of "requestStatus". Any value greater than or equal to 10.000 shall be considered as an error.**
- **On reception of an "*eMIP_FromIOP_xxx"* request, the partner will have to internally route this request to the relevant Operator, based on the field "*targetOperatorId*".**
- **Partner shall implement periodical HeartBeat messages.**

6 CPO Specific Implementation Guidelines

6.1 Typical CPO project steps for eMIP implementation and deployment

The integration with the GIREVE's Platform involves several tasks and actions, which will be managed as a project named "Connection project". This project, its phases, tasks and milestones, and the way to manage it, are described in the document [Cnx_Project_Mgt]. Please refer to this document.

The following steps for eMIP implementation and system deployment are not comprehensive but may constitute a guiding reference:

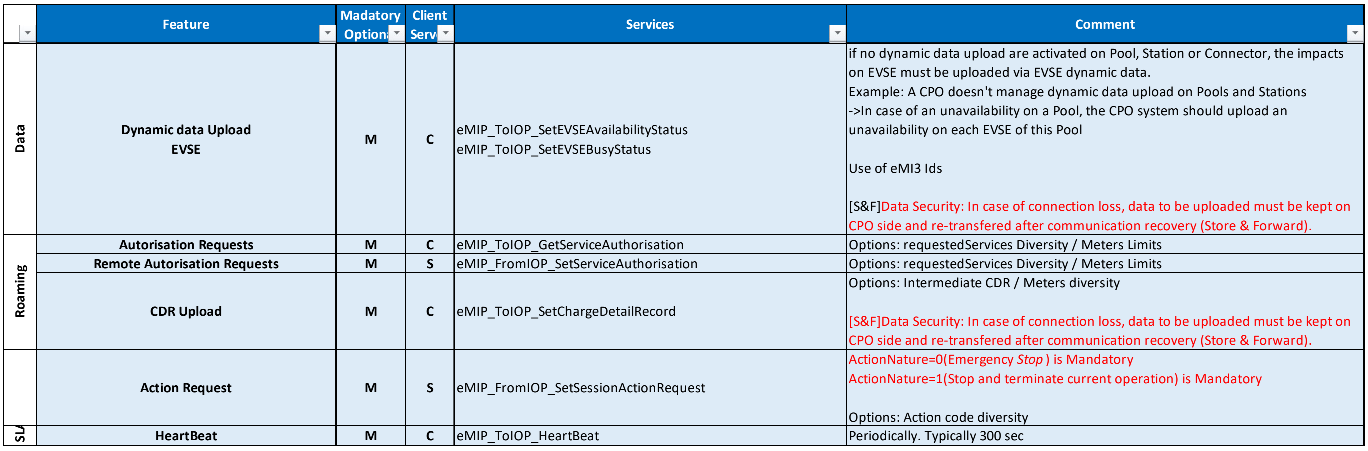
- Contact GIREVE to establish a contractual relationship
- Get and study the eMIP documentation (specification and this document) from GIREVE: if needed, GIREVE's technical team will answer any query.
- Get the eMIP IOP and CPO WSDL files from GIREVE: if needed, GIREVE's technical team can assist the CPO in the understanding and use of eMIP WSDL
- Contact GIREVE to get Operator/Partner credentials, Ids and communicate the eMI3 Ids if already available
- Contact GIREVE to communicate the CPO contractual relation-ship with the eMSPs: GIREVE can help the CPO in the process of the eMSP contractual relation-ship setup (linking, procedures …)
- Check if the CPO's EVCI is already referenced by GIREVE. If it's the case:
    - Check if all the EVCI is covered
    - Communicate additional CPO Ids and information to GIREVE
    - Exchange the eMI3 Ids if available

Actions to be done regarding the CPO back-end system:

- Define (if not available) mechanisms of change detection of the EVCI status :
    - Availability status for Pools, Stations, Points (EVSE) and Connectors
    - Occupation status for Points (EVSE)
- Implement the communication technical aspects: HeartBeat, Logs, Error management …
- Implement the "ToIOP_xxx" and "FromIOP_xxx" web services
- Test of the implementation before effective connection: to avoid disturbance on the operating IOP, each operator must go through a testing phase before connecting to the platform. This test phase is set up in accordance with GIREVE. It includes:
    - Schedule
    - Test Plan, test cases, anomaly management, dataset
    - Cross-configuration setting for the test environments
    - Perform the tests, detect anomalies, process and reiterate
- Get the conformity label from GIREVE to go on Deployment process
- Launch the deployment process: effective connection of the CPO system to the IOP. A particular supervision process is than operated by the GIREVE's technical team to insure that the CPO connected system is stable.

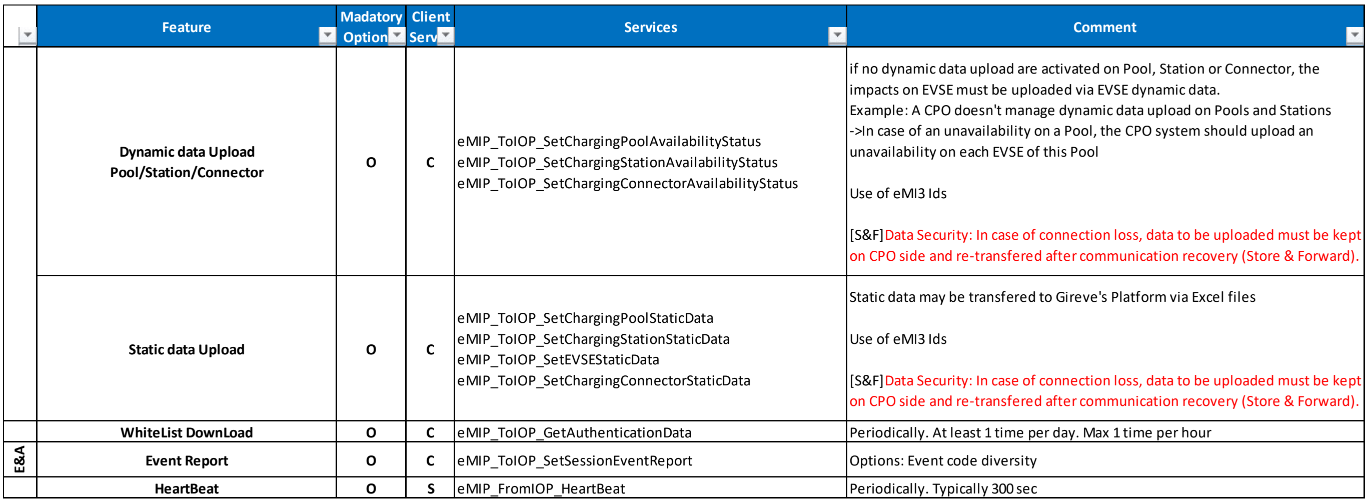
The following two paragraphs give an overview on the Web services to implement by a CPO.

6.1.1 Summary of mandatory WebServices implementation



	Feature	Madatory Option	Client Serv	Services	Comment
Data	Dynamic data Upload EVSE	M	C	eMIP_ToIOP_SetEVSEAvailabilityStatus eMIP_ToIOP_SetEVSEBusyStatus	if no dynamic data upload are activated on Pool, Station or Connector, the impacts on EVSE must be uploaded via EVSE dynamic data. Example: A CPO doesn't manage dynamic data upload on Pools and Stations ->In case of an unavailability on a Pool, the CPO system should upload an unavailability on each EVSE of this Pool Use of eMI3 Ids [S&F]Data Security: In case of connection loss, data to be uploaded must be kept on CPO side and re-transfered after communication recovery (Store & Forward).
Roaming	Autorisation Requests	M	C	eMIP_ToIOP_GetServiceAuthorisation	Options: requestedServices Diversity / Meters Limits
	Remote Autorisation Requests	M	S	eMIP_FromIOP_SetServiceAuthorisation	Options: requestedServices Diversity / Meters Limits
	CDR Upload	M	C	eMIP_ToIOP_SetChargeDetailRecord	Options: Intermediate CDR / Meters diversity [S&F]Data Security: In case of connection loss, data to be uploaded must be kept on CPO side and re-transfered after communication recovery (Store & Forward).
	Action Request	M	S	eMIP_FromIOP_SetSessionActionRequest	ActionNature=0(Emergency *Stop*) is Mandatory ActionNature=1(Stop and terminate current operation) is Mandatory Options: Action code diversity
SL	HeartBeat	M	C	eMIP_ToIOP_HeartBeat	Periodically. Typically 300 sec

6.1.2 Summary of optional WebServices implementation



	Feature	Madatory Option	Client Serv	Services	Comment
	Dynamic data Upload Pool/Station/Connector	O	C	eMIP_ToIOP_SetChargingPoolAvailabilityStatus eMIP_ToIOP_SetChargingStationAvailabilityStatus eMIP_ToIOP_SetChargingConnectorAvailabilityStatus	if no dynamic data upload are activated on Pool, Station or Connector, the impacts on EVSE must be uploaded via EVSE dynamic data. Example: A CPO doesn't manage dynamic data upload on Pools and Stations ->In case of an unavailability on a Pool, the CPO system should upload an unavailability on each EVSE of this Pool Use of eMI3 Ids [S&F]Data Security: In case of connection loss, data to be uploaded must be kept on CPO side and re-transfered after communication recovery (Store & Forward).
	Static data Upload	O	C	eMIP_ToIOP_SetChargingPoolStaticData eMIP_ToIOP_SetChargingStationStaticData eMIP_ToIOP_SetEVSEStaticData eMIP_ToIOP_SetChargingConnectorStaticData	Static data may be transfered to Gireve's Platform via Excel files Use of eMI3 Ids [S&F]Data Security: In case of connection loss, data to be uploaded must be kept on CPO side and re-transfered after communication recovery (Store & Forward).
E&A	WhiteList DownLoad	O	C	eMIP_ToIOP_GetAuthenticationData	Periodically. At least 1 time per day. Max 1 time per hour
	Event Report	O	C	eMIP_ToIOP_SetSessionEventReport	Options: Event code diversity
	HeartBeat	O	S	eMIP_FromIOP_HeartBeat	Periodically. Typically 300 sec

6.2 Data Upload

The global flow diagram is illustrated below.

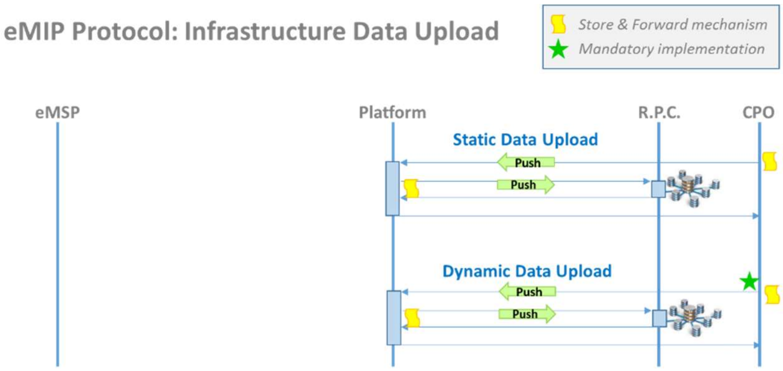


Figure 18: Data Upload flows

6.2.1 EVCI element Identifiers

Operators must identify Pool, EVSE, Station or Connector using eMI3 Id.

6.2.2 Update of EVCI dynamic information

The CPO shall submit any changes on its EVCI dynamic descriptions (status …) as soon as possible using the eMIP protocol. "Dynamic upload" is described in the eMIP Description Protocol document (cf doc [eMIP_Protocol_Descr]).

A typical approach is to notify the update, to the GIREVE's Platform using the eMIP protocol, once it is detected. The more real-time it happens, the more accurate it will be for the end-user.

Uploading EVSE status (availability and busy) is mandatory.

Uploading ChargingPool, ChargingStation status is optional, and gives a more detailed and relevant information. **If no dynamic data upload is activated on Pool, Station or Connector, the impacts on EVSE must be uploaded via EVSE dynamic data.** Example: In case of an unavailability on a Pool, the CPO system should upload an unavailability on each EVSE of this Pool.

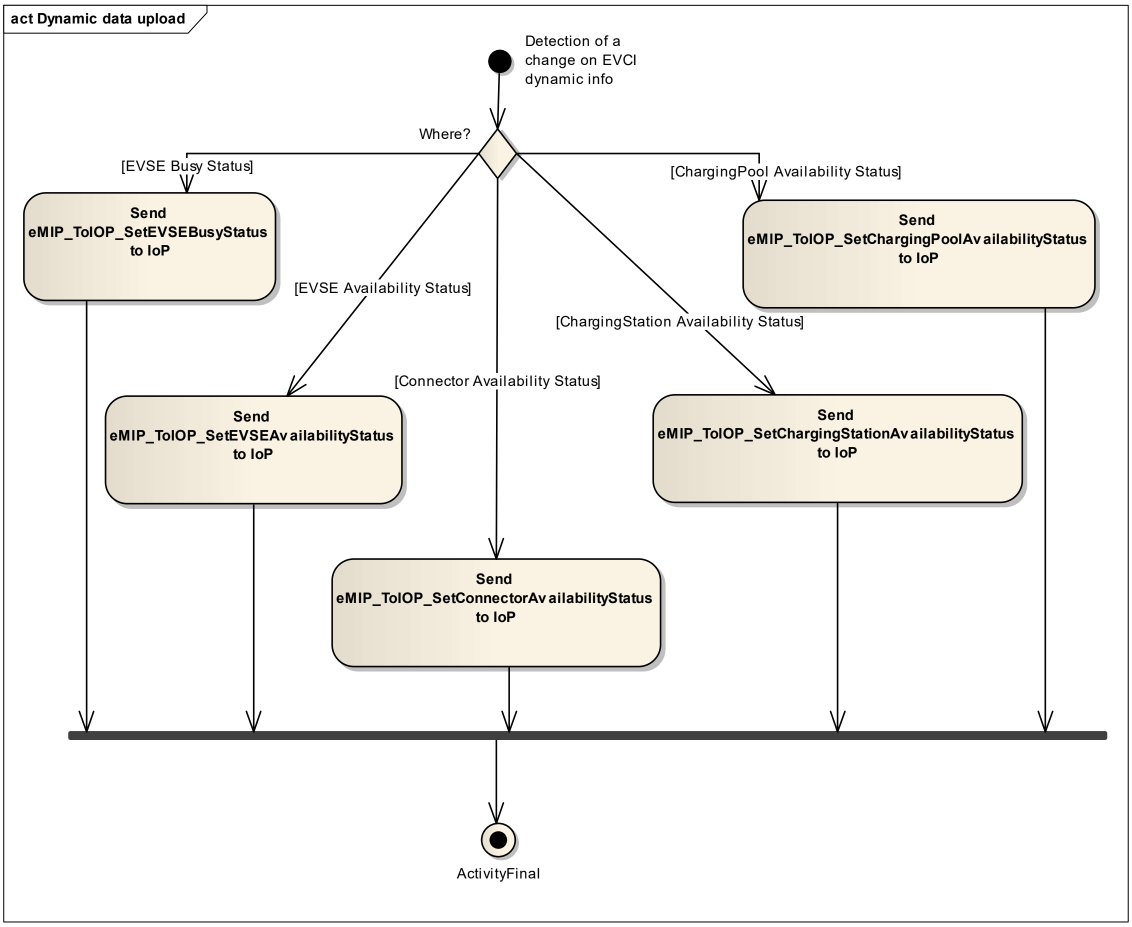

act Dynamic data upload
Detection of a change on EVCI dynamic info
Where?
[EVSE Busy Status]
[ChargingPool Availability Status]
Send eMIP_ToIOP_SetEVSEBusyStatus to IoP
Send eMIP_ToIOP_SetChargingPoolAvailabilityStatus to IoP
[EVSE Availability Status]
[ChargingStation Availability Status]
[Connector Availability Status]
Send eMIP_ToIOP_SetEVSEAvailabilityStatus to IoP
Send eMIP_ToIOP_SetChargingStationAvailabilityStatus to IoP
Send eMIP_ToIOP_SetConnectorAvailabilityStatus to IoP
ActivityFinal

Figure 19: Dynamic data upload

6.2.3 Update of EVCI static information

Please note**, that the EVCI Static-Data-Upload (SDU) is not currently active.* *These changes and updates must be notified to GIREVE's team (Google-SpreadSheets process, evse.report mail …).* ***The following chapter describes, for information, the SDU services that will be activated in a next step.

To maintain a high service level, the CPO Operator shall submit any changes on its EVCI static descriptions (identification, location …) as soon as possible using the eMIP protocol. "Static upload" is described in the eMIP Description Protocol document (cf doc [eMIP_Protocol_Descr]).

A typical approach is to upload the update to the GIREVE's Platform using the eMIP protocol once information has changed. Real-time is not expected here. For example, if a new WiFi free hotspot is installed on a Charging Pool, it is of course not expected to add this information in the IoP once the WiFi router is powering on. The update should happen as soon as possible at the end of the WiFi installation.

The CPO Operator may use the "Future" value of the "AvailabilityStatus" field to advert of a new EVCI before its installation. This may be a good opportunity for the CPO Operator to inform the eMSPs and end-users of the near availability of a new EVCI without enabling it yet.

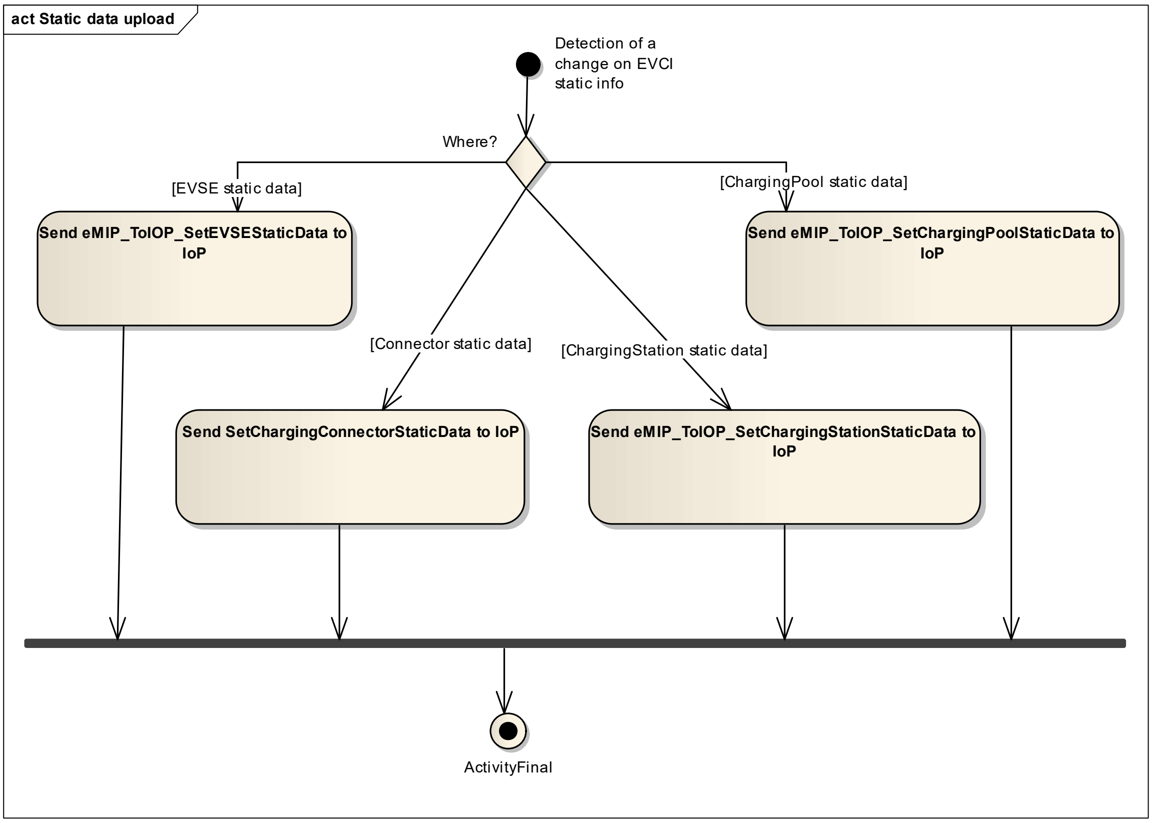

act Static data upload
Detection of a change on EVCI static info
Where?
[EVSE static data]
[ChargingPool static data]
Send eMIP_ToIOP_SetEVSEStaticData to IoP
Send eMIP_ToIOP_SetChargingPoolStaticData to IoP
[Connector static data]
[ChargingStation static data]
Send SetChargingConnectorStaticData to IoP
Send eMIP_ToIOP_SetChargingStationStaticData to IoP
ActivityFinal

Figure 20: Static data upload

6.2.4 Data Integrity for "Data Upload" services

A Store & Forward mechanism must be implemented to ensure that no data upload may be lost, in case of connection loss. Any data upload request that doesn't receive any response from IOP (TimeOut) must be stored on CPO side, and a retry process must be active. After the connection recovery, the SDU messages must be resend in a FIFO manner.

A "re-init" feature that would send all the availability and busy statuses for each EVSE (and optionally, Pool, Station and Connector) would be appreciated to improve the data quality and the problem management.

About Static data upload (SDU)

Please note*, *that the EVCI Static-Data-Upload (SDU) is not currently active. *These changes and updates must be notified to GIREVE's team (Google-SpreadSheets process, evse.report mail …).* ***The following chapter describes, for information, the SDU services that will be activated in a next step.***

When an Operator applies a large update of EVCI information (especially static data upload), incoherence of data between EVCI may appear during a small amount of time until all EVCI data are updated. For example, if a CPO would like to advert of a new Charging Station in an existing Charging Pool, the Charging Pool may still indicate the old number of Charging Stations even if the new Charging Station entry has already been added.

The data integrity issue is not very critical and is under the CPO responsibility.

Best practices to reduce this issue impact could be,

- In case of data updates, upload static description updates using the eMIP Protocol beginning with the Charging Connectors, then continuing with Charging Points, Charging Stations and finally Charging Pools.
- In case of creation, upload static description creations beginning with the Charging Pools, then Charging Stations, Charging Points and finally Charging Connectors.

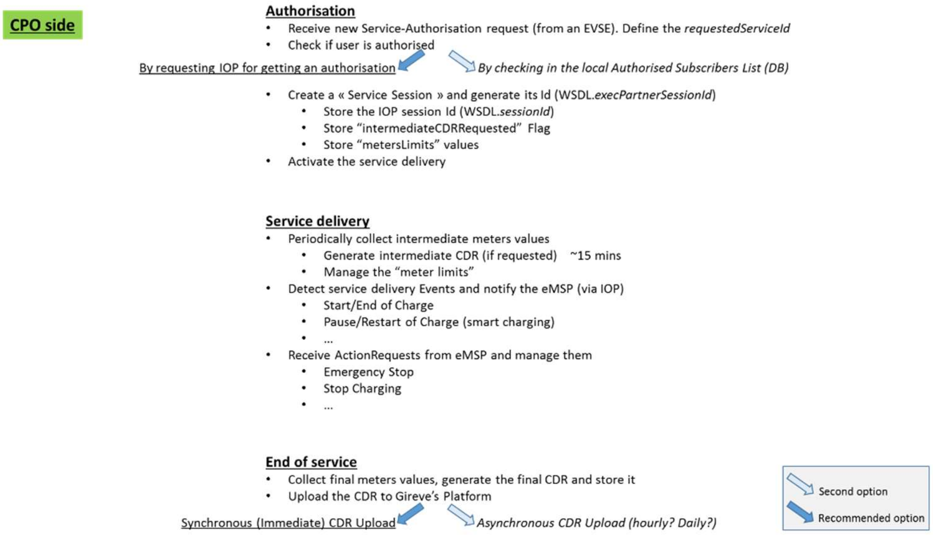
6.2.5 Data Upload – Main points of attention

- **Dynamic data upload is mandatory for Charging Points (= EVSE).**
- **If no dynamic data upload is activated on Pool, Station or Connector, the impacts on EVSE must be uploaded via EVSE dynamic data.**
- **CPO must identify Pool, EVSE, Station or Connector using eMI3 Id**
- **A Store & Forward mechanism must be implemented to ensure that no data upload may be lost, in case of connection loss.**
- **A “re-init” feature that would send all the availability and busy statuses for each EVSE (and optionally, Pool, Station and Connector) would be appreciated to improve the data quality and the problem management.**

6.3 Roaming

The roaming on CPO side means accept users that are customers of an eMSP with which the CPO is in contract (Roaming contract).

The main use-case starts when such a user swipes its badge on an EVSE managed by the CPO. The illustration below, describes the main steps and actions to be done by CPO.



CPO side

Authorisation

- Receive new Service-Authorisation request (from an EVSE). Define the *requestedServiceId*
- Check if user is authorised

By requesting IOP for getting an authorisation

By checking in the local Authorised Subscribers List (DB)

- Create a « Service Session » and generate its Id (WSDL.*execPartnerSessionId*)
  - Store the IOP session Id (WSDL.*sessionId*)
  - Store "intermediateCDRRequested" Flag
  - Store "metersLimits" values
- Activate the service delivery

Service delivery

- Periodically collect intermediate meters values
  - Generate intermediate CDR (if requested) ~15 mins
  - Manage the "meter limits"
- Detect service delivery Events and notify the eMSP (via IOP)
  - Start/End of Charge
  - Pause/Restart of Charge (smart charging)
  - …
- Receive ActionRequests from eMSP and manage them
  - Emergency Stop
  - Stop Charging
  - …

End of service

- Collect final meters values, generate the final CDR and store it
- Upload the CDR to Gireve's Platform

Synchronous (Immediate) CDR Upload

Asynchronous CDR Upload (hourly? Daily?)

Second option

Recommended option

Figure 21: Roaming seen from CPO side

6.4 Roaming - Authorisation

6.4.1 Authorisation processes

As described in the eMIP Protocol Description document (cf doc [eMIP_Protocol_Descr]), a CPO Operator has two possibilities to manage end-user authorisations, depending on the agreement (roaming contract) it shares with the eMSP: it can either wait the arrival of an end-user to check its authorisation with the IoP using "ToIOP_GetServiceAuthorisation" (synchronous use case), or it can retrieve regularly an end-users authorisations list from the IoP using "ToIOP_GetAuthenticationData" (asynchronous use case).

The global flow diagram is illustrated below.

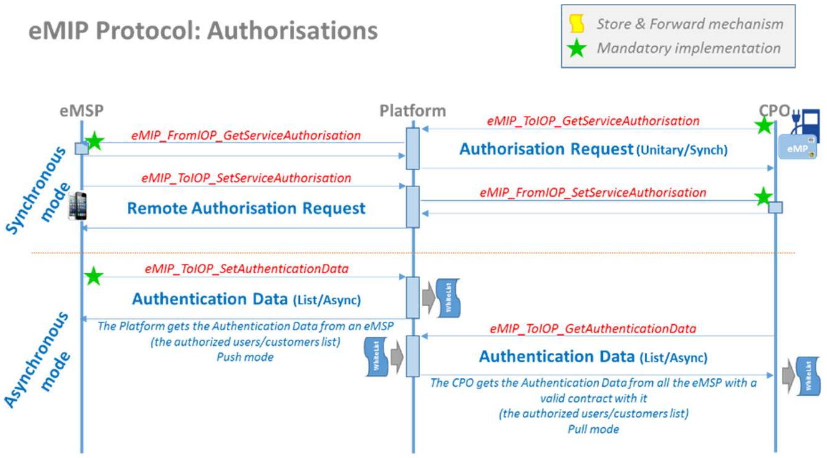

eMIP Protocol: Authorisations
Store & Forward mechanism
Mandatory implementation
eMSP
Platform
CPO
eMP
Synchronous mode
eMIP_FromIOP_GetServiceAuthorisation
eMIP_ToIOP_GetServiceAuthorisation
Authorisation Request (Unitary/Synch)
eMIP_ToIOP_SetServiceAuthorisation
eMIP_FromIOP_SetServiceAuthorisation
Remote Authorisation Request
Asynchronous mode
eMIP_ToIOP_SetAuthenticationData
Authentication Data (List/Async)
The Platform gets the Authentication Data from an eMSP (the authorized users/customers list) Push mode
WhiteList
eMIP_ToIOP_GetAuthenticationData
Authentication Data (List/Async)
The CPO gets the Authentication Data from all the eMSP with a valid contract with it (the authorized users/customers list) Pull mode
WhiteList

Figure 22: CPO-Roaming flows

GetServiceAuthorisation – Synchronous use case

In the synchronous use case, the CPO Operator will request (synchronously with the user badge swiping) IOP, to get (or not) an explicit authorisation. Thus, it will know for sure the authorisation status of the current end-user. However, a small latency might happen depending of the connection quality. In a nominal use case, this latency will be typically 2 seconds, and less than 10 seconds.

The synchronous mode,

- Ensures the CPO to have a real and explicit authorisation from the eMSP.
- Allows the eMSP to have its own specific authorisation decision algorithm: pre-paid method, dynamic tariff…
- Gives to the eMSP a real-time means to manage the relationship with its customer

GetAuthenticationData – Asynchronous use case

In the asynchronous use case, the CPO Operator shall store a local subscriber authorisation list and take care of its validity (manage white and black lists). The CPO will authorise (or not) the service for the user, by checking this list and applying its own algorithm. The main advantage is to reduce latency and eventually to simplify the process and connectivity requirements on EVSE.

The asynchronous mode,

- Put the authorisation decision act on the CPO side: the eMSP delegates the authorisation process to the CPO
- Permit semi-online EVCI management, regarding the authorisation process, by downloading the authentication data to the charging stations (relevant for small authorisation list).

We recommend, to implement the synchronous mode.

6.4.2 Authorisation – Asynchronous mode implementation

Expiry date

When using ToIOP_GetAuthenticationData, the CPO Operator will receive a set of modifications on the authorisation list: new authorised subscribers, modifications of the already registered subscribers and removed subscribers.

Regarding validity, there are two possible cases:

- **The authentication data element contains an expiry date**: the eMSP may provide an expiry date in the field "expiryDate" of the authorisation. In this case, the CPO Operator can safely authorise the end-user for the given service until this expiry date without updating its local list. **The expiry date is the responsibility of the eMSP.**
- **The authentication data element does not contain an expiry date**: the CPO shall take care of frequently updating this information to be sure the authorisation still applied, however it may authorise an end-user which does not have a valid contract with an eMSP anymore. **It is the responsibility of the CPO Operator to update its local list.** A typical approach is to update the local list hourly, but never less than daily.

Service Session

At the authorisation time, when a user request a service, the CPO backend system shall define a CPO-serviceSessionId to identify this service session. This Id will be mandatory to upload the CDR at the end of the service session.

See Ch. 5.5.1 *Service Session Management* page 30.

RequestedServiceId

See Ch.5.5.3 *RequestedServiceId* page 34.

6.4.3 Authorisation – Synchronous mode implementation

User identification

When a user authenticates on the Charging Station, the CPO backend system has to get its user identifier. The authorisation request (eMIP_ToIOP_GetServiceAuthorisation) shall content this user identifier, in the userId mandatory field. Because the user may be identified by several different ways, the userId may content different values. This is the reason why, this data field is associated with a userIdType data field that describes the nature of identifier. The current possible values list for the userIdType is described in the following table:

UserIdType		UserId
Code	**Signification**	**Format**
RFID-UID	RFID Tag Id	Characters String Must be provided (write): uppercase with leading zeros (up to 8 ou 14 char) Must be interpreted (read): non case sensitive. Leading zeros optional (up to 8 ou 14 char)
eMI3	eMI3 Token Id	See eMI3 ("Business objects" doc)
eMA	eMAid (see eMI3)	See eMI3 ("Business objects" doc)
EVCO	EVCOId (see 15118)	See ISO 15118
EMP-SPEC	eMSP specific Id	N/A

The current typical situation for identification is swiping a MIFARE badge. In this case,

- The relevant userIdType in such a situation is “RFID-UID”
- The relevant userId in such a situation is a characters string that shall contain the hexadecimal representation of the 4 or 7 bytes RFID UID (sector 0). Please note that the 7 bytes UID is preferred for interoperability reason. As an example: “1A2B3C4D5E6F70” shall be interpreted as
  - Lowest address byte contains “1A” and “1” is the most significant nibble (half byte) and “A” is the least significant nibble
  - Highest address byte contains “70”
  - Equivalent decimal value is “7365887390543728”

  The userId for RfID-UID is not case sensitive. We recommend to use uppercase characters

  Leading zeros must be provided to reach 8 characters in case of 4 bytes UID or 14 characters in case of 7 bytes UID

EVSE identification

The CPO should identify the EVSE with the eMI3 EVSE Id:

- EVSEIdType must be “eMI3”
- EVSEId must be the eMI3 EVSE Id (like “FR*123*Eabcdef”)

Service Session

At the authorisation time, when a user request a service, the CPO backend system shall define a CPO-serviceSessionId to identify this service session. This Id will be sent to IOP at authorisation time, and will be used during data and services exchanges with IOP and/or with the eMSP via IOP.

IOP will send back to the CPO backend system its own serviceSessionId, and the CPO backend system shall store it and associate it with its CPO-serviceSessionId. By this way the CPO will ensure the traceability about Service session with the IOP side.

See Ch. 5.5.1 *Service Session Management* page 30.

RequestedServiceId

See Ch.5.5.3 *RequestedServiceId* page 34.

The use of the value “0” for requestedServiceId is forbidden on CPO side: This value is reserved!

MeterLimits

MeterLimits are received by CPO when receiving the eMSP response.

See Ch. 5.5.2 *Meters, and Meter Limits* page 31*.*

The CPO will have to manage these “Meters Limits” and stop the service when at least one of these limits is reached.

authorisationValue

The authorisationValue is returned by eMSP to the CPO. The current value list is:

Authorisation	
Code	Signification
1	OK: Service is authorised
2	KO:Service is not authorised

The value “1” means that the service is authorised. It means that the service has to be delivered.

Other values stops the delivery, and rejects the user demand.

6.4.4 Authorisation – Remote Synchronous mode implementation

A CPO Operator may also receive a remote synchronous authorisation request from the IoP by receiving an “eMIP_FromIOP_SetServiceAuthorisation” request. It implies that an authorised end-user has directly contacted its eMSP to use a specific EVSE. The CPO Operator shall therefore authorise this end-user for this EVSE.

To enhance the end-user usability, the CPO Operator shall take care displaying the related eMI[3] identifier on the Charging Station, under each EVSE; it may even use a QrCode containing this identifier.

Service Session

At the authorisation time, when the eMSP request the CPO (using eMIP_FromIOP_SetServiceAuthorisation), the CPO backend system shall define a CPO-serviceSessionId to identify this service session. This Id will be sent to IOP, and will be used during data and services exchanges with IOP and/or with the eMSP via IOP.

IOP sends to the CPO backend system its own serviceSessionId, and the CPO backend system shall store it and associate it with its CPO-serviceSessionId. By this way the CPO will ensure the traceability about Service session with the IOP side.

See Ch. 5.5.1 *Service Session Management* page 30.

RequestedServiceId

See Ch.5.5.3 *RequestedServiceId* page 34.

The default value to be used is “1”, “Generic Charge Service”.

The value “0” is reserved and shall never be used by a CPO.

Meter Limits

See Ch. 5.5.2 *Meters, and Meter Limits* page 31*.*

The CPO will have to manage these “Meters Limits” and stop the service when at least one of these limits is reached.

See Ch. 5.5.2 *Meters, and Meter Limits* page 31*.*

authorisationValue

The authorisationValue is sent by eMSP to the CPO. The current value list is:

Authorisation	
Code	Signification
1	OK: Service is authorised
2	KO:Service is not authorised

The value “1” means that the service is authorised. It means that the service has to be delivered.

Other values stops the delivery, and rejects the user demand.

Note: As seen in Ch. 5.4.4 *Services and data exchange: Partner as a server, IOP as a client* page 29, on reception of a “FromIOP” request, the partner will have to route this request to the relevant Operator, based on the field “targetOperatorId” which contains the Id of this relevant operator.

6.5 Roaming - Charge Detail Record

6.5.1 Charge Detail Record Management

A Charge Detail Record (CDR) describes the result of a charging session. It is generated by the CPO and sent to the eMSP through IOP using the eMIP protocol.

Charge Detail Record Content

The CPO have to provide, at least two “meters” indicated in specified units.

- Duration (meterType = 1) unit=”min” for minutes

and

- Energy (meterType = 2) unit=”Wh” for Watt x hours

These two meters are mandatory. These meter units are mandatory.

The CPO may provide other meters. The following table indicates the meters and meter-units. M/O stands for M=Mandatory and O=Optional.

MeteridType			
Code	**M/O**	**Signification**	**Unit**
1	M	Total Duration (total duration of the time interval during which the service is available on the ChargingPoint)	"min" (for Minutes)
2	M	Total Energy (total energy for the charge session)	"Wh" (for Watt x hours)
3	O	B2B Service Cost	Local Currency ("EUR" for Euro)
4	O	*reserved for future use*	
5	O	Total veh-charge-duration (total duration of the time interval during which the vehicle consumed electricity)	"min" (for Minutes)
10xx	O	Charging duration at a given level of power ("x" value) Example: 1012 means "Charging duration at 12 kW"	"min" (for Minutes)
21	O	Average power, per step of 1 minutes (Repetitive meter)	"W" (for Watts)
25	O	Average power, per step of 5 minutes (Repetitive meter)	"W" (for Watts)
31	O	Maximum consumed power reached, per step of 1 minutes (Repetitive meter)	"W" (for Watts)
35	O	Maximum consumed power reached, per step of 5 minutes (Repetitive meter)	"W" (for Watts)
41	O	Delivered energy, per step of 1 minutes (Repetitive meter)	"Wh" (for Watt x hours)
45	O	Delivered energy, per step of 5 minutes (Repetitive meter)	"Wh" (for Watt x hours)

Meter value is a string coded decimal (XML). The decimal separator is "."

See, ch. 5.5.2 *Meters, and Meter Limits* page 31

Intermediate and Final Charge Detail Record

The CDR could be “final” or “intermediate”.

- The “intermediate” CDR will be uploaded periodically during the charge session, and will describe what happened from the beginning of the session and until the CDR creation.
- The “final” CDR will be uploaded at the end of the charge session, and will describe what happened from the beginning of the session and until the CDR creation which is the end of charge session.

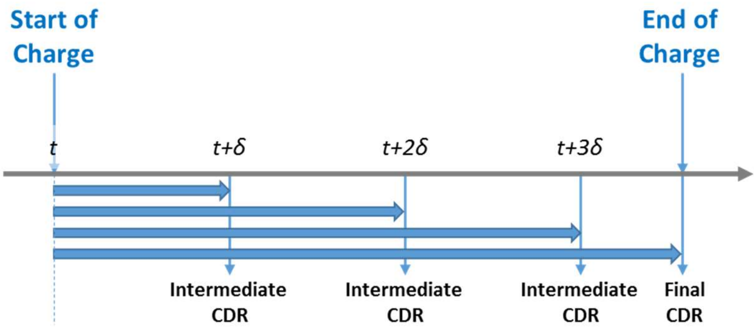


Figure 23: CDR: Intermediate and Final

Please note that the period for generating and uploading intermediate CDR (δ in the illustration), has to be defined by the CPO and is typically 5 – 10 minutes.

Charge Detail Record Upload

At the end of a service session, the CPO shall send a “ToIOP_SetChargeDetailRecord” message containing a Charge Detail Record to the IoP. The associated Charge Detail Record includes the meters value reflecting the level of the service usage (energy consumption, parking duration…).

If this has been requested during the authorisation phase with the field “IntermediateCDRRequested”, the CPO Operator shall send a CDR regularly with a “CDRNature” field set to “Intermediate” during the charging session. The typical value for intermediateCDR sending period is 15 minutes.

We recommend for a CPO, to implement intermediateCDR.

The CPO may postpone the upload of the “Final” CDR and deliver them in a batch mode, periodically. Typically: at the end of the day. This mode is called “Asynchronous mode”. It is not recommended.

We recommend for a CPO, to implement synchronous CDR upload.

Note: In this process, GIREVE only applies as a technical clearing house connector. **The financial compensation has to be handled between CPOs and eMSPs.**

Charge Detail Record Storage

The CPO Operator shall store all versions of the Charge Detail Records, the eventual “Intermediate” ones and especially the “Final” one.

Each Charge Detail Record shall be kept stored at least one year once the financial compensation for this service has been received by the CPO Operator. After this period, depending of the applicable local regulation, all data may be completely deleted.

6.6 Roaming – End to end messaging

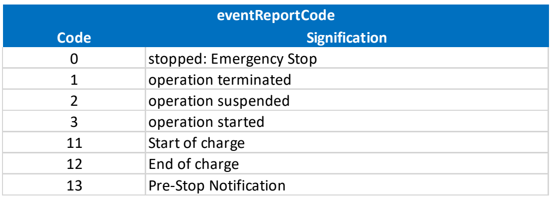
6.6.1 Event report

A CPO may send to the eMSP a message to notify the progress of the service delivery, or to notify a special event. These messages are associated to a serviceSession. The “eMIP_ToIOP_SetSessionEventReport” WebService (CPO as a client) implements this feature.

An Event-Report is a message that contains an Event description (see table below) with a field that describes the Nature of the Event (which event happened?). This field is named sessionEventNature.

sessionEventType				
	sessionEventNature	*Enum Integer*	*M*	Nature of the Event to be reported
	sessionEventId	*String (eventId)*	*M(F), O(T) **	Unique Id of the Event
	sessionEventDateTime	*DateTime*	*M*	Event DateTime
	sessionEventParameter	*String*	*O*	Event Parameter
	relatedSessionActionId	*String (actionId)*	*O*	If the Event is related to an action, contains the Id of the related Session
			** M(F), O(T)*	Means **M**andatory in "FromIOP" services, **O**ptional in "ToIOP" services

The eMIP protocol and the GIREVE’s IOP Platform are fully open for new event nature (new values for sessionEventNature). Nevertheless eMSP and CPO shall share the same “sessionEventNature values” catalog. The current content of this catalogue, and thus the possible values list is described in the following table:



eventReportCode	
Code	Signification
0	stopped: Emergency Stop
1	operation terminated
2	operation suspended
3	operation started
11	Start of charge
12	End of charge
13	Pre-Stop Notification

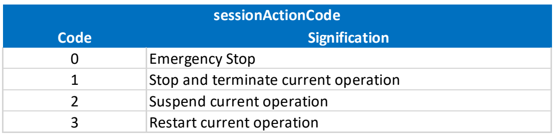
6.6.2 Action request

An eMSP may send to the CPO a message to request a specific action. These messages are associated to a serviceSession. The “eMIP_FromIOP_SetSessionActionRequest” WebService (CPO as a server) implements this feature.

An Action-Request is a message that contains an Action description (see table below) with a field that describes the Nature of the Action (what is required by the eMSP, to the CPO?). This field is named sessionActionNature.

sessionActionType				
	sessionActionNature	*Enum Integer*	*M*	Nature of the Action to be requested
	sessionActionId	*String (actionId)*	*M(F), O(T) **	Unique id of the Action
	sessionActionDateTime	*DateTime*	*M*	Action DateTime
	sessionActionParameter	*String*	*O*	Action Parameter
	relatedSessionEventId	*String (eventId)*	*O*	If the Event is related to an action, contains the Id of the related Session
			** M(F), O(T)*	Means **M**andatory in "FromIOP" services, **O**ptional in "ToIOP" services

The eMIP protocol and the GIREVE’s IOP Platform are fully open for new action nature (new values for sessionActionNature). Nevertheless eMSP and CPO shall share the same “sessionActionNature values” catalog. The current content of this catalogue, and thus the possible values list is described in the following table:



sessionActionCode	
Code	Signification
0	Emergency Stop
1	Stop and terminate current operation
2	Suspend current operation
3	Restart current operation

In case of a service session started by an eMSP via the “remote authorisation” mode, the right way, for the eMSP to request the end of such a service session is to send to the CPO an action request with the actionNature value set to “1: Stop and terminate current operation”.

The CPO shall implement this feature.

6.7 Roaming - Exchanges synthesis

6.7.1 Summary of the main flows

The following illustration shows the exchanges in a service session started via a “classic authorisation process”:

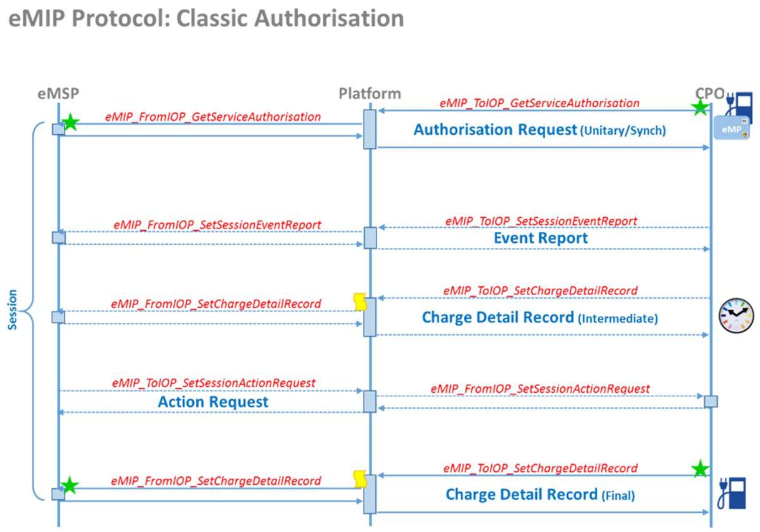

eMIP Protocol: Classic Authorisation
eMSP
Platform
CPO
eMIP_ToIOP_GetServiceAuthorisation
eMIP_FromIOP_GetServiceAuthorisation
Authorisation Request (Unitary/Synch)
eMIP
eMIP_FromIOP_SetSessionEventReport
eMIP_ToIOP_SetSessionEventReport
Event Report
Session
eMIP_FromIOP_SetChargeDetailRecord
eMIP_ToIOP_SetChargeDetailRecord
Charge Detail Record (Intermediate)
eMIP_ToIOP_SetSessionActionRequest
Action Request
eMIP_FromIOP_SetSessionActionRequest
eMIP_FromIOP_SetChargeDetailRecord
eMIP_ToIOP_SetChargeDetailRecord
Charge Detail Record (Final)

Figure 24: CPO-Roaming, flows summary (classic authorisation)

The following illustration shows the exchanges in a service session started via a “remote authorisation process”:

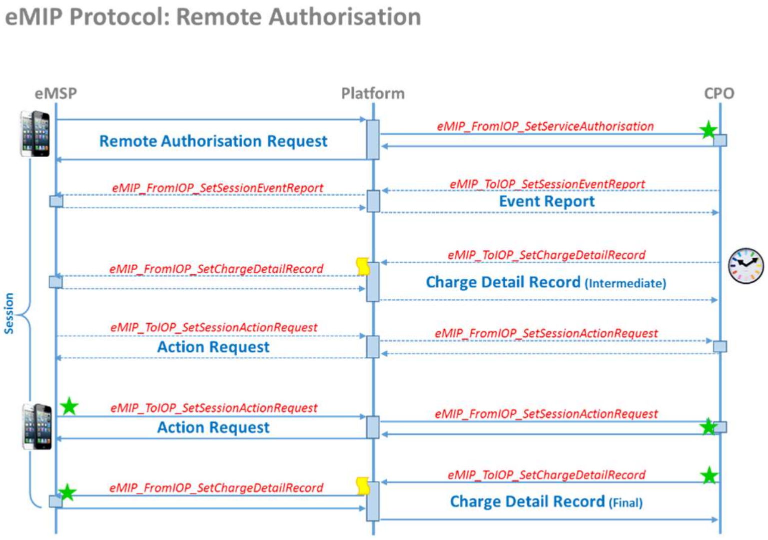

eMIP Protocol: Remote Authorisation
eMSP
Platform
CPO
Remote Authorisation Request
eMIP_FromIOP_SetServiceAuthorisation
eMIP_FromIOP_SetSessionEventReport
eMIP_ToIOP_SetSessionEventReport
Event Report
eMIP_FromIOP_SetChargeDetailRecord
eMIP_ToIOP_SetChargeDetailRecord
Charge Detail Record (Intermediate)
Session
eMIP_ToIOP_SetSessionActionRequest
Action Request
eMIP_FromIOP_SetSessionActionRequest
eMIP_ToIOP_SetSessionActionRequest
Action Request
eMIP_FromIOP_SetSessionActionRequest
eMIP_FromIOP_SetChargeDetailRecord
eMIP_ToIOP_SetChargeDetailRecord
Charge Detail Record (Final)

Figure 25: CPO-Roaming, flows summary (remote authorisation)

6.7.2 Main points of attention

- **For better user experience, we recommend for CPOs to implement the synchronous mode, for authorisation and CDR.**
- **In case of authentication via MIFARE badge, the relevant userIdType is “RFID-UID” and the relevant userId is a characters string that contains the hexadecimal representation of the 4 or 7 bytes RFID UID (sector 0) uppercase with leading zeros.**
- **CPO must identify EVSE using eMI3 Id**
- **CPO backend system shall define a CPO-serviceSessionId do identify each service session, and associate the serviceSessionId sent by IOP.**
- **The default value to be used for requestedServiceId is “1”, “Generic Charge Service”.**
- **We recommend for a CPO, to implement intermediateCDR. The typical period is 15 mins.**
- **The CPO shall include in each CDR, at least the meterTypes “Duration” and “Energy”.**
- **The meter unit for “Duration” shall be “min” for minutes.**
- **The meter unit for “Energy” shall be “Wh” for Watt x hours.**
- **The CPO shall manage the “Meters Limits” received from eMSP via IOP, and stop the service when at least one of these limits is reached.**
- **The CPO shall implement the “*eMIP_FromIOP_SetSessionActionRequest*”, and at least manage the “*sessionActionNature*” values “*0:Emergency Stop*” and “*1: Stop and terminate current operation*”**
- **On reception of an “*eMIP_FromIOP_SetServiceAuthorisation*” or “*eMIP_FromIOP_SetSessionActionRequest*” request, the partner will have to internally route this request to the relevant CPO Operator, based on the field “*targetOperatorId*”.**

7 eMSP Specific Implementation Guidelines

7.1 Typical eMSP project steps for eMIP implementation and deployment

The integration with the GIREVE's Platform involves several tasks and actions, which will be managed as a project named "Connection project". This project, its phases, tasks and milestones, and the way to manage it, are described in the document "Gireve_Tech_Cnx_ConnectionProjectManagement_x.y.z_en.pdf". Please refer to this document.

The following steps for eMIP implementation and system deployment are not comprehensive but may constitute a guiding reference:

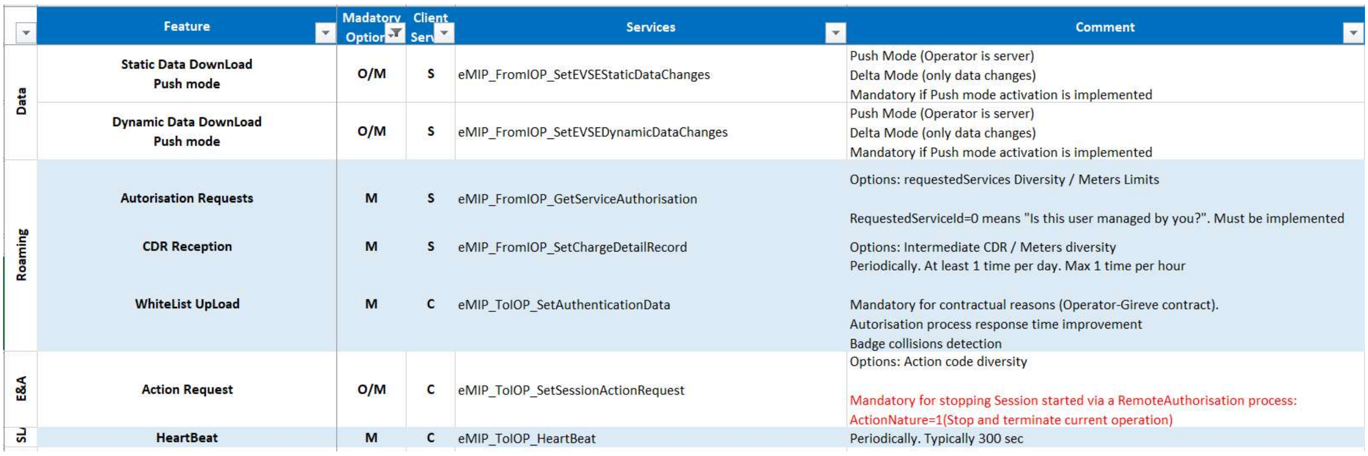
- Contact GIREVE to establish a contractual relationship
- Get and study the eMIP documentation (specification and this document) from GIREVE: if needed, GIREVE's technical team will answer any query.
- Get the eMIP eMSP WSDL from GIREVE: if needed, GIREVE's technical team can assist the eMSP in the understanding and use of eMIP WSDL
- Contact GIREVE to get Operator/Partner credentials, Ids and communicate the eMI3 Ids if already available
- Contact GIREVE to communicate the eMSP contractual relation-ship with the CPOs: GIREVE can help the eMSP in the process of the CPO contractual relation-ship setup (linking, procedures …)

Actions to be done regarding the eMSP back-end system:

- Implement the "ToIOP_xxx" and "FromIOP_xxx" web services
- Implement the communication technical aspects: HeartBeat, Logs, Error management …
- Test of the implementation before effective connection: to avoid disturbance on the operating IOP, each operator must go through a testing phase before connecting to the platform. This test phase is set up in accordance with GIREVE. It includes:
  - Schedule
  - Test Plan, test cases, anomaly management, dataset
  - Cross-configuration setting for the test environments
  - Perform the tests, detect anomalies, process and reiterate
- Get the conformity label from GIREVE to go on Deployment process
- Launch the deployment process: effective connection of the CPO system to the IOP. A particular supervision process is than operated by the GIREVE's technical team to insure that the eMSP connected system is stable.

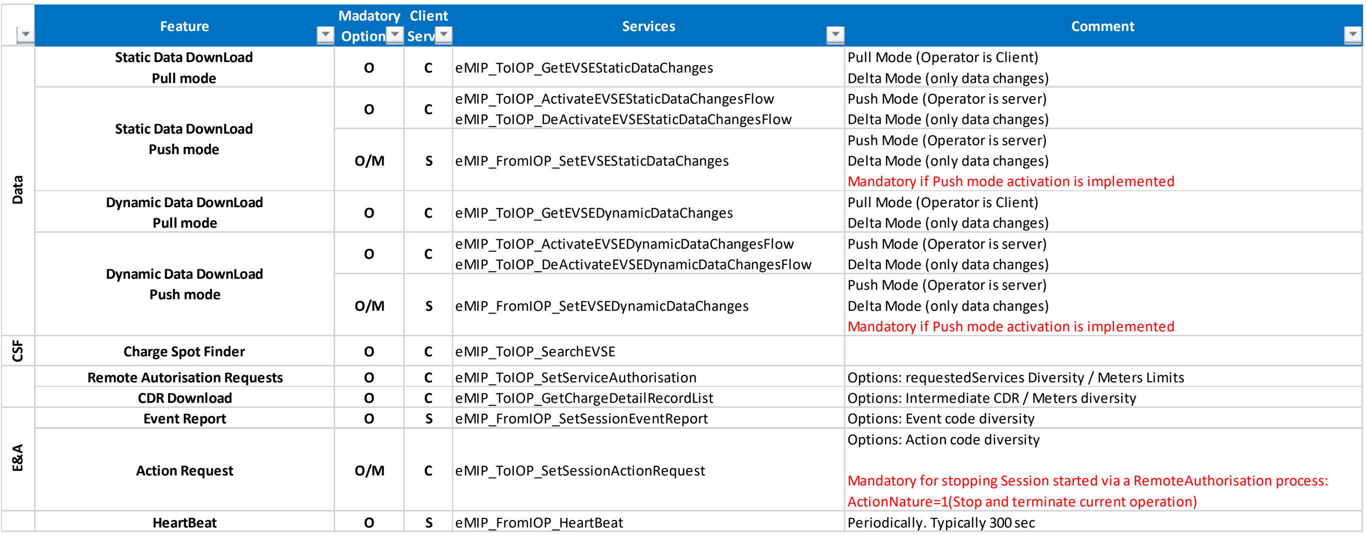
The following two paragraphs give an overview on the Web services to implement by an eMSP.

7.1.1 Summary of mandatory WebServices implementation



	Feature	Madatory Option	Client Serv	Services	Comment
Data	Static Data DownLoad Push mode	O/M	S	eMIP_FromIOP_SetEVSEStaticDataChanges	Push Mode (Operator is server) Delta Mode (only data changes) Mandatory if Push mode activation is implemented
	Dynamic Data DownLoad Push mode	O/M	S	eMIP_FromIOP_SetEVSEDynamicDataChanges	Push Mode (Operator is server) Delta Mode (only data changes) Mandatory if Push mode activation is implemented
Roaming	Autorisation Requests	M	S	eMIP_FromIOP_GetServiceAuthorisation	Options: requestedServices Diversity / Meters Limits RequestedServiceId=0 means "Is this user managed by you?". Must be implemented
	CDR Reception	M	S	eMIP_FromIOP_SetChargeDetailRecord	Options: Intermediate CDR / Meters diversity Periodically. At least 1 time per day. Max 1 time per hour
	WhiteList UpLoad	M	C	eMIP_ToIOP_SetAuthenticationData	Mandatory for contractual reasons (Operator-Gireve contract). Autorisation process response time improvement Badge collisions detection
E&A	Action Request	O/M	C	eMIP_ToIOP_SetSessionActionRequest	Options: Action code diversity Mandatory for stopping Session started via a RemoteAuthorisation process: ActionNature=1(Stop and terminate current operation)
SU	HeartBeat	M	C	eMIP_ToIOP_HeartBeat	Periodically. Typically 300 sec

7.1.2 Summary of optional WebServices implementation

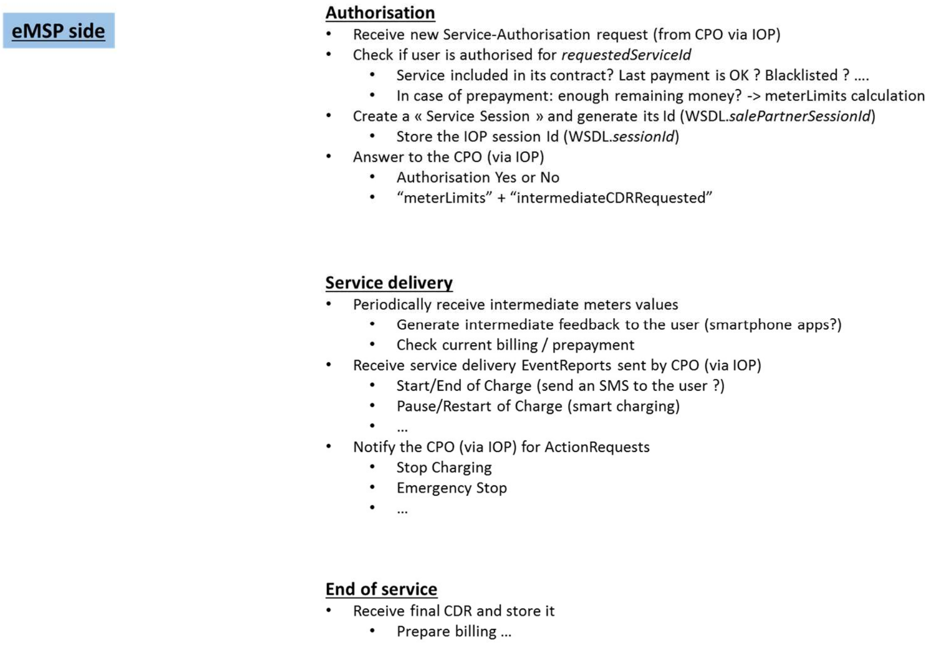


	Feature	Madatory Option	Client Serv	Services	Comment
Data	Static Data DownLoad Pull mode	O	C	eMIP_ToIOP_GetEVSEStaticDataChanges	Pull Mode (Operator is Client) Delta Mode (only data changes)
	Static Data DownLoad Push mode	O	C	eMIP_ToIOP_ActivateEVSEStaticDataChangesFlow eMIP_ToIOP_DeActivateEVSEStaticDataChangesFlow	Push Mode (Operator is server) Delta Mode (only data changes)
		O/M	S	eMIP_FromIOP_SetEVSEStaticDataChanges	Push Mode (Operator is server) Delta Mode (only data changes) Mandatory if Push mode activation is implemented
	Dynamic Data DownLoad Pull mode	O	C	eMIP_ToIOP_GetEVSEDynamicDataChanges	Pull Mode (Operator is Client) Delta Mode (only data changes)
	Dynamic Data DownLoad Push mode	O	C	eMIP_ToIOP_ActivateEVSEDynamicDataChangesFlow eMIP_ToIOP_DeActivateEVSEDynamicDataChangesFlow	Push Mode (Operator is server) Delta Mode (only data changes)
		O/M	S	eMIP_FromIOP_SetEVSEDynamicDataChanges	Push Mode (Operator is server) Delta Mode (only data changes) Mandatory if Push mode activation is implemented
CSF	Charge Spot Finder	O	C	eMIP_ToIOP_SearchEVSE	
	Remote Autorisation Requests	O	C	eMIP_ToIOP_SetServiceAuthorisation	Options: requestedServices Diversity / Meters Limits
	CDR Download	O	C	eMIP_ToIOP_GetChargeDetailRecordList	Options: Intermediate CDR / Meters diversity
E&A	Event Report	O	S	eMIP_FromIOP_SetSessionEventReport	Options: Event code diversity
	Action Request	O/M	C	eMIP_ToIOP_SetSessionActionRequest	Options: Action code diversity Mandatory for stopping Session started via a RemoteAuthorisation process: ActionNature=1(Stop and terminate current operation)
	HeartBeat	O	S	eMIP_FromIOP_HeartBeat	Periodically. Typically 300 sec

7.2 Roaming

The roaming on eMSP side means offer to customers to get services from CPOs with which the eMSP is in contract (Roaming contract).

The main use-case is started when such a user swipes its badge on an EVSE managed by a CPO. The illustration below, describes the main steps and actions to be done by eMSP.



eMSP side

Authorisation

- Receive new Service-Authorisation request (from CPO via IOP)
- Check if user is authorised for *requestedServiceId*
  - Service included in its contract? Last payment is OK ? Blacklisted ? ….
  - In case of prepayment: enough remaining money? -> meterLimits calculation
- Create a « Service Session » and generate its Id (WSDL.*salePartnerSessionId*)
  - Store the IOP session Id (WSDL.*sessionId*)
- Answer to the CPO (via IOP)
  - Authorisation Yes or No
  - “meterLimits” + “intermediateCDRRequested”

Service delivery

- Periodically receive intermediate meters values
  - Generate intermediate feedback to the user (smartphone apps?)
  - Check current billing / prepayment
- Receive service delivery EventReports sent by CPO (via IOP)
  - Start/End of Charge (send an SMS to the user ?)
  - Pause/Restart of Charge (smart charging)
  - …
- Notify the CPO (via IOP) for ActionRequests
  - Stop Charging
  - Emergency Stop
  - …

End of service

- Receive final CDR and store it
  - Prepare billing …

Figure 26: Roaming seen from eMSP

7.3 Roaming - Authorisation

7.3.1 Authorisation processes

As described in the eMIP Protocol Description document (cf doc [eMIP_Protocol_Descr]), the eMSP has two possibilities to manage end-user authorisations:

- Synchronous use case: manage the authorisation requests from the CPO via IOP ("FromIOP_GetServiceAuthorisation").
- Asynchronous use case: this use case allows identification of end-users by the IOP. The eMSPs are requested to regularly load their end-user authorisation list to the IoP using "ToIOP_SetAuthenticationData". Beside the fact that this mode allows better performance and reduce latency, it insure also a downgraded mode in case eMSP system goes off-line and make CPOs able to implement their own asynchronous use-case (semi-online EVCI).

These possibilities are summarised in Figure 27 below:

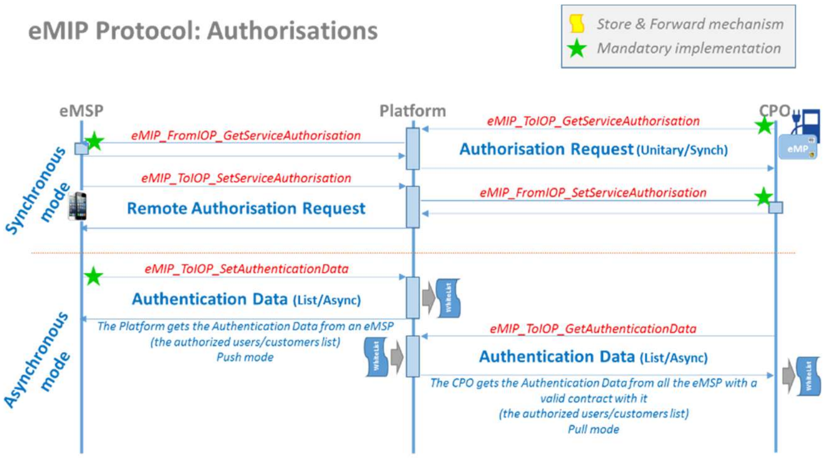

eMIP Protocol: Authorisations
Store & Forward mechanism
Mandatory implementation
eMSP
Platform
CPO
eMIP_FromIOP_GetServiceAuthorisation
eMIP_ToIOP_GetServiceAuthorisation
Authorisation Request (Unitary/Synch)
eMP
Synchronous mode
eMIP_ToIOP_SetServiceAuthorisation
Remote Authorisation Request
eMIP_FromIOP_SetServiceAuthorisation
eMIP_ToIOP_SetAuthenticationData
Authentication Data (List/Async)
WhiteList
Asynchronous mode
The Platform gets the Authentication Data from an eMSP
(the authorized users/customers list)
Push mode
eMIP_ToIOP_GetAuthenticationData
Authentication Data (List/Async)
The CPO gets the Authentication Data from all the eMSP with a valid contract with it
(the authorized users/customers list)
Pull mode

Figure 27: eMSP Authorisation Services

SetAuthenticationData – Asynchronous use case

The asynchronous use case shall always be supported by the eMSP Operators.

The eMSP shall submit any changes (Creation, Update and Deletion) on its user authorisation list to the IoP using the eMIP protocol as soon as possible.

The eMSP Operator may use the optional "expiryDate" field of an authorisation to specify its validity duration. However, if the end-user contract expires before the date, a CPO Operator may already have deployed this authorisation to its EVCI and may still authorise this end-user until the expiry date. The eMSP has to refresh frequently the Authentication data in IOP. Typically hourly. **The expiry date is the responsibility of the eMSP.**

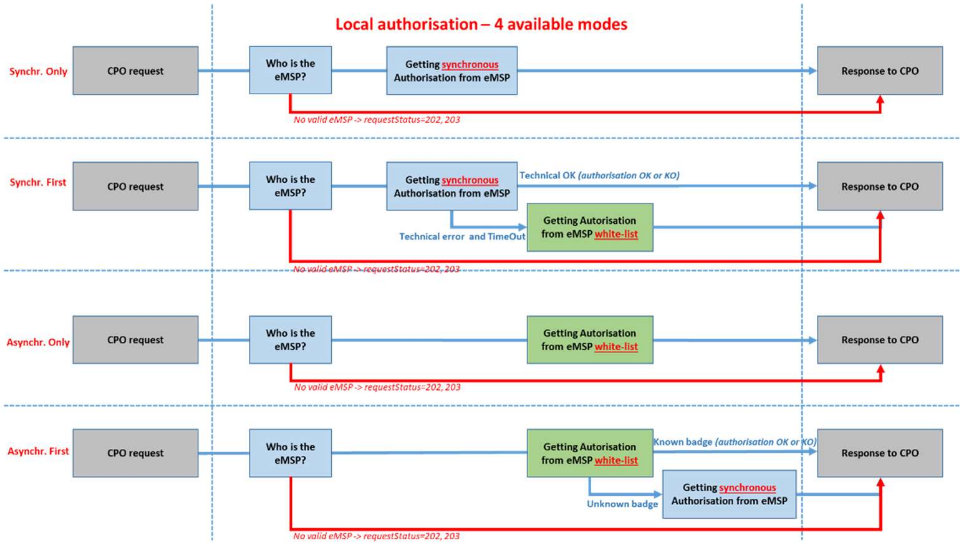
The asynchronous mode

- Put the authorisation decision act on the CPO side
- Doesn't give to eMSP any lever to improve its customer relationship

GetAuthorisationService – Synchronous use case

The eMSP Operator may support, or not, the synchronous use case. If it is supported, it shall discuss with the IoP technical team during the “Technical Setup” phase in order to select the IoP behaviour for this eMSP to be applied in case of “Authorisation request” received from a CPO:

- **Synchronous Only**: use the synchronous “GetServiceAuthorisation” use case.
- **Synchronous First**: start by using the synchronous “GetServiceAuthorisation” use case, but, if the eMSP does not reply, try to use data received asynchronously from “SetAuthenticationData”.
- **Asynchronous Only**: always use the data received asynchronously from “SetAuthenticationData”.
- **Asynchronous First**: start by using the data received asynchronously from “SetAuthenticationData”, but if no data is available, try with a synchronous “GetServiceAuthorisation” use case.


Local authorisation – 4 available modes
Synchr. Only
CPO request
Who is the eMSP?
Getting synchronous Authorisation from eMSP
Response to CPO
No valid eMSP -> requestStatus=202, 203
Synchr. First
CPO request
Who is the eMSP?
Getting synchronous Authorisation from eMSP
Technical OK (authorisation OK or KO)
Response to CPO
Technical error and TimeOut
Getting Autorisation from eMSP white-list
No valid eMSP -> requestStatus=202, 203
Asynchr. Only
CPO request
Who is the eMSP?
Getting Autorisation from eMSP white-list
Response to CPO
No valid eMSP -> requestStatus=202, 203
Asynchr. First
CPO request
Who is the eMSP?
Getting Autorisation from eMSP white-list
Known badge (authorisation OK or KO)
Response to CPO
Unknown badge
Getting synchronous Authorisation from eMSP
No valid eMSP -> requestStatus=202, 203

The synchronous mode

- Ensures the CPO of having a real and explicit authorisation from the eMSP.
- Allows the eMSP of having its own specific authorisation decision algorithm: pre-paid method, dynamic tariff…
- Gives to the eMSP a real time means to manage the relationship with its customer

We recommend for an eMSP, to implement the synchronous mode.

7.3.2 Authorisation – Asynchronous mode implementation

The eMSP has to deliver periodically the Authentication Data. The typical value for period is: “One time per day”. This data transfer is in “delta mode”. It means that the eMSP will only send Authentication Data changes, ie creations, updates and deletions.

Action Type

ActioType precises if the Authetication data line is a creation, an update or a deletion. ActionType is an enum-integer. The following table lists the valid values:

ActionCode	
Code	Signification
1	Insert
2	Update
3	Delete

Expiry date

When using ToIOP_SetAuthenticationData, the eMSP Operator will send a set of modifications on the authorisation list: new authorised subscribers, modifications on the already registered subscribers and removed subscribers.

In order to secure the authorisation process we recommend to provide an expiry date for each authentication data.

Period

The eMSP may implement periodical Authentication Data upload, or real-time Authentication Data Change upload.

Periodical

eMSP will periodically send the Authentication Data updates. This period is typically 1 time per day, but may be shorter.

- If the period is greater than (or equal to) 1 hour, we recommend to request an eMIP_ToIOP_SetAuthenticationData, even in case of empty list of changes: in case of no change during the last period, an empty list is uploaded.
- If the period is smaller than 1 hour, we recommend not to send any empty list of changes

We recommend a period between 1 day and 12 hours

Real-Time

eMSP send any change of its authentication data immediately when it occurs.

7.3.3 Authorisation – Synchronous mode implementation

Service Session

At the authorisation time, when a user request a service, the eMSP backend system shall define an eMSP-serviceSessionId do identify this service session. This Id will be send back to IOP, and will be used during data and services exchanges with IOP and/or with the CPO via IOP.

IOP sends to the eMSP backend system its own serviceSessionId, and the eMSP backend system shall store it and associate it with its own eMSP-serviceSessionId. By this way the eMSP will ensure the traceability about Service session with the IOP side.

See Ch. 5.5.1 *Service Session Management* page 30.

RequestedServiceId

See Ch.5.5.3 *RequestedServiceId* page 34.

The default value to be used is “1”, “Generic Charge Service”.

The value “0” is reserved and shall never be used by an eMSP in case of eMIP_ToIOP_GetServiceAuthorisation.

The value “0” is reserved by IOP and has a specific meaning for any eMSP: When an eMSP receive an eMIP_FromIOP_GetServiceAuthorisation request, it has to check if the requestedServiceId contains “0” or not.

- If requestedServiceId doesn’t contain “0”, it has to be interpreted as the Id of the requested service.
- Else, the eMSP backend server has to check if the user, is managed by the eMSP, and respond back to IOP a authorisation value relevant:
    - “1” means : Yes this user is managed by me
    - “2” means: No, this user is not managed by me

Please note that, in this case, the response fields partnerServiceSessionId, intermediateCDRRequested, userContractIdAlias and meterLimitList will not be used by IOP (not meaningful).

An eMSP has to implement the treatment of the requestedServiceId value “0” in case of eMIP_FromIOP_GetServiceAuthorisation.

User identification

The authorisation request (eMIP_FromIOP_GetServiceAuthorisation) contents a user identifier, in the userId mandatory field. Because the user may be identified by several different ways, the userId may content different values. This is the reason why, this data field is associated with a userIdType data field that describes the nature of identifier. The current possible values list for the userIdType is described in the following table:

UserIdType		UserId
Code	Signification	Format
RFID-UID	RFID Tag Id	Characters String Must be provided (write): uppercase with leading zeros (up to 8 ou 14 char) Must be interpreted (read): non case sensitive. Leading zeros optional (up to 8 ou 14 char)
eMI3	eMI3 Token Id	See eMI3 ("Business objects" doc)
eMA	eMAid (see eMI3)	See eMI3 ("Business objects" doc)
EVCO	EVCOId (see 15118)	See ISO 15118
EMP-SPEC	eMSP specific Id	N/A

The current typical situation for synchronous authorisation is identifying with “RFID-UID”. The userId for RFID-UID is not case sensitive. “A1B2C3D4” shall be interpreted the same way, as “a1B2c3D4”

Authorisation Value

This field contains the result of the authorisation process. The current value list is:

Authorisation	
Code	Signification
1	OK: Service is authorised
2	KO:Service is not authorised

The eMSP has to define the value regarding the result of its internal authorisation process.

Meter Limits

See Ch. 5.5.2 *Meters, and Meter Limits* page 31*.*

The eMSP may use this feature to limit the service delivery.

Note: As seen in Ch. 5.4.4 *Services and data exchange: Partner as a server, IOP as a client* page 29, on reception of a “FromIOP” request, the partner will have to route this request to the relevant Operator, based on the field “targetOperatorId” which contains the Id of this relevant operator.

7.3.4 Authorisation – Remote Synchronous mode implementation

The eMSP Operator may provide an external service to its end-users allowing them to select directly an EVSE (via an “EVSE Search” application, or using the identifier or the QrCode available on the EVSE) in order to authorise them to recharge their EV.

EVSE identification

The eMSP should identify the EVSE with the eMI3 EVSE Id:

- EVSEIdType shall be “eMI3”
- EVSEId shall be the eMI3 EVSE Id (like “FR*123*Eabcdef”)

User identification

The authorisation request (eMIP_ToIOP_SetServiceAuthorisation) shall content a user identifier, in the userId mandatory field. Because the user may be identified by several different ways, the userId may content different values. This is the reason why, this data field is associated with a userIdType data field that describes the nature of identifier. The current possible values list for the userIdType is described in the following table:

UserIdType		UserId
Code	Signification	Format
RFID-UID	RFID Tag Id	Characters String Must be provided (write): uppercase with leading zeros (up to 8 ou 14 char) Must be interpreted (read): non case sensitive. Leading zeros optional (up to 8 ou 14 char)
eMI3	eMI3 Token Id	See eMI3 ("Business objects" doc)
eMA	eMAid (see eMI3)	See eMI3 ("Business objects" doc)
EVCO	EVCOId (see 15118)	See ISO 15118
EMP-SPEC	eMSP specific Id	N/A

The current typical situation for remote authorisation is identifying with “contract Id”

- The relevant userIdType in such a situation could be “eMA” (eMAId from eMI3)
- The relevant userId in such a situation could be an eMSP specific Id, the one the user uses to log in its smartphone apps for example.
- The userId for RfID-UID is not case sensitive. We recommend to use uppercase characters
- Leading zeros must be provided to reach 8 characters in case of 4 bytes UID or 14 characters in case of 7 bytes UID

Service Session

At the authorisation time, when a user request a service, the eMSP backend system shall define an eMSP-serviceSessionId do identify this service session. This Id will be sent to IOP, and will be used during data and services exchanges with IOP and/or with the CPO via IOP.

IOP will send back to the eMSP backend system its own serviceSessionId, and the eMSP backend system shall store it and associate it with its own eMSP-serviceSessionId. By this way the eMSP will ensure the traceability about Service session with the IOP side.

See Ch. 5.5.1 *Service Session Management* page 30.

Authorisation Value

This field contains the result of the authorisation process. The current value list is:

Authorisation	
Code	Signification
1	OK: Service is authorised
2	KO:Service is not authorised

Meter Limits

See Ch. 5.5.2 Meter Report, and Meter Report List management page 30.

The eMSP may use this feature to limit the service delivery.

Authorisation Value

This field contains the result of the authorisation process. The current value list is:

Authorisation	
Code	Signification
1	OK: Service is authorised
2	KO:Service is not authorised

The only valid value for authorisationValue in case of a remote authorisation process is “1”.

7.4 Roaming - Charge Detail Record

7.4.1 Charge Detail Record Management

A Charge Detail Record (CDR) describes the result of a charging session. They are generated by the CPO and sent to the eMSP through the IoP using the eMIP protocol.

The eMSP may support or not support the synchronous “Charge Detail Record” reception by implementing the “FromIOP_SetChargeDetailRecord” request. In this case, it will receive the CDR as soon as the CPO Operator as send it to the IoP.

Otherwise, the eMSP Operator shall retrieve regularly its CDRs using the “ToIOP_GetChargeDetailRecord” or “ToIOP_GetChargeDetailRecordList” requests.

We recommend for an eMSP, to implement synchronous CDR download.

The eMSP can request to receive intermediate CDRs during the charging session using the field “IntermediateCDRRequested” during the authorisation phase.

Note: In this process, GIREVE only applies as a technical clearing house connector. **The financial compensation has to be handled between CPOs and eMSPs.**

7.4.2 Charge Detail Record Storage

The eMSP Operator shall store all versions of the Charge Detail Records, the eventual “Intermediate” ones and especially the “Final” one.

Each Charge Detail Record shall be kept stored at least one year once the financial compensation for this service has been done. After this period, depending of the applicable local regulation, all data may be completely deleted.

7.5 Roaming – End to end messaging

7.5.1 Event report

A CPO may send to the eMSP a message to notify the progress of the service delivery, or to notify a special event. These messages are associated to a serviceSession. The “eMIP_FromIOP_SetSessionEventReport” WebService (eMSP as a server) implements this feature.

An Event-Report is a message that contains an Event description (see table below) with a field that describes the Nature of the Event (which event happened?). This field is named sessionEventNature.

sessionEventType				
	sessionEventNature	*Enum Integer*	*M*	Nature of the Event to be reported
	sessionEventId	*String (eventId)*	*M(F), O(T) **	Unique Id of the Event
	sessionEventDateTime	*DateTime*	*M*	Event DateTime
	sessionEventParameter	*String*	*O*	Event Parameter
	relatedSessionActionId	*String (actionId)*	*O*	If the Event is related to an action, contains the Id of the related Session
			** M(F), O(T)*	Means **M**andatory in "FromIOP" services, **O**ptional in "ToIOP" services

The eMIP protocol and the GIREVE's IOP Platform are fully open for new event nature (new values for sessionEventNature). Nevertheless eMSP and CPO shall share the same "sessionEventNature values" catalog. The current content of this catalogue, and thus the possible values list is described in the following table:

eventReportCode	
Code	Signification
0	stopped: Emergency Stop
1	operation terminated
2	operation suspended
3	operation started
11	Start of charge
12	End of charge
13	Pre-Stop Notification

The eMSP shall implement this feature.

7.5.2 Action request

An eMSP may send to the CPO a message to request a specific action. These messages are associated to a serviceSession. The "eMIP_ToIOP_SetSessionActionRequest" WebService (eMSP as a client) implements this feature.

An Action-Request is a message that contains an Action description (see table below) with a field that describes the Nature of the Action (what is required by the eMSP, to the CPO?). This field is named sessionActionNature.

sessionActionType				
	sessionActionNature	*Enum Integer*	*M*	Nature of the Action to be requested
	sessionActionId	*String (actionId)*	*M(F), O(T) **	Unique id of the Action
	sessionActionDateTime	*DateTime*	*M*	Action DateTime
	sessionActionParameter	*String*	*O*	Action Parameter
	relatedSessionEventId	*String (eventId)*	*O*	If the Event is related to an action, contains the Id of the related Session
			** M(F), O(T)*	Means **M**andatory in "FromIOP" services, **O**ptional in "ToIOP" services

The eMIP protocol and the GIREVE's IOP Platform are fully open for new action nature (new values for sessionActionNature). Nevertheless eMSP and CPO shall share the same "sessionActionNature values" catalog. The current content of this catalogue, and thus the possible values list is described in the following table:

sessionActionCode	
Code	Signification
0	Emergency Stop
1	Stop and terminate current operation
2	Suspend current operation
3	Restart current operation

In case of a service session started by an eMSP via the "remote authorisation" mode, the right way, for the eMSP to request the end of such a service session is to send to the CPO an action request with the actionNature value set to "1: Stop and terminate current operation".

An eMSP that wants to implement the Remote authorisation mode, shall implement this feature.

7.6 Roaming - Exchanges synthesis

7.6.1 Summary of the main flows

The following illustration shows the exchanges in a service session started via a “classic authorisation process”:

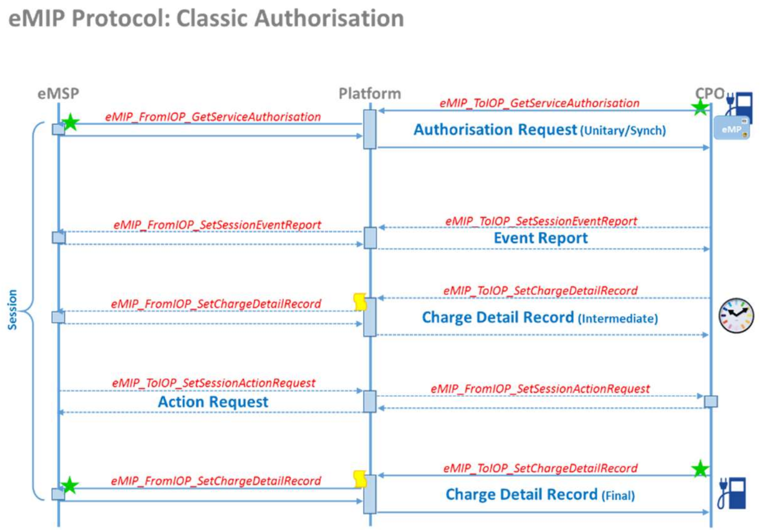

eMIP Protocol: Classic Authorisation
eMSP
Platform
CPO
eMIP_ToIOP_GetServiceAuthorisation
eMIP_FromIOP_GetServiceAuthorisation
Authorisation Request (Unitary/Synch)
eMIP_FromIOP_SetSessionEventReport
eMIP_ToIOP_SetSessionEventReport
Event Report
eMIP_FromIOP_SetChargeDetailRecord
eMIP_ToIOP_SetChargeDetailRecord
Charge Detail Record (Intermediate)
eMIP_ToIOP_SetSessionActionRequest
eMIP_FromIOP_SetSessionActionRequest
Action Request
eMIP_FromIOP_SetChargeDetailRecord
eMIP_ToIOP_SetChargeDetailRecord
Charge Detail Record (Final)
Session

Figure 28: eMSP-Roaming, flows summary (classic authorisation)

The following illustration shows the exchanges in a service session started via a “remote authorisation process”:

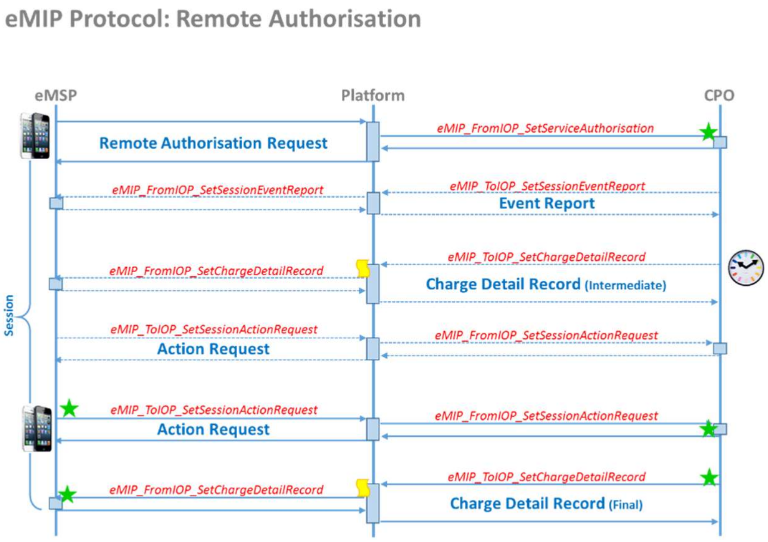

eMIP Protocol: Remote Authorisation
eMSP
Platform
CPO
Remote Authorisation Request
eMIP_FromIOP_SetServiceAuthorisation
eMIP_FromIOP_SetSessionEventReport
eMIP_ToIOP_SetSessionEventReport
Event Report
eMIP_FromIOP_SetChargeDetailRecord
eMIP_ToIOP_SetChargeDetailRecord
Charge Detail Record (Intermediate)
eMIP_ToIOP_SetSessionActionRequest
Action Request
eMIP_FromIOP_SetSessionActionRequest
eMIP_ToIOP_SetSessionActionRequest
Action Request
eMIP_FromIOP_SetSessionActionRequest
eMIP_FromIOP_SetChargeDetailRecord
eMIP_ToIOP_SetChargeDetailRecord
Charge Detail Record (Final)
Session

Figure 29: eMSP-Roaming, flows summary (remote authorisation)

7.6.2 Privacy concerns

eMSP Operators are highly recommended to never send the real user contract identifier to the IoP, and therefore to the CPOs. This may even be a requirement depending of the local regulation. An eMSP Operator can send instead a user contract identifier alias, changed regularly, which obfuscate all private information but is still unique for this eMSP.

7.6.3 Roaming – Main points of attention

- **For better user experience, we recommend for eMSP to implement the synchronous mode, for authorisation and CDR.**
- **In order to secure the authorisation process we recommend to provide an expiry date for each authentication data**
- **eMSP backend system shall define an eMSP-serviceSessionId do identify each service session, and associate the serviceSessionId sent by IOP**
- **The default value to be used for requestedServiceId is “1”, “Generic Charge Service”.**
- **The eMSP has to manage the requestedServiceId value “0” and respond depending on the fact that it manages (authorisationValue=1) or not (authorisationValue=2), the user for which the authorisation is requested.**
- **The eMSP shall implement the “*eMIP_FromIOP_ SetSessionEventReport*”, and at least trace the received message.**
- **An eMSP that wants to implement the Remote authorisation mode, shall implement the “*eMIP_ToIOP_SetSessionActionRequest*” feature.**
- **The userId associated with userIdType=”RFID-UID” shall be manage non case-sensitively**
- **On reception of an “*eMIP_FromIOP_GetServiceAuthorisation*” or “*eMIP_FromIOP_SetSessionEventReport*” request, the partner will have to internally route this request to the relevant eMSP Operator, based on the field “*targetOperatorId*”.**