OCPI v2.2.1 Protocol

IOP – OCPI Interface - Implementation Guide V1.0

Gireve

11/06/2024

OCPI Protocol Version Reference

This OCPI Protocol Implementation Guide document is based on OCPI v2.2.1, as defined in document:

- OCPI 2.2.1 Open Charge Point Interface 2.2.1, document version : 2.2.1 : https://github.com/ocpi/ocpi/tree/release-2.2.1-bugfixes?tab=readme-ov-file

Related documents

Doc reference	Content	Doc name
OCPI-2.2.1		Open Charge Point Interface 2.2.1, document version : 2.2.1

Document history

Version	Description	Date
1.0	First published version	11/06/2024

1 Introduction

1.1 Aims

This document describes guidelines to perform a proper connection with Gireve's platform using the OCPI protocol version 2.2.1.

1.2 Intended Audience

This document is dedicated to technical teams (system administrators, developers, etc.) of systems connected or to be connected to Gireve's platform through OCPI.

This document also covers some operational requirements that must be considered when implementing the OCPI 2.2.1 protocol.

1.3 Definitions and Abbreviations

Word	Meaning
IOP	Inter-operation Platform. IOP is the acronym of Gireve's eMobility Services Platform.
RPC	Référentiel des Points de Charge *(in French)* = Charge Points Repository *(in English)* The RPC is a system, built around a database that contains Electric Vehicles Charge Infrastructure (EVCI) description. It is connected to IOP. IOP's interfaces (eMIP, OCPI …) are the only way to access RPC, for partners systems.
ToIOP	Referring to flows for which an operator requests Gireve platform IOP. Partner system is client. IOP is server
FromIOP	Referring to flows for which Gireve platform IOP requests an operator. IOP is client. Partner system is server
Platform	A platform is a backend communicating with IOP through OCPI. A platform can manage one to several operators.
Operator	An operator is a business entity as a CPO or an eMSP and supervised by a platform.
Headers-To	Refers to OCPI 2.2.1 headers "OCPI-to-country-code" and "OCPI-to-party-id"
Headers-From	Refers to OCPI 2.2.1 headers "OCPI-from-country-code" and "OCPI-from-party-id"

1.4 Cardinality expression

To designate the cardinality of fields on data structures the following symbols are used :

Symbol	Description
?	Optional
1	Mandatory
*****	0 to n occurrences
+	1 to n occurrences

Gireve's data integration process includes quality controls that ensure the consistency of its data referential. The standard data quality level of Gireve is sustained by numerous mandatory attributes that are not all mandatory by the OCPI standard.

Information and Requirements

- Note that some attributes may not be mandatory by the OCPI standard but are required by Gireve for quality reasons. For further information about the cardinality of attributes by Gireve's quality standard, please refer to the document "Starting roaming operations as a CPO".

2 Integration Guidelines

2.1 Technical

2.1.1 Supported OCPI versions

IOP OCPI implementation supports two versions:

- OCPI 2.1.1
- OCPI 2.2.1

2.1.2 Security

IOP follows the security standard mechanisms of OCPI. To set up a connection with a new platform, Gireve provides connecting platform with a temporary Token to register. The Token should be used when the connection is initiated by the platform.

In the meantime, the platform can provide Gireve with temporary Token for IOP to initiate the connection. It is not mandatory as the platform is able to launch the Connection & Register process.

During the Connection & Register process, IOP and the platform exchange final Tokens to request each other, and endpoints of their respective OCPI modules.

Information and Requirements

- Contrary to OCPI 2.1.1, an OCPI 2.2.1 connection is multi-tenant and multi-role. This means that through a unique connection (i.e. OCPI Handshake) between a platform and IOP, every operator hosted by the platform can communicate with IOP, whatever their role.
- OCPI 2.2.1 requires the Authorization token, sent in headers of every request, to be encoded in base 64.
- Gireve requires all endpoints to be in HTTPS to secure communication between the partner and IOP.
- To secure data exchanges, IOP has an IP filtering for incoming requests.
- To increase security, Gireve is able to configure authentication via SSL client certificate on both ways.
  Client certificate used by the partner to request IOP is generated by Gireve.
  Client certificate used by IOP to request the partner shall be provided by the partner.

2.1.3 Platform and operator identification

OCPI is a protocol enabling the communication between operators (CPO, eMSP, …).

These operators are technically managed by OCPI platforms.

OCPI platforms are interconnected.

This means that when an operator requests another operator through OCPI, the request must contain the identifiers of the platform that requests, the operator that requests and the operator that is requested.

Information in request used for identification:

Identification of the platform that requests	Identification of the operator that requests	Identification of the requested operator
Authorization Token	Headers-From	Headers-To

Information and Requirements

- The Credentials module and flows are used to manage the OCPI connection between 2 platforms. It doesn't require the usage of Headers-From and Headers-To.
- The usage of Headers-To and Headers-From is mandatory for Gireve, except for Credentials module.
- Every request shall be sent to IOP operator, using the headers-To. IOP role as a "HUB" is to dispatch the information to other operators.

2.1.4 Traceability: X-Correlation-ID and X-Request-ID

Attributes	Description
X-Request-ID	Every request SHALL contain a unique request ID, the response to this request SHALL contain the same ID. The request ID is the ID of an exchange between two OCPI platforms.
X-Correlation-ID	Every request SHALL contain a unique correlation ID, every response to this request SHALL contain the same ID. The correlation ID is an ID enabling to trace a first request then all exchanges that are "consequences of the first request. *Example: an exchange between a CPO and IOP to update an EVSE status has the same correlation ID as all exchanges between IOP and eMSPs to forward this status update.*

Information and Requirements

- Every request sent by a partner to Gireve shall contain a X-Request-ID and a X-Correlation-ID.
- When IOP requests a platform, the platform shall respond using the same X-Request-ID and X-Correlation-ID values.
- Gireve suggests generating and using a UUID for X-Correlation-ID and X-Request-ID values.

2.1.5 Multi-tenant and multi-role capability

An OCPI 2.2.1 connection (i.e. OCPI Handshake) can be used to enable multiple operators to communicate, regardless of their roles.

Information and Requirements

- Only 1 OCPI handshake is needed to initialise the connection and enable the communication between several operators hosted by a unique platform and IOP.
- Gireve configures on its own the roles of an operator and rejects forbidden requests. *Example: If the operator is configured as eMSP only and requests the interface "CDRs Receiver" of IOP, the request is rejected.*

2.1.6 IOP is a "HUB"

Gireve presents the role "HUB" during the Credentials process. A "HUB", more than a single operator, processes and forwards the incoming request to appropriate targets.
Moreover, IOP as a "HUB", filters and cleans requests between operators.
Examples:
If a CPO is in roaming agreement with X eMSPs through Gireve, the CPO sends only 1 EVSE status update to IOP, then IOP forwards the information to the X eMSPs.
If a CPO sends several times in a row the same EVSE status for an EVSE, IOP takes into account and forwards the first, but ignores the following ones.

Information and Requirements

- IOP country_code is "FR". Its party_id is "107" on PreProduction and "007" on Production.
- Every request shall be sent to IOP operator, using the headers-To. IOP role as a "HUB" is to dispatch the information to other operators.

2.1.7 Client owned object push

OCPI introduces a specific use of resource identification mechanism, to manage situations where resource belongs to servers and situations where resource belongs to clients. See OCPI client owned object.

In OCPI 2.2.1, Objects managed through Rest protocol are owned by a CPO, an eMSP, a SCSP or a HUB.

In the same way, for OCPI 2.2.1, there are two « interfaces »: Sender/Receiver.

Module	Who is owner	Who could implement sender interface	Who could implement receiver interface
Credentials	***NA***	**An OCPI platform**	**An OCPI platform**
Locations	**CPO, NAP**	**CPO, NAP, HUB**	**All roles except CPO**
Tokens	**EMSP**	**EMSP, HUB**	**All roles except EMSP**
Commands	***NA***	**EMSP, SCSP, HUB**	**All roles except EMSP**
Sessions	**CPO**	**CPO, HUB**	**All roles except CPO**
CDRs	**CPO**	**CPO, HUB**	**All roles except CPO**
Tariffs	**CPO**	**CPO, HUB**	**All roles except CPO**
ChargingProfiles	**EMSP, SCSP**	**EMSP, SCSP, HUB**	**All roles**
HubClientInfo	**HUB**	**HUB**	**All roles**

Information and Requirements

- Connected to IOP, a platform can receive objects owned by operators behind IOP. The country_code and party_id of the object owner, in the URL and in the object, are the object owner ones.

2.1.8 Pagination

IOP implements pagination mechanisms described by OCPI. *See OCPI pagination mechanism.*

IOP requires operators to use pagination when they pull resources requesting IOP. If the operator does not use pagination arguments in its request, IOP will force it answering with the first **X** items and a Link to the second page if any.

2.1.9 Pulling Limits

For each module, IOP has its own max size limit per page. These limits can change with IOP evolutions, the operator implementation might be flexible regarding these limits.

Information and Requirements

- At the moment, IOP limits pulled objects to:
  - Locations module: 100 objects per page
  - Tokens module: 1000 objects per page
  - CDRs module: 20 objects per page
  - Tariffs module: 100 objects per page

2.1.10 List of OCPI modules

Module	Usage	Implemented by Gireve (Yes, No)
Credentials	Initialise and update IT connection information (endpoints, OCPI versions, …)	**Yes**
Locations	Transfer charge infrastructure information	**Yes**
Tokens	Transfer tokens and process Local authorizations	**Yes**
Commands	Send Commands to CPO (remote start, booking, …)	**Yes**
Sessions	Transfer information during charge	**Yes**
CDRs	Transfer the final charge detail report	**Yes**
Tariffs	Transfer tariff descriptions	**Yes**
ChargingProfiles	Transfer charging profiles (smart charge feature)	**No**
HubClientInfo	Transfer information about connection status to Hub	**No**

Information and Requirements

- Gireve, in its 1st version of OCPI 2.2.1, has implemented required features for CPOs and eMSPs to do roaming through OCPI 2.2.1. The scope implemented increases with time, updates are published in Gireve release notes and as updates in this document.

2.1.11 Gireve management of Ids

In its OCPI 2.1.1 implementation, Gireve replaces Object Ids sent by object owners by Ids managed by Gireve when objects are transferred to other parties. This transformation is required to ensure Ids unicity.
Example: An eMSP doesn't receive 2 Locations having the same Id if 2 CPOs connected to Gireve identify a Location with the same Id.

Thanks to OCPI 2.2.1 and the adding of object owner identifier in all objects (i.e. "country_code" and "party_id"), parties connected to Gireve in OCPI 2.2.1 receive Ids as sent by object owners in most modules.

Nevertheless, Gireve decides to extend Locations and Tariffs with "gireve_id" property to easier migrations of partners connected through eMIP or OCPI 2.1.1 and which would like to connect on OCPI 2.2.1.

Module	Object Id transferred to receiver in OCPI 2.1.1	Object Id transferred to receiver in OCPI 2.2.1	More information
Locations	Gireve internal Id (i.e. "gireve_id")	Object owner Id	Location object is extended with "gireve_id" when transferred to a receiver party.
Sessions	Gireve internal Id (i.e. "gireve_id")	Gireve internal Id (i.e. "gireve_id")	Gireve internal Id is used in Gireve ecosystem, including the Gireve connect place, to identify a charging session.
CDRs	Gireve internal Id (i.e. "gireve_id")	Object owner Id	-
Tariffs	Gireve internal Id (i.e. "gireve_id")	Object owner Id	Tariff object is extended with "gireve_id" when transferred to a receiver party.

2.2 Gireve management of Locations data

Gireve and its systems distinguish two natures of Location properties:

- Static data: Locations properties which almost never change.
- Dynamic data: Locations properties which can change frequently (e.g., EVSE status, Connector tariff_ids)

All Location properties are considered as static data except for the status of the EVSE, except "REMOVED" and "PLANNED" status, and the tariff_ids attached to a connector.

Information and Requirements

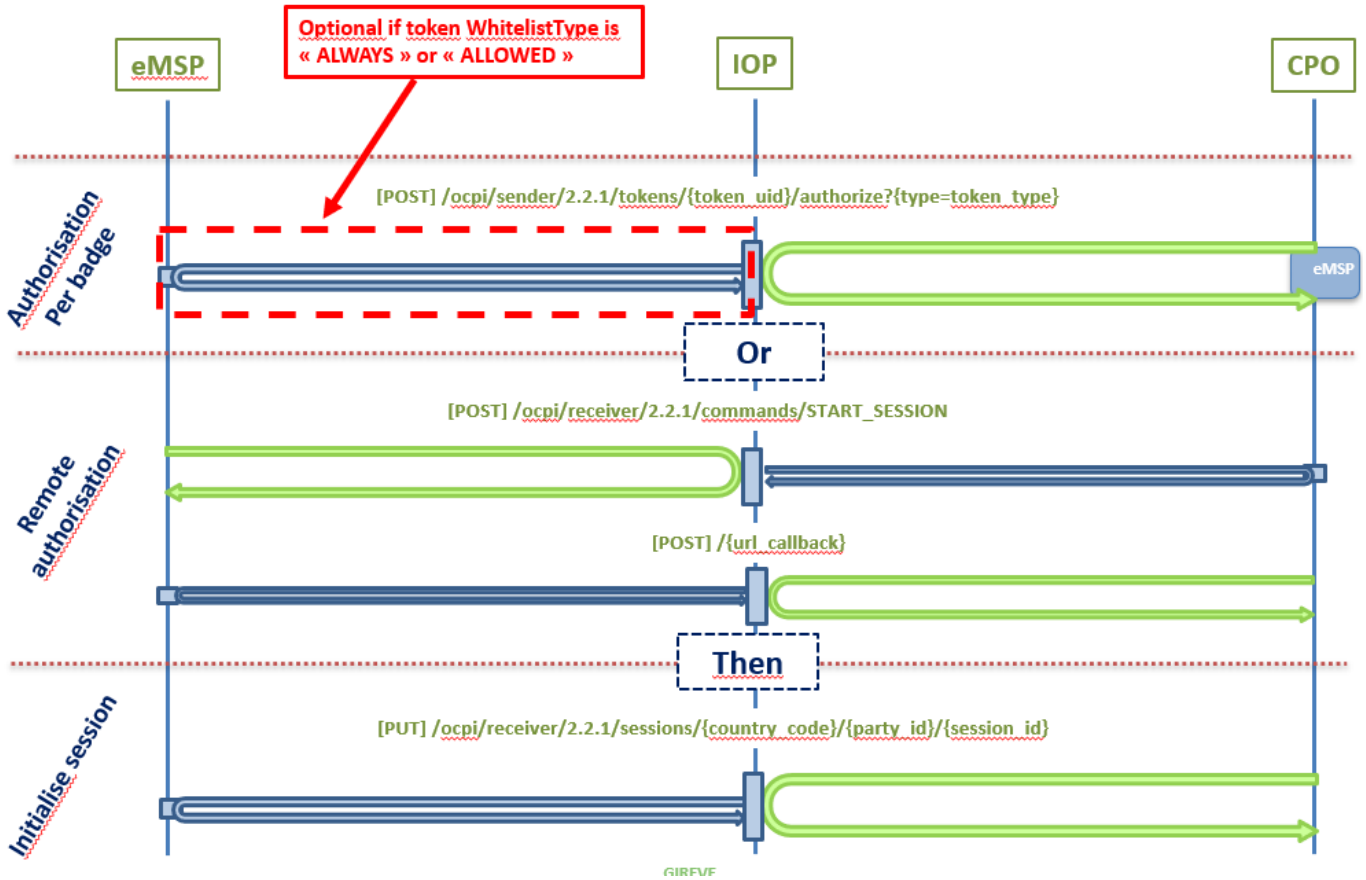
- GIREVE performs a specific process to first integrate static data of CPO Locations in its charge point repository then to integrate static data changes like changes on a Location of a CPO or new Locations or EVSEs. This process implies data quality tests and data completion of the CPO Locations. This process is asynchronous from the standard connection of the CPO with Gireve's IOP platform, meaning that new Locations of the CPO or updates on them can be seen in the Gireve charge point repository several hours after the first PUSH from the CPO to the Gireve IOP platform.

2.3 Roaming

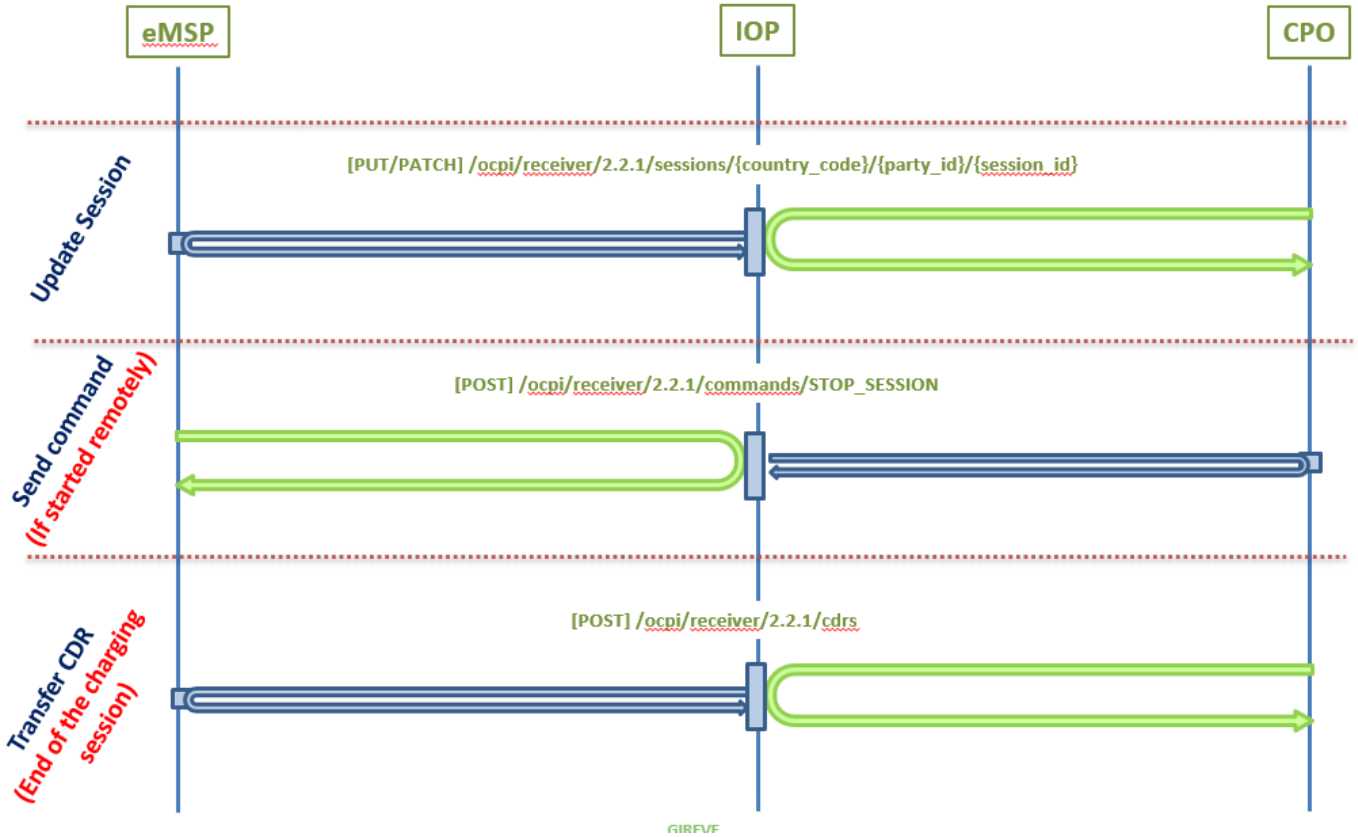
2.3.1 General workflow

An OCPI Roaming session through IOP should run the following workflow

Authorisation then initialise Session object



Session status information exchange, stop then Charge Detail Record



2.3.2 Management of B2B tariffs

The CPO connected to Gireve through OCPI have two options to manage and "publish" B2B tariffs:

B2B tariffs are described in roaming agreements: The description and the commitment related to tariffs are contained in roaming agreements signed on Gireve's Connect Place:

- Tariffs applied for a contract between an eMSP, and a CPO are defined and negotiated, using Gireve's Connect Place, before signature of the roaming agreement by both parts.
- In case of tariff updates, the CPO and the eMSP must sign an amendment whose management is fully automated.

In this case (tariffs defined in roaming agreements), the CPO should not use the OCPI Tariffs module.

B2B tariffs are not described in roaming agreements: Tariffs are not described in roaming agreements and the CPO transfers them through the OCPI Tariffs module.

- In this case, the CPO manages its tariffs on its own and eMSPs cannot give their insights or validation about applied tariffs.
- In case of tariff updates, the CPO and the eMSP do not need to sign an amendment.

Information and Requirements

- Currently, Gireve has not implemented OCPI 2.2.1 smart charge use-cases. CPOs can send their "FAST", "CHEAP" and "GREEN" tariffs, but they are not transferred to eMSPs.
- "Ad Hoc" tariffs sent by CPOs are not transferred to eMSPs either.

2.3.3 RFID Tokens

The current typical situation for identification is swiping a MIFARE badge. In this case, the relevant RFID tag in such a situation is a character 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 RFID is not case sensitive. We recommend using 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.

Information and Requirements

- For safety reasons, IOP rejects requests done with a bad RFID UID format.

3 CPO Specific Implementation Guidelines

3.1 General information

3.1.1 CPO operation definition and naming rules

A CPO operation is a homogeneous group of charging points. A CPO operation is defined by its unique eMI3 operation code also called "Operator Id" which is composed by a 2 ALPHA country code plus a 3 ALPHANUMERIC Spot Operator ID.

Example: FR*AB1

eMI3 standard official documentation can be downloaded here: https://emi3group.com/documents-links/

All EVSE included in each CPO operation must have an evse_id following the eMI3 standard naming rules defined in this specific document, page 27:

https://emi3group.com/wp-content/uploads/sites/5/2018/12/eMI3-standard-v1.0-Part-2.pdf

Specifically, all evse_id included into a given operation must always start with the Operator Id.

Example: FR*AB1*EABCDEFG*1

Information and Requirements

- It is not allowed to include in each CPO operation evse_ids starting with a different Operator-Id.
  *Example: A CPO "DE*CPO" having an "evse_id" not starting by "DECPO" or "DE*CPO" is not allowed*

3.1.2 CPO operation and roaming offers

A CPO Operation is the smallest entity that can constitute a roaming offer.

Although Gireve's technical platform (IOP) is built to manage bilateral communications between 1 operation (i.e. CPO) and another (i.e. eMSP), it is possible to aggregate several operations into a so-called network group on Gireve's Connect Place. It allows to publish a single roaming offer that includes several operations. The main advantages of such an offer structure are:

- Reduces the number of roaming agreements (1 agreement instead of several bilateral contracts – one per operation).
- Allows to add or remove operations from the network without terminating the current agreement or issuing amendments, enabling new tenants to benefit from already active roaming agreements.

3.2 Use cases covered by IOP

OCPI features are composed by several use cases that a CPO can choose to implement or not when connecting to an operator. In case of connection to Gireve, here is the list of use cases that a CPO can implement:

Use Case Family	Use case	Web Service
Credentials	**Register - FromIOP**	Get FromIOP_versions
		Get FromIOP_version-details
		Get ToIOP_versions
		Get ToIOP_version-details
		POST FromIOP_credentials
	Register - ToIOP (*)	Get FromIOP_versions
		Get FromIOP_version-details
		Get ToIOP_versions
		Get ToIOP_version-details
		POST ToIOP_credentials (*)
	Update Credentials - ToIOP	*PUT ToIOP_credentials*
	Update Credentials - FromIOP	*PUT FromIOP_credentials*
	Unregister - FromIOP	DELETE FromIOP_credentials
	Unregister - ToIOP	*DELETE ToIOP_credentials*
Locations	**Push EVCI - ToIOP**	PUT ToIOP_receiver_locations
		PUT ToIOP_receiver_locations-evse
		PUT ToIOP_receiver_locations-evse-connector
	Update EVCI - ToIOP	PATCH ToIOP_receiver_locations
		PATCH ToIOP_receiver_locations-evse
		PATCH ToIOP_receiver_locations-evse-connector
	Pull EVCI - FromIOP	GET FromIOP_sender_locations
	Check EVCI	*GET ToIOP_receiver_locations*
		GET ToIOP_receiver_locations-evse
		GET ToIOP_receiver_locations-evse-connector
Tokens exchange	*Pull Tokens - ToIOP*	*GET ToIOP_sender_tokens*

	Push Tokens - FromIOP	*PUT FromIOP_receiver_tokens*
		PATCH FromIOP_receiver_tokens
Local Authorization (Requested by CPO)	**Real-Time Authorization - ToIOP**	POST ToIOP_sender_tokens-authorize
Remote Authorization (Requested by eMSP)	**Remote start - FromIOP**	POST FromIOP_receiver_commands_START_SESSION
		POST ToIOP_sender_commands_START_SESSION-Callback
	Remote stop - FromIOP	POST FromIOP_receiver_commands_STOP_SESSION
		POST ToIOP_sender_commands_STOP_SESSION-Callback
Sessions	**Push Sessions - ToIOP**	PUT ToIOP_receiver_sessions
		PATCH ToIOP_receiver_sessions
	Pull Sessions - FromIOP	*GET FromIOP_sender_sessions*
	Check Sessions - ToIOP	*GET ToIOP_receiver_sessions*
	Push Sessions -FromIOP (smart charging)	*PUT FromIOP_sender_sessions-charging_preferences*
CDR	**Push CDRs - ToIOP**	POST ToIOP_receiver_cdrs
	Pull CDRs - FromIOP	*GET FromIOP_sender_cdrs*
	Check CDRs - ToIOP	*GET ToIOP_receiver_cdrs*
	CreditCDRs – Any	-
Tariffs	**Push Tariffs - ToIOP**	PUT ToIOP_receiver_tariffs
	Pull Tariffs - FromIOP	*GET FromIOP_sender_tariffs*
	Delete Tariffs - ToIOP	*DELETE ToIOP_receiver_tariffs*
	Check Tariffs – ToIOP	*GET ToIOP_receiver_Tariffs*
Hub Client Info	*PUSH Hub Client Info - FromIOP*	*PUT FromIOP_receiver_clientinfo*
	PULL Hub Client Info - FromIOP	*GET ToIOP_sender_clientinfo*
Booking	*Reserve now - FromIOP*	*POST FromIOP_receiver_commands_RESERVE_NOW*
		POST ToIOP_sender_commands_RESERVE_NOW-Callback
	Cancel reservation - FromIOP	*POST FromIOP_receiver_commands_CANCEL_RESERVATION*
		POST ToIOP_sender_commands_CANCEL_RESERVATION-Callback
Commands	*Unlock Connector – FromIOP*	*POST FromIOP_receiver_commands_UNLOCK_CONNECTOR*
		POST ToIOP_sender_commands_UNLOCK_CONNECTOR-Callback

(*) Not yet implemented by IOP

3.3 Use cases required by Gireve

Some use cases are required when connecting to Gireve

Always required

Use case	Why?
Register – FromIOP **Or Register - ToIOP**	These use cases are needed to initialise connection between a platform and IOP
Push EVCI - ToIOP	A CPO connected to Gireve must transfer "in real time" EVSE status changes of its EVSEs and tariff_ids of its Connectors
Pull EVCI - FromIOP	Gireve wants to be able to refresh EVCI data when needed.

If the CPO implements the "Roaming" feature

Use case	Why?
Real-Time Authorization - ToIOP	A CPO should be able to request eMSP through IOP when a driver uses his RFID badge or a PnC Contract certificate to charge.
Remote start - FromIOP	Remote authorisation and start features on CPO infrastructure are required by Gireve
Remote stop - FromIOP	Remote stop features on CPO infrastructure are required by Gireve
Push Sessions - ToIOP	A CPO must be able to send information about charging-sessions through Session objects (charge started, …)
Push CDRs - ToIOP	The CPO must send the CDR in real time after the end of the charging-session.

If the CPO doesn't commit and describe its tariffs in a roaming agreement

Use case	Why?
Push Tariffs - ToIOP	CPOs must inform in real-time, through IOP, eMSPs about tariff changes.

Information and Requirements

- The implementation and certification of the OCPI Tariffs module depends on the tariff's strategy of the CPO. It is not required if the CPO describes its tariffs through the Gireve connect place.
- CDRs shall be sent as soon as possible after the end of the charge.

3.4 Connection & Register specifications

IOP follows the OCPI 2.2.1 standard for Connection & Register process. See OCPI *2.2.1* specifications.

The Credentials module is used to manage the OCPI connection between 2 OCPI platforms. It doesn't require the usage of Headers-From and Headers-To because it is not used for an operator to communicate.
Unlike OCPI 2.1.1, the Authorization token shall be base 64 encoded in OCPI 2.2.1.

Also, Gireve presents the role "HUB" and will only presents the Operator FR*007 (Production environment) and or FR*107 (Preproduction environment).

Information and requirements

- Gireve has not implemented the "update Credentials". To update the OCPI connection information (i.e. Authorization Token and/or endpoints) the platform shall unregister with IOP then register again.
- During the registration process, a Partner can present a list of operators, but it will be ignored by IOP. Gireve manually creates/integrates the various operators ***(who have contracted with Gireve)*** a Partner can manage.
- Only 1 OCPI handshake is needed to initialise the connection and enable the communication between several operators hosted by a unique platform and IOP.
- The usage of Headers-To and Headers-From is mandatory for Gireve, except for Credentials module.
- Gireve has only implemented the "Register – FromIOP" and "Unregister – FromIOP" use cases.

3.5 Locations module specifications

IOP follows the OCPI 2.2.1 standard for Locations upload by a CPO. *See OCPI 2.2.1 specifications*.

3.5.1 Locations information required by Gireve

Gireve requires some information to be filled by CPOs in Locations data, even if they are optional in the OCPI 2.2.1 standard. This information is used to ensure data quality.

List of the required properties

Property	Reason
owner.name	Used to fill the information about the owner of the land where the charging station is located. The owner can also manage the charging station.
operator.name	Used to fill the "brand name" information
evses.connectors.tariff_ids	Used to dispatch EVSEs per tariff groups, even in case Tariffs are described through the Gireve's connect place.
evses.capabilities	Used to know, among others, if the EVSE is RFID or remote start capable.

3.5.2 Static and dynamic attributes

The attributes of the Location object are of 2 types:

- Static attributes are data attributes that do not change frequently (address, localisation …). These data are integrated in Gireve database through the Gireve quality process and could take some time before to be stored by Gireve and displayed to eMSPs.
- Dynamic attributes are data attributes that may change frequently (availability, occupied/free …). In OCPI 2.2.1, only "EVSE.status" and "Connector.tariff_ids" are considered as dynamic.
  These data are integrated in real-time by Gireve when CPOs send updates to IOP.

3.5.3 "tariff_ids" property

Gireve uses the "tariff_ids" information provided by CPOs in Locations to dispatch CPO's EVSEs into separated EVSE tariff groups. Also, CPOs can refer to these tariff groups when they describe tariffs directly in their roaming offer via Gireve's connect place.

If CPOs use the OCPI Tariffs module to send their tariffs, the management of tariffs and relations to the charging infrastructure follows the OCPI standard except that in Gireve systems, tariffs are linked to EVSEs and not to connectors.

In its current implementation of OCPI 2.2.1, Gireve stores all tariffs coming from CPOs, whatever their nature, but transfers only "B2B Regular" tariffs to eMSPs, whatever their protocol.

3.5.4 "publish" property

The "publish" information, added in OCPI 2.2.1 on Locations level, is used by CPO to inform other parties that the Location shall not be displayed on any support (i.e. a map on mobile application, …).
As this information doesn't exist in OCPI 2.1.1, Gireve doesn't send Locations with "publish" value false to eMSPs connected to IOP in OCPI 2.1.1.

3.5.5 Store and Forward – PUT and PATCH Locations

A Store and Forward mechanism shall be implemented by CPOs to ensure that no data upload may be lost, in case of a connection loss. Any data upload that didn't get a correct response (HTTP code: 2xx) from Gireve IOP platform must be stored on CPO side and a retry process must be active. After the connection recovery, the Data Upload messages must be resent in a FIFO manner.

Information and requirements

- Locations static data follow a quality process before they are integrated in Gireve's repository, meaning that Locations creation, update or deletion could take some time before being integrated by Gireve then displayed to other parties.
- As a reminder, the "evse_id" property is mandatory except if the EVSE has status "REMOVED". Moreover, this property containing the eMI3Id of the EVSE shall begin by the eMI3Id of the CPO.
- "connectors.tariff_ids", "owner.name", "operator.name" and "evses.capabilities" are mandatory for Gireve.
- Locations having "publish" flag set to false are not transferred to parties connected to Gireve in OCPI 2.1.1 protocol.

- In Gireve's model, the tariff is attached to the EVSE level and not to the connector level as in OCPI. IOP keeps "tariff_ids" attached to the first connector of the EVSE in the payload and applies them to the EVSE. "tariff_ids" of other connectors are ignored.
- A CPO shall retry every PUT and PATCH Locations in case of a technical error (i.e. timeout, http return code different than 2xx, …)

3.6 Tokens module specifications

IOP follows the OCPI standard for Tokens module. *See OCPI 2.2.1 specifications*.

3.6.1 Download of Tokens not recommended

Unlike OCPI 2.1.1, CPOs connected to Gireve in OCPI 2.2.1 don't need to download the whole list of eMSP Tokens.

In case they don't know the Token, the CPO requests IOP with a "POST Tokens-authorize" then receive the full description of the Token if the eMSP is connected to Gireve.

For this reason, Gireve doesn't include the download of Tokens by CPOs in its current implementation of OCPI 2.2.1.

3.6.2 "LocationReferences" mandatory in POST Tokens Authorize requests

As in OCPI 2.1.1, when requesting authorization, the CPO:

- Must specify 1 Location.
- Can set 0 to N EVSEs in its request. Gireve will select 1 and only 1 to continue the authorization request.

Information and requirements

- Gireve suggests that CPOs do not download Tokens of eMSPs and to send a POST Tokens authorize request in case of unknown Tokens. For this reason, Gireve doesn't include the download of Tokens by CPOs in its current implementation of OCPI 2.2.1.
- The POST Tokens authorize request shall contain the reference to a Location.

3.7 Commands module specifications

IOP follows the OCPI standard for Commands received by a CPO. *See OCPI specifications*.

3.7.1 List of available Commands

Command	Available?
START_SESSION	Yes
STOP_SESSION	Yes
RESERVE_NOW	No
CANCEL_RESERVATION	No
UNLOCK_CONNECTOR	No

3.7.2 New field "connector_id" in START_SESSION

A new field "connector_id" was added to the START_SESSION request in OCPI 2.2.1. As it is not present in OCPI 2.1.1 , eMSPs connected to IOP in OCPI 2.1.1 are not able to send this information, even if it is mandatory for CPO.

3.8 Sessions module specifications

IOP follows the OCPI standard for Sessions sent by a CPO. *See OCPI 2.2.1 specifications*.

3.8.1 Session initialisation

The CPO shall initialize a Session (i.e. send a PUT Session) after an allowed authorization and when the EV plugs to the EVSE.

This flow gives information to eMSP that the charge session of its customer has started.

3.8.2 Smart charging use cases

In its current implementation of OCPI 2.2.1, Gireve has not implemented smart charging use cases. An operator is not able to send its "Charging Preferences" on an ongoing charging session.

3.8.3 PATCH Sessions

Although Gireve has not implemented PATCH Sessions In its current OCPI 2.2.1 implementation, it is recommended for CPOs to send PATCH Sessions.

Connected partners won't have to change their OCPI implementation when Gireve makes the PATCH Sessions available.

3.8.4 Store and forward – PUT Sessions

A Store and Forward mechanism must be implemented to ensure that no Session may be lost, in case of a connection loss. Any PUT session that didn't get a correct response (HTTP code: 2xx) from Gireve IOP platform must be stored on the CPO's side and a retry process must be active. After the connection recovery, the session messages must be resent in a FIFO manner.

3.8.5 Advenir specific use case

Gireve enables CPOs to use their connection to IOP to transfer their consumption information (i.e. Sessions and CDRs) to the French subsidy program Advenir.

If used, CPOs shall send all consumption information of the charging station to IOP, even if it is not a roaming charge.

In case it is not a roaming charge, the CPO shall anonymise Sessions and CDRs with the following values

Attribute	Value
CdrToken.country_code	*$$(*)*
CdrToken.party_id	*ADV(*)*
CdrToken.uid	*12345671234567(*)*
CdrToken.type	*RFID(*)*
CdrToken.contract_id	*Gireve2Advenir(*)*

(*) Information to confirm by Gireve during onboarding

Information and requirements

- Unlike Locations and Tariffs where IOP forwards CPO Ids to eMSPs, Session Ids and authorization references are replaced by Gireve ones. This decision has been made to ensure the consistency with other protocols and Gireve systems.
- "PATCH Sessions" webservice has not been implemented in the Gireve current implementation. Nevertheless, it is recommended that CPOs already implement and use them.
- Session updates shall be stored and forwarded to Gireve in case of a technical error.
- Gireve billing feature doesn't take into account or calculate VAT. In case costs are calculated by Gireve and included in Sessions/CDRs, VAT information is not filled.

3.9 CDRs module specifications

IOP follows the OCPI standard for Sessions sent by a CPO. *See OCPI 2.2.1 specifications*.

3.9.1 CDR sending frequency

Gireve requires CDRs to be sent directly at the end of the charging-session.

eMSPs will therefore be able to display the charging-session price directly to their end customers.

3.9.2 CDR content

The "total_time" value is the total duration of this session (including the duration of charging and not charging).

It doesn't include the duration during which the EVSE is out of order so cannot supply the service. The out of order duration should be free of charge for eMSPs.

3.9.3 CreditCDR

Although Gireve has not implemented Credit CDRs use cases in its current OCPI 2.2.1 implementation, it is recommended for CPOs to manage and send them.
Connected partners won't have to change their OCPI implementation when Gireve makes this use case available.

3.9.4 Store and forward – POST CDRs

Similarly to a PUT sessions, Store and Forward mechanism must be implemented to ensure that no CDR can be lost, in case of a connection loss. Any POST Cdrs that didn't get a correct response (i.e. HTTP code: 2xx) from the Gireve platform IOP must be stored on CPO side and a retry process must be active. After the connection recovery, the Cdr messages must be resent in a FIFO manner.

3.9.5 Advenir specific use case

Gireve enables CPOs to use their connection to IOP to transfer their consumption information (i.e. Sessions and CDRs) to the French subsidy program Advenir.

If used, CPOs shall send all consumption information of the charging station to IOP, even if it is not a roaming charge.

In case it is not a roaming charge, the CPO shall anonymise Sessions and CDRs with the following values

Attribute	Value
CdrToken.country_code	*$$(*)*
CdrToken.party_id	*ADV(*)*
CdrToken.uid	*12345671234567(*)*
CdrToken.type	*RFID(*)*
CdrToken.contract_id	*Gireve2Advenir(*)*

(*) Information to confirm by Gireve during onboarding

Information and requirements

- Unlike Locations and Tariffs where IOP forwards CPO Ids to eMSPs, "Session Ids" and "authorization references" are replaced by Gireve ones. This decision has been made to ensure consistency with other protocols and Gireve systems.
- "Credit CDR" use case has not been implemented in Gireve's current implementation. Nevertheless, it is recommended that CPOs already implement and use them.
- CDRs shall be stored and forwarded to Gireve in case of technical error.
- Gireve billing feature doesn't consider or calculate VAT. In case costs are calculated by Gireve and included in Sessions/CDRs, VAT information is not filled.

3.10 Tariffs module specification

IOP follows the OCPI standard for Tariffs upload by a CPO. *See OCPI specifications.*

3.10.1 Locations tariff update

In case of tariff changes on a Location, Gireve suggests CPOs :

- Create a new Tariff with "start_date" at the date of the change.
- Update the "end_date" of the ongoing tariff.
- Attach the connectors to both tariffs, current one and next one.

3.10.2 Tariff shall be immutable

Management of tariffs is complex for CPOs and eMSPs, especially when tariff description changes (i.e. same tariff_id, different description). That's why Gireve suggests CPOs never change tariff properties attached to the price calculation.

The only information that can change shall be textual properties or end_date.

3.10.3 Differentiate tariff per eMSP

For some reason, a CPO might want to define different tariffs for a unique Location depending on the eMSP.

Example: For eMSP A on connector X, the B2B Regular tariff is 1€/h. For all other eMSPs, the tariff is 1,20€/h.

The connection to a Hub is not compatible with this use case, that's why Gireve has extended the Tariff object in OCPI 2.2.1 with 2 optional properties:

Attribute	Card.	Description
target_operator_country_code	?	Country code of the eMSP this tariff applies to. Shall be filled if target_operator_party_id is filled
target_operator_party_id	?	Party Id of the eMSP this tariff applies to. Shall be filled if target_operator_country_code is filled

This behaviour works with the following main principles:

- "tariff_ids" attached to a connector are the same, whatever the eMSP.
- For a single "tariff_id", the description of the tariff (OCPI 2.2.1 Tariffs module) can be different according to the eMSP.
- If "target_operator_country_code" and "target_operator_party_id" are null, the tariff description applies to all eMSPs (i.e. default tariff).
- When a CPO is pulled by IOP to get tariffs, it shall respond with multiple iterations for a single "tariff_id", one for each tariff description.

Example:
*CPO FR*CPO pushes 3 times tariff "tariff_A":*

tariff id	*Description*	*target_operator_country_code and target_operator_party_id values*
tariff_A	*0,50€/h*	*DE MP1*
tariff_A	*0,10€/kWh*	*NL MP2*
tariff_A	*0,80€/h*	-

"tariff_A" is 0,80€/h for all eMSPs
Except for eMSP NLMP2 for which "tariff_A" is 0,10€/kWh
And for eMSP DEMP1 for which "tariff_A" is 0,50€/h

3.10.4 Tariffs are attached to the EVSE level

In Gireve's model, the tariff is attached to the EVSE level and not to the connector level. IOP keeps "tariff_ids" attached to the first connector of the EVSE in the payload and applies them to the EVSE. "tariff_ids" of other connectors are ignored.

3.10.5 Store and forward – PUT Tariffs

Similarly to a POST Cdrs, Store and Forward mechanism must be implemented to ensure that no Tariffs can be lost, in case of a connection loss. Any PUT Tariffs that didn't get a correct response (HTTP code: 2xx) from Gireve IOP platform must be stored on CPO side and a retry process must be active. After the connection recovery, the Tariffs messages must be resent in a FIFO manner. eMSP Specific Implementation Guidelines.

Information and requirements

- In Gireve's model, the tariff is attached to the EVSE level and not to the connector level. IOP keeps "tariff_ids" attached to the first connector of the EVSE in the payload and applies them to the EVSE. "tariff_ids" of other connectors are ignored.
- Gireve extends "Tariff" object with 2 new properties target_operator_country_code and target_operator_party_id enabling a CPO to differentiate tariff description per eMSP. *See 3.10.3 Differentiate tariff per eMSP.*
- A tariff shall be immutable, that's why Gireve suggests CPOs never update a tariff description, except for text properties and "end_date"