OCPI, NDR & CDR Interface 1.0 (DRAFT v3)

- **OCPI** Open Charge point Interface
- **NDR** Notification Detail Record
- **CDR** Charge Detail Record

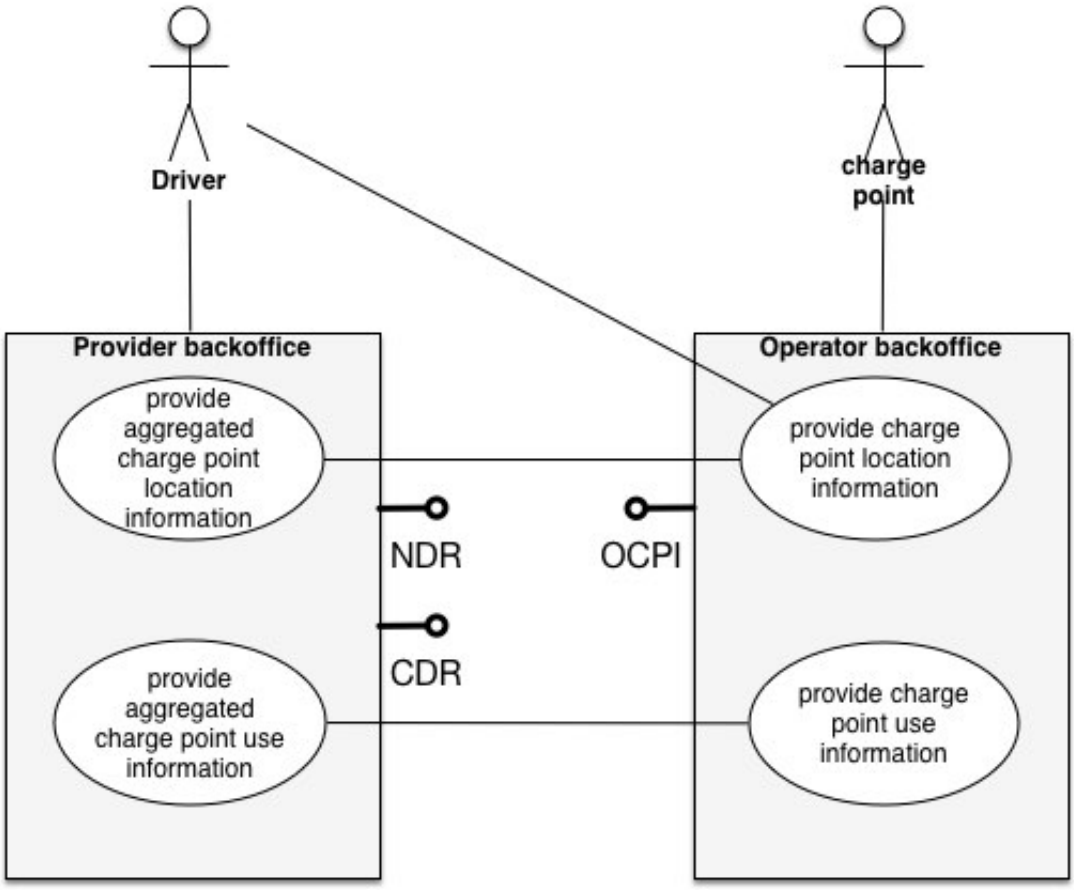
Introduction and background

The purpose of this standard is to inform EV drivers during their day to day use of charge points.

Starting in 2009, e-laad foundation and the forerunner of the eViolin association specified 2 standards in order to retrieve charge point details and active state. These are called the VAS interface and the Amsterdam interface. In 2011, eViolin combined these 2 interface into the OCPI interface allowing other parties to fetch charge point information and active state. In this same time period, the CDR format for the exchange of charge sessions was defined. This format is currently in use by the majority of the eViolin members.

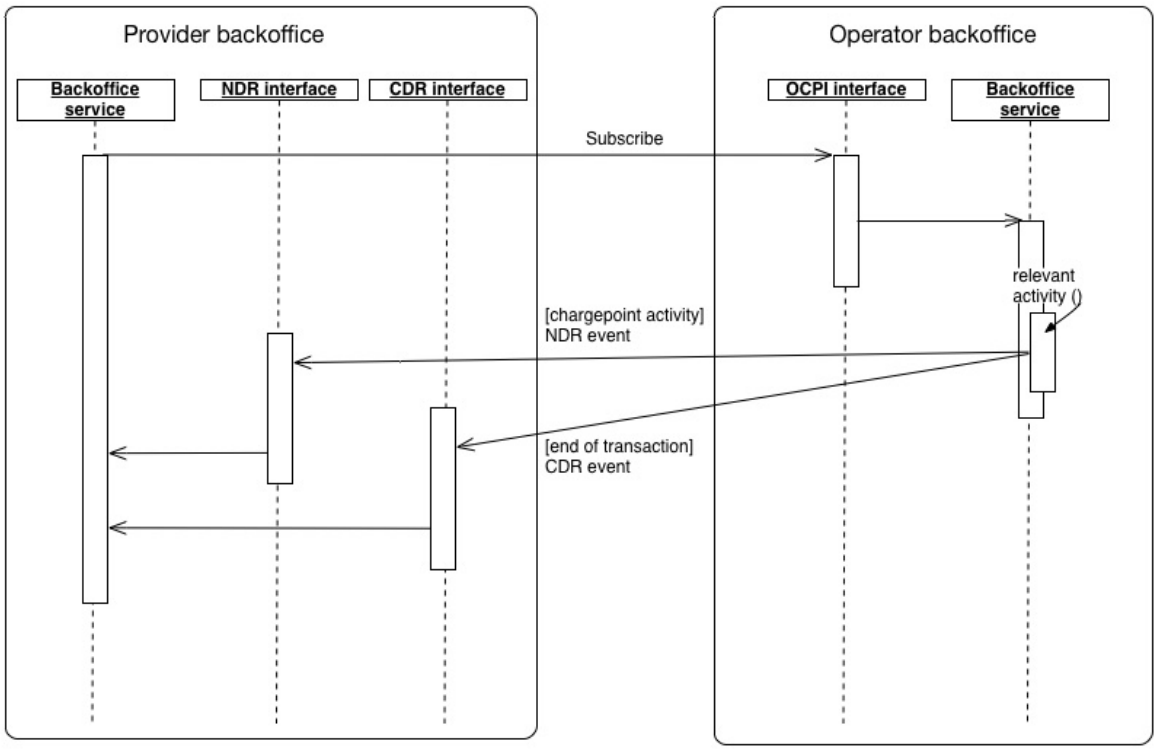
This document describes a combined set of standards based on the work done in the past. Next to that, the evolution of these standards and their use is taken into account and some elements have been updated to match nowadays use.

Overview



This overview shows the back-offices of the (Service) provider and the (charge point) operator. The charge point location information interface (OCPI) is served by the operator back-office and is used to find charge points. This will serve the static (and current status) information of the charge points. The option to subscribe to the dynamic information of the charge points is part of this interface. This dynamic information is pushed back to the provider via the NDR interface. A separate interface is available for the delivery of Charge Detail Records (CDRs) to the provider. Again this started based on a subscription by the provider.

Callback interfaces



All actions are initiated by the requesting party (mostly the provider) via the OCPI interface. A subscription ‘call’ tells the operator that this client is interested in dynamic updates. When the operator allows the client to receive these updates, the operator will start pushing information towards the NDR and/or CDR interface endpoint as given by the client during the subscription call.

The NDR and CDR interfaces are separate interfaces to allow non real-time delivery of CDR messages, not directly connected to charge point activity at that given moment.

provider perspective

The flow of a provider that makes use of these interfaces, will subscribe itself via the OCPI interface to dynamic events and CDR delivery. This allows the provider to ‘point’ out where the endpoints of the callback interfaces are for

both NDR and CDR messages. The provider will receive all generic NDR messages and specific NDR & CDR messages when their customer makes use of the charge points.

The provider may want to bundle all charge point information in order to deliver it in an aggregated way to the driver.

operator perspective

The operator registers the providers that want to make use of the OCPI based information. Part of the information maybe public (static / availability information) and could do without registration. At least the driver usage needs to be protected and only delivered to a known identity.

When a provider connects to OCPI and subscribes to the dynamic information, the operator will push all defined information connected to a driver that is a customer of this specific provider. As the operator may need time to build the Charge Detail Records, these final statements of a charge transactions are delivered via a separate interface that is not connected to a specific charge session in progress.

driver perspective

A driver is able to make use of applications that connect via their provider to these interfaces or via a third party application that makes use of the OCPI interface in a direct way.

Message format

The current structure of the interoperable interfaces is

based on JSON. The message structure is based on the OCPP2 standard when this is applicable.

The OCPP2 standard is currently under development and will be available via the Open Charge Alliance

Security

The interfaces are protected on HTTP transport level, with client side SSL and Basic Authentication. More information on basic authentication is found at the IETF

OCPI Interface operations

The OCPI interface is served by the operator.

Operation	Purpose
Find Charge points	Fetch a series of charge points given a number of search criteria
Subscribe to updates	Subscribe to status updates for charge points based on search criteria
Unsubscribe from updates	Unsubscribe to status updates for a list of charge points
Subscription status	Get the list of charge points currently subscribed to
Request Charging Profile	The ability to request a charging profile by the driver and inform the operator in order to adjust to the requested situation

Many of the details for charge points are defined as enumerations and types used within the domain of EV charging.

This document makes use of the OCPP2 spec for it’s types and enumerations.

Subscription to updates allows the caller to specify an endpoint address that will receive pushed messages. This endpoint needs to implement the NDR specified interface in order to receive messages correctly. Guaranteed message delivery may or may not be implemented, that is a decision up to the parties involved. Guaranteed message delivery SHOULD NOT put additional requirements or changes on the specified NDR format.

Find Charge points

request

Fetch a series of charge points given a number of search criteria.

Available search criteria:

Criteria	Optional	Possible values
area	o	GPS coordinates of NE and SW corners
operators	o	List of codes of the operator(s) to get charge points for
vehicleType	o	Type of vehicle to get charge points for (Car,Bike,Boat)
authorizationType	o	Charge points with certain type(s) of authorisation

response

The response contains a list of charge points matching the search criteria or an error message when the search failed. Charge points contain the following information

DO WE NEED TO COMMUNICATE THE EVSE ID IN THIS STRUCTURE ??

- identifier
- name: human readable form of the identifier
- operator
    - identifier
    - description (optional)
    - phone no (optional)
    - url (optional)
- connectors (list)
    - type: enumeration (see ConnectorType specification in OCPP2.0)
    - capabilities: enumeration (‘SetChargingProfile’) -> optional field to give additional options
    - chargeProtocol: enumeration (Unknown, Mode3, CHAdeMO, ISO15118, Uncontrolled)
    - status: enumeration (Available, Occupied, Charging, OutOfService)
    - power (replaces enum of DC50kWh / AC11kWh etc etc)
        - current (AC_1_Phase, AC_2_Phase, AC_3_Phase, DC)
        - voltage
        - amperage
    - priceSchemes (list of available options)
        - priceSchemaId: unique identifier of this schema

(at the operator)

- displayText: human readable form indicating this scheme (list of)
    - language: ISO0639–1
    - text: String
- startDate: ISO8601 date from this pricing scheme is valid (inclusive)
- expiryDate: ISO8601 date until this pricing scheme is valid (inclusive)
- tariff (list of components that build the price)
    - tariffId: identifier of this tariff (unique or at least within this schema)
    - validityRule: periodType (enum: Charging, Parking), time (iCalendar RRULE)
    - displayText: human readable form of this part of the tariff (multiple entries allowed)
        - language: ISO0639–1
        - text: String
    - currency: ISO 4217 code for currency
    - pricingUnit: enumeration of types of pricing (kWhToEV, OccupancyHours, ChargingHours, IdleHours, Session see OCPP2)
    - priceNet: amount (in smallest unit for relevant currency with an additional two decimal places, incl. VAT. e.g. euros = 0.2343, Japanese yen = 45.34)
    - priceGross: amount (Price of the unit excluding tax. Calculated as 100 ∗ priceNet / (100 + taxPct).

        - taxPct: percentage of tax
        - condition: Optional. The conditions under which this tariff is applicable. The format is not specified and left to the implementer. It is intended to be standardised in a next release of OCPP.
- reservedParking: integer with amount of reserved parking spaces for EV charging.
- vehicleType enumeration: Car, Bike, Boat (default = Car)
- authorizationTypes (list)
    - type: enumeration that described allowed identification (CIR, Bank, SMS, E-Clearing, Hubject, Provider App, Operator App, None (always usable))
    - description: human readable description
- location
    - address: address of the entry location in order to reach the charge point
    - entryLocation: GPS coordinates in order to reach the charge point (e.g. at a parking)
    - chargepointLocation: GPS coordinates of the charge point
    - floor (0 = ground floor)
    - note: human readable note to help charging
    - pictures: list of URLs (should be publicly available)
    - serviceName (name of the company/person that supports this location)
    - servicePhone (phone number to call for support (first line) at the location
    - serviceText (optional, additional text for support)
- availability
    - time (iCalendar RRULE, can be used e.g. by a mobile

app so it does not show you a charge point that is currently closed)
    - description (Opening days / hours in plain text)
    - restrictions (in text)

The iCalendar RRULE format is described in RFC 2445

NOTE: iCalendar specifies the FREEBUSY time type. Some research is required to see if this is a better match.

Subscribe

request

Fetch a series of charge points given a number of search criteria and subscribe to real-time status updates.

Available search criteria:

Criteria	Optional	Possible values
area	o	GPS coordinates of NE and SW corners
operators	o	List of codes of the operator(s) to get charge points for
vehicleType	o	Type of vehicle to get charge points for (Car,Bike,Boat)
authorizationType	o	Return charge points with certain type(s) of authorisation
identifier(s)	o	Return status updates for a (list of) specific charge point (s)

The request must also contain the endpoint URL for delivering the NDR messages. It may also include information to authenticate the user, if this is provided then the callbacks will contain contractIds for events relating to cards the user has access to.

Please note that the one providing this interface MAY put restrictions on the points that you will retrieve status updates for. It is advised to check the availability of the NDR interface at registration

When a subscribe is sent without any of the Criteria, updates of *all* charge points are expected.

response

The response contains a list of charge points (identifiers) that will be publishing events to the given endpoint, as well as a token id that can be used in future to validate received NDRs. When the request is sent without criteria, it should list *all* charge point identifiers it will send updates for. This helps the receiver to see if the list is as expected (not missing specific (new) points)

When the request is invalid or raised an error condition, an error message is returned.

The operator will thereafter publish messages to the provider via the NDR interface, the provider will return an OK to indicate that the message is accepted. When there is no OK returned, the operator will try it again until the operator sees no need in resending due to information irrelevancy

Unsubscribe

request

Unsubscribe with the token id received during subscription, plus a list of charge points that you no longer wish to receive updates for. Without a list of charge points, all notifications subscribed by this endpoint are stopped.

response

An OK or error response

SubscriptionStatus

request

Retrieve all charge points that the given token id is

subscribed for.

response

List of charge point Identifiers or an error response

RequestChargingProfile

request

message contains: evseId: Unique identifier of the EVSE that is attached to the session of the user List of [startDateTime + maxPower (in watts)] tariffType (specified in the CDR format, it is a string of 2 characters)

The EVSE is part of the message to specify the controller in use by this user. The unique EVSE number is given via the NDR interface the moment a session starts. As long as the session is active, the EVSE id is connected to Contract ID using the charge point.

As it is possible to set up a charging schedule, a list of start date time and the requested maxPower in watts is expected. The charge point will behave as specified based on it’s own current time and the part of the specified list is ‘active’. It is expected to have non-overlapping start date times. Absolute date times in the ISO8601 format are expected to prevent times relative to receiving these messages.

maxPower is specified in watts, to be compatible with the OCPP spec.

tariffType is a chosen string of 2 characters. The string is free and the specification is currently agreed upon between

operator and provider.

expected behaviour A charge point will always start charging in it’s default mode without waiting for this message as it is not said that this message will be sent / received. This message may be sent more than a single time. When it is sent more often, it’s only allowed to ask for more maxPower than before, otherwise the message should be denied. Next to that, the tariffType may only change when the maxPower is allowed to change. These restrictions are meant to prevent possible fraud as in the current processing, the last passed tariffType will be copied into the final CDR.

response

Response is an OK that it is received as expected. Actual acceptance / denial is provided via the NDR callback interface. The operator will send a ChargingProfileAccepted or ChargingProfileDenied. These are explained in the NDR interface chapter.

NDR interface

The NDR callback interface is served by the provider

This interface is implemented at the receiver side (the one that called the OCPI interface with a subscribe needs to implement this interface for receiving the NDR calls)

Please note that many different circumstances may allow the operator to select different timings to provide these messages. It should be taken into account, that the primary goal of the interface is to inform the driver. Moments to

publish the information should help the understanding of the driver of what is happening.

This interface will receive:

- operator : code of the operator
- tokenId : tokenId that matches the one returned in the subscribe response
- evseId: unique identifier of the EVSE inside the charge point
- connectorNo: connector no on the given charge point
- contractId : Contract Id that makes use of the charge point (be aware of privacy issues)
- event: specific event types are found in the table below.
- timestamp (ISO 8601)

The event type will be extended into these child objects, with their additional properties listed

Event Types

SessionStarted

SessionStarted is defined as the moment a charge point connector is occupied, this does not have to be physical, logically occupied (due to a card swipe) is another option.

contains: ∗ startDateTime in ISO8601 format

SessionEnded

SessionEnded is defined as the moment a charge point connector is available and ready for use by someone else.

contains: ∗ endDateTime in ISO8601 format

ChargingStarted

ChargingStarted is defined as the moment that actual charging takes place.

contains: ∗ startDateTime in ISO8601 format ∗ chargeSessionId

ChargingStopped

ChargingStopped is defined as the moment that actual charging has stopped.

contains ∗ endDateTime in ISO8601 format ∗ chargeSessionId ∗ wattHours

ChargingInterrupted

ChargingInterrupted is defined as the moment ?????

contains: ∗ chargeSessionId

ChargingInfoUpdated

ChargingInfoUpdated is a moment in time that charging is taken place and allows to provide information on the amount of watt hours transferred till that given moment.

contains: ∗ chargeSessionId ∗ wattHours

UserMessageCode

UserMessageCode can be transferred any given moment. This code is a string that is mapped to a certain human readable message. Commonly used codes are standardised and found in the table below.

contains: ∗ messageCode

Known UserMessageCodes

Code	Purpose
MoveYourCar	The operator likes to ask the driver to move the car currently occupying a parking spot / charge point connector

LocalBalancingActive

LocalBalancingActive indicates that at the operator level it is not possible to deliver the requested maxPower.

ChargingProfileAccepted

ChargingProfileAccepted indicates that the Requested Charging Profile is accepted and applied by the operator.

ChargingProfileDenied

ChargingProfileDenied indicates that the Requested Charging Profile is denied and will not be applied by the operator.

Small example:

When a driver parks the car, swipes a card and chooses for delayed charging, the driver expects to ‘see’

1. park and swipe: SessionStarted
2. the moment charging start (after the delay) : ChargingStarted
3. while charging: ChargingInfoUpdated till:
4. charging stops (battery is full) : ChargingStopped

Privacy note

The party publishing events should be aware that the contractId is linked to person and it’s of importance to

provide this field **only** to parties that are allowed to make use of that information**.**