

OCPI, NDR & CDR Interface 1.0 (DRAFT v4)

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

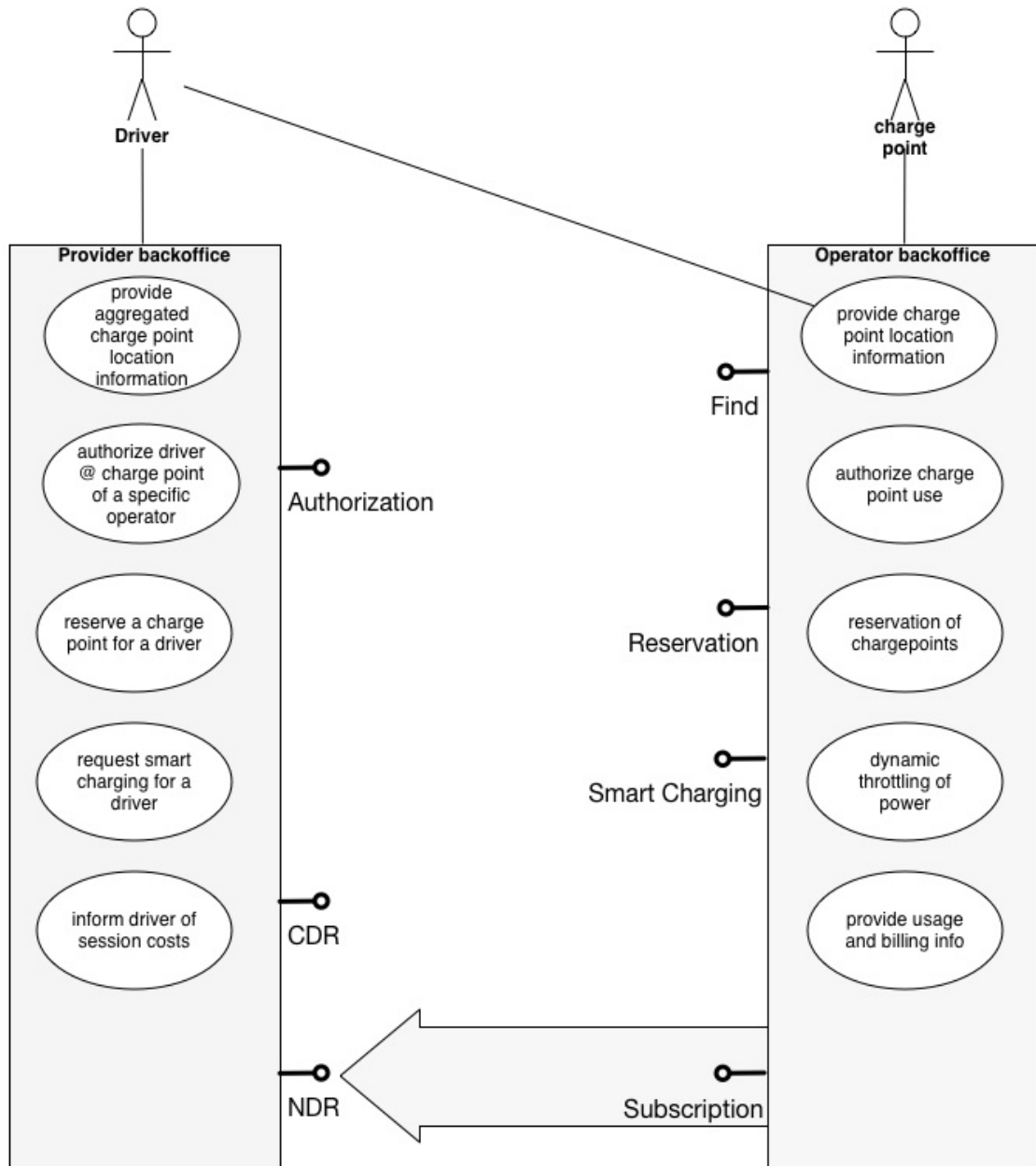
Introduction and background

The purpose of this standard is to inform EV drivers during their day to day use of charge points. Therefore, this standard is written to accommodate EV drivers in the way they make 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 interfaces 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, including the role of the driver. OCPI is a stack of interfaces that enable a driver to find and make use of a chargepoint in an informed way.

It all starts with finding a charge location that has a connector available that fulfills the needs of the driver. The 'Find' interface supports these actions in such a way that the driver is capable to query the operators on it's own or via an aggregator like the provider.

When an applicable charge location is found, a spot at that location may be reserved - or - the driver is able to keep track of status changes on the availability at that charge location via the Subscription mechanism. Again, this may be done directly at the operator - or - via an aggregating services like the provider.

The moment a driver start charging the car, there may be smart-charging capabilities available or automatically enabled. The Authorization and Smart Charging interface support several situations to get a charge session started and the Subscription interface delivering notifications (NDR) will keep the driver informed on progress and possible issues during the charge session.

Finally at the end of a charge session, a billing record in the form of a CDR provides all details and enables the complete financial handling of the full session.

Key concepts around a charge location



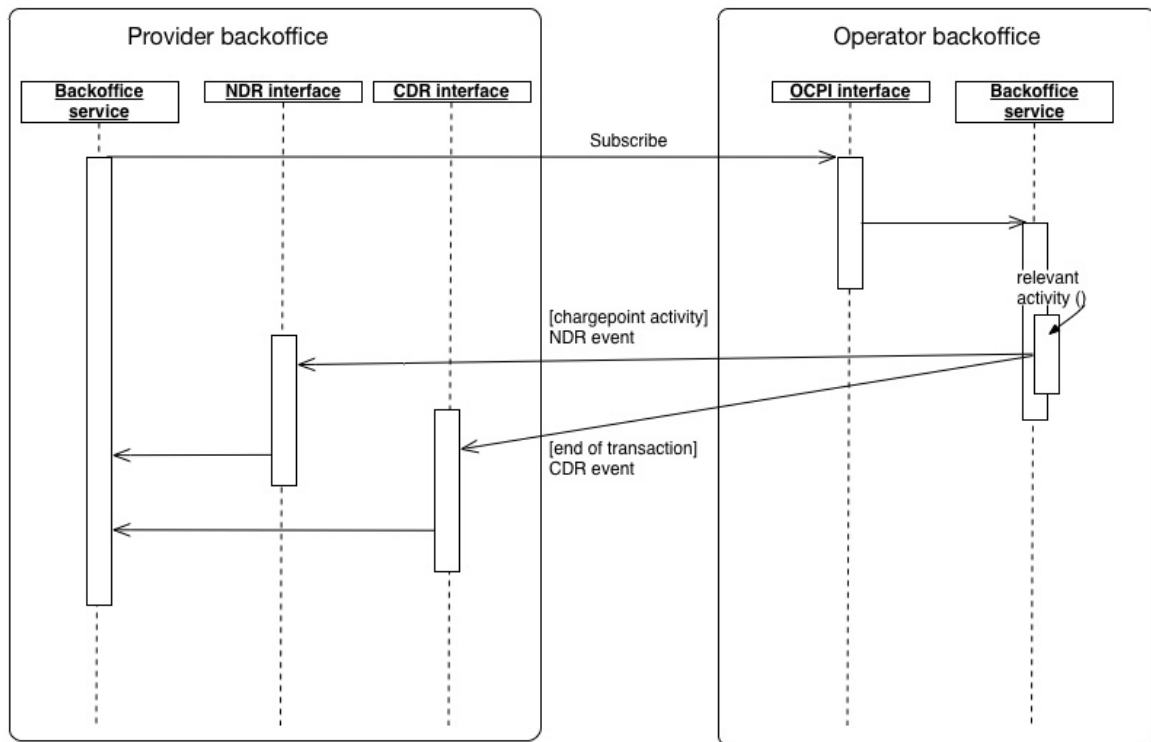
Charge location is an area that contains a number of charge points, all available via the same entry point. This allows simple aggregation for drivers and show locations that have at least a single connector of a specific type available.

Charge point is a single physical device that contains possible authentication and authorization mechanisms. In practice this is the spot where a charge card is swiped.

EVSE is the part that controls the power supply to a single EV in a single session.

Connector is the socket available for the EV to make use of. A single EVSE may provide multiple sockets but only 1 of these sockets can be in use at the same time.

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.

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 specify own endpoints of the callback interfaces 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 location 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 delivered only to a known identity.

When a provider connects to OCPI and subscribes to the dynamic information, the operator pushes all defined information for a specific 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 charge transactions may be delivered at a later stage.

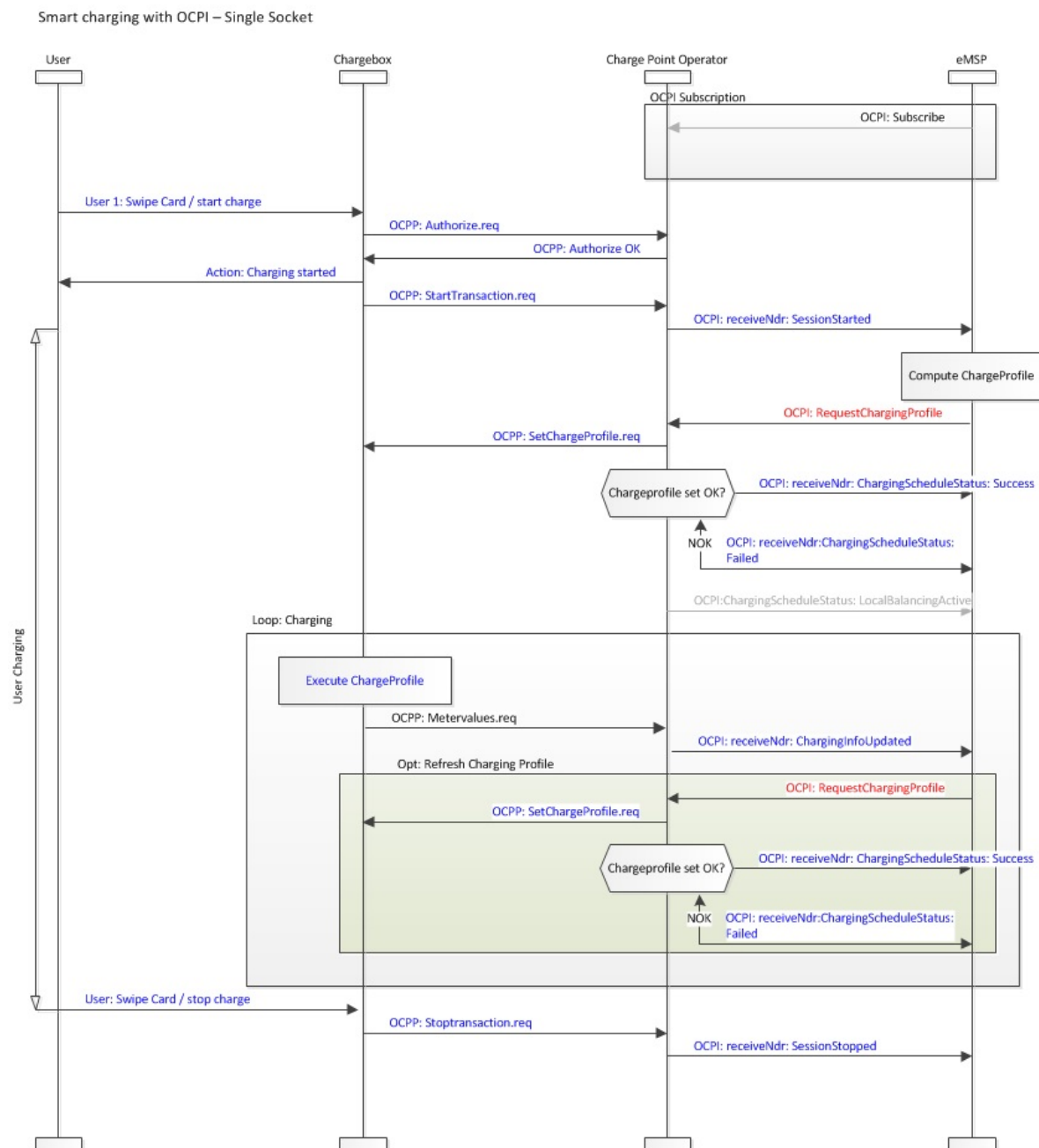
driver perspective

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. Some of the interfaces will not be available to the driver, like the CDR interface.

Smart charging support

The OCPI interfaces allows for smart charging. The Provider as

representative of the driver / customer is allowed to request a specific Charging Profile. The Operator evaluates this request and MAY change the charging profile provided to the car of the driver. The operator WILL notify the provider of its choice and actions via notifications.



This illustration shows the full flow for a smart charging situation. The communication between Provider and Operator is in scope for the

interfaces described in this document.

Security

The interfaces are protected on HTTP transport level, with SSL and Basic Authentication. Please note that this mechanism does **not** require client side certificates for authentication, only server side certificates in order to setup a secure SSL connection. More information on basic authentication is found at the [IETF](http://tools.ietf.org/html/rfc2617#page-5) See <http://tools.ietf.org/html/rfc2617#page-5>

Message format

The current structure of the interoperable interfaces is based on JSON. The message structure is based on the OCPP2 and OCHP standard when this is applicable.

The OCPP2 standard is currently under development and will be available via the [Open Charge Alliance](http://www.openchargealliance.org/?q=node/9) See <http://www.openchargealliance.org/?q=node/9> for more information.

OCHP is an open protocol for clearing house functionality. More information is found at [OCHP](http://www.ochp.eu) See <http://www.ochp.eu>.

Notes on generic data formats

Provider and Operator abbreviation

In this model, the provider and the operator play an important role. In order to target the right provider or operator, they need to be known upfront, at least between the cooperating parties.

In several standards, an issuing authority is mentioned that will keep a central registry of known Providers and Operators. At this moment, Germany and The Netherlands have an authority that keeps track of the known providers and operators. The dutch foundation, named [eViolin](#) has no publication of this registry.

The BDEW organisation keeps the registry for Germany.

- [Provider ID Liste](#) See <https://bdew-emobility.de/EMobility/ProviderIdList>
- [EVSE Operator ID Liste](#) See <https://bdew-emobility.de/EMobility/EvseOperatorIdList>

In order to keep track of the currently known providers and operators in the Netherlands, a [seperate list](#) is made available with this standard (for now). See <http://github.com/thenewmotion/ocpi/registry.md>

EVSEID

The EVSEID must follow the specification of ISO/IEC 15118-2 - Annex H "Specification of Identifiers".

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

<Country Code> <S> <EVSE Operator ID> <S> <ID Type> <Power Outlet ID>

- <Country Code> 2 ALPHA, two character country code according to ISO 3166-1 (Alpha-2-Code)
- <EVSE Operator ID> 3 (ALPHA | DIGIT) three alphanumeric characters
- <ID Type> "E"; one character "E" indicating that this ID represents an "EVSE"
- <Power Outlet ID> (ALPHA | DIGIT) * 30 (ALPHA | DIGIT <S>)
Sequence of alphanumeric characters or separators, start with alphanumeric character. * ALPHA = %x41-5A / %x61-7A; according to IETF RFC 5234 (7-Bit ASCII) DIGIT = %x30-39; according to IETF RFC 5234 (7-Bit ASCII) * <S> = *1 ("*" | "-") Optional separator

An example for a valid EVSEID 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 one of its power outlets.

EVSEID Semantics The following rules apply:

- Each EVSEID has a variable length with at least seven characters (two characters Country Code, three characters EVSE Operator ID, one character ID Type, one character Power Outlet ID) and at most thirty-seven characters (two characters Country Code, three characters EVSE Operator ID, one character ID Type, thirty- one characters Power Outlet ID).
- While the EVSE Operator ID shall be assigned by a central issuing

authority, each operator with an assigned EVSE Operator ID can choose the Power Outlet ID within the above mentioned rules freely.

- A Power outlet ID is specified as a single unit that controls the chargesession (the actual EVSE)

Backward Compatibility EVSE-IDs as defined in DIN SPEC 91286 MAY be used by applying the following mapping:

- The two digit country code "49" in Germany for geographic areas in ITU-T E.164:11/2010 is mapped onto the ISO-3166-1 (Alpha-2-Code).
- The three digit of spot operator ID is mapped 1:1 into the new alphanumeric scheme.
- All digits are mapped 1:1 into the new alphanumeric scheme.

Example: +4982312345678 *is interpreted as DE823E12345678*

Contract ID

The Contract ID is an unique identifier of a contract that is used to enable charging and related services.

The Contract ID must match the following structure (the notation corresponds to the augmented Backus-Naur Form (ABNF) as defined in RFC5234):

```
<Country Code> <S> <Provider ID> <S> <Instance> <S> <Check-Digit>
```

- <Country Code> 2 ALPHA, two character country code according to ISO 3166-1 (Alpha-2-Code)
- <Provider ID> 3 (ALPHA | DIGIT) three alphanumeric characters
- <Instance> (ALPHA | DIGIT) * 9 (ALPHA | DIGIT <S>) Sequence of alphanumeric characters or separators, start with alphanumeric character. * ALPHA = %x41-5A / %x61-7A; according to IETF RFC 5234 (7-Bit ASCII) DIGIT = %x30-39; according to IETF RFC 5234 (7-Bit ASCII) * <S> = *1 ("*" | "-") Optional separator
- <Check Digit> 1 DIGIT; according to computation

Backward Compatibility

Contract IDs as defined in DIN SPEC 91286 may be used as well by adding two zeros ("00") at the beginning of the Instance-part and the old check digit at position 14. A second check digit as referenced in this document may be calculated over the resulting ID and may be added on position 15. Example: The DIN-Contract-ID DE-8AA-123A56-3 must be set as EMAID DE- 8AA-00123A563-N.

computation of the Check Digit

TODO CHECK THIS *A unique value is determined for each of the first 11 characters of the Contract ID. The numbers (DIGIT) keep their value, the letters (ALPHA) are mapped to the values from A = 10 to Z = 35. The optional separator <S> is not taken into account and the mapping is case insensitive, i.e. "D" as well as "d" maps to the value 13.*

In the resulting list of digits, each digit is multiplied with the weight of $2^{\text{<pos>}}$ (2 to the power of <pos>) where <pos> is the position in the list. The first value in the list is positioned on number 0 (zero) The

resulting products are summed up to <checksum>

Note: There are at least 11 (if only numbers (DIGIT) are used in the Contract ID) and at most 22 (if only letters (ALPHA) are used in the Contract ID) resulting products.

Finally, <checksum> is taken "modulo 11". The result of this modulo-computation is a number between "0" and "10". The numbers between "0" and "9" are taken as Check Digit. For "10" the check digit is "X".

Note: Check digits based on the modulo-11 principle are able to identify single typing errors as well as single transposed characters.

TODO: ADD EXAMPLE HERE

Interface operations at the operator side

Operation	Purpose
Find Charge location	Fetch a series of charge locations given a number of search criteria
Subscribe	Subscribe to status updates for charge points based on search criteria
Unsubscribe	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
Request Reservation	The ability to request the operator to reserve a chargepoint

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.

Subscriptions are initiated by the provider and allows the provider to register callback interfaces for the Notifications, Authorization and CDR messages.

Subscriptions allows the caller to specify an endpoint address that will receive pushed messages. This endpoint needs to implement the callback 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, Authorization or CDR format.

Find Charge location

request

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

Available search criteria:

Criteria	Optional	Possible values
area	<input type="radio"/>	GPS coordinates of NW and SE corners
operators	<input type="radio"/>	List of codes of the operator(s) to get charge points for
vehicleType	<input type="radio"/>	Type of vehicle to get charge points for (Car,Bike,Boat)
authorizationType	<input type="radio"/>	Charge points with certain type(s) of authorisation

No Criteria will return all charge points.

A combination of the search criteria is allowed and will be interpreted as an AND query, thus limiting the results.

Additional parameters specify requests on the way the response is given.

- language : ask to only return a specific language

response

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

- identifier: identifier of this location
- name: human readable form of the identifier
- location: Entry location: where to go to in order to reach the chargepoint at the location
 - address: street address
 - postal_code
 - city
 - country
 - point: GPS coordinates
 - display_text: specifics in human readable form on the entry location
- operator: party that operates this chargepoint in it's network
 - identifier: unique identifier of the operator
 - phone (optional)
 - url (optional)
- connectors (list)
 - evse_id: the unique identifier of the EVSE used to provide the power via this connector

- type: enumeration (see ConnectorType specification in OCPP2.0)
- location: exact location of the connector (optional)
 - address: address of the entry location in order to reach the charge point
 - point: GPS coordinates in order to reach the charge point (e.g. at a parking)
 - floor: String indication of the floor according to the local situation.
- capabilities: enumeration (charging_profile_capable, reservable)
- charge_protocol: 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
- price_schemes (list of available options)
 - price_schema_id: unique identifier of this schema (at the operator)
 - display_text: human readable form indicating this scheme (list of)
 - language: ISO0639-1
 - text: String
 - start_date: ISO8601 date from this pricing scheme is

valid (inclusive)

- expiry_date: ISO8601 date until this pricing scheme is valid (inclusive)
- tariff (list of components that build the price)
 - tariff_id: identifier of this tariff (unique or at least within this schema)
 - validity_rule: period_type (enum: Charging, Parking), time (iCalendar RRULE)
 - display_text: human readable form of this part of the tariff
 - currency: ISO 4217 code for currency
 - pricing_unit: enumeration of types of pricing (kwhtoev, occupancyhours, charginghours, idlehours, session see OCPP2)
 - price_net: amount (in smallest unit for relevant currency with an additional two decimal places, incl. VAT. e.g. euros = 0.2343, Japanese yen = 45.34)
 - price_gross: amount (Price of the unit excluding tax. Calculated as $100 * priceNet / (100 + taxPct)$).
 - tax_percentage: percentage of tax
- vehicle_type enumeration: car, bike, boat (default = car)
- authorization_types (list)
 - type: enumeration that described allowed identification (cir, bank, sms, e-clearing, hsubject, providerapp, operatorapp, open (always usable))

- display_text: human readable description
- display_text: human readable note to help charging
- pictures: list of URLs (should be publicly available)
- service
 - name (name of the company/person that supports this location)
 - phone (phone number to call for support (first line) at the location)
 - display_text (optional, additional text for support)
- availability
 - start_date (start datetime for availability in ISO8601)
 - RRULE (iCalendar RRULE, can be used e.g. by a mobile app so it does not show you a charge point that is currently closed)
 - display text (Opening days / hours in plain text, may include restrictions)

Display text is always in the format that it contains a language marker and a text to allow multi language.

The iCalendar RRULE format is described in [RFC 2445](http://tools.ietf.org/html/rfc2445) See:
<http://tools.ietf.org/html/rfc2445#page-40>

NOTE: This is a description of the full format. The spec should contain a way to specify that you want to see a summarised format. This summarised format is to be defined based on the needs of application developers.

RequestReservation

request

response

Subscribe

request

Fetch a series of charge points given a number of search criteria and subscribe for notifications, authorization and billing exchange

Depending on the calling party, part of the notifications, authorization and billing may not be exchanged.

Available search criteria:

Criteria	Optional	Possible values
area	<input type="radio"/>	GPS coordinates of NW and SE corners
operators	<input type="radio"/>	List of codes of the operator(s)
vehicle_type	<input type="radio"/>	Type of vehicle to get charge points for (car,bike,boat)
authorization_type	<input type="radio"/>	Return charge points with certain type(s) of authorisation
identifier(s)	<input type="radio"/>	Return status updates for a (list of) specific charge point (s)
interface_type		Register for a specific interface type, being (NDR, Authorization or CDR)

The request must also contain the endpoint URL for delivering the NDR messages. It MAY include information to authenticate the user (the one that wants to subscribe to updates). When this is provided then the callbacks will contain contractIds for events relating to cards the user has access to.

Please note that the one pushing data to this interface MAY put restrictions on the information that you will receive status updates for. For the one pushing information - mostly the operator - : 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 subscription 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 different interfaces, 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.

security handshake between provider and operator for the callbacks.

When a provider wants to receive callback messages, the provider will give credentials along with the subscribe. In this way the operator makes use of the given credentials as Basic Authentication for the callback on the given URL in this request.

Unsubscribe

request

Unsubscribe with the subscription id received during subscription

response

An OK or error response

SubscriptionStatus

request

Retrieve all subscriptions and the evse_ids that these subscriptions support.

response

List of charge point Identifiers or an error response

RequestChargingProfile

request

message contains:

- evse-id: Unique identifier of the EVSE that is attached to the session of the user
- List of [start-datetime + max-power (in watts)]
- tariff-type (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 the wrong interpretation of time relative to receiving these messages.

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

TODO: OCPP allows for the use of Amperage, smart charging group is to decide the final format

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 its default mode without waiting for this message as it is not said that this message will be sent / received. Local Load Balancing may be applied according to the operator needs and will be notified via the NDR messages.

The RequestChargingProfile message may be sent more than a single time. When a change is applied by the operator, a new section in the chargesession will be registered for billing. (see the CDR format for more information)

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, ChargingProfileFailed or ChargingProfileDenied. These are explained in the NDR interface chapter.

Provider implemented interfaces

Notifications: NDR callback interface

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
- subscription_id : subscription id that matches the one returned in the subscribe response
- evse-id: unique identifier of the EVSE inside the charge point
- connector-no: connector no on the given charge point
- contract-id : 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

ChargeLocationAvailable

ChargeLocationOccupied

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:

- start-datetime in ISO8601 format

SessionEnded

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

contains:

- end-datetime in ISO8601 format

ChargingStarted

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

contains:

- start-datetime in ISO8601 format
- chargesession-id
- max_power (in watts)

ChargingStopped

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

contains:

- end-datetime in ISO8601 format
- chargesession-id
- watt-hours

ChargingInterrupted

The intent of ChargingInterrupted is to inform the driver of a abnormal stop of the charging session.

So when a car pauses charging and continues thereafter, there is no ChargingInterrupted event expected.

contains:

- chargesession-id

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:

- chargesession-id
- watt-hours

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:

- message-code
- display-text

Known UserMessageCodes

Code	Purpose
moveyourcar	The operator likes to ask the driver to move the car currently occupying a parking spot / charge point connector
emergency	Something at the charge location needs the immediate attention of the driver

LocalBalancingActive

localbalancing-active indicates that at the operator level it is not possible to deliver the requested maxPower.

ChargingProfileAccepted

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

ChargingProfileDenied

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

ChargingProfileFailed

chargingprofile-failed indicates that the Requested Charging Profile could 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: session-started
2. the moment charging start (after the delay) : charging-started
3. while charging: charginginfo-updated till:
4. charging stops (battery is full) : charging-stopped

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.

CDR interface

The CDR interface implemented by the provider is another callback interface. This interface allows the operator to deliver CDR messages of finished charge sessions.

- operator_id
- cdr_id
- evse_id
- contract_id
- chargesession_id (transactionid)
- start_datetime
- end_datetime
- chargepoint_type
- connector_type
- max_socket_power
- product_type
- meter_id
- currency

- vat percentage
- charging_periods
 - start_datetime
 - end_datetime
 - unit_price (itemPrice in OCHP)
 - unit_amount
 - period_cost (total cost of unit * amount, ex VAT)
 - tariff_type (parkingtime, usagetime, energy, power, servicefee)
- total_cost: sum of charging_periods, including VAT (allows calculation check) (? really required ??)

Authorization interface

The authorization interface is implemented by the provider and deals with real-time authorization requests of operators.

When a driver wants to make use of a chargepoint and the request for charging is initiated at the operator network (it is also possible that the provider initiates a charge session), the operator is able to verify the given token of identity at the service provider that handed out this token. Thereafter the service provider will issue the Contract ID that should be used to register this charge session on. That Contract ID will become part of the CDR when the session has ended and is used for specific NDR message delivery.

NOTE: In the situation of live authorizations, the provider MAY hand out temporary contract IDs that will be mapped to their customers after

CDR delivery. When central authorization is used (CIR), the Operator will report based on the contract ID found in the central authorization database.

JSON / HTTP implementation guide

Interface endpoint locations

As OCPI contains multiple interfaces, different endpoints are available for messaging. As the Provider calls the Operator and registers possible callback endpoints, the Operator interfaces need to be on a known location.

Operator interface	Endpoint
Find	/ocpi/find
Reservation	/ocpi/reserve
Smart Charging	/ocpi/smartcharge
Subscription	/ocpi/subscribe

The Provider interfaces are dynamically specified during the subscription calls and don't need a standard location

NOTE: At a later stage, the public available interfaces could be registered together at the Provider and Operator registry in order to ease interoperability.

A note on variable naming

In order to prevent issues with Capitals in variable names, the naming in JSON is not CamelCase but camel_case. All variables are lowercase and include an underscore for a space.

A note on error responses

When a request cannot be accepted, an HTTP error response code is expected including a JSON object that contains more details. HTTP status codes are described on [wikipedia](#) A 4XX type of response indicates that the request is not accepted in this way by the server. The client may adapt its request and retry.

A 5XX type of response indicates that the server is not capable of handling the request. The client should stop the request and not try it again until further notice.

The content that is send with the response is a 'application/json' type and contains a JSON object:

```
{
  "code" : "THEFAULTCODE",
  "display_text" : {
    "language": "en",
    "text": "Something went wrong, call the
Operator"
  }
}
```


Find chargepoints

Request

```
POST /api/ocpi/chargepoints
{
  "area" :
    "nw" : [-45.0, 45.0], // lon, lat
    "se" : [45.0, -45.0]  // lon, lat
  },
  "operators" : [ "ess", "tnm", "ela" ],
  "vehicle_type" : ["car"],
  "authorization_type" : ["cir", "hsubject"]
}
```

Coordinates are noted as upper left to lower right corner.

All object fields are optional, when multiple fields are noted, they work in 'AND' mode so it will limit the search results.

Note: In [GeoJSON](#) the correct coordinate order is longitude, latitude (X, Y) within coordinate arrays. This differs from many Geospatial APIs (e.g., Google Maps) that generally use the colloquial latitude, longitude (Y, X).

RESPONSE

```
200 OK
[
  {
    "authorization_type": [
```

```
{
  "display_text": [
    {
      "language": "nl",
      "text": "Standaard Tarief"
    },
    {
      "language": "en",
      "text": "Standard Tariff"
    }
  ],
  "type": "cir"
},
"availability": {
  "start_date": "2010-01-01T00:00Z",
  "rrule": "FREQ=DAILY;UNTIL=20201231T000000Z",
  "display_text": [
    {
      "language": "nl",
      "text": "Standaard Tarief"
    },
    {
      "language": "en",
      "text": "Standard Tariff"
    }
  ],
  "restrictions": {
    "display_text": [
      {
        "language": "nl",
        "text": "Standaard Tarief"
      },

```

```
        {
            "language": "en",
            "text": "Standard Tariff"
        }
    ]
}
},
"connectors": [
    {
        "capabilities": ["reservable",
"charge_profile_capable"],
        "charge_protocol": "mode3",
        "evse_id": "NL-ELA-0001",
        "location": {
            "address": "De Waag 6",
            "city": "Amersfoort",
            "country": "Nederland",
            "floor": "A1",
            "point": [
                5.3952443,
                52.1850078
            ],
            "postal_code": "3823 GE"
        },
        "power": {
            "amperage": 16,
            "current": "ac_1_phase",
            "voltage": 220
        },
        "price_schemes": [
            {
                "display_text": [
                    {
```

```
        "language": "nl",
        "text": "Standaard Tarief"
    },
    {
        "language": "en",
        "text": "Standard Tariff"
    }
],
"expiry_date": "2020-12-31T23:59Z",
"price_schema_id": "standard",
"start_date": "2010-01-01T00:00Z",
"tariff": [
    {
        "condition": "no",
        "currency": "eur",
        "display_text": [
            {
                "language": "nl",
                "text": "Standaard Tarief"
            },
            {
                "language": "en",
                "text": "Standard Tariff"
            }
        ],
        "price_gross": 0.1936,
        "price_net": 0.2343,
        "pricing_unit": "kwhtoev",
        "tariff_id": "kwrate",
        "tax_percentage": 21,
        "validity_rule": {
            "period_type": "charging"
        }
    }
]
```

```
        }
    ]
}

],
"status": "available",
"type": "ctype2"
}

],
"identifier": "CHARGE_LOCATION_001",
"location": {
    "address": "De Waag 6",
    "city": "Amersfoort",
    "country": "Nederland",
    "display_text": [
        {
            "language": "nl",
            "text": "Ga links af na de slagboom"
        },
        {
            "language": "en",
            "text": "Enter the parking and go left"
        }
    ],
    "point": [
        5.3952443,
        52.1850078
    ],
    "postal_code": "3823 GE"
},
"name": "Parking De Waag",
"operator": {
    "identifier": "ELA",
    "name": "e-laad",
```

```

        "phone": "08003522365",
        "url": "http://evnet.nl"
    },
    "pictures": [

"http://www.wijzijngroen.nl/imgsrc/800/600/user/img/album/dsc071

    ],
    "service": {
        "display_text": [
            {
                "language": "nl",
                "text": "Vraag naar de EV Desk"
            },
            {
                "language": "en",
                "text": "Ask for the EV Desk"
            }
        ],
        "name": "Gemeente Amersfoort",
        "phone": "033-423234234"
    },
    "vehicle_type": "car"
}
]

```

Subscriptions

Provider calls OCPI @ operator

REQUEST

```
POST /api/ocpi/subscribe (with Basic authentication header)
{
  "endpoint" : "https://myprovider.nl/api/ndr",
  "authentication" : {
    "type" : "basic",
    "credentials" : "aGVsbG86d29ybGQ="
  },
  "search" : {
    "area" : {
      "nw" : [-45.0, 45.0], // lon, lat
      "se" : [45.0, -45.0] // lon, lat
    },
    "operators" : [ "ess", "tnm", "ela" ],
    "vehicle_type" : ["car"],
    "authorization_type" : ["cir", "hubject"]
  }
}
```

REPLY

```
200 OK
{
  "subscription_id" : "JmO6uPb-U=1Kg7",
  "evse" : [ "evseId1", "evseId2", "evseId12" ]
}
```

Operator calls NDR @ provider

(on myprovider.nl, with basic auth "aGVsbG86d29ybGQ=")

subscription_id is uniek in combinatie met de operator

```
POST /api/ndr
{
  "operator": "NL-ELA",
  "subscription_id" : "JmO6uPb-U=1Kg7",
  , [ {
    "evse_id": "evseId2",
    "connector_no": 1,  // note 0 has a specific purpose in
OCPP and means ALL
    "contract_id":"NL-TNM-023232-X",
    "timestamp":"2014-11-11T12:56Z",
    "event" : {
      "type" : "session-started",
      "start_datetime":"2014-11-11T12:55Z"
    }
  ]
}
```

REPLY

```
200 OK // no data, received don't need to send it again.
```

Provider wants to influence a specific charging profile

Provider calls Operator


```
POST /api/chargepoint/<id> BASIC AUTH provider credentials
{
  "evse_id": "",
  "subscription_id": "",
  "schedule": {
    "start_schedule": "",
    "chargingschedule_periods": [
      {
        "max_current": "",
        "max_power": "",
        "start_period": ""
      }
    ],
    "duration": ""
  },
  "contract_id": "NL-TNM-234234-X"
  "tariff_type": ""
}
```

REPLY 200 OK / no data (when accepted) Specific notification records (NDR) will show the result of this request to the provider. This allows the operator to actually instruct a the specific EVSE to change it's behaviour and acknowledge the change.

TODO: sample needs to be fully specified

Provider / Driver wants to reserve a chargepoint

Provider calls operator

```
POST /api/xxxxxx Basic Auth operator credentials
```

```
{  
  "contract_id" : "NL-TNM-234234-3",  
  "evse_id" : "NL-ELA-23423"  
  "start_datetime" : "",  
  "expected_duration" : 120 // in minutes  
}
```

```
reply 200 OK // need for more information ?
```

Operator wants to Authorize a charge session

Operator calls Provider

```
POST /api/xxxxx Basic Auth subscription credentials
```

```
{  
  "token" : { // OR contract_id (both allowed)  
  
  }  
}
```

```
reply 200 OK
```

```
{  
  "contract_id" : "NL-TNM-273849-X"  
  ??? Indications on allowed charging ???  
}
```

****TODO:** complete sample and define message exchange based on OCHP tokens**