An Overview of EV Roaming Protocols

Dwidharma Priyasta[1,2,a], Hadiyanto[2,b], Reza Septiawan[1,c]

[1]*Research Center of Electronics, BRIN, Indonesia*

[2]*School of Postgraduate Studies, Diponegoro University, Indonesia*

[a)]*dwid002@brin.go.id*

[b)]*hadiyanto@che.undip.ac.id*

[c)]*reza001@brin.go.id*

Abstract. Roaming in electric mobility (EV roaming) enables EV users to make use of any charging station belonging to other networks with only a single user registration based on roaming agreement between operators. This paper presents an overview of the functionalities of the four major existing EV roaming protocols, namely the Open Charge Point Interface (OCPI), the Open InterCharge Protocol (OICP), the Open Clearing House Protocol (OCHP), and the eMobility Inter-operation Protocol (eMIP). This paper focuses on how each EV roaming protocol works in terms of data exchange with the Open Charge Point Protocol (OCPP) which is the de-facto protocol embedded in many charging stations available in the market. Related actors and their roles that are defined in each protocol will be presented. Examples are given in the form of a sequence diagram in order to depict the interaction between actors in the case of user registration, start a charging session, stop a charging session, and billing. This paper concludes that employing the sequence diagram is one effective method to achieve a fast learning curve while studying the EV roaming protocols.

1. Introduction

Roaming in electric mobility (EV roaming) plays an important role in the electric vehicle charging ecosystem. EV roaming provides the ability for different charging infrastructures to communicate with each other using a single protocol or multiple interoperable protocols [1]. Furthermore, EV roaming enables an EV user who does not (directly) have a contract with a Charge Point Operator (CPO) to make use of any charge point operated by the CPO at a condition that the EV user (directly) has a contract with a Mobility Service Provider (MSP) and the MSP also has a contract with the CPO either directly or via a roaming hub [2]. Therefore, if an MSP has many contracts with many CPOs, then this gives an opportunity for EV users to use more charge points with only a single user registration.

There are four major EV roaming protocols exist, namely the Open Charge Point Interface (OCPI), the Open InterCharge Protocol (OICP), the Open Clearing House Protocol (OCHP) and the eMobility Inter-operation Protocol (eMIP). These protocols are used in the communication domain between MSPs and CPOs. On the other hand, inside the CPO domain itself there usually exists another protocol, namely the Open Charge Point Protocol (OCPP).

OCPP is an application protocol for communication between a charge point and the central management system of the charge point. OCPP is the de-facto protocol embedded in many charging stations available in the market. Therefore, the main objective of this paper is to show how the four major existing EV roaming protocols (OCPI, OICP, OCHP and eMIP) work together with OCPP in terms of data exchange during a charging session.

This paper begins with the overview of OCPP, and follows with the overview of each EV roaming protocols based on a desk research of the actual protocol documentation of OCPP 1.6, OCPI 2.2.1, OICP 2.3, OCHP 1.4, and eMIP 1.0.14. Related actors and their roles that are defined in each document will be presented. Examples are given in the form of a sequence diagram in order to depict the interaction between actors in the case of user registration, start a charging session, stop a charging session, and billing. These examples intend to show how these EV roaming protocols work together with the OCPP to enable services for EV users.

2. Overviews

2.1. Open Charge Point Protocol (OCPP)

OCPP is an open protocol for communication between two actors in the CPO domain, the Charge Point (technical term for EV charging stations) and the Central System (technical term for central system that manages EV charging stations). OCPP is managed by the Open Charge Alliance (OCA) and was designed to accommodate any type of charging technique [3]. The latest version to date is OCPP 2.0.1, but this paper only covers OCPP 1.6. So, the word 'OCPP' without version which appears elsewhere in this paper always means OCPP 1.6.

OCPP is based on JSON/RESTful API with WebSocket as the basis communication protocol. The Charge Point plays the role of a WebSocket client, whereas the Central System is the WebSocket server. OCPP defines a set of messages such as:

- **Authorize()** is sent by the Charge Point to the Central System to request for the authorization of an identifier belongs to an EV user who wants to start or to stop charging,
- **RemoteStartTransaction()** is sent by the Central System to the Charge Point to request the Charge Point to start a charging session,
- **StartTransaction()** is sent by the Charge Point to the Central System to inform about a charging session that has been started,
- **MeterValues()** is sent by the Charge Point to the Central System to provide extra information about its electrical meter values or other sensor values periodically,
- **RemoteStopTransaction()** is sent by the Central System to the Charge Point to request the Charge Point to stop a charging session, and
- **StopTransaction()** is sent by the Charge Point to the Central System to notify that a charging session has stopped.

OCPP messages contain one or more required and optional fields. An example is shown in Table 1 which is taken from the OCPP 1.6 specification document.

Table 1. Required and optional fields of RemoteStartTransaction()

Field Name	Description
connectorId	Optional. Number of the connector on which to start the transaction. connectorId SHALL be > 0.
idTag	Required. The identifier that Charge Point must use to start a transaction.
chargingProfile	Optional. Charging Profile to be used by the Charge Point for the requested transaction. ChargingProfilePurpose MUST be set to TxProfile.

One charging cycle according to the OCPP scheme can be as shown in Figure 1. It is based on the request-response mechanism. An EV user who wants to use the Charge Point must be registered in the Central System and recognized as a valid user of the Charge Point. The user authorization is based on RFID tags or smart cards. Fig. 1 depicts the interaction between the Charge Point and the Central System during a charging session, which can be interpreted as follows:

1. The Charge Point sends Authorize.req() which contains the unique identifier (UID) of RFID or smart card belonging to an EV user for the authorization. The Central System responds with Authorize.conf() which contains the status to indicate whether or not the user request can be granted.
2. If charging occurs, the Charge Point sends StartTransaction.req() to inform the Central System that a charging session has been started for the EV user. The Central System responds with StartTransaction.conf() that includes the transaction identifier and the authorization status of the EV user.
3. To stop the charging session, the Charge Point needs to verify whether the EV user is the one that initiated the charging session or someone else who is allowed to terminate the charging session. Therefore, the Charge Point sends another Authorize.req() to the Central System.
4. Once authorized and the event stop, the Charge Point sends StopTransaction.req() to notify the Central System which in turn responds with StopTransaction.conf(). The request contains the transaction identifier to appoint which event has stopped.

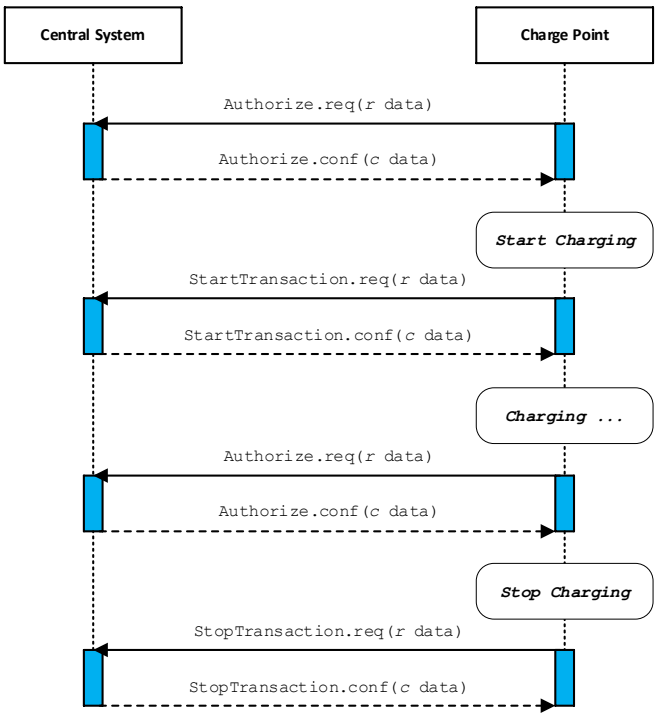

Fig. 1. One charging cycle based on OCPP 1.6

2.2. Open Charge Point Interface (OCPI)

OCPI is a scalable and automated EV roaming protocol for communication between MSPs and CPOs [4]. OCPI is managed by the Nederlands Kennisplatform Laadinfrastructuur (NKL). OCPI is publicly available without registration, and is published under the Creative Commons Attribution-NoDerivatives 4.0 International Public License. The latest version to date is OCPI 2.2.1.

OCPI is based on JSON/RESTful API with HTTP as the basis communication protocol. It is a real-time protocol and supports synchronous as well as asynchronous operations. Actors and roles that are defined in OCPI and relevant to the main objective of this paper are as follows:

- The **e-Mobility Service Provider (eMSP)** provides services for charging electric vehicles (a role of an MSP as mentioned elsewhere in this paper). eMSPs connect to CPOs to provide roaming services.
- The **Charge Point Operator (CPO)** operates EV charging stations (a role of a CPO as mentioned elsewhere in this paper). CPOs connect to eMSPs to enable roaming services.
- The **Charge Point** delivers energy to electric vehicles.
- The **EV user** connects to a charging station through an eMSP.

OCPI also supports the following features:

- **Roaming via hub:** OCPI enables data exchange between eMSPs and CPOs via one or multiple hubs that are a module in OCPI.
- **Roaming peer-to-peer:** eMSPs and CPOs can connect directly via OCPI.
- **Authorization:** OCPI has a certain module to share tokens data from eMSPs with CPOs. Authorization can be done real-time ('token.whitelist=NEVER') or via a whitelist ('token.whitelist=ALWAYS').
- **Billing:** OCPI supports sending Charge Detail Records (CDRs) for the purpose of invoicing.
- **Remote start/stop:** eMSPs can start or stop a charging session.

2.2.1. OCPI : User Registration

EV users are identified by tokens. Tokens may be in the form of RFID tags or smart cards or others that are given by eMSPs at a user registration process. eMSPs could share tokens data with CPOs, so that CPOs can manage the whitelist of valid user tokens, which in a certain case can be used as a reference by CPOs to authorize an EV user.

According to OCPI, there are two ways to share tokens data with CPOs:

1. The CPO determines an end-point with the following format:

   ```
   {endpoint_url}/{country_code}/{party_id}/{token_uid}[?type={type}]
   ```

   , for example:

   ```
   https://www.server.com/ocpi/cpo/2.2/tokens/NL/TNM/01234567
   ```

   , so that the eMSP can upload tokens data to the CPO as shown in Figure 2 with a HTTP PUT as follows:

   ```
   PUT To URL:
   https://www.server.com/ocpi/cpo/2.2/tokens/NL/TNM/01234567?type=RFID
   {
     "country_code": "NL",
     "party_id": "TNM",
   ```

```
  "uid": "01234567",
  "type": "RFID",
  "valid": true,
  "whitelist": "ALWAYS"
  ...
}
```

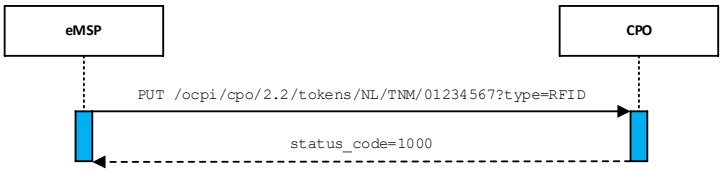

Fig. 2. OCPI HTTP PUT user tokens

2. The eMSP determines an end-point, so that the CPO can download tokens data as shown in Figure 3 with a HTTP GET as follows:

```
GET From URL:
https://www.server.com/ocpi/emsp/2.2/tokens/?offset=50&limit=100
```

Fig. 3. OCPI HTTP GET user tokens

2.2.2. OCPI : Start Transaction

The start transaction event begins with the eMSP sending a Command() module which contains the START_SESSION command to the CPO as shown in Figure 4. The Command() module also includes other information, such as the token that triggers the transaction. The token data is to be assigned to parameter idTag of the RemoteStartTransaction.req() sent by the CPO to the Charge Point in order to start a charging session.

The authorization of a user token depends on the value of parameter ‘token.whitelist’. If ‘token.whitelist=NEVER’, then the token must be authorized real-time, not white-listed. In this case, the CPO can request the eMSP to execute the authorization and send the result to the CPO as shown in Fig. 4. Whereas if ‘token.whitelist=ALWAYS’, then the CPO has to do the authorization based on the whitelist of valid user tokens it has. After that, the transaction can be proceed as defined in OCPP.

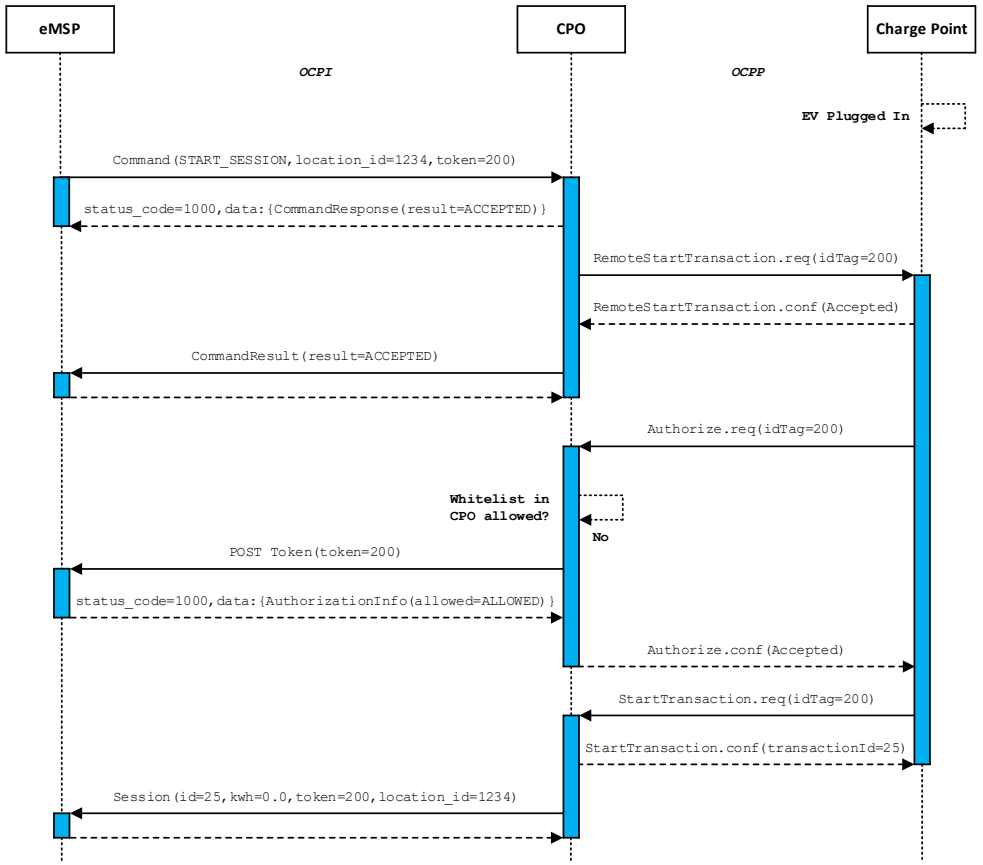

Fig. 4. OCPI Command START_SESSION with 'token.whitelist=NEVER'

2.2.3. OCPI : Stop Transaction

The stop transaction event begins with the eMSP sending a Command() module which contains the STOP_SESSION command to the CPO as shown in Figure 5. The Command() module also includes other parameters, such as the session_id. This data is to be assigned to parameter transactionId of the RemoteStopTransaction.req() which is sent by the CPO to the Charge Point to stop the charging session.

The Charge Point can use the transaction id as a reference to obtain the token that request a charging session to stop. Generally, this event requires authorization as well, and the process depends on the value of 'token.whitelist'. If 'token.whitelist=NEVER', then the token must be authorized real-time, not white-listed. In this case, the CPO can request the eMSP to execute the authorization and send the result to the CPO. Whereas if 'token.whitelist=ALWAYS', then the CPO has to do the authorization based on the whitelist of valid user tokens it has, as shown in Fig. 5. After that, the transaction can be proceed as defined in OCPP.

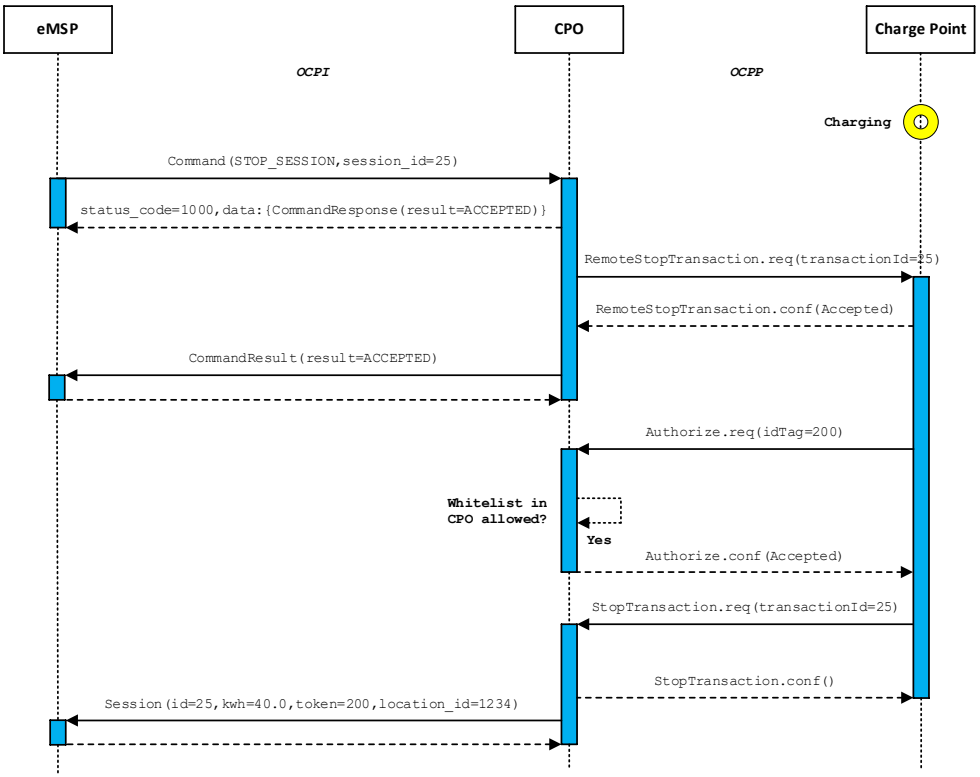

Fig. 5. OCPI Command STOP_SESSION with 'token.whitelist=ALWAYS'

2.2.4. OCPI : Billing

After a charging session has ended, the CPO can prepare a Charge Detail Record (CDR) to be shared with the eMSP. The CDR contains the description of the concluded charging session. The CDR is the only object that relevant to billing. Although there is no requirement to share CDRs in real-time, it is considered a good practice to do it as soon as possible. OCPI allows both parties to determine the time for sharing the CDR based on an agreement [2].

According to OCPI, there are two ways to share CDRs with eMSPs:

1. The eMSP determines an end-point with the following format:

   ```
   {endpoint_url}/?[date_from={date_from}]&[date_to={date_to}]&[offset={o
   ffset}]&[limit={limit}]
   ```

   , for example:

   ```
   https://www.server.com/ocpi/2.2/cdrs/
   ```

   , so that the CPO can upload CDRs to the eMSP as shown in Figure 6 with a HTTP POST request as follows:

   ```
   POST To URL: https://www.server.com/ocpi/2.2/cdrs/
   {
     "country_code": "BE",
     "party_id": "BEC",
     "id": "12345",
     "cdr_location": {
   ```

```
    "id": "LOC1",
    ...
  },
  "currency": "EUR",
  "charging_periods": [{
    "start_date_time": "2015-06-29T21:39:09Z",
    "dimensions": [{
      "type": "TIME",
      "volume": 1.973
    }],
    "tariff_id": "12"
  }],
  "total_energy": 15.342,
  "total_time_cost": {
    "excl_vat": 4.00,
    "incl_vat": 4.40
  },
  ...
}
```


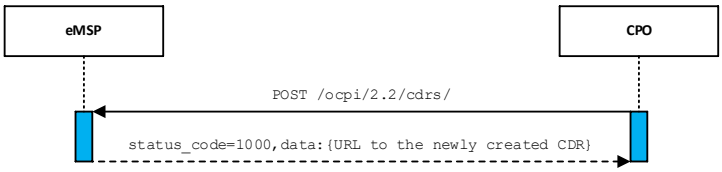

Fig. 6. OCPI HTTP POST CDR

2. The eMSP determines an end-point, so that the CPO can download CDRs as shown in Figure 7 with a HTTP GET as follows:

```
GET From URL: https://www.server.com/ocpi/2.2/cdrs/?offset=50
```

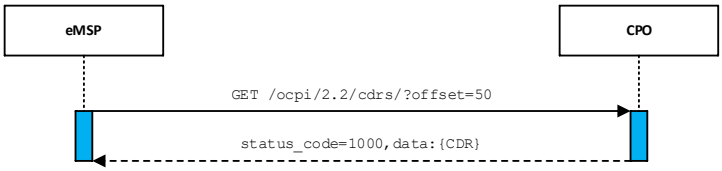

Fig. 7. OCPI HTTP GET CDR

2.3. Open InterCharge Protocol (OICP)

OICP is an EV roaming protocol that enables communication between MSPs and CPOs via the Hubject Brokering System [5], [6]. Hubject's stakeholders are all German companies including BMW Group, Daimler, Volkswagen, Siemens, Bosch and others. It is publicly available at no cost and without registration. OICP is publised under the Creative Commons Attribution-ShareAlike 4.0 International License. The latest version to date is OICP 2.3.

OICP is based on SOAP and JSON/RESTful API (select one) with HTTP as the basis communication protocol. OICP is a real-time protocol and supports synchronous as well as asynchronous operations. Actors and roles that are defined in OICP and relevant to the main objective of this paper are as follows:

- The **Hubject Brokering System (HBS)** facilitates communication between EMPs and CPOs.

- The **Emobility Service Provider (EMP)** provides services for charging electric vehicles (a role of an MSP as mentioned elsewhere in this paper). EMPs connect to CPOs via HBS to enable roaming services.
- The **Charge Point Operator (CPO)** operates EV charging stations (a role of a CPO as mentioned elsewhere in this paper). CPOs connect to EMPs via HBS to enable roaming services.
- The **Electric Vehicle Supply Equipment (EVSE)** delivers energy to electric vehicles.
- The **EV user** directly connects to a charging station or through an EMP.

OICP also supports the following features:

- **Roaming via hub:** The backend system of the EMP and the backend system of the CPO connect to the HBS e-roaming platform. Both parties (the EMP and the CPO) have a roaming contract via the HBS.
- **Authorization:** OICP has a procedure to share tokens data from EMPs with HBS. Authorization can be done real-time (the EMP is online and online authorization is possible) or via a whitelist (the EMP is offline).
- **Billing:** OICP supports sending Charge Detail Records (CDRs) via the HBS for the purpose of invoicing.
- **Remote start/stop:** EMPs can start or stop a charging session.

2.3.1. OICP : User Registration

EV users are identified by tokens (the authentication data). Tokens may be in the form of RFID tags or smart cards or others that are provided by the EMP at a user registration process. If the EMP does not have real-time connection for authorization to the HBS (offline EMP), then the EMP must upload the tokens data to the HBS as shown in Figure 8 with a HTTP POST request as follows:

```
POST To URL:
https://service.hubject.com/api/oicp/authdata/v21/providers/{providerID}/push
-request
{
  "ActionType": "fullLoad",
  "ProviderAuthenticationData": {
    "ProviderID": "DE-DCB"
    "AuthenticationDataRecord": [{
      "Identification": {
        "RFIDMifareFamilyIdentification": {
          "UID": "string"
        },
        "QRCodeIdentification": {
        },
        "PlugAndChargeIdentification": {
        },
        "RemoteIdentification": {
        },
      }
    }],
  }
}
```

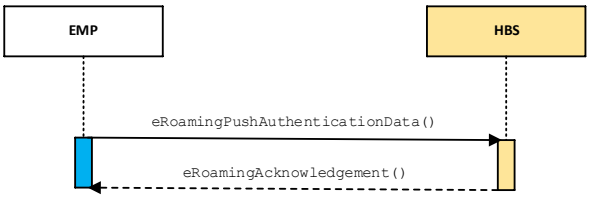

Fig. 8. OICP sharing tokens data

2.3.2. OICP : Start Transaction

This event can start at the EVSE. The CPO does not recognize the authentication data, so the CPO can request the HBS to authorize the token and send the result to the CPO. If the EMP has a real-time connection for authorization to the HBS (online EMP), the HBS can forward the authorization task to the EMP as shown in Figure 9. While if the EMP has no real-time connection for authorization to the HBS (offline EMP), the HBS can do the authorization based on the authentication data it has. After that, the transaction can be proceed as defined in OCPP. Note that the token data in parameter idTag of the Authorize.req() from the EVSE to the CPO is to be assigned to the corresponding parameter of the eRoamingAuthorizeStart() which is sent by the CPO to the HBS to authorize the EV user who wants to start charging.

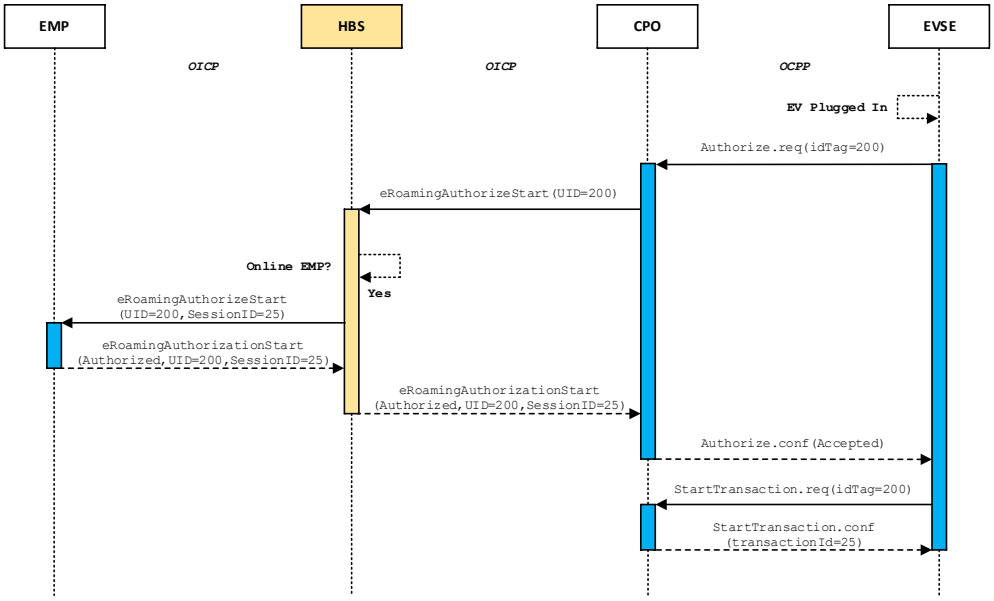

Fig. 9. OICP start transaction at the EVSE (online EMP)

For start transaction that begins at the EMP (for example via mobile apps), the EMP can send the eRoamingAuthorizeRemoteStart() to the HBS which contains data of the token that triggers the transaction (e.g. UID). The HBS should then forward the message to the CPO with the additional parameter SessionID. The token data is to be assigned to parameter idTag of the RemoteStartTransaction.req() sent by the CPO to the EVSE in order to start a charging session, and parameter SessionID is referred by parameter transactionId of the StartTransaction.conf() as shown in Figure 10.

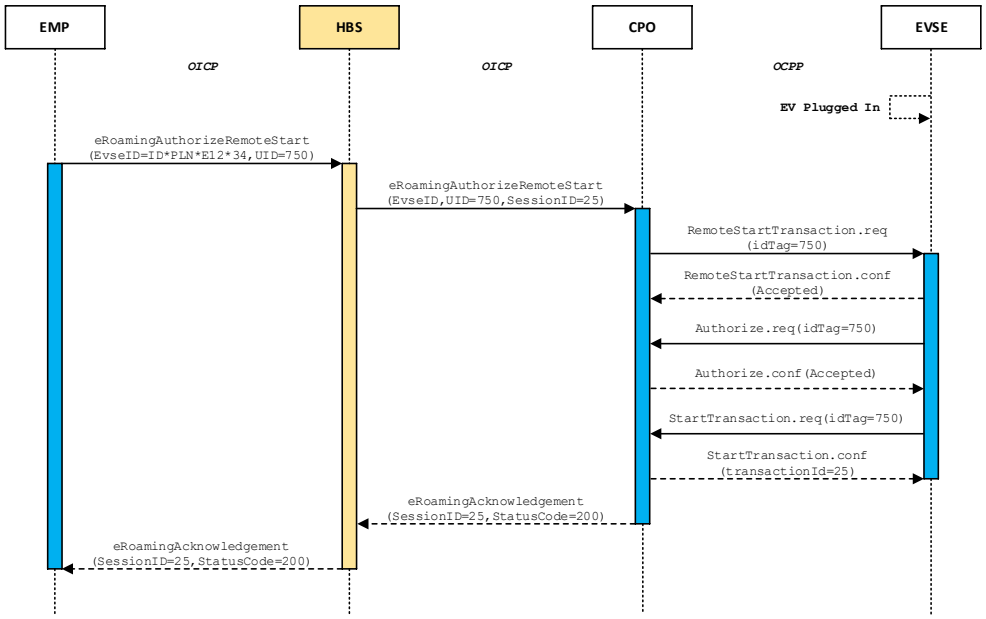

Fig. 10. OICP start transaction at the EMP

2.3.3. OICP : Stop Transaction

This event is similar to the start transaction. The HBS can forward the authorization task to the EMP (online EMP) or do the authorization based on the authentication data it has (offline EMP) as shown in Figure 11. After that, the transaction can be proceed as defined in OCPP. Note that the token data in parameter idTag of the Authorize.req() from the EVSE to the CPO is to be assigned to the corresponding parameter of the eRoamingAuthorizeStop() which is sent by the CPO to the HBS to authorize the EV user who wants to stop charging.

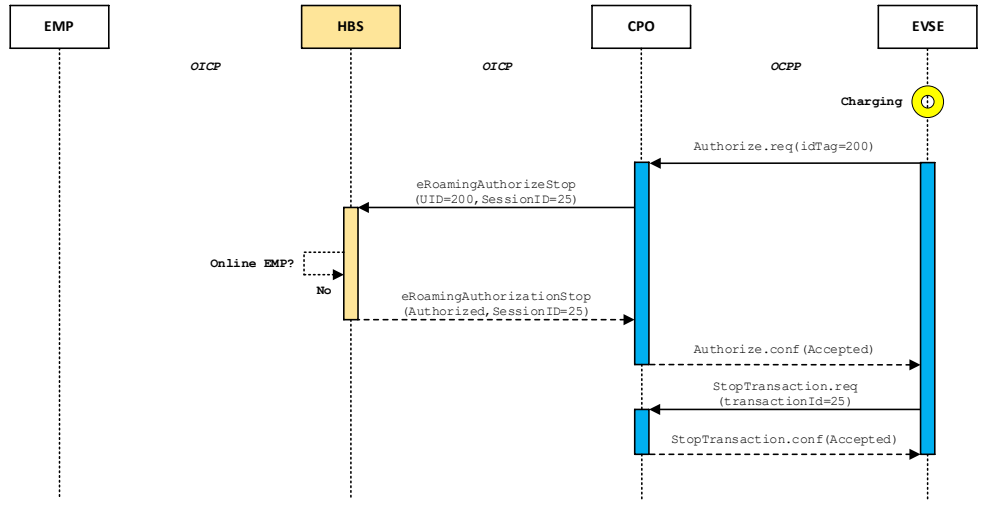

Fig. 11. OICP stop transaction at the EVSE (offline EMP)

For stop transaction that begins at the EMP (for example via mobile apps), the EMP can send the eRoamingAuthorizeRemoteStop() to the HBS which contains information of the session id that was generated by the HBS after the initial eRoamingAuthorizeRemoteStart()

message. The HBS should then forward the request to the CPO. After that, the transaction can be proceed as defined in OCPP. The session id is to be assigned to parameter transactionId of the RemoteStopTransaction.req() which is sent by the CPO to the EVSE in order to stop the charging session as shown in Figure 12.

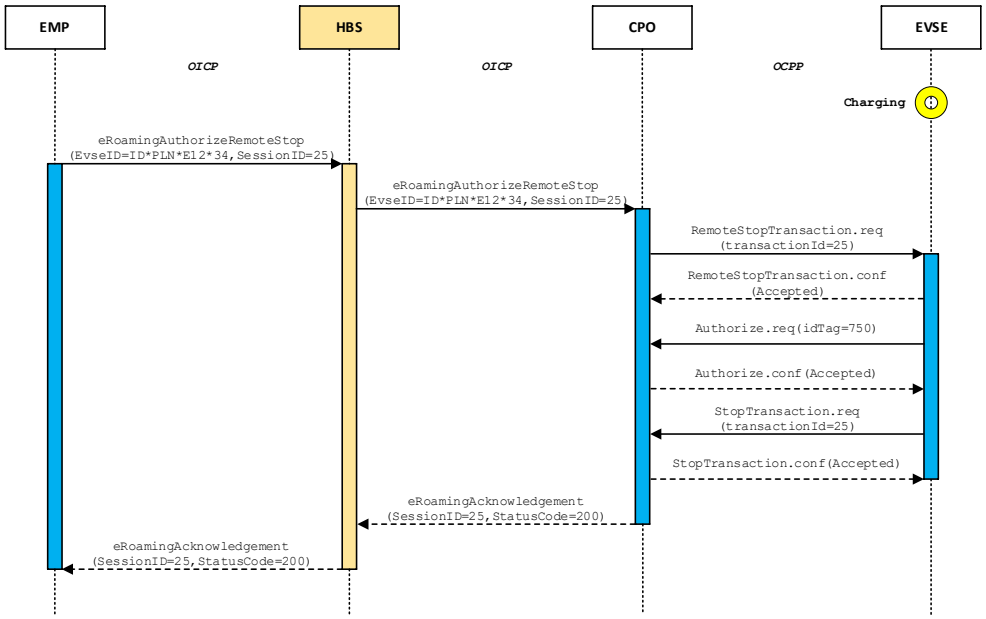

Fig. 12. OICP stop transaction at the EMP

2.3.4. OICP : Billing

After the end of a charging session, the CPO can prepare a Charge Detail Record (CDR) to be shared with the EMP. The CDR contains the description of the concluded charging session and is the only object that relevant to billing. Although there is no requirement to share CDRs in real-time, it is considered a good practice to do it as soon as possible. OICP allows both parties to determine the time for sharing the CDR based on an agreement [2].

According to OICP, there are two ways to share CDRs with EMPs:

1. The CPO sends the CDR to the HBS with a HTTP POST request as follows:

```
POST To URL:
https://service.hubject.com/api/oicp/cdrmgmt/v22/operators/{operatorID
}/charge-detail-record
{
  "SessionID": "string",
  "ConsumedEnergy": 10,
  "Identification": {
    "RFIDMifareFamilyIdentification": {
      "UID": "string"
    },
    ...
  },
  ...
}
```

, so that the HBS can forward the CDR to the EMP as shown in Figure 13.

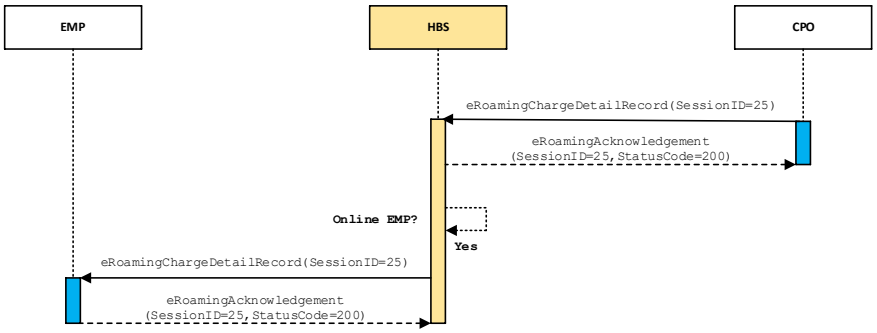

Fig. 13. OICP sharing CDRs (online EMP)

2. The CPO uploads the CDR to the HBS with the HTTP POST request as described in number (1), so that the EMP can download the CDR from the HBS as shown in Figure 14 with a HTTP POST request as follows:

```
POST To URL:
https://service.hubject.com/api/oicp/cdrmgmt/v22/providers/{providerID
}/get-charge-detail-records-request
{
  "CDRForwarded": false,
  "From": "2020-08-23T14:20:10.285Z",
  "OperatorID": "DE*ABC",
  "ProviderID": "DE-DCN",
  "To": "2020-09-23T14:20:10.285Z",
  "SessionID": [
    "string"
  ]
}
```

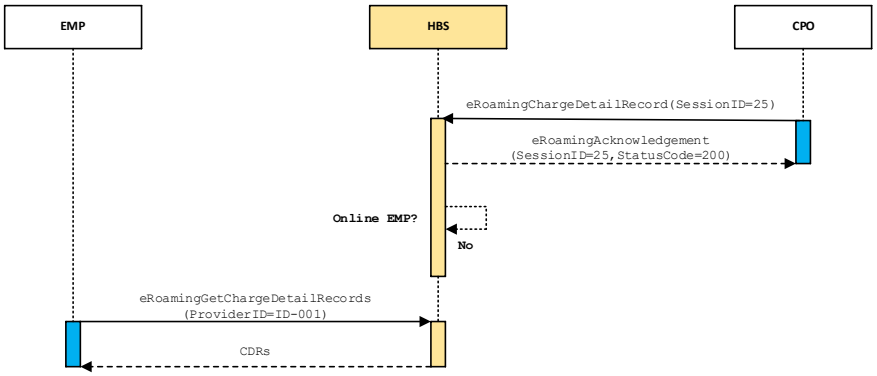

Fig. 14. OICP sharing CDRs (offline EMP)

2.4. Open Clearing House Protocol (OCHP)

OCHP is an EV roaming protocol that enables communication between MSPs and CPOs via a roaming hub [7]. OCHP is develop by Smartlab and ElaadNL which are organizations from Germany and the Netherlands respectively. Up to date, OCHP is used by e-clearing.net, a roaming hub which is operated by Smartlab and owned by Smartlab and ElaadNL. There is also an extension named OCHP-direct that supports for peer-to-peer communication, but this paper does not cover that extension.

OCHP is based on SOAP with HTTP as the basis communication protocol. It relies on asynchronous operations. OCHP is publicly available without registration, and is published under the MIT License which allows for free distribution and modification. The latest version to date is OCHP 1.4.

OCHP does not authorize EV users in real-time using data present at the MSP, but instead authorizes EV users using the whitelist of valid users. Actors and roles that are defined in OCHP and relevant to the main objective of this paper are as follows:

- The **Clearing House (CH)** runs a software platform named Clearing House to enable data exchange between EVSPs and EVSE Operators.
- The **Electric Vehicle Service Provider (EVSP)** provides services for charging electric vehicles (a role of an MSP as mentioned elsewhere in this paper). EVSPs connect to EVSE Operators via CH to enable roaming services.
- The **Electric Vehicle Supply Equipment Operator (EVSE Operator)** operates EV charging stations (a role of a CPO as mentioned elsewhere in this paper). EVSE Operators connect to EVSPs via CH to provide roaming services.
- The **Electric Vehicle Supply Equipment (EVSE)** delivers energy to electric vehicles.
- The **EV user** directly connects to a charging station.

OCHP also supports the following features:

- **Roaming via hub:** Roaming hub runs a software platform to connect the relevant parties. Up to date, OCHP is used by a roaming hub named e-clearing.net. However, the OCHP documentation does not specifically state that e-clearing.net should be the only one.
- **Roaming peer-to-peer:** Direct connection scheme via the CH (only if OCHP-direct apply).
- **Authorization:** There is a procedure to share tokens data from EVSPs with EVSE Operators. Authorization is conducted via a whitelist.
- **Billing:** OCHP supports sending Charge Detail Records (CDRs) via the CH for the purpose of invoicing.
- **Remote start/stop:** EVSPs can start or stop a charging session (only if OCHP-direct apply).

2.4.1. OCHP : User Registration

Tokens are used to authorize EV users. Tokens may be in the form of RFID tags or smart cards or others that are provided by the EVSP at a user registration process. The Master Data Management System (MDM) of the EVSP must upload the whitelist of valid user tokens (the roaming authorization list) to the Clearing House System (CHS) of CH as shown in Figure 15 with a HTTP request which contains the following XML elements:

```
<s11:Body>
  <ns1:SetRoamingAuthorisationListRequest xmlns:ns1='http://ochp.eu/1.4'>
    <ns1:roamingAuthorisationInfoArray>
      <ns1:EmtId representation='?XXX?'>
        <ns1:instance>???</ns1:instance>
        <ns1:tokenType>???</ns1:tokenType>
        <ns1:tokenSubType>???</ns1:tokenSubType>
      </ns1:EmtId>
      ...
    </ns1:roamingAuthorisationInfoArray>
  </ns1:SetRoamingAuthorisationListRequest>
</s11:Body>
```

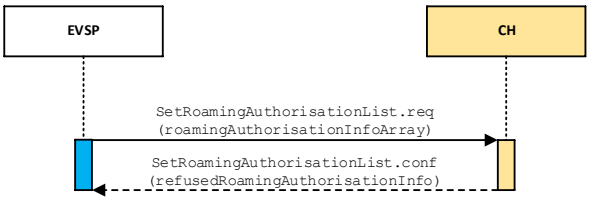

Fig. 15. OCHP uploading tokens data

,and then the Charge Point Management System (CMS) of the EVSE Operator can download the tokens data from the CH as shown in Figure 16 with a HTTP request which contains the following XML elements:

```
<s11:Body>
  <ns1:GetRoamingAuthorisationListRequest xmlns:ns1='http://ochp.eu/1.4' />
</s11:Body>
```

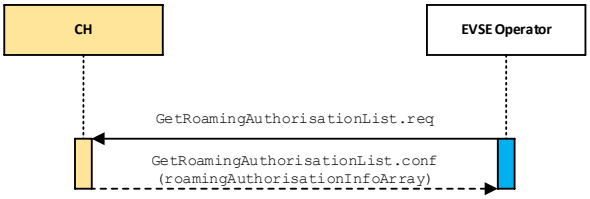

Fig. 16. OCHP downloading tokens data

In case that the roaming authorization list needs to be updated, the EVSP can send the UpdateRoamingAuthorisationList.req() to the CH which contains the authorization records to update. Then, the EVSE Operator can send the GetRoamingAuthorisationListUpdates.req() to the CH which contains information that specifies the last time the authorization records was changed in the request to download the updated version.

2.4.2. OCHP : Start Transaction

This event can start at the EVSE. To authorize an EV user, if the authorization data in the roaming authorization list managed by the EVSE Operator is not up-to-date, then the EVSE Operator can request the CH to authorize the token that triggers the transaction and send the result to the EVSE Operator as shown in Figure 17. The token data in parameter idTag of the Authorize.req() from the EVSE to the EVSE Operator is to be assigned to parameter emtId of the RequestSingleRoamingAuthorisation.req() which is sent by the EVSE Operator to the CH. While if the authorization data is up-to-date, then the EVSE Operator can do the authorization based on the authorization data it has, as shown in Figure 18. After that, the transaction can be proceed as defined in OCPP.

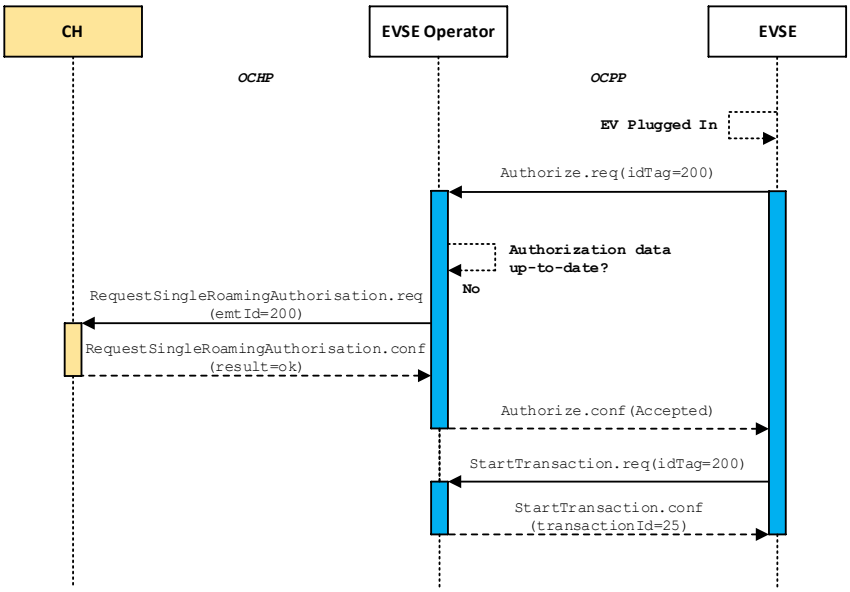

Fig. 17. OCHP start transaction at the EVSE (the authorization data is not up-to-date)

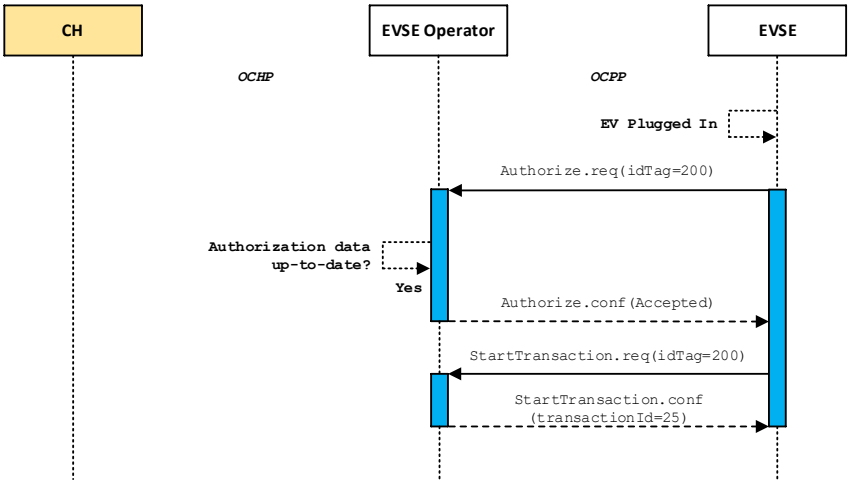

Fig. 18. OCHP start transaction at the EVSE (the authorization data is up-to-date)

2.4.3. OCHP : Stop Transaction

This event can start at the EVSE. The EVSE Operator can directly authorize the same token that triggered the charging session at the start transaction without having to request to the CH to do so, as shown in Figure 19. The required steps are conducted as defined in OCPP.

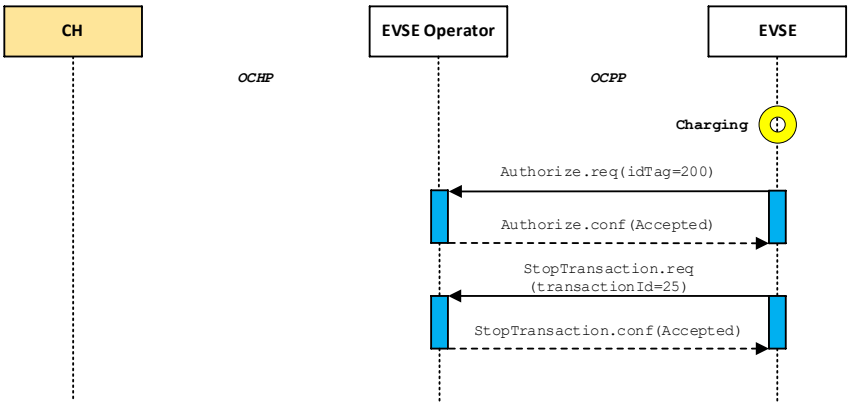

Fig. 19. OCHP stop transaction at the EVSE

2.4.4. OCHP : Billing

After a charging session is over, the EVSE Operator can prepare a Charge Detail Record (CDR) to be shared with the EVSP. The CDR contains the description of the concluded charging session. The CDR is the only object that relevant to billing. Although there is no requirement to share CDRs in real-time, it is considered a good practice to do it as soon as possible. In addition, OCHP allows both parties to determine the time for sharing the CDR based on an agreement [2].

The EVSE Operator (also known as the CDR-"Originator") can upload CDRs to the CH as shown in Figure 20 with a HTTP request which contains the following XML elements:

```
<s11:Body>
  <ns1:AddCDRsRequest xmlns:ns1='http://ochp.eu/1.4'>
    <ns1:cdrInfoArray>
      <ns1:CdrId>???</ns1:CdrId>
      <ns1:evseId>???</ns1:evseId>
      <ns1:emtId representation='?XXX?'>
        <ns1:instance>???</ns1:instance>
        <ns1:tokenType>???</ns1:tokenType>
        <ns1:tokenSubType>???</ns1:tokenSubType>
      </ns1:emtId>
      <ns1:totalCost>?999.99?</ns1:totalCost>
      <ns1:currency>???</ns1:currency>
      ...
    </ns1:cdrInfoArray>
  </ns1:AddCDRsRequest>
</s11:Body>
```

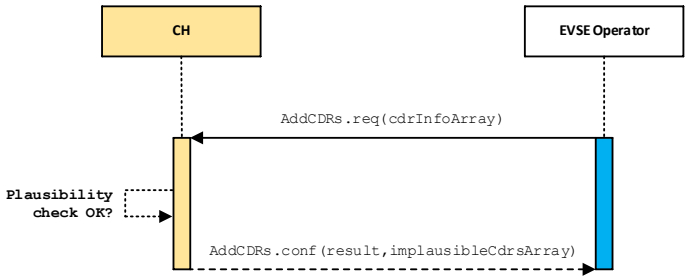

Fig. 20. OCHP uploading CDRs

Upon receiving the CDR, the CH will perform a basic plausibility check to determine whether the CDR can be accepted. If not, then the CDR will be sent back to the EVSE Operator using the CheckCDRs.req() for revision and resend to the CH. Plausible CDRs will be marked as 'accepted' by the CH and ready to be downloaded by the EVSP (also known as the CDR-"Owner") in order to get approval from the EVSP.

The EVSP can download all relevant CDRs from the CH using the GetCDRs() as shown in Figure 21. Following an internal validation check, the EVSP will give the status 'approved' to the approved CDRs. Whereas the rejected CDRs will be given the status 'declined'. This information will be sent to the CH using the ConfirmCDRs.req(). Further, the approved CDRs will not be included in the response to the GetCDRs() message from the EVSP.

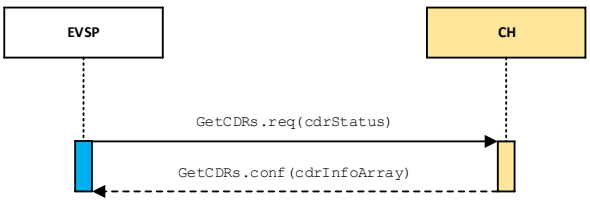

Fig. 21. OCHP downloading CDRs

2.5. eMobility Inter-operation Protocol (eMIP)

eMIP is an EV roaming protocol that enables communication between MSPs and CPOs via a roaming hub [8]. The eMIP specifications were designed and managed by GIREVE which was founded by EDF, Renault, CNR and Caisse des Dépôts, all French companies. GIREVE is the only roaming hub available, and to connect to the GIREVE platform requires a certain certification.

eMIP is publicly available, but requires registration. It is based on SOAP with HTTP as the basis communication protocol. Although it was designed as a real-time protocol, eMIP also supports asynchronous operations. The latest version to date is eMIP 1.0.14.

Actors and roles that are defined in eMIP and relevant to the main objective of this paper are as follows:

- The **GIREVE** runs the GIREVE's eMobility Services Platform (the GIREVE's IoP) to provide technical and functional means to intermediate services between eMSPs and CPOs.
- The **eMobility Services Provider (eMSP)** provides services for charging electric vehicles (a role of an MSP as mentioned elsewhere in this paper). eMSPs have a valid "Subscription contract to GIREVE's Platform" in order to connect to CPOs via GIREVE to enable roaming services.
- The **Charge Point Operator (CPO)** operates EV charging stations (a role of a CPO as mentioned elsewhere in this paper). CPOs have a valid "Subscription contract to GIREVE's Platform" in order to connect to eMSPs via GIREVE to provide roaming services.
- The **Charge Point (CP)** delivers energy to electric vehicles.
- The **EV user** directly connects to a charging station or through an eMSP.

eMIP also supports the following features:

- **Roaming via hub:** eMSPs and CPOs are connected to the GIREVE platform based on web services. The eMIP documentation is explicitly states GIREVE as the roaming hub.

- **Authorization:** There are 3 authorization modes available, namely the Synchronous Authorisation (the main and nominal process), the Asynchronous Authorisation, and the Asynchronous Authentication Data Exchange & Synchronous Authorisation (the Mixed mode).
- **Billing:** eMIP supports sending Charge Detail Records (CDRs) via the GIREVE for invoicing. Asynchronous Exchange scenario enables the eMSP to download a list of CDR that has been sent by the CPO to the GIREVE. While Synchronous Exchange enables the GIREVE to forward the CDR to the eMSP.
- **Remote start/stop:** eMSPs can start or stop a charging session.

2.5.1. eMIP : User Registration

Tokens are used to authorize EV users. Tokens may be in the form of RFID tags or smart cards or others that are provided by the eMSP at a user registration process. After that, the eMSP must upload the whitelist of valid user tokens to the GIREVE as shown in Figure 22 with a HTTP POST request which contains XML elements as follows:

```
<soap:Body xmlns:m="https://api-iop.gireve.com/schemas/AuthorisationV1/">
  <m:eMIP_ToIOP_SetAuthenticationDataRequest>
    <transactionId>TRANSACTION_46151</transactionId>
    <partnerId>FR*MSP</partnerId>
    <operatorId>FR*798</operatorId>
    <reqAuthenticationDataList>
      <reqAuthenticationData>
        <actionType>1</actionType>
        <userIdType>RFID-UID</userIdType>
        <userId>1234567890ABCD</userId>
        ...
      </reqAuthenticationData>
      <reqAuthenticationData>
        <actionType>2</actionType>
        ...
      </reqAuthenticationData>
      ...
    </reqAuthenticationDataList>
    ...
  </m:eMIP_ToIOP_SetAuthenticationDataRequest>
</soap:Body>
```

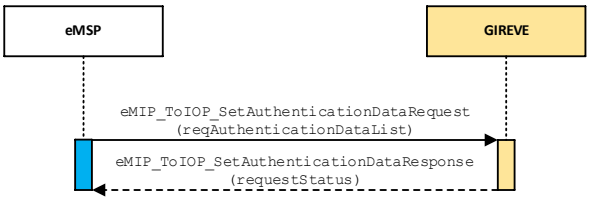

Fig. 22. eMIP uploading tokens data

,and then the CPO can download the tokens data from the GIREVE as shown in Figure 23 with a HTTP POST request which contains XML elements as follows:

```
<soap:Body xmlns:m="https://api-iop.gireve.com/schemas/AuthorisationV1/">
  <m:eMIP_ToIOP_GetAuthenticationDataRequest>
    <transactionId>TRANSACTION_46151</transactionId>
    <partnerId>FR*CPO</partnerId>
    <operatorId>FR*489</operatorId>
    <dateTimeFrom>2015-10-26T21:32:52</dateTimeFrom>
    ...
  </m:eMIP_ToIOP_GetAuthenticationDataRequest>
```

```
</soap:Body>
```

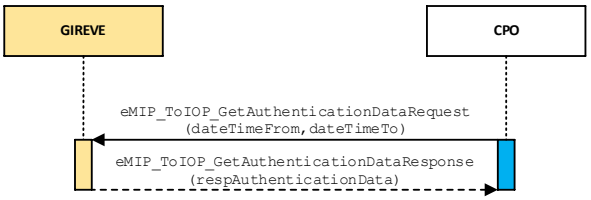

Fig. 23. eMIP downloading tokens data

2.5.2. eMIP : Start Transaction

This event can start at the CP. If the Synchronous Authorisation mode apply, the CPO can request the eMSP via the GIREVE to authorize the token that triggers the transaction and send the result to the CPO, as shown in Figure 24. Whereas if the Asynchronous Authorisation mode apply, the CPO can do the authorization based on the whitelist of valid user tokens it has, as shown in Figure 25. As for the Mixed mode, the CPO can request the GIREVE to do the authorization based on the whitelist of valid user tokens it has and send the result to the CPO, as shown in Figure 26. After that, the transaction can be proceed as defined in OCPP. Note that for the Synchronous Authorisation mode and the Mixed mode, the token data in parameter idTag of the Authorize.req() which is sent by the CP to the CPO is to be assigned to parameter userId of the eMIP_ToIOP_GetServiceAuthorisationRequest().

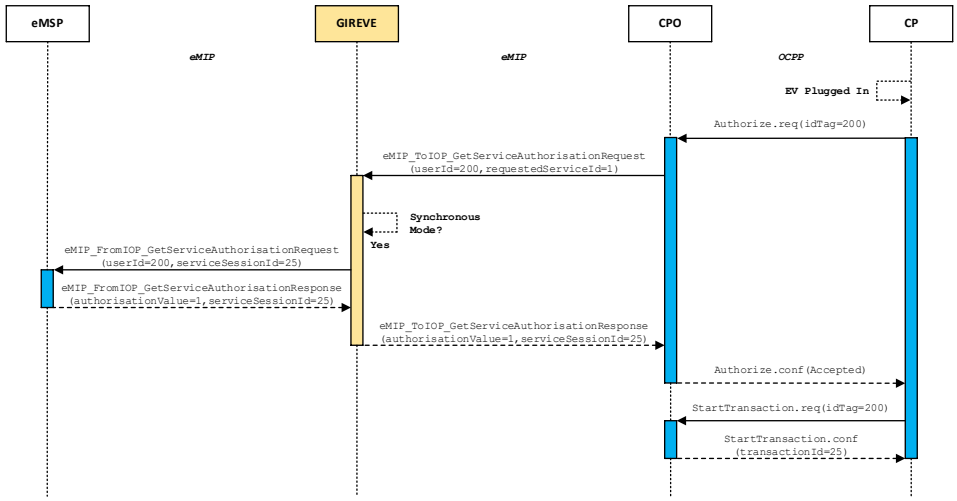

Fig. 24. eMIP start transaction at the CP (the Synchronous Authorisation mode)

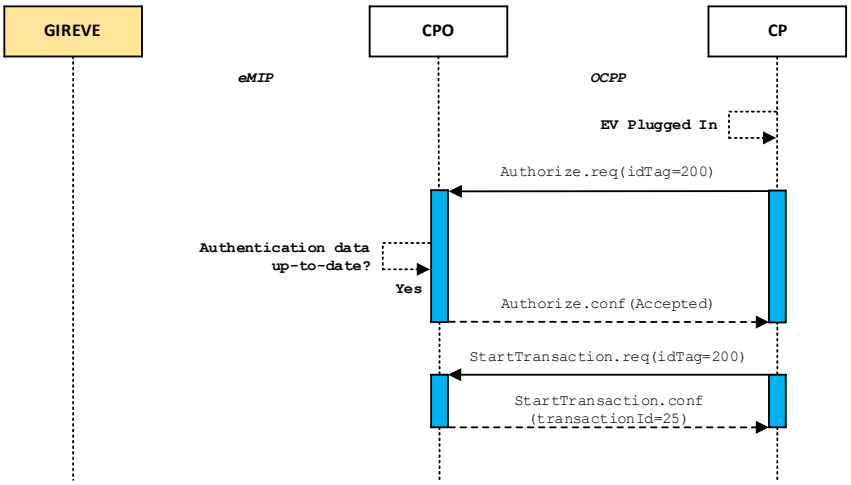

Fig. 25. eMIP start transaction at the CP (the Asynchronous Authorisation mode)

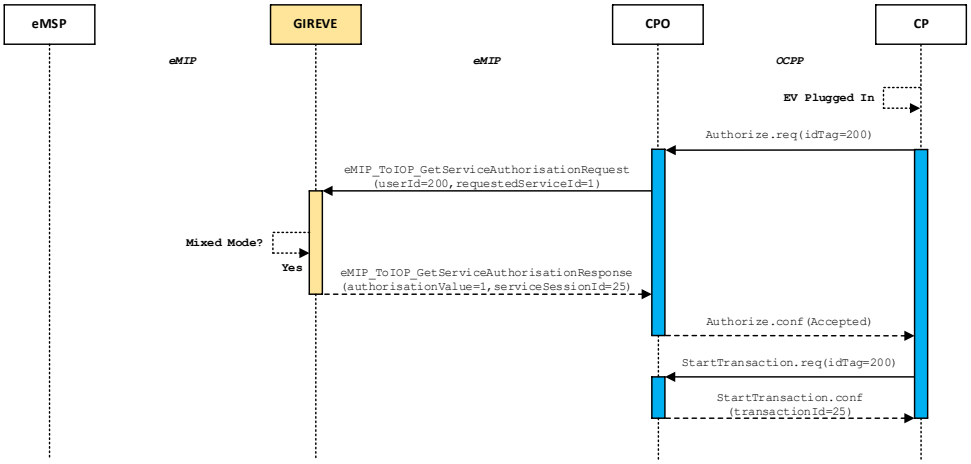

Fig. 26. eMIP start transaction at the CP (the Mixed mode)

For start transaction that begins at the eMSP (for example via mobile apps), the eMSP can send the eMIP_ToIOP_SetServiceAuthorisationRequest() to the GIREVE which contains data of the token that triggers the transaction along with parameter authorisationValue which is set to 1. The GIREVE should then forward the message to the CPO with the additional parameter serviceSessionId. After that, the transaction can be proceed as defined in OCPP. The token data is to be assigned to parameter idTag of the RemoteStartTransaction.req() which is sent by the CPO to the CP to request the start of a charging session, and parameter serviceSessionId becomes a reference for parameter transactionId of the StartTransaction.conf() as shown in Figure 27.

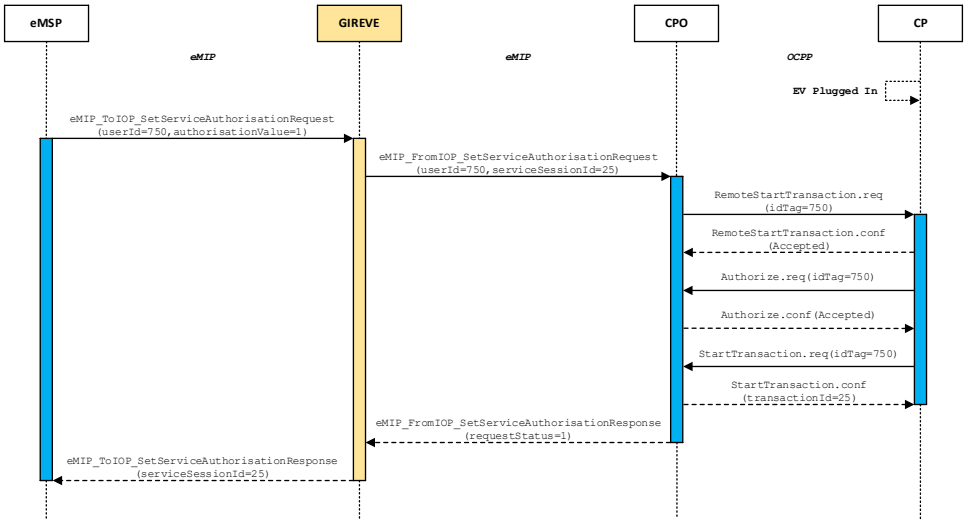

Fig. 27. eMIP start transaction at the eMSP

2.5.3. eMIP : Stop Transaction

This event can start at the CP with a similar procedure to the start transaction. After the authorization of an EV user who wants to stop charging, the transaction can be proceed as defined in OCPP. However, this time the StopTransaction.req() message takes part. There are 3 authorization modes: the Synchronous Authorisation, the Asynchronous Authorisation, and the Mixed mode. This selection must match the authorization at the start transaction.

For stop transaction that begins at the eMSP (for example via mobile apps), the eMSP can send the eMIP_ToIOP_SetServiceAuthorisationRequest() to the GIREVE which contains data of the token that triggers the transaction along with parameter authorisationValue which is set to 2. The GIREVE should then forward the message to the CPO with the additional parameter serviceSessionId. After that, the transaction can be proceed as defined in OCPP. The session id is to be assigned to parameter transactionId of the RemoteStopTransaction.req() which is sent by the CPO to the CP in order to stop the charging session as shown in Figure 28.

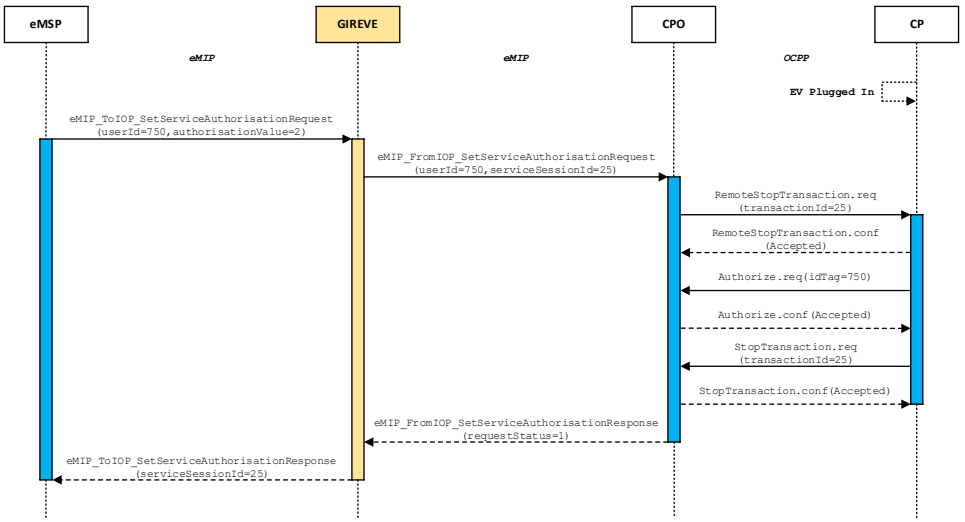

Fig. 28. eMIP stop transaction at the eMSP

2.5.4. eMIP : Billing

After the end of a charging session, the CPO can prepare a Charge Detail Record (CDR) to be shared with the eMSP. The CDR contains the description of the concluded charging session and is the only object that relevant to billing. Although there is no requirement to share CDRs in real-time, it is considered a good practice to do it as soon as possible. eMIP allows both parties to determine the time for sharing the CDR based on an agreement [2].

According to eMIP, there are two ways to share CDRs with eMSPs:

1. The CPO sends the CDR to the GIREVE with a HTTP POST request which contains XML elements as follows:

```
<soap:Body xmlns:m="https://api-
iop.gireve.com/schemas/AuthorisationV1/">
  <m:eMIP_ToIOP_SetChargeDetailRecordRequest>
    <transactionId>TRANSACTION_46151</transactionId>
    <partnerId>FR*CPO</partnerId>
    <operatorId>FR*489</operatorId>
    <chargeDetailRecord>
      <serviceSessionId>IOP-SID-GIR-V-IOPFT01</serviceSessionId>
      <meterReportList>
        <meterReport>
          <meterTypeId>1</meterTypeId>
          <meterValue>240</meterValue>
          <meterUnit>min</meterUnit>
        </meterReport>
        <meterReport>
          <meterTypeId>2</meterTypeId>
          ...
        </meterReport>
        ...
      </meterReportList>
      ...
    </chargeDetailRecord>
    ...
  </m:eMIP_ToIOP_SetChargeDetailRecordRequest>
</soap:Body>
```

, so that the GIREVE can forward the CDR to the eMSP as shown in Figure 29.

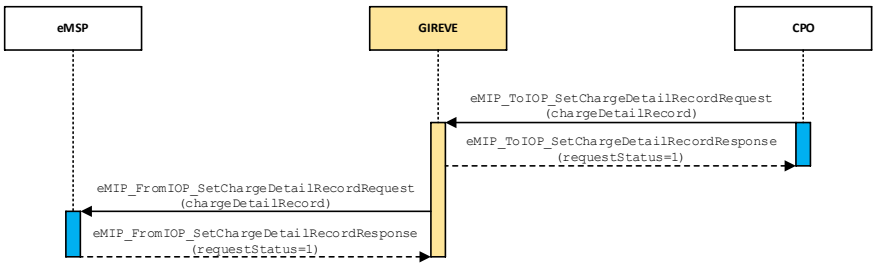

Fig. 29. eMIP sharing CDRs (Synchronous Exchange)

2. The CPO uploads the CDR to the GIREVE with the HTTP POST request as described in number (1), so that the eMSP can download the CDR from the GIREVE as shown in Figure 30 with a HTTP POST request which contains XML elements as follows:

```
<soap:Body xmlns:m="https://api-
iop.gireve.com/schemas/AuthorisationV1/">
  <m:eMIP_ToIOP_GetChargeDetailRecordRequest>
```

```
      <transactionId>TRANSACTION_46151</transactionId>
      <partnerId>FR*MSP</partnerId>
      <operatorId>FR*798</operatorId>
      <serviceSessionId>IOP-SID-GIR-V-IOPFT01</serviceSessionId>
      ...
    </m:eMIP_ToIOP_GetChargeDetailRecordRequest>
  </soap:Body>
```

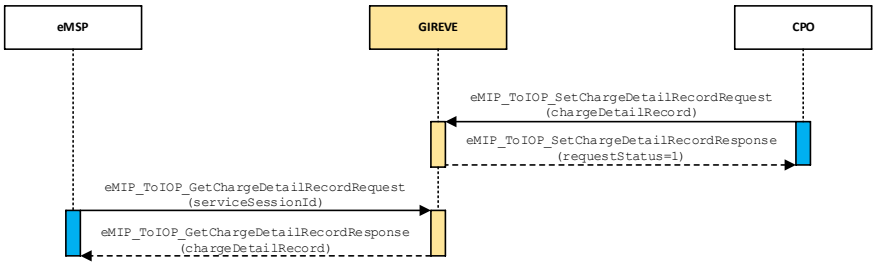

Fig. 30. eMIP sharing CDRs (Asynchronous Exchange)

3. Conclusion

In this paper, we provided an overview of the functionalities of the four major EV roaming protocols, namely the Open Charge Point Interface (OCPI), the Open InterCharge Protocol (OICP), the Open Clearing House Protocol (OCHP), and the eMobility Inter-operation Protocol (eMIP). We described how each protocol works in terms of data exchange with the Open Charge Point Protocol (OCPP) which commonly embedded in many charging stations nowadays. We confirmed that the EV roaming protocols define similar actors and roles, but use different terminology to call them. Interaction between actors is conducted via a roaming hub or a brokering system, even some of them also support peer-to-peer connectivity. Also, we included some examples in the form of a sequence diagram in order to depict the interaction between actors in the case of user registration, start a charging session, stop a charging session, and billing.

A transaction can start at the charging station as well as at the service provider (e.g. via mobile apps). This charging event always begins with the authorization of an EV user who wants to start or to stop charging based on valid user tokens. We find from some scripts on the user registration topic that various types of tokens can be provided by the service provider to be used for user authorization. These tokens include RFID tags, smart cards, QR codes, EV identities, and so forth. We understand that for transactions that start at the charging station, the tokens should be in the form of RFID tags or smart cards, since they are explicitely specified by the Open Charge Point Protocol (OCPP). While for transactions that start at the service provider, we realize that the tokens might be in any form that can be translated into alphanumeric. Besides tokens data, each transaction also exchanges charging information, and ended with sharing a Charge Detail Record (CDR) containing the description of the concluded charging session which can be used as the basis for invoicing.

We conclude that one effective method that can be used to achieve a fast learning curve in studying the EV roaming protocols is by employing a graphical representation to describe the roaming system, such as using a sequence diagram. The sequence diagram is relatively simple but is able to depict the interaction between actors from the service provider up to the charging station, thus gives a straightforward picture and clear understanding how the system works.

This research did not receive any specific grant from funding institutions whatsoever in the public, commercial, or not-for-profit sectors.

References

1. R. Ferwerda, M. Bayings, M. van der Kam and R. Bekkers, "Advancing E-roaming in Europe: Towards a single 'language' for the European charging infrastructure," in *World Electric Vehicle Journal*, *9*(4), (2018).
2. M. van der Kam and R. Bekkers, "Comparative analysis of standardized protocols for EV roaming - Report D6.1 for the evRoaming4EU project," (2020), retrieved 8th of July 2021 from https://evroaming.org/app/ uploads/2020/06/D6.1-Comparative-analysis-of-standardized-protocols-for-EV-roaming.pdf
3. Open Charge Alliance, "Open Charge Point Protocol 1.6," (2017).
4. NKL, "Open Charge Point Interface 2.2.1," (2021), retrieved 11th of March 2022 from https://evroaming.org/ app/uploads/2021/11/OCPI-2.2.1.pdf
5. Hubject, "Open InterCharge Protocol for Charge Point Operators 2.3," (2020), retrieved 15th of June 2022 from https://github.com/hubject/oicp/blob/master/OICP-2.3/OICP%202.3%20CPO/00_README_CPO.md
6. Hubject, "Open InterCharge Protocol for Emobility Service Providers 2.3," (2020), retrieved 15th of June 2022 from https://github.com/hubject/oicp/blob/master/OICP-2.3/OICP%202.3%20EMP/00_README_EMP.md
7. Smartlab and ElaadNL, "Open Clearing House Protocol 1.4," (2016), retrieved 8th of March 2022 from https://github.com/e-clearing-net/OCHP/blob/master/OCHP.md
8. GIREVE, "eMIP Protocol - Protocol Description 1.0.14," (2020), retrieved 11th of September 2021 from https:// www.gireve.com/wp-content/uploads/2020/12/Gireve_Tech_eMIP-V0.7.4_ProtocolDescription_ 1.0.14-en.pdf