OCPP & QR code payment in Japan

v1.0 June 2026

Table of Contents

OCA White Paper

Authors: Open Charge Alliance

Relevant for OCPP version: 2.x

1. Introduction

In Japan, mobile wallet-type payments using QR codes are rapidly spreading. They are commonly used in retail, hospitality, and service environments in the cities. The most widely used services for this are:

- PayPay
- Rakuten-Pay
- d-Barai
- au PAY
- LINE Pay

For public EV charging infrastructure, QR code payment offers a convenient way to enable ad hoc charging sessions without requiring RFID cards, proprietary charging apps, or physical payment terminals.

However, integrating these QR code payment methods into EV charging requires a different architectural approach than an open interbank payment rail. Although these methods allow payments funded by linked bank accounts, it operates as a managed wallet ecosystem and relies on API-generated dynamic payment requests rather than open static merchant QR codes.

This paper describes how this can be integrated into OCPP-compliant charging networks, focusing on dynamic QR and API-based flows. It also explains how OCPP supports prepaid and authorization-based payment models.

2. Integration models for QR code payments for EV charging

QR code payments commonly used in Japan have the following characteristics:

- Merchant-presented mode for QR payments, in which the merchant presents a (dynamic) QR code and the customer scans it
- Customer-presented mode for QR payments, in which the customer presents a dynamic QR code and the merchant scans it
- API integration with payment systems to generate dynamic QR codes with amounts and expiration dates
- Payment information is centrally managed by the payment provider, and transaction history can be checked on the user app.
- Authorization (reservation) and delayed capture
- Support for refunds of a remainder of a prepaid amount

Since EV charging stations are generally unmanned, and since charging stations do not have a QR code scanner, the customer-presented mode is not an option — there is no one to scan the customer QR code. EV charging therefore relies on the merchant-presented mode. There are three ways to display a QR code for payment:

1. Standard OCPP 2.1 QR code to initiate payment via CSMS webpage
2. Payment service provider’s QR code coming from CSMS for payment via app

3. Payment service provider's QR code coming from charging station for payment via app

The following sections describe these models in more detail.

2.1. Standard OCPP 2.1 QR code to initiate payment via CSMS webpage

This model uses the process described for an ad hoc payment via a QR code that is described in OCPP 2.1 as use case C25. The QR code that the charger displays and that the user will scan to initiate a payment, should be **a dynamic QR code as defined in the OCPP 2.1 specification**.

In this model:

1. The charger displays a dynamic QR containing a secure HTTPS URL.
2. The driver scans the QR using their phone camera.
3. The CSMS webpage opens, showing charger details and pricing.
4. The user selects the QR code payment app as payment method.
5. The CSMS creates a QR code payment request via the payment provider's API.
6. The webpage launches the QR code payment app with a pre-filled payment via a deeplink.
7. The user authorizes payment.
8. The CSMS receives webhook confirmation.
9. The CSMS starts the transaction.

Advantages:

- Allows presentation of pricing and terms
- Supports multiple payment methods
- Enables refund and capture management
- Simplifies compliance and audit logging

This model is especially suited for OCPP 2.1 (or higher) deployments.

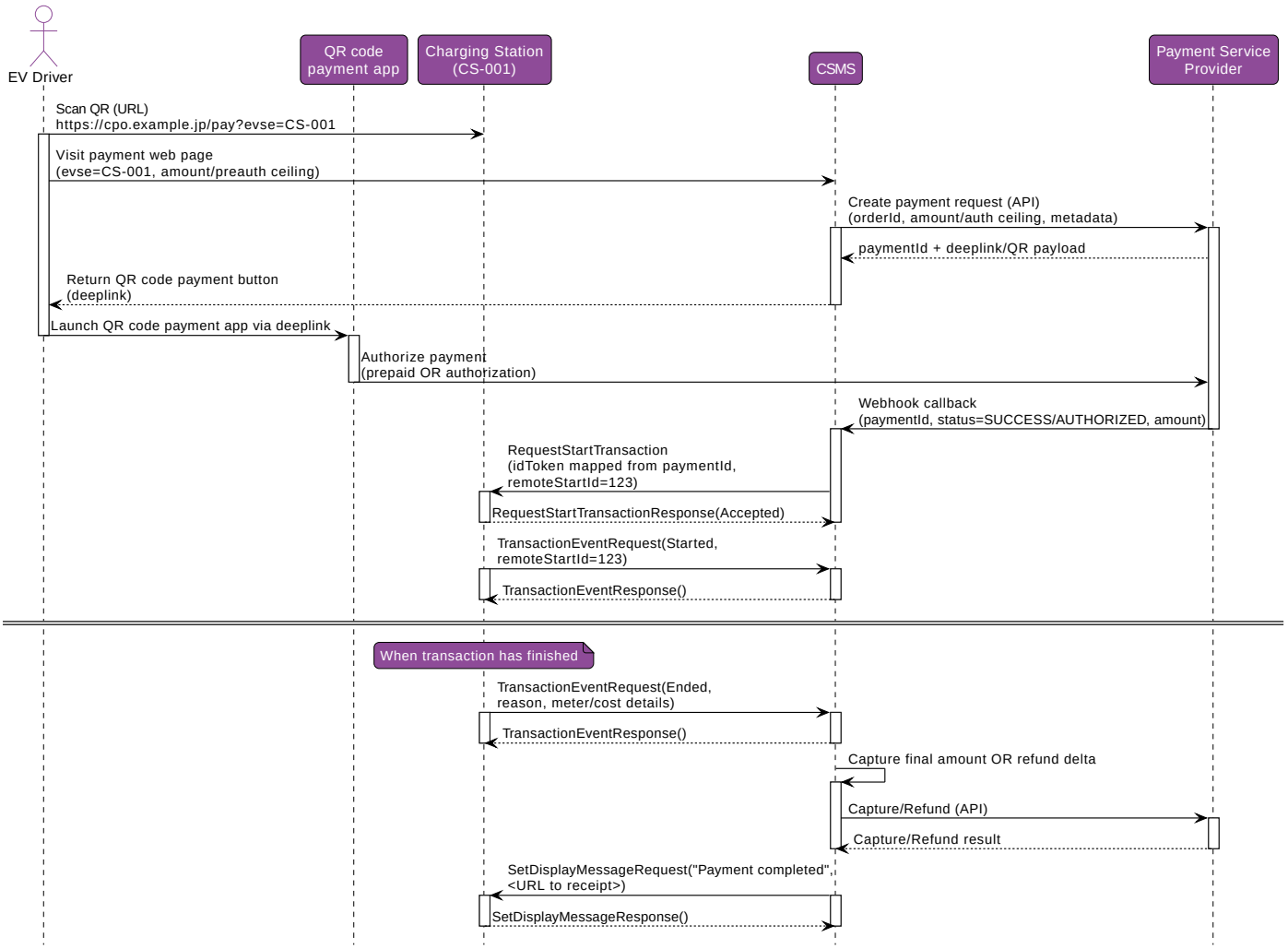

Figure 1. Sequence diagram for payment via CSMS webpage, initiated by scanning QR code on charging station

2.2. Payment service provider's QR code coming from CSMS for payment via app

This model is also similar to the ad hoc payment via a QR code that is described in OCPP 2.1 as use case C25, but it differs in the way how the dynamic QR code is generated.

In the standardized process in OCPP 2.1 the dynamic QR code is generated by the charging station according to an algorithm defined in the OCPP specification. This dynamic QR code changes fairly frequently, i.e. every minute or at least every hour.

On the other hand, for QR code payment services commonly used in Japan **the dynamic QR codes are generated through APIs** provided by payment providers. Such a QR code is generated only once when a payment transaction is requested, although it can be refreshed on demand.

In this model, when the user selects "Pay with QR code" on the display of the charging station it needs to trigger CSMS to start a payment process with the payment provider and request a QR code. Since there is no standard OCPP message that the charging station can use to inform CSMS, we need a customization for this in the form of a DataTransfer message. After receiving this trigger the CSMS calls the payment provider's API to generate a dynamic QR code. The generated dynamic QR code payload (including QR code display URLs and payment identification tokens) is sent via the OCPP SetDisplayMessageRequest(format= `QRCODE`), and the charging station shows the QR code on the display.

The format=`QRCODE` was introduced in OCPP 2.1. For a charging station running OCPP 2.0.1 the following approach needs to be taken to display the text from a DisplayMessageRequest as a QR code:

- The charging station has a device model component **QRCodeDisplay** with a variable Enabled = true.
- CSMS can add this component as the *display* parameter in a SetDisplayMessageRequest in case the *messageContent* needs to be rendered as a QR code.
- CSMS will only use this *display* parameter if the variable QRCodeDisplay.Enabled has been reported as true.

It is up to the implementation whether using the component QRCodeDisplay displays the QR code on the main screen of the charging station or on a dedicated QR code display.

In short:

1. The EV driver selects “Pay with QR code” on the charger.
2. The charging station sends message to CSMS to start a payment session.
3. The CSMS calls the payment provider’s API to create a payment request.
4. Payment provider returns a dynamic QR payload to use for the payment.
5. The CSMS sends this dynamic QR code payload via a SetDisplayMessageRequest to the charging station to be displayed.
6. The driver scans the QR using the QR code payment app.
7. The payment app processes the payment with the payment provider.
8. The CSMS receives real-time confirmation from the payment provider.
9. The CSMS instructs the charging station to start the charging session.

Characteristics:

- Fully automated payment confirmation
- Real-time webhook integration
- Payment linked to specific charger and EVSE
- Supports prepaid and authorization flows
- Secure against QR tampering through short-lived QR generation
- A custom DataTransfer message is required to initiate the payment

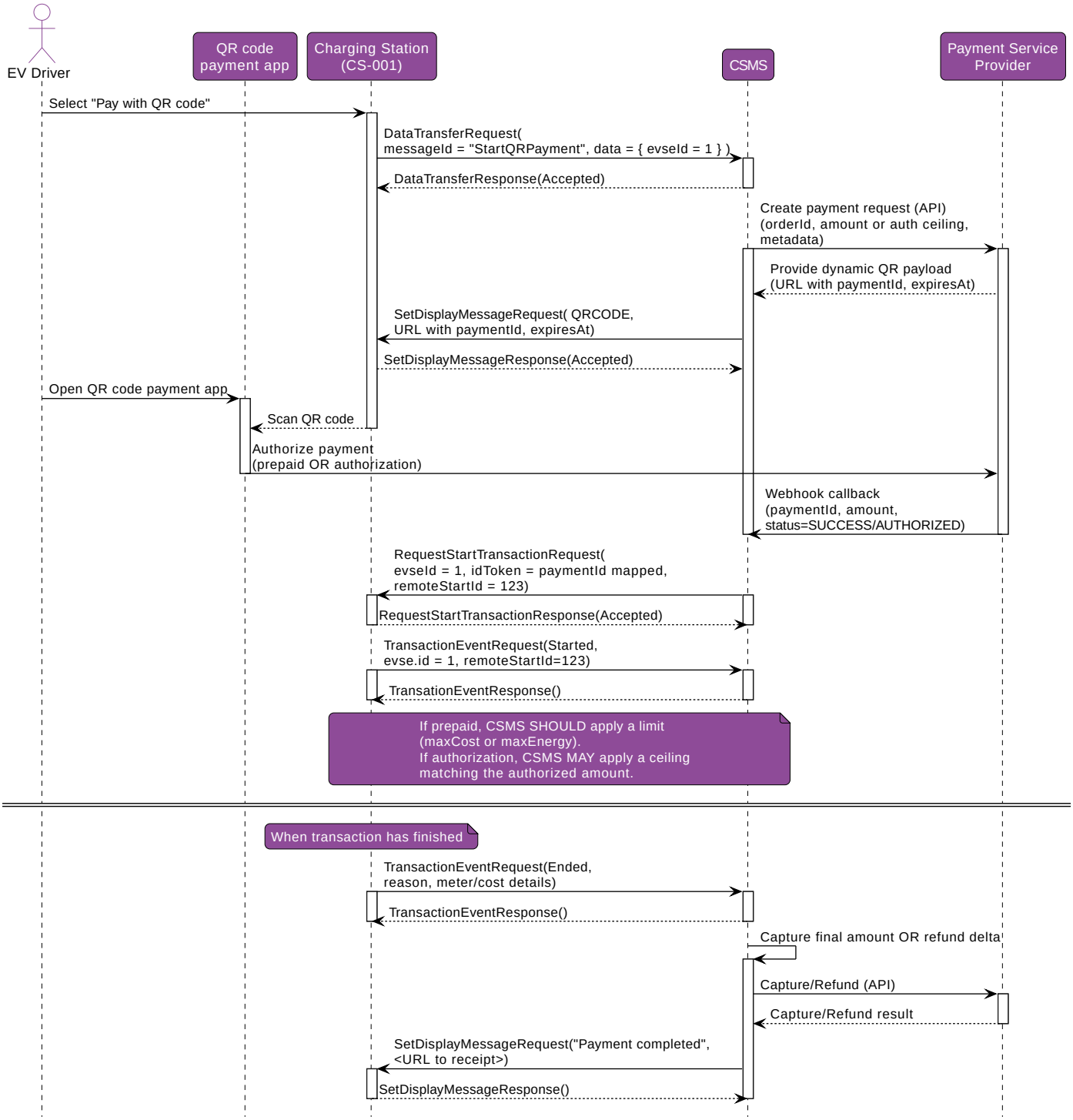

Figure 2. Sequence diagram for app payment via a QR code, that is managed by CSMS

Custom DataTransfer message

The simple DataTransfer message that is needed to trigger CSMS to start a QR code payment process, has the following content:

```
DataTranferRequest {
  "vendorId": "org.openchargealliance.qrpayment",
  "messageId": "StartQRPayment"
  "data": { "evseId": 1 }
}

DataTransferResponse {
  "status": "Accepted" or "Rejected"
}
```

2.3. Payment service provider's QR code coming from charging station for payment via app

This integration model circumvents the need for a custom DataTransfer message to trigger CSMS when the user selects "Pay with QR code" by handling all payment provider interactions locally on the charging station. This assumes that the **charging station has internet connectivity to the payment provider service**.

In short:

1. The EV driver selects "Pay with QR code" on the charger.
2. The charging station calls the payment provider API to create a payment request.
3. The payment provider returns a dynamic QR payload.
4. The charger displays the dynamic QR code.
5. The driver scans the QR using the QR code payment app.
6. The payment provider authorizes the payment.
7. The charging station receives real-time confirmation from the payment provider.
8. The charging station starts the charging session.

The characteristics are the same as mentioned in Payment service provider's QR code coming from CSMS for payment via app with the following addition:

- Requires credentials and configuration for a payment provider API in every charging station
- No custom DataTransfer message required.

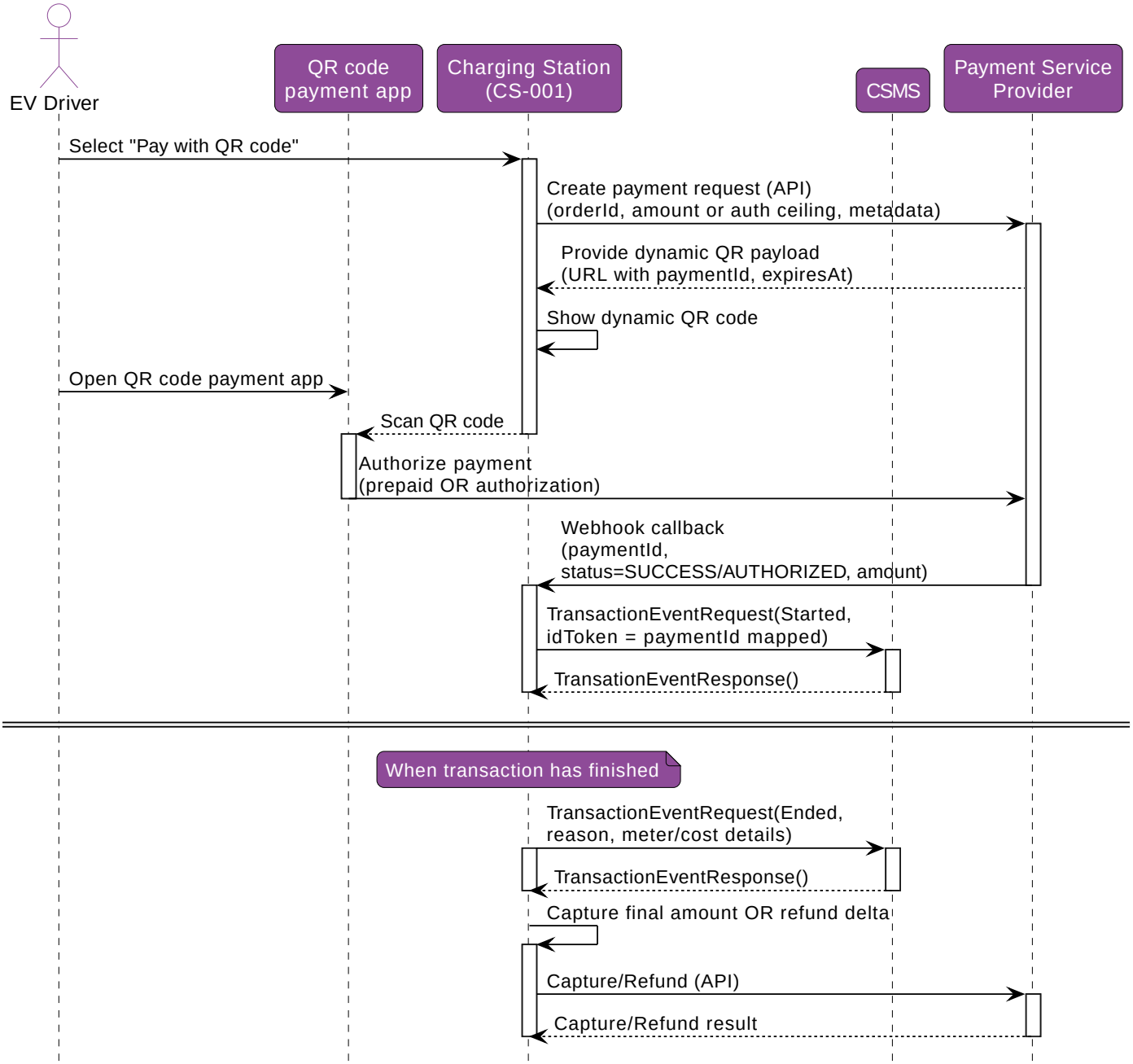

Figure 3. Sequence diagram app payment via a QR code, that is managed by charging station

3. Prepaid versus authorization models

The following two main payment methods are used by major QR code payment services in Japan that can be applied to EV charging.

3.1. Prepaid

- User pays a fixed amount upfront
- Charging session is limited to that amount
- If the actual amount spent is less than the prepaid amount a refund for the unused portion is required

3.2. Authorization with delayed capture

- A maximum amount that can be expected is authorized (reserved in the account)
- Charging session runs without a predetermined amount limit
- The final amount is captured (settled) when the session finishes
- Unused reserved funds are automatically released
- No refund workflow required

For EV charging, authorization with delayed capture provides a cleaner operational model and better user experience.

4. Security considerations

The most secure configuration is when a charging station only connects to the CSMS (with the highest security profile #3) and does not allow any other connections. This configuration clearly limits communication paths and authentication methods, minimizing the attack surface.

If Payment service provider's QR code coming from charging station for payment via app is used, however, then every charging station needs its own internet connection to the payment provider service. Not only does this open up an additional attack surface, it also requires the operator to configure every charging station with the credentials for the payment provider service. Care must be taken to ensure that these credentials cannot be stolen or misused even if a charging station is compromised.