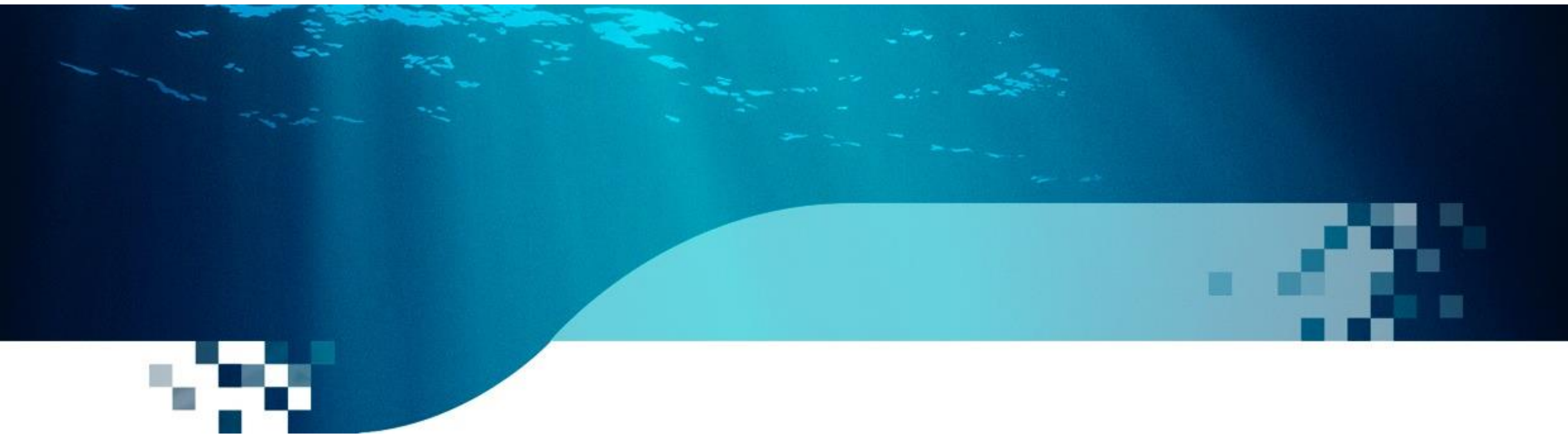


Technical Reference EVO E-PAY

EMV 3-D Secure Integration

April 27, 2022

Changes in contrast to the previous document version

Chapter no.	Key word	Content of change	Author (initials)	Date
3.1.2	**URLSuccess**, **URLFailure**, **URLNotify**	General notes on **URLSuccess**, **URLFailure** and **URLNotify**	HD	Apr. 27, 2022

About this document

Explanation of role-related terms

Three main roles exist in the complex scenario of payment processing. Depending on the perspective and situation at hand, two of these can act as customers, or two of them as service providers. To enable a clear distinction, systematic use will be made of the following terms:

Merchant or Client

Uses, in its capacity as merchant or services provider, the products of EVO Payments to settle payments for the goods or services it offers. Is the contract partner and thus the direct "customer" of EVO Payments.

Customer or end customer

Customers who purchase goods from the merchant or services from the service provider; are contractual partners of the merchant / service partner and not of EVO Payments.

EVO Payments

EVO Payments provides services connected with payment processing and acts as the link between the parties, especially the merchant / service provider and other establishments such as card organizations and other institutions involved in the processing of payments.

Terms in the EVO Payments XML language

All terms belonging to the EVO Payments XML language are displayed in "Consolas" font in this document (e. g. **`PaymentTransactionType`**). If such XML terms are separated by line breaks, they do not have a hyphen, as this might lead to misunderstandings in the spelling of the programming terms. Thus, if an XML term is separated by a line break in this document, the line break must be ignored when programming.

Example:
Text in this document text in this document text in this document text in this document **`Payment`**
`TransactionType` text in this document text in this Document Text in …

Text when programming: **`PaymentTransactionType`**

Table of Contents

All statements made in this document refer to the production date of this document. All statements may be subject to changes and amendments. This document may not be reproduced or distributed without our prior written permission.

1. Regulatory Requirements

1.1 EBA Mandate

The European Banking Authority (EBA) mandated that all payer who access their payment account online and initiate electronic payment transactions through a remote channel must be strongly authenticated (aka Strong Customer Authentication (SCA)) commencing September 14, 2019. The card organizations seized this opportunity to overhaul the established 3-D Secure protocol for cardholder authentication and to address several issues that curbed adoption in the market.

1.2 3DS 2.0

Previously, internet merchants had the choice to either present a cardholder challenge (e.g. TAN / password) or to give 3DS a pass entirely. Some adopted a dynamic approach based on PSP or own risk assessment, but many merchants valued a frictionless checkout and high conversion rates more than the potential benefits of a liability shift. The card organization's overall strategy for 3DS 2.0 is to reduce friction through an improved cardholder experience (device awareness) and to leverage exemptions from SCA based on robust transaction risk analysis (TRA) with the ultimate goal of delivering optimal authorization performance and conversion rates. Thus, TRA is key to delivering frictionless payment experiences for low-risk remote transactions. Therefore the 3DS 2.0 protocol introduced a plethora of additional data points that can be transferred to the issuer to aid transaction risk analysis and to apply exemptions from SCA.

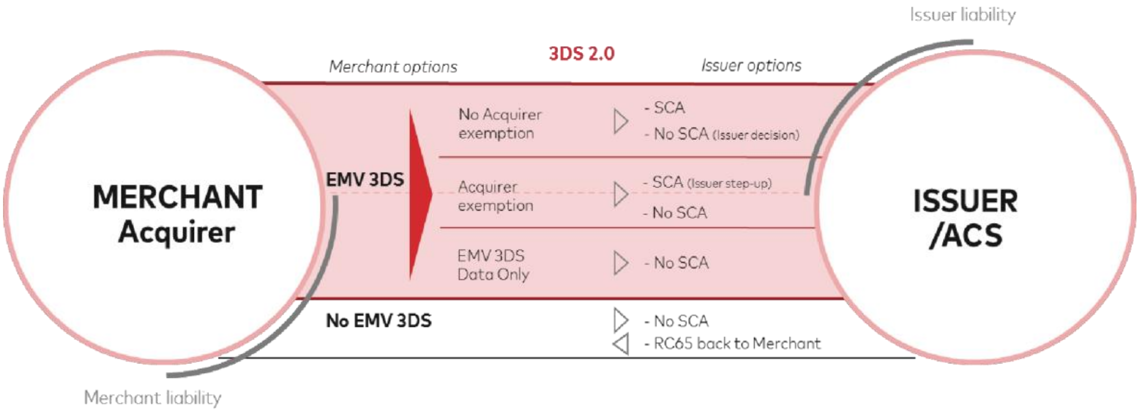
SCA will be required when:

- The transaction is not out of scope of the PSD2 RTS
- No PSD2 SCA exemption applies for a payment transaction
- Adding a card to a Merchant’s file (card-on-file)
- Starting a recurring payment arrangement for fixed and variable amounts, including setting the initial mandate for Merchant-Initiated Transactions
- Changing a recurring payment agreement for a higher amount (premium offering for example)
- Setup of white-listing (or viewing/amending white-lists)
- Binding a device to a Cardholder

1.3 Liability Shift

As a rule of thumb, when cardholder authentication was performed through 3-D Secure, merchants are typically protected against e-commerce fraud-related disputes and liability shifts from the merchant / acquirer to the issuer. There are exceptions to merchant dispute protection though. In the context of 3DS 2.0 merchants are regularly not protected if granted exemptions according to PSD2 RTS were actively requested by merchant / acquirer.

The following diagram depicts options and liabilities under PSD2 RTS requirements according to Mastercard.

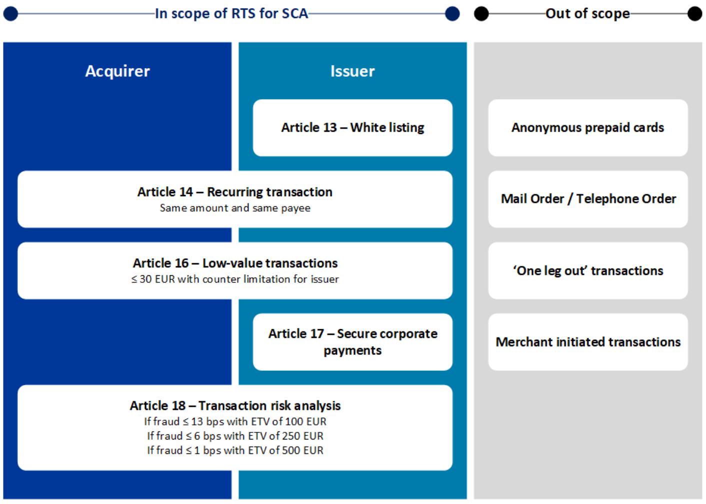


1.4 3DS 2.0 and GDPR Compliance

Cardholders must be provided with detailed information about how their data is collected, used and processed. This can be ensured via a Privacy Notice including at a minimum the types of data being processed, the purposes of their processing, data uses, etc. Card organizations and Issuers will not use EMV 3DS data for other purposes than fraud prevention and authentication. It excludes the usage of personal data for other purposes, such as sales, marketing and data mining (other than fraud prevention as purpose) activities.

1.5 PSD2 SCA Exemptions and Exclusions

There are some important exemptions to SCA according to the regulatory technical standards (RTS) that may apply in various conditions which are depicted in the following diagram.



In scope of RTS for SCA

Out of scope

Acquirer

Issuer

Article 13 – White listing

Article 14 – Recurring transaction
Same amount and same payee

Article 16 – Low-value transactions
≤ 30 EUR with counter limitation for issuer

Article 17 – Secure corporate payments

Article 18 – Transaction risk analysis
If fraud ≤ 13 bps with ETV of 100 EUR
If fraud ≤ 6 bps with ETV of 250 EUR
If fraud ≤ 1 bps with ETV of 500 EUR

Anonymous prepaid cards

Mail Order / Telephone Order

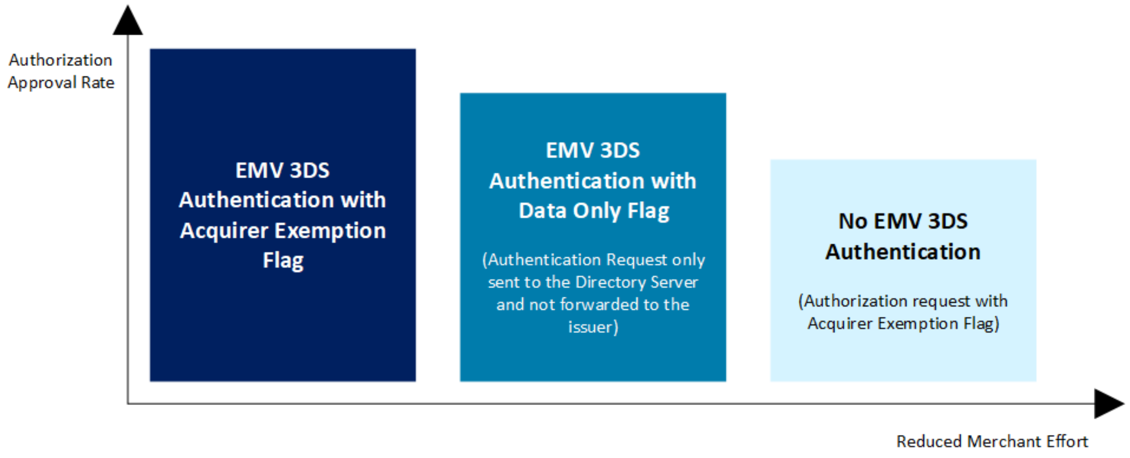
'One leg out' transactions

Merchant initiated transactions

2. EVO E-PAY

2.1 Authentication Options

An acquirer may be allowed to not apply SCA due to low fraud rates and TRA. For these exemptions, there are various processing options available as depicted in the diagram below.



As a standard, EVO Payments will submit (where supported) applicable exemptions through the EMV 3DS authentication flow to the issuer to achieve best possible authorization approval rates.

EBA-Op-2018-04, Paragraph 47 - Clarification on PSP (Acquirer Fraud Rates)
The fraud rate as defined in Annex A of the RTS is calculated for all credit transfer transactions and all card payment transactions and cannot be defined per individual payee (e.g. merchant) or per channel (whether app or web interface). The fraud rate that determines whether or not a PSP qualifies for the SCA exemption cannot be calculated for specific merchants only, i.e. where the payer wants to make a payment to a specific merchant and this specific merchant has a fraud risk that is below the threshold. While the payee's PSP (acquirer) may contractually agree to 'outsource' its transaction risk analysis monitoring to a given merchant, or allow only certain predefined merchants to benefit from that PSP's exemption (based on a contractually agreed low fraud rate), the fraud rate making a given PSP eligible for an exemption under Article 18 would still need to be calculated on the basis of the payee PSP's executed or acquired transactions, rather than on the merchant's transactions.

2.2 Message Version 2

To handle the amount of additional non-payment data and to maintain downward compatibility as much as possible EVO Payments decided to version its EVO E-PAY card interface via the additional data element MsgVer. The upgraded API is still based on key-value pairs but relies heavily on Base64 encoded JSON objects to aid readability and client-side scripting.

2.2.1 Whitelisting of trusted beneficiaries

A cardholder might opt to add a merchant to a list of trusted beneficiaries maintained at the issuer to exempt this particular merchant from SCA with future payments. This will usually occur during a cardholder challenge but cardholder's might also be able to manage a list of trusted beneficiaries through their banking app for instance.
Merchants may benefit from a whitelist exemption if requested and if a cardholder challenge is not required otherwise.

Please note that whitelisting is available with 3DS version 2.2 and higher. Currently issuer most support 3DS 2.1.

2.2.2 Recurring transactions

Recurring transactions are a series of transactions processed following an agreement between a cardholder and a merchant where the cardholder purchases goods or services over a period of time and through a number of separate transactions with the same amount. The initial transaction must be authenticated (i.e. cardholder initiated transaction (CIT)). Subsequent recurring payments are out of scope of RTS SCA since they are regularly merchant initiated (i.e. without customer being in session).

2.2.3 Low-value transactions

Issuers may exempt transactions from SCA provided that the following conditions are met:

- the payment amount does not exceed EUR 30,
- the cumulative amount of previous payment transactions without SCA does not exceed EUR 100,
- the number of previous payment transactions without SCA does not exceed five consecutive payment transactions.

Please note that low-vale exemptions must be requested to be considered for a frictionless authentication flow.

2.2.4 Transaction risk analysis

Acquirers and issuers are allowed not to apply SCA provided the overall fraud rate is not higher than the reference fraud rate for the exemption threshold value (ETV) specified in the table below and where the risk-based assessment of each individual transaction can be considered as low risk.

ETV	Card-based payments
EUR 500	1 bps
EUR 250	6 bps
EUR 100	13 bps

2.2.5 One-leg out transactions

One-leg out transactions are such transactions where either the payer's payment service provider or the payee's payment service provider are located outside the European Union.

Payment service provider in the context of a card based transaction and in the spirit of the PSD2 are regularly acquirer and issuer.

Thus, neither the nationality of the cardholder nor the merchant's business location are relevant for the assessment whether a transaction is out of scope due to the 'one-leg out' rule.

3. Integration Methods EN

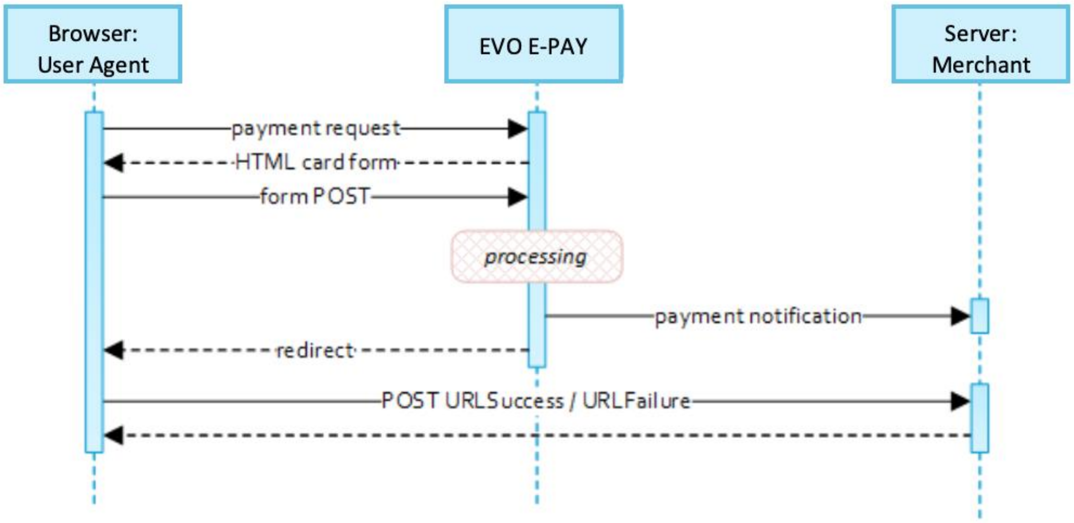
- EVO E-PAY interface: via form (paySSL) EN

3.1 EVO E-PAY interface: via form (paySSL) EN

When requesting card payments via EVO Payments hosted forms the complexity of 3-D Secure is completely removed from the merchant implementation.

From a merchant point of view the sequence itself does not differ between 3DS authenticated and non-authenticated payments though 3DS requires consideration of additional data elements in the request and response.

3.1.1 Simplified Sequence Diagram



3.1.2 Payment Request

To retrieve an EVO Payments card form please submit the following data elements via HTTP POST request method to https://spg.evopayments.eu/pay/payssl.aspx.

> As a general rule it is strongly recommended to always send conditional required data (C) elements to avoid unnecessary friction and declines.

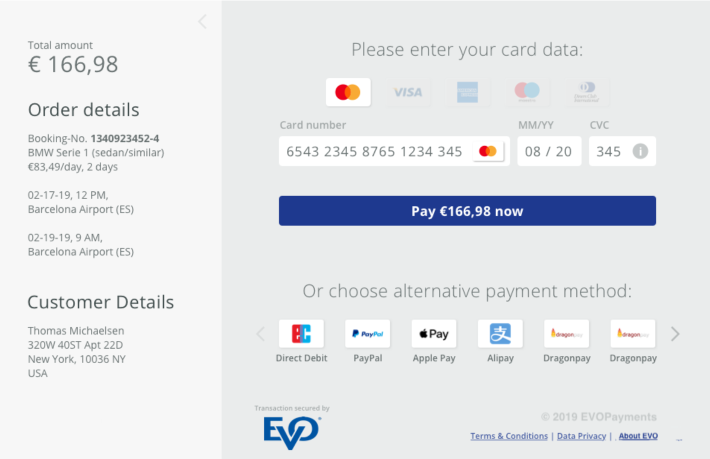
	Key	Format	Con-dition	Description
1	MerchantID	ans..30	M	Merchant identifier assigned by EVO Payments.
2	MsgVer	ans..5	M	Message version. Values accepted > 2.0
3	TransID	ans..64	M	Transaction identifier supplied by the merchant. Shall be unique for each payment.
4	RefNr	ans..20	M	Merchant’s unique reference number, which serves as payout reference. The following characters are permitted: > Digits (0..9) > Capital letters (A..Z) > Separators: point (.), hyphen (-) and slash (/)
5	Amount	n..10	M	Transaction amount in it smallest unit of the submission currency.
6	Currency	a3	M	ISO 4217 three-letter currency code.
7	Capture	ans..6	O	Determines the type and time of payment completion (i.e. dual message systems). Values accepted:

	Key	Format	Condition	Description
				> `AUTO` = completion immediately after authorization (default value) > `MANUAL` = completion made by the merchant > `NUMBER` = Delay in hours until the completion (whole number; 1 to 696).
8	billingDescriptor	ans..22	O	A descriptor to be printed on a cardholder's statement. Please also refer to the additional comments made elsewhere for more information about rules and regulations.
9	OrderDesc	ans..768	O	Order description.
10	AccVerify	a3	O	Indicator to request an account verification (aka zero value authorization). If an account verification is requested the submitted amount will be optional and ignored for the actual payment transaction (e.g. authorization). Values accepted > Yes
11	threeDSConfig	JSON	O	Object specifying merchant, acquirer and login data to be used for 3DS authentication. If submitted values override configuration data stored at the `MerchantID`.
12	threeDSPolicy	JSON	O	Object specifying authentication policies and exemption handling strategies.
13	priorAuthenticationInfo	JSON	O	Prior Transaction Authentication Information contains optional information about a 3DS cardholder authentication that occurred prior to the current transaction.
14	accountInfo	JSON	O	The account information contains optional information about the customer account with the merchant.
15	billToCustomer	JSON	C	The customer that is getting billed for the goods and / or services. Required for EMV 3DS unless market or regional mandate restricts sending this information.
16	shipToCustomer	JSON	C	The customer that the goods and / or services are sent to. Required if different from billToCustomer.
17	billingAddress	JSON	C	Billing address. Required for EMV 3DS (if available) unless market or regional mandate restricts sending this information.
18	shippingAddress	JSON	C	Shipping address. If different from billingAddress, required for EMV 3DS (if available) unless market or regional mandate restricts sending this information.
19	credentialOnFile	JSON	C	Object specifying type and series of transactions using payment account credentials (e.g. account number or payment token) that is stored by a merchant to process future purchases for a customer. Required if applicable.
20	merchantRiskIndicator	JSON	O	The Merchant Risk Indicator contains optional information about the specific purchase by the customer. If no `shippingAddress` is present it is strongly recommended to populate the `shippingAddressIndicator` property with an appropriate value such as `shipToBillingAddress`, `digitalGoods` or `noShipment`.
21	URLNotify	an..256	M	A FQDN URL to submit the final payment result (HTTP POST).
22	URLSuccess	an..256	M	A FQDN URL for redirection of the client in case the payment was processed successfully (HTTP POST).
23	URLFailure	an..256	M	A FQDN URL for redirection of the client in case the payment could not be processed successfully (HTTP POST).
24	userData	ans..1024	O	Base64 encoded custom value that will be returned in responses and notifications.

	Key	Format	Con-dition	Description
25	MAC	an64	M	Hash Message Authentication Code (HMAC) with SHA-256 algorithm.

➡ General notes on **`URLSuccess`**, **`URLFailure`** and **`URLNotify`**:
- We recommend using the **`response=encrypted`** parameter to get an encrypted response from EVO E-PAY.
- Fraudsters can copy the encrypted **`DATA`** element sent to **`URLFailure`** and fraudulently send **`DATA`** to **`URLSuccess`** / **`URLNotify`**. Therefore, be sure to check the **`code`** value of the **`DATA`** element. Only a response with **`code=00000000`** should be considered successful.

EVO E-PAY will return an HTML document in the response body representing the requested card form. The form may be included in the merchant checkout page or used as a standalone page to redirect the cardholder to.



Total amount
€ 166,98

Order details

Booking-No. **1340923452-4**
BMW Serie 1 (sedan/similar)
€83,49/day, 2 days

02-17-19, 12 PM,
Barcelona Airport (ES)

02-19-19, 9 AM,
Barcelona Airport (ES)

Customer Details

Thomas Michaelsen
320W 40ST Apt 22D
New York, 10036 NY
USA

Please enter your card data:

Card number: 6543 2345 8765 1234 345
MM/YY: 08 / 20
CVC: 345

Or choose alternative payment method:

Cardholder authentication and payment authorization will take place once the the cardholder entered all required card details and submitted the form data to EVO E-PAY.

Note: In case you are using your own templates (Corporate Payment Page), please make sure you include Cardholder name on your custom template. Cardholder name is mapped to Paygate API parameter "CreditCardHolder". Cardholder name field must not contain any special characters and must have minimal length of 2 characters and maximum length of 45 characters.

When the payment is completed EVO E-PAY will send a notification to the merchant server (i.e. **URLNotify**) and redirect the browser to the **URLSuccess** respectively to the **URLFailure**.

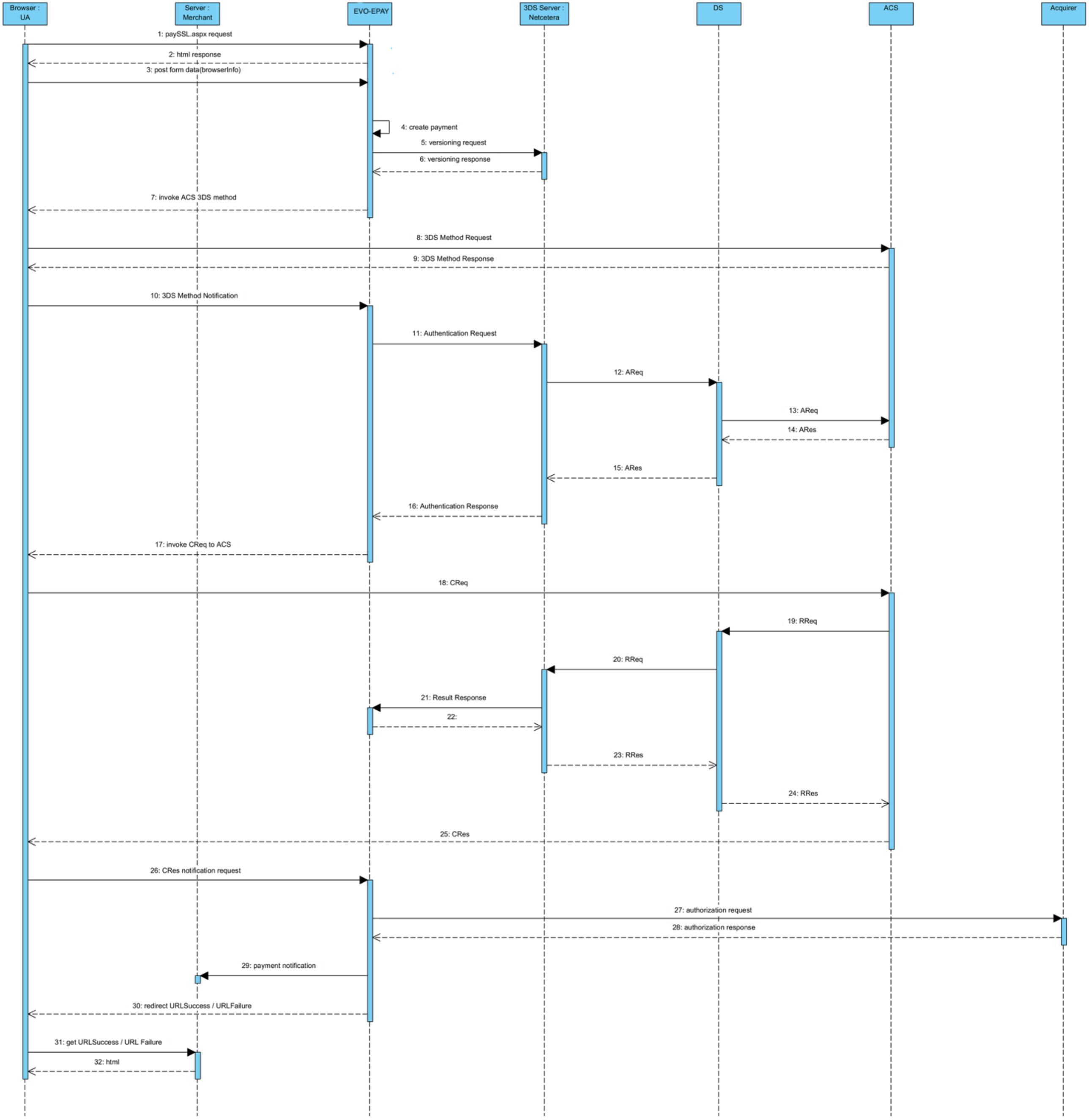
The blowfish encrypted data elements as listed in the following table are transferred via **HTTP POST** request method to the **URLNotify** and **URLSuccess**/**URLFailure**.

➡ Please note that the call of **URLSuccess** or **URLFailure** takes place with a **GET** in case of fallback to 3-D Secure 1.0. Therefore your systems should be able to receive parameters both via **GET** and via **POST**.

3.1.3 HTTP POST to URLSuccess / URLFailure / URLNotify

Key	Format	Condition	Description
MID	ans..30	M	Merchant identifier assigned by EVO Payments.
MsgVer	ans..5	M	Message version. Accepted values: • `2.0`
PayID	ans32	M	Payment/transaction identifier assigned by EVO Payments.
XID	ans64	M	ID assigned by EVO E-PAY for the operation performed on the payment.
TransID	ans..64	M	Transaction identifier supplied by the merchant.
schemeReferenceID	ans..64	C	Card scheme specific transaction ID required for subsequent credential-on-file payments, delayed authorizations and resubmissions.
Status	a..20	M	Status of the transaction. Values accepted: • `Authorized` • `OK` (Sale) • `FAILED` In case of *Authentication-only* the *Status* will be either `OK` or `FAILED`.
Description	ans..1024	M	Textual description of the code.
Code	n8	M	EVO E-PAY response code.
card	JSON	M	Card response data.
ipInfo	JSON	C	Object containing IP information. Presence depends on the configuration for the merchant.
threeDSData	JSON	M	Authentication data.
resultsResponse	JSON	C	In case the authentication process included a cardholder challenge additional information about the challenge result will be provided.
userData	ans..1024	C	Base64 encoded custom value as submitted in the request.
MAC	an64	M	Hash Message Authentication Code (HMAC) with SHA-256 algorithm.

3.1.4 Extended Sequence Diagram



3.2 EVO E-PAY interface: via server-to-server (direct.aspx) EN

Please note that the Server-2-Server Integration is **only** relevant for **subsequent transactions** with a **pseudo card number** and corresponding **credentialOnFile** and **chemeReferenceID** parameters.
Initial transactions must be submitted via form (paySSL).

3.2.1 Overview

A 3DS 2.0 payment sequence may comprise the following distinct activities:

- Versioning
  - Request ACS and DS Protocol Version(s) that correspond to card account range as well as an optional 3DS Method URL
- 3DS Method
  - Connect the cardholder browser to the issuer ACS to obtain additional browser data
- Authentication
  - Submit authentication request to the issuer ACS
- Challenge
  - Challenge the carholder if mandated
- Authorization
  - Authorize the authenticated transaction with the acquirer

Server-2-Server Sequence Diagram

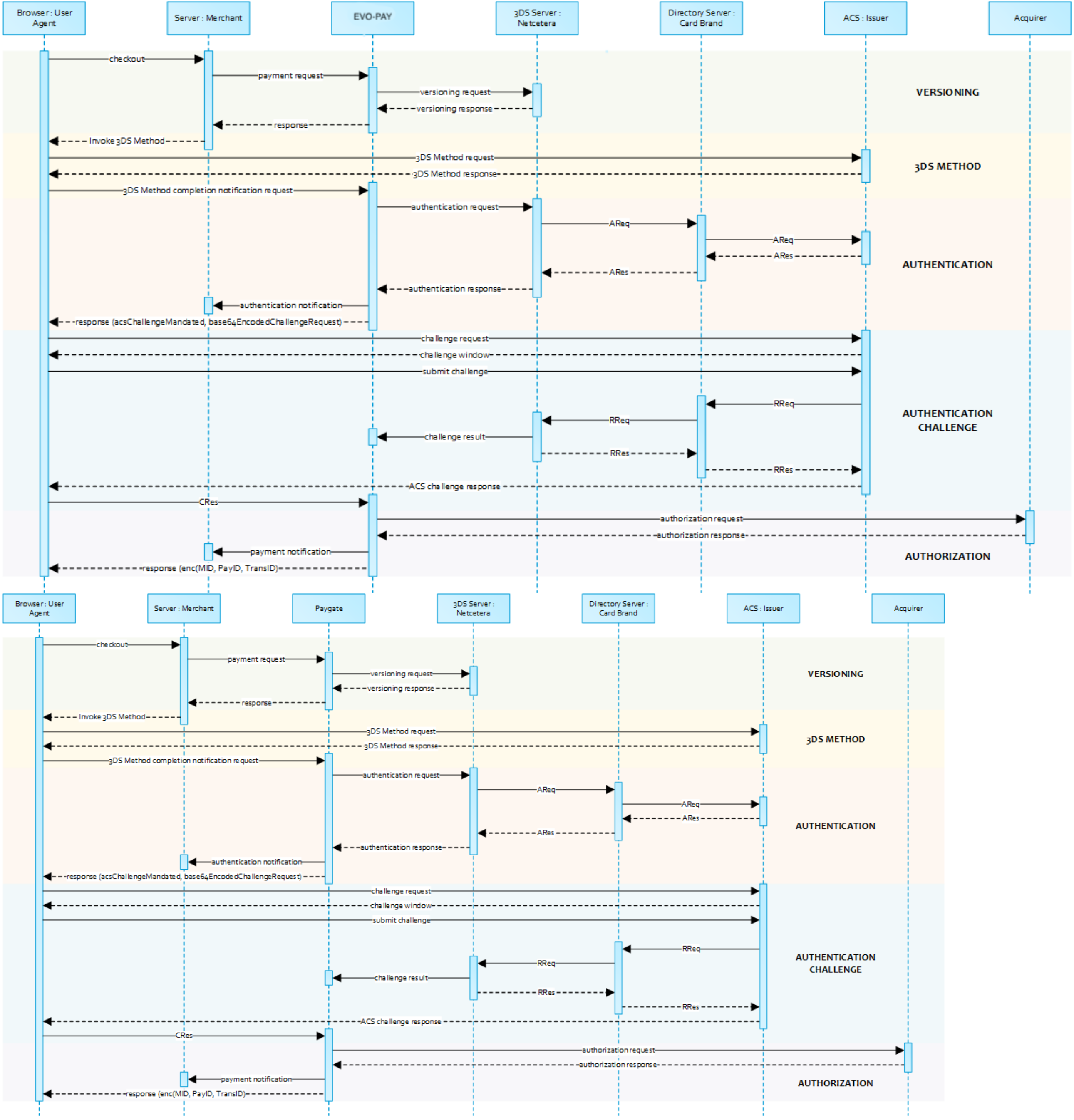
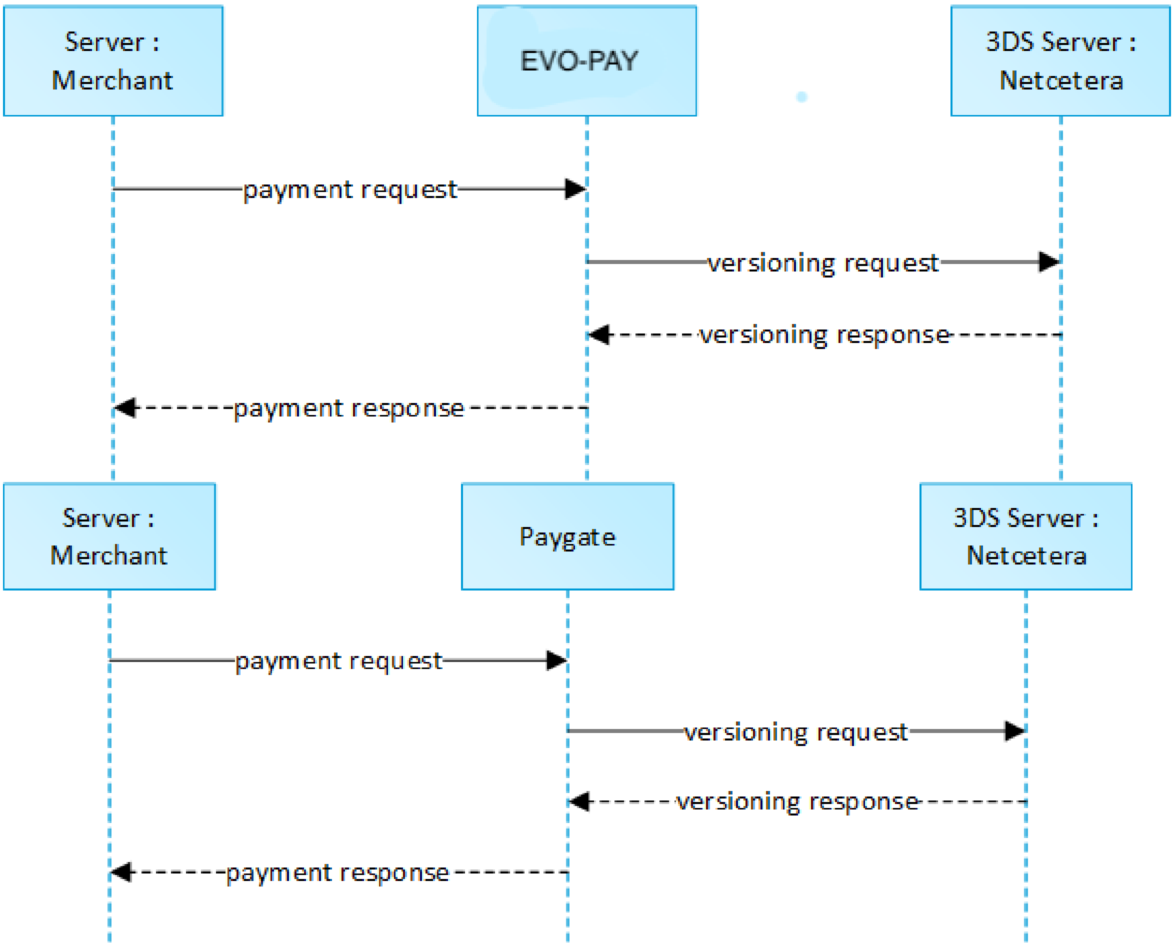


Figure 1 Server-2-Server Sequence Diagram

Please note that the communication between client and Access Control Server (ACS) is implemented through iframes. Thus, responses arrive in an HTML subdocument and you may establish correspondent event listeners in your root document. Alternatively you could solely rely on asynchronous notifications delivered to your backend. In those cases you may have to consider methods such as long polling, SSE or websockets to update the client.

3.2.2 Payment Initiation

The initial request to EVO E-PAY will be the same regardless of the underlying 3DS Protocol.



In order to start a server-to-server 3-D Secure card payment sequence please post the following key-value-pairs to https://spg.evopayments.eu/pay/direct.aspx.

3.2.2.1 Request Elements

Notice: In case of a merchant initiated recurring transaction the JSON objects (besides credentialOnFile), the URLNotify and TermURL are not mandatory parameters, because no 3-D Secure and no risk evaluation is done by the card issuing bank and the payment result is directly returned within the response.

	Key	Format	Con-dition	Description
1	MerchantID	ans..30	M	Merchant identifier assigned by EVO Payments.
2	MsgVer	ans..5	M	Message version. Values accepted:

	Key	Format	Condition	Description
				• 2.0
3	TransID	ans..64	M	Transaction identifier supplied by the merchant. Shall be unique for each payment.
4	RefNr	ans..20	M	Merchant's unique reference number, which serves as payout reference. The following characters are permitted: > Digits (0..9) > Capital letters (A..Z) Separators: point (.), hyphen (-) and slash (/)
5	schemeReferenceID	ans..64	C	Card scheme specific transaction ID required for subsequent credential-on-file payments, delayed authorizations and resubmissions.
6	Amount	n..10	M	Transaction amount in it smallest unit of the submission currency.
7	Currency	a3	M	ISO 4217 three-letter currency code.
8	card	JSON	M	Card data.
9	Capture	ans..6		Determines the type and time of payment completion (i.e. dual message systems). Values accepted: • AUTO = completion immediately after authorization (default value). • MANUAL = completion made by the merchant. • <Number> = Delay in hours until the completion (whole number; 1 to 696).
10	channel	a..20	C	Indicates the type of channel interface being used to initiate the transaction. Values accepted: • Browser • App • 3RI If not present the value Browser is implied.
11	billingDescriptor	ans..22	O	A descriptor to be printed on a cardholder's statement. Please also refer to the additional comments made elsewhere for more information about rules and regulations.
12	OrderDesc	ans..768	O	Order description.
13	TermURL	ans..256	M	In case of 3DS 1.0 fallback: the URL the customer will be returned to at the end of the 3DS 1.0 authentication process.
14	AccVerify	a3	O	Indicator to request an account verification (aka zero value authorization). If an account verification is requested the submitted amount will be optional and ignored for the actual payment transaction (e.g. authorization). Values accepted: • Yes
15	threeDSConfig	JSON	O	Object specifying merchant, acquirer and login data to be used for 3DS authentication. If submitted values override configuration data stored at the MerchantID.
16	threeDSPolicy	JSON	O	Object specifying authentication policies and exemption handling strategies.
17	threeDSData	JSON	C	Object detailing authentication data in case authentication was performed through a third party or by the merchant.
18	priorAuthenticationInfo	JSON	O	Prior Transaction Authentication Information contains optional information about a 3DS cardholder authentication that occurred prior to the current transaction.

	Key	Format	Condition	Description
19	browserInfo	JSON	C	Accurate browser information are needed to deliver an optimized user experience. Required for 3DS 2.0 transactions.
20	accountInfo	JSON	O	The account information contains optional information about the customer account with the merchant. Optional for 3DS 2.0 transactions.
21	billToCustomer	JSON	C	The customer that is getting billed for the goods and / or services. Required unless market or regional mandate restricts sending this information.
22	shipToCustomer	JSON	C	The customer that the goods and / or services are sent to. Required (if available and different from billToCustomer) unless market or regional mandate restricts sending this information.
23	billingAddress	JSON	C	Billing address. Required for 3DS 2.0 (if available) unless market or regional mandate restricts sending this information.
24	shippingAddress	JSON	C	Shipping address. If different from billingAddress, required for 3DS 2.0 (if available) unless market or regional mandate restricts sending this information.
25	credentialOnFile	JSON	C	Object specifying type and series of transactions using payment account credentials (e.g. account number or payment token) that is stored by a merchant to process future purchases for a customer. Required if applicable.
26	merchantRiskIndicator	JSON	O	The Merchant Risk Indicator contains optional information about the specific purchase by the customer.
27	URLNotify	an..256	M	The merchant URL that receive asynchronous requests during the authentication process.
28	userData	ans..1024	O	Base64 encoded custom value that will be returned in responses and notifications.
29	MAC	an64	M	Hash Message Authentication Code (HMAC) with SHA-256 algorithm.

3.2.2.2 Response Elements

	Key	Format	Condition	Description
1	MID	ans..30	M	Merchant identifier assigned by EVO Payments.
2	PayID	ans32	M	Payment/transaction identifier assigned by EVO Payments.
3	XID	ans64	M	ID assigned by EVO E-PAY for the operation performed on the payment.
4	TransID	ans..64	M	Transaction identifier supplied by the merchant. Shall be unique for each payment.
5	Code	n8	M	EVO E-PAY response code.
6	Status	a..20	M	Status of the transaction. Values accepted: • `AUTHENTICATION_REQUEST` • `PENDING` • `FAILED`
7	Description	ans..1024	M	Textual description of the code.
8	versioningData	JSON	M	The Card Range Data data element contains information that indicates the most recent EMV 3-D Secure version supported by the ACS that hosts that card range. It also may optionally contain the ACS URL for the 3DS Method if supported by the ACS and the DS Start and End Protocol Versions which support the card range.
9	threeDSLegacy	JSON	M	Object containing the data elements required to construct the Payer Authentication request in case of a fallback to 3DS 1.0.

	Key	Format	Con-dition	Description
10	userData	ans..1024	C	Base64 encoded custom value as submitted in the request.
11	MAC	an64	M	Message Authentication Code (HMAC) with SHA-256 algorithm.

The `versioningData` object will indicate the EMV 3DS protocol versions (i.e. 2.1.0 or higher) that are supported by Access Control Server of the issuer.

If the corresponding protocol version fields are NULL it means that the BIN range of card issuer is not registered for 3DS 2.0 and a fallback to 3DS 1.0 is required for transactions that are within the scope of PSD2 SCA.

When parsing `versioningData` please also refer to the subelement `errorDetails` which will specify the reason if some fields are not pupoluated (e.g. Invalid cardholder account number passed, not available card range data, failure in encoding/serialization of the 3DS Method data etc.).

```
{
  "threeDSServerTransID": "14dd844c-b0fc-4dfe-8635-366fbf43468c",
  "acsStartProtocolVersion": "2.1.0",
  "acsEndProtocolVersion": "2.1.0",
  "dsStartProtocolVersion": "2.1.0",
  "dsEndProtocolVersion": "2.1.0",
  "threeDSMethodURL": "http://www.acs.com/script",
  "threeDSMethodDataForm": "eyJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMI-
joiaHR0cHM6Ly93d3cuY29tcHV0b3AtcGF5Z2F0ZS5jb20vY2JUaHJlZURTLmFzcHg_YWN0aW9uPW10aGROdG
ZuIiwidGhyZWVEU1NlcnZlclRyYW5zSUQiOiIxNGRkODQ0Yy1iMGZjLTRkZmUtODYzNS0zNjZmYmY0MzQ2OG-
MifQ==",
  "threeDSMethodData": {
    "threeDSMethodNotificationURL": "https://spg.evopayments.eu/pay/
https://www.computop-paygate.com/cbThreeDS.aspx?action=mthdNtfn",
    "threeDSServerTransID": "14dd844c-b0fc-4dfe-8635-366fbf43468c"
  }
}
```

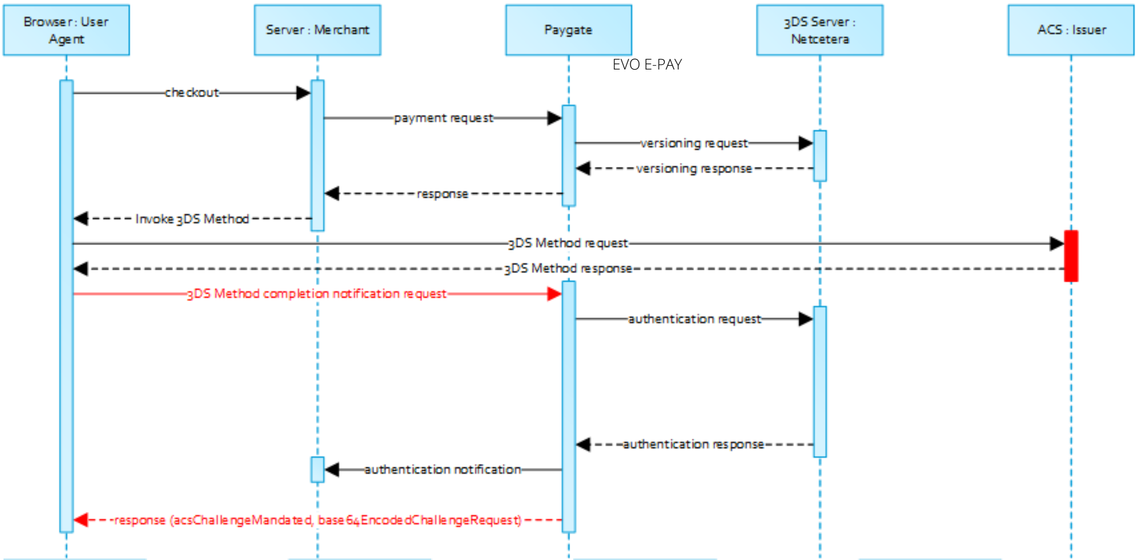
Code Block 1 versioningData

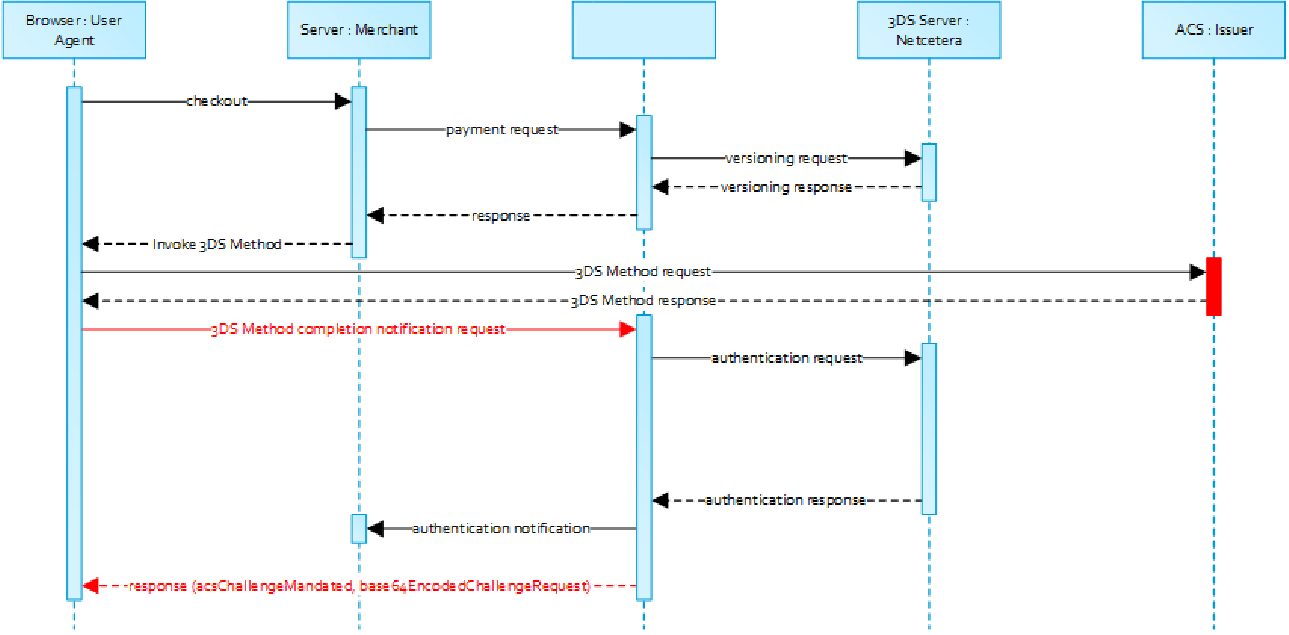
3.2.3 3DS Method

The 3DS Method allows for additional browser information to be gathered by an ACS prior to receipt of the authentication request message (AReq) to help facilitate the transaction risk assessment. Support of 3DS Method is optional and at the discretion of the issuer.

The `versioningData` object contains a value for `threeDSMethodURL`. The merchant is supposed to invoke the 3DS Method via a hidden HTML iframe in the cardholder browser and send a form with a field named `threeDSMethodData` via HTTP POST to the ACS 3DS Method URL.

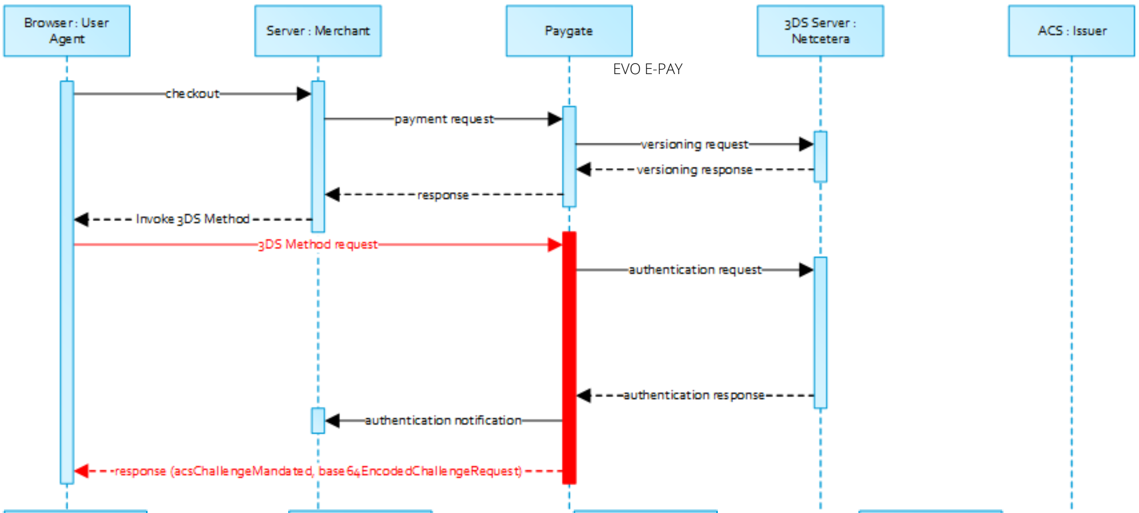
3DS Method: `threeDSMethodURL`

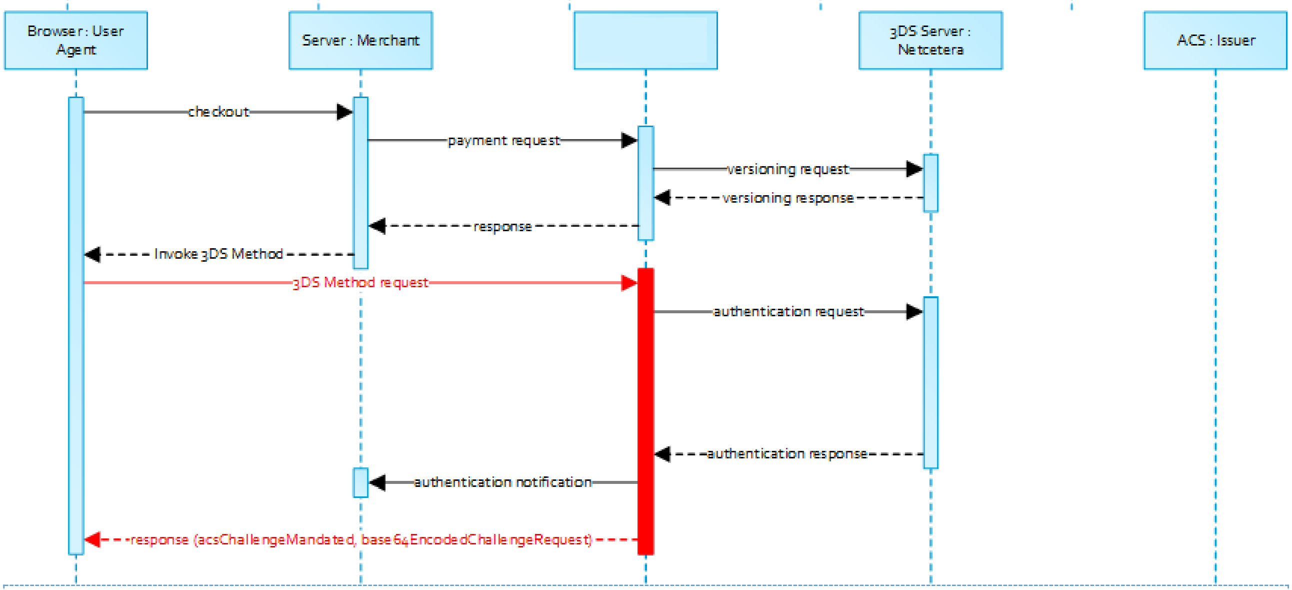




Please not that the `threeDSMethodURL` will be populated by EVO E-PAY if the issuer does not support the 3DS Method. The 3DS Method Form Post as outlined below must be performed independently from whether it is supported by the issuer. This is necessary to facilitate direct communication between the browser and EVO E-PAY in case of a mandated challenge or a frictionless flow.

3DS Method: No issuer `threeDSMethodURL`





```
<form name="frm" method="POST" action="Rendering URL">
  <input                    type="hidden"                    name="threeDSMethodData"
value="eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6IjNhYzdjYWE3LWFhNDItMjY2My03OT-
FiLTJhYzA1YTU0MmM0YSIsInRocmVlRFNNZXRob2ROb3RpZmljYXRpb25VUkwi-
OiJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIn0">
</form>
```

Code Block 2 3DS Method Form Post

The ACS will interact with the Cardholder browser via the HTML iframe and then store the applicable values with the 3DS Server Transaction ID for use when the subsequent authentication message is received containing the same 3DS Server Transaction ID.

Netcetera 3DS Web SDK
You may use the operations `init3DSMethod` or `createIframeAndInit3DSMethod` at your discretion from the nca3DSWebSDK in order to initiate the 3DS Method. Please refer to the Integration Manual at https://mpi.netcetera.com/3dsserver/doc/current/integration.html#Web_Service_API.

Once the 3DS Method is concluded the ACS will instruct the cardholder browser through the iFrame response document to submit `threeDSMethodData` as a hidden form field to the 3DS Method Notification URL.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8"/>
  <title>Identifying...</title>
</head>
<body>
<script>
  var                              tdsMethodNotificationValue                              =
'eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6ImUxYzFlYmViLTc0ZTgtND-
NiMi1iMzg1LTJlNjdkMWFhY2ZhMiJ9';

  var form = document.createElement("form");
  form.setAttribute("method", "post");
  form.setAttribute("action", "notification URL");

  addParameter(form, "threeDSMethodData", tdsMethodNotificationValue);

  document.body.appendChild(form);
  form.submit();

  function addParameter(form, key, value) {
    var hiddenField = document.createElement("input");
    hiddenField.setAttribute("type", "hidden");
    hiddenField.setAttribute("name", key);
    hiddenField.setAttribute("value", value);
    form.appendChild(hiddenField);
  }
</script>
</body>
</html>
```

Code Block 3 ACS Response Document

```
<form name="frm" method="POST" action="3DS Method Notification URL">
  <input                    type="hidden"                    name="threeDSMethodData"
value="eyJ0aHJlZURTU2VydmVyVHJhbnNJRCI6ImUxYzFlYmViLTc0ZTgtND-
NiMi1iMzg1LTJlNjdkMWFhY2ZhMiJ9">
</form>
```

Code Block 4 3DS Method Notification Form

Please note that the `threeDSMethodNotificationURL` as embedded in the Base64 encoded `threeDSMethodData` value points to EVO E-PAY and must not be modified. The merchant notification is delivered to the URLNotify as provided in the original request or as configured for the MerchantID in EVO E-PAY.

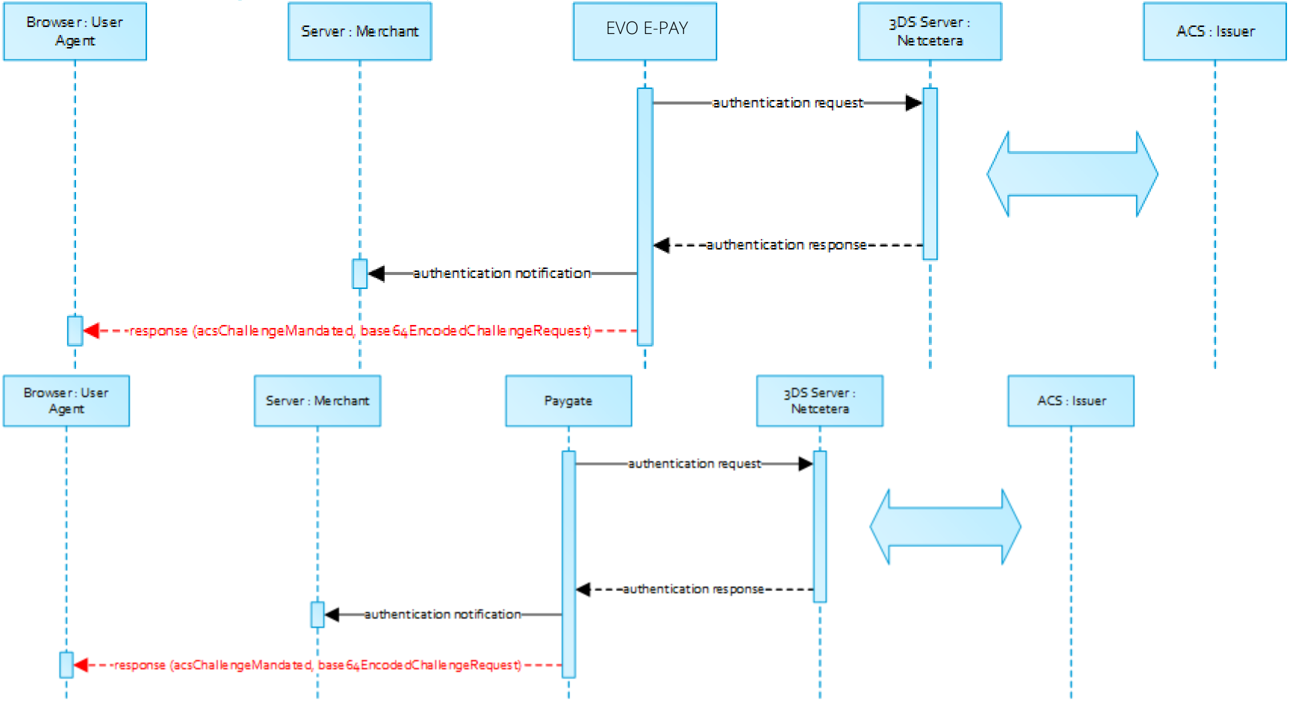
3.2.4 Authentication

If 3DS Method is supported by the issuer ACS and was invoked by the merchant EVO E-PAY will automatically continue with the authentication request once the 3DS Method has completed (i.e. 3DS Method Notification).

The authentication result will be transferred via HTTP POST to the `URLNotify`. It may indicate that the Cardholder has been authenticated, or that further cardholder interaction (i.e. challenge) is required to complete the authentication.

In case a cardholder challenge is mandated EVO E-PAY will transfer a JSON object within the body of HTTP browser response with the elements `acsChallengeMandated`, `challengeRequest`, `base64EncodedChallengeRequest` and `acsURL`. Otherwise, in a frictionless flow, EVO E-PAY will automatically continue and respond to the cardholder browser once the authorization completed.

Cardholder Challenge: Browser Response



3.2.4.1 Browser Challenge Response

Data Elements

	Key	Format	Condition	Description
1	acsChallengeMandated	boolean	M	Indication of whether a challenge is required for the transaction to be authorized.
2	challengeRequest	object	M	Challenge request object.
3	base64EncodedChallengeRequest	string	M	Base64-encoded Challenge Request object.
4	acsURL	string	M	Fully qualified URL of the ACS to be used to post the Challenge Request.

Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "acsChallengeMandated": {"type": "boolean"},
    "challengeRequest": {"type": "object"},
    "base64EncodedChallengeRequest": {"type": "string"},
    "acsURL": {"type": "string"}
  },
  "required": ["acsChallengeMandated", "challengeRequest", "base64EncodedChallengeRequest", "acsURL"],
  "additionalProperties": false
}
```

Code Block 5 Scheme: Browser Challenge Response

Sample

```
{
  "acsChallengeMandated": true,
  "challengeRequest": {
    "threeDSServerTransID": "8a880dc0-d2d2-4067-bcb1-b08d1690b26e",
    "acsTransID": "d7c1ee99-9478-44a6-b1f2-391e29c6b340",
    "messageType": "CReq",
    "messageVersion": "2.1.0",
    "challengeWindowSize": "01",
    "messageExtension": [
      {
        "name": "emvcomsgextInChallenge",
        "id": "tc8Qtm465Ln1FX0nZprA",
        "criticalityIndicator": false,
        "data": "messageExtensionDataInChallenge"
      }
    ]
  },
  "base64EncodedChallengeRequest": "base64-encoded-challenge-request",
  "acsURL": "acsURL-to-post-challenge-request"
}
```

Code Block 6 Sample: Browser Challenge Response

3.2.4.2 Authentication Notification

The data elements of the authentication notification are listed in the table below.

	Key	Format	Condition	Description
1	MID	ans..30	M	Merchant identifier assigned by EVO Payments.
2	PayID	ans32	M	Payment/transaction identifier assigned by EVO Payments.
3	TransID	ans..64	M	Transaction identifier supplied by the merchant. Shall be unique for each payment.
4	Code	n8	M	EVO E-PAY response code.
5	authenticationResponse	JSON	M	Response object in return of the authentication request with the ACS.

	Key	Format	Condition	Description
6	MAC	an64	M	Hash Message Authentication Code (HMAC) with SHA-256 algorithm.

3.2.4.3 Browser Challenge

If a challenge is mandated (see `acsChallengeMandated`) the browser challenge will occur within the cardholder browser. To create a challenge it is required to post the value `base64EncodedChallengeRequest` via an HTML iframe to the ACS URL.

```
<form name="challengeRequestForm" method="post" action="acsChallengeURL">
  <input type="hidden" name="creq" value="ewogICAgInRocmVlRFNTZXJ2ZXJUcmFuc0lEIjogI-
jhhODgwZGMwLWQyZDItNDA2Ny1iY2IxLWIwOGQxNjkwYjI2ZSIsCiAgICAi-
YWNzVHJhbnNJRCI6ICJkN2MxZWU5OS05NDc4LTQ0YTYtYjFmMi0zOTFlMjljNmIzNDAiLAogI-
CAgIm1lc3NhZ2VUeXBlIjogIkNSZXEiLAogICAgIm1lc3NhZ2VWZXJzaW9uIjogIjIuMS4wIiwKICAgICJ-
jaGFsbGVuZ2VXaW5kb3dTaXplIjogIjAxIiwKICAgICJtZXNzYWdlRXh0ZW5zaW9uIjog-
WwoJCXsKCQkJIm5hbWUi-
OiAiZW12Y29tc2dleHRJbkNoYWxsZW5nZSIsCgkJCSJpZCI6ICJ0YzhRdG00NjVMbjFGWDBuWn-
ByQSIsCgkJCSJjcml0aWNhbGl0eUluZGljYXRvciI6IGZhbHNlLAoJCQki-
ZGF0YSI6ICJtZXNzYWdlRXh0ZW5zaW9uRGF0YUluQ2hhbGxlbmdlIgoJCX0KICAgIF0KfQ==">
</form>
```

Code Block 7 Challenge Request

You may use the operations `init3DSChallengeRequest` or `createIFrameAndInit3DSChallengeRequest` from the nca3DSWebSDK in order submit the challenge message through the cardholder browser.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script src="nca-3ds-web-sdk.js" type="text/javascript"></script>
  <title>Init 3DS Challenge Request - Example</title>
</head>
<body>
<!-- This example will show how to initiate Challenge Reqeuests for different window
sizes. -->
<div id="frameContainer01"></div>
<div id="frameContainer02"></div>
<div id="frameContainer03"></div>
<div id="frameContainer04"></div>
<div id="frameContainer05"></div>
<iframe id="iframeContainerFull" name="iframeContainerFull" width="100%"
height="100%"></iframe>

<script type="text/javascript">
  // Load all containers
  iFrameContainerFull = document.getElementById('iframeContainerFull');
  container01 = document.getElementById('frameContainer01');
  container02 = document.getElementById('frameContainer02');
  container03 = document.getElementById('frameContainer03');
  container04 = document.getElementById('frameContainer04');
  container05 = document.getElementById('frameContainer05');


  // nca3DSWebSDK.init3DSChallengeRequest(acsUrl, creqData, container);
  nca3DSWebSDK.init3DSChallengeRequest('http://example.com', 'base64-encoded-chal-
lenge-request', iFrameContainerFull);

  // nca3DSWebSDK.createIFrameAndInit3DSChallengeRequest(acsUrl, creqData, chal-
lengeWindowSize, frameName, rootContainer, callbackWhenLoaded);
  nca3DSWebSDK.createIFrameAndInit3DSChallengeRequest('http://example.com', 'base64-
encoded-challenge-request', '01', 'threeDSCReq01', container01);
  nca3DSWebSDK.createIFrameAndInit3DSChallengeRequest('http://example.com', 'base64-
encoded-challenge-request', '02', 'threeDSCReq02', container02);
  nca3DSWebSDK.createIFrameAndInit3DSChallengeRequest('http://example.com', 'base64-
encoded-challenge-request', '03', 'threeDSCReq03', container03);
  nca3DSWebSDK.createIFrameAndInit3DSChallengeRequest('http://example.com', 'base64-
encoded-challenge-request', '04', 'threeDSCReq04', container04);
  nca3DSWebSDK.createIFrameAndInit3DSChallengeRequest('http://example.com', 'base64-
encoded-challenge-request', '05', 'threeDSCReq05', container05, () => {
    console.log('Iframe loaded, form created and submitted');
  });
</script>

</body>
</html>
```

Code Block 8 Init 3DS Challenge Request - Example

Once the cardholder challenge is completed, was canceled or timed out the ACS will instruct the browser to post the results to the notification URL as specified in the challenge request and to send a Result Request (RReq) via the Directory Server to the 3DS Server.

Please note that the notification URL submitted in the challenge request points to EVO E-PAY and must not be changed.

3.2.5 Authorization

After successful cardholder authentication or proof of attempted authentication/verification is provided EVO E-PAY will automatically continue with the payment authorization.

In case the cardholder authentication was not successful or proof of attempted authentication/verification cannot be provided EVO E-PAY will not continue with an authorization request.

In both cases, EVO E-PAY will deliver a final notification to the merchant specified `URLNotify` with the data elements as listed in the table below.

3.2.5.1 Payment Notification

	Key	Format	Condition	Description
1	MID	ans..30	M	Merchant identifier assigned by EVO Payments.
2	MsgVer	ans..5	M	Message version. Accepted values: • `2.0`
3	PayID	ans32	M	Payment/transaction identifier assigned by EVO Payments.
4	XID	an32	M	ID assigned by EVO E-PAY for the operation performed on the payment.
5	TransID	ans..64	M	Transaction identifier supplied by the merchant. Shall be unique for each payment.
6	schemeReferenceID	ans..64	C	Card scheme specific transaction ID required for subsequent credential-on-file payments, delayed authorizations and resubmssions.
7	TrxTime	an21	M	Transaction time stamp in format DD.MM.YYYY HH:mm:ssff.
8	Status	a..20	M	Status of the transaction. Values accepted: • `Authorized` • `OK` (Sale) • `PENDING` • `FAILED` In case of **Authentication-only** the **Status** will be either `OK` or `FAILED`.
9	Description	ans..1024	M	Textual description of the code.
10	Code	n8	M	EVO E-PAY response code.
11	card	JSON	M	Card data.
12	ipInfo	JSON	O	Object containing IP information.
13	threeDSData	JSON	M	Authentication data.
14	resultsResponse	JSON	C	In case the authentication process included a cardholder challenge additional information about the challenge result will be provided.
15	MAC	an64	M	Hash Message Authentication Code (HMAC) with SHA-256 algorithm.

3.2.5.2 Browser Payment Response

Additionally the JSON formatted data elements as listed below are transferred in the HTTP response body to the cardholder browser. Please note that the data elements (i.e. `MID`, `Len`, `Data`) are base64 encoded.

Data Elements

	Key	Format	Condition	Description
1	MID	string	M	Merchant identifier assigned by EVO Payments.
2	Len	integer	M	Length of the unencrypted `Data` string.
3	Data	string	M	Blowfish encrypted string contain a JSON object with `MID`, `PayID` and `TransID`.

Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "MID": {
      "type": "string"
    },
    "Len": {
      "type": "integer"
    },
    "Data": {
      "type": "string"
    }
  },
  "required": ["MID", "Len", "Data"],
  "additionalProperties": false
}
```

Merchants are supposed to forward these data elements to their server for decryption and mapping against the payment notification. Based on the payment results the merchant server may deliver an appropriate response to the cardholder browser (e.g. success page).

Decrypted Data

	Key	Format	Condition	Description
1	MID	ans..30	M	Merchant identifier assigned by EVO Payments.
2	PayID	ans32	M	Payment/transaction identifier assigned by EVO Payments.
3	TransID	ans..64	M	Transaction identifier supplied by the merchant.

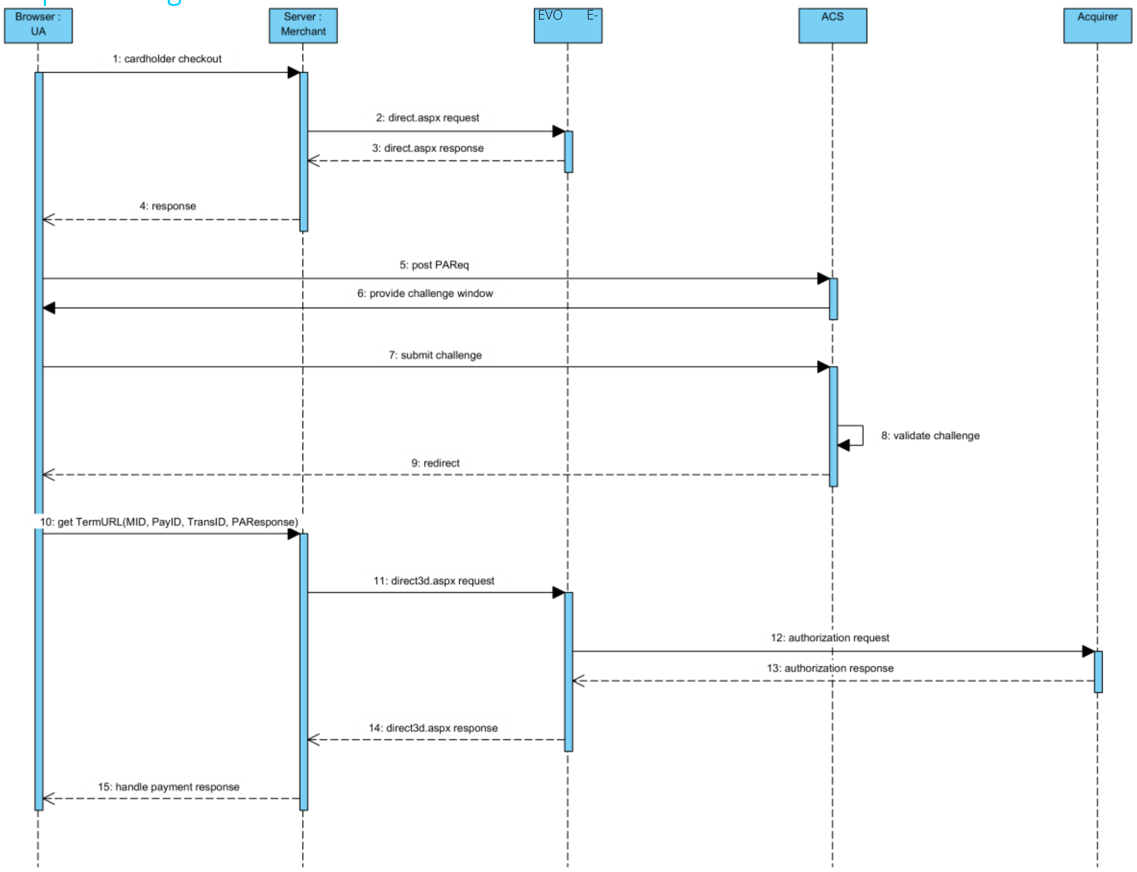
Sample decrypted Data

```
MID=YourMID&PayID=PayIDassignedbyPaygateEVOEPay&TransID=YourTransID
```

3.2.6 3DS 1.0 Fallback EN

In case the Access Control Server (ACS) of the cardholder's bank does not support any EMV 3DS protocol version (i.e. 2.0 or higher, see `acsStartProtocolVersion)` the `threeDSMethodDataForm` element of the versioningData object in the payment response will be Null.

Sequence Diagram


Browser : UA
Server : Merchant
EVO E-
ACS
Acquirer
1: cardholder checkout
2: direct.aspx request
3: direct.aspx response
4: response
5: post PAReq
6: provide challenge window
7: submit challenge
8: validate challenge
9: redirect
10: get TermURL(MID, PayID, TransID, PAResponse)
11: direct3d.aspx request
12: authorization request
13: authorization response
14: direct3d.aspx response
15: handle payment response

3.2.6.1 3DS 1.0 Authentication

In order to a 3DS 1.0 authentication request through the cardholder browser it is required to construct a form with the data elements provided in `threeDSLegacy` and to post it to the `acsURL`.

The form fields that are sent to the ACS are listed in the table below:

	Form Element	Description
1	PAReq	A constructed, Base64 encoded and compressed field carrying the Payer Authentication Request Message Fields. The compression algorithm used is a combination of LZ77 and Huffman coding as specified in RFC 1951.
2	TermURL	The merchant URL the ACS will redirect the cardholder to after the authentication has concluded. Note that EVO E-PAY adds the fields `PayID`, `TransID` and `MID` in the query string to the base URL. Please do not alter the TermURL!
3	MD	The MD (i.e. Merchant Data) field can carry whatever data the merchant needs to continue the session. Please note that this field must be present in the form even though it is not used.

```
<html>
  <head>
    <script language=\"javascript\">
      <!--
        function sendpareq()
          {
            document.pareq_form.submit();
          }
      // -->
    </script>
  </head>

  <body onload="javascript:sendpareq();">
    <form           action="https://pit.3dsecure.net/VbVTestSuiteService/pit1/acsSer-
vice/paReq?summary=ZTIwOWMwYmEtNTVhOC00NDExLThkZDktYzllODk1NmZlNDQ0"    method="POST"
name="pareq_form">
      <input type="hidden" name="PaReq" value="eJxVUst22jAQ/RUfL7rpMZKFiQ0dK4dXgAVOT-
muSpjvVGsApfkSWA+TrK/Fo0t29M6M7M3cEt4di57yhav-
KqjF2/Q10Hy6ySebmJ3VV650Wu02hRSrGrSozdIzbuLYd0qxAnPzBrFXJYY-
tOIDTq5jN1aCIEioyzywkhILwh7gddnFD1JMVyv15HfYz2Xw8PwO75yuPTmp-
nWHAblSo6myrSg1B5G9jhYJD266jHWBXCgUqBYTPk4fR4+M+jdAz-
gEoRYG8zrXGRn+dFb/nzhdR1N+ccQXklIOsa-
kutjpyF5tWVQKt2fKt1PSBkv993sqqoW13VHYlAbA7Ix0gPrUWN0Trkkv+aLVnyvjkuZ6tD8vS8Tya7l/un-
BXt+n8ZAbAVIoZGbMSPaY4HjB4MuHQR9IKc4iMIOwX1KzXpnDLVtMfyU+BwA47syd-
zryfhiZHa4M8FCbM5kKY+U/DBKbjKfGD9PQQiAfC4zn1uFMG+vm+V06bad/Zi+rn6rrJ20xWt4P49h6fiqw8r
nxyo/8s74lQKwEuZyTXP6CQf/9kb8b1MvQ">
      <input  type="hidden"  name="TermUrl"  value="http://localhost:40405/test/3DTer-
mURL.aspx?PayID=dc67820e15f049c9b6c1f0420729da8a&TransID=20180524-162741-084&MID=gus-
tav">
      <input type="hidden" name="MD" value="Optional merchant session data">
    </form>
  </body>
</html>
```

Code Block 9 Sample: PAReq form passed through the Cardholder to the ACS URL

Once the authentication has been completed or the canceled by the cardholder the ACS will redirect the cardholder through the cardholder's browser to the `TermURL` as specified in the initial payment request.

> The Payer Authentication Response (`PaRes`) will be transferred via **HTTP POST** method while `MID`, `PayID` and `TransID` are sent in the HTTP query string (i.e. **HTTP GET**).

Data Elements transferred to the TermURL

	Key	Format	Condition	Description
1	MID	ans..30	M	Merchant identifier assigned by EVO Payments.
2	PayID	ans32	M	Payment/transaction identifier assigned by EVO Payments.
3	TransID	ans..64	M	Transaction identifier supplied by the merchant. Shall be unique for each payment.
4	PARes	--	M	The PARes (Payer Authentication Response) message sent by the ACS in response to the PAReq regardless of whether authentication is successful.

3.2.6.2 Authorization

In order to authorize a 3DS 1.0 authenticated payment you must POST the parameter as listed in the table below to https://spg.evopayments.eu/pay/direct3d.aspx.

Request Elements

	Key	Format	Condition	Description
1	MerchantID	ans..30	M	Merchant identifier assigned by EVO Payments.
2	PayID	ans32	M	Payment/transaction identifier assigned by EVO Payments.
3	TransID	ans..64	M	Transaction identifier supplied by the merchant. Shall be unique for each payment.
4	PAResponse	--	M	The PARes (Payer Authentication Response) message sent by the ACS.

Response Elements

	Key	Format	Condition	Description
1	MID	ans..30	M	Merchant identifier assigned by EVO Payments.
2	PayID	ans32	M	Payment/transaction identifier assigned by EVO Payments.
3	XID	ans64	M	ID assigned by EVO E-PAY for the operation performed on the payment.
4	TransID	ans..64	M	Transaction identifier supplied by the merchant. Shall be unique for each payment.
5	Status	a..20	M	Status of the transaction. Values accepted: • `Authorized` • `OK` (Sale) • `FAILED`
6	Description	ans..1024	M	Textual description of the code.
7	Code	n8	M	EVO E-PAY response code.
8	card	JSON	C	Card data.
9	ipInfo	JSON	O	Object containing IP information.
10	threeDSData	JSON	M	Authentication data.

3.2.6.3 Payer Authentication Request Message Fields EN

The Payer Authentication Request (PAReq) message field is a data element constructed by EVO Payments' Merchant Server Plug-in (MPI).

The MPI builds the XML PAReq, in canonical format according to the DTD. It passes the XML stream to an RFC1951-compliant compressor, which produces an RFC1950-compliant output stream which turn is Base64 encoded.

For educational purposes the PAReq data elements are listed in the table below.

PAReq

	Data Element	Condition	Description
1	Message Version Number	M	Message Version Number as received in the Verify Enrollment Response (VERes).

	Data Element	Condition	Description
			Values accepted: • `1.0.1` • `1.0.2`
2	Acquirer Bank Identification Number (BIN)	M	This field must match the acquirer BIN used in the Verify Enrollment Request.
3	Merchant Identifier (ID) Number	M	This field must match the Merchant ID used in the Verify Enrollment Request. This field also must match the Merchant ID used by the acquirer with the card networks for authorizations and clearing.
4	Merchant Name	M	This field must contain the name of the online merchant at which cardholder is making the purchase. The maximum length is 25 characters. The merchant name must match the name submitted for authorization and clearing.
5	Merchant Country Code	M	This field must contain the ISO 3166 three digit country code value.
6	Merchant URL	M	This field must contain the fully qualified URL of the merchant site.
7	Transaction Identifier	M	Unique transaction identifier determined by merchant. Contains a 20 byte statistically unique value that has been Base64 encoded, giving a 28 byte result.
8	Purchase Date & Time	M	Date and time of purchase expressed in GMT in the following format: YYYYMMDD HH:MM:SS.
9	Purchase Amount	M	This field must contain the value of the purchase being made by the cardholder. It is a value up to 12 digits with punctuation removed.
10	Purchase Currency	M	The appropriate ISO 4217 three-digit currency code for the transaction currency between the cardholder and merchant must be used.
11	Currency Exponent	M	The minor units of currency as defined in ISO 4217.
12	Order Description	O	Brief description of items purchased, determined by the merchant. Maximum size is 125 characters, but merchant should consider the characteristics of the cardholder’s device when creating the field.
13	Recurring Payment Data	C	A Recur element must be included if the merchant and cardholder have agreed to recurring payments.
14	Installment Payment Data	C	An integer greater than one indicating the maximum number of permitted authorizations for installment payments. Must be included if the merchant and cardholder have agreed to installment payments.
15	Account Identifier	M	The content of this field is a data string useful to the ACS; it must not reveal the PAN and must be generated using an algorithm that is likely to generate unique values, even if the same PAN is being presented.
16	Card Expiry Date	M	Expiration Date supplied to merchant by cardholder (YYMM).
17	Message Extension	O	Any data necessary to support the requirements that are not otherwise defined in the PAReq message must be carried in an instance of Message Extension.

Recurring Payment Data

	Data Element	Condition	Description
1	Recurring Frequency	M	An integer indicating the minimum number of days between authorizations.

	Data Element	Condition	Description
2	Recurring Expiry	M	The date after which no further authorizations should be performed. (YYYYMMDD format).

4. JSON Objects EN

Please note that all JSON objects must be **Base64** encoded.
EVO E-PAY validates JSON objects on all requests which contain parameter "MsgVer=2.0". This is independently of the fact whether 3DSecure2 is activated for your MerchantID.
Please make sure no empty parameters or objects are submitted. Under such circumstances EVO E-PAY assumes an error an rejects the transaction.

- accountInfo EN
- card EN
- credentialOnFile EN
- customerInfo EN
- merchantRiskIndicator EN
- priorAuthenticationInfo EN
- threeDSConfig EN
- threeDSPolicy EN

4.1 accountInfo EN

The account information contains optional information about the cardholder account with the merchant.

Cardholder Account Information data elements used to define a time period can be included as either: the *specific date or an approximate indicator* for when the action occurred. 3DS Requestors can use either format.

4.1.1 Data Elements

	Key	Format	Condition	Description
1	accountIdentifier	string	O	The account ID of the cardholder within merchant environment / website (e.g. customer number).
2	authenticationInformation	object	O	This element contains optional information about how the cardholder authenticated during login to their account in the merchant environment (e.g. website).
3	accountAgeIndicator	string	O	Length of time that the customer has had the payment instrument / payment account with the merchant. Values accepted: • `guestCheckout` • `thisTransaction` • `lessThan30Days` • `from30To60Days` • `moreThan60Days`

	Key	Format	Condition	Description
4	accountChangeDate	string	O	Date that the customer's payment instrument (account) with the merchant was last changed, including billing or shipping address, new payment account, or new user(s) added (YYYY-MM-DD).
5	accountChangeIndicator	string	O	Length of time since the customer's account information with the merchant was last changed, including billing or shipping address, new payment account, or new user(s) added. Values accepted: • `thisTransaction` • `lessThan30Days` • `from30To60Days` • `moreThan60Days`
6	accountCreationDate	string	O	Date that the customer opened the account with the merchant in format YYYY-MM-DD.
7	passwordChangeDate	string	O	Date that customer's account with the merchant had a password change or account reset in format YYYY-MM-DD.
8	passwordChangeDateIndicator	string	O	Indicates the length of time since the customer account had a password change or account reset. Values accepted: • `noChange` • `thisTransaction` • `lessThan30Days` • `from30To60Days` • `moreThan60Days`
9	nbrOfPurchases	integer	O	Number of purchases in the last 6 months.
10	addCardAttemptsDay	integer	O	Number of Add Card attempts in the last 24 hours.
11	nbrTransactionsDay	integer	O	Number of transactions (successful and abandoned) in the previous 24 hours.
12	nbrTransactionsYear	integer	O	Number of transactions (successful and abandoned) in the previous year.
13	paymentAccountAge	string	O	Date that the payment account was enrolled in the customer account in format YYYY-MM-DD.
14	paymentAccountAgeIndicator	string	O	Indicates the length of time that the payment account was enrolled in the customer account. Values accepted: • `guestCheckout` • `thisTransaction` • `lessThan30Days` • `from30To60Days` • `moreThan60Days`
15	shipAddressUsageDate	string	O	Date when the shipping address used for this transaction was first used in format YYYY-MM-DD.
16	shipAddressUsageIndicator	string	O	Indicates when the shipping address used for this transaction was first used. Values accepted: • `thisTransaction` • `lessThan30Days` • `from30To60Days` • `moreThan60Days`
17	suspiciousAccActivity	boolean	O	Indicates whether the merchant has experienced suspicious activity (including previous fraud) on the customer account.

4.1.1.1 authenticationInformation

	Key	Format	Condition	Description
1	authenticationData	string	C	This data element can carry specific authentication attestation data such as FIDO if applicable.
2	authenticationMethod	string	M	This data element specifies the mechanism used by the Cardholder to authenticate to the merchant. Values accepted: • `guest` • `merchantCredentials` • `federatedID` • `issuerCredentials` • `thirdPartyAuthentication` • `FIDO` • `signedFIDO` • `SRCassuranceData`
3	authenticationTimestamp	string	M	Date and time (see RFC 3339) in **UTC** of the cardholder authentication. YYYY-MM-DD**T**HH:MM:SS+00:00

4.1.2 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://spg.evopayments.eu/pay/schemas/accountInfo.json",
  "title": "accountInfo",
  "description": "Customer Account Information",
  "type": "object",
  "properties": {
    "accountIdentifier": {
      "type": "string",
      "maxLength": 64
    },
    "authenticationInformation": {
      "type": "object",
      "properties": {
        "authenticationData": {
          "type": "string",
          "maxLength": 20000
        },
        "authenticationMethod": {
          "type": "string",
          "enum": ["guest", "merchantCredentials", "federatedID", "issuerCredentials",
"thirdPartyAuthentication", "FIDO", "signedFIDO", "SRCassuranceData"]
        },
        "authenticationTimestamp": {
          "type": "string",
          "format": "date-time"
        }
      },
      "required": ["authenticationMethod", "authenticationTimestamp"],
      "additionalProperties": false
    },
    "accountAgeIndicator": {
      "type": "string",
      "enum": ["guestCheckout", "thisTransaction", "lessThan30Days", "from30To60Days",
"moreThan60Days"],
      "description": "Length of time that the customer has had the account with the
merchant."
    },
    "accountChangeDate": {
      "type": "string",
      "format": "full-date",
      "description": "YYYY-MM-DD"
    },
    "accountChangeIndicator": {
      "type": "string",
      "enum": ["thisTransaction", "lessThan30Days", "from30To60Days",
"moreThan60Days"],
      "description": "Length of time since the customer account information was last
changed."
    },
    "accountCreationDate": {
      "type": "string",
      "format": "full-date",
      "description": "YYYY-MM-DD"
    },
    "passwordChangeDate": {
      "type": "string",
      "format": "full-date",
      "description": "YYYY-MM-DD"
    },
    "passwordChangeDateIndicator": {
      "type": "string",
      "enum": ["noChange", "thisTransaction", "lessThan30Days", "from30To60Days",
"moreThan60Days"],
      "description": "Indicates the length of time since the customer account had a
password change or account reset."
    },
    "nbrOfPurchases": {
      "type": "integer",
      "maximum": 9999,
      "description": "Number of purchases in the last 6 months."
    },
    "addCardAttemptsDay": {
      "type": "integer",
      "maximum": 999,
      "description": "Number of Add Card attempts in the last 24 hours."
    },
    "nbrTransactionsDay": {
      "type": "integer",
      "maximum": 999,
      "description": "Number of transactions (successful and abandoned) in the previous
24 hours."
    },
    "nbrTransactionsYear": {
      "type": "integer",
      "maximum": 999,
      "description": "Number of transactions (successful and abandoned) in the previous
year."
    },
    "paymentAccountAge": {
      "type": "string",
      "format": "full-date",
      "description": "Date that the payment account was enrolled in the customer ac-
count in format YYYY-MM-DD."
    },
    "paymentAccountAgeIndicator": {
      "type": "string",
      "enum": ["guestCheckout", "thisTransaction", "lessThan30Days", "from30To60Days",
"moreThan60Days"],
      "description": "Indicates the length of time that the payment account was en-
rolled in the customer account."
    },
    "shipAddressUsageDate": {
      "type": "string",
      "format": "full-date",
      "description": "Date when the shipping address used for this transaction was
first used in format YYYY-MM-DD."
    },
    "shipAddressUsageIndicator": {
      "type": "string",
      "enum": ["thisTransaction", "lessThan30Days", "from30To60Days",
"moreThan60Days"],
      "description": "Indicates when the shipping address used for this transaction
was first used."
```

```
    },
    "suspiciousAccActivity": {
      "type": "boolean",
      "description": "Indicates whether the merchant has experienced suspicious activity (including previous fraud) on the customer account."
    }
  },
  "additionalProperties": false
}
```

4.1.3 Sample

```
{
  "accountIdentifier": "joe.bloggs@acme.com"
  "authenticationInformation": {
    "authenticationMethod": "merchantCredentials",
    "authenticationTimestamp": "2021-10-05T04:36:18+00:00"
  },
  "accountAgeIndicator": "moreThan60Days",
  "accountChangeDate": "2019-01-23",
  "accountChangeIndicator": "from30To60Days",
  "accountCreationDate": "2016-01-01",
  "passwordChangeDate": "2018-06-08",
  "passwordChangeDateIndicator": "lessThan30Days",
  "nbrOfPurchases": 4,
  "addCardAttemptsDay": 0,
  "nbrTransactionsDay": 0,
  "nbrTransactionsYear": 5,
  "paymentAccountAge": "2018-03-20",
  "paymentAccountAgeIndicator": "thisTransaction",
  "shipAddressUsageDate": "2017-10-14",
  "shipAddressUsageIndicator": "moreThan60Days",
  "suspiciousAccActivity": true
}
```

4.2 address EN

4.2.1 Data Elements

	Key	Format	Condition	Description
1	city	string	C	City. Required unless market or regional mandate restricts sending this information. For shipping details this data element might not be available (e.g. digital goods).
2	country	object	C	Alpha-3 country code according to ISO 3166-1:2013. Required unless market or regional mandate restricts sending this information. For shipping details this data element might not be available (e.g. digital goods).
3	addressLine1	object	C	First line of the street address. Required unless market or regional mandate restricts sending this information. For shipping details this data element might not be available (e.g. digital goods).

	Key	Format	Condition	Description
4	addressLine2	string	C	Second line of the street address (e.g. apartment, suite, floor, PO Box, etc.). Required unless market or regional mandate restricts sending this information. For shipping details this data element might not be available (e.g. digital goods).
5	addressLine3	string	C	Third line of the street address. Required unless market or regional mandate restricts sending this information. For shipping details this data element might not be available (e.g. digital goods).
6	postalCode	string	C	ZIP or other postal code. Required unless market or regional mandate restricts sending this information. For shipping details this data element might not be available (e.g. digital goods).
7	state	string	C	Alpha-2 code of state or province according to ISO 3166-2. Required unless market or regional mandate restricts sending this information, or State is not applicable for this country. For shipping details this data element might not be available (e.g. digital goods).

4.2.1.1 addressLine1

	Key	Format	Condition	Description
1	street	string	M	Street name.
2	streetNumber	string	C	Street or house number.

4.2.2 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://spg.evopayments.eu/pay/schemas/address.json",
  "title": "address",
  "description": "Address",
  "type": "object",
  "properties": {
    "city": {
      "type": "string"
    },
    "country": {
      "type": "object",
      "properties": {
        "countryName": {
          "type": "string",
          "description": "Name of the country."
        },
        "countryA2": {
          "type": "string",
          "description": "ISO-3166 alpha-2 code."
        },
        "countryA3": {
          "type": "string",
          "description": "ISO 3166-1:2013 alpha-3"
        },
        "countryNumber": {
          "type": "string",
          "description": "ISO-3166 numeric code."
        }
      },
      "required": ["countryA3"],
      "additionalProperties": false
    },
    "addressLine1": {
      "type": "object",
      "properties": {
        "street": {
          "type": "string"
        },
        "streetNumber": {
          "type": "string"
        }
      },
      "required": ["street"],
      "additionalProperties": false
    },
    "addressLine2": {
      "type": "string"
    },
    "addressLine3": {
      "type": "string"
    },
    "postalCode": {
      "type": "string"
    },
    "state": {
      "type": "string",
      "minLength": 2,
      "maxLength": 2,
      "description": "Alpha-2 code of state or province according to ISO 3166-2:2013
where applicable"
    }
  },
  "required": ["country", "addressLine1", "postalCode"],
  "additionalProperties": false
}
```

4.2.3 Sample

```
{
  "city": "New York",
  "country": {
    "countryA3": "USA"
  },
  "addressLine1": {
    "street": "Park Avenue",
    "streetNumber": "270"
  },
  "postalCode": "10017-2070",
  "state": "NY"
}
```

4.3 card EN

- card:request EN
- card:response EN

4.3.1 card:request EN

4.3.1.1 Data Elements

	Key	Format	Condition	Description
1	securityCode	string	O	Card security value.
2	expiryDate	string	M	Card expiry in format YYYYMM.
3	startDate	string	C	Card start in format YYYYMM (only applicable to some UK debit cards).
4	cardholderName	string	M	Name of the cardholder as printed on the card.
5	issueNumber	string	C	Issue number of the card (only applicable to some UK debit cards).
6	number	string	M	Pseudo card number (PCN)/ card token.
7	brand	string	M	Card network.

4.3.1.2 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id":    "https://spg.evopayments.eu/pay/https://www.computop-paygate.com/schemas/card.json",
  "title": "card",
  "description": "Card Information",
  "type": "object",
  "properties": {
    "securityCode": {
      "type": "string",
      "minLength": 3,
      "maxLength": 4
    },
    "expiryDate": {
      "type": "string",
      "description": "YYYYMM",
      "minLength": 6,
      "maxLength": 6
    },
    "startDate": {
      "type": "string",
      "description": "YYYYMM (only applicable to some UK debit cards)",
      "minLength": 6,
      "maxLength": 6
    },
    "cardholderName": {
      "type": "string",
      "maxLength" : 45,
      "minLength" : 2,
      "description" : "The name of the cardholder as printed on the card. Alphanumeric special characters, listed in EMV Book 4, “Appendix B”."
    },
    "issueNumber" : {
      "type" : "string",
      "maxLength" : 2,
      "minLength" : 1,
      "description" : "only applicable to some UK debit cards"
    },
    "number": {
      "type" : "string",
      "maxLength" : 19,
      "minLength" : 12
    },
    "brand": {
      "type": "string",
      "enum": [
        "MasterCard",
        "VISA",
        "AMEX",
        "DINERS",
        "CBN",
        "JCB",
        "Dankort",
        "Maestro",
        "Cartes Bancaires",
        "DISCOVER",
        "Bancontact",
        "Hipercard",
        "Elo",
        "Aura",
        "Carte 4Etoiles",
        "AirPlus",
        "CUP",
        "NARANJA",
        "SHOPPING",
        "CABAL",
        "ARGENCARD",
        "CENCOSUD",
        "KOOKMIN",
        "KEB",
        "BC",
        "SHINHAN",
        "SAMSUNG",
        "HYUNDAI",
        "LOTTE",
        "1euro",
        "echequevacances",
        "cofidis3xcb",
        "cofidis4xcb",
        "facilypay-3x",
        "facilypay-3xsansfrais",
        "facilypay-4x",
        "facilypay-4xsansfrais",
        "RuPay"
      ]
    }
  },
  "required": ["expiryDate", "number", "brand"],
  "additionalProperties": false
}
```

4.3.1.3 Sample

```
{
  "securityCode": "569",
  "expiryDate": "202208",
  "cardholderName": "William Thomas",
  "number": "4186665161011901",
  "brand": "VISA"
}
```

4.3.2 card:response EN

4.3.2.1 Data Elements

	Key	Format	Condition	Description
1	cardholderName	string	C	Presence depends on merchant configuration. Name of the cardholder as printed on the card.
2	number	string	C	Presence depends on merchant configuration. If present this element entails either the masked card number or EVO E-PAY card token.

	Key	Format	Condition	Description
3	expiryDate	string	C	Present if number entails EVO E-PAY card token.
4	bin	object	M	Bank Identification Number including account range if applicable.
5	brand	string	M	Card network name (e.g. 'Visa', 'Mastercard').
6	product	string	C	Card product name (if available) (e.g. 'Business Premium Debit').
7	source	string	C	Card funding source (if available). Values accepted: • `DEBIT` • `CREDIT` • `DEFERRED DEBIT` • `PREPAID` • `CHARGE`
8	type	string	C	The card type specifies the program, application or card level attached to the card if any (e.g. Classic, Standard, Gold, Business etc.).
9	country	JSON	M	Country the card was issued in.
10	issuer	string	C	Card issuer (if available).

bin

	Key	Format	Condition	Description
1	accountBIN	string	M	The leading six digits of the account number also known as bank identification number (**BIN**).
2	accountRangeLow	string	C	The account number at the low end of the account range.
3	accountRangeHigh	string	C	The account number at the high end of the account range.

4.3.2.2 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id":   "https://spg.evopayments.eu/pay/https://www.computop-paygate.com/schemas/card_response.json",
  "title": "card",
  "description": "Card Information",
  "type": "object",
  "properties": {
    "cardholderName": {
      "type": "string",
      "maxLength" : 45,
      "minLength" : 2,
      "description" : "The name of the cardholder as printed on the card. Alphanumeric special characters, listed in EMV Book 4, “Appendix B”."
    },
    "number": {
      "type" : "string",
      "maxLength" : 19,
      "minLength" : 12
    },
    "expiryDate": {
      "type": "string",
      "description": "YYYYMM",
      "minLength": 6,
      "maxLength": 6
    },
    "bin": {
      "type": "object",
      "properties": {
        "accountBin": { "type": "string" },
        "accountRangeLow": { "type": "string" },
        "accountRangeHigh": { "type": "string" }
      },
      "additionalProperties": false,
      "required": ["accountBin"]
    },
    "brand": {
      "type": "string",
      "enum": [
        "MasterCard",
        "VISA",
        "AMEX",
        "Diners",
        "CBN",
        "JCB",
        "Dankort",
        "Maestro",
        "Cartes Bancaires",
        "DISCOVER",
        "Bancontact",
        "Hipercard",
        "Elo",
        "Aura",
        "Carte 4Etoiles",
        "AirPlus",
        "CUP",
        "NARANJA",
        "SHOPPING",
        "CABAL",
        "ARGENCARD",
        "CENCOSUD",
        "KOOKMIN",
        "KEB",
        "BC",
        "SHINHAN",
        "SAMSUNG",
        "HYUNDAI",
        "LOTTE",
        "1euro",
        "echequevacances",
        "cofidis3xcb",
        "cofidis4xcb",
        "facilypay-3x",
        "facilypay-3xsansfrais",
        "facilypay-4x",
        "facilypay-4xsansfrais",
        "RuPay"
      ]
    },
    "product": {
      "type": "string"
    },
    "source": {
      "type": "string",
      "enum": ["DEBIT", "CREDIT", "DEFERRED DEBIT", "PREPAID", "CHARGE"]
    },
    "type": {
      "type": "string"
    },
    "country": {
      "type": "object",
      "properties": {
        "countryName": {
          "type": "string",
          "description": "Name of the country."
        },
        "countryA2": {
          "type": "string",
          "description": "ISO-3166 alpha-2 code."
        },
        "countryA3": {
          "type": "string",
          "description": "ISO 3166-1:2013 alpha-3"
        },
        "countryNumber": {
          "type": "string",
          "description": "ISO-3166 numeric code."
        }
      },
      "required": ["countryA3"],
      "additionalProperties": false
    },
    "issuer": {
      "type": "string"
```

```
      }
    },
    "required": ["bin", "brand", "country"],
    "additionalProperties": false
  }
```

4.3.2.3 Sample

```
  {
    "cardholderName": "John Doe",
    "bin": {
      "accountBin": "492947",
      "accountRangeLow": "492947000000",
      "accountRangeHigh": "492948387999"
    },
    "brand": "VISA",
    "product": "Business",
    "source": "CREDIT",
    "type": "CLASSIC",
    "country": {
      "countryName": "United Kingdom of Great Britain and Northern Ireland",
      "countryA2": "GB",
      "countryA3": "GBR",
      "countryNumber": "826"
    },
    "issuer": "BARCLAYS BANK PLC"
  }
```

4.4 credentialOnFile EN

	Key	Format	Condition	Description
1	type	object	M	Type of credential on file payment.
2	initialPayment	boolean	M	Indicates whether a card on file transaction is the first one in a series of transactions (establishment) or a subsequent transaction.

4.4.1 type

	Key	Format	Condition	Description
1	recurring	object	C	Recurring payments are a series of transactions processed pursuant to an agreement between a cardholder and a merchant where the cardholder purchases goods or services over a period of time through a number of separate transactions. Please note that in the context of PSD2 and SCA requirements the European Banking Authority (EBA) describes recurring payments as a series of transactions with the same amount and with the same payee.
2	unscheduled	string	C	Value indicating the party that initiates a credential on file transaction that does not occur on a fixed schedule. Values accepted: • `CIT` = Customer Initiated Transaction • `MIT` = Merchant Initiated Transaction

4.4.1.1 recurring

	Key	Format	Condition	Description
1	recurringFrequency	integer	M	Indicates the number of days between authorizations.
2	recurringStartDate	string	O	Determines the date of the first authorization according to the recurring mandate.
3	recurringExpiryDate	string	M	Date after which no further authorizations shall be performed.

4.4.2 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://spg.evopayments.eu/pay/schemas/credentialOnFile.json",
  "title": "credentialOnFile",
  "description": "Credential-on-File Transactions",
  "type": "object",
  "properties": {
    "type": {
      "type": "object",
      "properties": {
        "recurring": {
          "type": "object",
          "properties": {
            "recurringFrequency": {
              "type": "integer",
              "minimum": 1,
              "maximum": 9999,
              "description": "Indicates the minimum number of days between recurring authorizations."
            },
            "recurringStartDate": {
              "type": "string",
              "format": "full-date",
              "description": "YYYY-MM-DD"
            },
            "recurringExpiryDate": {
              "type": "string",
              "format": "full-date",
              "description": "YYYY-MM-DD"
            }
          },
          "required": ["recurringExpiryDate", "recurringFrequency"],
          "additionalProperties": false
        },
        "unscheduled": {
          "type": "string",
          "enum": ["CIT", "MIT"]
        }
      },
      "oneOf": [
        {"required": ["recurring"]},
        {"required": ["installments"]},
        {"required": ["unscheduled"]}
      ],
      "additionalProperties": false
    },
    "initialPayment": {
      "type": "boolean"
    }
  },
  "required": ["type", "initialPayment"],
  "additionalProperties": false
}
```

4.4.3 Sample Recurring

```
{
  "type": {
    "recurring": {
      "recurringFrequency": 30,
      "recurringStartDate": "2019-09-14",
      "recurringExpiryDate": "2020-09-14"
    }
  },
  "initialPayment": true
}
```

4.4.4 Sample Unscheduled CIT

```
{
  "type": {
    "unscheduled": "CIT"
  },
  "initialPayment": false
}
```

4.5 customerInfo EN

	Key	Format	Condition	Description
1	consumer	object	C	Object describing private customers. Required if the customer is a person.
2	business	object	C	Object describing business customers. Required if the customer is a legal entity.
3	phone	object	C	Phone number. Required (if available), unless market or regional mandate restricts sending this information.
4	mobilePhone	object	C	Mobile phone number. Required (if available) unless market or regional mandate restricts sending this information.
5	email	string	C	Email address. Required unless market or regional mandate restricts sending this information.

4.5.1 consumer

	Key	Format	Condition	Description
1	salutation	string	O	Salutation Values accepted: • Mr • Mrs • Miss
2	firstName	string	M	Customers' first name.
3	lastName	string	M	Customer's last name.
4	birthDate	string	O	Customer's birthdate in format YYYY-MM-DD.

4.5.2 business

	Key	Format	Condition	Description
1	legalName	string	M	Business legal name.
2	dbaName	string	O	Doing Business As.
3	registrationNumber	string	O	Business registration number.

4.5.3 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://spg.evopayments.eu/pay/schemas/customerInfo.json",
  "title": "customerInfo",
  "description": "Customer Information",
  "type": "object",
  "properties": {
    "consumer": {
      "properties": {
        "salutation": {
          "type": "string",
          "enum": ["Mr", "Mrs", "Miss"]
        },
        "firstName": {
          "type": "string",
          "maxLength": 30
        },
        "lastName": {
          "type": "string",
          "maxLenght": 30
        },
        "birthDate": {
          "type": "string",
          "format": "full-date",
          "description": "YYYY-MM-DD"
        }
      },
      "required": ["firstName", "lastName"],
      "additionalProperties": false
    },
    "business": {
      "properties": {
        "legalName": {
          "type": "string",
          "maxLenght": 50
        },
        "dbaName": {
          "type": "string",
          "maxLenght": 50,
          "description": "Doing business as. Name of the company as usually known to
consumers."
        },
        "registrationNumber": {
          "type": "string",
          "maxLenght": 20
        }
      },
      "required": ["legalName"],
      "additionalProperties": false
    },
    "phone": {
      "type": "object",
      "properties": {
        "countryCode": {
          "type": "string",
          "minLength": 1,
          "maxLenght": 3
        },
        "subscriberNumber": {
          "type": "string",
          "maxLenght": 12
        }
      },
      "required": ["countryCode", "subscriberNumber"],
      "additionalProperties": false
    },
    "mobilePhone": {
      "type": "object",
      "properties": {
        "countryCode": {
          "type": "string",
          "minLength": 1,
          "maxLenght": 3
        },
        "subscriberNumber": {
          "type": "string",
          "maxLenght": 12
        }
      },
      "required": ["countryCode", "subscriberNumber"],
      "additionalProperties": false
    },
    "email": {
      "type": "string",
      "maxLenght": 50,
      "format": "idn-email"
    }
  },
  "oneOf": [
    {"required": ["consumer"]},
    {"required": ["business"]}
  ],
  "additionalProperties": false
}
```

4.5.4 Sample

```
{
  "consumer": {
    "salutation": "Mr",
    "firstName": "Napoleon",
    "lastName": "Bonaparte",
    "birthDate": "1769-08-15"
  },
  "mobilePhone": {
    "countryCode": "33",
    "subscriberNumber" : "12345678910"
  },
  "email": "napoleon.bonaparte@france.com"
}
```

4.6 ipInfo EN

	Key	Format	Condition	Description
1	ipAddress	string	M	IP address.
2	country	object	M	Country of IP origin.
3	state	string	C	States and provinces (that is, the first-level administrative division) in all countries where they exist.
4	city	string	M	City, localized spelling.
5	longitude	string	M	The longitude of the identified location, expressed as a floating point number with range of -180 to 180, with positive numbers representing East and negative numbers representing West.
6	latitude	string	M	The latitude of the identified location, expressed as a floating point number with range of -90 to 90, with positive numbers representing North and negative numbers representing South. Latitude and longitude are derived from the city or postal code.

4.6.1 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://www.computop-paygate.com/schemas/ipInfo.json",
  "title": "ipInfo",
  "description": "IP Information",
  "type": "object",
  "properties": {
    "ipAddress": {
      "type": "string",
      "oneOf": [{"format": "ipv4"},{"format": "ipv6"}]
    },
    "country": {
      "type": "object",
      "properties": {
        "countryName": {
          "type": "string"
        },
        "countryA2": {
          "type": "string",
          "minLength": 2,
          "maxLength": 2
        },
        "countryA3": {
          "type": "string",
          "minLength": 3,
          "maxLength": 3
        },
        "countryNumber": {
          "type": "string",
          "minLength": 3,
          "maxLength": 3
        }
      },
      "required": ["countryName", "countryA2", "countryA3", "countryNumber"],
      "additionalproperties": false
    },
    "state": {
      "type": "string"
    },
    "city": {
      "type": "string"
    },
    "longitude": {
      "type": "string"
    },
    "latitude": {
      "type": "string"
    }
  },
  "required": ["ipAddress", "country", "city", "longitude", "latitude"],
  "additionalproperties": false
}
```

4.6.2 Sample

```
{
  "ipAddress": "178.37.173.82",
  "country": {
    "countryName": "poland",
    "countryA2": "pl",
    "countryA3": "pol",
    "countryNumber": "616"
  },
  "state": "wielkopolskie",
  "city": "poznan",
  "longitude": "16.83739",
  "latitude": "52.4136"
}
```

4.7 merchantRiskIndicator EN

	Key	Format	Condition	Description
1	deliveryEmail	string	O	For Electronic delivery, the e-mail address to which the merchandise was delivered.
2	deliveryTimeframe	string	O	Indicates the merchandise delivery timeframe.
3	giftCardAmount	integer	O	For prepaid or gift card purchase, the purchase amount total of prepaid or gift cards in smallest currency unit.
4	giftCardCount	integer	O	For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased.
5	giftCardCurr	string	O	For prepaid or gift card purchase, ISO 4217 three-letter currency code of the gift card.
6	preOrderDate	string	O	For a pre-ordered purchase, the expected date that the merchandise will be available (YYYY-MM-DD).
7	preOrderPurchaseIndicator	boolean	O	Indicates whether customer is placing an order for merchandise with a future availability or release date.
8	reorderItemsIndicator	boolean	O	Indicates whether the customer is reordering previously purchased merchandise.
9	shippingAddressIndicator	string	O	Indicates shipping method chosen for the transaction. If one or more items are included in the sale, use the Shipping Indicator code for the physical goods, or if all digital goods, use the Shipping Indicator code that describes the most expensive item.

4.7.1 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://spg.evopayments.eu/pay/schemas/merchantRisk.json",
  "title": "merchantRisk",
  "description": "Merchant Risk Information",
  "type": "object",
  "properties": {
    "deliveryEmail": {
      "type": "string",
      "maxLength": 50,
      "format": "idn-email",
      "description": "For Electronic delivery, the email address to which the merchandise was delivered."
    },
    "deliveryTimeframe": {
      "type": "string",
      "enum": ["electronicDelivery", "sameDayDelivery", "nextDayDelivery", "twoOrMoreDaysDelivery"]
    },
    "giftCardAmount": {
      "type": "integer",
      "maximum": 999999999999
    },
    "giftCardCount": {
      "type": "integer",
      "maximum": 99,
      "description": "Number of prepaid or gift cards used for this purchase."
    },
    "giftCardCurr": {
      "type": "string",
      "minLength": 3,
      "maxLength": 3,
      "description": "ISO 4217 three-letter currency code of the gift card."
    },
    "preOrderDate": {
      "type": "string",
      "format": "full-date",
      "description": "The expected date that the merchandise will be available in format YYYY-MM-DD."
    },
    "preOrderPurchaseIndicator": {
      "type": "boolean",
      "description": "Indicates whether customer is placing an order for merchandise with a future availability or release date."
    },
    "reorderItemsIndicator": {
      "type": "boolean",
      "description": "Indicates whether the customer is reordering previously purchased merchandise."
    },
    "shippingAddressIndicator": {
      "type": "string",
      "enum": [
        "shipToBillingAddress",
        "shipToVerifiedAddress",
        "shipToNewAddress",
        "shipToStore",
        "digitalGoods",
        "noShipment",
        "other"
      ],
      "description": "Indicates shipping method chosen for the transaction. If one or
more items are included in the sale, use the Shipping Indicator code for the physical
goods, or if all digital goods, use the Shipping Indicator code that describes the most
expensive item."
    }
  },
  "additionalProperties": false
}
```

4.7.2 Sample

```
{
  "deliveryEmail": "joe.bloggs@acme.com",
  "deliveryTimeframe": "twoOrMoreDaysDelivery",
  "giftCardAmount": 5000,
  "giftCardCount": 2,
  "giftCardCurr": "EUR",
  "preOrderDate": "2020-03-15",
  "preOrderPurchaseIndicator": true,
  "reorderItemsIndicator": true,
  "shippingAddressIndicator": "shipToStore"
}
```

4.8 priorAuthenticationInfo EN

	Key	Format	Condition	Description
1	priorAuthenticationData	string	O	Data that documents and supports a specific authentication process performed by the merchant such as FIDO.
2	priorAuthenticationMethod	string	O	Mechanism used by the Cardholder to previously authenticate. Values accepted: • `frictionless` • `ACSchallenge` • `AVSverified` • `other`
3	priorAuthenticationTimestamp	string	O	Date and time (see RFC 3339) in **UTC** of the prior cardholder authentication. YYYY-MM-DD**T**HH:MM:SS+00:00
4	priorAuthenticationReference	string	O	This data element contains an ACS Transaction ID for a prior authenticated transaction (for example, the first recurring transaction that was authenticated with the cardholder).

4.8.1 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://spg.evopayments.eu/pay/schemas/priorAuthenticationInformation.json",
  "title": "Prior Authentication Information",
  "type": "object",
  "properties": {
    "prioAuthenticationData": {
      "type": "string",
      "maxLength": 2048
    },
    "priorAuthenticationMethod": {
      "type": "string",
      "enum": ["frictionless", "ACSchallenge", "AVSverified", "other"]
    },
    "priorAuthenticationTimestamp": {
      "type": "string",
      "format": "date-time"
    },
    "priorAuthenticationReference": {
      "type": "string",
      "maxLength": 36
    }
  },
  "additionalProperties": false
}
```

4.8.2 Sample

```
{
  "priorAuthenticationMethod": "frictionless",
  "priorAuthenticationTimestamp": "2021-10-05T04:36:18+00:00",
  "priorAuthenticationReference": "d7c1ee99-9478-44a6-b1f2-391e29c6b340"
}
```

4.9 resultsResponse EN

Please note that all data elements listed below will be present in `resultsResponse` but might hold an *empty string* based on the *condition*.

	Key	Format	Condition	Description
1	threeDSServerTransID	string	M	EVO E-PAY PayID in canonical format as specified in IETF RFC 4122.
2	acsTransID	string	M	Universally Unique transaction identifier assigned by the ACS to identify a single transaction.
3	acsRenderingType	object	C	Required unless ACS Decoupled Confirmation = true.

	Key	Format	Condition	Description
4	authenticationType	string	C	Required if the Transaction Status = Y or N. Indicates the type of authentication method the Issuer will use to challenge the cardholder. Required if the Transaction Status = C or D. Values accepted: • `01 = static` • `02 = dynamic` • `03 = oob` Future implementation. Protocol Version 2.2.0 onwards - • `04 = decoupled`
5	authenticationValue	string	C	Required if Transaction Status = Y or A.
6	challengeCancel	string	C	Indicator informing that the authentication has been canceled. Values accepted: • `01` = Cardholder selected “Cancel” • `02` = Reserved for future EMVCo use (values invalid until defined by EMVCo). • `03` = Transaction Timed Out—Decoupled Authentication • `04` = Transaction Timed Out at ACS—other timeouts • `05` = Transaction Timed Out at ACS—First CReq not received by ACS • `06` = Transaction Error • `07` = Unknown • `08` = Transaction Timed Out at SDK
7	dsTransID	string	M	Universally unique transaction identifier assigned by the DS to identify a single transaction.
8	eci	string	C	Payment System-specific value provided by the ACS or DS to indicate the results of the attempt to authenticate the Cardholder. The requirements for the presence of this field are DS specific.
9	interactionCounter	string	M	Indicates the number of authentication cycles attempted by the Cardholder.
10	messageCategory	string	M	Identifies the category of the message for a specific use case. Values accepted: • `01` = PA • `02` = NPA
11	messageExtension	string	C	Data necessary to support requirements not otherwise defined in the 3-D Secure message are carried in a Message Extension. Conditions to be set by each DS.
12	messageType	string	C	Identifies the type of message that failed in case of an error. Values accepted: • `ARes` • `RReq`
13	messageVersion	string	M	Protocol version identifier.
14	sdkTransID	string	M	Future use. Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction.
15	transStatus	string	M	Indicates whether a transaction qualifies as an authenticated transaction. Values accepted: • `Y` = Authentication Verification Successful. • `N` = Not Authenticated /Account Not Verified; Transaction denied. • `U` = Authentication/ Account Verification Could Not Be Performed; Technical or other problem, as indicated in ARes or RReq. • `A` = Attempts Processing Performed; Not Authenticated/Verified, but a proof of attempted authentication/verification is provided.

	Key	Format	Condition	Description
				- C = Challenge Required; Additional authentication is required using the CReq/CRes. - D = Challenge Required; Decoupled Authentication confirmed. - R = Authentication/ Account Verification Rejected; Issuer is rejecting authentication/verification and request that authorization not be attempted. - I = Informational Only; 3DS Requestor challenge preference acknowledged.
16	transStatusReason	string	C	Provides information on why the Transaction Status field has the specified value. Required if the Transaction Status field = N, U, or R. Values accepted: - 01 = Card authentication failed - 02 = Unknown Device - 03 = Unsupported Device - 04 = Exceeds authentication frequency limit - 05 = Expired card - 06 = Invalid card number - 07 = Invalid transaction - 08 = No Card record - 09 = Security failure - 10 = Stolen card - 11 = Suspected fraud - 12 = Transaction not permitted to cardholder - 13 = Cardholder not enrolled in service - 14 = Transaction timed out at the ACS - 15 = Low confidence - 16 = Medium confidence - 17 = High confidence - 18 = Very High confidence - 19 = Exceeds ACS maximum challenges - 20 = Non-Payment transaction not supported - 21 = 3RI transaction not supported - 22 = ACS technical issue - 23 = Decoupled Authentication required by ACS but not requested by 3DS Requestor - 24 = 3DS Requestor Decoupled Max Expiry Time exceeded - 25 = Decoupled Authentication was provided insufficient time to authenticate cardholder. ACS will not make attempt - 26 = Authentication attempted but not performed by the cardholder
17	whiteListStatus	string	C	**Future use.** Only supported with protocol version 2.2.0 onwards. Enables the communication of trusted beneficiary/whitelist status. Values accepted: - Y = 3DS Requestor is whitelisted by cardholder - N = 3DS Requestor is not whitelisted by cardholder - E = Not eligible as determined by issuer - P = Pending confirmation by cardholder - R = Cardholder rejected - U = Whitelist status unknown, unavailable, or does not apply
18	whiteListStatusSource	string	C	**Future use.** Only supported with protocol version 2.2.0 onwards. This data element will be populated by the system setting Whitelist Status. Values accepted: - 01 = 3DS Server

	Key	Format	Condition	Description
				• 02 = DS • 03 = ACS

4.9.1 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://www.computop-paygate.com/schemas/resultsResponse.json",
  "type": "object",
  "properties": {
    "threeDSServerTransID": {
      "type": "string",
      "maxLength": 36
    },
    "acsTransID": {
      "type": "string",
      "maxLength": 36
    },
    "acsRenderingType": {
      "type": "object",
      "properties": {
        "acsInterface": {
          "type": "string",
          "enum": ["native", "html", ""],
          "description": "The ACS interface that the challenge will present to the
cardholder."
        },
        "acsUiTemplate": {
          "type": "string",
          "enum": ["text", "singleSelect", "multiSelect", "oob", "other",""],
          "description": "Identifies the UI Template format that the ACS first presents
to the consumer."
        }
      },
      "required": ["acsInterface", "acsUiTemplate"],
      "additionalProperties": false
    },
    "authenticationType": {
      "type": "string",
      "enum": ["01", "02", "03", "04", ""]
    },
    "authenticationValue": {
      "type": "string",
      "maxLength": 28
    },
    "challengeCancel": {
      "type": "string",
      "enum": ["01", "02", "03", "04", "05", "06", "07", "08", ""]
    },
    "dsTransID": {
      "type": "string",
      "maxLength": 36
    },
    "eci": {
      "type": "string",
      "maxLength": 2
    },
    "interactionCounter": {
      "type": "string",
      "maxLength": 2
    },
    "messageCategory": {
      "type": "string",
      "enum": ["01", "02"]
    },
    "messageExtension": {
      "type": "string",
      "maxLength": 81920
    },
    "messageVersion": {
      "type": "string",
      "minLength": 5,
      "maxLength": 8
    },
    "sdkTransID": {
      "type": "string",
      "maxLength": 36
    },
    "transStatus": {
      "type": "string",
      "enum": ["Y", "N", "U", "A", "C", "D", "R", "I", ""]
    },
    "transStatusReason": {
      "type": "string",
      "enum": ["01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", ""]
    }
  },
  "required": ["threeDSServerTransID", "acsTransID", "acsRenderingType", "authenticationType", "authenticationValue", "challengeCancel", "dsTransID", "eci", "interactionCounter", "messageCategory", "messageExtension", "messageVersion", "sdkTransID", "transStatus", "transStatusReason"],
  "additionalProperties": false
}
```

4.9.2 Sample

```
{
  "threeDSServerTransID":"9e944d5d-56f3-461d-a393-80a666d346d1",
  "acsTransID":"1e43b52f-3623-4e5d-8917-41c5c15b7218",
  "acsRenderingType":{
    "acsInterface":"01",
    "acsUiTemplate":"01"
  },
  "authenticationType":"02",
  "authenticationValue":"JAmi21makAifmwqo2120cjq1AAA=",
  "challengeCancel":"",
  "dsTransID":"c626e8a0-f2ba-42b3-aa6d-620658421f3a",
  "eci":"05",
  "interactionCounter":"01",
  "messageCategory":"01",
  "messageExtension":"",
  "messageVersion":"2.1.0",
  "sdkTransID":"",
  "transStatus":"Y",
  "transStatusReason":""
}
```

4.10 threeDSData EN

- threeDSData:response EN

4.10.1 threeDSData:response EN

	Key	Format	Condition	Description
1	authenticationStatus	boolean	M	Indicates whether a cardholder has been authenticated or not.
2	acsProtocolVersion	string	M	The protocol version used for authentication.
3	authenticationValue	string	C	Payment system specific value to provide proof of authentication.
4	eci	string	M	Payment system specific Electronic Commerce Indicator.
5	threeDSServerTransID	string	C	3DS 2.0 only. EVO E-PAY PayID in canonical format as specified in IETF RFC 4122.
6	acsXID	string	C	3DS 1.0 only. ACS assigned transaction ID.

4.10.1.1 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://www.computop-paygate.com/schemas/threeDSData_response.json",
  "title": "3DS Data",
  "description": "3DS Data",
  "type": "object",
  "properties": {
    "authenticationStatus": {
      "type": "boolean"
    },
    "acsProtocolVersion": {
      "minLength": 5,
            "maxLength": 8
    },
    "authenticationValue": {
            "type": "string",
            "maxLength": 28
        },
    "eci": {
      "type": "string",
      "minLength": 2,
      "maxLength": 2
    },
    "threeDSServerTransID": {
      "type": "string",
            "maxLength": 36
        },
    "ACSXID": {
      "type": "string",
      "maxLength": 40
    }
  },
  "additionalProperties": false,
  "required": ["authenticationStatus", "acsProtocolVersion", "eci"]
}
```

4.11 threeDSConfig EN

This data element can be used to override configuration data on file for the Merchant ID.

	Key	Format	Condition	Description
1	loginID	string	M	3DS Login ID.
2	acquirerBIN	string	M	Acquirer’s Bank Identification Number.
3	merchantCategoryCode	string	M	Merchant Category Code (MCC).
4	merchantCountry	object	M	Merchant country.
5	merchantName	string	M	Merchant name.

4.11.1 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://spg.evopayments.eu/pay/schemas/threeDSConfig.json",
  "title": "3DS Config",
  "description": "3DS Configuration Data",
  "type": "object",
  "properties": {
    "loginID": {
      "type": "string"
    },
    "acquirerBIN": {
      "type": "string",
      "minLength": 6,
      "maxLength": 8
    },
    "merchantCategoryCode": {
      "type": "string",
      "minLength": 4,
      "maxLength": 4
    },
    "merchantCountry": {
      "type": "object"
    },
    "merchantName": {
      "type": "string"
    }
  },
  "additionalProperties": false,
  "required":  ["loginID",  "acquirerBIN",  "merchantCategoryCode",  "merchantCountry",
"merchantName"]
}
```

4.11.2 Sample

```
{
  "loginID": "123456ABCDEF",
  "acquirerBIN": "123456",
  "merchantCategoryCode": "5965",
  "merchantCountry": {
    "countryA3": "pol"
  },
  "merchantName": "ACME INC."
}
```

4.12 threeDSPolicy EN

	Key	Format	Condition	Description
1	skipThreeDS	string	O	Indicates whether and under which conditions authentication should be skipped or performed as data share only (dataOnly). By default, all transactions and SCA exemptions will be requested through EMV 3DS if not specified otherwise. Values accepted: • `thisTransaction` • `outOfScope` • `dataOnly`
2	threeDSExemption	object	O	Object detailing requested SCA exemptions.

4.12.1 threeDSExemption

	Key	Format	Condition	Description
1	exemptionReason	string	M	Designates the type of SCA exemption (e.g. Acquirer TRA or MIT) to be applied. Values accepted: • `transactionRiskAnalysis` • `delegatedAuthority` • `merchantInitiatedTransaction` • `lowValue` Note: Acquirer exemptions and Merchant Initiated Transactions (MIT) may be also requested through an authorization without authentication (i.e. EMV 3DS or EMV 3DS Data Only). Note: `merchantInitiatedTransactionis` only valid in combination with credentialOnFile.
2	merchantFraudRate	integer	O	Merchant fraud rate in bps taking into account all Merchant sites and card volumes, calculated as per PSD2 RTS Article 19. The merchant fraud rate is optional and has to be calculated by the Acquirer. The submission of this data point might be beneficial to increase the level of confidence of the ACS/issuer into the ongoing transaction. Also, Issuers may use it to decide if a merchant should be eligible for the white listing exemption.

4.12.2 Scheme

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://spg.evopayments.eu/pay/schemas/threeDSPolicy.json",
  "title": "threeDSPolicy",
  "description": "3DS Policy",
  "type": "object",
  "properties": {
    "skipThreeDS": {
      "type": "string",
      "enum": ["thisTransaction", "outOfScope", "dataOnly"]
    },
    "threeDSExemption": {
      "type": "object",
      "properties": {
        "exemptionReason": {
          "type": "string",
          "enum": ["transactionRiskAnalysis", "delegatedAuthority", "merchantInitiatedTransaction", "lowValue"]
        },
        "merchantFraudRate": {
          "type": "integer",
          "minimum": 1,
          "maximum": 99
        }
      },
      "required": ["exemptionReason"],
      "additionalProperties": false
    }
  },
  "additionalProperties": false
}
```

4.12.3 Sample

```
{
  "skipThreeDS": "outOfScope",
  "threeDSExemption": {
    "exemptionReason": "merchantInitiatedTransaction",
    "merchantFraudRate": 4
  }
}
```

5. Important Notes EN

- 3DS Authentication Hosting EN
- Account Verification EN
- Amazon Payments MFA EN
- Dynamic Billing Descriptors EN
- Mandatory and Conditional Required Data Elements for EMV 3DS EN
- Multi-party Ecommerce / Agent Model EN
- Non-payment authentications for Card Add EN
- schemeReferenceID EN

- Stored Credentials EN

5.1 Dynamic Billing Descriptors EN

5.1.1 General Requirements

The `billingDescriptor` element is used to override the merchant name that is sent to a cardholder's bank where applicable.

The merchant name is the most important factor in cardholder recognition of a transaction record typically printed on a cardholder statement. It should reflect the operating name of a company (i.e. 'Doing Business As' (DBA) name), as opposed to the legal name, by which a cardholder would recognize the merchant to avoid any confusion and to minimize copy requests.

As a standard, acquirers usually forward the merchant name they got on file with the merchant's account. Please note that the card organizations established strict rules around merchant names on cardholder statements.

However, there are a number of specific exceptions where supplementary data can be added to the DBA name depending on the use case and industry (e.g. airlines, railway services, car rental, fuel stations etc.).

5.1.2 Formatting the merchant name

The authorization and clearing systems of the card organizations provide varying sizes for merchant names. The smallest common denominator is 22 characters. Thus, merchant names longer than 22 characters will not fit into the merchant name field and must be abbreviated in a way that it is still recognizable to the cardholder.

5.1.2.1 Purchase of Goods or Services

For regular purchases of goods and services additional information may be included after the merchant name and an **asterisk (*)** to indicate an order number, reference number, or other information to identify a transaction.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
G	R	E	A	T		B	R	A	N	D		L	T	D	*	0	8	1	5	3	7

For vehicle rental and hotel merchants, the merchant name must not be truncated in order to place supplemental information into the merchant name field.

5.1.2.2 No-Show Transactions

May also include the words "NO SHOW" after the merchant name.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
H	.	C	A	L	I	F	O	R	N	I	A		N	O		S	H	O	W		

5.1.2.3 Purchase of an airline Ticket (or passenger railway tickets in the US Region)

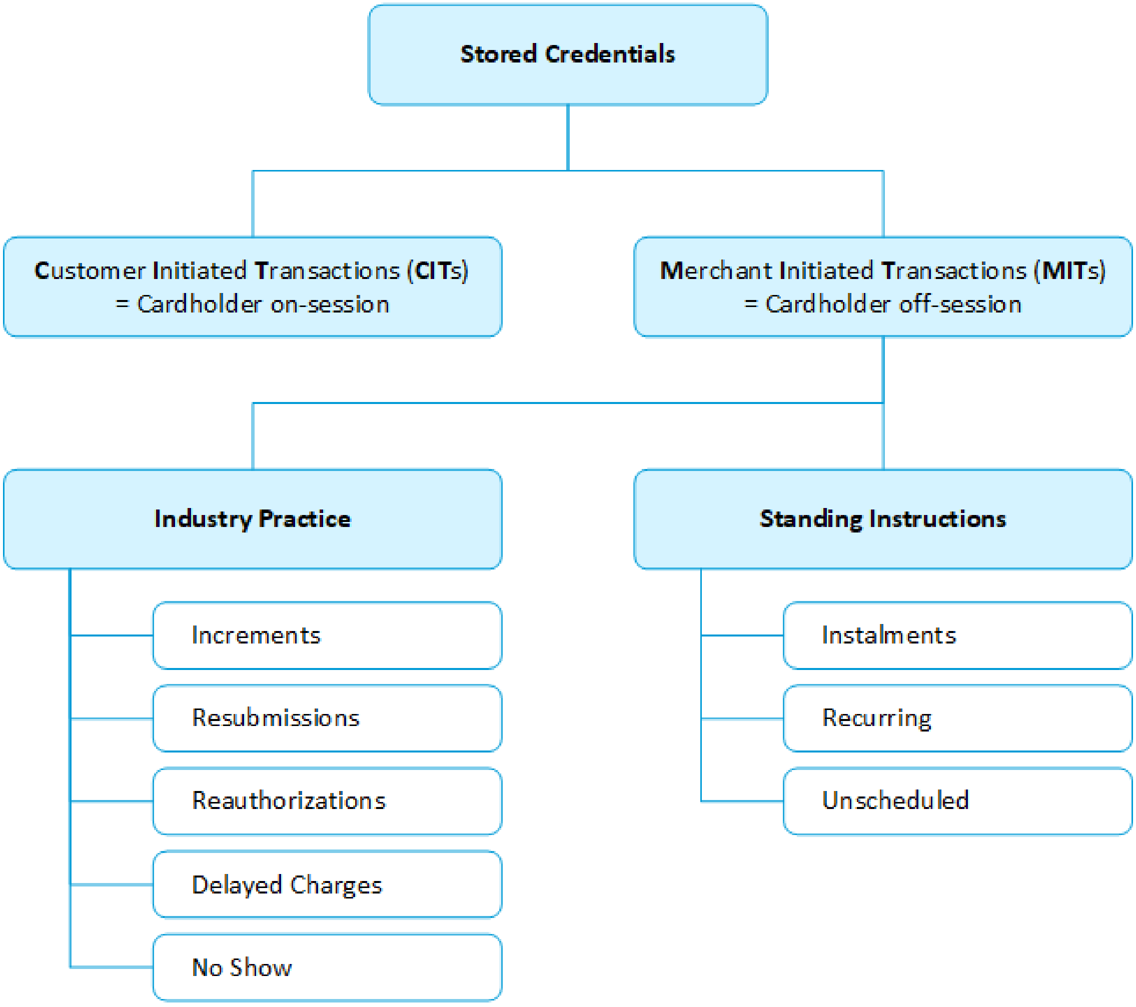
Must contain all of the following:

- An abbreviated airline (or US railway) name in the first 11 or 12 positions
- A blank in position 12 if applicable
- Airline (or US railway) ticket identifier beginning position 13

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
F	L	Y		L	O	W		P	L	C		1	2	3	4	5	6	7	8	9	0

5.2 Stored Credentials EN

Whenever card credentials (i.e. cardholder name, card number/token and/or expiry date) are stored for future use a prior consent by the cardholder is required.
During the establishment of such a mandate the cardholder should be informed about the exact reason for storage of the credentials with the merchant. That means an authorization request that establishes a mandate for stored credentials also must indicate the kind of potential subsequent transactions.
These subsequent transactions with a stored payment credential that the cardholder has consented to are broadly categorized into Customer Initiated Transactions (CITs) and Merchant Initiated Transactions (MITs).



The decisive difference between **CITs** and **MITs** is that the latter are out of scope of the RTS for SCA. This is because the Cardholder regularly is off-session and thus, practically not available for an authentication.

There are various use cases for MITs that can be generally categorized into transactions following a certain Industry Practice and Standing Instructions.
In EVO E-PAY CITs and MITs for Standing Instructions are flagged via the JSON object credentialOnFile.

Please note that all unscheduled MIT transactions are not supported in 3DS 2.0 and thus, will be directly sent for authorization without entering the EVO E-PAY 3DS sequence.
MIT Recurring transactions however will be sent through the 3DS 2.0 protocol to the issuer in order to garantuee best possible acceptance rates.

Please note that with each initial CIT that establishes a mandate for subsequent MITs you will receive a `schemeReferenceID` that must be included in follow-up transactions in order to link the sequence.
Once SCA becomes mandatory on September 14, 2019 existing MITs covered by cardholder agreements can continue to be processed without a `schemeReferenceID` if the mandates were setup before that date (i.e. grandfathering). Please do not submit any dummy values. EVO E-PAY will automatically apply appropriate values in the authorization protocol to indicate so called grandfathering.

Cardholder Initiated Transaction (CIT)

A cardholder-initiated transaction is any transaction where the cardholder actively participates in the transaction. This can be either at a terminal in-store or through a checkout experience online, or with a stored payment credential that the cardholder has previously consented to store with the merchant.

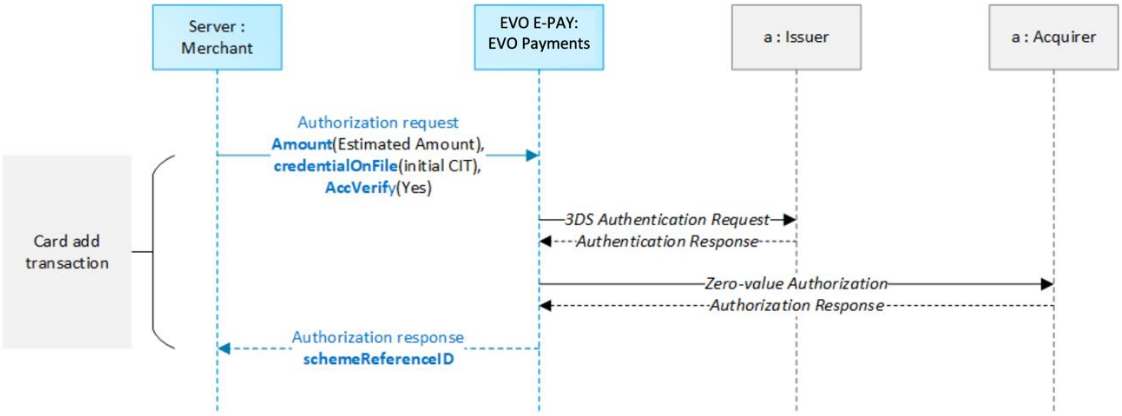
Merchant Initiated Transaction (MIT)

Any transaction that relates to a previous cardholder-initiated transaction but is conducted without the active participation of the cardholder. As a result, the merchant cannot perform any cardholder validation. In all cases, a merchant-initiated transaction must refer to the cardholder's original interaction.

5.2.1 Real-time service via mobile app with payment after service completion EN

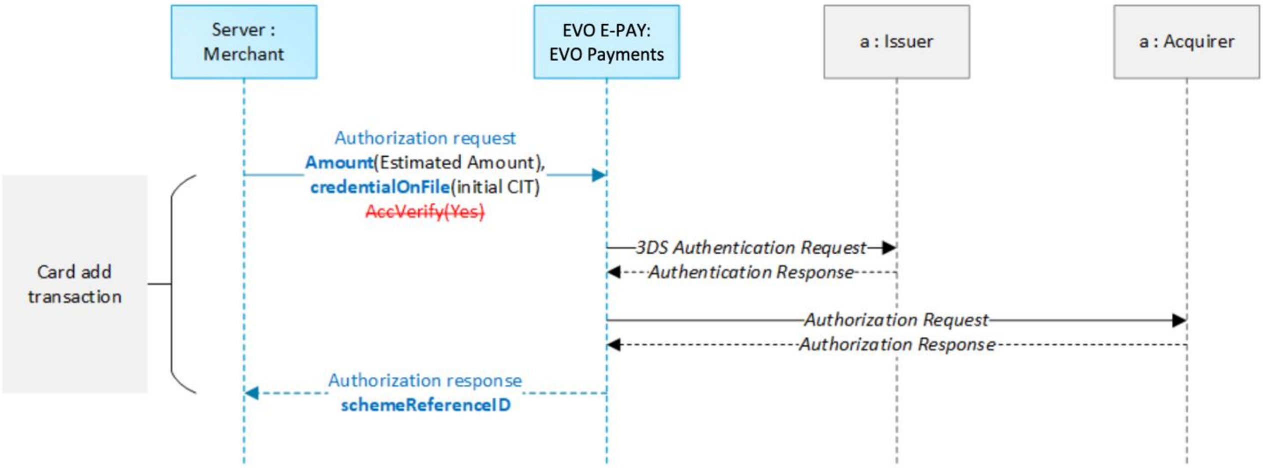
In many scenarios within the sharing economy such as car sharing or bike sharing a customer's mobile device is an essential part in the service delivery and payment system. Card credentials are often getting stored with the cardholder's account at the service provider for optimal user experience.
The sequence of actions to be performed are outlined in the diagrams below.

5.2.1.1 Card Add as part of a none payment transaction (NPA)

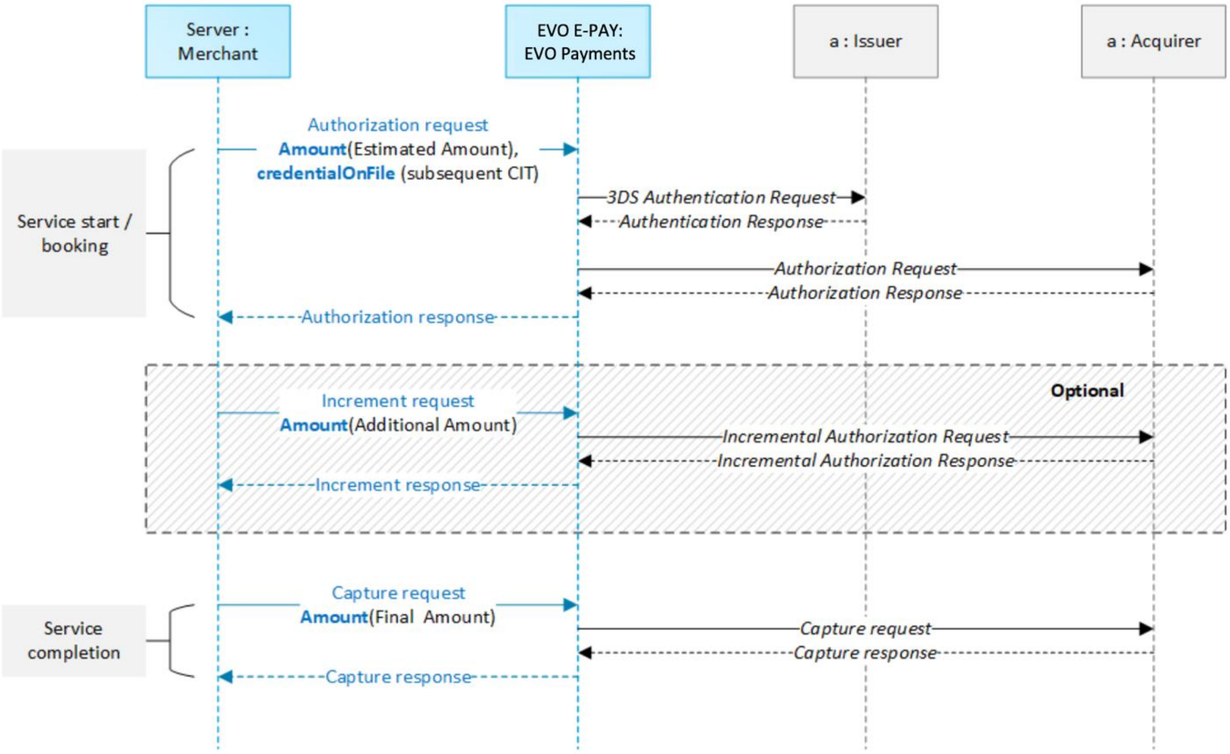


If the **Card Add** is <u>NOT</u> part of a payment transaction it is mandatory to perform an account verification (see `AccVerify`).

5.2.1.2 Card Add as part of a payment transaction



5.2.1.3 Service Provisioning



Unscheduled Credential-on-File (UCOF) **MITs** are not applicable in scenarios where the cardholder is on-session at the time of service completion and thus, is available for authentication. This is regularly the case for car sharing or ride hailing apps for example.

> If the estimated amount is lower than the final amount it is recommended to perform a full reversal on the originally authorized amount and to submit a new authorization for the final amount.

5.2.1.4 Amount

Card Add

A zero amount or any estimated amount. Please note that the amount is usually displayed to the cardholder during an authentication challenge and thus, should be within the customer's reasonable expectation.

Service Provisioning

The amount in the authorization request should be an estimated for the service provision according to reasonable customer expectations. Once the service has been completed incremental authorizations may be used before the final amount is captured.

5.2.1.5 credentialOnFile

Card Add

The UCOF flag is submitted to establish a mandate for storing the credentials and to obtain an initial `schemeReferenceID`. The card issuer is obliged to perform a step-up during authentication.

```
{
  "type": {
    "unscheduled": "CIT"
  },
  "initialPayment": true
}
```

Service Provisioning

The CIT flag is submitted in order to enable UCOF transactions without card security value.

```
{
  "type": {
    "unscheduled": "CIT"
  },
  "initialPayment": false
}
```

5.2.1.6 AccVerify

All Card Adds that are not performed as part of a payment transaction require and account verification.

5.2.2 Delayed Shipment EN

In some instances a merchant may receive an order from a customer that will not be available for shipment within an authorization hold period of 7 (i.e. final authorization) respectively 30 days (i.e. pre-authorization). This is typically the case for:

- highly configured products that are **Build to Order (BTO)** such as bicycles, computer servers, furniture or other manufactured items that exceed standard specifications

- pre-orders of upcoming products such as new phone models
- out of stock items

Please note that if the authorization is to take place a considerable time after the initial order it is best practice to send a reminder to the cardholder a couple of days before authorization to maximize the likelihood for funds to be available.

To maintain a potential liability shift of 3DS authentication it is recommended follow a two step process:

1. Initial Card Add as part of a none payment transaction (NPA)
2. Subsequent unscheduled COF MIT with authentication data from step 1 (UCOF MIT)

5.2.2.1 Initial Card Add as part of a none payment transaction (NPA)

To establish the card on file mandate and to authenticate the cardholder please submit an authorization request to EVO E-PAY.

Amount

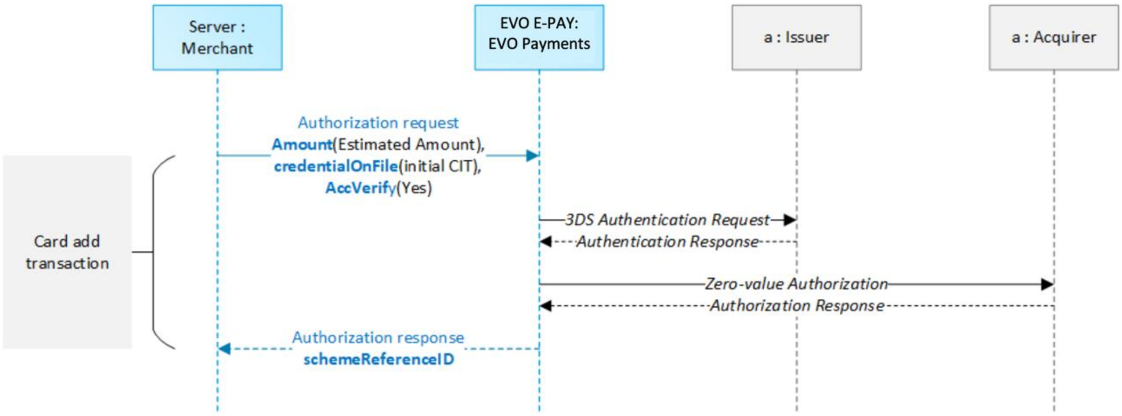
The specified `Amount` will be used within the 3DS authentication process and displayed during the cardholder challenge to the customer. It should be the final amount that is to be charged to the customer once the order is fulfilled as it will be also the maximum amount for the liability shift.

COF

The cardholder challenge will be enforced through the `credentialOnFile` indicator labelling the transaction as a mandate establishment for subsequent MITs.

Account Verification

To suppress an immediate authorization hold of the full order amount on the cardholder's account and to comply with the card scheme rules for card add transactions it is required to perform an account verification (aka zero-value authorization) transaction by submitting the key `AccVerify` with the value 'Yes'.



The response JSON object `threeDSData` in combination with the conditional (i.e. challenge flow) JSON object `resultsResponse` contains the authentication data needed for the subsequent MIT authorization request once the order is ready to be shipped.

5.2.2.2 UCOF MIT

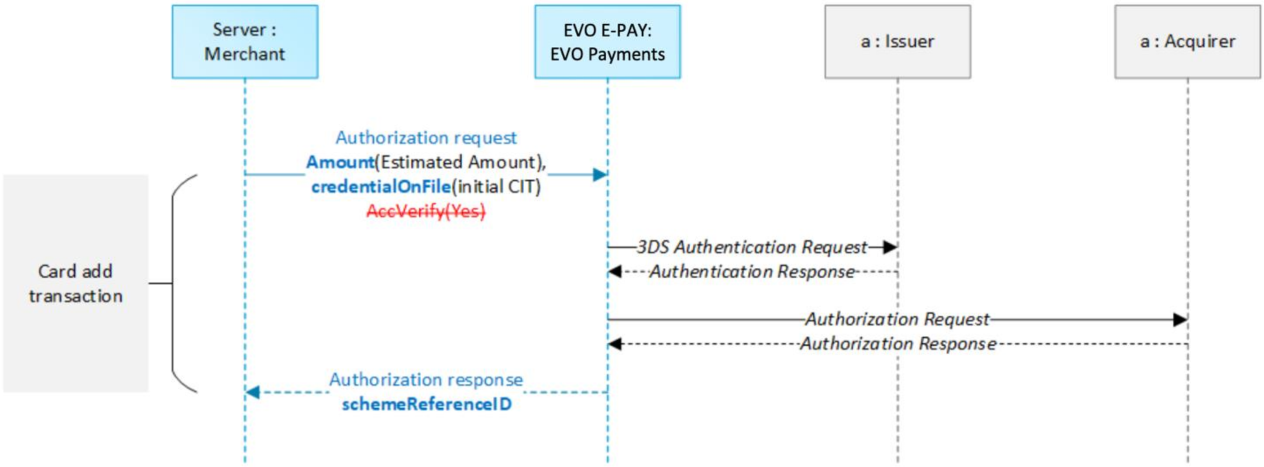
Once the product or service becomes available and is ready to be shipped or at any other date that permits to put an authorization hold on the cardholder's account please submit an authorization request flagged as UCOF MIT in combination with the authentication data obtained in initial COF establishment transaction.

COF

The `credentialOnFile` flagging of MIT prevents the issuer from requesting a cardholder challenge and links the transaction via the `schemeReferenceID` to the initial COF.

threeDSData

Liability protection can be established by submitting the `threeDSData` object containing the authentication data of the initial COF CIT transaction.



5.3 Account Verification EN

An account verification also known as a zero value authorization is a request to check whether a card account is in good standing (i.e. the card is not stolen and the card account can be used for payments).

5.3.1 3DS and Account Verification

Please note that if an account verification (i.e. AccVerify=Yes) in combination with 3DS is requested the authorization will be always performed with a zero-value. The authentication however will be carried out with the amount correspondent to the value submitted in the `Amount` field. If the transferred `Amount` is a zero-value the related liability shift will be also for a zero amount.

5.3.1.1 Credential on File

Please note that non-payment transactions that establish a `credentialOnFile` (aka Card Add) require an account verification.

5.4 Non-payment authentications for Card Add EN

Please use `AccVerify` to request an Account Verification when adding a card to COF without a payment. Non-payment authentications for Card Add transactions always require step-up authentication (i.e. challenge).

Provisioning, such as Card Add is generally considered as an

> *action through a remote channel which may imply a risk of payment fraud or other abuses*

pursuant to article 97(1)(C) PSD2. There are no exemptions provided for these actions.

When a card is added to COF and a payment is requested at the same time, only one SCA is needed.

Use Case	Flags
Add COF without a payment	`AccVerify` `credentialOnFile`
Add COF as part of a payment	`credentialOnFile`

5.5 Mandatory and Conditional Required Data Elements for EMV 3DS EN

Please note that some data elements in EMV 3DS that are marked as conditional required are further specified as:

> *Required unless market or regional mandate restricts sending this information.*

This applies for instance to data elements such as e-mail, phone number or billing address.
Considering the legal landscape and depending on the issuer *(or more precisely on the vendor of the issuer Access Control Server (ACS))* this specification might be interpreted as strictly required within the European Economic Area (EEA).
Contrary to the EMV 3DS protocol specification:

> *A.1 Missing Required Fields*
> *. . .*
> *Unless explicitly noted, if a required field is missing, the receiving component returns an Error Message . . . This applies whether the field is always Required or Conditionally required.*

the card schemes mandated that issuers must not decline EMV 3DS messages when one or more of conditional fields are absent. But the card schemes also stipulate that merchants must send conditional fields in EMV 3DS messages in accordance with the applicable data protection law.
The following elements are considered key and it is strongly advised to provide these data points:

- Cardholder Information
  - Name
  - Email address
  - Home phone number
  - Mobile phone number
  - Billing address
  - Shipping address
- Browser Information (depending on the integration type)
  - IP Address

As a general rule it is strongly recommended to always send conditional required data elements to avoid unnecessary friction and declines.

5.5.1 VISA

5.5.1.1 Travel and Hospitality Merchants

Scenario 1: Authentication performed by a Travel Agent on behalf of one single merchant
When the Travel Agent is facilitating authentication on behalf of a single merchant at the time of booking the process is as follows:

- The Travel Agent discloses to the cardholder appropriate T&Cs and follows other requirements associated with the potential future MIT type the merchant may have to process.
- The Travel Agent authenticates the transaction for the total booking amount (Merchant descriptor name = "Travel Agent Name * name of merchant ")
- The Travel Agent may optionally perform an account verification to check validity of the card before passing the card details to the merchant (If an account verification is performed, it must not include the CAVV as this is required by the end merchant)
- The Travel Agent passes the authentication data to the merchant.
- The Merchant submits a normal authorization request to EVO E-PAY including the required data in the threeDSData [Request] JSON object.

In case the merchant wants to perform the payment at a later time than he should:

1. Perform an Account Verification (i.e. AccVerify=Yes) with the authentication data received from the Agent within 24 hours.
2. Subsequently when the amount is due send an authorization flagged as MIT including the schemeReferenceID from the previous account verification transaction.

Scenario 2: Authentication performed by a Travel Agent on behalf of several other merchants.
This scenario covers the case when a customer makes a travel reservation online via a Travel Agent which includes the delivery of services by multiple merchants.

- The Travel Agent discloses to the cardholder appropriate T&Cs and follows other requirements associated with the potential future MIT type the merchant may have to process
- The Travel Agent then authenticates the transaction for the total booking amount (Merchant descriptor name = "Travel Agent Name")
- An additional 3DS 3RI authentication request is required to provide CAVVs for each further merchant who will perform an authorization.
- The Travel Agent submits authentication data to the merchant.
- The Merchant submits a regular authorization *request* to EVO E-PAY including the required data in the threeDSData [Request] JSON object.

In case the merchant wants to perform the payment at a later time than he should:

1. Perform a zero-value ***account verification*** with the authentication values received from the Agent within 24 hours.
2. Subsequently when the amount is due send an authorization flagged as MIT including the schemeReferenceID from the previous account verification transaction.

5.5.1.2 Other Merchants

Scenario 3: Authenticated CIT from the Agent and subsequent MIT transaction from the merchant
This is the case when the Agent performs the authentication and subsequently uses the VISA CAVV for its own authorization payload. In such a case as the Agent already used the CAVV and he may not be able to use the 3DS/3RI feature (for having a new CVV for the merchant) than he can provide to the merchant only the schemeReferenceID of the initial CIT together with the payload so that the merchant may initiate his authorizations flagged as an subsequent MIT (i.e UCOF, Delayed Auth)

As a difference worthy to be highlighted between VISA and MC is such use cases is that:
a) Authentication Value (CAVV/AAV)
In case of VISA, the Agent who is performing authentication on behalf of several merchants needs through the ***3DS/3RI*** request (only possible with EMV 3DS 2.2 version) to provide separate CAVV's values for each merchant separately.
Mastercard on the other hand is stating that for the Agent Model where a single authentication is linked to multiple authorizations the same authentication code/ AVV could be used for multiple transactions.

b) schemeReferenceID

Under the third scenario the Agent can provide to the merchant together with the payload only the '*schemeReferenceID*' from the initial CIT with no Authentication data in such a way that the merchant is able to initiate subsequent MIT transactions referring to the initial establishment.
For VISA the "*schemeRefernceID*" (transactionID) is usually provided as a single response parameter while for Mastercard the "*schemeReferenceID*" is a concatenation of fields like: *(SettlementDate+FinancialNetworkCode& BanknetReference).

-*SettlementDate* -> n..4
-*FinancialNetworkCode* -> an..3
-*BanknetReference* -> an..9

* Agents using this type of scenario (with CIT/MIT) are recommended to provide their merchants with the required Mastercard reference data where the merchant can subsequently submit it to EVO E-PAY "*schemeReferenceID*" during MIT transactions.
Please also be advised that the above approach for handling each of these scenarios serves only as recommendation, therefore, merchants and Acquirers can choose alternative options that complement their business model, as long as they remain compliant with the key principles of the PSD2 SCA.

5.6 schemeReferenceID EN

This is a unique transaction identifier provided directly from the card schemes like VISA and MC in order to uniquely reference a transaction in the whole payment ecosystem. It was introduced initially by VISA in accordance to their Framework specifications like COF (Credential On File) and MIT (Merchant Initiated Transactions), relevant to use cases involving transaction types such as Recurring, UCOF (MIT), Incremental, Delayed Authorization, Resubmission etc.
With the release of EMV 3DS specifications it came as a requirement also for Mastercard to make use of such a unique identifier which they called "traceID" or "grandfathering ID". The logic behind it, is that the Issuer could rely on this identifier to link the initial payment with all the subsequent ones related to a standing order in a COF or MIT regime. This will allow the Issuer to apply different transaction rules (i.e. no CVV/CVC, no additional authentication in EMV 3DS) for all the subsequent payments.

In the current situation for Mastercard/Maestro transactions on which the initial payment (Establishment of an Agreement) have been submitted before 14th of September 2019, merchants have not been provided with the "*schemeReferenceID*" in authorization responses. EVO Payments will require those merchants dealing with the above use cases to leave this parameter empty on all the subsequent (COF/MIT) transactions.
For **Initial payments** (Establishment of an Agreement) after 14th of September 2019 the merchants **must save** the "*schemeReferenceID*" value provided in the response and submit it to EVO E-PAY in all the subsequent payments related to that Initial agreement. As for VISA the "*schemeReferenceID*" will be the equivalent of the previous EVO E-PAY parameter "*TID*" that the merchants are currently submitting in accordance to COF & MIT Frameworks.

6. Test Cards EN

6.1 Card Numbers

	Visa	Mastercard	Test Scenario
1	4000012892688323	5232125125401459	Browser challenge
2	4000016435940133	5232122189301469	Browser challenge
3	4000012699048523	5232127264637786	Browser frictionless; missing DS Transaction ID
4	4000011744135012	5232122741507017	Not authenticated browser frictionless
5	4000019966199434	5232122422543299	Authenticated browser frictionless
6	4000015573198637	5232128083944791	Browser challenge missing ACS URL
7	4000017873485953	5232122596907270	Authentication protocol error
8	4000014730366880	5232124106987982	Browser challenge; authenticated transaction; missing authentication value

6.2 One-time Passwords (OTPs)

Notice: Please confirm the One-Time-Password in case of a challenge with mouse click instead of Enter key, because otherwise the "Cancel"-button is selected and the authentication is terminated.

	otpValue	transStatus	transStatusReason	ECI	authenticationValue
1	1234	Y		01	JAmi21makAifmwqo2120cjq1AAA=
2	1111	N	01	01	
3	2222	R	01	01	
4	3333	U	01	01	
5	6666	Y	01	01	
6	7777	A		01	JAmi21makAifmwqo2120cjq1AAA=
7	8888	N	10		
8	9999	N	08		
9	0001	N	01		
10	0002	N	02		
11	0003	N	03		
12	0004	N	04		
13	0005	N	05		
14	0006	N	06		
15	0007	N	07		
16	0009	N	09		
17	0010	N	10		
18	0011	N	11		

6.2.1 transStatus

	transStatus	Description
1	Y	Authentication Verification Successful.
2	N	Not Authenticated /Account Not Verified; Transaction denied.

	transStatus	Description
3	U	Authentication/ Account Verification Could Not Be Performed; Technical or other problem, as indicated in ARes or RReq.
4	A	Attempts Processing Performed; Not Authenticated/Verified, but a proof of attempted authentication/verification is provided.
5	C	Challenge Required; Additional authentication is required using the CReq/CRes.
6	D	Challenge Required; Decoupled Authentication confirmed.
7	R	Authentication/ Account Verification Rejected; Issuer is rejecting authentication/verification and request that authorization not be attempted.
8	I	Informational Only; 3DS Requestor challenge preference acknowledged.

6.2.2 transStatusReason

	transStatusReason	Description
1	01	Card authentication failed
2	02	Unknown Device
3	03	Unsupported Device
4	04	Exceeds authentication frequency limit
5	05	Expired card
6	06	Invalid card number
7	07	Invalid transaction
8	08	No Card record
9	09	Security failure
10	10	Stolen card
11	11	Suspected fraud
12	12	Transaction not permitted to cardholder
13	13	Cardholder not enrolled in service
14	14	Transaction timed out at the ACS
15	15	Low confidence
16	16	Medium confidence
17	17	High confidence
18	18	Very High confidence
19	19	Exceeds ACS maximum challenges
20	20	Non-Payment transaction not supported
21	21	3RI transaction not supported
22	22	ACS technical issue
23	23	Decoupled Authentication required by ACS but not requested by 3DS Requestor
24	24	3DS Requestor Decoupled Max Expiry Time exceeded
25	25	Decoupled Authentication was provided insufficient time to authenticate cardholder. ACS will not make attempt
26	26	Authentication attempted but not performed by the cardholder

7. Terms and Definitions EN

7.1 Mandatory and conditional data elements

Conditions in 3DS 2.0 are often described as *'Required unless market or regional mandate restricts sending this information.'*

This applies for example to the e-mail address of the cardholder. Please note that some vendors of ACS software and some issuer might consider these data elements technically as mandatory within the EEA since there are currently no known restrictions. Thus, it is highly recommended to submit these data elements if possible.

7.2 Condition Codes

Code	Condition
M	**Mandatory**. Signifies that the data element shall be included in that message.
O	**Optional**. The data element may or may not be present in a message.
C	**Conditional**. The data element shall be included (i.e. mandatory) when specified conditions are met.

7.3 Definitions

Term	Definition
Authorization	An authorization is an approval or guarantee of funds given by the card issuer to the acquirer.
Authorization Advice	The acquirer advises the card issuer of authorization already given (e.g. Authorization by Voice).
Capture	Capture is the process of combining the approval amount and authorization code of a transaction and turn it into a billable transaction record. It is essentially the instruction to deduct the funds from the debtor's account. The acquirer usually submits a capture file to the card network (dual messaging system). In Host Capture Systems the merchant usually sends a Capture Advice message to the acquiring host. For Terminal Capture Systems the card acceptor (e.g. merchant) either submits a capture file (most common) to the acquirer or performs a batch upload. The capture records submitted by the card acceptor are usually validated at the acquiring host and then added to the acquirer's capture file for the corresponding card network.
Sale	A Sale is an instruction from the merchant to the acquirer to request an authorization and a capture of the transaction completed at the Point of Sale within a single message. That means a successful authorization will be added to the acquirer's capture file automatically without the need for further instructions or completion messages. However, some terminal capture systems may require Sale transactions to be included in the capture file or in the batch upload.
Terminal Capture	Terminal Capture means that the terminal submits Authorizations, Sales, and Reversals to the host throughout the day. The terminal stores all of these transactions as well as any transactions performed locally (offline), so that the terminal can perform a batch submission at the end of the merchant's processing day. Terminal capture processing offers the merchant the ability to perform offline transactions that are included only in the batch. Offline transactions include for instance returns, prior sales and tip adjustments.
Host Capture	In Host Capture processing, transaction batches are managed by the acquiring host. Merchants transmit transactions to the host as they occur at the point of sale and the host records the transactions in a batch. In ISO 8583 based message protocols this is often referred to as capture advice. The batch is then closed either by request from the merchant's system (manual batch release) or at a schedule time each day (host auto-close). The auto-close option is the most common mode.

Term	Definition
Recurring	Recurring Transactions are a series of transactions processed following agreement between a Cardholder and a Merchant where the Cardholder purchases goods or services over a period of time through a number of separate transactions.
Installment	Installment Transactions represent a single purchase of goods or services billed to a Cardholder’s account in multiple segments, over a period of time that has been agreed between the Cardholder and a Merchant.

8. Syntax

EVO E-PAY response codes are 8 digits and are constructed according to the syntax described below.
Format: N8, (NNNNNNNN)

- `N (status)`
- `NNN (category)`
- `NNNN (detail)`

8.1 Sample

`22060203`

- `2 Error`
- `206 3DS credit card adapater for authorization protocol GICC`
- `0203 Card brand does not support 3DS`

8.2 Status Codes (1)

Code	Meaning	Description
0	Ok	Operation completed successfully.
2	Error	Operation failed.
4	Fatal Error	Operation failed and data that were processed might be lost.
6	Continue / Transient	Operation is not completed. Final status will be transferred asynchronous.
7	EMV 3DS Info	Intermediate states in the EMV 3DS sequence

8.3 Category (2-4)

The category code is a 3 digit value designating the payment adapter or payment protocol. For 3DS 2.0 those category codes will be in the range of 100 to 299 depending on the card connector.

8.4 Detail (5-8)

Status	Category	Detail	Description
2	xxx	0101	Message received invalid
2	xxx	0102	Message version number not supported
2	xxx	0103	Sent messages limit exceeded. Only used for PReq
2	xxx	0201	Required element missing
2	xxx	0202	Critical message extension not recognized

Status	Category	Detail	Description
2	xxx	0203	Format on one or more elements is invalid according to the specs
2	xxx	0204	Duplicate data element
2	xxx	0301	Transaction id is not recognized
2	xxx	0302	Data decryption failure
2	xxx	0303	Access denied, invalid endpoint
2	xxx	0304	ISO code is not valid
2	xxx	0305	Transaction data is not valid
2	xxx	0306	Merchant category code is not valid for payment system
2	xxx	0307	Serial number is not valid
2	xxx	0402	Transaction timed out
2	xxx	0403	Transient system failure
2	xxx	0404	Permanent system failure
2	xxx	0405	System connection failure
2	xxx	0911	UnionPay specific error code. Present when Data fields relevance check failed (ECI value and AV appearance are inconsistent with transaction status).
2	xxx	0912	UnionPay specific error code. Present when duplicated transaction ID (Transaction ID should be unique for each AReq request).
2	xxx	0985	3DS 2.0 is not supported by this card. The merchant has to follow the fallback process.
2	xxx	3002	Invalid parameter BROWSERINFO
2	xxx	3006	Invalid parameter BILLINGADDRESS
7	000	0000	3DS 2.0 versioning successful
7	000	0001	Authentication response --> challenge mandated

9. ECI Codes EN

The ECI value is provided by the issuer ACS. It indicates the level of authentication that was performed on the transaction. The ECI value received from authentication is forwarded in the authorization request and also determines whether a transaction receives liability protection.

9.1 Visa

ECI	Description	3DS Version(s)	Merchant Liability Shift
05	Cardholder authentication successful (this includes successful authentication using risk-based authentication and/or a dynamic password)	3DS 1.0 EMV 3DS (2.0)	Yes
06	Merchant attempted to authenticate the cardholder • For 3DS 1.0.2, the ECI 06 value may be utilized as an authentication response from the Issuer ACS, at the issuer's discretion. For example, issuers that use risk-based authentication may provide an ECI = 06 for a transaction that does not require step-up, also known as frictionless authentication. These issuers may reserve an ECI = 05 for transactions that were successfully stepped-up.	3DS 1.0 EMV 3DS (2.0)	Yes

ECI	Description	3DS Version(s)	Merchant Liability Shift
	• For 3DS 2.0, the ECI 06 value can only be used to indicate that a "Merchant attempted to authenticate the cardholder".		
07	Non-authenticated e-commerce transaction	3DS 1.0 EMV 3DS (2.0)	No

9.2 Mastercard

ECI	Description	3DS Version(s)	Merchant Liability Shift
00	Non-authenticated e-commerce transaction	3DS 1.0 EMV 3DS (2.0)	No
01	Merchant attempted to authenticate the cardholder and received authentication value (Accountholder Authentication Value (AVV))	3DS 1.0 EMV 3DS (2.0)	Yes
02	Cardholder authentication successful (this includes successful authentication using risk-based authentication and/or a dynamic password)	3DS 1.0 EMV 3DS (2.0)	Yes
04	Data share only: non-authenticated e-commerce transaction but merchants have chosen to share data via the 3DS flow with the issuer to improve authorization approval rates	EMV 3DS (2.0)	No
06	Acquirer exemption	EMV 3DS (2.0)	No
07	Recurring payments might be applicable for initial or subsequent transaction) • If this value is received on initial recurring payments merchant will have liability shift • Subsequent transactions are considered as MIT and liability remains with the merchant	EMV 3DS (2.0)	Yes

10. 3DS 2.0 Merchant Use-Cases & Testing of 3-D Secure 2.0 EN

What can you expect in this area?
Due to various scenarios that can arise with 3-D Secure 2.X, we will subdivide the following into three thematic areas: 1. General technical adjustments (relevant for all merchants) 2. Use cases for transaction flagging (different handling depending on merchant scenario) 3. Test 3-D Secure 2.X via EVO Payments

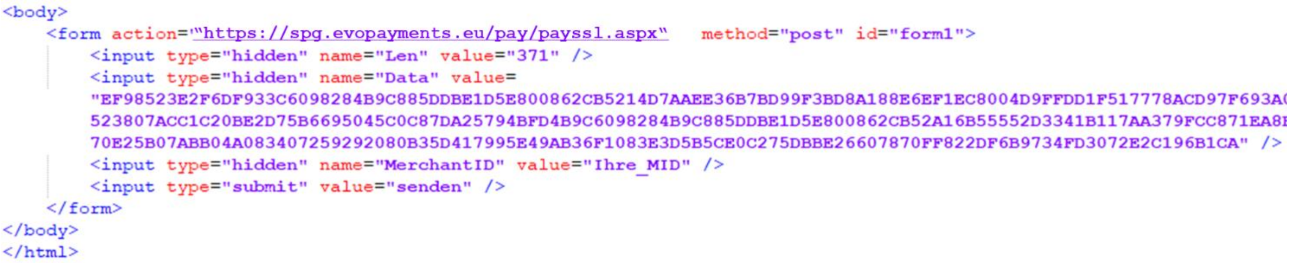
1. General technical adjustments

Which request types does 3-D Secure 2.0/SCA affect?

- Affected request types are: Authorization and Sale
- Capture (booking) and credit notes are excluded from changes

How will data transfer to/from EVO Payments look like in the future?

- REQUEST: During the implementation of 3-D Secure 2.0 and the necessary delivery of larger amounts of data, we recommend you to call our forms via Form-POST Method. Please note that the option iFrame is still available. Background are possible browser restrictions, which can lead to the fact that the sent data string is cut off.
- Example:



```
<body>
    <form action="“https://spg.evopayments.eu/pay/payssl.aspx“   method="post" id="form1">
        <input type="hidden" name="Len" value="371" />
        <input type="hidden" name="Data" value=
        "EF98523E2F6DF933C6098284B9C885DDBE1D5E800862CB5214D7AAEE36B7BD99F3BD8A188E6EF1EC8004D9FFDD1F517778ACD97F693A
        523807ACC1C20BE2D75B6695045C0C87DA25794BFD4B9C6098284B9C885DDBE1D5E800862CB52A16B55552D3341B117AA379FCC871EA8
        70E25B07ABB04A083407259292080B35D417995E49AB36F1083E3D5B5CE0C275DBBE26607870FF822DF6B9734FD3072E2C196B1CA" />
        <input type="hidden" name="MerchantID" value="Ihre_MID" />
        <input type="submit" value="senden" />
    </form>
</body>
</html>
```

- RESPONSE:
  Please also note a change to the final redirect to the URLSuccess | URLFailure. This will be executed as a body POST in the case of a 3-D Secure 2.0 transactions. Therefore, you should be able to receive both a GET and a POST response on the URLSuccess | URLFailure.

How can I choose between 3DS 1.0 or 2.0?

- IMPORTANT: To be able to use and test 3-D Secure 1.0 or 3-D Secure 2.0, we have to configure 3-D Secure on our EVO E-PAY on your behalf. Please contact the responsible EVO Payments Implementation Manager if you have not yet started this process.
- By default, each payment is made following the 3-D Secure 1.0 process.
- If you want to follow the 3-D Secure procedure 2.0, please use the request parameter MsgVer=2.0. This applies to tests as well as in production at a later stage.
  - Parameter: MsgVer
    Value: 2.0

Use of JSON objects becomes mandatory

- Please note that a mandatory extension of existing parameters comes with the implementation of 3-D Secure 2.0 . For this reason, EVO E-PAY expects and returns relevant additional data as JSON Object. The JSON Object must be Base64 encoded and regularly transmitted with all other parameters in the encrypted Blowfish data to EVO E-PAY.
- Please pass JSON Objects with values only. Empty or zero-filled objects/parameters lead to a rejection.

JSON Example request

https://spg.evopayments.eu/pay/payssl.aspx?MerchantID=Generic3DSTest&len=1800&data=CDC44E5A9D2C8A559CEDF1CCA97C9FBD3D90E046BFBF96F06ADA9A00FB0BC3494317E8D924FF44729671B93348B477F880541ACFF12C8E3A868CD55FEA95219C245CF7F4716FCF3462167A8B63D11424FA7BD30891504F8465C56805975115EB71C0A04E5D7466D771495035749FFF94D3087529F578DEF518003EA1422F6DE7B7DFD78A0DD695550623A42BF41A422EC219012318FE26D2B757F12BDFE046EA4CB8D35079ABAB6859691FEE1B03483471495035749FFF94D3087529F578DEF518003EA1422F6DE7D4E20259A484D23A9EFC7F4ADB209DD67D8EDE5BD2AC0CB2682D7CF26A6624A54BCF4E93219ADD89ABA6214820D4BAA5A9A184DD7F8AF3E2BE98C5B63113276B023B92DA5AADCCD7387B71B6651A0E7E4E42F8790122386AA9A184DD7F8AF3E23CFEC0086B59B6A9D98EB96DFDA496D97D85706A4A810056FE48AB878EFC1E976DB7504D402F4B96778B45ADE1DF3E217EFFBA566359677AB73F514F1E75F11DBE3E15983BA530E7D5B13A87D1A2ED19A9A184DD7F8AF3E21D32D652A71B2A49A58F3A30256097DA11388C26E7CBEB12E65B31C485C94DE8179CEACDE9237EF4C426A05E594E28069E10B19AE173D25A93A546845C5D78C44112031D6D5DE9B4ABA6214820D4BAA5A9A184DD7F8AF3E260C35EC59CD2FAA2435CD631BFC801AA7C72A1BAE39879C0BF733EDC45DD99F3A9A184DD7F8AF3E2DDA25A6458507ACE3B3CAAC3A4B293A9C6177F7F00EBFB6924050D9DF661DE8EC204863D819ABF9564498E9F2D72BEFF2E04

0214C4961D8737821BA1F638BE05FB01E1B382733FC42D6B04AB80D66218C75E691B9475C5F6CF13AD
357057BC6B5864EE113DF2272EF6572101D5E45CB634F3E941FA7B3EA7E636EAEF751C67C82F8E8D9B6
18E69826221B2A42D7F694D9E10B19AE173D25A6EB48BD63BFFE0FAFC78722BD9FFA39623B5D40494B
96D2A9E10B19AE173D25A188DA61C8E3401850C400A3144C3547808A0C82C7B8E9863D017852B02FBF
E6D62983EBC372B1A8108D832C13F92E88535C213D0FDA1B1A5C426A05E594E28069E10B19AE173D25
A92AD74641E23F21D1D66F1B352AFCCD408B1727FACC2405AA9A184DD7F8AF3E29B3106F31EE7D473A
854D99576FDD5620141A96DEF638FCE4362F90866AED8044E42F8790122386AA9A184DD7F8AF3E20F6B
F2E070199426696A900FEEBC7848B6F72D445F2CB9F0ED160CC32B1A3C40C426A05E594E28069E10B19
AE173D25A201E55FC81E8F7CD78FFD98E342897C11AB2BE505B3E8421C63E936DCCF29058C31D72A36
97DA2C89EFC7F4ADB209DD67D8EDE5BD2AC0CB2682D7CF26A6624A54BCF4E93219ADD89ABA6214820
D4BAA5A9A184DD7F8AF3E2BE98C5B63113276B023B92DA5AADCCD7387B71B6651A0E7E4E42F8790122
386AA9A184DD7F8AF3E23CFEC0086B59B6A9D98EB96DFDA496D93F669AB8A34E11706F7B3F762241F74
9A9A184DD7F8AF3E286587E20CD9A354709F67B1501183CFC5D6FD3FD6E23B0D4FA9746B8925D4A4FA
9A184DD7F8AF3E21D32D652A71B2A49A58F3A30256097DA11388C26E7CBEB120758D07B77A47DB34E3
59C7AE383D69BC426A05E594E28069E10B19AE173D25A93A546845C5D78C44112031D6D5DE9B4ABA62
14820D4BAA5A9A184DD7F8AF3E260C35EC59CD2FAA2435CD631BFC801AA7C72A1BAE39879C0BF733ED
C45DD99F3A9A184DD7F8AF3E2DDA25A6458507ACE3B3CAAC3A4B293A9C6177F7F00EBFB6924050D9DF
661DE8EC204863D819ABF9564498E9F2D72BEFF2E040214C4961D8737821BA1F638BE05FB01E1B38273
3FC46AA58C2847221D78069144B06DE3755A6C88EADD3B3FCCD6F6572101D5E45CB634F3E941FA7B3E
A7B08783F57D9AD1BAB2071FAB9B93B3C13FF102AD44B6A493B5C341FB37BF525B0A0E4F490BE1D46A
4C5B8F691A2020868119A0AEB9E9BCD4F9D783FEA316723E17976FBB4909040AE279D66AF13B8441582
CB00BB30835AB6401E5CDDF295F533AEE31D2677314D288F2C15BFB16837EF4A779C1E39E4AA1CEE13F
ABDB2B89D9A7A89ED81EC005BCD416330CFCE5CF716A316FDF29A9CFF3F25490656C800BCA582CB00B
B30835AB-
ABD19D247E68289A52F1387D978126C967F9BBB890618AF5A0E5136C7DC2892D2460687217D2779B58
36D3F1FFAE8F3B582CB00BB30835ABEE02C59E0AAF8C913339B61F9DDFB7DAC4FF2460869E4876C5DF
D5D39E79330D427654226D9E37E72D7A4C332F59563DF70B3A840877E2B1BF739A2347A73347F7DA9F
100EEBC189ADE92F98BE65BCBE29FE1A3DFE89E44EEEBF9C902BBAA7C2F68CBC48C724B889A53EA1489
88B56CC52D52743C045F57844F6607DDEA75FE613EAC80E2C02BCEA89B71E52E64D7538DC9B82EB274
0B82C698F43B6A62D770935233D5F10E593D0519511BAAD615B0035D7524B097C29BA39EBEBEDB9342
5EB7824B9CCDB1397E716993ED326500615B4B1853A59F760A0E06373BDFE1CC6695A93B15851F56428
&template=evo_responsive&language=en

```
MerchantID=Generic3DSTest

&MsgVer=2.0

&TransID=1234567890

&RefNr=20200124

&Amount=100

&Currency=EUR

&URLNotify=

https://www.shop.com

&URLSuccess=

https://www.shop.com

&URLFailure=

https://www.shop.com

&billToCustomer=ew0KICAgICJjb25zdW1lciI6IHsNCiAgICAgICAgIn-
NhbHV0YXRpb24iOiAiTXIiLA0KICAgICAgICAiZmlyc3ROYW1lIjogIk5hcG9sZW9uIiwNCiAgICAgI-
CAgImxhc3ROYW1lIjogIkJvbmFwYXJ0ZSIsDQogICAgICAgICJiaXJ0aERhdGUiOiAiMTc2OS0wOC0xNSIN-
CiAgICB9LA0KICAgICJtb2JpbGVQaG9uZSI6IHsNCiAgICAgICAgImNvdW50cnlDb2RlIjogIjMzIiwN-
CiAgICAgICAgInN1YnNjcmliZXJOdW1iZXIiIDogIjEyMzQ1Njc4OTEwIg0KICAgIH0sDQogI-
CAgImVtYWlsIjogIm5hcG9sZW9uLmJvbmFwYXJ0ZUBmcmFuY2UuY29tIg0KfQ==

&billingAddress=ew0KICAgICJjaXR5IjogIkFqYWNjaW8iLA0KICAgICJjb3VudHJ5Ijogew0KICAgI-
CAgICAiY291bnRyeUEzIjogIkZSQSINCiAgICB9LA0KICAgICJhZGRyZXNzTGluZTEiOiB7DQogICAgI-
CAgICJzdHJlZXQiOiAiRXhpbGVzdHJlZXQiLA0KICAgICAgICAic3RyZWV0TnVtYmVyIjogIjI3MCINCiA-
gICB9LA0KICAgICJwb3N0YWxDb2RlIjogIjIwMTY3Ig0KfQ==

&shipToCustomer=ew0KICAgICJjb25zdW1lciI6IHsNCiAgICAgICAgIn-
NhbHV0YXRpb24iOiAiTXIiLA0KICAgICAgICAiZmlyc3ROYW1lIjogIkx1ZHdpZyIsDQogICAgI-
CAgICJsYXN0TmFtZSI6ICJYVklJSSIsDQogICAgICAgICJiaXJ0aERhdGUiOiAiMTc1NS0xMS0xNyINCiA-
gICB9LA0KICAgICJtb2JpbGVQaG9uZSI6IHsNCiAgICAgICAgImNvdW50cnlDb2RlIjogIjMzIiwNCiAgI-
CAgICAgInN1YnNjcmliZXJOdW1iZXIiIDogIjEyMzQ1Njc4OTEwIg0KICAgIH0sDQogICAgImVtYWlsI-
jogIkx1ZHdpZ0Byb3lhbC5mcmFuY2UuY29tIg0KfQ==

&shippingAddress=ew0KICAgICJjaXR5IjogIlBhcmlzIiwNCiAgICAiY291bnRyeSI6IHsNCiAgICAgI-
CAgImNvdW50cnlBMyI6ICJGUkEiDQogICAgfSwNCiAgICAiYWRkcmVzc0xpbmUxIjogew0KICAgICAgI-
CAic3RyZWV0IjogIlBsYWNlIGRlIGxhIENvbmNvcmRlIiwNCiAgICAgI-
CAgInN0cmVldE51bWJlciI6ICIxIg0KICAgIH0sDQogICAgInBvc3RhbENvZGUiOiAiNzUwMDEiDQp9

&credentialOnFile=ew0KICAgICJ0eXBlIjogew0KICAgICAgICAidW5zY2hlZHVsZWQi-
OiAiQ0lUIg0KICAgIH0sDQogICAgImluaXRpYWxQYXltZW50IjogdHJ1ZQ0KfQ==

&OrderDesc=Test:0000

billToCustomer=ew0KICAgICJjb25zdW1lciI6IHsNCiAgICAgICAgIn-
NhbHV0YXRpb24iOiAiTXIiLA0KICAgICAgICAiZmlyc3ROYW1lIjogIk5hcG9sZW9uIiwNCiAgICAgI-
CAgImxhc3ROYW1lIjogIkJvbmFwYXJ0ZSIsDQogICAgICAgICJiaXJ0aERhdGUiOiAiMTc2OS0wOC0xNSIN-
```

```
CiAgICB9LA0KICAgICJtb2JpbGVQaG9uZSI6IHsNCiAgICAgICAgImNvdW50cnlDb2RlIjogIjMzIiwNCiAgICAgICAgInN1YnNjcmliZXJOdW1iZXIiIDogIjEyMzQ1Njc4OTEwIg0KICAgIH0sDQogICAgImVtYWlsIjogIm5hcG9sZW9uLmJvbmFwYXJ0ZUBmcmFuY2UuY29tIg0KfQ==

{

"consumer":
{ "salutation": "Mr", "firstName": "Napoleon", "lastName": "Bonaparte", "birthDate":
"1769-08-15" }

,

"mobilePhone":
{ "countryCode": "33", "subscriberNumber" : "12345678910" }

,

"email": "napoleon.bonaparte@france.com"

}

billingAddress=ew0KICAgICJjaXR5IjogIkFqYWNjaW8iLA0KICAgICJjb3VudHJ5Ijogew0KICAgICAgICAiY291bnRyeUEzIjogIkZSQSINCiAgICB9LA0KICAgICJhZGRyZXNzTGluZTEiOiB7DQogICAgICAgICJzdHJlZXQiOiAiRXhpbGVzdHJlZXQiLA0KICAgICAgICAic3RyZWV0TnVtYmVyIjogIjI3MCINCiAgICB9LA0KICAgICJwb3N0YWxDb2RlIjogIjIwMTY3Ig0KfQ==

{

"city": "Ajaccio",

"country":
{ "countryA3": "FRA" }

,

"addressLine1":
{ "street": "Exilestreet", "streetNumber": "270" }

,

"postalCode": "20167"

}

shipToCustomer=ew0KICAgICJjb25zdW1lciI6IHsNCiAgICAgICAgInNhbHV0YXRpb24iOiAiTXIiLA0KICAgICAgICAiZmlyc3ROYW1lIjogIkx1ZHdpZyIsDQogICAgICAgICJsYXN0TmFtZSI6ICJYVklJSSIsDQogICAgICAgICJiaXJ0aERhdGUiOiAiMTc1NS0xMS0xNyINCiAgICB9LA0KICAgICJtb2JpbGVQaG9uZSI6IHsNCiAgICAgICAgImNvdW50cnlDb2RlIjogIjMzIiwNCiAgICAgICAgInN1YnNjcmliZXJOdW1iZXIiIDogIjEyMzQ1Njc4OTEwIg0KICAgIH0sDQogICAgImVtYWlsIjogIkx1ZHdpZ0Byb3lhbC5mcmFuY2UuY29tIg0KfQ==

{

"consumer":
{ "salutation": "Mr", "firstName": "Ludwig", "lastName": "XVIII", "birthDate": "1755-
11-17" }
,

"mobilePhone":
{ "countryCode": "33", "subscriberNumber" : "12345678910" }

,

"email": "Ludwig@royal.france.com"

}

shippingAddress=ew0KICAgICJjaXR5IjogIlBhcmlzIiwNCiAgICAiY291bnRyeSI6IHsNCiAgICAgI-
CAgImNvdW50cnlBMyI6ICJGUkEiDQogICAgfSwNCiAgICAiYWRkcmVzc0xpbmUxIjogew0KICAgICAgI-
CAic3RyZWV0IjogIlBsYWNlIGRlIGxhIENvbmNvcmRlIiwNCiAgICAgI-
CAgInN0cmVldE51bWJlciI6ICIxIg0KICAgIH0sDQogICAgInBvc3RhbENvZGUiOiAiNzUwMDEiDQp9

{

"city": "Paris",

"country":
{ "countryA3": "FRA" }

,

"addressLine1":
{ "street": "Place de la Concorde", "streetNumber": "1" }

,

"postalCode": "75001"

}

credentialOnFile=ew0KICAgICJ0eXBlIjogew0KICAgICAgICAidW5zY2hlZHVsZWQi-
OiAiQ0lUIg0KICAgIH0sDQogICAgImluaXRpYWxQYXltZW50IjogdHJ1ZQ0KfQ==

{

"type":
{ "unscheduled": "CIT" }

,

"initialPayment": true

}
```

Key Parameter / Object

- If you do not use your own template, we a new template for the first tests for you. All you have to do is add "Template=evo_responsive" to the encrypted data and the cardholderName entered by the customer will automatically be adopted by EVO Payments for the 3D 2.0 process. For the planned / upcoming 3DS-2.0 rollout, EVO Payments will adapt the standard templates accordingly and make them available to you.

- If you use your own merchant template and the cardholder query is not yet integrated in it, you need to integrate the cardholderName yourself.
- Example XSL file:

```
<!-- Cardholdername -->
  <div class="row ccholder">
    <span class="label">
      <xsl:value-of select="EVOEPay/language/strCCHolder"/>
    </span>
    <div class="input">
      <input type="text" value="" id="creditCardHolder" name="creditCardHolder">
        <xsl:attribute     name="value"><xsl:value-of     select="EVOEPay/creditCard-
Holder"/></xsl:attribute>
      </input>
    </div>
  </div>
```

- Example XML file:

```
For each language used:

<strCCHolder>Cardholdername</strCCHolder>
```

- For PaySSL.aspx the cardholderName is a key value pair.

JSON Object – accountInfo

- The more data you transfer to us, the higher the probability that smooth payment processing (frictionless mode) will take effect.
- You should therefore check which data you already have and evaluate internally which data you would like to transfer.

JSON Object – customerInfo (billToCustomer | shipToCustomer)

- Please note that the transfer of address data is mandatory for 3-D Secure 2.0. IMPORTANT: If the delivery address is not identical to the billing address, both addresses must be transferred! In the case of digital goods, the billing address is sufficient.

JSON Object – merchantRiskIndicator

- We strongly recommend to pass the merchantRiskIndicator (shipping method). The shipping type is transferred in the JSON object merchantRiskIndicator in the JSON parameter shippingAddressIndicator.
  This can have a positive effect on smooth payment processing (frictionless mode).

2. Use cases for transaction flagging

Scenario 01 – Credit Card One-Time Payment

- You offer your customers payment by credit card
- Each payment is a one-time payment, and therefore always a newly initiated payment
- You do not use a pseudo card number to store and reuse the card data

Credentials on File (CoF)

- You must use 3-D Secure
- No further adjustments are necessary

Scenario 02 – Credit Card Subscriptions

- You offer your customers payment by credit card
- Customers enter into a subscription with you that ALWAYS maintains the same amount and payment interval

- You use the pseudo card number to store and reuse the card data
- IMPORTANT: The following initial payment is subject to the liability shift for you as a merchant. In the case of the subsequent payment, however, this expires, so that there is no liability shift.

Credentials on File (CoF) – Initial Subscription Payment

- Applies to PaySSL.aspx
- 3-D Secure is mandatory
- Necessary adjustments:
    - Example:
        - JSON object credentialOnFile with JSON parameter recurring (3 keys included)
        - JSON object credentialOnFile with JSON parameter initialPayment and the value "true"
        -
        - Example Initial Subscription Payment:

```
{
  "type": {
    "recurring": {
      "recurringFrequency": 30,
      "recurringStartDate": "2019-09-14",
      "recurringExpiryDate": "2020-09-14"
    }
  },
  "initialPayment": true
}
```

Scenario 03 – Credit Card Recurring Payment / Down Payment / Final Payment

- You offer your customers payment by credit card
- Customers shop repeatedly in your shop using the same credit card data
- You use the pseudo card number to store and reuse the card data
- IMPORTANT: The following initial payment is subject to the liability shift for you as a merchant. In the case of the subsequent payment, however, this expires, so that there is no liability shift.

Credentials on File (CoF) - Initial Recurring Payment

- Applies to PaySSL.aspx
- 3-D Secure is mandatory
- Necessary adjustments:
    - Example:
        - JSON object credentialOnFile with JSON parameter unscheduled and the value "CIT"
        - JSON object credentialOnFile with JSON parameter initialPayment and the value "true"
        - Example Initial Recurring Payment:

```
{
  "type": {
    "unscheduled": "CIT"
  },
  "initialPayment": true
}
```

Credentials on File (CoF) – Subsequent Recurring Payment

- Applies to Direct.aspx
- 3-D Secure is **NOT** mandatory
- Necessary adjustments:
    - Example:

        - Please always send the schemereferenceID from the initial payment, so that the downstream systems can link the two transactions accordingly.
        - JSON object credentialOnFile with JSON parameter unscheduled and the value "MIT"
        - JSON object credentialOnFile with JSON parameter initialPayment and the value "false"
        - Example Subsequent Recurring Payment:

```
{
  "type": {
    "unscheduled": "MIT"
  },
  "initialPayment": false
}
```

Scenario 04 – Credit Card Account Verification

- You offer your customers payment by credit card
- In this scenario, you only want to validate the customer's credit card
- You use the pseudo card number to store and reuse the card data
- IMPORTANT: Currently and in the future, schemes/card brands want to prevent merchants from carrying out card data validations with a minimum amount (e.g. 1 cent authorization). Therefore, EVO E-PAY offers you the corresponding "ZeroValueAuthentication". This is controlled by passing the additional parameter "AccVerify" in the encrypted data – see the example below for details. Please make sure that your credit card acquirer supports this function for you.

<u>Credentials on File (CoF) - Validation Request</u>

- Applies to PaySSL.aspx
- 3-D Secure is mandatory
- Necessary adjustments:
    - Example:
        - Please send the parameter AccVerify=Yes in the encrypted data (for further details please refer to our programming manual)
        - JSON object credentialOnFile with JSON parameter unscheduled and the value "CIT"
        - JSON object credentialOnFile with JSON parameter initialPayment and the value "true"
        - Example Account Verification:

```
{
  "type": {
    "unscheduled": "CIT"
  },
  "initialPayment": true
}
```

Scenario 05 – Credit Card Token Storage / Form Prefill

- You offer your customers payment by credit card
- Customers buy in your shop and you store the credit card data in the form of the pseudo card number
- When the customer returns, you prefill the credit card form with the stored data

<u>Credentials on File (CoF) - Initial Payment for Token Storage</u>

- Applies to PaySSL.aspx
- 3-D Secure is mandatory
- Necessary adjustments:
    - Example:
        - JSON object credentialOnFile with JSON parameter unscheduled and the value "CIT".

        - JSON object credentialOnFile with JSON parameter initialPayment and the value "true"
        - Example Initial Payment for Token Storage:

```
{
  "type": {
    "unscheduled": "CIT"
  },
  "initialPayment": true
}
```

<u>Credentials on File (CoF) - Subsequent Payment for Token Storage</u>

- Applies to PaySSL.aspx
- 3-D Secure is mandatory
- Necessary adjustments:
    - Example:
        - JSON object credentialOnFile with JSON parameter unscheduled and the value "CIT"
        - JSON object credentialOnFile with JSON parameter initialPayment and value "false"
        - Example Subsequent payment für Token Storage:

```
{
  "type": {
    "unscheduled": "CIT"
  },
  "initialPayment": false
}
```

Scenario 06 – Credit Card Recurring Payment incl. Liability Shift (e.g. Travel business)

- IMPORTANT: The following scenario only applies to PCI-certified systems
- There are several scenarios for the travel industry that allow recurring payments to also be subject to liability shift
- Example:
    - Customer books a hotel room via a booking platform, enters his card data and executes 3-D Secure 2.0. This is processed via a separate PSP. This transaction only serves to validate the card data -ZeroValueAuthentication-.
      This results in an Authenticate Status = CAVV, which the central booking platform then reports to the hotel operator (and any other service providers such as rental car agencies, insurance agencies, etc.). The hotel operator makes a NON-3DS 2.0 payment via EVO E-PAY, but including the CAVV and any other data. The second transaction also contains the corresponding liability shift.
- The basis for this to work and for the liability shift to take place is the passing on of the Authenticate Status (CAVV). This is determined via a so-called "External 3DS Authentication". Two steps are necessary:
    a. The external merchant system that initiated the first payment (AccVerify/ZeroValueAuthentication) stores the authentication status
    b. Subsequently, a recurring payment can be made via EVO E-PAY. In this case, the merchant must include the JSON object threeDSData in the JSON data as well as the original card data of the initial authenticate (Card-JSON). The card data must therefore be transferred in its original form from the booking platform to all relevant service providers / agencies in compliance with PCI. For this purpose a separate section explains the necessary steps.
- All necessary technical information can be found in the Multi-party Ecommerce / Agent Model section.

Scenario 07 – Credit Card MoTo (MailOrder / TelephoneOrder) via PaySSL

- You offer your customers payment by credit card, which is collected by telephone.
- The credit card data is entered in a separate call center application which triggers a payment via EVO E-PAY using PaySSL.aspx
- You use the pseudo card number to store and reuse the card data
- IMPORTANT: MoTo payments are not subject to the liability shift as 3-D Secure is not possible. (Out of Scope)

Credentials on File (CoF) - Initial MoTo Payment

- Applies to PaySSL.aspx
- 3-D Secure is not possible (Out of Scope)
- Necessary adjustments:
    - Example:
        - JSON object credentialOnFile with JSON parameter unscheduled and the value "MIT"
        - JSON object credentialOnFile with JSON parameter initialPayment and the value "true"
        - Example Initial MoTo Payment:

```
{
  "type": {
    "unscheduled": "MIT"
  },
  "initialPayment": true
}
```

Scenario 08 – Credit Card MoTo (MailOrder / TelephoneOrder) via Virtual Terminal

- You offer your customers payment by credit card, which is collected by telephone.
- The credit card data is entered via Virtual Terminal.
- IMPORTANT: MoTo payments are not subject to the liability shift as 3-D Secure is not possible. (Out of Scope)

Credentials on File (CoF)

- By using the Virtual Terminal no further adjustments are necessary.

Scenario 09 – Batch Processing

- Due to the constant need for adjustments in the area of batch procedures, please contact the responsible EVO Payments Implementation Manager if you have any questions.

Scenario 10 – Extended Transaction Management (ETM)

- When using the EVO Payments ETM, EVO E-PAY takes care of the correct flagging of transactions for you.

3. Test 3-D Secure 2.0 via EVO Payments

Take the opportunity to test 3-D Secure 2.0 now!
While not all downstream systems currently offer 3-D Secure for testing, you can perform a test simulation within EVO E-PAY. This allows you to perform 3-D Secure authentication including different return values.
Please proceed as follows for testing:

- Activate 3-D Secure 2.0 for your EVO Payments MerchantID. If you are unsure whether it has already been activated, please contact the responsible EVO Payments Implementation Manager.
- In the encrypted data request, use the default parameter OrderDesc with the value "Test:0000". This will give you a correspondingly successful authorization after successful authentication. IMPORTANT: In simulation mode, the schemereferenceID of the initial payment is not returned because this ID is generated by the downstream systems. These systems are not involved in the Simulation.
- Perform 3-D Secure Authentication

3. Test 3-D Secure 2.0 via EVO Payments
• Please **ONLY** use the available Testcards (expiration date always in the future + CVV/CVC may contain any value) • Depending on the desired scenario (e.g. Browser 3-D Secure 2.0 challenge, frictionless browser authentication, etc.), please use the appropriate One-Time Passwords