

Direct/API Integration

ACS Timeout Amendment

v1.1

03/01/2023

Table of Contents

Table of Contents	2
Introduction	3
Intended Audience.....	3
ACS Timeout in 3DS 2.0 Transactions.....	4
Introduction.....	4
Resolution.....	5
Appendix A: Sample code to implement timer	6
Appendix B: 3DS2 Transactions flows	7
Transaction Flow – 3D Secure v2 Frictionless Authentication (Browser)	7
Transaction Flow – 3D Secure v2 Challenge Authentication (Browser)	9
Appendix C: Direct Integration Documentation.....	12
Appendix D: Example Messages.....	13
Response–CardDetailsTransactionResponse (Not requiring 3D Secure authentication)	13
Response–CardDetailsTransactionResponse (Requiring 3D Secure v2 authentication)	14

Introduction

Intended Audience

This document should be used by the developers to make changes to the Direct Integration done with Payment Gateway. It assumes that the reader has knowledge and understanding of internet protocols like HTTPs, SSL and XML/SOAP. However, this introduction section will also provide valuable, non-technical information to any interested non- developer.

ACS Timeout in 3DS 2.0 Transactions

Introduction

The full 3DSv2 flow for frictionless and challenge transactions are detailed below in Appendix B – please refer to these diagrams for a broader understanding of 3DS 2.0 Transaction processing.

As part of the 3DS 2.0 transaction flow, the Payment Gateway will respond to a Card Details Transaction message with “Issuer Authentication Required” and a “ThreeDSecureOutputData” object as shown in Appendix C.

In this instance, the Merchant website or plugin posts the “Method Data” to the “Method URL” and the device fingerprinting process takes place. The expected result is that the Merchant website receives a new “Method Data” which is subsequently sent to the Gateway as part of the “ThreeDSecureEnvironment” request in step 7 of the diagram in Appendix B.

This instructs the Payment Gateway to continue with transaction Authentication either following the Frictionless flow, where the transaction is sent to the Acquirer for Authorisation or the Challenge flow, where the Issuer requests further cardholder Authentication and the user must enter a One Time Password or authorise the transaction via their banking app.

In certain circumstances transactions are not progressing as the Payment Gateway never receives the subsequent “ThreeDSecureEnvironment” request message. In this instance the transaction status remains as “Issuer Authentication Required” or is set to “Issuer Authentication Expired” after a set time limit.

Investigation has shown that this is most likely to be occurring due to the Device Fingerprint process failing either in the ACS or in the Merchant website and the follow-on messages to the Payment Gateway not being completed as a result.

Resolution

After receiving the “MethodData” and “MethodURL” from the Card Details Transaction Response (Appendix C), the Merchant integration should track the time taken to perform the iFrame redirect.

The iFrame redirect process should be completed within 10 seconds however, if it is not completed within 12 seconds the process can be considered as having failed.

The Merchant integration should send the “ThreeDSecureEnvironment” request to the Payment Gateway containing the MethodData originally received in the “CardDetailsTransaction” response.

The Payment Gateway will then use the original Method Data to determine that a failure has occurred and process the transaction accordingly. In this instance it is highly likely that a Challenge Authentication will be required.

It is advisable that the timeout value is set to 12 seconds at this stage. We also recommend that this value is made configurable to account for future changes to the 3DS 2.0 specification being issued by EMVCo.

The 12 second value is derived from EMVCo specification detailing a 10 second failure timeout for the Fingerprinting process and to account for any potential network delay between the Merchant and ACS.

Appendix A gives a suggested example using JavaScript to create a timer to track the iFrame Redirect.

Appendix A: Sample code to implement timer

Note: szActionURL is the Merchant's payment form URL.

```
szScript = "var paymentFormSessionClientID;" +
    "var form1;" +
    "var input;" +
    "form1 = document.createElement(\"form\");" +
    "form1.id = \"ThreeDSecureFingerprintForm\";" +
    "form1.method = \"POST\";" +
    "form1.action = \"\" + szMethodURL + \"\";" +
    "form1.target = \"\" + ifFingerprintFrame.ClientID + \"\";" +

    "input = document.createElement(\"input\");" +
    "input.id = \"threeDSMethodData\";" +
    "input.name = \"threeDSMethodData\";" +
    "input.type = \"hidden\";" +
    "input.value = \"\" + szMethodData + \"\";" +
    "form1.appendChild(input);" +

    "document.body.appendChild(form1);" +
    "form1.submit();" +

    "var form2;" +
    "paymentFormSessionClientID = parent.document.getElementById(\"\" +
PaymentFormSession.ClientID + \"\");" +
    "if(paymentFormSessionClientID == null ||" +
    "paymentFormSessionClientID === undefined)" +
    "{" +
        "throw new Error(\"Error 6345: Couldn't find session holder.\");" +
    "}" +

    "form2 = document.createElement(\"form\");" +
    "form2.id = \"FingerprintNotificationForm\";" +
    "form2.method = \"POST\";" +
    "form2.action = \"\" + szActionURL + \"\";" +

    "input = document.createElement(\"input\");" +
    "input.id = \"threeDSMethodData\";" +
    "input.name = \"threeDSMethodData\";" +
    "input.type = \"hidden\";" +
    "input.value = \"\" + szMethodData + \"\";" +
    "form2.appendChild(input);" +

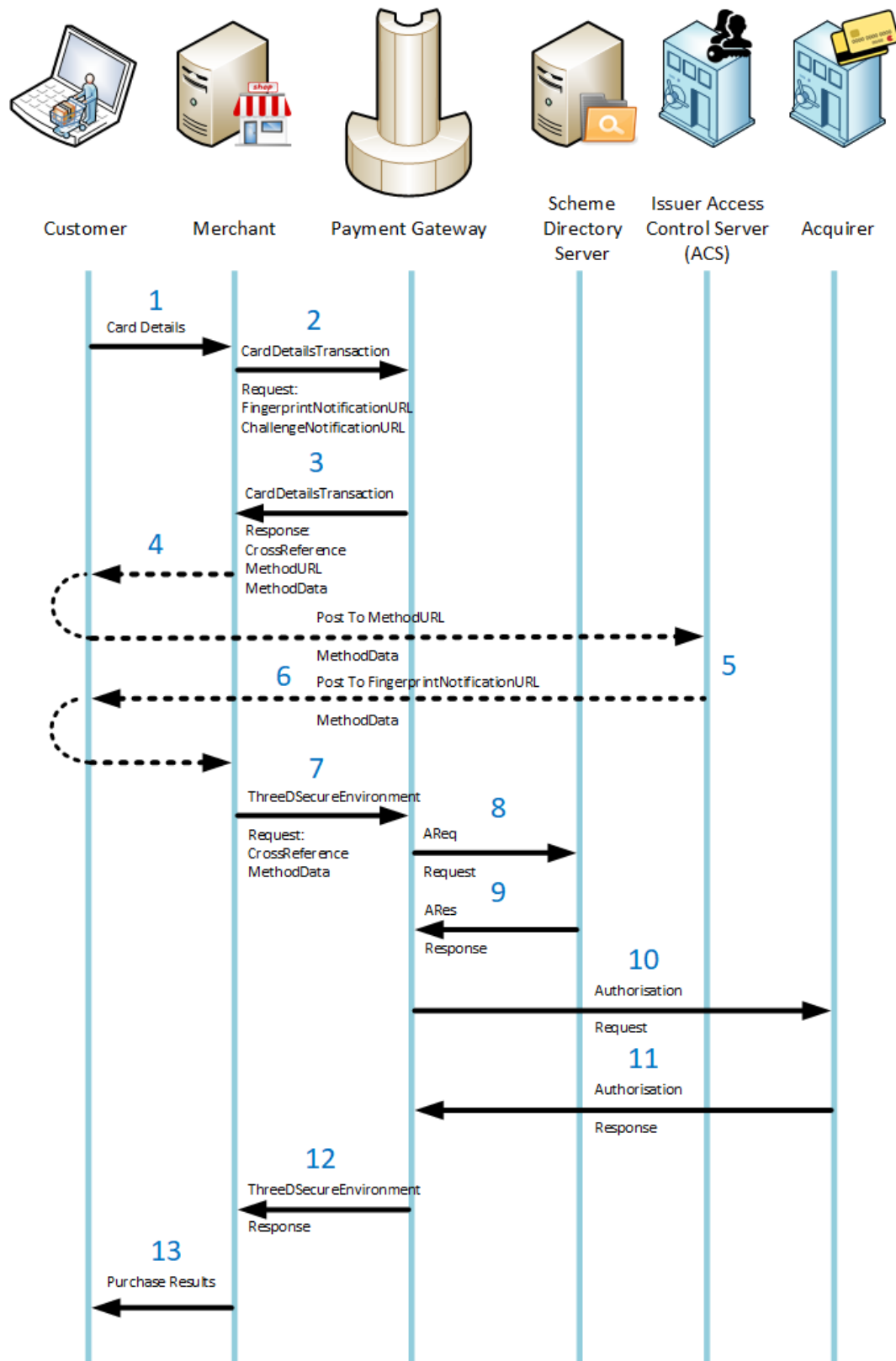
    "input = document.createElement(\"input\");" +
    "input.id = \"PaymentFormSession\";" +
    "input.name = \"PaymentFormSession\";" +
    "input.type = \"hidden\";" +
    "input.value = paymentFormSessionClientID.value;" +
    "form2.appendChild(input);" +

    "document.body.appendChild(form2);" +
    "setTimeout(() => { form2.submit(); }, " + "12000" + ");";

ScriptManager.RegisterStartupScript(this, GetType(), "ThreeDSecureFingerprintForm",
szScript, true);
```

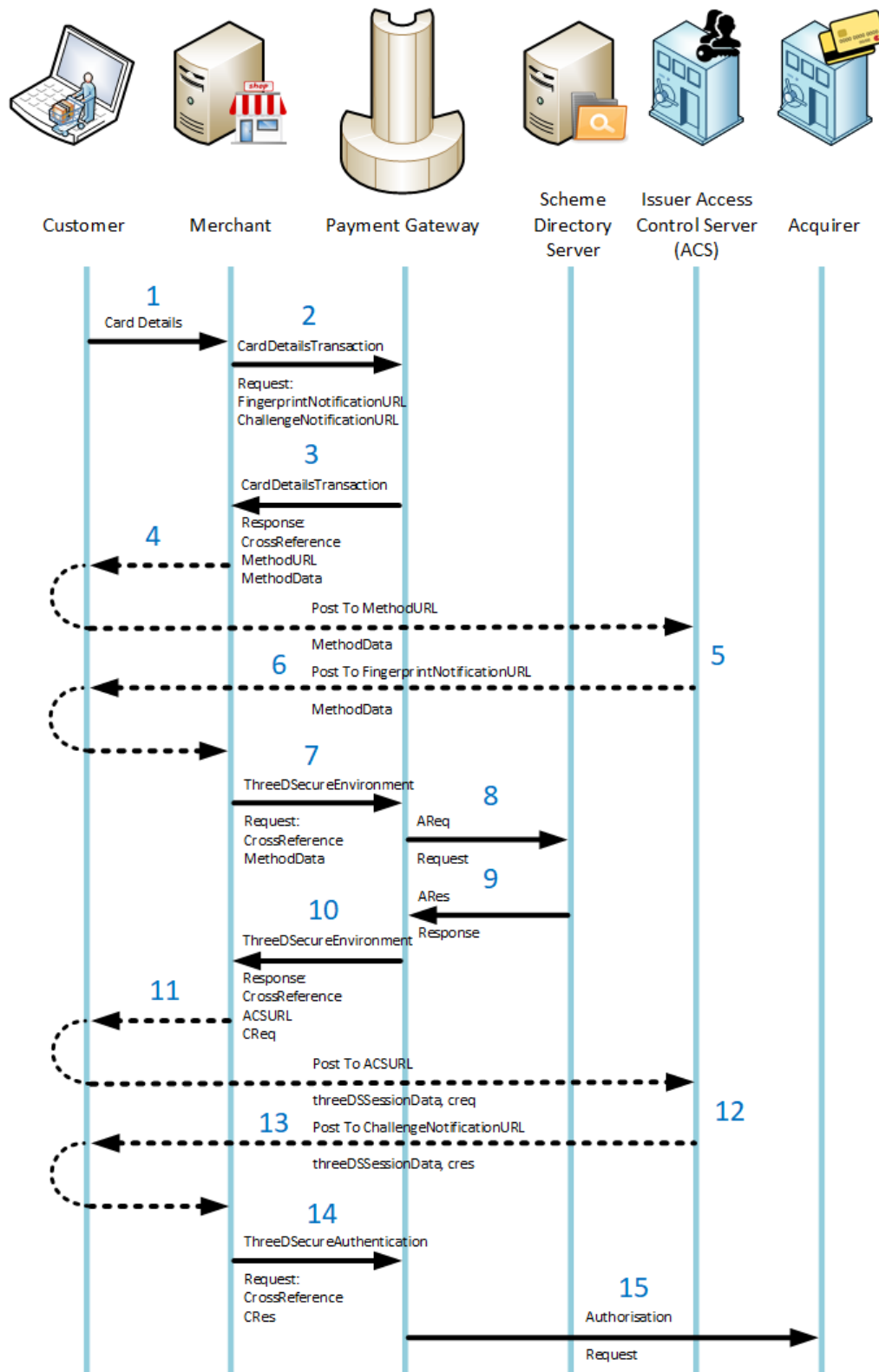
Appendix B: 3DS2 Transactions flows

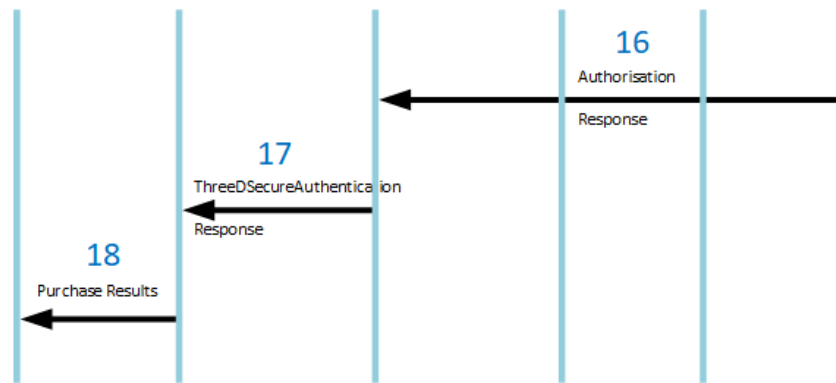
Transaction Flow – 3D Secure v2 Frictionless Authentication (Browser)



- 1) As part of the checkout process, the cardholder enters their card details and submits the details to the merchant's webshop
- 2) The merchant sends the purchase details in a CardDetailsTransaction request to the Payment Gateway, which includes values for FingerprintNotificationURL and ChallengeNotificationURL
- 3) The Payment Gateway checks the card details, and on determining that the card is enrolled on the 3D Secure scheme, requires device fingerprint analysis as part of the 3D Secure checkout process. The Payment Gateway parks the transaction details and responds with a CardDetailsTransaction response that includes the fields CrossReference, MethodData and MethodURL
- 4) The merchant's webshop stores the CrossReference using a mechanism that can be retrieved later (e.g. in a cookie) and creates a hidden iframe that contains a form which forces the customer's browser to post the MethodData over to the MethodURL
- 5) The Issuer's Access Control Server (ACS) performs device fingerprint analysis on the customer's browser
- 6) The ACS automatically forces the customer's browser to post the resultant MethodData back to the merchant's nominated FingerprintNotificationURL
- 7) The merchant's webshop retrieves the previously stored CrossReference and sends this and the MethodData back to the Payment Gateway in a ThreeDSecureEnvironment message
- 8) The Payment Gateway sends an Authorisation Request (AReq) to the Scheme Directory Server
- 9) The Scheme Directory Server (DS) analyses the transaction details (including the device fingerprint results) and responds with an Authorisation Response (ARes) that indicates the transaction can continue without further authentication
- 10) The Payment Gateway sends an authorisation request to the merchant's Acquirer
- 11) The Acquirer sends the transactions onto the customer's issuing bank's authorisation host, who authorise or decline the transaction and return the results of the authorisation back to the Payment Gateway
- 12) The Payment Gateway returns the transaction response to the merchant's webshop in a ThreeDSecureEnvironment response
- 13) The merchant's webshop displays the results of the purchase to the customer

Transaction Flow – 3D Secure v2 Challenge Authentication (Browser)





- 1) As part of the checkout process, the cardholder enters their card details and submits the details to the merchant's webshop
- 2) The merchant sends the purchase details in a CardDetailsTransaction request to the Payment Gateway, which includes values for FingerprintNotificationURL and ChallengeNotificationURL
- 3) The Payment Gateway checks the card details, and on determining that the card is enrolled on the 3D Secure scheme, requires device fingerprint analysis as part of the 3D Secure checkout process. The Payment Gateway parks the transaction details and responds with a CardDetailsTransaction response that includes the fields CrossReference, MethodData and MethodURL
- 4) The merchant's webshop stores the CrossReference using a mechanism that can be retrieved later (e.g. in a cookie) and creates a hidden iframe that contains a form which forces the customer's browser to post the MethodData over to the MethodURL
- 5) The Issuer's Access Control Server (ACS) performs device fingerprint analysis on the customer's browser
- 6) The ACS automatically forces the customer's browser to post the resultant MethodData back to the merchant's nominated FingerprintNotificationURL
- 7) The merchant's webshop retrieves the previously stored CrossReference and sends this and the MethodData back to the Payment Gateway in a ThreeDSecureEnvironment message
- 8) The Payment Gateway sends an Authorisation Request (AReq) to the Scheme Directory Server
- 9) The Scheme Directory Server (DS) analyses the transaction details (including the device fingerprint results) and responds with an Authorisation Response (ARes) that indicates the customer must further authenticate themselves with their issuing bank
- 10) The Payment Gateway builds a Challenge Request (CReq) and returns this with the ACSURL and a CrossReference to the merchant's webshop in a ThreeDSecureEnvironment response
- 11) The merchant's webshop builds a form that forces the customer's browser to post the CReq (as "creq") and the CrossReference (as "threeDSSessionData") over to the ACSURL
- 12) The customer completes the challenge authentication with their issuing bank's ACS
- 13) The ACS automatically forces the customer's browser to post the resultant Challenge Response (as "cres") and CrossReference (as "threeDSSessionData") back to the merchant's nominated ChallengeNotificationURL
- 14) The merchant's webshop sends the CrossReference and the CRes back to the Payment Gateway in a ThreeDSecureAuthentication message
- 15) The Payment Gateway checks the authentication status and determines whether the transaction should be declined or sent onto the acquirer for authorisation. If the transaction should continue for authorisation, the Payment Gateway sends an authorisation request to the merchant's Acquirer
- 16) The Acquirer sends the transactions onto the customer's issuing bank's authorisation host, who authorise or decline the transaction and return the results of the authorisation back to the Payment Gateway
- 17) The Payment Gateway returns the transaction response to the merchant's webshop in a ThreeDSecureAuthentication response
- 18) The merchant's webshop displays the results of the purchase to the customer

Appendix C: Direct Integration Documentation

This amendment references the “CardDetailsTransction” and “ThreeDSecureEnvironment” messages on pages 8 and 26 respectively.



DirectIntegration.p
df

Note: The documentation is v2.6.5 issued in December 2022. Newer versions of the Direct Integration Documentation may be available via the MMS.

Appendix D: Example Messages

Response – CardDetailsTransactionResponse (Not requiring 3D Secure authentication)

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/enve
lope/">
  <soap:Body>
    <CardDetailsTransactionResponse xmlns="https://www.thepaymentgateway.net/">
      <CardDetailsTransactionResult AuthorisationAttempted="true">
        <StatusCode>0</StatusCode>
        <Message>Auth Code: 123456</Message>
      </CardDetailsTransactionResult>
      <TransactionOutputData CrossReference="07010101010010101010102"
ExternalCrossReference="58792488326099422839"
ExternalClientReference="15764860733312436602"
ExternalTransactionUID="00342830812924528849">
        <AuthCode>123456</AuthCode>
        <AddressNumericCheckResult>UNKNOWN</AddressNumericCheckResult>
        <PostCodeCheckResult>UNKNOWN</PostCodeCheckResult>
        <CV2CheckResult>PASSED</CV2CheckResult>
        <CardTypeData>
          <CardType>MAESTRO_INTERNATIONAL</CardType>
          <Issuer>HSBC</Issuer>
        </CardTypeData>
        <AmountReceived>1000</AmountReceived>
        <GatewayEntryPoints>
          <GatewayEntryPoint EntryPointURL="https://gw1.paymentprocessor.net"
Metric="100" />
          <GatewayEntryPoint EntryPointURL="https://gw2.paymentprocessor.net"
Metric="200" />
        </GatewayEntryPoints>
      </TransactionOutputData>
    </CardDetailsTransactionResponse>
  </soap:Body>
</soap:Envelop
e>
```

Response – CardDetailsTransactionResponse (Requiring 3D Secure v2 authentication)

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CardDetailsTransactionResponse xmlns="https://www.thepaymentgateway.net/">
      <CardDetailsTransactionResult AuthorisationAttempted="true">
        <StatusCode>2</StatusCode>
        <Message>Environment analysis required</Message>
      </CardDetailsTransactionResult>
      <TransactionOutputData CrossReference="070101010100101010102">
        <CardTypeData>
          <CardType>MAESTRO_INTERNATIONAL</CardType>
          <Issuer>HSBC</Issuer>
        </CardTypeData>
        <AmountReceived>1000</AmountReceived>
        <ThreeDSecureOutputData>
          <MethodData>
            eyJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIjoiaHR0cHM6XC9cL2d3MS5pcmlkaXVtY29ycC5uZXRcL
            0Rpc3BsYXlBbGxwYXJpYWJsZXMuYXNweCIsInRocmVlRFNTZXJ2ZXJUcmFuc01EIjoimTViZmZjYzYtZGU1Mi
            00MDgwLTllZjMtMzc3OTIyMDliMGUzIn0
          </MethodData>
          <MethodURL>https://www.bank.com/acs</MethodURL>
        </ThreeDSecureOutputData>
        <GatewayEntryPoints>
          <GatewayEntryPoint EntryPointURL="https://gw1.paymentprocessor.net"
            Metric="100" />
          <GatewayEntryPoint EntryPointURL="https://gw2.paymentprocessor.net"
            Metric="200" />
        </GatewayEntryPoints>
      </TransactionOutputData>
    </CardDetailsTransactionResponse>
  </soap:Body>
</soap:Envelope>
```