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

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 Computop

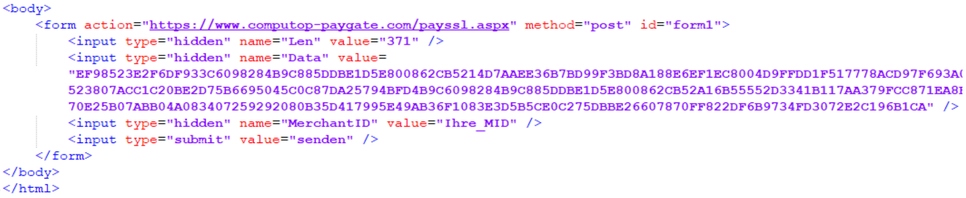
General technical adjustments

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

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

How will data transfer look like with 3-D Secure 2.x?

- 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://www.computop-paygate.com/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 excecuted 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 3-D Secure 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 Paygate on your behalf. Please contact Computop Helpdesk 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, Computop Paygate 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 Computop Paygate.
- Please pass JSON Objects with values only. Empty or zero-filled objects/parameters lead to a rejection.

JSON Example request - encrypted data

ⓘ BASEURL=https://www.computop-paygate.com/

```
{{BASEURL}}payssl.aspx?
MerchantID=Generic3DSTest&len=1800&data=CDC44E5A9D2C8A559CEDF1CCA97C9FBD3D90E046BFBF96F06ADA9A00FB0BC3494317E8D92
4FF44729671B93348B477F880541ACFF12C8E3A868CD55FEA95219C245CF7F4716FCF3462167A8B63D11424FA7BD30891504F8465C5680597
5115EB71C0A04E5D7466D771495035749FFF94D3087529F578DEF518003EA1422F6DE7B7DFD78A0DD695550623A42BF41A422EC219012318
FE26D2B757F12BDFE046EA4CB8D35079ABAB6859691FEE1B03483471495035749FFF94D3087529F578DEF518003EA1422F6DE7D4E20259A4
84D23A9EFC7F4ADB209DD67D8EDE5BD2AC0CB2682D7CF26A6624A54BCF4E93219ADD89ABA6214820D4BAA5A9A184DD7F8AF3E2BE98C
5B63113276B023B92DA5AADCCD7387B71B6651A0E7E4E42F8790122386AA9A184DD7F8AF3E23CFEC0086B59B6A9D98EB96DFDA496D97D
85706A4A810056FE48AB878EFC1E976DB7504D402F4B96778B45ADE1DF3E217EFFBA566359677AB73F514F1E75F11DBE3E15983BA530E7D
5B13A87D1A2ED19A9A184DD7F8AF3E21D32D652A71B2A49A58F3A30256097DA11388C26E7CBEB12E65B31C485C94DE8179CEACDE9237E
F4C426A05E594E28069E10B19AE173D25A93A546845C5D78C44112031D6D5DE9B4ABA6214820D4BAA5A9A184DD7F8AF3E260C35EC59CD2
FAA2435CD631BFC801AA7C72A1BAE39879C0BF733EDC45DD99F3A9A184DD7F8AF3E2DDA25A6458507ACE3B3CAAC3A4B293A9C6177F7F
00EBFB6924050D9DF661DE8EC204863D819ABF9564498E9F2D72BEFF2E040214C4961D8737821BA1F638BE05FB01E1B382733FC42D6B04A
B80D66218C75E691B9475C5F6CF13AD357057BC6B5864EE113DF2272EF6572101D5E45CB634F3E941FA7B3EA7E636EAEF751C67C82F8E8
D9B618E69826221B2A42D7F694D9E10B19AE173D25A6EB48BD63BFFE0FAFC78722BD9FFA39623B5D40494B96D2A9E10B19AE173D25A188
DA61C8E3401850C400A3144C3547808A0C82C7B8E9863D017852B02FBFE6D62983EBC372B1A8108D832C13F92E88535C213D0FDA1B1A5C4
26A05E594E28069E10B19AE173D25A92AD74641E23F21D1D66F1B352AFCCD408B1727FACC2405AA9A184DD7F8AF3E29B3106F31EE7D473
A854D99576FDD5620141A96DEF638FCE4362F90866AED8044E42F8790122386AA9A184DD7F8AF3E20F6BF2E070199426696A900FEEBC7848
B6F72D445F2CB9F0ED160CC32B1A3C40C426A05E594E28069E10B19AE173D25A201E55FC81E8F7CD78FFD98E342897C11AB2BE505B3E84
21C63E936DCCF29058C31D72A3697DA2C89EFC7F4ADB209DD67D8EDE5BD2AC0CB2682D7CF26A6624A54BCF4E93219ADD89ABA6214820
D4BAA5A9A184DD7F8AF3E2BE98C5B63113276B023B92DA5AADCCD7387B71B6651A0E7E4E42F8790122386AA9A184DD7F8AF3E23CFEC00
86B59B6A9D98EB96DFDA496D93F669AB8A34E11706F7B3F762241F749A9A184DD7F8AF3E286587E20CD9A354709F67B1501183CFC5D6FD3
FD6E23B0D4FA9746B8925D4A4FA9A184DD7F8AF3E21D32D652A71B2A49A58F3A30256097DA11388C26E7CBEB120758D07B77A47DB34E35
9C7AE383D69BC426A05E594E28069E10B19AE173D25A93A546845C5D78C44112031D6D5DE9B4ABA6214820D4BAA5A9A184DD7F8AF3E260
C35EC59CD2FAA2435CD631BFC801AA7C72A1BAE39879C0BF733EDC45DD99F3A9A184DD7F8AF3E2DDA25A6458507ACE3B3CAAC3A4B29
3A9C6177F7F00EBFB6924050D9DF661DE8EC204863D819ABF9564498E9F2D72BEFF2E040214C4961D8737821BA1F638BE05FB01E1B38273
3FC46AA58C2847221D78069144B06DE3755A6C88EADD3B3FCCD6F6572101D5E45CB634F3E941FA7B3EA7B08783F57D9AD1BAB2071FAB9
B93B3C13FF102AD44B6A493B5C341FB37BF525B0A0E4F490BE1D46A4C5B8F691A2020868119A0AEB9E9BCD4F9D783FEA316723E17976FB
B4909040AE279D66AF13B8441582CB00BB30835AB6401E5CDDF295F533AEE31D2677314D288F2C15BFB16837EF4A779C1E39E4AA1CEE13
FABDB2B89D9A7A89ED81EC005BCD416330CFCE5CF716A316FDF29A9CFF3F25490656C800BCA582CB00BB30835ABABD19D247E68289A52
F1387D978126C967F9BBB890618AF5A0E5136C7DC2892D2460687217D2779B5836D3F1FFAE8F3B582CB00BB30835ABEE02C59E0AAF8C913
339B61F9DDFB7DAC4FF2460869E4876C5DFD5D39E79330D427654226D9E37E72D7A4C332F59563DF70B3A840877E2B1BF739A2347A73347
F7DA9F100EEBC189ADE92F98BE65BCBE29FE1A3DFE89E44EEEBF9C902BBAA7C2F68CBC48C724B889A53EA148988B56CC52D52743C045
F57844F6607DDEA75FE613EAC80E2C02BCEA89B71E52E64D7538DC9B82EB2740B82C698F43B6A62D770935233D5F10E593D0519511BAAD
615B0035D7524B097C29BA39EBEBEDB93425EB7824B9CCDB1397E716993ED326500615B4B1853A59F760A0E06373BDFE1CC6695A93B1585
1F56428&template=ct_responsive_ch&language=en
```

JSON Example request - request parameter before encryption

```
MerchantID=Generic3DSTest
&MsgVer=2.0
&TransID=1234567890
&RefNr=20200124
&Amount=100
&Currency=EUR
&URLNotify=https://www.computop.com
&URLSuccess=https://www.computop.com
&URLFailure=https://www.computop.com
&billToCustomer=ew0KICAgICJjb25zdW1lciI6IHsNCiAgICAgICAgInNhbHV0YXRpb24iOiAiTXIiLA0KICAgICAgICAiZmlyc3ROYW1lI
jogIk5hcG9sZW9uIiwNCiAgICAgICAgImxhc3ROYW1lIjogIkJvbmFwYXJ0ZSIsDQogICAgICAgICJiaXJ0aERhdGUiOiAiMTc2OS0wOC0xNS
INCiAgICB9LA0KICAgICJtb2JpbGVQaG9uZSI6IHsNCiAgICAgICAgImNvdW50cnlDb2RlIjogIjMzIiwNCiAgICAgICAgInN1YnNjcmliZXJ
OdW1iZXIiIDogIjEyMzQ1Njc4OTEwIg0KICAgIH0sDQogICAgImVtYWlsIjogIm5hcG9sZW9uLmJvbmFwYXJ0ZUBmcmFuY2UuY29tIg0KfQ==
&billingAddress=ew0KICAgICJjaXR5IjogIkFqYWNjaW8iLA0KICAgICJjb3VudHJ5Ijogew0KICAgICAgICAiY291bnRyeUEzIjogIkZSQ
SINCiAgICB9LA0KICAgICJhZGRyZXNzTGluZTEiOiB7DQogICAgICAgICJzdHJlZXQiOiAiRXhpbGVzdHJlZXQiLA0KICAgICAgICAic3RyZW
V0TnVtYmVyIjogIjI3MCINCiAgICB9LA0KICAgICJwb3N0YWxDb2RlIjogIjIwMTY3Ig0KfQ==
&shipToCustomer=ew0KICAgICJjb25zdW1lciI6IHsNCiAgICAgICAgInNhbHV0YXRpb24iOiAiTXIiLA0KICAgICAgICAiZmlyc3ROYW1lI
jogIkx1ZHdpZyIsDQogICAgICAgICJsYXN0TmFtZSI6ICJYVklJSSIsDQogICAgICAgICJiaXJ0aERhdGUiOiAiMTc1NS0xMS0xNyINCiAgIC
B9LA0KICAgICJtb2JpbGVQaG9uZSI6IHsNCiAgICAgICAgImNvdW50cnlDb2RlIjogIjMzIiwNCiAgICAgICAgInN1YnNjcmliZXJOdW1iZXI
iIDogIjEyMzQ1Njc4OTEwIg0KICAgIH0sDQogICAgImVtYWlsIjogIkx1ZHdpZ0Byb3lhbC5mcmFuY2UuY29tIg0KfQ==
&shippingAddress=ew0KICAgICJjaXR5IjogIlBhcmlzIiwNCiAgICAiY291bnRyeSI6IHsNCiAgICAgICAgImNvdW50cnlBMyI6ICJGUkEi
DQogICAgfSwNCiAgICAiYWRkcmVzc0xpbmUxIjogew0KICAgICAgICAic3RyZWV0IjogIlBsYWNlIGRlIGxhIENvbmNvcmRlIiwNCiAgICAgI
CAgInN0cmVldE51bWJlciI6ICIxIg0KICAgIH0sDQogICAgInBvc3RhbENvZGUiOiAiNzUwMDEiDQp9
&credentialOnFile=ew0KICAgICJ0eXBlIjogew0KICAgICAgICAidW5zY2hlZHVsZWQiOiAiQ0lUIg0KICAgIH0sDQogICAgImluaXRpYWx
```

```
QYXltZW50IjogdHJ1ZQ0KfQ==
&OrderDesc=Test:0000


------

where:
billToCustomer=ew0KICAgICJjb25zdW1lciI6IHsNCiAgICAgICAgInNhbHV0YXRpb24iOiAiTXIiLA0KICAgICAgICAiZmlyc3ROYW1lIj
ogIk5hcG9sZW9uIiwNCiAgICAgICAgImxhc3ROYW1lIjogIkJvbmFwYXJ0ZSIsDQogICAgICAgICJiaXJ0aERhdGUiOiAiMTc2OS0wOC0xNSI
NCiAgICB9LA0KICAgICJtb2JpbGVQaG9uZSI6IHsNCiAgICAgICAgImNvdW50cnlDb2RlIjogIjMzIiwNCiAgICAgICAgInN1YnNjcmliZXJO
dW1iZXIiIDogIjEyMzQ1Njc4OTEwIg0KICAgIH0sDQogICAgImVtYWlsIjogIm5hcG9sZW9uLmJvbmFwYXJ0ZUBmcmFuY2UuY29tIg0KfQ==
is base64-encoding of
{
"consumer":
{ "salutation": "Mr", "firstName": "Napoleon", "lastName": "Bonaparte", "birthDate": "1769-08-15" },
"mobilePhone": { "countryCode": "33", "subscriberNumber" : "12345678910" },
"email": "napoleon.bonaparte@france.com"
}

billingAddress=ew0KICAgICJjaXR5IjogIkFqYWNjaW8iLA0KICAgICJjb3VudHJ5Ijogew0KICAgICAgICAiY291bnRyeUEzIjogIkZSQS
INCiAgICB9LA0KICAgICJhZGRyZXNzTGluZTEiOiB7DQogICAgICAgICJzdHJlZXQiOiAiRXhpbGVzdHJlZXQiLA0KICAgICAgICAic3RyZWV
0TnVtYmVyIjogIjI3MCINCiAgICB9LA0KICAgICJwb3N0YWxDb2RlIjogIjIwMTY3Ig0KfQ==
is base64-encoding of
{
"city": "Ajaccio",
"country": { "countryA3": "FRA" },
"addressLine1": { "street": "Exilestreet", "streetNumber": "270" },
"postalCode": "20167"
}

shipToCustomer=ew0KICAgICJjb25zdW1lciI6IHsNCiAgICAgICAgInNhbHV0YXRpb24iOiAiTXIiLA0KICAgICAgICAiZmlyc3ROYW1lIj
ogIkx1ZHdpZyIsDQogICAgICAgICJsYXN0TmFtZSI6ICJYVklJSSIsDQogICAgICAgICJiaXJ0aERhdGUiOiAiMTc1NS0xMS0xNyINCiAgICB
9LA0KICAgICJtb2JpbGVQaG9uZSI6IHsNCiAgICAgICAgImNvdW50cnlDb2RlIjogIjMzIiwNCiAgICAgICAgInN1YnNjcmliZXJOdW1iZXIi
IDogIjEyMzQ1Njc4OTEwIg0KICAgIH0sDQogICAgImVtYWlsIjogIkx1ZHdpZ0Byb3lhbC5mcmFuY2UuY29tIg0KfQ==
is base64-encoding of
{
"consumer": { "salutation": "Mr", "firstName": "Ludwig", "lastName": "XVIII", "birthDate": "1755-11-17" },
"mobilePhone": { "countryCode": "33", "subscriberNumber" : "12345678910" },
"email": "Ludwig@royal.france.com"
}

shippingAddress=ew0KICAgICJjaXR5IjogIlBhcmlzIiwNCiAgICAiY291bnRyeSI6IHsNCiAgICAgICAgImNvdW50cnlBMyI6ICJGUkEiD
QogICAgfSwNCiAgICAiYWRkcmVzc0xpbmUxIjogew0KICAgICAgICAic3RyZWV0IjogIlBsYWNlIGRlIGxhIENvbmNvcmRlIiwNCiAgICAgIC
AgInN0cmVldE51bWJlciI6ICIxIg0KICAgIH0sDQogICAgInBvc3RhbENvZGUiOiAiNzUwMDEiDQp9
is base64-encoding of
{
"city": "Paris",
"country": { "countryA3": "FRA" },
"addressLine1": { "street": "Place de la Concorde", "streetNumber": "1" },
"postalCode": "75001"
}

credentialOnFile=ew0KICAgICJ0eXBlIjogew0KICAgICAgICAidW5zY2hlZHVsZWQiOiAiQ0lUIg0KICAgIH0sDQogICAgImluaXRpYWxQ
YXltZW50IjogdHJ1ZQ0KfQ==
is base64-encoding of
{
"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=ct_responsive_ch" to the encrypted data and the cardholderName entered by the customer will automatically be adopted by Computop for the 3-D 2.0 process. For the planned / upcoming 3DS-2.0 rollout, Computop 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="paygate/language/strCCHolder"/>
                </span>
                <div class="input">
                        <input type="text" value="" id="creditCardHolder" name="creditCardHolder">
                                <xsl:attribute name="value"><xsl:value-of select="paygate/creditCardHolder"
/></xsl:attribute>
                        </input>
                </div>
        </div>
```

- Example XML file:

```
For each language used:
<strCCHolder>Cardholdername</strCCHolder>
```

- For PaySSL.aspx | PayNow.aspx, the cardholderName is a key value pair.
- For Direct.aspx, the cardholderName is a JSON Parameter of the JSON-Objects "Card".

JSON Object – browserInfo

- For PaySSL.aspx Computop captures the browser data on your behalf and there is no need for you to take any action.
- For a server-to-server connection via Direct.aspx or the use of PayNow.aspx, the individual additional queries must be implemented in the shop.

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).

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 + PayNow.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
}
```

CB2A - flexible Subscription

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

Credentials on File (CoF) – Subsequent Subscription 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 recurring (3 keys included)
        - JSON object credentialOnFile with JSON parameter initialPayment and value "false"
        - Example Subsequent Subscription Payment:

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

CB2A - flexible Subscription

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

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 + PayNow.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 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, Computop Paygate 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.

Credentials on File (CoF) - Validation Request

- Applies to PaySSL.aspx + PayNow.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 – PayNow interface

ⓘ The processes described below are subject to the liability shift for you as a merchant.

- 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

A: When the customer returns, you prefill the credit card form with the saved data. In the case of CIT with initial=false, flagging is used if the merchant prefills the pseudo card number to the customer using a template prefill option.

B: When the customer returns, you prefill the credit card form with the saved data. If the customer deletes the existing card number and stores a new one or, if necessary, adds another card, then the flagging for CIT must be used again (initial=true), if the customer also wants this card to be preassigned.

Credentials on File (CoF) – Initial Payment for Token Storage

- Applies to PayNow.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
}
```

Credentials on File (CoF) – Subsequent Payment for Token Storage

- Applies to PayNow.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"

        - Please always send the schemereferenceID from the initial payment (COF-CIT-TRUE) so that the downstream systems can link the two transactions accordingly
        - Example subsequent payment für Token Storage:

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

Szenario 05 – Credit Card Token Storage / Form Prefill – PaySSL interface

The obligatory control of the initial and recurring payments is managed by Paygate for you. So that we can activate this function for you, please contact the Computop Helpdesk.

Please note the following requirements in relation to merchant specific or Computop standard templates (ct_responsive, Cards_v1).

- Applies to PaySSL.aspx
- 3-D Secure is mandatory
- The shift in liability is subject to the liability shift for you as a merchant
- Necessary adjustments (merchant template):
    - Please integrate a prefill checkbox using the code snippet provided below
    - Please activate (comment in) the code contained in the XSL file for preassigning the pseudo card number
    - If the customer decides to save the card data (clicks the checkbox), the merchant system will receive the additional, encrypted response parameter "prefill=on". If the checkbox is not clicked, there is no additional notification about the prefill parameter.
    - If you want to prefill the form with the card data for the registered customer, please deliver in the encrypted data the Card JSON object and within it send the existing pseudo card number (token), expiration date, card brand and the credit card holder.
    - Please do NOT submit a Credential On File object for the initial request as well as for further pre-filled payment.
- Necessary adjustments (Computop standard form):
    - If you would like the prefill checkbox to be displayed on Computop's own form, please let us know via the Computop Helpdesk. Then we will configure the complete function (checkbox + COF control) for you.
    - Please do NOT send a Credential On File object for the initial request as well as for further pre-filled payment.
    - If the customer decides to save the card data (clicks the checkbox), the merchant system will receive the additional, encrypted response parameter "prefill=on". If the checkbox is not clicked, there is no additional notification about the prefill parameter.
    - If you want to prefill the form with the card data for the registered customer, please deliver in the encrypted data the Card JSON object and within it send the existing pseudo card number (token), expiration date, card brand and the credit card holder

JavaScript

```
// Adds a Change-Event to the checkbox called 'cbPrefill'
$("#cbPrefill").change(function() {
 if ($("#cbPrefill").is(':checked'))
{   // If the checkbox was activated, the value of hiddenfield with name 'prefill' is set to 'on'   $("input
[type='hidden'][name='prefill']").val('on');  }

else
{   // If the checkbox was deactivated, the value of hiddenfield with name 'prefill' is deleted again
$("input[type='hidden'][name='prefill']").val('');  }

});
// In case of retries (If form gets called a second time due to errors),
// the last status will be set
if ($("input[type='hidden'][name='prefill']").val() == 'on') {
 $("#cbPrefill").attr('checked', true);
}
```

XSL

```
<!-- Hiddenfield, which tells Paygate that prefill function should get activated -->
<!-- value="on" means that Paygate will return 'prefill=on' in the response to merchant -->
<!-- value=""  means that Paygate will not return 'prefill=on' in the response to merchant -->
<input type="hidden" name="prefill"><xsl:attribute name="value"><xsl:value-of select="paygate/prefill"/><
/xsl:attribute></input>

<!-- Checkbox to (de)activate prefill function -->
 <div id="div_cbPrefill" class="div_cbPrefill">
  <input type="checkbox" name="cbPrefill" id="cbPrefill"></input>
  <span><xsl:value-of select="paygate/language/strCCSaveData" disable-output-escaping="yes"/></span>
  <div class="row"></div>
 </div>
```

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 Computop Paygate, 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:
    1. The external merchant system that initiated the first payment (AccVerify/ZeroValueAuthentication) stores the authentication status
    2. Subsequently, a recurring payment can be made via Computop Paygate. 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, Direct or PayNow

- You offer your customers payment by credit card, which is collected by telephone.
- The credit card data is entered in a separate call centre application which triggers a payment via Computop Paygate using PaySSL.aspx, Direct.aspx or Paynow.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, PayNow.aspx und Direct.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,
     "useCase": "ucof"
}
```

Credentials on File (CoF) - Subsequent MoTo Payment

- Applies to automated payment initiation via Direct.aspx
- 3-D Secure is not possible (Out of Scope)
- 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 value "false"
        - Example Subsequent MoTo Payment:

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

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

The requirement of the card brands to mark recurring transactions correctly is already an older requirement; we provided information on this in January 2019. In the course of the PSD2-SCA implementation, this will become fully mandatory so that recurring transactions can be clearly identified and the downstream systems can recognize why 3-D Secure was not processed in these cases. Since in those cases no end customer actively participates in the payment process, as consequence, the implementation is mandatory for all merchants.

Below you will find our descriptions & details, i.e. in the first step the initial payment from the store has to be marked as CredentialOnFile and based on that the recurring batch transactions will follow.

***initial shop transaction or AccountVerification:**

On our application page the correct initial flagging (CredentialOnFile) is described for different use cases and the following chapter + scenario will apply to you.

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

Chapter: 2. **use cases for transaction identification**
Scenario 03 - Credit Card Recurring Payment / Deposit Payment / Final Payment

***Recurring batch submission**
Since the schemeReferenceID was generated from the initial store request and reported back, that initial value must be used for all subsequent batch submissions + RTF=M (M=Merchant Initiated Transaction) needs to be included & specified.
Card+processing+EN

Chapter: **Batch usage of the interface**

affected batch format / action:

CC,Sale,<Amount>,<Currency>,<TransID>,(<RefNr>),<CCBrand>,<CCNr|PCNr>,<CCExpiry>,<OrderDesc>,<textfeld1>,<textfeld2>,<RTF>,<cardholder>,<transactionID>,<schemeReferenceID>
CC,Authorize,<Amount>,<Currency>,<TransID>,(<RefNr>),<CCBrand>,<CCNr|PCNr>,<CCExpiry>,<OrderDesc>,<textfeld1>,<textfeld2>,<RTF>,<cardholder>,<transactionID>,<schemeReferenceID>

In the batch request there exists the "transactionID" on the one hand and in addition the "schemeReferenceID" as well and we recommend that you take field schemeReferenceID.

Examples Batch Versions 1.2 schemeReferenceID field delivery:
CC,Sale,<Amount>,<Currency>,<TransID>,(<RefNr>),<CCBrand>,<CCNr|PCNr>,<CCExpiry>,<OrderDesc>,<textfeld1>,<textfeld2>,<RTF>,<cardholder>,<1234567890>

The Scheme Identifier must be transferred in batch format by using the TransactionID / schemeReferenceID, i.e. you will receive the schemeReferenceID / TransactionID for the initial 3DS-2.X transaction. This value must then be included in the batch format so that all recurring payments are correctly identified. The field cardHolder must be present but can also be submitted as an empty field and using schemeReferenceID the TransactionID field also empty.

Example:

CC,Sale,999,EUR,987456321,123456789,<VISA>,<0123456789123456>,<052029>,<mein Warenkorb>,,,M,,,<1234567890>

***Token/PCN Mandate - SchemeReferenceID**

You can try to submit payments without using a schemeReferenceID, but we cannot prevent card transactions to be declined.
In case of a rejection, we recommend to execute a separate transaction as Card Check (AccVerify=YES) transaction via our ...PaySSL.aspx, i.e. the customer receives a payment link which he can use to process an initial 3DS-2.X transaction and the schemeReferenceID is then reported back and can be included in the process of recurring payments again.

Scenario 10 – Extended Transaction Management (ETM)

- When using the Computop ETM, Computop Paygate takes care of the correct flagging of transactions for you.

Test 3-D Secure 2.0

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 Computop Paygate. 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 Computop MerchantID. If you are unsure whether it has already been activated, please contact Computop Helpdesk
- 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.
- Perform 3-D Secure Authentication
- 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