EMV® 3-D Secure

White Paper

Business Overview, Technical Features and Use Cases

Version 2.0

March 2025

Legal Notice

This document is subject to change by EMVCo at any time. This document does not create any binding obligations upon EMVCo or any third party regarding the subject matter of this document, which obligations will exist, if at all, only to the extent set forth in separate written agreements executed by EMVCo or such third parties. In the absence of such a written agreement, no product provider, test laboratory or any other third party should rely on this document, and EMVCo shall not be liable for any such reliance.

No product provider, test laboratory or other third party may refer to a product, service or facility as EMVCo approved, in form or in substance, nor otherwise state or imply that EMVCo (or any agent of EMVCo) has in whole or part approved a product provider, test laboratory or other third party or its products, services, or facilities, except to the extent and subject to the terms, conditions and restrictions expressly set forth in a written agreement with EMVCo, or in an approval letter, compliance certificate or similar document issued by EMVCo. All other references to EMVCo approval are strictly prohibited by EMVCo.

Under no circumstances should EMVCo approvals, when granted, be construed to imply any endorsement or warranty regarding the security, functionality, quality, or performance of any particular product or service, and no party shall state or imply anything to the contrary. EMVCo specifically disclaims any and all representations and warranties with respect to products that have received evaluations or approvals, and to the evaluation process generally, including, without limitation, any implied warranties of merchantability, fitness for purpose or non-infringement. All warranties, rights and remedies relating to products and services that have undergone evaluation by EMVCo are provided solely by the parties selling or otherwise providing such products or services, and not by EMVCo, and EMVCo will have no liability whatsoever in connection with such products and services.

This document is provided "AS IS" without warranties of any kind, and EMVCo neither assumes nor accepts any liability for any errors or omissions contained in this document. EMVCO DISCLAIMS ALL REPRESENTATIONS AND WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AS TO THIS DOCUMENT.

EMVCo makes no representations or warranties with respect to intellectual property rights of any third parties in or in relation to this document. EMVCo undertakes no responsibility to determine whether any implementation of this document may violate, infringe, or otherwise exercise the patent, copyright, trademark, trade secret, know-how, or other intellectual property rights of third parties, and thus any person who implements any part of this document should consult an intellectual property attorney before any such implementation.

Without limiting the foregoing, this document may provide for the use of public key encryption and other technology, which may be the subject matter of patents in several countries. Any party seeking to implement this document is solely responsible for determining whether its activities require a license to any such technology, including for patents on public key encryption technology. EMVCo shall not be liable under any theory for any party's infringement of any intellectual property rights in connection with this document.

Contents

Tables

1 Introduction

The purpose of this EMV® 3-D Secure White Paper (White Paper) is to promote a better understanding of certain EMV® 3-D Secure (3DS) features and provide example use cases that highlight their benefits.

The 3DS protocol is a security measure designed to provide an additional layer of protection for online transactions and to reduce the risk of fraud in e-commerce transactions.

This document describes the two primary 3DS flows: the Frictionless Flow (Section 2) and the Challenge Flow (Section 4). It also includes sections dedicated to specific transactions or technical features, such as Recurring and Instalment Transactions (Section 3), OOB authentication (Section 5), the 3DS Message Extensions (Section 6), and the Split-SDK (Section 7), along with example use cases. In addition, an overview and list of relevant 3DS documentation is provided in Section 8.

The use cases presented in this document are not exhaustive and other use cases may exist. This document does not describe the practical implementation of any specific use case, and such implementations may vary by 3DS Programme.

This EMV® 3-D Secure White Paper accompanies and complements the EMV® 3-D Secure Protocol and Core Functions Specification (Core Specification) and supporting documentation. Where additional relevant information on a given subject is available in any of these documents, references to specific sections in the documents are provided to avoid duplication.

This White Paper applies to the protocol versions for which testing support is available at the time of release of this document, and highlights the differences that may exist between those versions for certain features.

1.1 Audience and Structure

Structure

This White Paper is structured to provide a comprehensive understanding of the Core Specification, presenting its key features across different perspectives.

Each key feature section begins with a business overview, which outlines the functionalities, objectives and benefits, and the ways in which they enhance 3DS transactions, their security and user experience. This is followed by a flow diagram showing how the feature can be used during a 3DS transaction, along with related data. Lastly, for certain features, a sequence of screens illustrates the user experience.

Where applicable, the White Paper also explains how the key features can be leveraged depending on which Core Specification version is supported (version 2.2, version 2.2 in combination with the EMV® 3-D Secure Bridging Message Extension, or version 2.3.1).

Audience

The document is aimed at diverse groups with varying levels of expertise.

Firstly, it addresses stakeholders seeking a high-level overview of 3DS capabilities, presenting its business value and potential impact on improving user security and trust in online transactions.

It then provides information on the technical details of the Core Specification for experts involved in the development or implementation of 3DS applications.

Lastly, it is aimed at those interested in the user experience aspect and in learning how 3DS aims to streamline and enhance the authentication process for end users.

By addressing the needs of these distinct audiences, the document ensures comprehensive coverage of the Core Specification from different perspectives, and is intended for use by all participants of the 3DS ecosystem.

1.2 Notational Conventions

Abbreviations

For a list of abbreviations used in this White Paper, refer to Table 1.4 in Section 1.9 of the Core Specification.

Terminology and Conventions

The White Paper uses the following words which have a specific meaning:

3DS Requestor – Merchant in the context of a purchase transaction.

Assumptions

Where provided, assumptions for a given use case are specific to that use case example, but not the wider use case. Different assumptions are part of the same use case but would refer to a different use case example.

Preconditions

Preconditions for a given use case are those which must occur in order for the use case to exist.

2 Improving Risk Analysis and Frictionless Flow

2.1 Business Overview

The use of risk-based authentication allows Issuers to accept transactions without having to challenge Cardholders, which results in a frictionless process for both Cardholders and Merchants. The Merchants benefit from adopting the 3DS protocol by protecting against fraudulent chargebacks and ensuring that the Cardholders are secure from fraudulent transactions, while the Cardholders have a seamless experience using the Merchants' platform as they are not being challenged. This reduction in Challenge Flow interactions may lower the drop-off rate caused by using the 3DS protocol.

In the Frictionless Flow, the Issuer verifies the Cardholder's identity automatically without the need for additional authentication steps or Cardholder interaction.

The Frictionless Flow is achieved through real-time risk assessment that takes into account various types of information, such as:

- details of the transaction (amount, currency, Merchant, recurring or non-recurring…)
- Cardholder information
- device used by the Cardholder to perform the transaction
- the Cardholder's transaction history and relationship with the Merchant
- technical information such as device location or IP address

to determine the level of risk associated with the transaction.

To enhance transaction risk assessment, the Issuer can use two 3DS features:

- **Trust List** enables the Cardholder to create a list of preferred Merchants. Enabling the Cardholder to provide their spending habits improves the Issuer's risk assessment. For additional details, refer to *Trust List* in the Technical Features section.

- **Device Binding** enables the Cardholder to link the Device used for e-commerce transactions to their payment card (Cardholder Account Number). In return, the Issuer uses this information in transaction risk assessment as an indicator that the genuine Cardholder is performing the transaction using the same payment card on the same device. For additional details, refer to *Device Binding* in the Technical Features section.

The Frictionless Flow of the 3DS protocol provides a convenient and secure experience for online transactions. The automatic verification process based on real-time risk assessment helps to reduce the risk of fraud while keeping the transaction secure. The integration of the Frictionless Flow into the transaction provides a seamless shopping experience to both the Cardholder and the Merchant, making it an important aspect of online payment security.

2.2 Technical Features

2.2.1 Device Information for the App-Based Flow

In the Core Specification, Device Information is defined to allow rich device data collection on App-based transactions. The EMV® EMV 3-D Secure—SDK Device Information specification defines common and operating system-specific device elements to be captured by the 3DS SDK during an Authentication Request (AReq). This rich device data set assists in assessing transaction risk. In addition, it may provide information on the capabilities of the device to inform presentation of authentication methods during a challenge request.

Device Information is applicable to all App-based implementations, including the Default-SDK and the Split-SDK (and its variants). Some examples of device information may include:

- device manufacturer and model
- operating system and version
- geolocation data, including IP address and locale
- device IP and fingerprint
- telecom data (SIM metadata)
- user interface preferences and settings

A benefit of App-based authentication requests is that it enables the 3DS SDK to gather rich device data natively based on operating system-specific elements. The Device Information is a required parameter for 3DS processing (between the DS and ACS), but the specific data elements comprising the Device Information object are neither required nor optional. It is strongly recommended to populate all the fields accessible to the 3DS SDK (based on device and runtime permissions) to better inform network and Issuer processing systems.

Benefits by Actor

- Merchant – may increase frictionless authentication outcomes and/or authentication success rates by providing rich device information data
- Issuer – provides greater insight into the Cardholder's device details for risk assessment on the authentication request and informed device capabilities for authentication methods during a challenge request (if applicable)
- Cardholder – improves the authentication experience during checkout by potentially reducing friction or modifying challenge methods to align with device capabilities (for example, modifying screen resolution and fonts during a challenge prompt).

Technical Overview

The Device Information data is collected by the 3DS SDK using the built-in operating system APIs. The data is then encrypted by the 3DS SDK using the DS public key and sent to the 3DS Server for forwarding on the Authentication Request (AReq) as SDK Encrypted Data (sdkEncData). Once received by the DS, the SDK Encrypted Data will be decrypted and the

Device Information will be sent to the ACS for use in risk assessment and/or Cardholder verification.

The decrypted Device Information will include one of the following groups of data elements[1];

- Common Device Identification Parameters and the OS-specific Device Parameters,

  OR

- Platform Provider-specific Device Parameters (as determined by the 3DS SDK)

Within the OS-specific Device Parameters, there are varying degrees of data elements available, ranging from probabilistic device identifiers to user profile metadata on the device. These data elements are captured in JSON name/value pairs, using an Identifier name key:

- Common Device Identification Parameters (C001-C018)
- Android-specific Device Parameters (A001-A169)
- iOS-specific Device Parameters (I001-I015)
- Platform Provider-specific Device Parameters (D001-D035)

This format allows for flexibility in the capturing of the Device Information as the operating systems continually evolve. With operating systems typically releasing at least one major version per year, the Identifier name key and the format definition of the data elements in the Device Information specification can be adapted quickly when certain data elements are modified or restricted by the OS platform providers.

Versioning

The SDK Device Information specification has received multiple publications, each incrementing the version of the Device Information object (name/value pairs). All 3DS components are expected to support the latest Device Information version to maintain compatibility with each OS platform provider.

Security

The SDK Device Information specification enables secure transfer of data to the ACS by encrypting the device data in transit from the 3DS SDK using the DS public key. This prevents the 3DS Requestor or any unauthorised party from accessing or modifying device-level data during an Authentication Request.

The DS public key must be obtained from each supported network and the process is subject to network program rules (i.e., requires a valid EMVCo Letter of Approval). The connection between the 3DS SDK and the DS is required to be secured via the DS public key, even though the 3DS Server is facilitating the connectivity. This preserves the privacy of the Cardholder and upholds the security model established via the EMV 3DS protocol. Similar to the 3DS Method URL, all device information is accessed only by the ACS.

[1] If Platform Provider-specific Device parameters are used (for example, with a Split-SDK implementation), then the other Device Parameters (Common, Android-specific, iOS-specific) are not present.

Permissions

In line with regional regulations and/or operating system provider policy, the 3DS Requestor is required to provide prominent disclosure of the use of data in the application at the time of submission to the operating system provider. This includes specifying which data elements may be collected and used as part of a 3DS authentication.

At the time of installation, the 3DS Requestor App must prominently disclose to the user that:

- the 3DS Requestor App will access (and use) the sensitive user data (for example, phone number); and/or
- the 3DS Requestor App will provide access to the sensitive user data to the 3DS SDK (and ACS) for transaction risk assessment.

When use of sensitive user data is invoked for a 3DS authentication, the 3DS SDK must verify user consent and permission prior to requesting any sensitive user data from the 3DS Requestor App. This shall be implemented in accordance with the Permission designation provided in the SDK Device Information specification.

Note: The 3DS SDK shall never prompt for user consent or permission to any data within a 3DS authentication.

Supporting Documentation

- EMV 3-D Secure—SDK Specification
- Core Specification [Req 1–25]

Table 2.1 below lists the data elements that may be provided in relation to Device Information for the App-based flow.

Table 2.1: 3DS Data Elements Related to Device Information for the App-Based Flow

Data Element	Description	Version
Device Information	Device information gathered by the 3DS SDK from a Consumer Device. This is JSON name/value pairs that as a whole is Base64url-encoded. This will be populated by the DS as unencrypted data to the ACS obtained from SDK Encrypted Data.	2.3.1 2.2
SDK Encrypted Data	JWE Object (represented as a string) as defined in Section 6.2.2.1 containing data encrypted by the 3DS SDK for the DS to decrypt.	2.3.1 2.2

2.2.2 Device Information for the Browser-Based Flow

For each Browser-based transaction, the 3DS Server populates a set of Browser-specific data as follows:

- Browser Accept Headers
- Browser IP Address
- Browser Java Enabled
- Browser JavaScript Enabled
- Browser Language
- Browser Screen Color Depth
- Browser Screen Height
- Browser Screen Width
- Browser Time Zone
- Browser User-Agent

In addition to this data, the 3DS Method URL is a mechanism defined in the 3DS protocol that allows the ACS to gather detailed device and browser information during the checkout process, prior to authentication, enhancing risk-based decisioning during a 3DS authentication.

Executing the 3DS Method URL early in the checkout process maximises the likelihood that the Issuer has the necessary information for risk assessment before authentication, potentially improving authentication success rates and reducing customer abandonment. This involves loading a URL in a hidden iframe to execute JavaScript for browser fingerprinting, gathering data such as browser device characteristics.

Merchants must run the 3DS Method URL if requested by the Issuer, as failure to do so can result in higher step-up rates and increased authentication failures. The collected data helps Issuers to make informed risk decisions based on the device used by the Cardholder for the transaction and accurately assess the transaction contexts. The process is asynchronous and transparent to consumers, typically completing within 5 seconds. If the 3DS Method URL does not complete within 5 seconds, the process fails, but the Merchant can proceed with the Authentication Request (AReq) message.

Benefits by Actor

- Merchant – may increase frictionless authentication outcomes and/or increase authentication success rates by executing the 3DS Method URL and allowing it to complete
- Issuer – offers greater insight into the Cardholder's browser and device details for risk assessment on the authentication request, providing higher confidence in authentication result decisioning to mitigate fraud
- Cardholder – improves the authentication experience during checkout by potentially reducing friction

Technical Features

Refer to Section 5.8.1 in the Core Specification for the requirements related to the execution of the 3DS Method URL.

Table 2.2 below lists the data elements that may be provided in relation to Device Information for the Browser-based flow.

Table 2.2: 3DS Data Elements Related to Device Information for the Browser-Based Flow

Data Element	Description	Version
Browser Accept Headers	Exact content of the HTTP accept headers as sent to the 3DS Requestor from the Cardholder Browser.	2.3.1 2.2
Browser IP Address	IP address of the Browser as returned by the HTTP headers to the 3DS Requestor.	2.3.1 2.2
Browser Java Enabled	Boolean that represents the ability of the Cardholder Browser to execute Java.	2.3.1 2.2
Browser JavaScript Enabled	Boolean that represents the ability of the Cardholder Browser to execute JavaScript.	2.3.1 2.2
Browser Language	Value representing the Browser language as defined in IETF BCP47.	2.3.1 2.2
Browser Screen Color Depth	Value representing the bit depth of the colour palette for displaying images, in bits per pixel.	2.3.1 2.2
Browser Screen Height	Total height of the Cardholder's screen in pixels.	2.3.1 2.2
Browser Screen Width	Total width of the Cardholder's screen in pixels.	2.3.1 2.2
Browser Time Zone	Time zone offset in minutes between UTC and the Cardholder Browser local time.	2.3.1 2.2
Browser User-Agent	Exact content of the HTTP user-agent header.	2.3.1 2.2
3DS Method Completion Indicator	Indicates whether the 3DS Method successfully completed.	2.3.1 2.2
3DS Method ID	Contains the 3DS Server Transaction ID used during the previous execution of the 3DS Method.	2.3.1 2.2
Card Range Data	Card range data from the DS indicating the most recent Protocol Versions supported by the ACS, and, optionally, the DS that hosts that range, and, if configured, the ACS URL for the 3DS Method.	2.3.1 2.2

Data Element	Description	Version
3DS Method Notification URL	The URL that will receive the notification of 3DS Method completion from the ACS. This is sent in the initial request to the ACS from the 3DS Requestor executing the 3DS Method.	2.3.1 2.2
ACS Protocol Versions	Array of objects containing the list of Protocol Versions supported by the ACS for the card range, with their associated ACS Information Indicator, the 3DS Method URL and the list of Supported Message Extension. • Version • ACS Information Indicator • 3DS Method URL • Supported Message Extension	2.3.1 2.2
3DS Method URL	The ACS URL that will be used by the 3DS Method for a particular Protocol Version. The 3DS Method URL data element may be omitted if not supported by the ACS for this specific card range.	2.3.1 2.2

2.2.3 Cardholder Information

The importance of Cardholder Information in 3DS Authentication is critical for enhancing transaction security and ensuring successful ACS processing. Cardholder information comprises various data types sourced from multiple origins. Key data points include:

- the Cardholder's intrinsic information, such as name, address, and email;
- the Cardholder’s relationship with the Merchant, including account length and history of previous authenticated transactions; and
- the authentication methods employed by the Merchant to log in the customer, ranging from basic username and password to federated login and FIDO-based authentication.

This information is essential for establishing trust and assessing risk. The richer data exchange in 3DS authentication enables businesses to send detailed, transaction-specific, and contextual data to the ACS. The Merchant or 3DS Server should provide all available information as accurately as possible. When the ACS trusts and validates this data, the transaction can proceed seamlessly; otherwise, it may enter a challenge flow for additional verification.

Benefits by Actor

- Merchant: may increase frictionless authentication outcomes
- Issuer: offers greater insight into the Cardholder information for risk assessment on the authentication request
- Cardholder: improves the authentication experience during checkout by potentially reducing friction

Technical Features

The Merchant provides the 3DS Server with two main sets of data related to the Cardholder:

- Cardholder information – data collected by the Merchant during the transaction to facilitate its completion and product or service delivery; and
- Cardholder relationship with the Merchant – information held by the Merchant that describes previous Cardholder activities or interactions.

The Merchant should provide the most accurate and complete information possible, as this directly impacts the ACS's ability to authenticate the transaction. Incomplete or incorrect information (such as dummy values) that does not accurately represent the Cardholder or the context of the transaction can lead to increased challenges or, in the worst case, declined authentication.

Table 2.3 below lists the data elements that may be provided in relation to Cardholder information.

Table 2.3: 3DS Data Elements Related to Cardholder Information

Data Element	Description	Version
Address Match Indicator	Indicates whether the Cardholder Shipping Address and Cardholder Billing Address are the same.	2.3.1 2.2
Cardholder Account Identifier	Additional information about the account optionally provided by the 3DS Requestor.	2.3.1 2.2
Cardholder Billing Address City	The city of the Cardholder billing address associated with the card used for this purchase.	2.3.1 2.2
Cardholder Billing Address Country	The country of the Cardholder billing address associated with the card used for this purchase.	2.3.1 2.2
Cardholder Billing Address Line 1	First line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.	2.3.1 2.2
Cardholder Billing Address Line 2	Second line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.	2.3.1 2.2

Data Element	Description	Version
Cardholder Billing Address Line 3	Third line of the street address or equivalent local portion of the Cardholder billing address associated with the card used for this purchase.	2.3.1 2.2
Cardholder Billing Address Postal Code	ZIP or other postal code of the Cardholder billing address associated with the card used for this purchase.	2.3.1 2.2
Cardholder Billing Address State	The state or province of the Cardholder billing address associated with the card used for this purchase.	2.3.1 2.2
Cardholder Email Address	The email address associated with the account that is either entered by the Cardholder or is on file with the 3DS Requestor.	2.3.1 2.2
Cardholder Home Phone Number	The home phone number provided by the Cardholder.	2.3.1 2.2
Cardholder Mobile Phone Number	The mobile phone number provided by the Cardholder.	2.3.1 2.2
Cardholder Name	Name of the Cardholder.	2.3.1 2.2
Cardholder Shipping Address City	City portion of the shipping address requested by the Cardholder.	2.3.1 2.2
Cardholder Shipping Address Country	Country of the shipping address requested by the Cardholder.	2.3.1 2.2
Cardholder Shipping Address Line 1	First line of the street address or equivalent local portion of the shipping address requested by the Cardholder.	2.3.1 2.2
Cardholder Shipping Address Line 2	The second line of the street address or equivalent local portion of the shipping address requested by the Cardholder.	2.3.1 2.2
Cardholder Shipping Address Line 3	The third line of the street address or equivalent local portion of the shipping address requested by the Cardholder.	2.3.1 2.2

Data Element	Description	Version
Cardholder Shipping Address Postal Code	The ZIP or other postal code of the shipping address requested by the Cardholder.	2.3.1 2.2
Cardholder Shipping Address State	The state or province of the shipping address associated with the card being used for this purchase.	2.3.1 2.2
Cardholder Work Phone Number	The work phone number provided by the Cardholder	2.3.1 2.2
Tax ID	Cardholder's tax identification.	2.3.1 2.2

Note: The Address Match Indicator allows the 3DS Requestor to indicate to the ACS whether the Cardholder's billing and shipping address are the same.

3DS Requestors can use the Address Match Indicator to identify that the Cardholder selected a checkbox indicating that the shipping address is the same as the billing address. This could be helpful in regions with privacy mandates that prohibit providing billing and shipping address details.

3DS Requestors should still provide billing and shipping address information (assuming no privacy mandates exist in the region), even when the Address Match Indicator has been provided.

Some 3DS Requestors always provide this indicator as part of their checkout process, even if it may not be meaningful – for example, in the case of digital goods for which the shipping address is irrelevant.

Table 2.4 below lists the data elements that may be provided in relation to the Cardholder's relationship with the Merchant.

Table 2.4: 3DS Data Elements Related to the Cardholder's Relationship with the Merchant

Data Element	Description	Version
Cardholder Account Information	Additional information about the Cardholder's account provided by the 3DS Requestor.	2.3.1 2.2

Data Element	Description	Version
3DS Requestor Authentication Information	Information about how the 3DS Requestor authenticated the Cardholder before or during the transaction.	2.3.1 2.2
3DS Requestor Prior Transaction Authentication	Information about how the 3DS Requestor authenticated the Cardholder as part of a previous 3DS transaction.	2.3.1 2.2

2.2.4 Trust List Managed by the ACS / Issuer – Overview

The Issuer/ACS offers the Cardholder the option to add their preferred or trusted Merchant to their trust list during a 3DS challenge when in direct communication with the Cardholder. The Issuer controls the selection of Merchants proposed in the Trust List (for example, only offering the Trust List service for low-risk Merchants). The Issuer will consider the risk associated with the Merchant type and market, as well as the Cardholder's transaction history.

The 3DS Trust List feature may be used for the trusted beneficiary exemption in countries in scope of PSD2 (Revised Payment Services Directive).

The Core Specification does not prevent Issuers from providing alternative channels to Cardholders to manage the trusted beneficiaries list (for example, e-banking).

An alternative use case is the Trust List managed by the DS as described in Section 2.2.6.

Benefits by Actor

- Merchant
  - fewer challenges
  - faster transactions
- Issuer
  - better knowledge of Cardholder purchasing habits
  - opportunity to pre-select "trusted" Merchants
  - reduced need to challenge
- Cardholder
  - fewer challenges
  - faster transactions

2.2.5 Trust List Flow and Data

Preconditions

The ACS has a Trust List Management System and can display the Trust List prompt/screen to the Cardholder during a 3DS challenge.

Optional: The ACS indicates support of the Trust List in the Card Range Data (ACS Information Indicator - 04 = Trust List Supported).

Note: The ACS uses some or all of the Merchant information (Merchant Name, 3DS Requestor Name, 3DS Requestor ID) to manage the Trust List. Therefore, it is essential that the Merchant and/or the 3DS Server provide consistent Merchant information across the Trust List enrolment and subsequent transactions.

Sequence Diagram

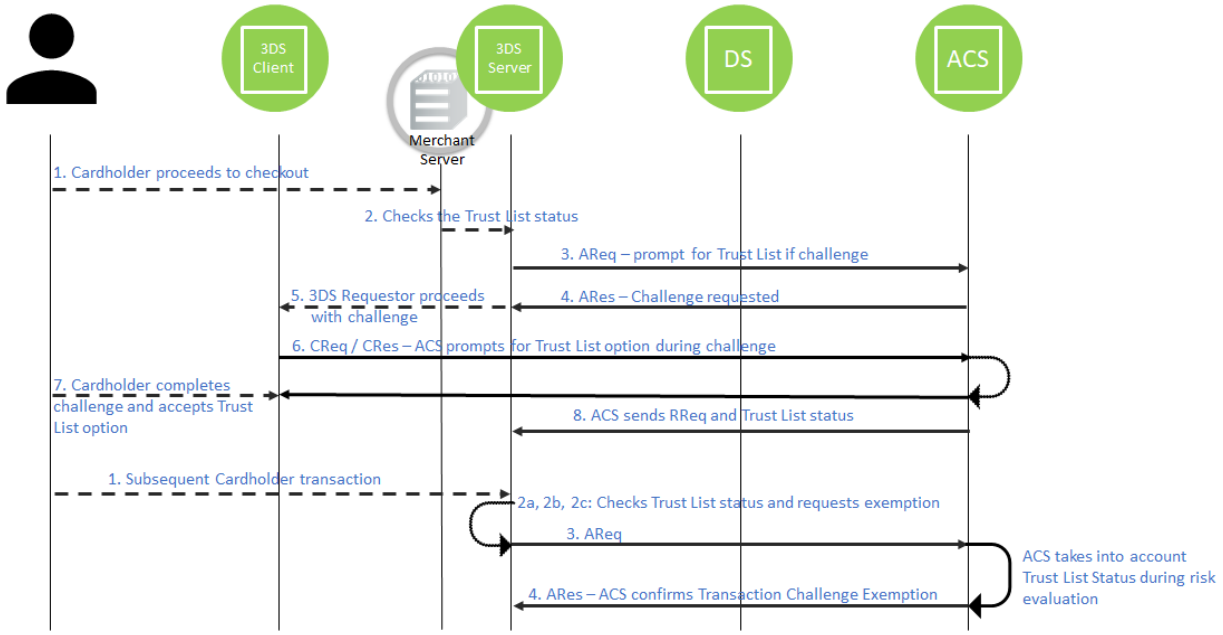
The Cardholder enrols a Merchant on their Trust List that is managed by the Issuer/ACS.

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Requestor/3DS Server may:
   a. Check if the ACS supports the Trust List by confirming that ACS Information Indicator = 04 (Trust List Supported)
   b. Set the 3DS Requestor Challenge Indicator to 09 (= Challenge requested – Trust List prompt requested if challenge required) in the Authentication Request (AReq) message to indicate to the ACS that it should prompt for the Trust List during the challenge.
3. The 3DS Server sends the AReq message.
4. The ACS responds with an Authentication Response (ARes) message requesting a challenge.
5. The 3DS Server proceeds with the challenge.
6. The ACS proceeds with the challenge and provides the prompt for the Trust List option.
7. The Cardholder completes the challenge and accepts the Trust List option (enrols the Merchant on the Trust List).
8. The ACS provides the outcome of the authentication in a Results Request (RReq) message, and optionally the Trust List Status using the Trust List Status and the Trust List Status Source.

In a subsequent transaction with the same Cardholder and Merchant:

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Requestor/3DS Server may:
   a. Check the Trust List Status of the Cardholder.
   b. Check if the ACS supports the Trust List exemption by confirming that ACS Information Indicator = 09 (Trust List Exemption Supported)
   c. Set the 3DS Requestor Challenge Indicator to 08 (= No challenge requested – use Trust List exemption if no challenge required) in the AReq message.
3. The 3DS Server sends the AReq message.
4. As a result of the risk assessment, the ACS may apply the Trust List exemption, and may report it in the Transaction Challenge Exemption (= 08) in an ARes message.

Figure 2.1: Trust List Flow



Note: Step 6. CReq/CRes: refer to Trust List templates for the user interface.

Table 2.5 below lists the data elements that may be provided in relation to the Trust List.

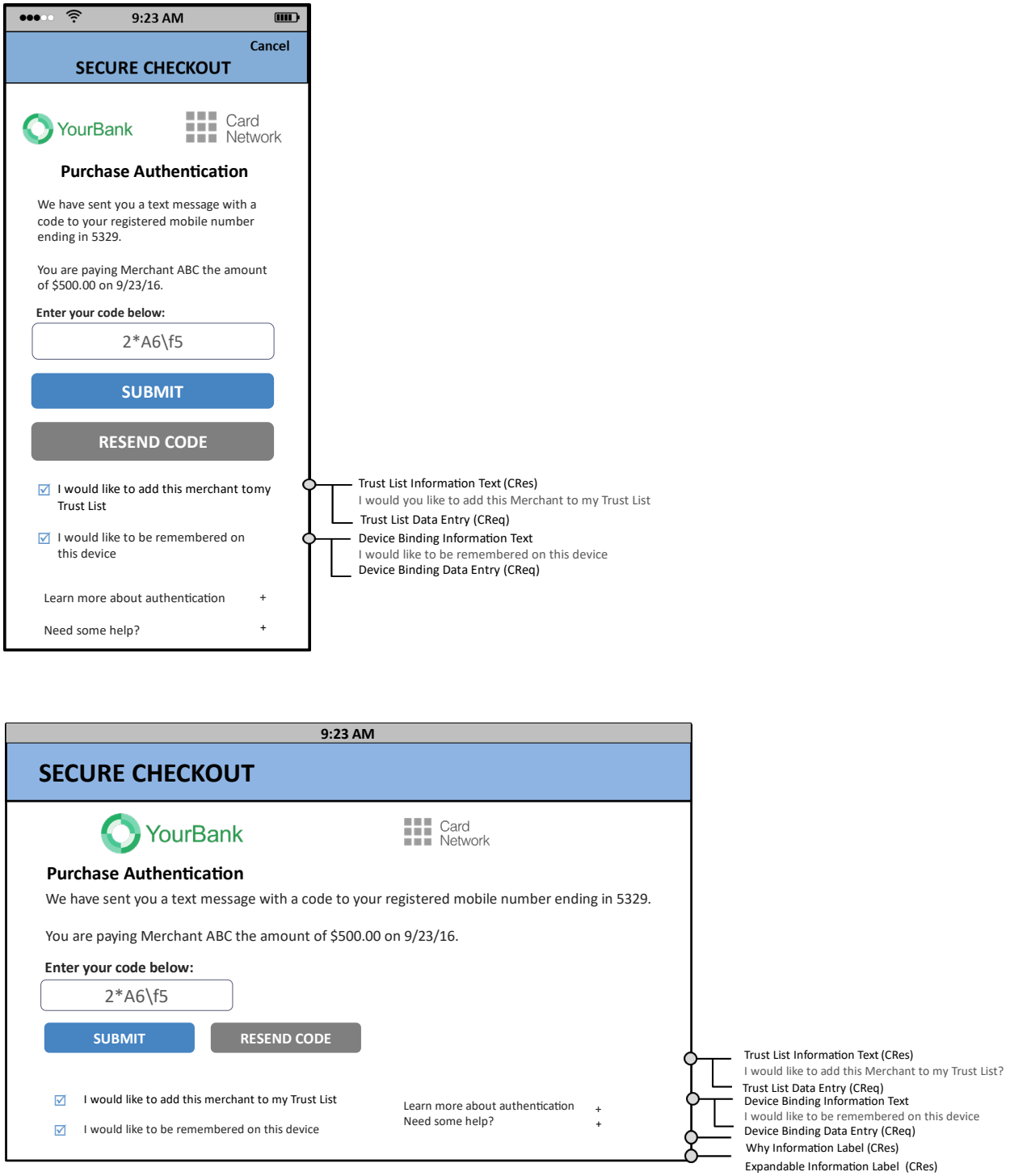
Table 2.5: 3DS Data Elements Related to the Trust List

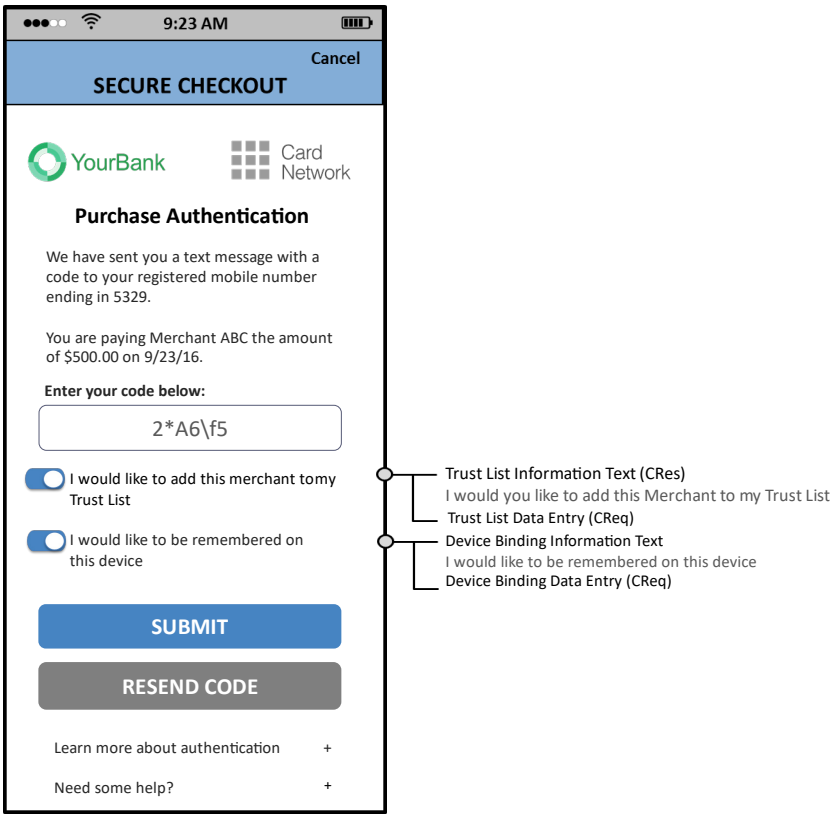
Data Element	Description	Version
3DS Requestor Challenge Indicator	Indicates whether a challenge is requested for this transaction.	2.3.1 2.2
3RI Indicator	Indicates the type of 3RI request. This data element provides additional information to the ACS to determine the best approach for handling a 3RI request. A value of 10 indicates a Trust List Status check.	2.3.1 2.2
ACS Information Indicator	Provides additional information for a particular Protocol Version to the 3DS Server. The element lists all applicable values for the card range.	2.3.1 2.2

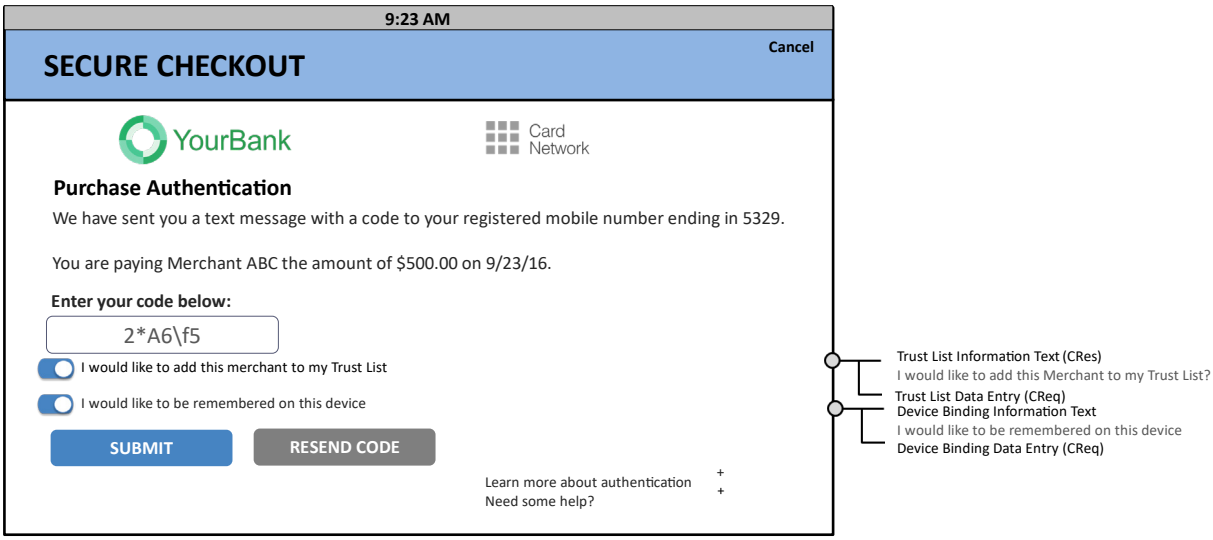
Data Element	Description	Version
Card Range Data	Card range data from the DS indicating the most recent Protocol Versions supported by the ACS, and, optionally, the DS that hosts that range, and, if configured, the ACS URL for the 3DS Method. Additionally, it identifies the 3DS features supported by the ACS in the ACS Information Indicator, such as Trust List or Decoupled Authentication. Trust List indicators are defined in the ACS Information Indicator: - 04 = Trust List Supported for v2.2 and v2.3 - 09 = Trust List Exemption Supported for v2.3	2.3.1 2.2
Toggle Position Indicator	Indicates if the Trust List and/or Device Binding prompt should be presented below or above the action buttons.	2.3.1
Transaction Challenge Exemption	Exemption applied by the ACS to authenticate the transaction without requesting a challenge.	2.3.1 2.2 + Bridging Message Extension
Trust List Data Entry	Indicator provided by the 3DS SDK to the ACS to confirm whether the Cardholder gives consent to the Trust List.	2.3.1 2.2
Trust List Information Text	Text provided by the ACS to the Cardholder during a Trust List transaction.	2.3.1 2.2
Trust List Status	Enables the communication of Trust List Status between the ACS, the DS and the 3DS Requestor.	2.3.1 2.2
Trust List Status Source	This data element will be populated by the system setting Trust List Status.	2.3.1 2.2

Note: The term “Trust List” is used in version 2.3.1 of the Core Specification, replacing the terms “Whitelist” and “Whitelisting” used in version 2.2.

Figure 2.2: App Flow – User Interface Related to the Trust List







Note: Checkbox, radio button or any relevant user interface may be used to offer the Trust List and Device Binding options.

2.2.6 Alternative Use Case – Trust List Managed by the DS

Preconditions

The DS has a Trust List Management System and an agreement with the ACS to manage the Trust List on its behalf.

The ACS is able to display the Trust List prompt/screen to the Cardholder during the 3DS challenge.

Optional: The ACS or DS indicates support of the Trust List in the Card Range Data (ACS Information Indicator – 04 = Trust List Supported)

Sequence Diagram

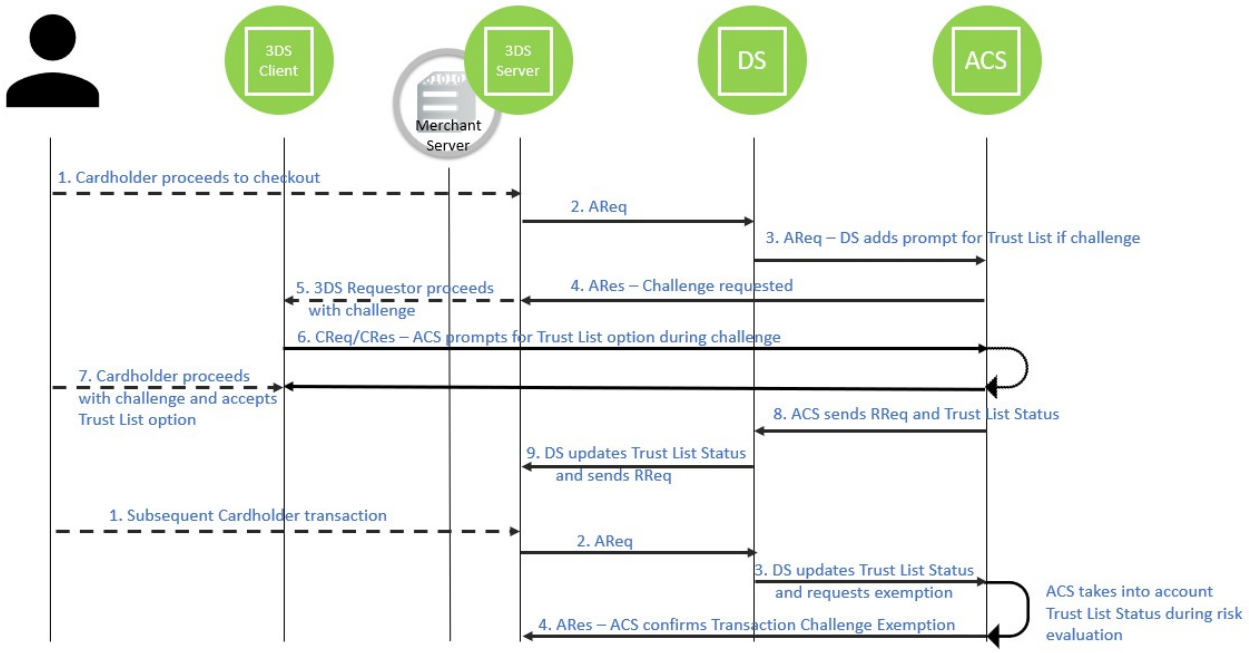
The Cardholder enrols a Merchant on their Trust List that is managed by the DS.

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Server sends an AReq message.
3. The DS sets the 3DS Requestor Challenge Indicator to 09 (= Challenge requested – Trust List prompt requested if challenge required) to indicate to the ACS that it should prompt for the Trust List during the challenge.
4. The ACS responds with an ARes message requesting a challenge.
5. The 3DS Server proceeds with the challenge.
6. The ACS proceeds with the challenge and provides the prompt for the Trust List option.
7. The Cardholder accepts the Trust List option and completes the challenge.
8. The ACS provides the outcome of the authentication in the RReq message, and the Trust List Status to the DS using the Trust List Status and Trust List Status Source.
9. The DS updates the Trust List Status for this Cardholder account in its Trust List management system, and optionally provides the feedback to the 3DS Server using the Trust List Status and the Trust List Status Source in the RReq message.

In a subsequent transaction with the same Cardholder and Merchant:

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Server sends an AReq message.
3. The DS updates the AReq message from the 3DS Server with the Trust List Status and Trust List Status Source, and sets the 3DS Requestor Challenge Indicator to 08 (= No challenge requested – use Trust List exemption if no challenge required).
4. As a result of the risk assessment, the ACS may apply the Trust List exemption, and may report it in the Transaction Challenge Exemption (= 08) in an ARes message.

Figure 2.3: Trust List Managed by the DS



Note: Step 6. CReq/CRes: refer to Trust List templates for the user interface.

2.2.7 Device Binding managed by the ACS / Issuer – Overview

In this White Paper, Device Binding is understood to denote the process to link the Consumer Device used for a transaction to the Cardholder Account.

Device Binding may be managed by any 3DS component (refer to the different use cases in this section).

The ACS, the DS or the 3DS Server may be the source of the Device Binding Status information.

Benefits by Actor

- Merchant – reduced need to challenge for returning Cardholders
- Issuer
  - better knowledge of Cardholder purchasing habits
  - reduced need to challenge
- Cardholder – feels more secure as transactions not performed on the device are more likely to be challenged

Use Case Overview

The ACS offers the Cardholder the option to link the device used for the transaction to the Cardholder Account Number during a 3DS challenge. The Device Binding Status provides to the ACS additional information that could be used for transaction risk assessment.

The Core Specification does not prevent Issuers from providing alternative channels to Cardholders to manage the Device Binding information (for example, online banking).

The Core Specification does not define how the ACS identifies the Consumer Device.

Preconditions

The ACS has a Device Binding management system and is able to display the Device Binding prompt/screen to the Cardholder during the 3DS challenge.

The ACS is able to identify the Consumer Device.

Note: How the ACS identifies the Consumer Device is outside the scope of the Core Specification.

Sequence Diagram

The Cardholder accepts the Device Binding option that is managed by the ACS.

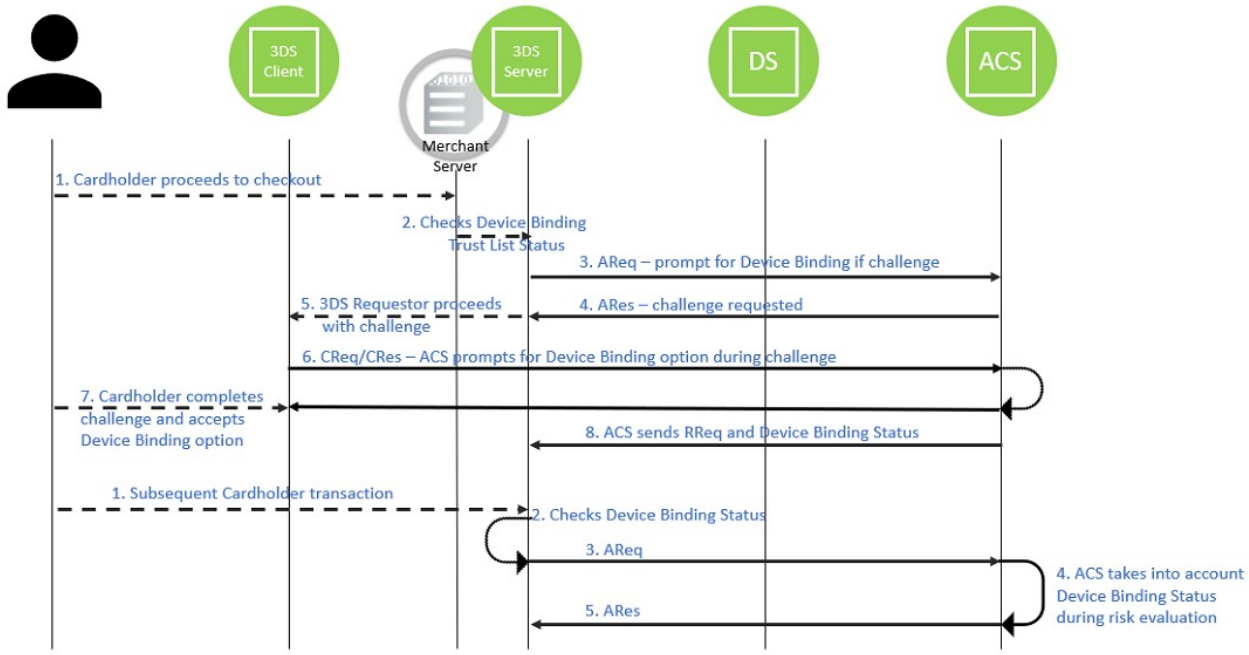
1. The Cardholder makes a purchase and proceeds to checkout.
2. Optionally, the 3DS Requestor/3DS Server:
   a. Checks if the ACS supports Device Binding by confirming that ACS Information Indicator = 05 (Device Binding Supported).
   b. Sets the 3DS Requestor Challenge Indicator to 12 (= Challenge requested – Device Binding prompt requested if challenge required) in the AReq message to indicate to the ACS that it should prompt for Device Binding during the challenge.
   c. Initiates a 3DS authentication.
3. The 3DS Server sends the AReq message.
4. The ACS responds with an ARes message requesting a challenge.
5. The 3DS Server proceeds with the challenge (opens an iframe for a Browser flow or provides the relevant data to the 3DS SDK for an App flow).
6. The ACS proceeds with the challenge and provides the prompt for the Device Binding option.
7. The Cardholder completes the challenge and accepts the Device Binding option.
8. The ACS stores the Device Binding information for this Cardholder/Account, provides the outcome of the authentication in the RReq message, and optionally the Device Binding Status using the Device Binding Status and the Device Binding Status Source to the 3DS Server/3DS Requestor.

In a subsequent transaction with the same Cardholder and Merchant:

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Requestor/3DS Server prepares the AReq message by:
   a. Checking the Device Binding Status of the Cardholder, and/or
   b. Providing the Device Binding Status and Device Binding Status Source (if known).
3. The 3DS Server sends the AReq message.

4. The ACS uses the Device Binding Status information as part of transaction risk assessment.
5. The ACS returns an ARes message.

Figure 2.4: Device Binding Flow



Note: Step 6. CReq/CRes: refer to Device Binding templates for the user interface.

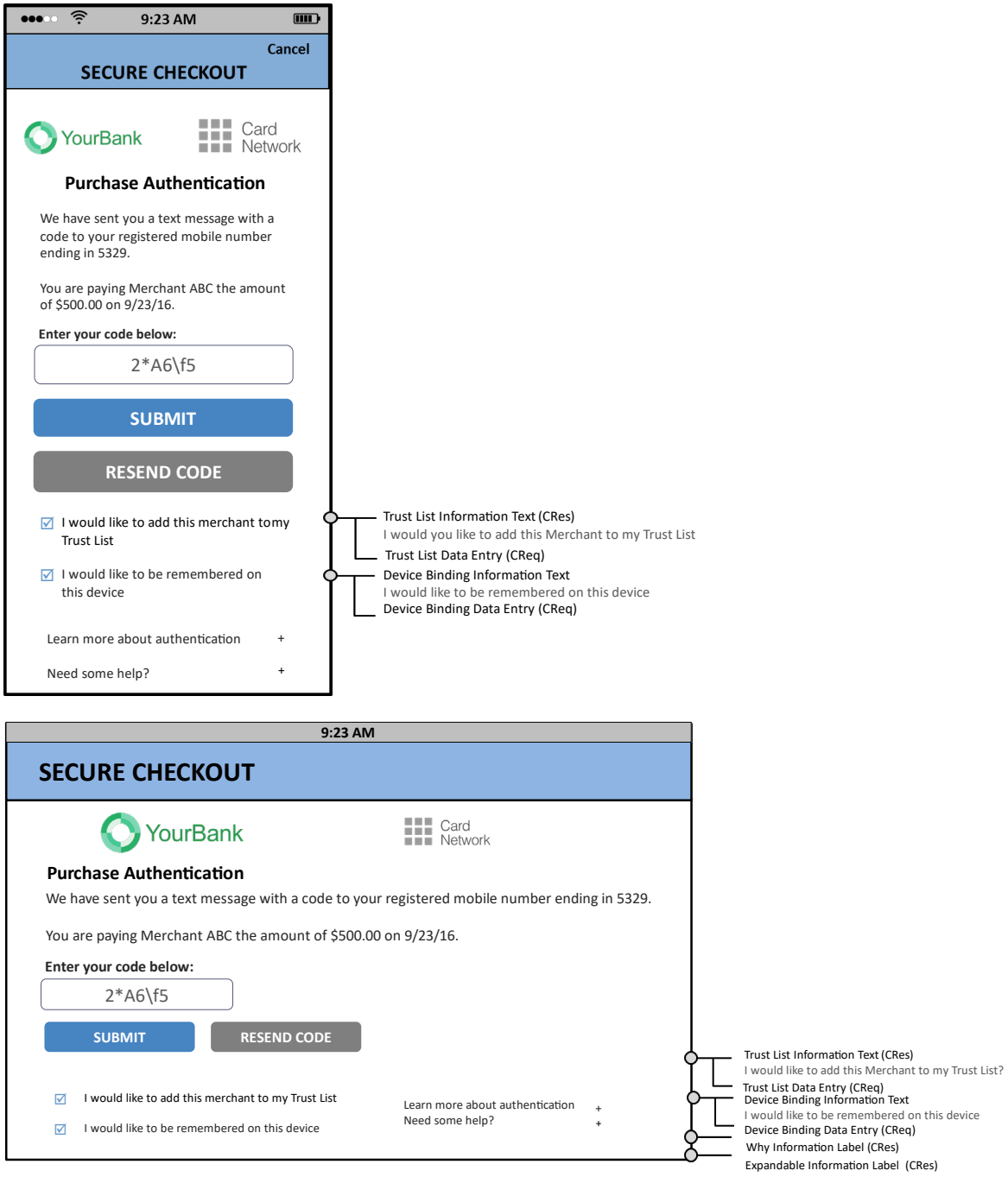
Table 2.6 below lists the data elements that may be provided in relation to Device Binding.

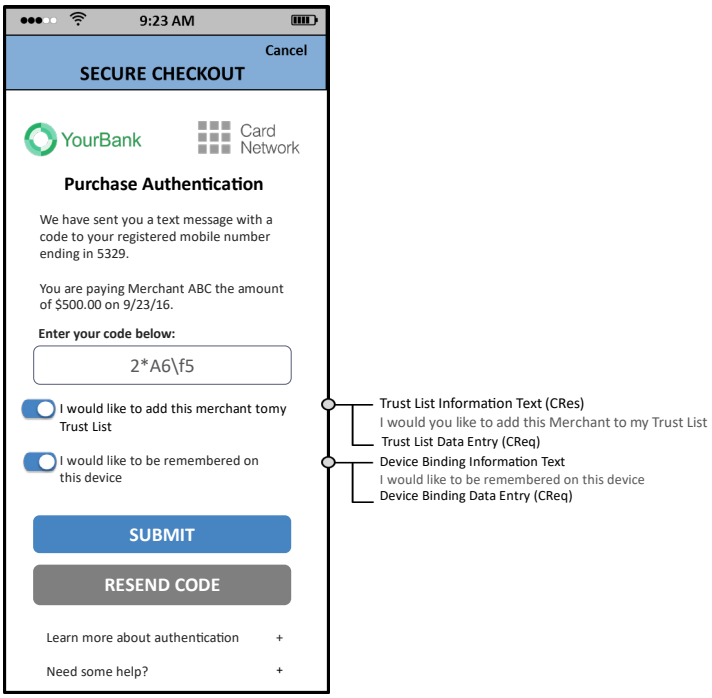
Table 2.6: 3DS Data Elements Related to Device Binding

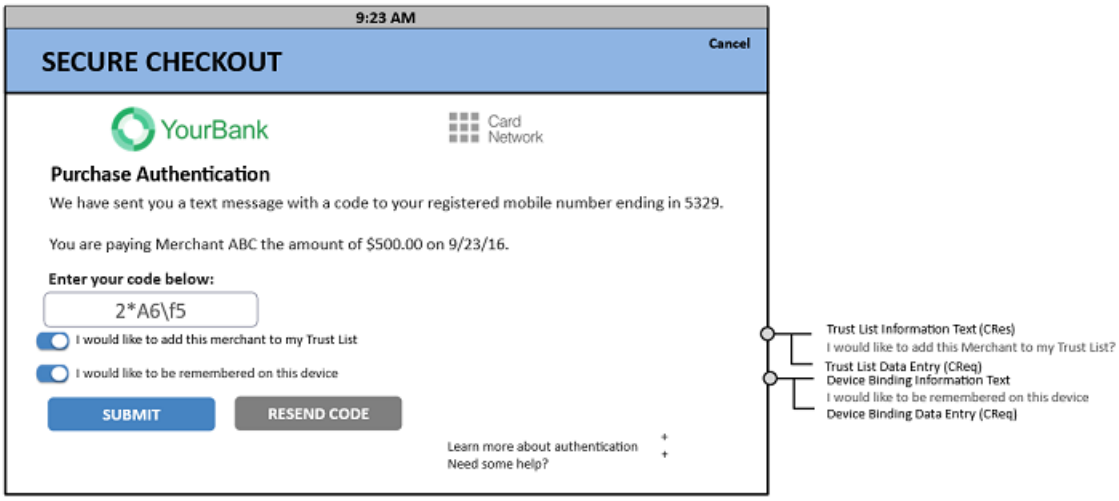
Data Element	Description	Version
3DS Requestor Challenge Indicator	Indicates whether a challenge is requested for this transaction.	2.3.1
3RI Indicator	Indicates the type of 3RI request. This data element provides additional information to the ACS to determine the best approach for handling a 3RI request. A value of 10 indicates a Trust List Status check.	2.3.1
ACS Information Indicator	Provides additional information for a particular Protocol Version to the 3DS Server. The element lists all applicable values for the card range.	2.3.1

Data Element	Description	Version
Card Range Data	Card range data from the DS indicating the most recent Protocol Versions supported by the ACS, and, optionally, the DS that hosts that range, and, if configured, the ACS URL for the 3DS Method. Additionally, it identifies the 3DS features supported by the ACS, such as Trust List or Decoupled Authentication. The Device Binding indicator is defined in the ACS Information Indicator: - 05 = Device Binding Supported	2.3.1
Device Binding Data Entry	Indicator provided by the 3DS SDK to the ACS to confirm whether the Cardholder gives consent to bind the device.	2.3.1
Device Binding Information Text	Text provided by the ACS to the Cardholder during the Device Binding process.	2.3.1
Device Binding Status	Enables the communication of Device Binding Status between the ACS, the DS and the 3DS Requestor. For bound devices (value = 11–14), Device Binding Status also conveys the type of binding that was performed.	2.3.1
Device Binding Status Source	This data element will be populated by the system setting Device Binding Status.	2.3.1
Toggle Position Indicator	Indicates if the Trust List and/or Device Binding prompt should be presented below or above the action buttons.	2.3.1

Figure 2.5: User Interface Related to Device Binding







Note: Checkbox, radio button or any relevant user interface may be used to offer the Trust List and Device Binding options.

Alternative Use Case – Device Binding Managed by the 3DS Server/3DS Requestor

Preconditions

The 3DS Server (and/or 3DS Requestor) has a Device Binding management system.

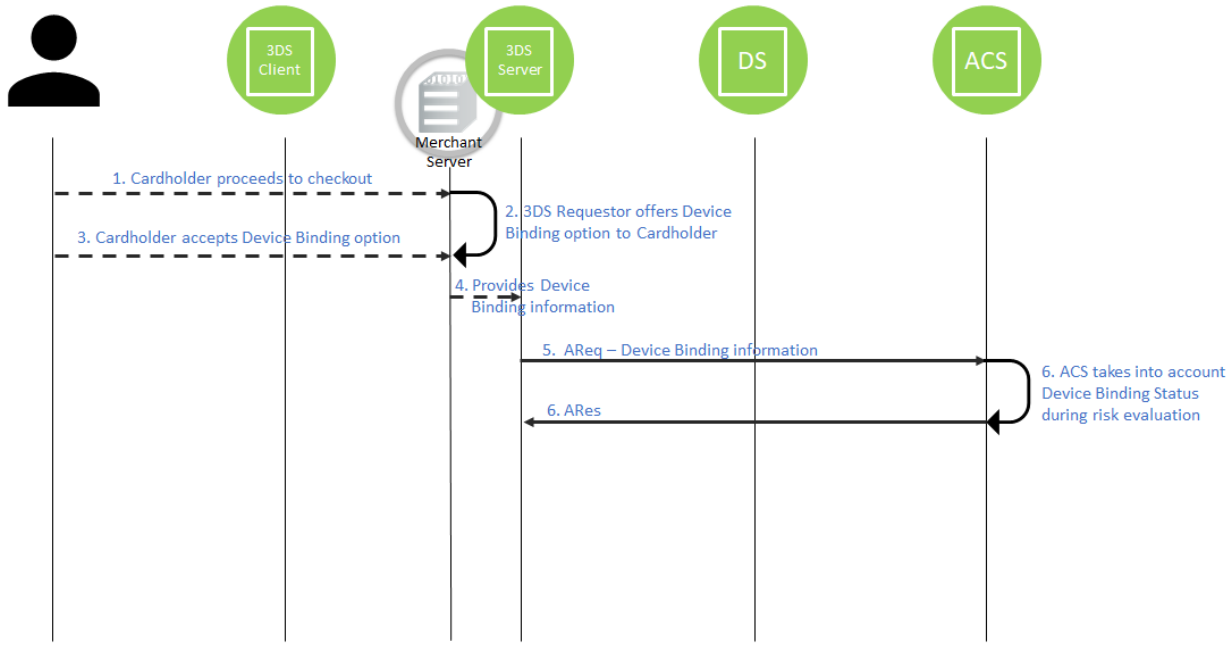
The 3DS Server (and/or 3DS Requestor) is able to identify the Device used by the Cardholder.

Sequence Diagram

The Cardholder accepts the Device Binding option that is managed by the 3DS Server/3DS Requestor.

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Requestor offers the Cardholder the option to link their Cardholder Account Number with the Device used for this transaction during the checkout process.
3. The Cardholder accepts the Device Binding option.
4. The 3DS Requestor provides the transaction information to the 3DS Server with the Device Binding information.
5. The 3DS Server sends an AReq message and the Device Binding information, using the Device Binding Status and Device Binding Status Source.
6. The ACS uses the Device Binding Status information as part of transaction risk assessment.
7. The ACS returns an ARes message.

Figure 2.6: Device Binding Managed by the 3DS Server/3DS Requestor



Alternative Use Case – Device Binding Managed by the DS

Preconditions

The DS has a Device Binding management system, and is able to identify the Consumer Device.

Note: How the DS identifies the Consumer Device is outside the scope of the Core Specification.

The DS and ACS have an agreement for the management of the Device Binding information.

The ACS is able to display the Device Binding prompt/screen to the Cardholder during the 3DS challenge.

Sequence Diagram

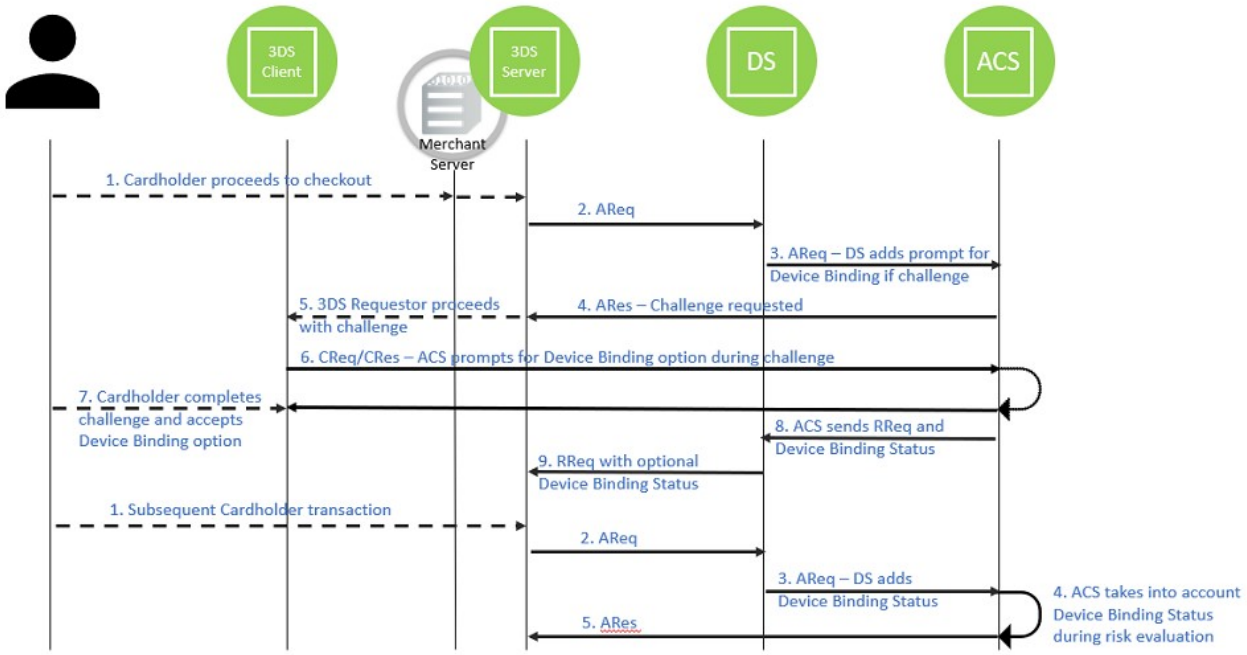
The Cardholder accepts the Device Binding option that is managed by the DS.

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Server sends an AReq message.
3. The DS receives the AReq message, sets the 3DS Requestor Challenge Indicator to 12 (= Challenge requested – Device Binding prompt requested if challenge required) to indicate to the ACS that it should prompt for Device Binding during the challenge, and sends the AReq message to the ACS.
4. The ACS responds with an ARes message requesting a challenge.
5. The 3DS Server proceeds with the challenge (opens an iframe for a Browser flow or provides the relevant data to the 3DS SDK for an App flow).
6. The ACS proceeds with the challenge and provides the prompt for the Device Binding option.
7. The Cardholder accepts the Device Binding option and completes the challenge.
8. The ACS provides the Device Binding information for this Cardholder/Account, the outcome of the authentication in the RReq message to the DS.
9. The DS optionally provides the Device Binding status in the RReq message to the 3DS Server.

In a subsequent transaction with the same Cardholder and Merchant:

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Requestor/3DS Server prepares the AReq message and sends it to the DS.
3. The DS updates the AReq message with the Device Binding Status and Device Binding Status Source.
4. The ACS uses the Device Binding Status information as part of transaction risk assessment.
5. The ACS returns an ARes message.

Figure 2.7: Device Binding Managed by the DS



2.2.8 Delegated Authentication

Delegated authentication enhances the shopping experience for Cardholders, Merchants, and Issuers, ensuring a seamless Cardholder authentication process. EMV 3DS is designed to encourage frictionless authentication, which can be achieved by Merchants providing specific information. Delegated authentication can increase the likelihood of a frictionless authentication.

In delegated authentication, Issuers transfer the responsibility of Cardholder authentication to a third party, which may be the Merchant or an authorised representative. This arrangement allows the third party to handle the authentication process.

Upon authenticating the Cardholder, the Merchant relays the authentication confirmation to the ACS. The ACS then assesses the Cardholder's authentication details as part of its transaction risk analysis. It will confirm the authentication unless the risk is deemed excessive, or the information provided is insufficient. In such instances, the ACS retains the option to challenge the Cardholder.

Furthermore, the regulations (such as PSD2) support delegated authentication by Merchants or third parties, provided that the Merchant adheres to the Strong Customer Authentication (SCA) requirements. This regulatory framework ensures that the authentication process remains secure while allowing flexibility in how it is conducted. For this to work effectively, Merchants and Issuers must establish agreements regarding the authentication methods, either through bilateral contracts or services offered by payment systems, which are outside the scope of the Core Specification.

Benefits by Actor

- Merchant: increased frictionless authentication outcomes, reduction of the transactions lost due to Cardholder abandonment or 3DS authentication failure during the 3DS process
- Issuer: reduced need to request a challenge
- Cardholder: improvements to the authentication experience during checkout by potentially reducing friction

Technical Features

Preconditions

The Merchant has an agreement with the Issuer/ACS for delegated authentication, for example as defined in PSD2.

The Merchant has an authentication process that meets the Issuer's requirements or preferences.

Overview

When the Cardholder creates an account or makes a purchase with the Merchant, the Cardholder goes through a secure registration process that complies with agreed ACS requirements (such as a FIDO-based authentication).

During the checkout process, the Cardholder is prompted to authenticate using the method established during the registration. The Merchant's system then verifies the authentication data to validate the Cardholder.

After the successful Cardholder authentication, the Merchant communicates this information using the 3DS Requestor Authentication Method and 3DS Requestor Authentication Data in the AReq message to the ACS.

The ACS receives the confirmation of the Cardholder authentication, it can approve the transaction, allowing the transaction to be processed. This streamlined process reduces the need for additional authentication steps providing a frictionless checkout experience for the Cardholder.

Table 2.7 below lists the data elements that may be provided in relation to 3DS Requestor Authentication Information.

Table 2.7: 3DS Data Elements Related to 3DS Requestor Authentication Information

Data Element	Description	Version
3DS Requestor Authentication Data	Data that documents and supports a specific authentication process. In the current version of the specification, this data element is not defined in detail. However, the intention is that, for each 3DS Requestor Authentication Method, this field carry data that the ACS can use to verify the authentication process. For example, if the 3DS Requestor Authentication Method is: • 03, then this element can carry information about the provider of the federated ID and related information. • 06, then this element can carry the FIDO Assertion and/or Attestation Data. • 07, then this element can carry FIDO Assertion and/or Attestation. Data with the FIDO Assurance Data signed by a trusted third party. • 08, then this element can carry the SRC Assurance Data. For 3DS Requestor Authentication Method = 06 or 07, refer to the *EMV® 3-D Secure White Paper – Use of FIDO® Data in 3-D Secure Messages* for the 3DS Requestor Authentication Data content and format.	2.3.1 2.2
3DS Requestor Authentication Method	Length: 2 characters JSON Data Type: String Values accepted: • 01 = No 3DS Requestor authentication occurred (i.e., Cardholder “logged in” as guest) • 02 = Login to the Cardholder account at the 3DS Requestor system using 3DS Requestor’s own credentials • 03 = Login to the Cardholder account at the 3DS Requestor system using federated ID • 04 = Login to the Cardholder account at the 3DS Requestor system using Issuer credentials • 05 = Login to the Cardholder account at the 3DS Requestor system using third-party authentication • 06 = Login to the Cardholder account at the 3DS Requestor system using FIDO Authenticator • 07 = Login to the Cardholder account at the 3DS Requestor system using FIDO Authenticator (FIDO Assertion or Attestation data signed) • 08 = SRC Assurance Data	2.3.1 2.2

Data Element	Description	Version
	• 09 = SPC Authentication • 10 = Electronic ID Authentication Data • 11–79 = Reserved for EMVCo future use (values invalid until defined by EMVCo) • 80–99 = Reserved for DS use	
3DS Requestor Authentication Timestamp	Date and time of the Cardholder authentication converted into UTC.	2.3.1 2.2
DS Authentication Information Verification Indicator	Supplied by the DS, the value that represents the signature verification performed by the DS on the mechanism (e.g., FIDO) used by the Cardholder to authenticate to the 3DS Requestor. The DS populates this data element prior to passing to the ACS. Values accepted: • 01 = Verified • 02 = Failed • 03 = Not performed	2.3.1 2.2

2.2.9 Exemptions

Some markets require Issuers to validate all e-commerce transactions with the Cardholder (i.e., PSD2 – Strong Customer Authentication (SCA) requirements). However, the regulations may allow certain exemptions from the SCA requirements.

One of such exemptions is the **Transaction Risk Analysis** exemption. Merchants or their acquiring banks assess the risk associated with the transaction. If the transaction risk is low (below a certain threshold), the Merchants can request an SCA exemption.

Other exemptions include:

- **Low Transaction Value:** transactions below a pre-defined amount.
- **Trust List:** if the Issuer supports it, the Cardholder can add Merchants to their Trust List after a successful SCA transaction.
- **Secure Corporate Payments**: transactions initiated by secure corporate accounts (cards held by third-party agents or virtual cards).

Merchants can request exemptions for qualifying transactions. If the exemptions are accepted (granted) by the Issuer, they allow for a frictionless checkout, reducing the risk of abandonment linked to the challenge.

Benefits by Actor

- Merchant: increased frictionless authentication outcomes, fewer transactions lost due to Cardholder abandonment or 3DS authentication failure during the 3DS process.

- Issuer: reduced need to request a challenge
- Cardholder: improved shopping experience due to reduced friction

Technical Features

Preconditions

The Merchant or 3DS Server is able to evaluate the context of the transaction and its risk, and request an exemption.

The ACS is able to manage the challenge exemption.

Optional: The ACS indicates the supported exemption in the Card Range Data (ACS Information Indicator – 08, 09, 10 or 11)

Overview

During the checkout process, the 3DS Server may:
- verify if the ACS supports the exemptions using the information from the ACS Information Indicator in the Card Range Data
- request an exemption using the 3DS Requestor Challenge Indicator in the AReq message.

The ACS evaluates the risk associated with the transaction, taking into account the request for an exemption (if present). If an exemption is applicable, the ACS may provide this information in the Transaction Challenge Exemption in the ARes message.

Note: The Core Specification supports the exchange of information related to the exemption between the 3DS Server and the ACS. The presence and use of this information is optional, and depends on the regulations applicable in the relevant market.

Table 2.8 below lists the data elements that may be provided in relation to exemptions.

Table 2.8: 3DS Data Elements Related to Exemptions

Data Element	Description	Version
3DS Requestor Challenge Indicator	Indicates whether a challenge is requested for this transaction. Values related to the exemptions • 05 = No challenge requested (transactional risk analysis is already performed) • 08 = No challenge requested (use Trust List exemption if no challenge required) • 10 = No challenge requested (use low value exemption) • 11 = No challenge requested (Secure corporate payment exemption)	2.3.1 2.2

Data Element	Description	Version
Transaction Challenge Exemption	Exemption applied by the ACS to authenticate the transaction without requesting a challenge. Values related to the exemptions • 05 = Transaction Risk Analysis exemption • 08 = Trust List exemption • 10 = Low Value exemption • 11 = Secure Corporate Payments exemption • 79 = No exemption applied	2.3.1 2.2 + Bridging Message Extension
ACS Information Indicator	Provides additional information for a particular Protocol Version to the 3DS Server. The element lists all applicable values for the card range. Values related to exemption • 08 = Transaction Risk Analysis Exemption Supported • 09 = Trust List Exemption Supported • 10 = Low Value Exemption Supported • 11 = Secure Corporate Payments Exemption Supported	2.3.1 2.2

Use Case 1 – Low-Value Exemption

The transaction value is below a predefined threshold, so the Merchant and/or 3DS Server can request an SCA exemption.

Before sending an Authentication Request with 3DS Requestor Challenge Indicator = 10 (No challenge requested - use low value exemption), the 3DS Server may check that the ACS supports the Low-Value Exemption using ACS Information Indicator (10 = Low Value Exemption Supported).

The ACS verifies that the transaction value is below the threshold, determines that a Cardholder challenge is not necessary, and applies the Low Value Exemption. It may return Transaction Challenge Exemption = 10 (Low Value exemption) with its response.

Merchant / 3DS Server	Issuer / ACS
Low Value Exemption threshold = 50 € (example) Transaction data elements • Purchase Amount = 999 • Purchase Currency = 978 (€) • Purchase Currency Exponent = 2 • Purchase Date & Time = 20241118092600 • 3DS Requestor Challenge Indicator = **10**	**Transaction data elements** • Transaction Status = Y • Transaction Challenge Exemption = **10**

Use Case 2 – Transaction Risk Analysis Exemption

The Merchant or their acquiring bank assesses the risk associated with the transaction. If the transaction risk is low (below a certain threshold), the Merchant and/or 3DS Server can request an SCA exemption.

Before sending an Authentication Request with 3DS Requestor Challenge Indicator 05 = No challenge requested - transactional risk analysis is already performed), the 3DS Server may check that the ACS supports the Transaction Risk Analysis Exemption using ACS Information Indicator (08 = Transaction Risk Analysis Exemption Supported).

The ACS determines that the Merchant/Acquirer performed the transaction risk analysis and that a Cardholder challenge is not necessary, it applies the Transaction Risk Analysis exemption. It may return Transaction Challenge Exemption = 05 (Transaction Risk Analysis exemption) with its response.

Merchant/Acquirer	Issuer
The Merchant assesses that the transaction risk is low. Transaction data elements • Purchase Amount = 9999 • Purchase Currency = 978 (€) • Purchase Currency Exponent = 2 • Purchase Date & Time = 20241118104200 • 3DS Requestor Challenge Indicator = **05**	**Transaction data elements** • Transaction Status = Y • Transaction Challenge Exemption = **05**

Use Case 3 – Trust List Exemption

The Merchant or the 3DS Server knows that the Merchant is listed on the Cardholder's Trust List (refer to Section 2.2.4 for more information on the Trust List), the Merchant and/or 3DS Server can request an SCA exemption.

Before sending an Authentication Request with 3DS Requestor Challenge Indicator = 08 (No challenge requested - use Trust List exemption if no challenge required), the 3DS Server may check that the ACS supports the Trust List Exemption using ACS Information Indicator = 09 (Trust List Exemption supported).

The ACS verifies that the Merchant is on the Cardholder's Trust List, determines that a Cardholder challenge is not necessary, and applies the Trust List exemption. It may return Transaction Challenge Exemption = 08 (Trust List exemption) with its response.

Merchant/Acquirer	Issuer
The Merchant knows that it is on the Cardholder's Trust List. Transaction data elements • Purchase Amount = 9999 • Purchase Currency = 978 (€) • Purchase Currency Exponent = 2 • Purchase Date & Time = 20241118111000 • 3DS Requestor Challenge Indicator = **08**	**Transaction data elements** • Transaction Status = Y • Transaction Challenge Exemption = **08**

Use Case 4 – Secure Corporate Payments Exemption

The Merchant or the 3DS Server knows that the card used for the transaction is a corporate payment card (i.e., card used for business expenditures or card account used for business-to-business payments), the Merchant and/or 3DS Server can request an SCA exemption.

Before sending an authentication request with 3DS Requestor Challenge Indicator = 11 (No challenge requested - Secure corporate payment exemption), the 3DS Server may check that the ACS supports the Secure Corporate Payments Exemption using ACS Information Indicator = 11 (Secure Corporate Payments Exemption Supported).

The ACS verifies that the type of card used for the transaction, determines that a Cardholder challenge is not necessary, and applies the Secure Corporate Payments exemption. It may return Transaction Challenge Exemption = 11 (Secure Corporate Payments Exemption) with its response.

Merchant/Acquirer	Issuer
The Merchant knows that the card used for the transaction is a corporate card. Transaction data elements • Purchase Amount = 9999 • Purchase Currency = 978 (€) • Purchase Currency Exponent = 2 • Purchase Date & Time = 20241118111000 • 3DS Requestor Challenge Indicator = **11**	**Transaction data elements** • Transaction Status = Y • Transaction Challenge Exemption = **11**

3 Recurring and Instalment Transactions

3.1 Business Overview

Recurring payments involve Cardholders granting permission for Merchants to automatically charge their payment cards to cover subscription-type agreements, providing peace of mind for Cardholders and an easier collection process for Merchants. There are many types of recurring payments, depending on whether the amount and the frequency are fixed. For example, the frequency can be predefined, such as each week, month or year or can be non-fixed and triggered by a specific usage event (for example, when the balance on a prepaid card falls below $5, reload with $20. The amount can also be fixed (for example, "reload by $20" in the previous example) or variable – when the amount itself is dependent on usage (for example, a utility bill). Fixed amount does not necessarily mean fixed frequency, and vice versa. The use case determines the parameters of the recurring or instalment payment.

An instalment payment is a payment made over time according to a pre-agreed schedule for goods and services that have been fully delivered or performed. Instalment transactions are explained in more detail in Use Case 7 below.

Merchants enjoy several advantages with recurring and instalment payments. They experience fewer late payments, as the automated system ensures that payments are collected on time. This contributes to consistent cash flow and peace of mind. Recurring and instalment payments also save time and resources for Merchants, as they eliminate the need for manual invoicing and payment collection. Additionally, Merchants can build better customer relationships by offering the convenience of recurring and instalment payments. Cardholders appreciate not having to remember to make additional payments or re-enter payment information, which results in greater satisfaction and loyalty.

Good communication between the Merchant, the Cardholder and the ACS/Issuer is essential during the set-up of recurring and instalment transactions to prevent disputes or declined transactions. The Core Specification enables the Merchant to provide detailed information on recurring and instalment transactions using the data elements available in the 3DS protocol.

Benefits by Actor

- Merchant
  - may help in solving disputes with Cardholders as to whether the recurring or instalment payment was put in place
  - leverages a single authentication to set up a recurring or instalment transaction at the same time as a purchase
- Issuer – receives detailed information about the recurring or instalment transaction to make it clear that an agreement is entered into
- Cardholder
  - enjoys the convenience of using a recurring or instalment transaction rather than initiating multiple transactions

- receives detailed information about the recurring or instalment transaction before proceeding with the transaction

3.2 Technical Features

Preconditions

Depending on the use case, the appropriate Core Specification version or the Bridging Message Extension must be supported.

Additionally, to initiate 3RI payment authentications for subsequent payments in a recurring or instalment transaction, EMV 3DS version 2.2 or 2.3.1 is required. The DS Transaction ID and/or the ACS Transaction ID, which was received in the initial authentication, is kept and used in each associated 3RI transaction.

3DS Data Elements Related to Recurring and Instalment Transactions

Table 3.1 below lists the data elements that may be provided by 3DS Servers to support recurring and instalment transactions.

For additional information, refer to Table A.1 in the Core Specification and to the EMV 3-D Secure Bridging Message Extension.

Table 3.1: 3DS Data Elements Related to Recurring and Instalment Transactions

Data Element	Description	Version
3DS Requestor Authentication Indicator	Indicates the type of Authentication request. This data element provides additional information to the ACS to determine the best approach for handling an authentication request. A value of 02 indicates that this authentication is requested for a recurring transaction. A value of 03 indicates that this authentication is requested for an instalment transaction.	2.3.1 2.2
3DS Requestor Prior DS Transaction ID	This data element is within the 3DS Requestor Prior Transaction Authentication Information object and contains a DS Transaction ID for a prior authenticated transaction (for example, the first recurring transaction that was authenticated with the Cardholder).	2.3.1

Data Element	Description	Version
3DS Requestor Prior Transaction Reference	This data element is within the 3DS Requestor Prior Transaction Authentication Information object and contains an ACS Transaction ID for a prior authenticated transaction (for example, the first recurring transaction that was authenticated with the Cardholder).	2.3.1 2.2
3RI Indicator	Indicates the type of 3RI request. This data element provides additional information to the ACS to determine the best approach for handling a 3RI request. A value of 01 indicates that this authentication is requested for a recurring transaction (PA). A value of 02 indicates that this authentication is requested for an instalment transaction (PA). A value of 05 indicates that this authentication is requested for an account verification with recurring payment data for information (NPA). 3RI requests are used when the Merchant decides to authenticate subsequent transactions.	2.3.1 2.2
Instalment Payment Data	Indicates the maximum number of authorisations permitted for instalment payments.	2.3.1 2.2
Purchase Amount	Purchase amount in minor units of currency with all punctuation removed. The purchase amount is the amount payable at the time of purchase, which includes both: • the amount of the one-time purchase (when there is one) and • the amount of the recurring transaction also payable that day (if there is one). The amount payable at the set-up of a recurring payment can be the recurring amount itself, a promotional amount (i.e. a percentage of the recurring amount) or even zero if no amount is due at the time of purchase.	2.3.1 2.2
Purchase Currency	Currency in which the Purchase Amount is expressed.	2.3.1 2.2

Data Element	Description	Version
Purchase Currency Exponent	Minor units of currency as specified in the ISO 4217 currency exponent. Examples: • USD = 2 • JPY = 0	2.3.1 2.2
Purchase Date & Time	Date and time of the authentication converted into UTC.	2.3.1 2.2
Recurring Amount	Recurring amount in minor units of currency with all punctuation removed. Recurring amount is specified if the recurring payment is a fixed amount. In the case of instalment payments, the instalment amount is included in this Recurring Amount field.	2.3.1 2.2 + Bridging Message Extension
Recurring Currency	Currency in which the Recurring (or instalment) Amount is expressed.	2.3.1 2.2 + Bridging Message Extension
Recurring Currency Exponent	Minor units of currency as specified in the ISO 4217 currency exponent. Examples: • USD = 2 • JPY = 0	2.3.1 2.2 + Bridging Message Extension
Recurring Date	Effective date of the new authorised amount following the first/promotional payment in a recurring or instalment transaction. Recurring date is specified if the date is fixed.	2.3.1 2.2 + Bridging Message Extension
Recurring Expiry	Date after which no further authorisations are performed. This applies to both recurring and instalment payments. Recurring expiry is often not specified for a recurring payment in cases where there is no known expiry date.	2.3.1 2.2 + Bridging Message Extension 2.2

Data Element	Description	Version
Recurring Frequency	Indicates the minimum number of days between authorisations for a recurring or instalment transaction.	2.3.1 2.2 + Bridging Message Extension 2.2
Recurring Indicator	Indicates whether the recurring or instalment payment has a fixed or variable amount and frequency. The Recurring Indicator object contains: • the Amount Indicator • the Frequency Indicator	2.3.1 2.2 + Bridging Message Extension 2.2

3.2.1 Cardholder-Initiated Flow (App-Based or Browser-Based Device Channels)

Overview

For the initial set-up of a recurring transaction agreement, the Cardholder is present (i.e. the Cardholder is initiating the payment transaction). 3DS Servers should provide the relevant recurring transaction data elements to allow the ACS to determine the appropriate authentication action (i.e., Frictionless Flow or Cardholder challenge). If the transaction is challenged, the recurring transaction data elements are used to determine the information to display to the Cardholder. Upon a successful authentication, a DS Transaction ID and an ACS Transaction ID are provided by the ACS and returned to the 3DS Server in an ARes message (or RReq message, in the case of a Cardholder challenge). The DS Transaction ID and/or ACS Transaction ID should be provided in future authentication requests directly related to the recurring transaction to help the ACS reference the details of the initial authentication.

Sequence Diagram

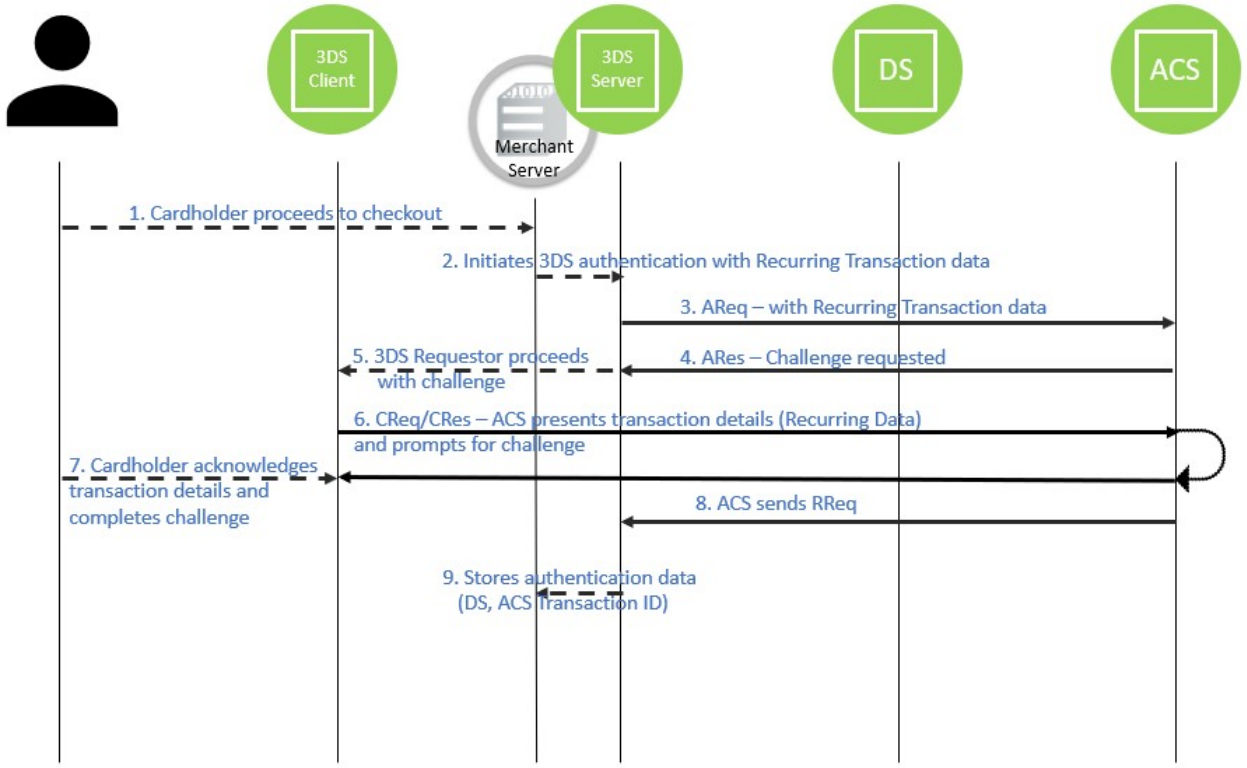
The Cardholder and the Merchant agree on the set-up of a recurring or instalment transaction.

1. The Cardholder makes a purchase that includes a recurring or instalment payment.
2. The 3DS Requestor initiates a 3DS authentication and provides the details of the purchase, in particular the recurring transaction data elements.
3. The 3DS Server sends an AReq message.
4. The ACS responds with a challenge (ARes).
5. The 3DS Server proceeds with the challenge (opens an iframe for a Browser flow or provides the relevant data to the 3DS SDK for an App flow).
6. The ACS proceeds with the challenge and provides the UI that includes the transaction information. The ACS uses the transaction data from the AReq to provide the recurring transaction details to the Cardholder (amount, frequency, expiry date…).
7. The Cardholder acknowledges the transaction details and completes the challenge.

Note: How the Cardholder acknowledges the transaction details is an implementation decision from the ACS.

8. The ACS may store the details of the authentication for future 3RI processing, and provides the outcome of the authentication in the RReq message to the DS and 3DS Server.
9. The 3DS Requestor stores the details of the authentication (DS Transaction ID and/or ACS Transaction ID, Authentication Value) for future authentication.

Figure 3.1: Cardholder-Initiated Flow



3.2.2 Merchant-Initiated Flow (3RI Device Channel)

Overview

For subsequent payments in a recurring transaction, 3DS Requestors should use the 3RI Indicator to indicate that this is a recurring transaction (01 = Recurring transaction) or instalment transaction (02 = instalment). Additionally, 3DS Requestors should provide the DS Transaction ID and/or the ACS Transaction ID, which was received in the initial authentication, in the 3DS Requestor Prior Transaction Authentication Information object of the AReq message as it allows the ACS to reference the authentication from the initial set-up.

Note: 3RI payment authentications are supported in EMV 3DS version 2.2 and above.

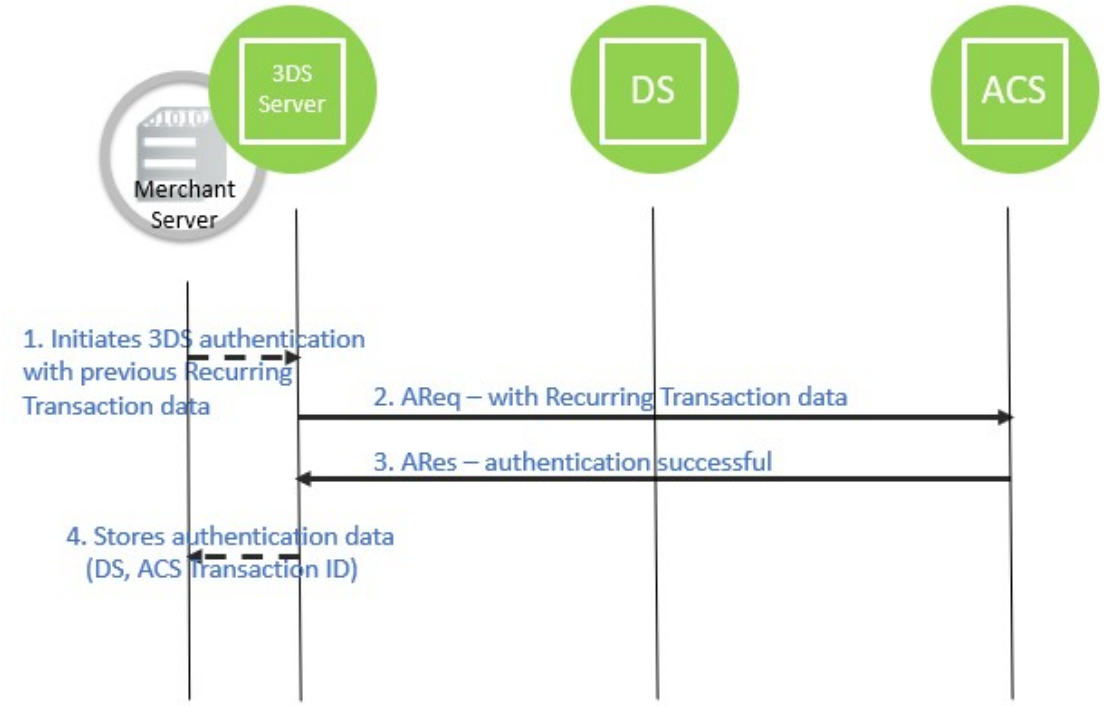
Sequence Diagram

In a subsequent transaction with the same Merchant:

1. The Merchant needs to renew the recurring transaction before it expires (assuming the Merchant uses 3RI transactions to authenticate on an ongoing basis).

The 3DS Requestor initiates a 3DS authentication and provides the details of the previous transaction (DS Transaction ID and/or ACS Transaction ID) in the 3DS Requestor Prior Transaction Reference.
2. The 3DS Server sends a 3RI Authentication Request (AReq).
3. The ACS matches the references provided in the 3DS Requestor Prior Transaction Reference to the initial Cardholder-initiated transaction, responds with an approval (ARes) to the DS and 3DS Server.
4. The 3DS Requestor stores the details of the authentication (DS Transaction ID and/or ACS Transaction ID, Authentication Value) for future authentication.

Figure 3.2: Merchant-Initiated Flow



3.3 Use Cases

The following use cases illustrate the technical capabilities of 3DS with recurring data elements and cover the most common types of recurring or instalment transactions. Payment systems may impose additional requirements on the use of recurring data, including for the purposes of ensuring compliance with market regulations.

3.3.1 Use Cases for Version 2.2

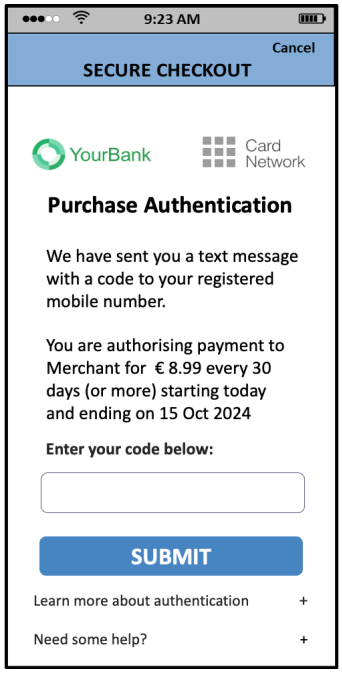
With version 2.2 of the Core Specification, the Merchant has a limited set of data available to provide to the ACS about a recurring or instalment transaction. For example, the Merchant cannot indicate if the recurring transaction has a variable amount, or if the instalment amount is different from the initial amount (purchase amount).

Presented below are example use cases for recurring or instalment transactions in version 2.2:

1. Recurring payment with a fixed frequency
2. Instalment payment

Use Case 1: Recurring Payment with a Fixed Frequency

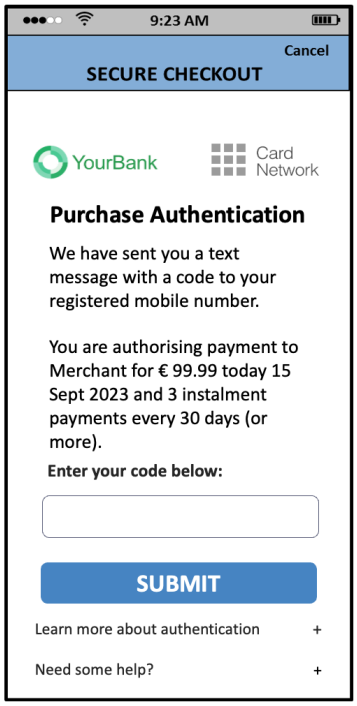
In this use case, the amount due at recurring payment set-up is the same amount that will be due on a recurring basis. In the example below, the Cardholder is committing to pay €8.99 monthly, starting on the day of the purchase and ending on 15 October 2024.

Merchant/Acquirer	Issuer	Cardholder
Existing recurring data elements • Purchase Amount = 899 • Purchase Currency = 978 (€) • Purchase Currency Exponent = 2 • Purchase Date & Time = 20230915120000 • Recurring Expiry = 20241015 • Recurring Frequency = 30 • 3DS Requestor Authentication Indicator = 02	Determines that a Cardholder challenge is necessary and builds the message with the recurring payment information. Message is displayed to the Cardholder in the 3DS challenge window.	

Use Case 2: Instalment Payment

An instalment payment is a payment made over time according to a pre-agreed schedule for goods and services that have been fully delivered or performed.

One example is the purchase for a total of €999.99, to be paid in 4 instalments (i.e., 3 times €300.00 each month after the first payment of €99.99), with the first payment occurring on the day of the purchase. The Merchant cannot provide information on the different amounts between the first payment and the 3 successive instalment payments. Similarly, if anything else is purchased at the same time as the instalment set-up, the amount of that purchase would be added to the purchase amount with the first instalment payment.

Merchant/Acquirer	Issuer	Cardholder
Existing recurring data elements • Purchase Amount = 9999 • Purchase Currency = 978 (€) • Purchase Currency Exponent = 2 • Purchase Date & Time = 20230915120000 • Recurring Frequency = 30 • Instalment Payment Data = 04 • 3DS Requestor Authentication Indicator = 03	Determines that a Cardholder challenge is necessary and builds the message with the recurring payment information. Message is displayed to the Cardholder in the 3DS challenge window.	

3.3.2 Use Cases for Version 2.3.1

With version 2.3.1 of the Core Specification, the Merchant has a large set of data available to provide to the ACS about a recurring or instalment transaction. For example, the Merchant can indicate if the recurring transaction has a variable amount, or if the recurring amount is different from the initial amount (purchase amount).

The use cases are also possible with version 2.2 of the Core Specification if the Bridging Message Extension with a Recurring Data object is present and supported by the ACS and the 3DS Server:

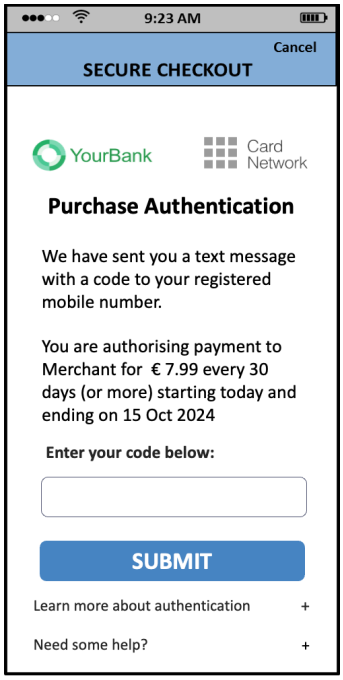
1. Recurring payment with a fixed amount and a fixed frequency.
2. Recurring payment with a fixed amount, fixed frequency, and a promotional rate.
3. Recurring payment with a variable amount and a fixed frequency.
4. Recurring payment with a variable amount and a variable frequency.
5. Recurring payment with a fixed amount and a variable frequency.
6. Recurring payment combined with one-time purchase.
7. Instalment payments.

Use Case 1: Recurring Payment with a Fixed Amount and a Fixed Frequency

In this scenario, the amount due at recurring payment set-up is the amount that will be due on a recurring basis. Use Case 2 covers the scenario when the two amounts differ.

In the example below, the Cardholder is committing to pay €7.99 monthly, starting on the day of the purchase and ending on 15 October 2024.

Merchant/Acquirer	Issuer	Cardholder

Existing recurring data elements
• Purchase Amount = 799
• Purchase Currency = 978 (€)
• Purchase Currency Exponent = 2
• Purchase Date & Time = 20230915120000
• Recurring Expiry = 20241015
• Recurring Frequency = 30
• 3DS Requestor Authentication Indicator = 02
 	
Additional elements
• Recurring Amount = 799
• Recurring Indicator
  ○ Amount Indicator = 01
  ○ Frequency Indicator = 01
• Recurring Currency = 978 (€)
• Recurring Currency Exponent = 2
• Recurring Date = 20231015
	Determines that a Cardholder challenge is necessary and builds the message with the recurring payment information.
	
	Message is displayed to the Cardholder in the 3DS challenge window.
		

Use Case 2: Recurring Payment with a Fixed Amount, Fixed Frequency, and a Promotional Rate

A rate is considered promotional when the amount to be paid at set-up is not the same as the amount to be paid on an ongoing basis. In the case of an ongoing subscription of €7.99/month, the data listed below need to be provided.

If the first month is free:

- The purchase amount will be zero.
- The recurring amount will be €7.99.

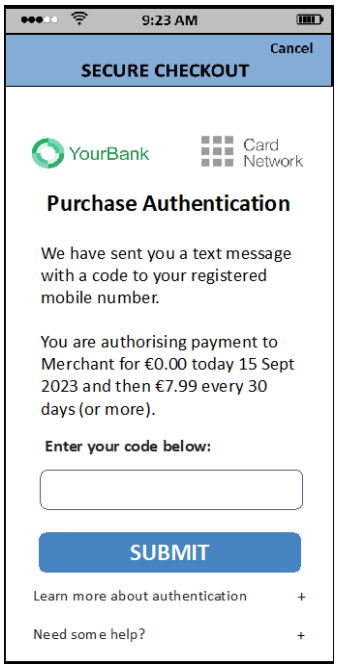
If there is a 50% discount:

- The purchase amount will be €3.99.
- The recurring amount will be €7.99.

If there is no promotional rate and the €7.99 is also due on the day of the purchase, refer to Use Case 1.

In the example below, the first month is free, the recurring amount is €7.99, and there is no end date provided. When there is no end date, there is no need to (and it is recommended not to) convey this information.

Merchant/Acquirer	Issuer	Cardholder

Existing recurring data elements
• Purchase Amount = 0
• Purchase Currency = 978 (€)
• Purchase Currency Exponent = 2
• Purchase Date & Time = 20230915120000
• Recurring Frequency = 30
• 3DS Requestor Authentication Indicator = 02
 	
Additional elements
• Recurring Indicator
  ○ Amount Indicator = 01
  ○ Frequency Indicator = 01
• Recurring Amount = 799
• Recurring Currency = 978 (€)
• Recurring Currency Exponent = 2
• Recurring Date = 20231015
	Determines that a Cardholder challenge is necessary and builds the message with the recurring payment information.
	
	Message is displayed to the Cardholder in the 3DS challenge window.
		

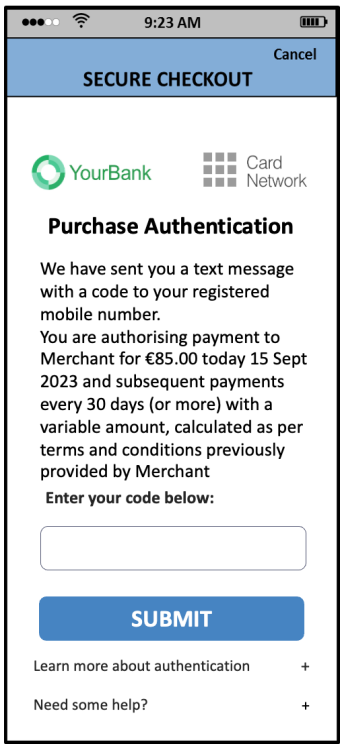
Use Case 3: Recurring Payment with a Variable Amount and a Fixed Frequency

In the case of a recurring payment with a variable amount, the method of calculating the amount is typically communicated in the Merchant's terms and conditions of the recurring payment set-up. For example, in the case of an electricity bill, a Merchant will typically inform the Cardholder that the amount to be charged will depend on usage and will be calculated on display €x per kW of usage.

It is recommended that Issuers find a generic way to convey the meaning of "variable amount" by using terms such as "of an amount calculated as per the terms and conditions previously displayed by the merchant". When there is no end date, there is no need to (and it is recommended not to) convey this information.

The example below shows data provided when the Merchant has required a payment of €85 on the day of the purchase (considered the average monthly payment) and payment on usage every month starting one month after that date.

Merchant/Acquirer	Issuer	Cardholder

Existing recurring data elements
• Purchase Amount = 8500
• Purchase Currency = 978 (€)
• Purchase Currency Exponent = 2
• Purchase Date & Time = 20230915120000
• Recurring Frequency = 30
• 3DS Requestor Authentication Indicator = 02
 	
Additional elements
• Recurring Indicator
  ○ Amount Indicator = 02
  ○ Frequency Indicator = 01
• Recurring Date = 20231015
	Determines that a Cardholder challenge is necessary and builds the message with the recurring payment information.
	
	Message is displayed to the Cardholder in the 3DS challenge window.
		

Use Case 4: Recurring Payment with a Variable Amount and a Variable Frequency

In the case of a recurring payment with:

- a variable amount – the method of calculating the amount is typically communicated as part of the Merchant's terms and conditions of the recurring payment set-up;
- a variable frequency – the event that will trigger a charge/payment is typically described to the Cardholder in the Merchant's terms and conditions of the recurring payment set-up.

For example, in the case of a payment to be collected by a highway operator when the Cardholder's transponder is used on a highway, the operator informs the Cardholder that a payment will be charged at the end of the day, every time the highway is used on that day, for an amount based on the distance driven.

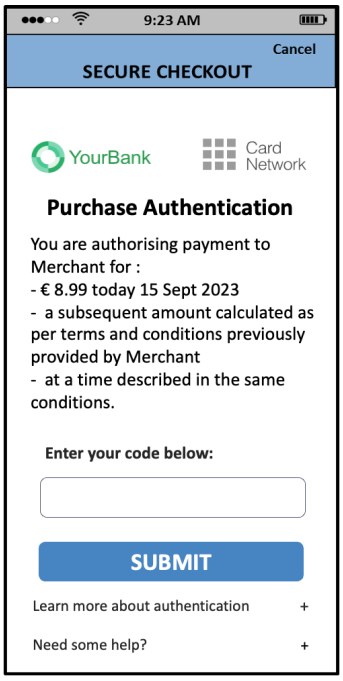
It is recommended that Issuers find a generic way to convey the meaning of "variable amount" and "variable frequency" to Cardholders by using terms such as:

- for amount – "of an amount calculated as per the terms and conditions previously displayed by the merchant";
- for frequency – "at a time described in the terms and conditions previously displayed by the merchant".

When both amount and frequency are variable, Issuers should try to avoid displaying the wording “described in the terms and conditions previously described by the merchant” twice, for example, as per the image below.

Note: Payment systems may impose additional requirements on the use of recurring data or may set limits such as a maximum amount.

Merchant/Acquirer	Issuer	Cardholder

Existing recurring data elements
• Purchase Amount = 899
• Purchase Currency = 978 (€)
• Purchase Currency Exponent = 2
• Purchase Date & Time = 20230915120000
• 3DS Requestor Authentication Indicator = 02
 	
Additional elements
• Recurring Indicator
  ○ Amount Indicator = 02
  ○ Frequency Indicator = 02
	Determines that a Cardholder challenge is necessary and builds the message with the recurring payment information.
	
	Message is displayed to the Cardholder in the 3DS challenge window.
		

Use Case 5: Recurring Payment with a Fixed Amount and a Variable Frequency

In the case of payments with a variable frequency, the event that will trigger a charge/payment is typically described to the Cardholder in the Merchant’s terms and conditions of the recurring payment set-up.

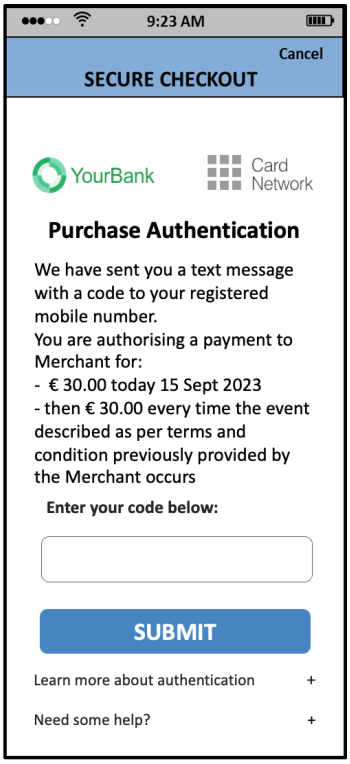
For example, in the case of a payment to be collected by a transit operator that provides prepaid transit cards, the triggering event could be the transit card balance falling below an amount set by the Cardholder or set by the operator and communicated to the Cardholder.

The terms and conditions set forth by the operator may state, for example, that a reload of €30.00 will occur when the transit card balance falls below €15.00.

In the example below, at set-up, the card is loaded with €30 and a reload of €30 will occur when the balance falls below €15.00.

Note: Payment systems may impose additional requirements on the use of these data or may set limits such as a maximum transaction per recurring period.

Merchant/Acquirer	Issuer	Cardholder

Existing recurring data elements
• Purchase Amount = 3000
• Purchase Currency = 978 (€)
• Purchase Currency Exponent = 2
• Purchase Date & Time = 20230915120000
• 3DS Requestor Authentication Indicator = 02
 	
Additional elements
• Recurring Indicator
  ○ Amount Indicator = 01
  ○ Frequency Indicator = 02
• Recurring Amount = 3000
• Recurring Currency = 978 (€)
• Recurring Currency Exponent = 2
	Determines that a Cardholder challenge is necessary and builds the message with the recurring payment information.
	
	Message is displayed to the Cardholder in the 3DS challenge window.
		

Use Case 6: Recurring Payment, Combined with a One-Time Purchase

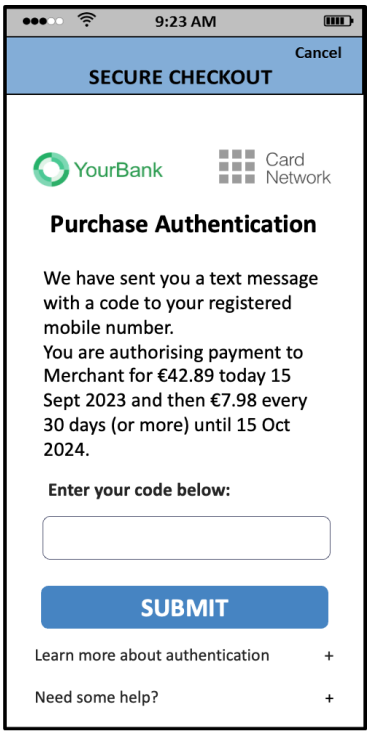
In every use case, the amount to be sent in the purchase amount is the amount the Cardholder must pay on the day of the authentication. This will include:

- the amount of the one-time purchase when there is one
- the amount of the recurring agreement also payable that day:
  - if there is a promotion and no amount is payable that day, this amount is zero, but if any amount is payable that day, this amount must be added to the amount of the one-time purchase;
  - if both the amount of the one-time purchase and a recurring amount is to be paid that day, it is not possible to indicate the individual amount for each.

The amounts and frequency to be provided in the recurring data amount and frequency should follow the principles set forth in Use Cases 1–5.

For example, if the one-time purchase has a value of €42.89 and the fixed recurring payment is free for the first month and payable only in the second month (see example below), then the purchase amount should be sent as €42.89. If 50% of the recurring amount is due on the day of the purchase (€3.99), in addition to the purchase of €42.89, the amount to be sent in the purchase amount would be €46.88 (sum of €42.89 and €3.99).

Merchant/Acquirer	Issuer	Cardholder

Existing recurring data elements
• Purchase Amount = 4289
• Purchase Currency = 978 (€)
• Purchase Currency Exponent = 2
• Purchase Date & Time = 20230915120000
• Recurring Expiry = 20241015
• Recurring Frequency = 30
• 3DS Requestor Authentication Indicator = 02
 	
Additional elements
• Recurring Indicator
  ○ Amount Indicator = 01
  ○ Frequency Indicator = 01
• Recurring Amount = 798
• Recurring Currency = 978 (€)
• Recurring Currency Exponent = 2
• Recurring Date = 20231015
	Determines that a Cardholder challenge is necessary and builds the message with the recurring payment information.
	
	Message is displayed to the Cardholder in the 3DS challenge window.
		

Use Case 7: Instalment Payment

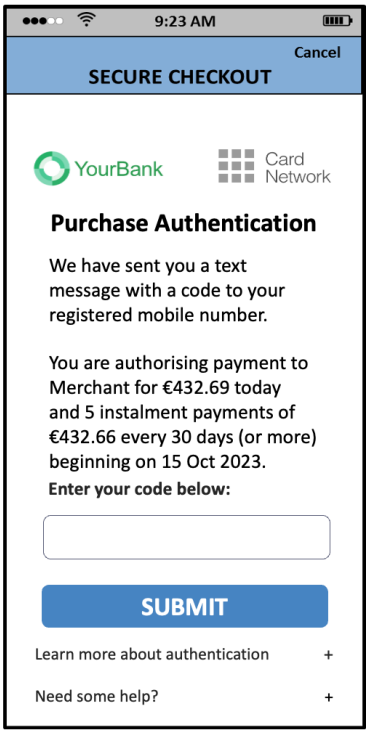
An instalment payment is a payment made over time according to a pre-agreed schedule for goods and services that have been fully delivered or performed.

One example is the purchase of a sofa for a total of €2595.99, to be paid in 6 instalments (i.e. €432.66 each after the first payment of €432.69), with the first payment occurring on the day of the purchase. As the first instalment is paid on the day of the purchase, only 5 instalments remain. The value provided in the Instalment Payment Data corresponds to the maximum number of authorisations permitted for instalment payments (6 in this example).

If any other purchase was made at the same time as the instalment set-up, the amount of that purchase would be added to the purchase amount with the first instalment payment, following the principles set forth in Use Case 6 above.

Note: Payment systems may impose different requirements on the use of these data. For example, they may require that the total amount (€2595.99) be provided in the Purchase Amount.

Merchant/Acquirer	Issuer	Cardholder

Existing recurring data elements
• Purchase Amount = 43269
• Purchase Currency = 978 (€)
• Purchase Currency Exponent = 2
• Purchase Date & Time = 20230915120000
• Recurring Frequency = 30
• Instalment Payment Data = 06
• 3DS Requestor Authentication Indicator = 03
 	
Additional elements
• Recurring Indicator
  ○ Amount Indicator = 01
  ○ Frequency Indicator = 01
• Recurring Amount = 43266
• Recurring Currency = 978 (€)
• Recurring Currency Exponent = 2
• Recurring Date = 20231015
	Determines that a Cardholder challenge is necessary and builds the message with the recurring payment information.
	
	Message is displayed to the Cardholder in the 3DS challenge window.
		

3.3.3 Best Practices for Defining Recurring Frequency Values

The Recurring Frequency data element defines the minimum number of days between authorisations. This is a limitation, as Merchants often charge on a fixed interval basis – not necessarily based on the number of days but on a calendar interval (week, month, quarter…). It is recommended that Merchants use the Recurring Frequency values indicated in Table 4.2 to ensure that the Issuer's message is expressed as a calendar interval rather than a number of days. Issuers receiving those Recurring Frequency values should use the corresponding calendar intervals to display recurring transaction information to Cardholders.

Table 3.2 below provides the recommended Issuer messaging for Recurring Frequency values.

Table 3.2: Recommended Issuer Messaging for Recurring Frequency Values

Recurring Frequency value	Issuer messaging
7	Every week
14	Biweekly
28	Every month
59	Bimonthly
89	Quarterly
181	Twice a year
365	Annually

For example, for a Recurring Frequency of 28 days, the ACS should interpret the frequency as a monthly payment and provide the following message to the Cardholder:
"You are authorising payment to *[Merchant abc]* every **month**".

If the ACS does not interpret the 28 days as a monthly payment, it may provide the following message to the Cardholder:
"You are authorising payment to *[Merchant abc]* every **28 days (or more)**".

3.3.4 Recurring Transactions and the Bridging Message Extension

For more information on using the Bridging Message Extension for recurring transactions, please refer to Section 6.4.

4 Challenge Flow

4.1 Business Overview

The 3DS Challenge flow occurs when a Cardholder's transaction cannot be authenticated through a Frictionless Flow or when the ACS determines that the transaction is deemed high-risk. The Cardholder may also be required to do this when the Cardholder agreed to their payment details being used for subsequent purchases like recurring payments.

For the Challenge flow, the Cardholder must provide additional information directly to the ACS in order for the transaction to take place. This may include entering a one-time code sent to their mobile device or validating the transaction using their mobile banking application (Out-of-Band Authentication).

The 3DS Challenge flow significantly reduces fraud by requiring this extra layer of authentication, which in turn increases consumer confidence in online shopping. Cardholders are more likely to complete a purchase when they feel confident in the security measures in place.

The challenge process provides valuable information on transaction patterns, as well as Cardholder and ACS behaviour, helping 3DS Requestors to improve their fraud prevention strategies, and balance security and user experience based on their specific risk profiles.

In some regions, regulations require Strong Customer Authentication (SCA) for online payments. Implementing 3DS and supporting the challenge process helps businesses to comply with these regulations.

Overall, the challenge process in 3DS enhances the security of online transactions while providing benefits to Cardholders, 3DS Requestors and banks (ACSs).

Benefits per actor

- 3DS Requestor
  - helps to significantly reduce the risk of fraudulent transactions
  - implementing 3DS helps 3DS Requestors to comply with regulations requiring SCA for online transactions
- ACS
  - helps to significantly reduce the risk of fraudulent transactions
  - various authentication methods available to accommodate bank preferences and market requirements
- Cardholder
  - may view online transactions as safer due to the challenge providing an extra layer of security
  - similar user experience across all 3DS Requestors

4.2 Technical Features

This section offers a general introduction to the Challenge Flow and the subsequent sections detail the more complex challenge flows such as SPC, Out-of-Band and Decoupled Authentication. For the common challenge methods such as One-Time Passcode (OTP) or multi-select, please refer to the Core Specification.

After reviewing the information provided in the AReq message, the ACS may decide that further checks are required to complete the authentication. The ACS will respond to the 3DS Server with a Challenge Request (instead of an approval for the Frictionless Flow). For example, a challenge may be necessary because the transaction is deemed high-risk (above a certain threshold) or requires Cardholder authentication due to country mandates or regulations.

The 3DS Requestor can decide whether to proceed with the challenge or terminate the authentication process. For the challenge, the 3DS Client is in direct communication with the ACS through a secure connection, so the ACS can directly interact with the Cardholder for the authentication.

3DS Data Elements Related to the Challenge Flow

Note: There are numerous data elements related to challenge processing. Table 4.1 below only lists those data elements that relate to the 3DS Requestor request in terms of the challenge and response from the ACS. The other data elements depend on the channel (Browser, App) and on the challenge method used by the ACS (OOB, SPC, etc.). For additional information, please refer to the Core Specification or the relevant sections of this White Paper.

Table 4.1: 3DS Data Elements Related to the Challenge Flow

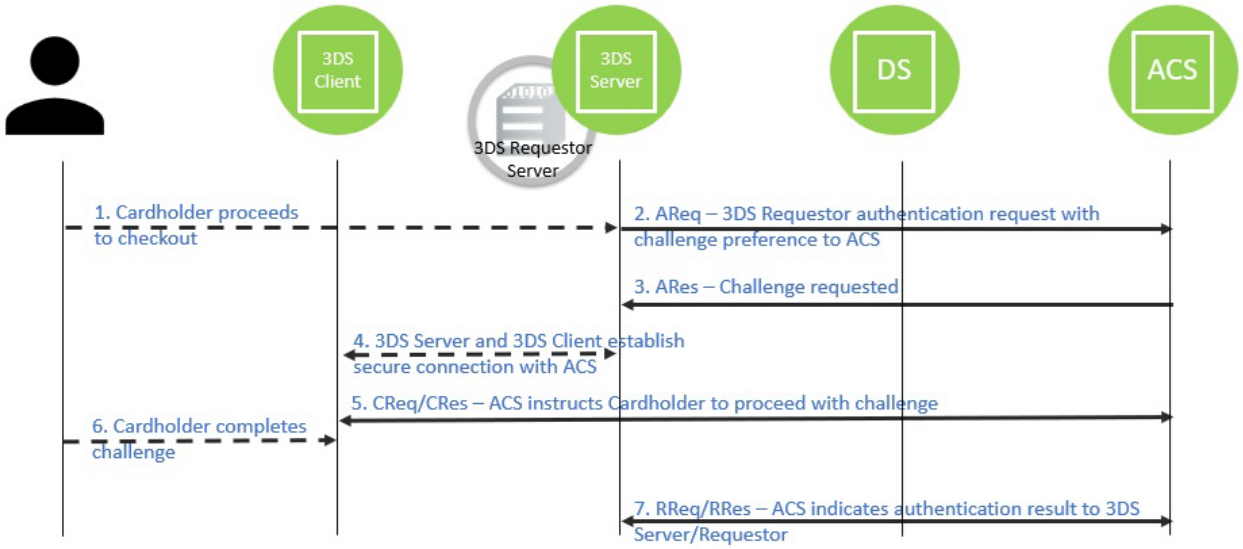
Data Element/ Field Name	Description	Version
3DS Requestor Challenge Indicator	Indicates whether a challenge is requested for this transaction.	2.3.1 2.2
3DS Requestor Decoupled Request Indicator	Indicates whether the 3DS Requestor requests the ACS to use Decoupled Authentication and agrees to use Decoupled Authentication if the ACS confirms its use.	2.3.1 2.2
ACS Challenge Mandate Indicator	Indication of whether a challenge is required for the transaction to be authorised due to local/regional mandates or other variable.	2.3.1 2.2
Transaction Status	Indicates whether a transaction qualifies as an authenticated transaction or account verification. Note: the ACS uses the Transaction Status to request a Challenge to the 3DS Server	2.3.1 2.2

Sequence Diagram

1. The Cardholder makes a purchase and proceeds to checkout.

2. The 3DS Server sends the AReq message indicating its authentication preference (Frictionless, Challenge etc.)
3. The ACS responds with an Authentication Response (ARes) message requesting a challenge.
4. The 3DS Server (or 3DS Requestor) accepts to proceed with the challenge (opens an iframe for a Browser flow or provides the relevant data to the 3DS SDK for an App-based flow).
5. The 3DS Client (or 3DS Server) establishes a secure connection with the ACS. The ACS then proceeds with the challenge and instructs the Cardholder how to complete the challenge.
6. The Cardholder provides the requested information to the ACS to complete the challenge. The ACS evaluates the responses and decides whether to continue or stop the challenge.
7. The ACS provides the outcome of the authentication in a Results Request (RReq) message.

Figure 4.1: Challenge Flow



4.3 WebAuthn and SPC

4.3.1 Business Overview

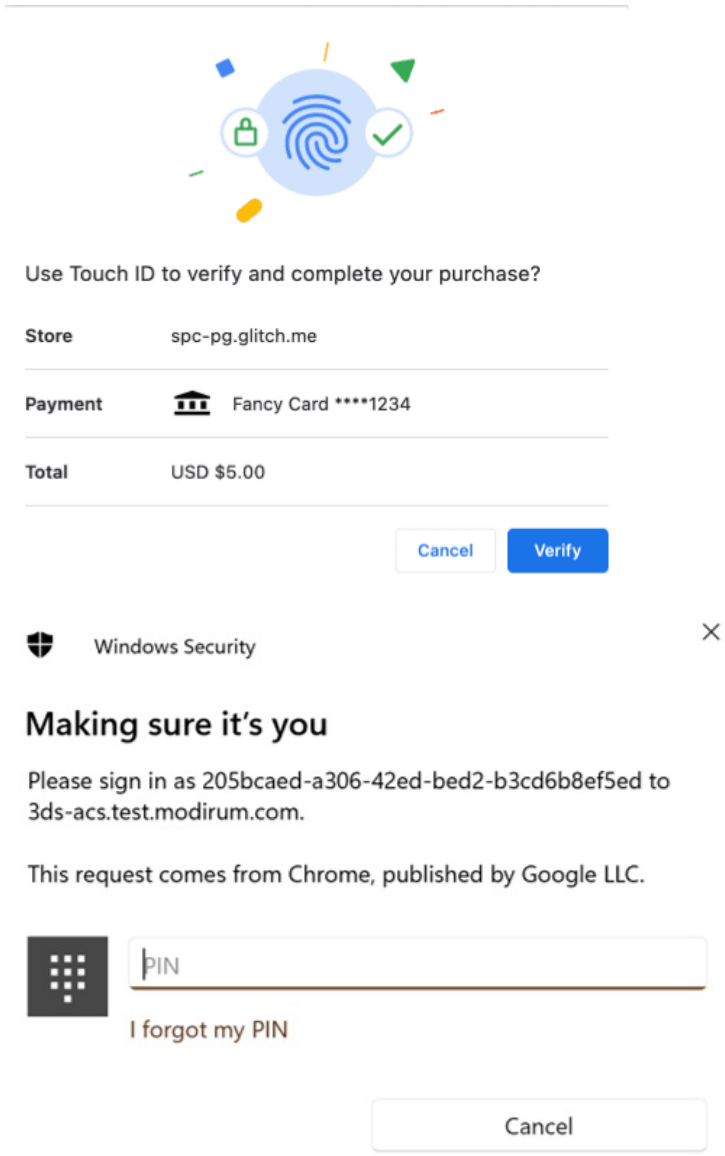
Secure Payment Confirmation (SPC) is a proposed web standard that allows customers to authenticate with their card issuer, bank, or other payment service provider using a platform authenticator:

- Touch ID / Face ID on a macOS device
- Windows Hello on a Windows device

SPC is designed to enable streamlined SCA for the purpose of completing online payment transactions. It enables a consistent authentication experience across websites/Merchants and

provides cryptographic evidence that the Cardholder has accepted the terms of the transaction. These terms include the Merchant details, the payment instrument, and the total amount of the transaction. When a Merchant or Issuer invokes the SPC API, the Browser displays the elements of the transaction in a secure modal window, and the Cardholder is asked to verify as illustrated below.

Figure 4.2: WebAuthn User Experience



Once the SPC Transaction Data has been verified by the Cardholder, the Cardholder is then prompted to authenticate using the Platform Authenticator integrated with their device.

There are two steps to use SPC – registration and authentication:

- **Registration:** the Cardholder links their device to a Relying Party. The Relying Party may be a payment network, bank, or other payment service provider.
- **Authentication:** the Cardholder uses the registered device to confirm their identity with the Relying Party directly from the Merchant's platform before confirming payments.

This White Paper focuses on the Authentication stage of SPC and assumes that registration of the Cardholder's credential has already occurred and is outside the scope of the Core Specification.

Benefits by Actor

- Merchant – enables seamless shopping while ensuring Cardholder privacy and maintaining the highest levels of security
- Issuer – allows Issuers to leverage FIDO standards for authentication approval
- Cardholder – provides a consistent authentication experience across different websites and platforms

4.3.2 Technical Features

Overview

SPC is built upon WebAuthn and designed specifically for payment purposes. As WebAuthn credentials are registered for specific domains, these credentials cannot be used to authenticate on unregistered sites that may be impersonating a Merchant. This feature makes WebAuthn effective against phishing attacks. SPC is available for browsers built using Chromium software (such as Microsoft Edge or Google Chrome).

SPC adds a payment information layer on top of WebAuthn enabling the card issuer or bank to provide a consistent payment experience. Once a payer registers an authenticator with the Relying Party, it can be used to authenticate on different Merchant sites. The Relying Party can also choose to use the payment credential as a regular WebAuthn credential.

Preconditions

The ACS has an enrolled FIDO Authenticator on the device for the Cardholder.

The 3DS Requestor and/or the ACS have detected that the Cardholder Browser supports the related SPC APIs (`allow="payment *; publickey-credentials-get *"`). For the ACS, this information can be obtained via the Browser User-Agent data element or via data obtained using the 3DS Method.

Table 4.2 below lists the data elements that may be provided in relation to SPC, whereas Table 4.3 lists the data elements that may be provided in relation to the SPC Transaction Data Object.

Table 4.2: 3DS Data Elements Related to Secure Payment Confirmation

Data Element/ Field Name	Description	Version
3DS Requestor Authentication Information	Information about how the 3DS Requestor authenticated the Cardholder before or during the transaction.	2.3.1
3DS Requestor Prior Transaction Authentication Information	Information about how the 3DS Requestor authenticated the Cardholder as part of a previous 3DS transaction.	2.3.1
3DS Requestor SPC Support	Indicate if the 3DS Requestor supports the SPC authentication. Note: If present, this field contains the value Y.	2.3.1

ACS Information Indicator	Provides additional information for a particular Protocol Version to the 3DS Server. The element lists all applicable values for the card range.
	Example:
	`{`
	`"acsInfoInd":["01","02","03","04",`
	`"05","06","07"]`
	`}`
		2.3.1

Authentication Method	Indicates the list of authentication types the Issuer will use to challenge the Cardholder, when in the ARes message or what was used by the ACS when in the RReq message. Note: For 03-3RI, only present for Decoupled Authentication.	2.3.1
SPC Transaction Data	Information that the 3DS Requestor passes in the SPC API for display in the Smart Modal Window.	2.3.1
Transaction Status	Indicates whether a transaction qualifies as an authenticated transaction or account verification. The Final CRes message can only contain a value of Y or N or D. Transaction Status = C or S is not allowed for Device Channel = 3RI.	2.3.1
WebAuthn Credential List	List of credential IDs registered for the Cardholder Account Number.	2.3.1

Table 4.3: 3DS Data Elements Related to the SPC Transaction Data Object

Data Element/ Field Name	Description	Version
Additional Data	For SPC API enhancement, to be defined in a future 3DS specification release	2.3.1

Data Element/ Field Name	Description	Version
Challenge	Random string generated by the ACS to prevent replay attacks.	2.3.1
Challenge Information Text	Text provided by the ACS to be displayed during the SPC authentication.	2.3.1
Currency	Transaction amount currency to be displayed during the SPC authentication	2.3.1
Display Name	Card or product name (Payment Instrument) to be displayed during the SPC authentication.	2.3.1
Icon	Card image (Payment Instrument) URL or Data URL to be displayed during the SPC authentication.	2.3.1

Issuer Image SPC	Issuer logo or Image URLs or Data URLs to be displayed during the SPC authentication.
	Includes at minimum the Default Image and at maximum the three Fully Qualified URLs or Data URLs defined as default, dark mode or monochrome images of the Issuer Image SPC.
	Default Image
	Field Name: `default`
	Dark Mode Image
	Field Name: `dark`
	Monochrome Image
	Field Name: `monochrome`
	Example Fully Qualified URL:
	`"issuerImageSpc":{ "default":"https://acs.com/defaultspcimage.png"}`
	Example Data URL:
	`"issuerImageSpc":{ "default":"..."`
		2.3.1

Payee Name	The display name of the payee that this SPC call is for (e.g., the Merchant). Matches the Merchant Name from the AReq message.	2.3.1
Payee Origin	The origin of the payee that this SPC call is for (e.g. the Merchant). Matches the Payee Origin from the AReq message.	2.3.1
Payment System Image SPC	Payment System logo or Image URLs to be displayed during the SPC authentication.	2.3.1

Data Element/ Field Name	Description	Version

	Includes at minimum the Default Image and at maximum the three Fully Qualified URLs defined as default, dark mode or monochrome images of the Payment System Image SPC.
	Default Image
	Field Name: `default`
	Dark Mode Image
	Field Name: `dark`
	Monochrome Image
	Field Name: `monochrome`
	Example Fully Qualified URL:
	`"psImageSpc":{ "default":"https://ds.com/defaultspcimage.png"}`
	Example Data URL:
	`"psImageSpc":{ "default":" c2RzYWRhc2Q..."`	

Timeout	The number of milliseconds before the request to sign the transaction details times out.	2.3.1
Value	Transaction amount as a decimal value to be displayed during the SPC authentication.	2.3.1
WebAuthn SPC Extension Indicator	For SPC and WebAuthn API enhancement.	2.3.1

4.3.3 Merchant-Initiated SPC Flow

Overview

The SPC authentication can be initiated by the 3DS Requestor via an extra AReq/ARes message pair instead of the standard browser challenge flow.

Sequence Diagram

1. The Cardholder interacts with the 3DS Requestor using a browser on their device.
2. The 3DS Requestor initiates communication with the 3DS Server and provides the necessary 3DS data to the 3DS Server to start the Cardholder authentication.
3. The 3DS Server sends the Authentication Request (AReq) and indicates that the 3DS Requestor can support SPC authentication for the transaction and that the Browser supports the SPC API.
    a. Support is indicated by setting the 3DS Requestor SPC Support = Y.

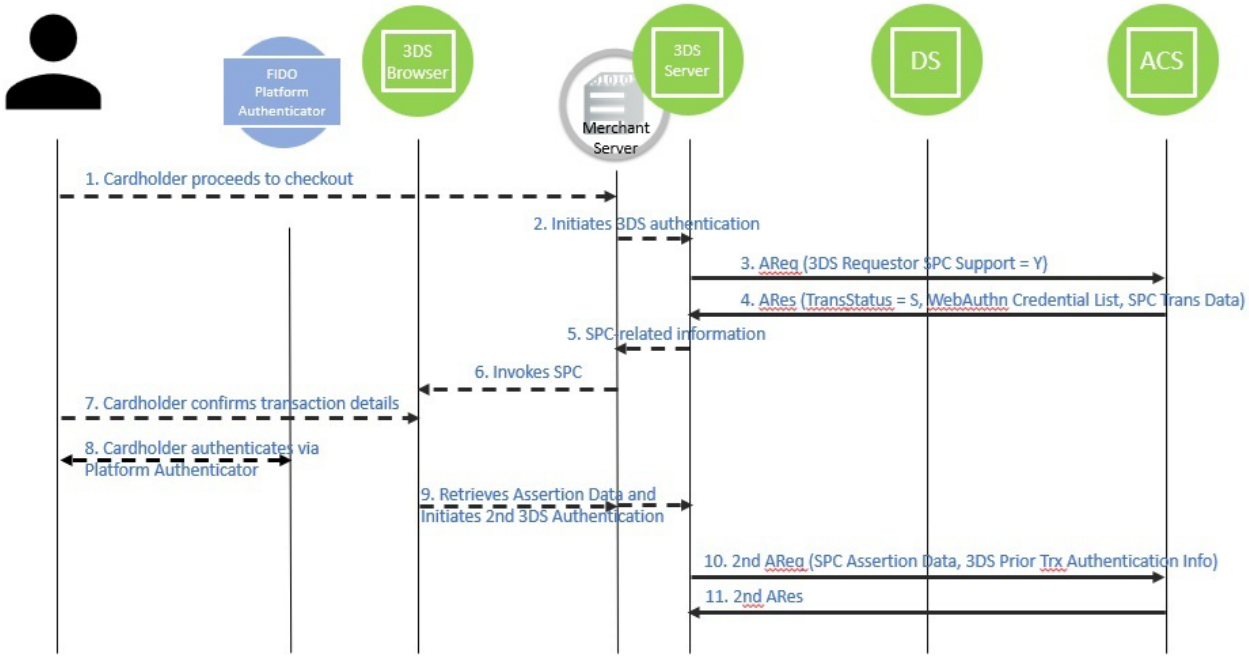
4. The ACS receives the AReq and recognises that SPC-based authentication is supported. It determines that SPC is the selected authentication method and returns an Authentication Response (ARes) containing:
    a. Transaction Status = S
    b. WebAuthn Credential List: The list of enrolled FIDO credentials associated with the Cardholder.
    c. SPC Transaction Data containing:
        i. Challenge data: Random string generated by the ACS to prevent replay attacks.
        ii. Challenge Information Text: Text provided by the ACS to be displayed during the SPC authentication.
        iii. Currency: Transaction amount currency to be displayed during the SPC authentication.
        iv. Display Name: Card or product name (Payment Instrument) to be displayed during the SPC authentication.
        v. Icon: Card image URL or Data URL to be displayed during the SPC authentication.
        vi. Issuer Image SPC: Issuer logo or Image URLs or Data URLs to be displayed during the SPC authentication.
        vii. Payee Name: The display name of the payee that the SPC call is for (the Merchant).
        viii. Payee Origin: The origin of the payee that the SPC call is for (the Merchant)
        ix. Payment System Image SPC: The Payment System logo or Image URLs to be displayed during the SPC authentication.
        x. Timeout: The number of milliseconds before the request to sign the transaction details times out.
        xi. Value: Transaction amount as a decimal value to be displayed during the SPC authentication.
        xii. WebAuthn SPC Extension Indicator: For SPC and WebAuthn API enhancement.
5. The 3DS Server sends the necessary information from the ARes message to the 3DS Requestor, in particular, the WebAuthn Credential List, and the SPC Transaction Data.
6. The 3DS Requestor invokes the SPC API against the WebAuthn Credential List returned in the ARes and provides the SPC Transaction Data as an input.
7. The Cardholder confirms the transaction details.
8. The Cardholder authenticates using the FIDO Authenticator on their device.
9. The 3DS Requestor initiates a second 3DS Authentication Request with the 3DS Server and provides the SPC Assertion Data output from the SPC API to the 3DS Server.

10. The 3DS Server sends the second AReq message containing:
    a. The FIDO Assertion Data via the 3DS Requestor Authentication Data object using a 3DS Requestor Authentication Method = 09.
    b. A new 3DS Server Transaction ID.
    c. The 3DS Requestor Prior Transaction Authentication Information object, which includes:
        i. 3DS Requestor Prior Transaction Reference = ACS Transaction ID from the prior ARes message indicating that SPC authentication is to be performed.
        ii. 3DS Prior Transaction Authentication Method = 05 (SPC authentication).
        iii. 3DS Requestor Prior Transaction Authentication Timestamp.
11. The ACS receives the second AReq and determines the disposition of the transaction by evaluating the Assertion Data and responds with an ARes message. The ACS does its evaluation by verifying:
    a. The signature in the Assertion Data
    b. The consistency of the transaction data between the first and second AReq messages
    c. The consistency of the Assertion Data with the data from the AReq message.

Note: It is expected that, if the Assertion Data is verified correctly, no further challenge is needed and the 3DS Server will then receive an ARes message with Transaction Status = Y. However, the ACS is able to respond with any applicable Transaction Status, including Transaction Status = C, if the ACS determines that an additional challenge is necessary.

Note: The DS may act as the FIDO Relying Party and perform some or all the actions described for the ACS within the SPC flow.

Figure 4.3: Merchant-Initiated SPC Flow



User Experience

Figure 4.4: Checkout Page

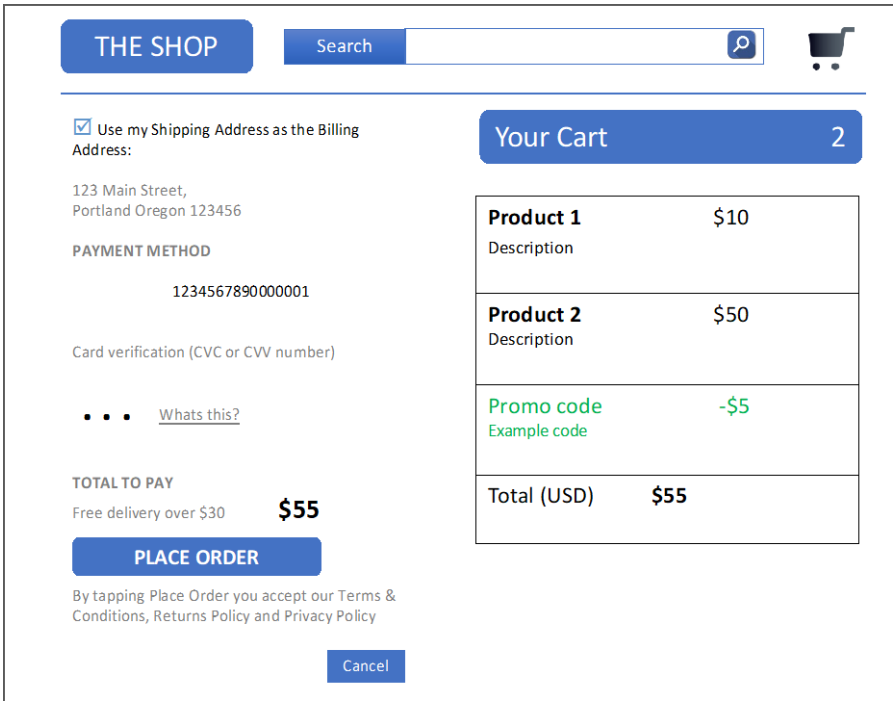


Figure 4.5: First AReq/ARes

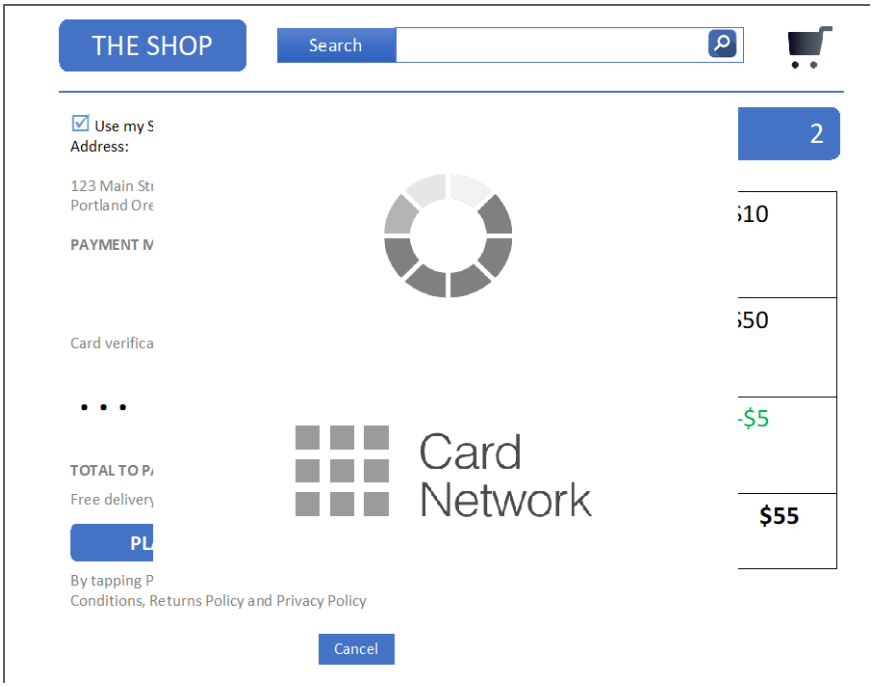


Figure 4.6: SPC Modal Window

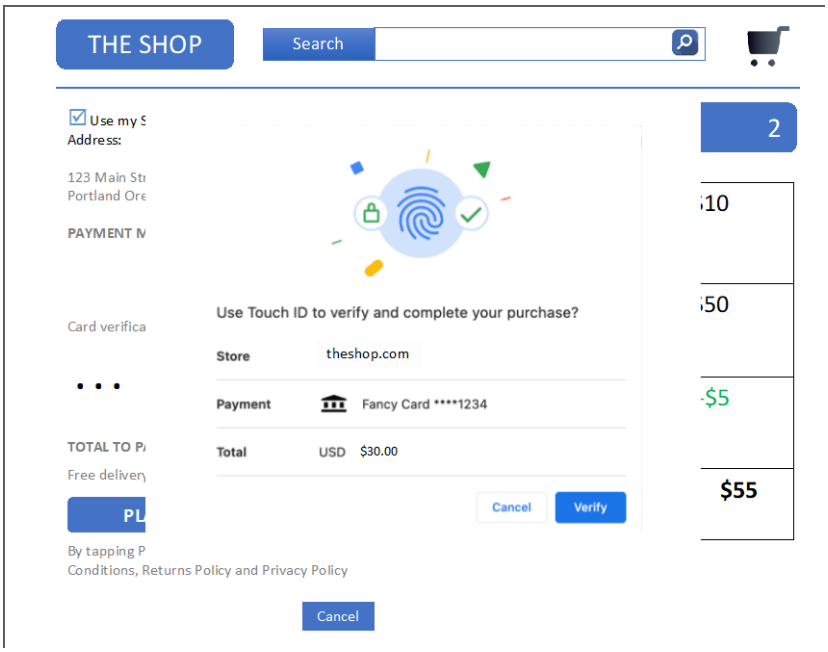


Figure 4.7: FIDO Authentication

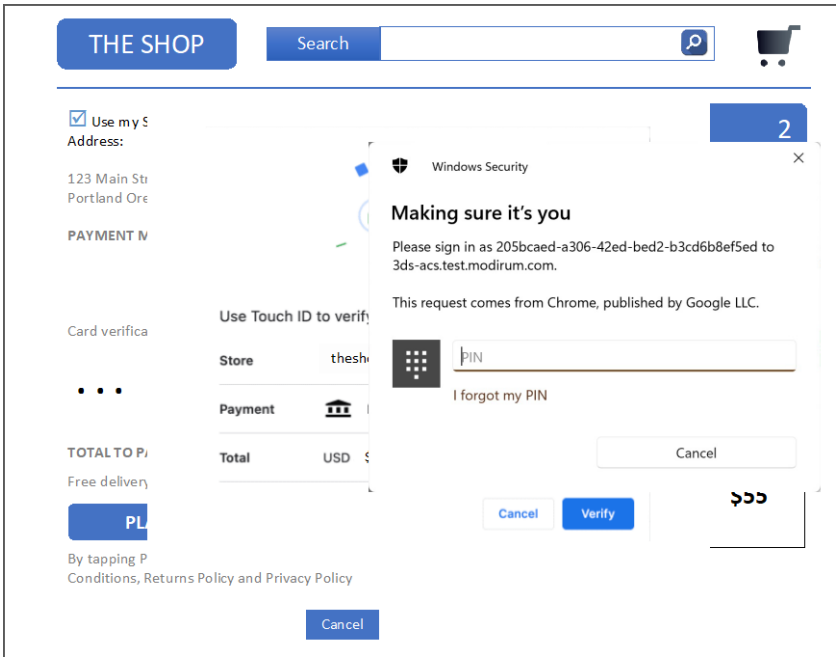


Figure 4.8: Merchant Provides Outcome of SPC Authentication to ACS – Processing Screen

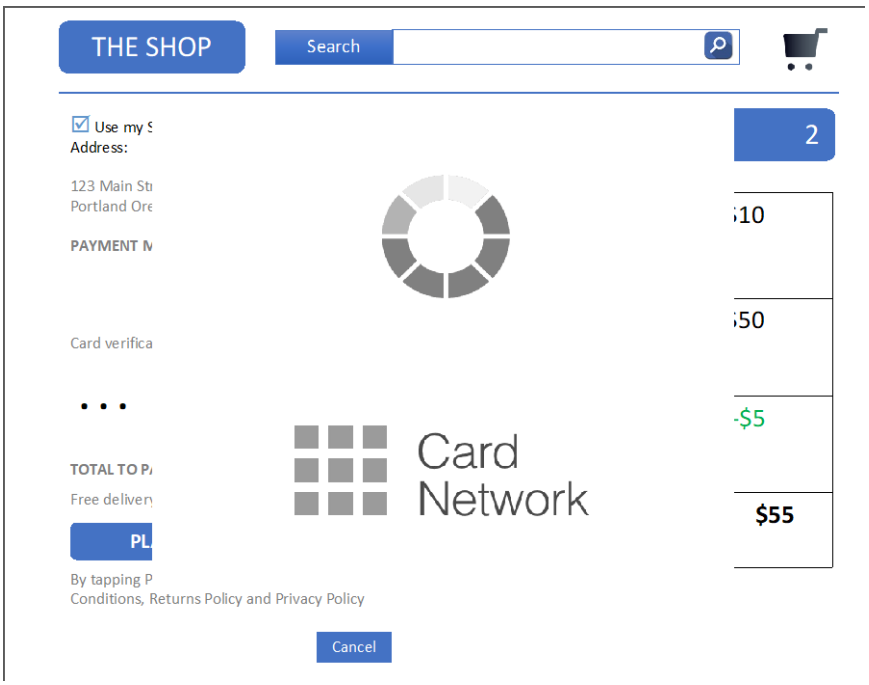
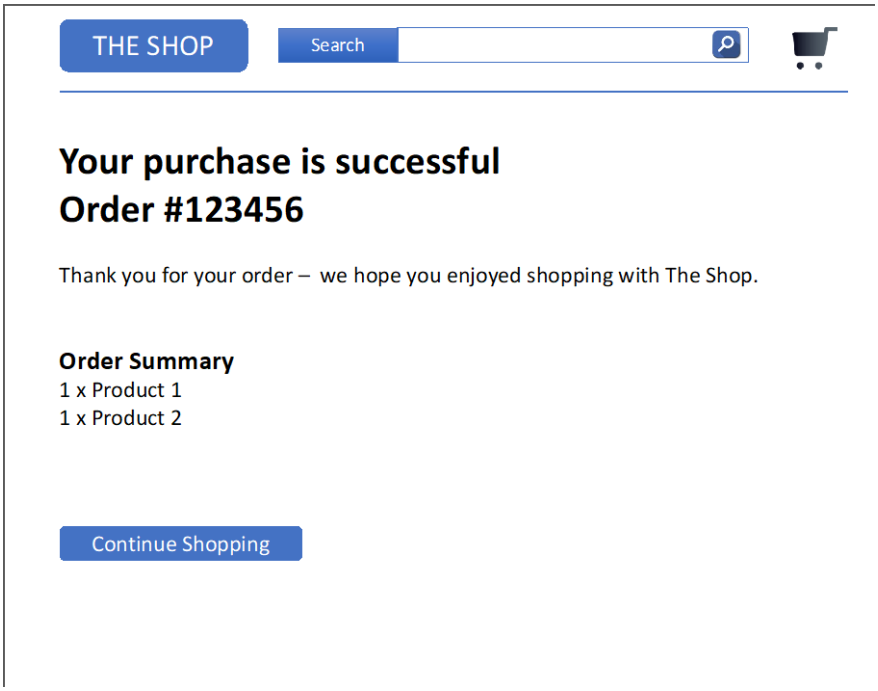


Figure 4.9: SPC Authentication Successful, Purchase Complete



4.3.4 Issuer-Initiated SPC Flow

Overview

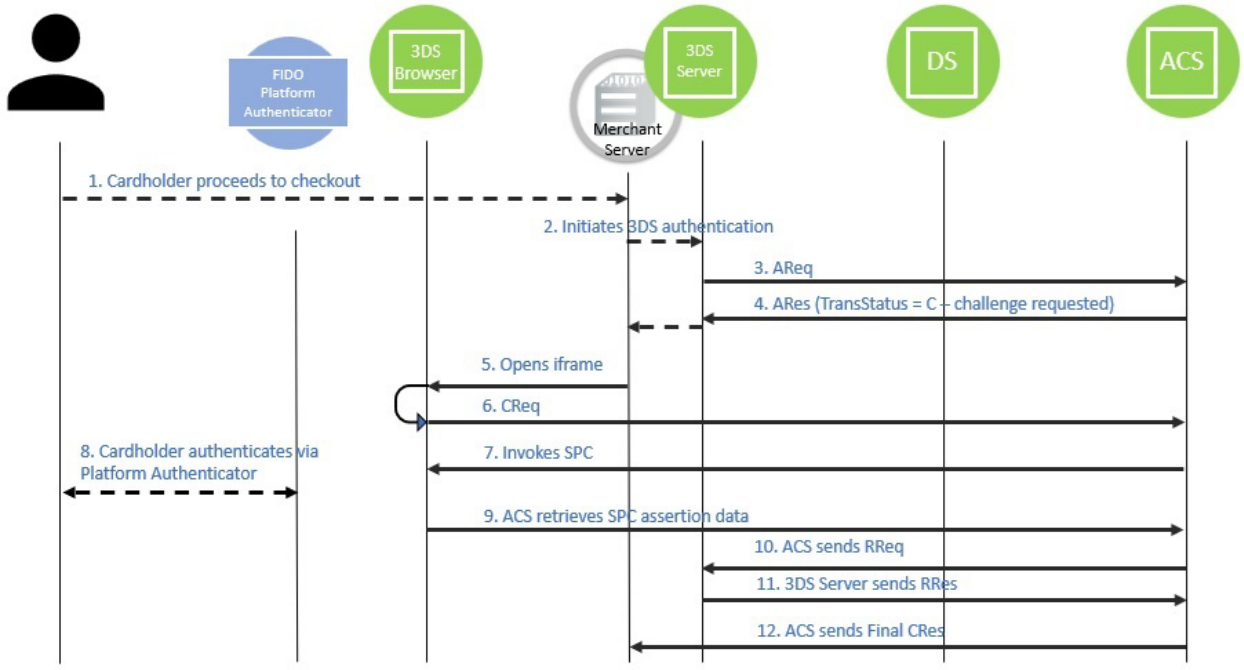
When the ACS initiates and performs an SPC authentication as part of a challenge, the steps are identical to a standard Browser flow, where SPC authentication is used instead of the other 3DS challenge methods. This section outlines some of the details and values for specific steps when an ACS performs an SPC authentication as part of the Challenge Flow.

Sequence Diagram

1. The Cardholder interacts with the 3DS Requestor using a Browser on a consumer device.
2. The 3DS Requestor initiates communications with the 3DS Server and provides the necessary 3DS data to the 3DS Server to initiate Cardholder authentication.
3. The 3DS Server sends the Authentication Request (AReq).
4. The ACS receives the AReq and recognises that there is a pre-registered FIDO Authenticator on the device for the Cardholder. It determines that SPC will be selected as the authentication method and returns the 3DS Server an Authentication Response (ARes) containing:
   a. Transaction Status = C
   b. Authentication Method = 14 (SPC)

5. The 3DS Requestor accepts the request for challenge and posts the CReq message using an HTTP POST through the Cardholder Browser HTML iframe to the ACS URL received in the ARes message.
6. The ACS receives and processes the Challenge Request (CReq) message from the Browser.
7. The ACS invokes the SPC authentication (SPC API) against the credentials registered for the Cardholder and the device using, for example, a JavaScript.
8. The Cardholder authenticates using the FIDO Authentication on their device (for example, using Windows Hello or Apple Touch ID).
9. The ACS retrieves, evaluates, and verifies the Assertion Data from the SPC Browser API.
10. The ACS determines the challenge outcome and sends the Results Request (RReq) message to the 3DS Server through the DS.
11. The 3DS Server receives and processes the RReq message, and responds with a Results Response (RRes) message to the ACS.
12. The ACS receives the RRes message and responds with the Final Challenge Response (CRes) message.

Figure 4.10: SPC Issuer-Initiated Flow



User Experience

Figure 4.11: Checkout Page

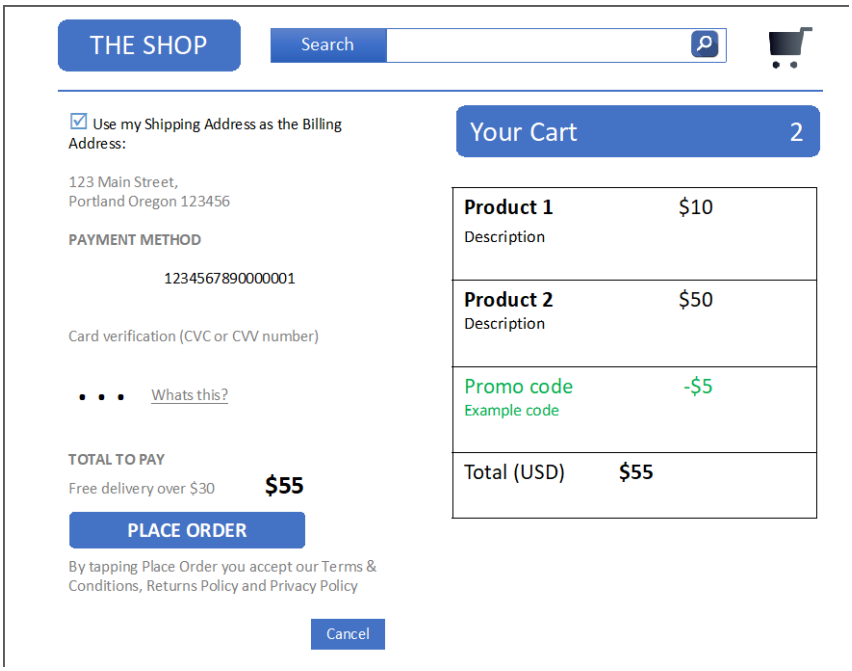


Figure 4.12: First AReq/ARes Processing Screen

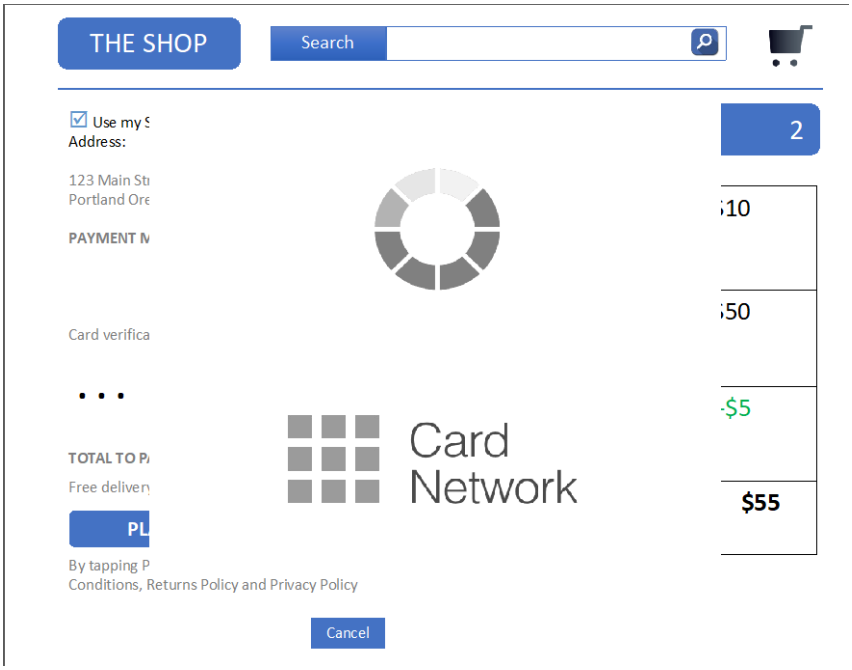


Figure 4.13: Merchant Opens iframe, ACS Provides UI

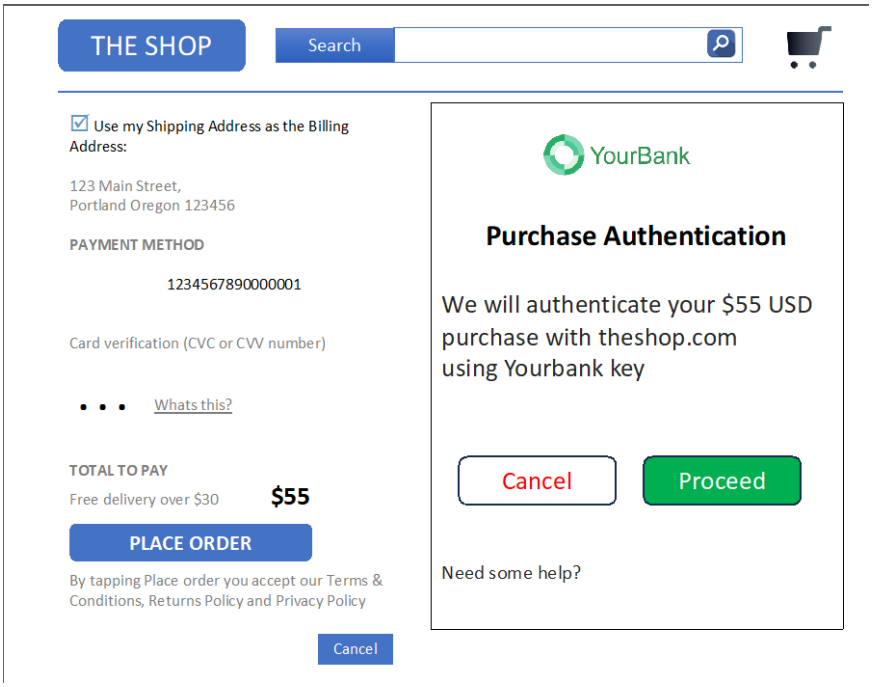


Figure 4.14: SPC Modal Window

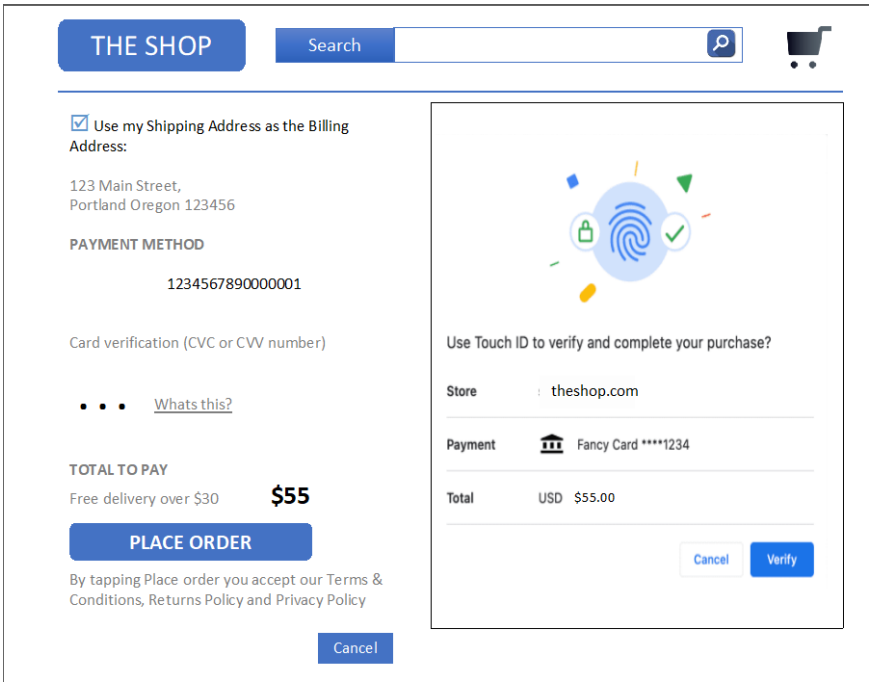


Figure 4.15: Authenticator

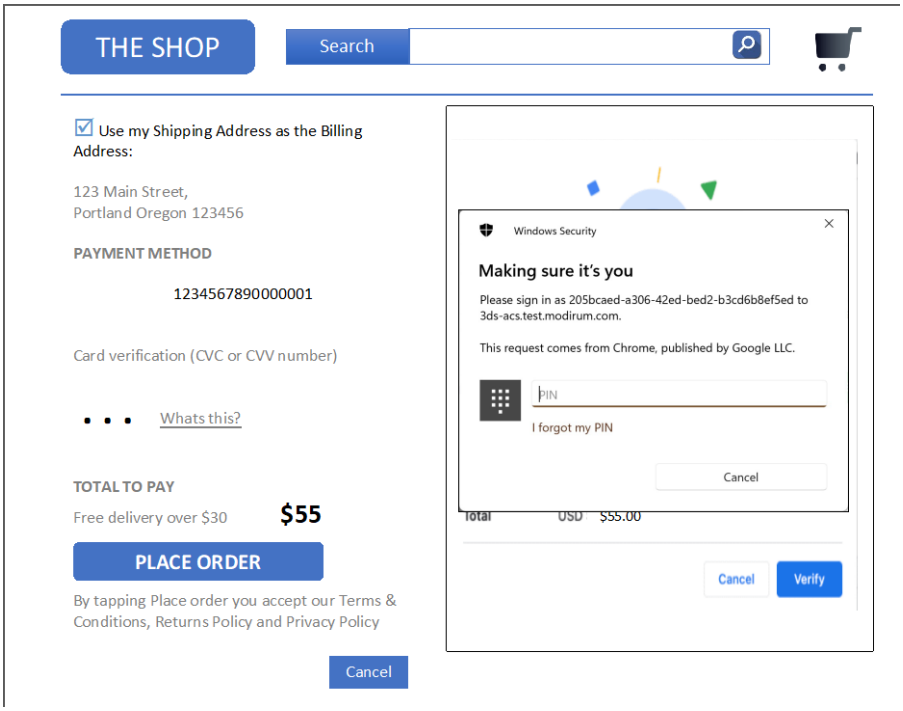


Figure 4.16: SPC Authentication Successful

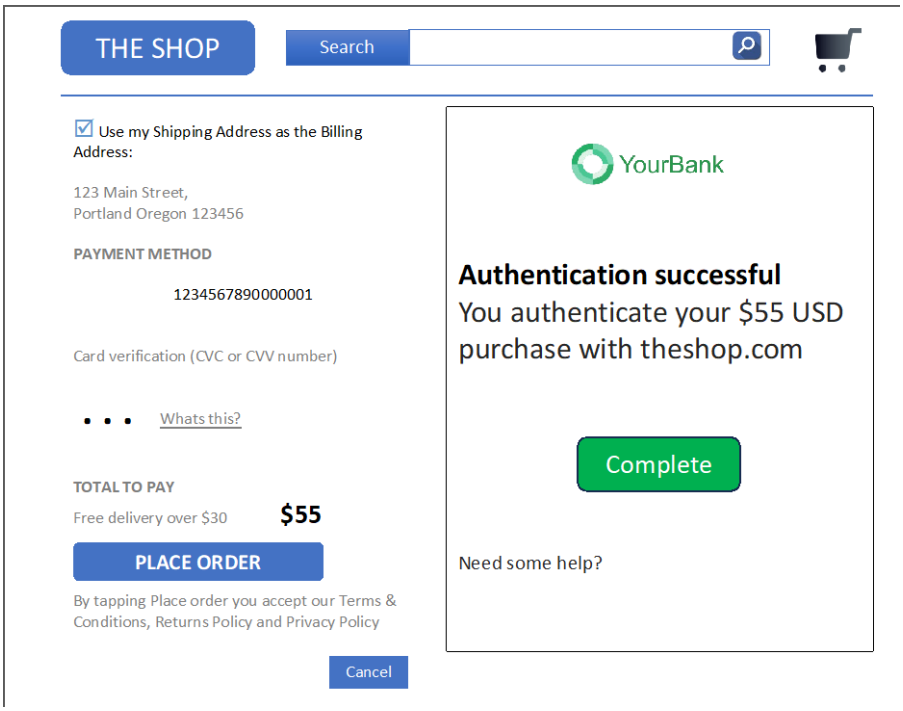
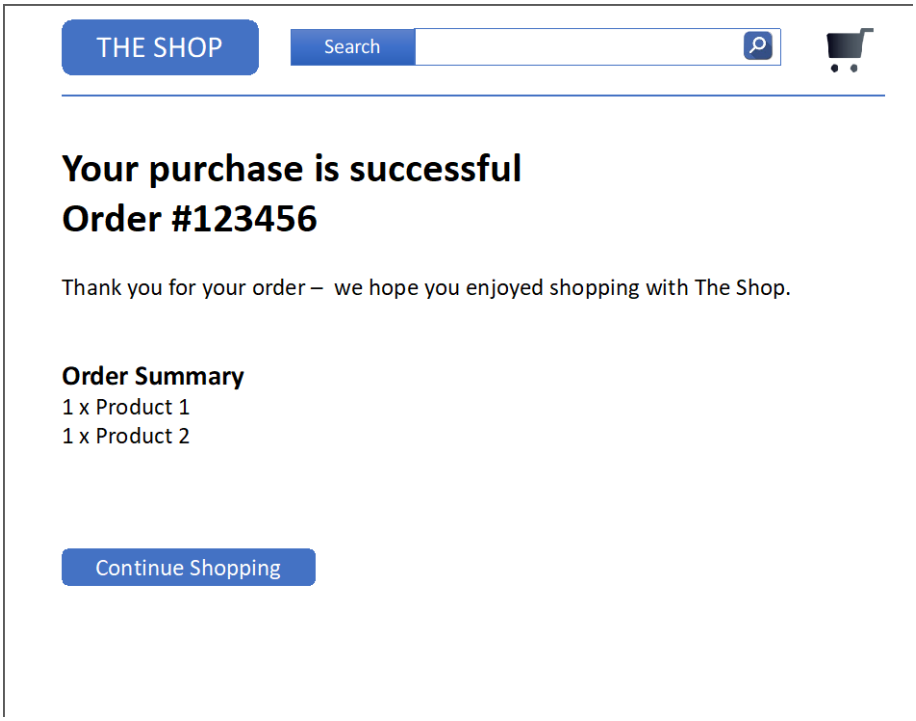


Figure 4.17: Merchant Purchase Complete



4.4 Decoupled Authentication

4.4.1 Business Overview

Decoupled authentication is a 3DS feature that allows for an alternative authentication method when the primary authentication method (i.e., challenge) is not possible, available or fails.

In the 3DS protocol, the primary authentication method is typically the challenge flow where the Cardholder is directed to their card issuer (ACS) to complete the authentication process. However, there may be situations where the challenge is not possible or successful, for example, when:

- the Cardholder's device or browser does not support the required technology (for example, JavaScript not supported, voice assistant) for redirect-based authentication;
- the ACS authentication is temporarily unavailable or experiencing issues;
- the Cardholder is unable to complete the authentication process for some reason (for example, poor internet connection, technical difficulties); or
- the Cardholder is not present, and the Merchant initiates the authentication process with the card issuer, for example, in Mail Order/Telephone Order (MOTO) transactions.

In such cases, the Core Specification allows for a Decoupled Authentication, where the authentication process is performed separately from the payment transaction flow. While the authentication method used for Decoupled Authentication is outside the scope of this White Paper, example methods could include a text message, an email, a phone call, or a push

notification to a banking app that completes authentication, and then sends the results to the ACS.

Typically, this involves the following steps:

1. The Merchant initiates the 3DS authentication process as usual, but the Issuer recognises that the primary authentication method is not available or has failed.

2. The Issuer then triggers an alternative authentication method, such as prompting the Cardholder to authenticate using a mobile banking app or another secure channel.

3. The Cardholder completes the alternative authentication process

4. The Issuer provides the Merchant with an Authentication Result.

Decoupled Authentication ensures that the payment transaction can still be completed, even if the primary 3DS authentication method is not available or fails. Decoupled Authentication is applicable to all Device Channels, but it is the only authentication method available to facilitate Cardholder challenges for 3RI transactions. This helps to improve the overall user experience and reduce the risk of abandoned transactions.

Note: Although Decoupled Authentication is a 3DS Challenge method, the flow may differ from the general Challenge Flow as the CReq/CRes messages are not always present.

Benefits by Actor

- 3DS Requestor – authentication is performed after the checkout process
- ACS
  - authentication is possible, also when the device does not support the 3DS challenge (for example, a voice assistant)
  - authentication can use a different channel in case of a fraudulent transaction or compromised device

3DS Data Elements Related to Decoupled Authentication

Table 4.4 lists the data elements that may be provided by 3DS Servers and ACSs to support Decoupled Authentication.

For additional information, refer to Table A.1 in the Core Specification and to the EMV 3-D Secure Bridging Message Extension.

Table 4.4: 3DS Data Elements Related to Decoupled Authentication

Data Element	Description	Version
3DS Requestor Decoupled Request Indicator	Indicates whether the 3DSRequestor requests the ACS to use Decoupled Authentication and agrees to use Decoupled Authentication if the ACS confirms its use.	2.3.1.1 2.2

Data Element	Description	Version
3DS Requestor Decoupled Max Time	Indicates the maximum amount of time that the 3DS Requestor will wait for an ACS to provide the results of a Decoupled Authentication transaction (in minutes).	2.3.1.1 2.2
3DS Requestor Decoupled Request Indicator	Indicates whether the 3DS Requestor requests the ACS to use Decoupled Authentication and agrees to use Decoupled Authentication if the ACS confirms its use. Note: if the element is not provided, the expected action is for the ACS to interpret as N (Do not use Decoupled Authentication).	2.3.1.1 2.2
3DS Requestor Prior Transaction Authentication Information	Information about how the 3DS Requestor authenticated the Cardholder as part of a previous 3DS transaction. Required for 3RI in the case of Decoupled Authentication Fallback or for SPC.	2.3.1.1 2.2
3RI Indicator	Indicates the type of 3RI request. This data element provides additional information to the ACS to determine the best approach for handling a 3RI request.	2.3.1.1 2.2
ACS Decoupled Confirmation Indicator	Indicates whether the ACS confirms use of Decoupled Authentication and agrees to use Decoupled Authentication to authenticate the Cardholder.	2.3.1.1 2.2
Authentication Method	Indicates the list of authentication types the Issuer will use to challenge the Cardholder, when in the ARes message or what was used by the ACS when in the RReq message. Note: For 03-3RI, only present for Decoupled Authentication.	2.3.1.1 2.2
Card Range Data – ACS Information Indicator	Provides additional information for a particular Protocol Version to the 3DS Server. The element lists all applicable values for the card range.	2.3.1.1 2.2
Cardholder Information Text	Text provided by the ACS/Issuer to Cardholder during a Frictionless or Decoupled transaction. The Issuer can provide information to Cardholder. For example, “Additional authentication is needed for this transaction, please contact (Issuer Name) at xxx-xxx-xxxx” with optionally the Issuer and Payment System images.	2.3.1.1 2.2

Data Element	Description	Version
	Refer to Section A.20 in the *Core Specification* for UI example.	
Challenge Cancelation Indicator	Indicator informing the ACS and the DS that the authentication has been cancelled.	2.3.1.1 2.2
Results Message Status	Indicates the status of the Results Request message from the 3DS Server to provide additional data to the ACS. This will indicate if the message was successfully received for further processing or will be used to provide more detail on why the Challenge could not be completed from the 3DS Client to the ACS.	2.3.1.1 2.2
Transaction Status	Indicates whether a transaction qualifies as an authenticated transaction or account verification. The Final CRes message can only contain a value of Y or N or D. Transaction Status = C or S is not allowed for Device Channel = 3RI.	2.3.1.1 2.2
Transaction Status Reason	Provides information on why the Transaction Status field has the specified value.	2.3.1.1 2.2

4.4.2 3DS Requestor-Initiated Flow (3RI Device Channel)

Overview

The 3DS Requestor needs to authenticate a transaction with the Issuer, but the Cardholder is not present. This may happen in the case of:

- Mail Order/Telephone Order (MOTO) transactions
- Recurring transactions/subscriptions
- Split/delayed shipments
- Flash sales.

To determine if the ACS supports Decoupled Authentication as an authentication method, the 3DS Server should refer to the ACS Information Indicator data element.

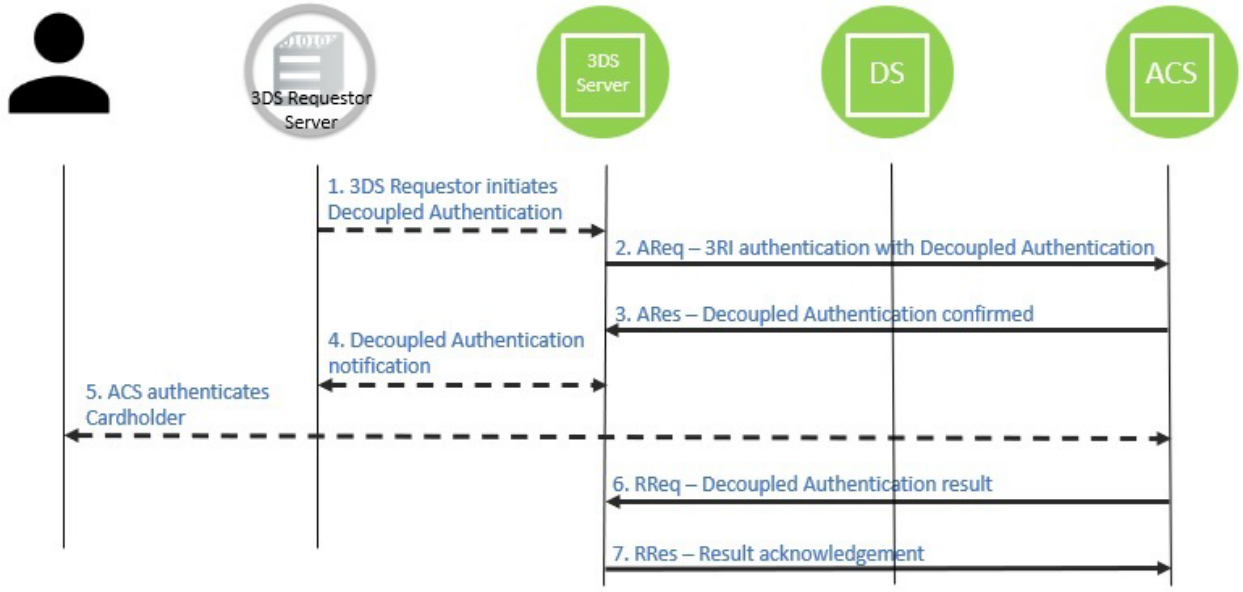
Sequence Diagram

1. Along with any details about the transaction, 3DS Requestor sets its preferences for Decoupled Authentication using the following elements:
   a. 3DS Requestor Decoupled Max Time
   b. 3DS Requestor Decoupled Request Indicator
   c. 3DS Requestor Prior Transaction Authentication Information

d. 3RI Indicator

2. The 3DS Server sends a 3RI Authentication Request (AReq).
3. The ACS evaluates the transaction and determines if a Cardholder challenge is required and determines if Decoupled Authentication is appropriate given the conditions provided by the 3DS Requestor. The ACS specifies its intent to use Decoupled Authentication as the authentication method in the AReq message using the following elements:
   a. ACS Decoupled Confirmation Indicator
   b. Authentication Method
   c. Cardholder Information Text
   d. Transaction Status
4. The 3DS Server may notify the 3DS Requestor that Decoupled Authentication will be used as the authentication method.
5. The ACS authenticates the Cardholder using a method that is separate from the 3DS Challenge Flow.
6. The ACS provides the results of the authentication in an RReq message.
7. The 3DS Server receives the authentication results and sends an RRes message.

Figure 4.18: 3DS Requestor-Initiated Flow



4.4.3 Decoupled Authentication as a Challenge Method

Overview

The 3DS Requestor initiates an authentication with the card issuer, the ACS assesses the risk associated with the transaction and selects Decoupled Authentication as challenge method. This may happen if:

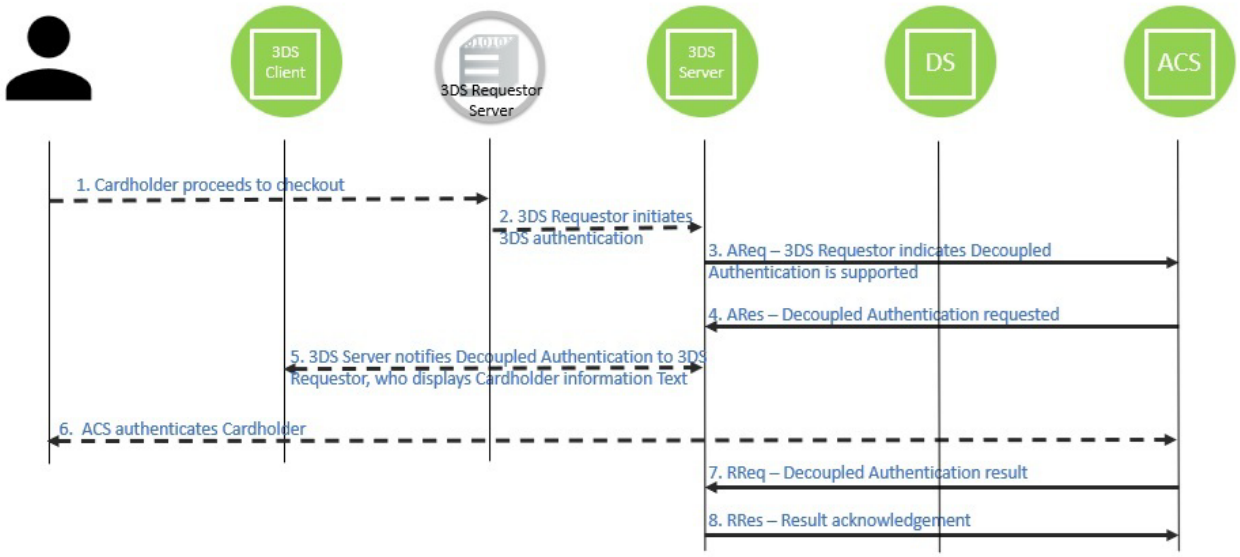
- The ACS knows that it is not possible to use the usual authentication method or to reach the Cardholder.
- The ACS has assessed that the authentication was initiated from an untrusted or stolen device.

The ACS verifies that Decoupled Authentication is supported by the 3DS Server by checking the 3DS Requestor Decoupled Request Indicator.

Sequence Diagram

1. The Cardholder interacts with the 3DS Requestor website using a Browser or the 3DS Requestor App on a Consumer Device.
2. The 3DS Requestor initiates communications with the 3DS Server and provides the necessary 3DS data to the 3DS Server to initiate Cardholder authentication.
   Along with any details about the transaction, the 3DS Requestor sets its preferences for Decoupled Authentication using the following elements:
   a. 3DS Requestor Decoupled Max Time
   b. 3DS Requestor Decoupled Request Indicator
   c. 3DS Requestor Prior Transaction Authentication Information
3. The 3DS Server sends an Authentication Request (AReq).
4. The ACS evaluates the transaction and determines if a Cardholder challenge is required and determines if Decoupled Authentication is appropriate given the conditions provided by the 3DS Requestor. The ACS specifies its intent to use Decoupled Authentication in the ARes message as the authentication method by using the following elements:
   a. ACS Decoupled Confirmation Indicator
   b. Authentication Method
   c. Cardholder Information Text
   d. Transaction Status
5. 3DS Server may notify the 3DS Requestor that Decoupled Authentication will be used as authentication method, and pass the Cardholder Information Text for display to the Cardholder.
6. The ACS authenticates the Cardholder using a method that is separate from the 3DS Challenge Flow.
7. The ACS provides the results of the authentication in the RReq message.
8. The 3DS Server receives the authentication results and sends an RRes message.

Figure 4.19: Decoupled Authentication as a Challenge Method



4.4.4 Decoupled Authentication Fallback

Overview

During a challenge, the Cardholder or the ACS may experience technical issues that prevent its normal completion. The ACS has the option to invoke Decoupled Authentication as an alternative way to complete the challenge, as long as it is supported by the 3DS Server and the 3DS Requestor.

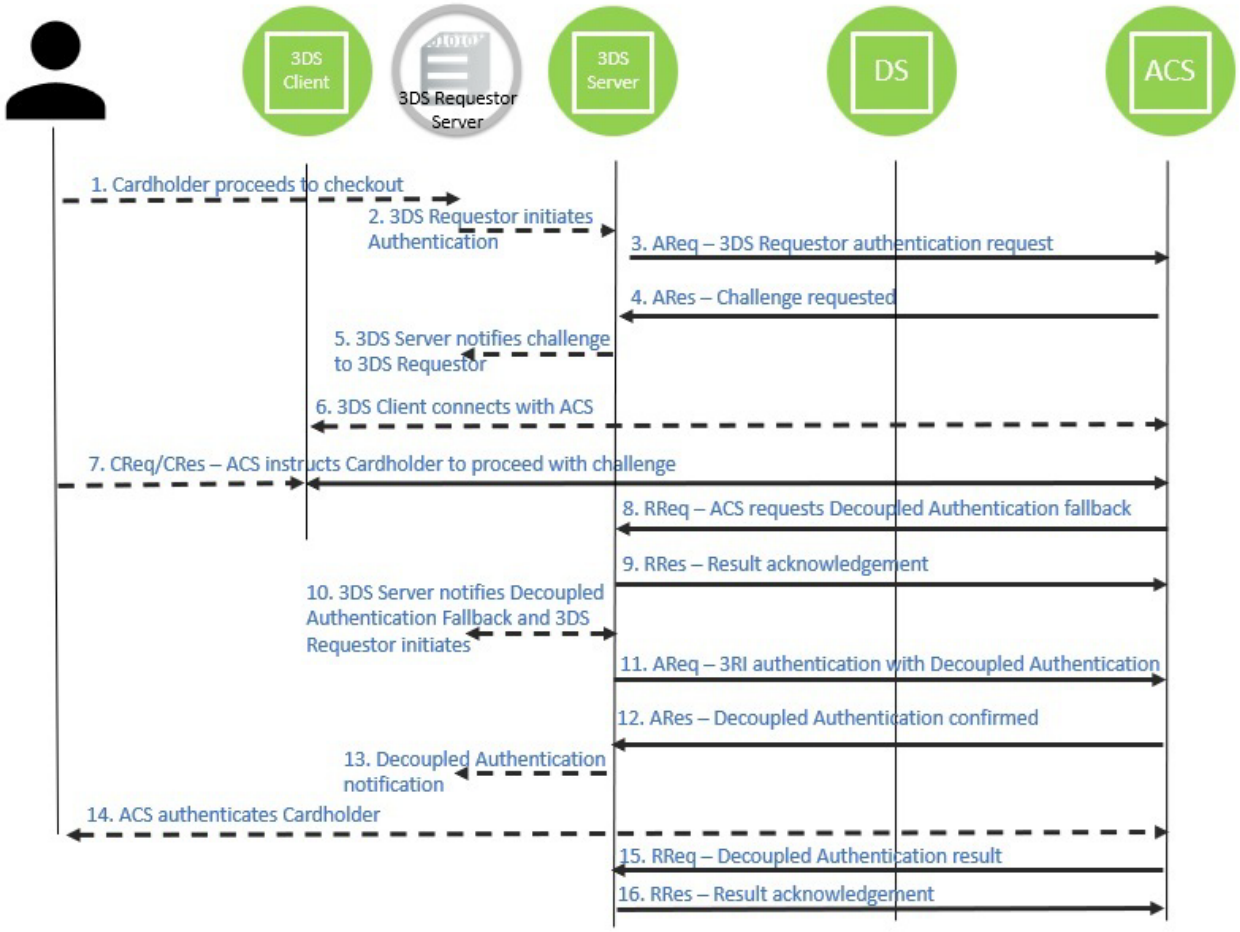
The Decoupled Authentication Fallback flow is identical to the Decoupled Authentication flow initiated by the Merchant, except it starts after the challenge when the ACS sends Transaction Status = D to the 3DS Requestor in an RReq message. The 3DS Server confirms its support in an RRes message (Result Message Status = 04), and then initiates a 3RI transaction with Decoupled Authentication as the challenge method.

Sequence Diagram

1. The Cardholder interacts with the 3DS Requestor website using a Browser or the 3DS Requestor App on a Consumer Device.
2. The 3DS Requestor initiates an authentication with the 3DS Server.
3. The 3DS Server sends the AReq message indicating its authentication preference (for example, Frictionless or Challenge).
4. The ACS responds with an Authentication Response (ARes) message requesting a challenge.
5. The 3DS Server (or 3DS Requestor) accepts to proceed with the challenge (opens an iframe for a Browser-based flow or provides the relevant data to the 3DS SDK for an App-based flow).

6. The 3DS Client (or 3DS Server) establishes a secure connection with the ACS. Then the ACS proceeds with the challenge and provides instructions to the Cardholder on to how to complete the challenge.
7. The Cardholder provides the requested information to the ACS to complete the challenge. The ACS evaluates the responses and decides to request Decoupled Authentication Fallback.
8. The ACS provides the request for a Decoupled Authentication Fallback in a Results Request (RReq) message (Transaction Status =D), the ACS may use the Cardholder Information Text to notify the Cardholder of the Decoupled Authentication.
9. The 3DS Server acknowledges the Decoupled Authentication Fallback setting Result Message Status = 04 and sends an RRes message.
10. The 3DS Server notifies the Decoupled Authentication Fallback to the 3DS Requestor, the 3DS Requestor sets its preferences for Decoupled Authentication using the following elements:
    a. 3DS Requestor Decoupled Max Time
    b. 3DS Requestor Decoupled Request Indicator
    c. 3DS Requestor Prior Transaction Authentication Information
    d. 3RI Indicator
11. The 3DS Server sends a 3RI Authentication Request (AReq).
12. The ACS evaluates the transaction information, and ACS specifies its intent to use Decoupled Authentication as the authentication method in the ARes message using the following elements:
    a. ACS Decoupled Confirmation Indicator
    b. Authentication Method
    c. Cardholder Information Text
    d. Transaction Status
13. The 3DS Server notifies the 3DS Requestor that Decoupled Authentication will be used as the authentication method.
14. The ACS authenticates the Cardholder using a method that is separate from the 3DS Challenge Flow.
15. The ACS provides the results of the authentication in an RReq message.
16. The 3DS Server receives the authentication results and sends an RRes message.

Figure 4.20: Decoupled Authentication Fallback



4.5 Use of the Challenge Error Reporting Data Element

In a version 2.2 3DS authentication, there is limited visibility into errors that may occur during an App-based challenge between the ACS and the 3DS SDK. When an error message is received from the 3DS SDK or a CReq message is in error, the ACS uses the Challenge Cancelation Indicator (06 = Transaction Error) in the RReq message to report the issue. However, these errors can arise for various reasons, such as missing data elements, incorrect values, or cryptographic errors, which makes them difficult to identify and resolve due to the lack of detailed information.

To address this, two new values were added in the Core Specification v2.3.1.1 to the Challenge Cancelation Indicator to better describe when the CReq or CRes message is in error. Additionally, the ACS now provides a copy of the Error Message exchanged with the 3DS SDK in the RReq message under a new data field called Challenge Error Reporting.

For a v2.2 3DS authentication, these errors can be reported in Additional Data using the Bridging Message Extension.

Table 4.5 below lists the data elements that may be provided in relation to Challenge Error Reporting.

Table 4.5: 3DS Data Elements Related to Challenge Error Reporting

Data Element	Description	Version
Challenge Cancelation Indicator	Indicator informing the ACS and the DS that the authentication has been cancelled.	2.3.1.1 2.2 + Bridging Message Extension
Challenge Error Reporting	Copy of the Erro Message sent or received by the ACS in case of error in the CReq/CRes messages.	2.3.1.1 2.2 + Bridging Message Extension

4.6 Challenge Autofill

The autofill feature on mobile devices allows the device operating system to automatically fill in forms, login credentials, and other information on websites and apps. It is designed to save the user's time and effort by pre-populating fields with previously entered or saved information.

When autofill is enabled on a mobile device, it can store and retrieve various types of information, such as:

- Names and addresses
- Phone numbers and email addresses
- Login credentials (usernames and passwords)
- Other personal details

Autofill can be triggered in various ways, for example:

- When the user starts typing in a form field
- When the user taps on a form field
- When the user uses a password manager or a browser's built-in autofill feature

Autofill is available on most mobile devices, including Android and iOS devices, and can be managed through the device's settings or through individual apps, such as web browsers or password managers.

In the context of a 3DS challenge on a mobile device, the autofill function could be used to automatically fill a password or SMS OTP on the challenge UI.

Benefits by Actor

- Issuer: reduced risk of Cardholder error during the challenge
- Cardholder: improved challenge experience by reducing friction

Technical Features

Preconditions

The authentication is initiated from a 3DS SDK with a 2.3.1 or higher protocol version.

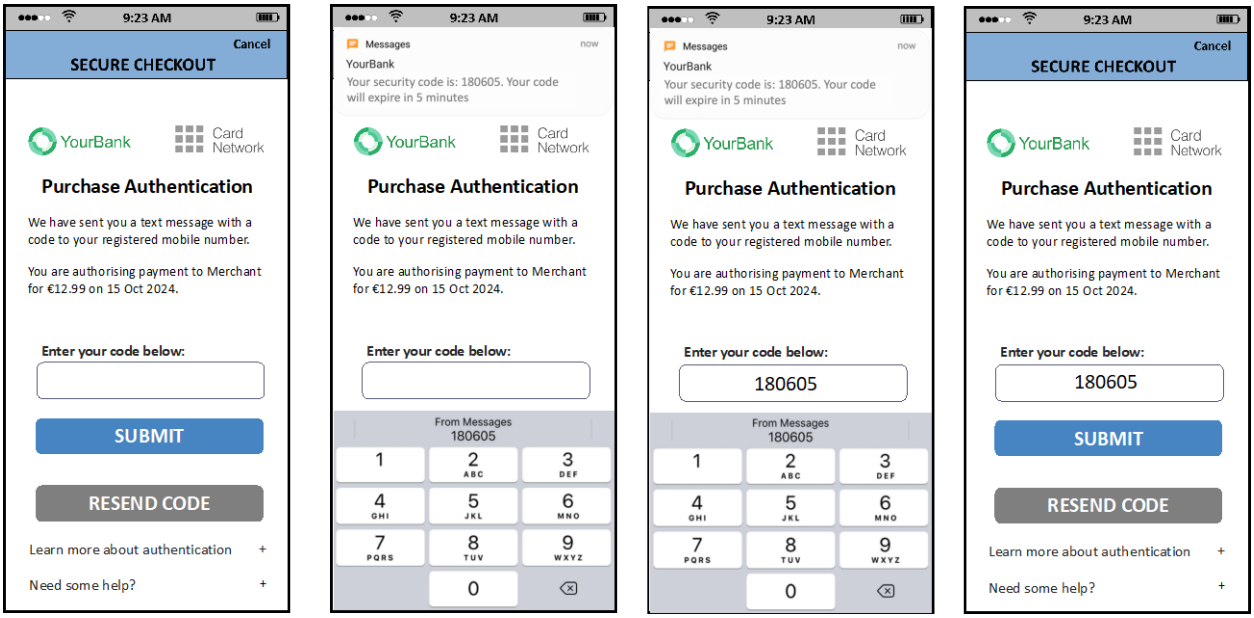
The ACS has chosen to use the autofill feature during the challenge.

Overview

During a challenge, the ACS has the option to enable the Autofill option for the ACS UI template = Text. The ACS can select between OTP or Password.

When the Autofill option is enabled, the OS will display to the Cardholder a pop-up with the suggested input field. The Cardholder validates the input and submits their response.

Figure 4.21: Example Cardholder Experience Involving OTP



The ACS shall ensure that a text message is sent to the Cardholder's phone.

Note: The device operating system may implement its own heuristics to determine whether the input field is for login or a verification code, which may lead to incorrect suggested input. The recommendation for the ACS is to conduct extensive tests before enabling the Autofill option. To learn more, please refer to the mobile operating systems' detailed autofill information (Android and iOS).

Table 4.6 below lists the data elements that may be provided in relation to 3DS Autofill.

Table 4.6: 3DS Data Elements Related to 3DS Autofill

Data Element	Description	Version
Challenge Entry Box	Defines the setting of an entry box in the Native UI OTP/Text Template: • Challenge Data Entry Keyboard Type • Challenge Data Entry Autofill • Challenge Data Entry Autofill Type • Challenge Data Entry Length Maximum • Challenge Data Entry Label • Challenge Data Entry Masking • Challenge Data Entry Masking Toggle	2.3.1
Challenge Entry Box 2	Defines the setting of an entry box in the Native UI OTP/Text Template: • Challenge Data Entry Keyboard Type • Challenge Data Entry Autofill • Challenge Data Entry Autofill Type • Challenge Data Entry Length Maximum • Challenge Data Entry Label • Challenge Data Entry Masking • Challenge Data Entry Masking Toggle	2.3.1 2.2
Challenge Data Entry Autofill	Indicates if the 3DS SDK enables the autofill option for the Challenge Data Entry. When enabled, the 3DS SDK/OS automatically copies the received or saved code or password in the Challenge Data Entry. If Challenge Data Entry Autofill is not present, the option is not enabled.	2.3.1
Challenge Data Entry Autofill Type	Indicates the type of data expected when the Challenge Data Entry Autofill is active. Refer to the following for Android or iOS.	2.3.1

5 Out-of-Band (OOB) Authentication

5.1 Business Overview

Out-of-band (OOB) authentication adds an extra layer of security to the authentication process by requiring the Cardholder to authenticate with their bank through a separate channel. The use of a different channel makes the authentication process more resistant to attacks such as man-in-the-middle attacks, where an attacker intercepts and modifies the communication between the Cardholder and the authentication server.

In the context of 3DS, OOB authentication can be used to verify the identity of the Cardholder during a transaction. For example, the Cardholder initiates a payment, and the Issuer decides that a challenge is needed to confirm the transaction. Instead of conducting the challenge in the Merchant environment (App or Browser), the Issuer instructs the Cardholder to use a separate authentication app to verify their identity using an OOB channel. In the authentication app, the Issuer can request to the Cardholder any preferred authentication process. Issuers typically use their banking website or mobile banking apps that they fully control and trust. Once the Cardholder has been authenticated using the OOB channel, the Issuer can notify the Merchant that the authentication was successful.

Overall, the use of OOB authentication in 3DS can help reduce the risk of fraud and improve the security of online transactions, providing greater protection for both Cardholders and Merchants.

OOB authentication is an effective authentication mechanism that involves two signals from two separate channels. This method is used to block fraudulent users who have access to only one of the channels. OOB authentication is known to be effective in preventing fraudulent attacks, especially in e-commerce. The key benefit of 3DS OOB authentication is that it gives the Issuer full control over the selection of Cardholder authentication methods, which include biometric authentication, tokens, and one-time password via SMS or email. OOB authentication is an ideal choice to protect Cardholders while enabling Issuers to customise services according to their preferences.

The Core Specification supports OOB authentication for both Browser- and App-based transactions by providing a specific user interface template, and automation of the transition from the Merchant app to the OOB App in the context of mobile devices. Another key benefit is the ability to leverage consistent authentication methods across 3DS and other channels, such as online banking.

5.2 OOB – Introduction

OOB authentication is a challenge activity that is completed outside of, but in parallel to, the 3DS flow. OOB authentication methods or implementations are not in scope of the Core Specification.

Benefits by Actor

- Merchant
  - Cardholders are used to the authentication process defined by the ACS, so there is less abandonment or failure
  - automated App-to-App transfer between merchant and OOB apps when on the same device (App-based flow in 3DS version 2.3)
- Issuer
  - consistent authentication methods for Cardholders
  - simpler customer education and support
- Cardholder – similar user experience across all Merchants

Use Case Overview

During a challenge, the ACS directs the Cardholder to use a specific channel and application to authenticate the transaction, instead of using the 3DS challenge window to authenticate the Cardholder. For example, the Issuer requests the use of the mobile banking app to authenticate and validate the transaction.

The OOB flow depends on:

- the Core Specification version;
- the channel used by the Cardholder for the transaction and the channel used by the ACS for the authentication;
- whether the OOB Authentication App is on the same device as the transaction – for an App-based transaction;
- whether the transition from the 3DS Requestor checkout page to the OOB Authentication App, and the return, is manual or automated.

Table 5.1 below shows all the possible options and indicates whether automation of the transition between the merchant app and the OOB Authentication App is possible.

Table 5.1: OOB Authentication per Channel and Automation

Merchant Channel	OOB App Channel	Same Device	OOB App Transition Automation
Browser	Browser	Yes	No
Browser	App	Yes	No
App	Browser	Yes	No
App	App	Yes	2.2, 2.2 + Bridging Message Extension and 2.3.1
App/Browser	App/Browser	No	No

Note: This table assumes an implementation fully compliant with the Core Specification – in particular, the setting of the iframe for the challenge in the Browser flow, and the use of Universal App Link for the App flow.

5.3 OOB Flow for the Browser Channel

For a Browser-based transaction and all versions of the Core Specification, during the challenge, the ACS instructs the Cardholder to manually switch from the payment/checkout page to the OOB Authentication App. The OOB Authentication App may be accessible using a Browser or an app on the same or different device. When the Cardholder has completed the authentication, the Cardholder returns to the challenge window in the payment/checkout page and must select the Complete button.

Note: Unlike in the App-based flow, in the Browser-based flow it is not possible to automate the switch between the Browser – 3DS Requestor page and the OOB Authentication App.

Preconditions

The ACS has defined, deployed, and communicated an OOB authentication process to the Cardholder.

The Issuer has a pre-established authentication process with the Cardholder using the OOB Authentication App, available as a web service or a mobile app, and accessed on any device or a specific device at the ACS's preference.

Assumptions

The 3DS Requestor Website and the OOB Authentication App do not need to be on the same device.

Sequence Diagram

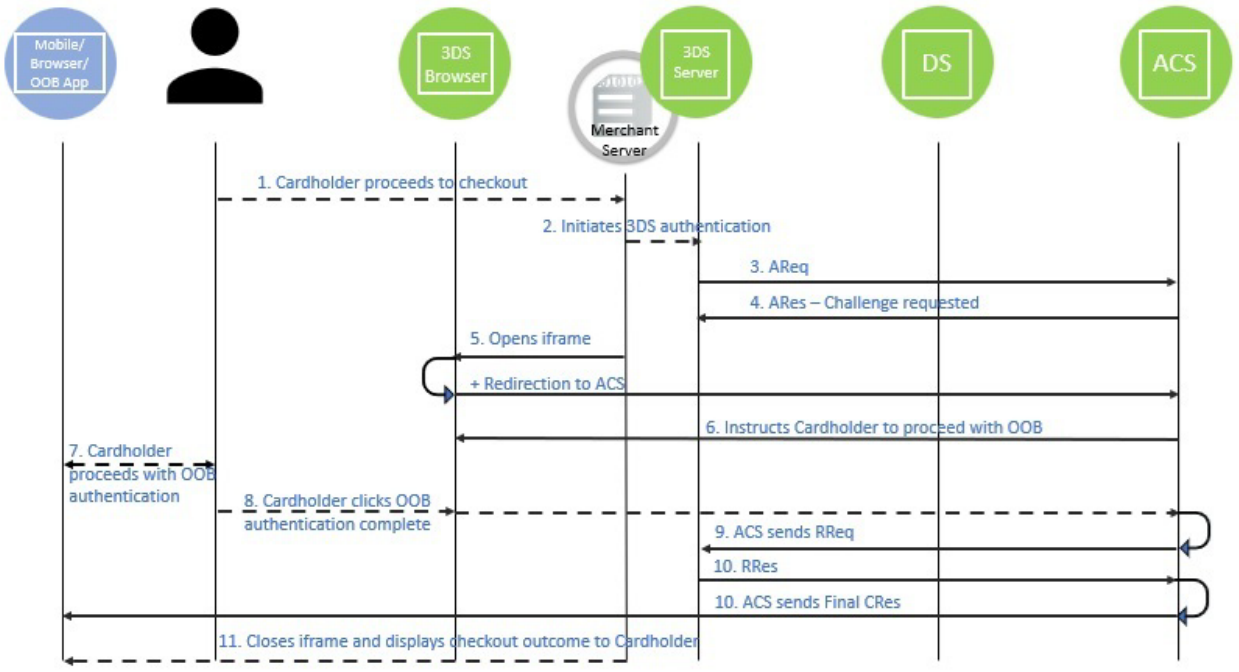
The Cardholder authenticates the transaction using an OOB Authentication App provided by the ACS.

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Requestor initiates a 3DS authentication.
3. The 3DS Server sends an AReq message.
4. The ACS responds with an ARes message requesting a challenge.
5. The 3DS Requestor proceeds with the challenge, opens an iframe in its checkout page and makes the redirection to the ACS.
6. The ACS provides the UI in the iframe and instructs the Cardholder to proceed with an OOB authentication.
7. The Cardholder switches to the OOB App, which may be available on a Browser or as a mobile app, on the same or different device. The Cardholder completes the authentication with the OOB App as instructed by the ACS or authentication system provider. The OOB authentication is defined and controlled by the ACS, and thus falls outside the scope of the Core Specification.
8. The Cardholder manually switches to the 3DS Requestor checkout page and selects the Complete button.

9. The ACS sends the result of the authentication in the RReq message to the 3DS Server through the DS.
10. After receiving the RRes message from the 3DS Server through the DS, the ACS sends a Final Challenge Response (CRes) message through the iframe to the 3DS Requestor to indicate the end of the challenge and the outcome of the authentication.
11. The 3DS Requestor closes the iframe and updates the UI according to the outcome of the authentication and/or authorisation.

Note: In Step 9, the ACS may continue the challenge if the OOB authentication was not performed or if it failed, before sending the Final CRes message.

Figure 5.1: OOB Flow – Browser Channel



Note: Automation (URLs to and from the OOB Authentication App) of the OOB flow in the Browser channel is not possible.

5.3.1 Browser Channel – Alternative OOB Flow

In this alternative OOB flow, the ACS directly accesses the result of the OOB authentication and sends the RReq and Final CRes messages before the Cardholder manually switches back to the 3DS Requestor checkout page and selects the Complete button.

Note: This flow is recommended as the Cardholder does not need to click the Complete button for the challenge to complete.

Preconditions

The ACS has defined, deployed, and communicated an OOB authentication process to the Cardholder. The OOB Authentication App may be available as a web service or as a mobile app, and may be accessed on any device, or on a specific device at the ACS's preference.

The ACS receives or knows the result of the OOB authentication (pass or fail) before the Cardholder confirms completion (selects the Complete button) and does not perform additional challenges after the OOB authentication.

Benefits

The main benefits are as follows:

- The 3DS Requestor receives the completion information from the ACS, and can close the iframe and update the UI according to the outcome of the authentication and/or authorisation, without the Cardholder having to click the Complete button in the iframe.
- If the iframe is still open, the Cardholder may select the Complete button – this does not impact the outcome of the transaction.
- This alternative flow prevents the Authentication from failing after the 10-minute timeout due to lack of Cardholder interaction.

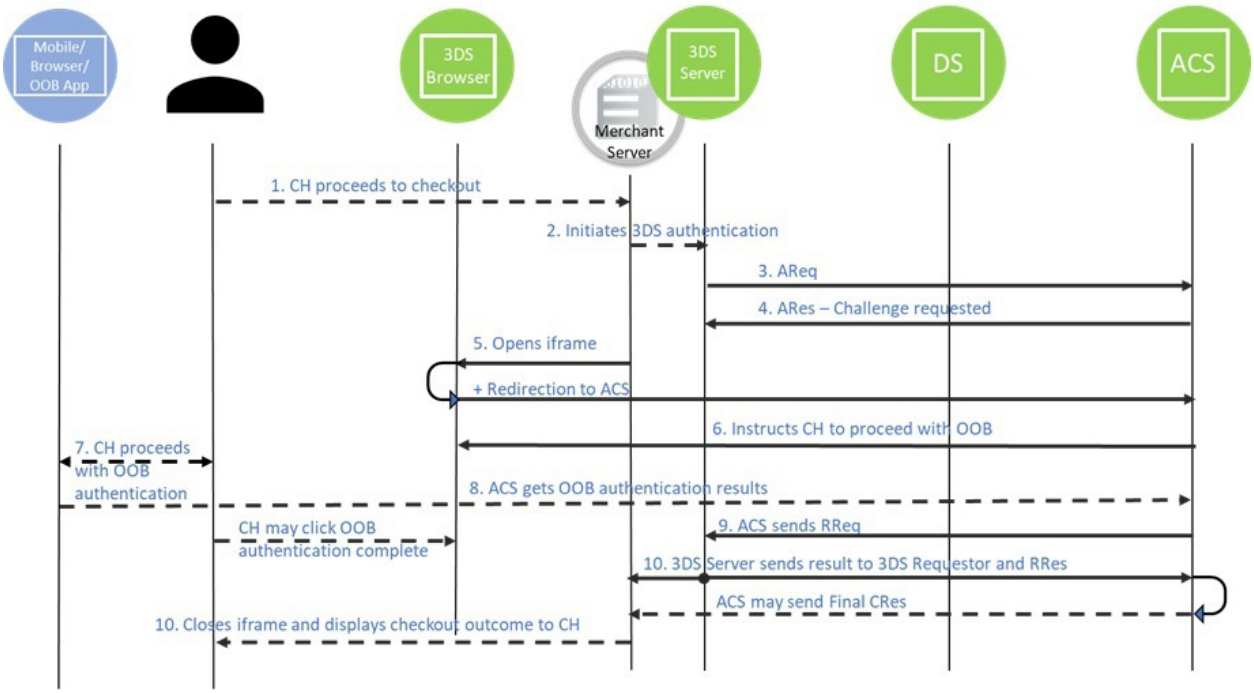
Sequence Diagram

The Cardholder authenticates the transaction using an OOB Authentication App provided by the ACS.

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Requestor initiates a 3DS authentication.
3. The 3DS Server sends an AReq message.
4. The ACS responds with an ARes message requesting a challenge.
5. The 3DS Requestor proceeds with the challenge, opens an iframe in its checkout page and makes the redirection to the ACS.
6. The ACS provides the UI in the iframe and instructs the Cardholder to proceed with an OOB authentication.
7. The Cardholder switches to the OOB App, which may be available on a Browser or as a mobile app, on the same or different device. The Cardholder completes the authentication with the OOB App as instructed by the ACS or authentication system provider. The OOB authentication is defined and controlled by the ACS, and thus falls outside the scope of the Core Specification.
8. The ACS receives the result of the OOB authentication and does not need to wait for the authentication completion information from the Cardholder (the Complete button).
9. The ACS sends the results of the authentication in the RReq message through the DS to the 3DS Server.
10. The 3DS Server provides the authentication result to the 3DS Requestor and sends the RRes message to the ACS through the DS. The 3DS Requestor closes the iframe and updates the UI according to the outcome of the authentication and/or authorisation.

Note: In parallel to Step 8 and 10, and if the iframe is still open, the Cardholder may manually switch to the 3DS Requestor checkout page and select the Complete button. After receiving the RRes message from the 3DS Server, the ACS may send the Final CRes message through the iframe.

Figure 5.2: Alternative OOB Flow - Browser Channel



5.3.2 Browser Channel – Mobile Browser

Background

Some 3DS Requestors and ACSs have reported a lower success rate for OOB authentications initiated from a mobile browser.

These failures manifest in the ACS sending declined authentication responses to the 3DS Server, such as failed challenge interactions (for example, mainly through timeouts with Challenge Cancelation Indicator 04 or 05), or in the 3DS Requestor being unaware that the Cardholder has successfully completed the challenge.

Error Scenario

This section describes the likely error scenario during an OOB authentication when a Cardholder is interacting with the 3DS Requestor on a mobile Browser, and the OOB authentication is performed on an app on the same device.

1. During the challenge, the ACS instructs the Cardholder to authenticate the transaction using the OOB Authentication App.
2. The Cardholder switches from the mobile Browser (Merchant webpage) to the OOB Authentication App (usually a mobile banking app).

3. The device operating system (OS) manages the switch of applications, puts the Browser in the background, and starts the OOB Authentication App, bringing the user interface to the foreground. If the memory available on the device is limited, the OS instructs the apps in the background (for example, the Browser) to free up memory, or, in a worst-case scenario, kill certain apps. In turn, the Browser deactivates tabs to comply with the OS request.
4. When the Cardholder returns to the Browser, the Browser reactivates the tab the Cardholder was on. However, the reactivation process essentially behaves as a refresh of the tab.
5. If the 3DS Requestor did not receive the authentication result from the 3DS Server in the meantime (RReq message from the ACS/Directory Server), the 3DS Requestor Environment reloads the checkout page, opens the challenge iframe, and posts a CReq message to the ACS to restore the challenge context of the transaction.
6. Depending on how the ACS handles the challenge and the receipt of multiple CReq messages, the ACS may return an RReq message with a failed authentication result or an Error Message to the 3DS Server, or the transaction may time out.

In theory, this scenario is possible for a transaction initiated from a 3DS Requestor application (Merchant app), but it is unlikely due to its smaller memory footprint compared to a Browser.

The memory and app management by the OS depends on multiple factors, including memory size, the number of apps open simultaneously, or the OS's capabilities to save and restore the app's context. Therefore, it is not possible to predict the OOB authentication error. Feedback from major OS providers indicates that lower-end devices with limited RAM and processing power contribute to this issue, but higher-end devices are also not immune to this behaviour.

Recommendations

The following recommendations and solutions are proposed to develop best practices for handling OOB challenges and harmonise the Merchant, 3DS Server, and ACS implementations.

1. 3DS Servers should share with the 3DS Requestor, the authentication status (Transaction Status) from the RReq message as soon as it is received. To prevent any delay, the 3DS Server should push the information to the Merchant (instead of the Merchant itself having to retrieve the information).

2. In case of a Browser restart, the 3DS Requestor shall always restore the challenge iframe so that the ACS can send the Final CRes message. The 3DS Requestor should refresh the payment page, reopen an iframe in the checkout page and post the same CReq message as the initial one through the iframe to the ACS.

3. The ACS should verify the completion of the OOB authentication directly from the authentication server and immediately send the RReq message when the OOB authentication is completed. The ACS should not wait for the Cardholder to confirm the completion through Browser interaction (for example, the Continue button), as the challenge window is lost when the Browser is put to sleep or closed by the device OS, which results in a timeout as the Cardholder interaction is no longer possible.

4. ACSs shall accept multiple CReq messages for a mobile Browser-based transaction, as a possible option under Req 442 (in *Core Specification* version 2.3.1 and Specification Bulletin 214 for version 2.2). The ACS should verify the consistency of the IP address, 3DS Requestor Session Data, and CReq message content before continuing the challenge with

the Cardholder. The ACS should check the OOB authentication status before taking any action.

- If the OOB authentication was completed, the ACS should confirm challenge completion with the Cardholder and send the Final CRes message.
- If the OOB authentication was not completed or performed, the ACS should restart the OOB authentication with the Cardholder.

Note: If the ACS receives the CReq message more than once after attempting an OOB authentication, it should evaluate whether the OOB authentication was not successful or if it has not already received multiple CReq messages before sending a new CRes message, the ACS should avoid remaining in the loop: “OOB authentication not performed – receipt of a new CReq message” and exit the loop if necessary, and fail the authentication.

If a CReq message is received after the RReq/RRes message, the ACS should accept the CReq message (not return an Error message), and confirm challenge completion with the Cardholder, and send the Final CRes.

If the ACS has already sent the RReq message and lost the connection to the challenge iframe, the ACS may not be able to send the Final CRes message if the 3DS Requestor does not re-open the challenge iframe.

5. Issuers should be mindful of the size of the OOB Authentication App and keep it minimal to avoid device memory issues. Heavy applications can increase the likelihood of the OS requiring the Browser to free up resources.

Addressing these issues through standardised practices and leveraging new technologies is expected to improve the challenge success rate, leading to better user experiences and more reliable transactions.

Sequence Diagram

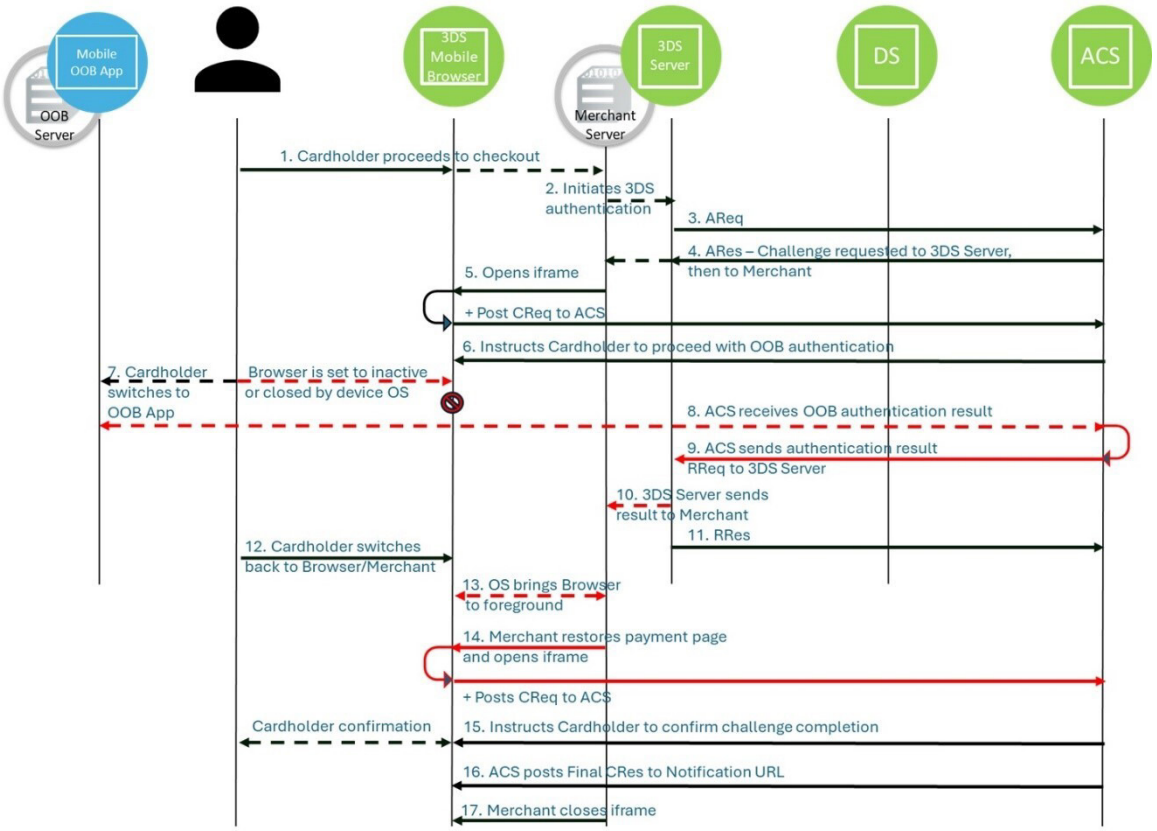
The following sequence diagram illustrates the 3DS Browser-based flow, taking into account the above recommendations to prevent the failure of the OOB authentication when performed on a mobile device.

1. The Cardholder makes a purchase and proceeds to checkout.
2. The 3DS Requestor initiates a 3DS authentication.
3. The 3DS Server sends an AReq message.
4. The ACS responds with an ARes message requesting a challenge.
5. The 3DS Requestor proceeds with the challenge, opens an iframe in their checkout page and posts the CReq message through the iframe to the ACS.
6. The ACS provides the UI in the iframe and instructs the Cardholder to proceed with an OOB authentication.
7. The Cardholder switches to the mobile OOB App that is on the same device. The device OS sets the Browser to an inactive state or closes the Browser to free up device resources.

8. The ACS actively retrieves the result of the authentication from the OOB authentication server.
9. The ACS sends the result of the authentication in an RReq message to the 3DS Server through the DS.
10. Immediately upon receiving the RReq message, the 3DS Server shares the authentication result with the 3DS Requestor.
11. The 3DS Server sends an RRes message to the ACS through the DS.
12. In the meantime, the Cardholder switches from the OOB App back to the Browser.
13. The device OS brings the Browser back to the foreground.
14. The 3DS Requestor refreshes the payment page, reopens an iframe in the checkout page and posts the same CReq message as the first one through the iframe to the ACS.
15. The ACS provides the UI in the iframe and instructs the Cardholder to confirm the challenge completion.
16. After receiving the Cardholder's confirmation, the ACS sends the Final CRes message through the iframe to the 3DS Requestor to indicate the end of the challenge and the outcome of the authentication.

The 3DS Requestor closes the iframe and updates the UI according to the outcome of the authentication and/or authorisation.

Figure 5.3: Browser Channel - Mobile Browser



5.4 OOB Flow: App Channel – Manual Switching

During the challenge, the ACS instructs the Cardholder to manually switch from the payment/checkout page to the OOB Authentication App. The OOB Authentication App may be on the same or different device. When the Cardholder has completed the authentication, the Cardholder returns to the challenge window in the payment/checkout page and must select the Complete button.

This flow is applicable to all versions of the Core Specification.

Preconditions

The ACS has defined, deployed, and communicated an OOB authentication process to the Cardholder.

The ACS and the 3DS Requestor do not use the 3DS Requestor App URL and the OOB App URL (version 2.2 and 2.3.1).

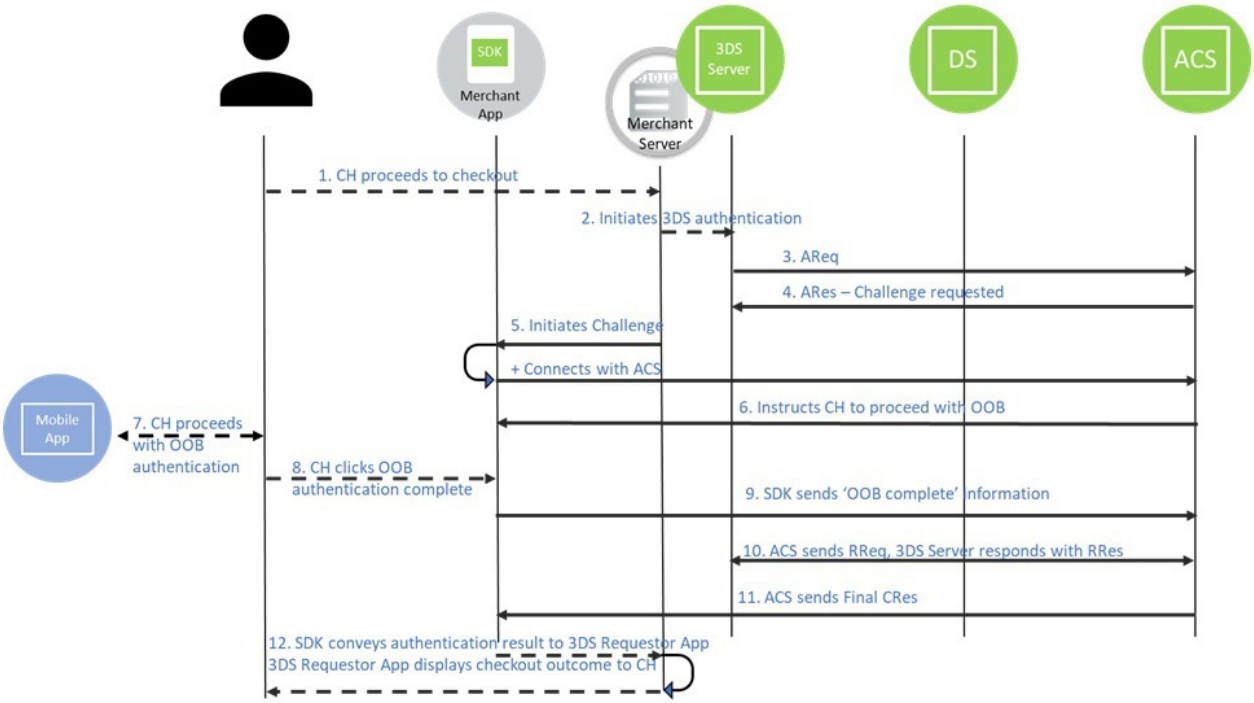
Assumptions

The 3DS Requestor App and the OOB Authentication App do not need to be on the same device.

Sequence Diagram

1. The Cardholder makes a purchase on the 3DS Requestor App and proceeds to checkout.
2. The 3DS Requestor initiates a 3DS authentication.
3. The 3DS Server sends an AReq message.
4. The ACS responds with an ARes message requesting a challenge.
5. The 3DS Requestor triggers the 3DS SDK to proceed with a challenge. The 3DS SDK connects to the ACS.
6. The ACS provides the UI to the 3DS SDK and instructs the Cardholder to proceed with the OOB authentication.
7. The Cardholder switches to the OOB App, which may be available on a Browser or as a mobile app, on the same or different device. The Cardholder completes the authentication with the OOB App as instructed by the ACS or authentication system provider. The OOB authentication is defined and controlled by the ACS, and thus falls outside the scope of the Core Specification.
8. The Cardholder manually switches to the 3DS Requestor App and selects the Complete button displayed by the 3DS SDK.
   If the OOB App and the 3DS Requestor App are on the same device, the 3DS SDK will automatically send a CReq message when the 3DS Requestor App is moved to the foreground.
9. The 3DS SDK sends the “OOB complete” information to the ACS.
10. The ACS sends the results of the authentication in an RReq message through the DS to the 3DS Server, and the 3DS Server acknowledges it by sending the RRes message.
11. The ACS sends a Final CRes message to the 3DS SDK, the 3DS SDK conveys the information to the 3DS Requestor App.
12. The 3DS Requestor App updates the UI according to the outcome of the authentication and/or authorisation.

Figure 5.4: OOB Flow - App Channel - Manual Switching



Note: In Step 9, the ACS may continue the challenge if the OOB authentication was not performed or if it failed, before sending the Final CRes message.

Note: If the ACS receives or knows the result of the OOB authentication (pass or fail) before the Cardholder confirms completion, it may send an RReq message before receiving the “OOB complete” information from the 3DS SDK.

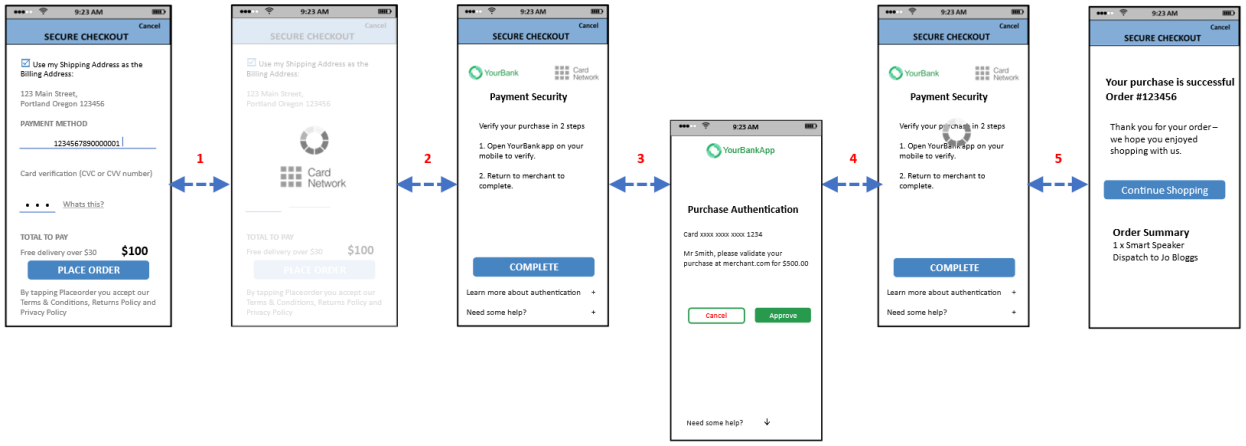
User Experience

The Cardholder authenticates the transaction using an OOB Authentication App provided by the ACS.

1. The Cardholder makes a purchase and proceeds to checkout. The 3DS Requestor App displays the processing screen while waiting for the ACS’s response.
2. The 3DS SDK displays the UI provided by the ACS to proceed with an OOB authentication.
3. The Cardholder manually switches to the OOB App.
4. When the OOB authentication is completed, the Cardholder manually switches back to the 3DS Requestor App.
5. The Cardholder selects the Complete button.
   If the OOB App and the 3DS Requestor App are on the same device, the 3DS SDK will automatically send a CReq message when the 3DS Requestor App is moved to the foreground, and the Cardholder will not need to select the Complete button.
6. The 3DS Requestor App displays the purchase completion information.

Note: After Step 4, the ACS may continue the challenge if the OOB authentication was not performed or if it failed OR send the Final CRes message as shown.

Figure 5.5: OOB Flow - App Channel - Manual Switching: User Experience



5.4.1 3DS Version 2.2 and 2.3.1 Data Elements

Table 5.2 below lists the data elements that may be provided in relation to OOB – manual switching.

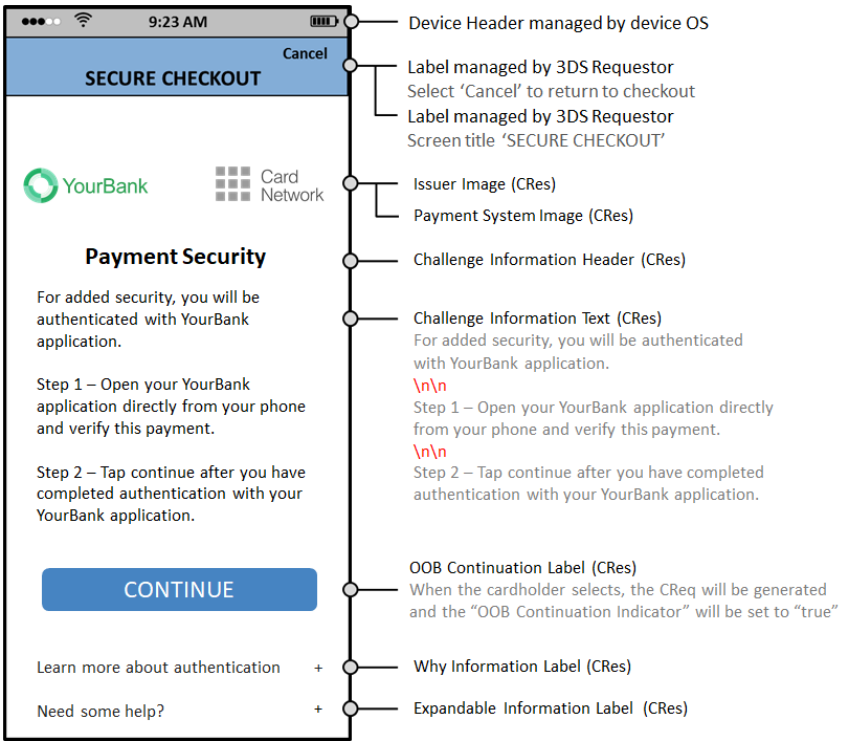
Table 5.2: 3DS Data Elements Related to OOB – Manual Switching

Data Element	Description	Version
ACS Interface	The interface that the challenge presents to the cardholder.	2.3.1 2.2
ACS UI Template	Identifies the UI Template format that the ACS first presents to the Cardholder.	2.3.1 2.2
ACS UI Type	User interface type that the 3DS SDK will render, which includes the specific data mapping and requirements.	2.3.1 2.2
Authentication Method	Indicates the list of authentication types the Issuer will use to challenge the Cardholder, when in the ARes message or used by the ACS in the RReq message. The authentication approach that the ACS used to authenticate the Cardholder for this specific transaction.	2.3.1 2.2
Authentication Type	Indicates the type of authentication method the Issuer will use to challenge the Cardholder, whether in the ARes message or used by the ACS in the RReq message.	2.2

Data Element	Description	Version
OOB Continuation Label	Label to be used in the UI for the button that the Cardholder selects when they have completed the OOB authentication.	2.3.1 2.2
SDK UI Type	Lists all UI types supported by the device for displaying specific challenge user interfaces within the 3DS SDK.	2.3.1 2.2

5.4.2 OOB User Interface for 3DS Version 2.2 and 2.3.1

Figure 5.6: Native User Interface



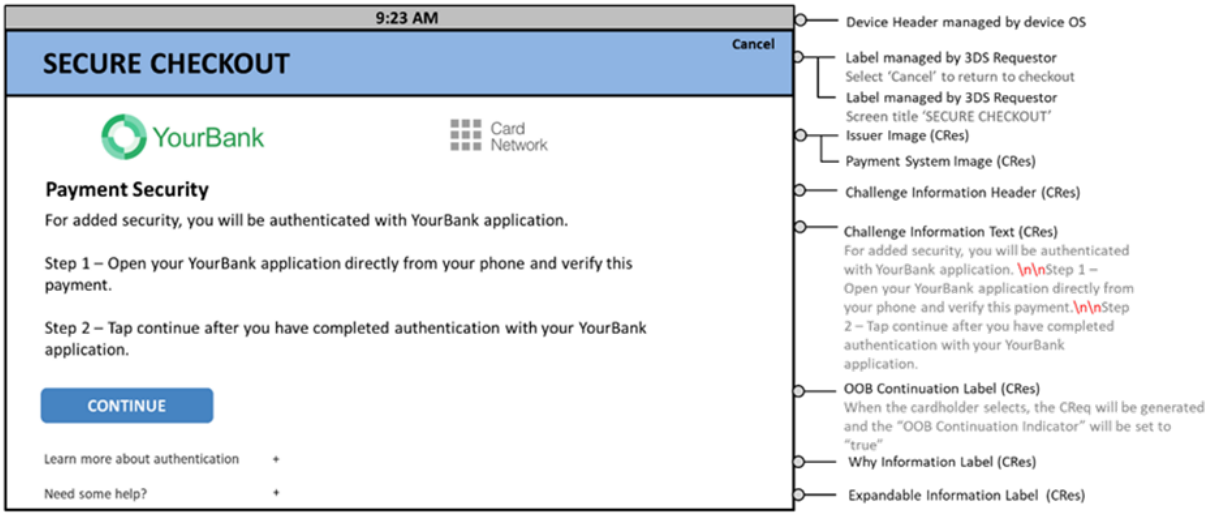
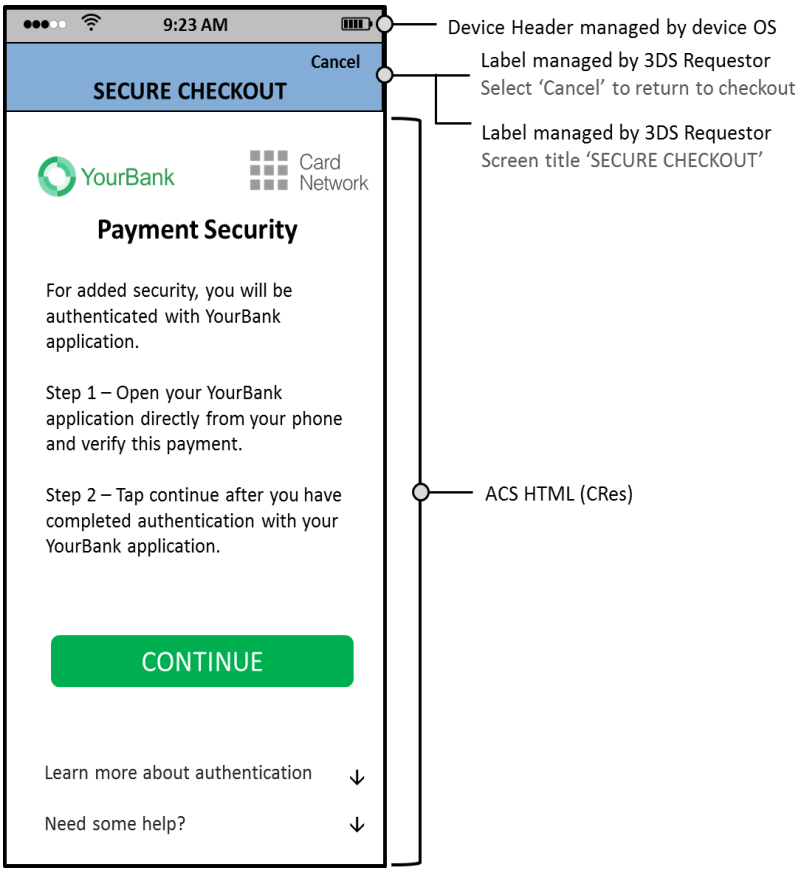
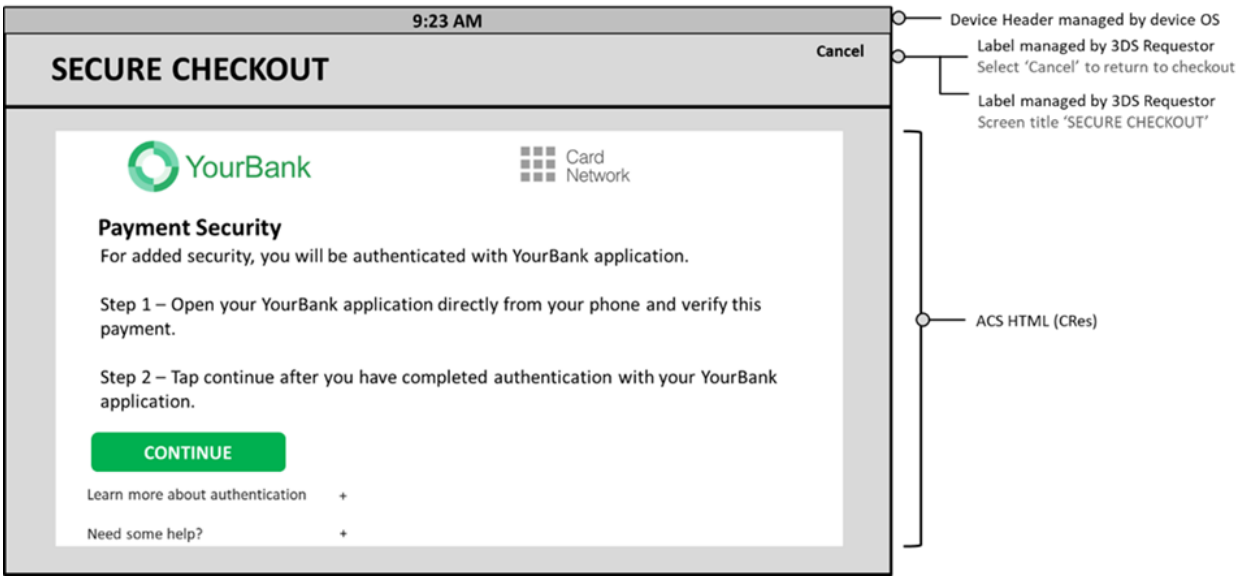


Figure 5.7: HTML User Interface





5.5 OOB Flow App Channel – Automatic Switching to the 3DS Requestor App

For version 2.2 and above of the Core Specification, it is possible to automate the switching from the OOB Authentication App to the 3DS Requestor App.

During the challenge, the ACS instructs the Cardholder to manually switch from the payment/checkout page to the OOB Authentication App. When the Cardholder has completed the authentication, the Authentication App will automatically return to the Challenge screen on the 3DS Requestor App, assuming that the two apps are on the same device.

Preconditions

The ACS has defined, deployed, and communicated an OOB authentication process to the Cardholder.

The OOB Authentication App can handle the 3DS Requestor App URL.

The 3DS Requestor provides the 3DS Requestor App URL to the 3DS SDK.

The ACS provides the 3DS Requestor App URL to the OOB Authentication App.

The 3DS Requestor App and the OOB Authentication App are on the same device.

The 3DS Requestor App URL is a Universal App Link (refer to Table 1.3 in version 2.2 of the Core Specification).

Sequence Diagram

The Cardholder authenticates the transaction using an OOB Authentication App that is on the Device used for the purchase.

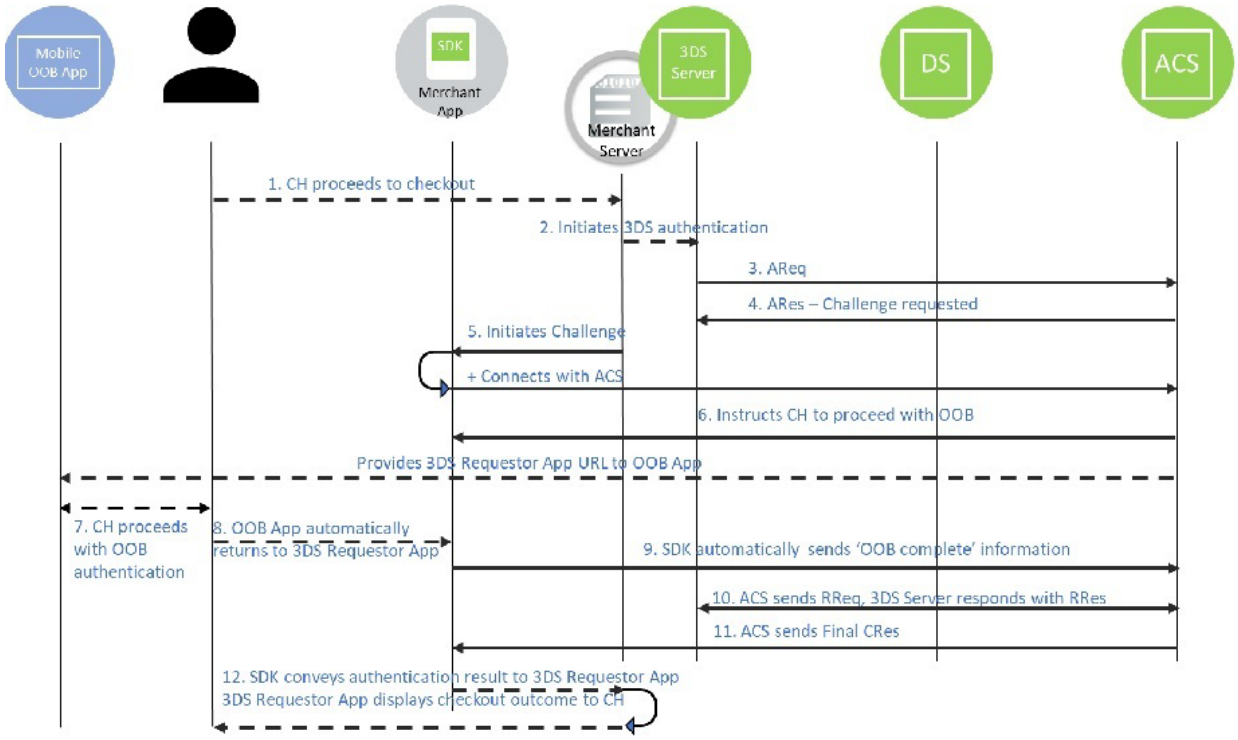
1. The Cardholder makes a purchase on the 3DS Requestor App and proceeds to checkout.
2. The 3DS Requestor initiates a 3DS authentication.

3. The 3DS Server sends an AReq message.
4. The ACS responds with an ARes message requesting a challenge.
5. The 3DS Requestor triggers the 3DS SDK to proceed with a challenge. The 3DS SDK connects to the ACS and provides the 3DS Requestor App URL to the ACS.
6. The ACS provides the UI to the 3DS SDK and instructs the Cardholder to proceed with an OOB authentication.
   The ACS conveys the 3DS Requestor App URL to the OOB Authentication App.
   Note: The ACS also displays the Complete button if the OOB Authentication App is on a different device.
7. The Cardholder switches to the OOB App on the same device, and completes the authentication as instructed by the ACS or authentication system provider.
   The OOB authentication is defined and controlled by the ACS, and thus falls outside the scope of the Core Specification.
8. The OOB Authentication App uses the 3DS Requestor App URL (Universal App Link) to automatically return the Cardholder to the 3DS Requestor App.
9. The Cardholder does not need to select the Complete button, the 3DS SDK detects that the 3DS Requestor App is back in the foreground and sends a Challenge Request (CReq) message to the ACS (Automatic CReq). The 3DS SDK sends the “OOB complete” information to the ACS.
10. The ACS sends the results of the authentication in an RReq message through the DS to the 3DS Server, and the 3DS Server acknowledges it by sending an RRes message.
11. The ACS sends a Final CRes message to the 3DS SDK, the 3DS SDK conveys the information to the 3DS Requestor App.
12. The 3DS Requestor App updates the UI according to the outcome of the authentication and/or authorisation.

Note: In Step 9, the ACS may continue the challenge if the OOB authentication was not performed or if it failed, before sending the Final CRes message.

Note: If the ACS receives or knows the result of the OOB authentication (pass or fail) before the Cardholder confirms completion, it may send the RReq before receiving the “OOB complete” information from the 3DS SDK.

Figure 5.8: OOB Flow App Channel - Automatic Switching

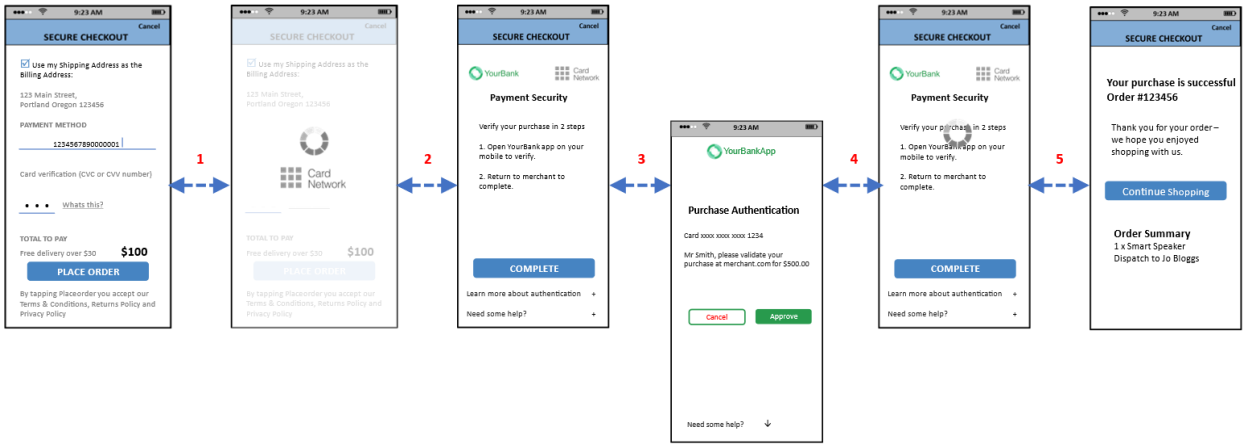


User Experience

1. The Cardholder makes a purchase and proceeds to checkout. The 3DS Requestor App displays the processing screen while waiting for the ACS's response.
2. The 3DS SDK displays the UI provided by the ACS to proceed with an OOB authentication.
3. The Cardholder manually switches to the OOB App.
4. When the OOB authentication is completed, the OOB App invokes the 3DS Requestor App URL, and the Cardholder is automatically taken back to the 3DS Requestor App.
5. The 3DS SDK detects that it is in the UI foreground and automatically sends the OOB complete information. The Cardholder does not need to select the Complete button.
6. The 3DS Requestor App displays the purchase completion information.

Note: After Step 4, the ACS may continue the challenge if the OOB authentication was not performed or if it failed, OR send the Final CRes message as shown.

Figure 5.9: OOB Flow App Channel - Automatic Switching: User Experience



5.5.1 Technical Variant: the Device Operating System Cannot Match the 3DS Requestor App URL to an Installed App

User Experience

1. The Cardholder makes a purchase and proceeds to checkout. The 3DS Requestor App displays the processing screen while waiting for the ACS's response.
2. The 3DS SDK displays the UI provided by the ACS to proceed with an OOB authentication.

Note: The ACS also displays the Complete button in case the OOB Authentication App is on a different device.

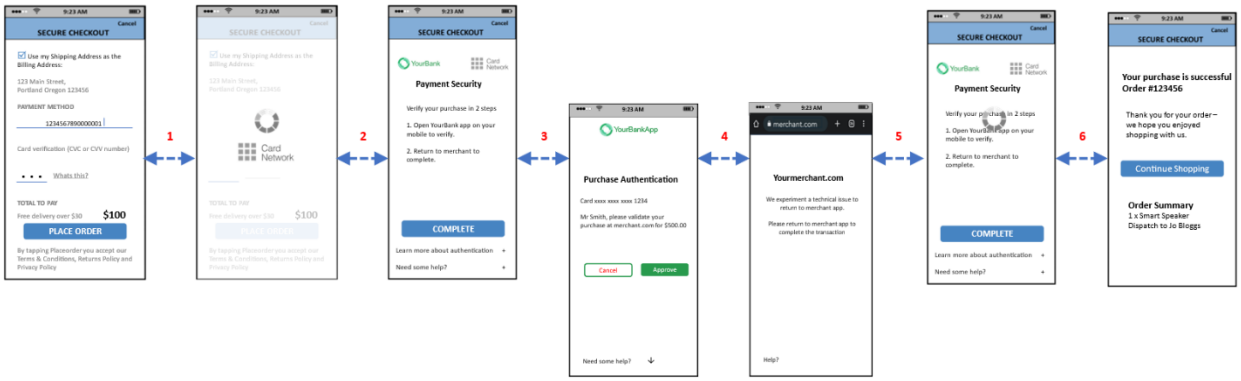
3. The Cardholder manually switches to the OOB App.
4. When the OOB authentication is completed, the OOB Authentication App invokes the 3DS Requestor App URL (Universal App Link) to return to the 3DS Requestor App, but the Device Operating System cannot resolve the URL and opens the default Device Browser.

Note: The 3DS Requestor would need to provide a landing page to instruct the Cardholder to manually switch to the 3DS Requestor App.

5. The Cardholder manually switches to the 3DS Requestor App.
6. The 3DS SDK detects that it is in the UI foreground and automatically sends the OOB complete information to the ACS. The Cardholder does not need to select the Complete button.
7. The 3DS Requestor App displays the purchase completion information.

Note: After Step 4, the ACS may continue the challenge if the OOB authentication was not performed or if it failed OR send the Final CRes message as shown.

Figure 5.10: User Experience: Device OS Cannot Match 3DS Requestor App URL to App



5.5.2 Technical Variant: the 3DS Requestor App URL Is Invalid or Is Based on a Custom Device Operating System

User Experience

1. The Cardholder makes a purchase and proceeds to checkout. The 3DS Requestor App displays the processing screen while waiting for the ACS's response.
2. The 3DS SDK displays the UI provided by the ACS to proceed with an OOB authentication.

Note: The ACS also displays the Complete button in case the OOB Authentication App is on a different device.

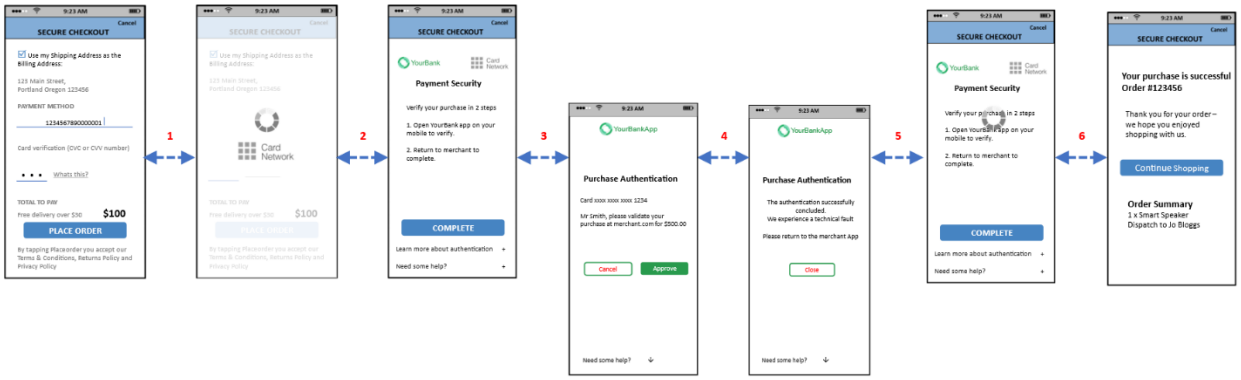
3. The Cardholder manually switches to the OOB App.
4. When the OOB authentication is completed, the OOB App invokes the 3DS Requestor App URL to return to the 3DS Requestor App, but the Device Operating System cannot resolve the URL and returns an error to the OOB Authentication App. The OOB Authentication App displays a page to instruct the Cardholder to manually switch to the 3DS Requestor App.

Note: The OOB Authentication App needs to interpret the error returned by the Device Operating System and be able to display the instructions to the Cardholder.

5. The Cardholder manually switches to the 3DS Requestor App.
6. The 3DS SDK detects that it is in the UI foreground and automatically sends the OOB complete information to the ACS. The Cardholder does not need to select the Complete button.
7. The 3DS Requestor App displays the purchase completion information.

Note: After Step 4, the ACS may continue the challenge if the OOB authentication was not performed or if it failed OR send the Final CRes message as shown.

Figure 5.11: User Experience: 3DS Requestor App URL Invalid or Based on Custom Device Operating System



5.5.3 3DS Version 2.2 and Above Data Elements

Table 5.3 below lists the data elements that may be provided in relation to OOB – automatic switching to the 3DS Requestor App.

Table 5.3: 3DS Data Elements Related to OOB – Automatic Switching to the 3DS Requestor App

Data Element	Description	Version
3DS Requestor App URL	3DS Requestor App declaring its URL within the CReq message so that the Authentication App can call the 3DS Requestor App after OOB authentication has occurred. Each transaction would require a unique Transaction ID by using the SDK Transaction ID.	2.3.1 2.2
ACS UI Template	Identifies the UI Template format that the ACS first presents to the consumer.	2.3.1 2.2
ACS UI Type	User interface type that the 3DS SDK will render, which includes the specific data mapping and requirements.	2.3.1 2.2
Authentication Method	Authentication approach that the ACS used to authenticate the Cardholder for this specific transaction.	2.3.1 2.2
Authentication Type	Indicates the type of authentication method the Issuer will use to challenge the Cardholder, whether in the ARes message or what was used by the ACS when in the RReq message.	2.3.1 2.2

Data Element	Description	Version
OOB Continuation Label	Label to be used in the UI for the button that the Cardholder selects when they have completed the OOB authentication.	2.3.1 2.2
SDK UI Type	Lists all UI types that the device supports for displaying specific challenge user interfaces within the 3DS SDK.	2.3.1 2.2

5.5.4 OOB User Interface for 3DS Version 2.2 and Above

Figure 5.12: Native User Interface

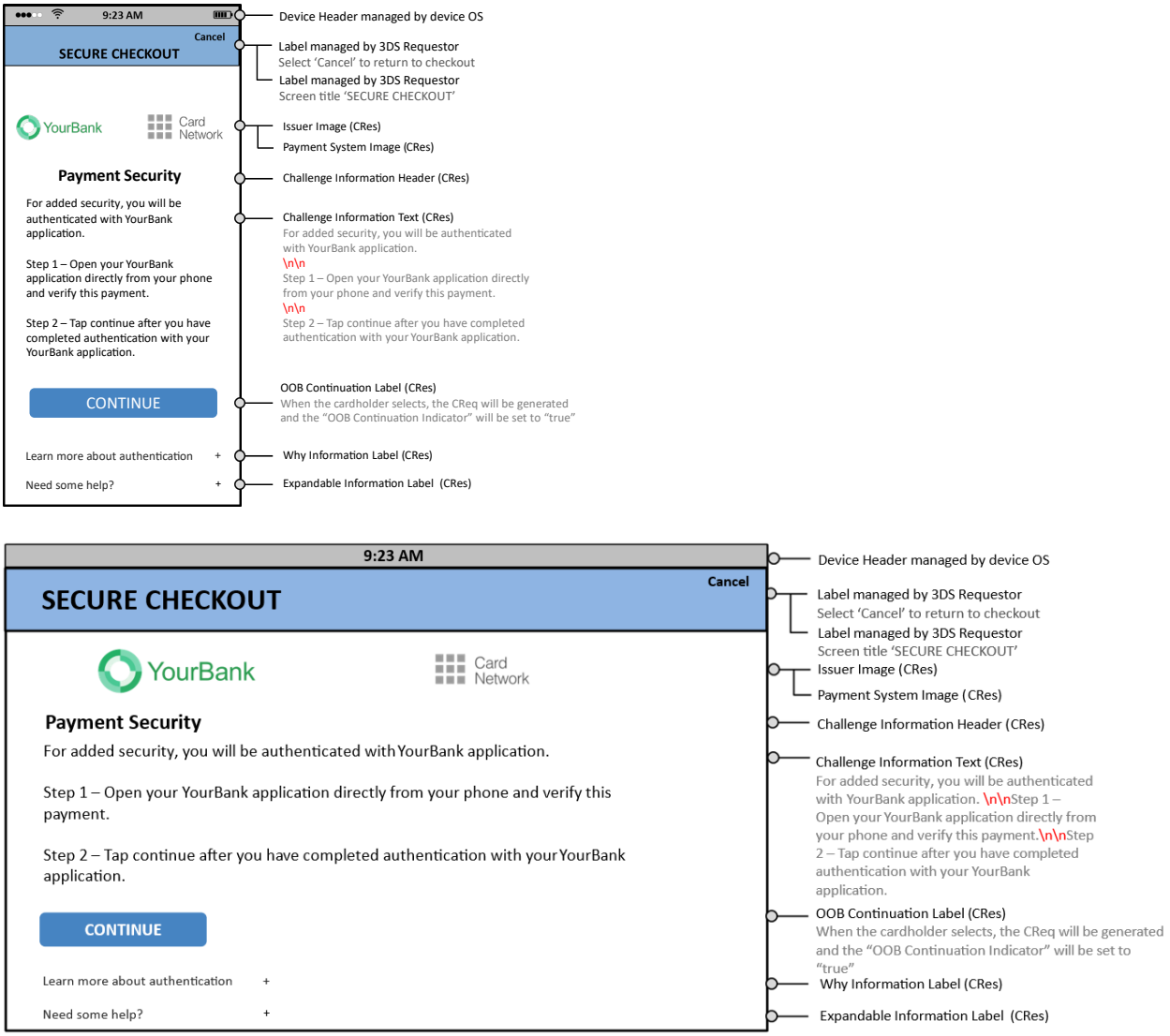
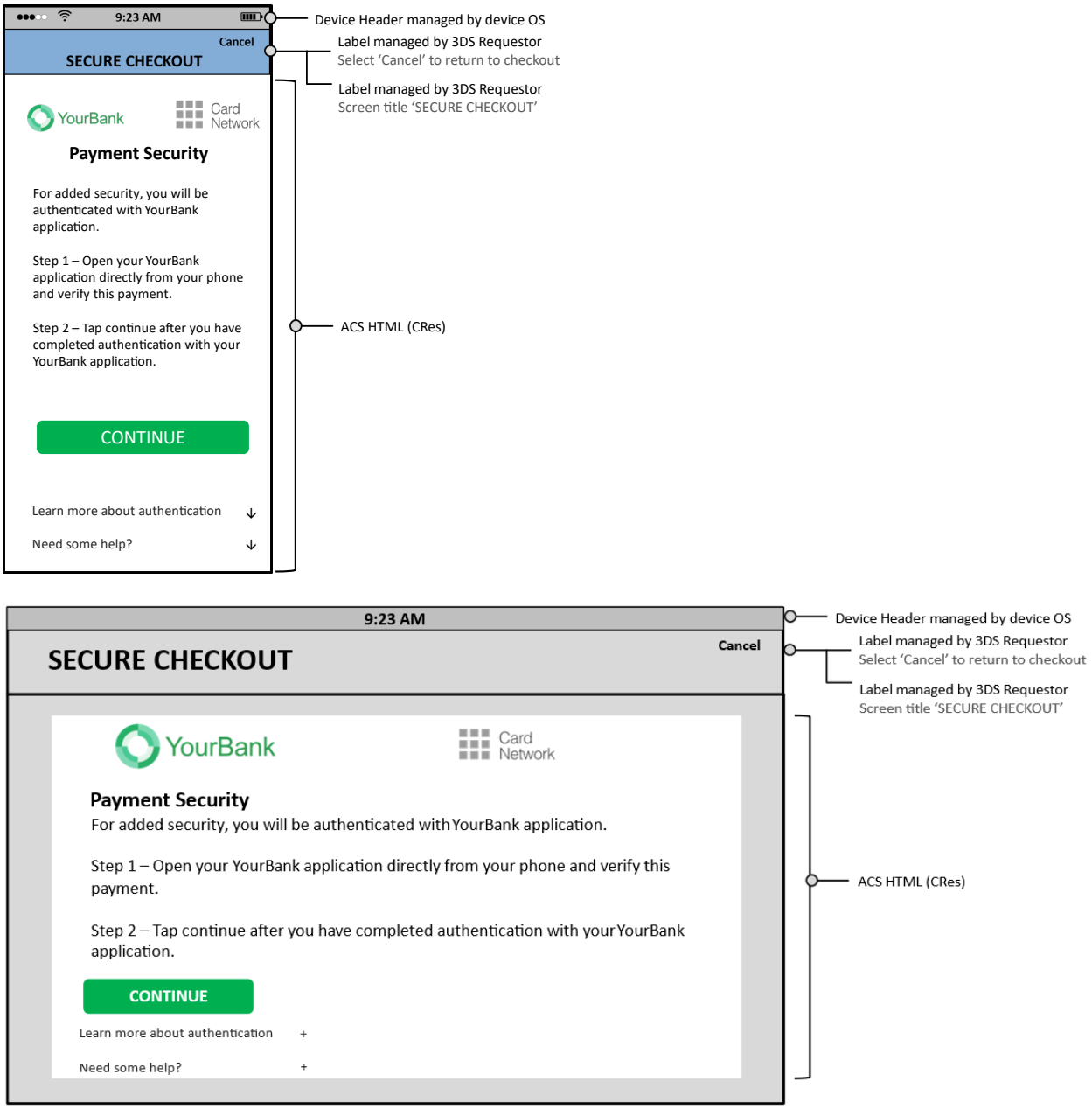


Figure 5.13: HTML User Interface



5.6 OOB Flow: App Channel – Automatic Switching to the OOB App

For version 2.3.1 of the Core Specification, it is possible to automate the switching from the 3DS Requestor App to the OOB Authentication App. This flow is also possible with version 2.2 of the Core Specification if the Bridging Message Extension with the Challenge Data object is present and supported by the ACS and the 3DS SDK.

During the challenge, the ACS instructs the Cardholder to switch from the payment/checkout page to the Authentication App using the provided button on the screen. When the Cardholder has completed the authentication, the OOB Authentication App will automatically return the Cardholder to the Challenge screen on the 3DS Requestor App.

Refer to Section 3.5 for details on automatic switching from the OOB Authentication App to the 3DS Requestor App.

Preconditions

The ACS has defined, deployed, and communicated an OOB authentication process to the Cardholder.

The ACS provides the OOB App URL to the 3DS SDK.

The OOB Authentication App can handle the 3DS Requestor App URL.

The 3DS Requestor provides the 3DS Requestor App URL to the 3DS SDK.

The 3DS Requestor App and the OOB Authentication App are on the same device.

The 3DS SDK and ACS communicate via the 3DS Requestor App URL Indicator and OOB App URL Indicator that they support the URLs for automatic switching.

Sequence Diagram

The Cardholder authenticates the transaction using an OOB Authentication App that is on the Device used for the purchase. The switching between the 3DS Requestor and the OOB Authentication App is automated.

1. The Cardholder makes a purchase and proceeds to checkout. The 3DS Requestor App displays the processing screen while waiting for the ACS's response.
2. The 3DS Requestor initiates a 3DS authentication.
3. The 3DS Server sends an AReq message.
4. The ACS responds with an ARes message requesting a challenge.
5. The 3DS Requestor triggers the 3DS SDK to proceed with a challenge. The 3DS SDK connects to the ACS.
6. The ACS provides the UI to the 3DS SDK and instructs the Cardholder to proceed with an OOB authentication. In particular, the ACS provides the OOB App URL (Universal App Link) and OOB App Label that the 3DS SDK uses to display a button to automatically switch to the OOB App.
   The ACS conveys the 3DS Requestor App URL to the OOB Authentication App.
7. The Cardholder selects the "OOB App" button, and the 3DS SDK invokes the OOB App URL that opens the OOB App. The Cardholder is automatically taken to the OOB App.

8. The Cardholder proceeds with the OOB authentication. When completed, the OOB App invokes the 3DS Requestor App URL (Universal App Link) to automatically return to the 3DS Requestor App.
9. The Cardholder does not need to select the Complete button, the 3DS SDK detects that the 3DS Requestor App is back in the foreground and sends a CReq message to the ACS (Automatic CReq). The 3DS SDK sends the “OOB complete” information to the ACS.
10. The ACS sends the results of the authentication in an RReq message through the DS to the 3DS Server, and the 3DS Server acknowledges it by sending an RRes message.
11. The ACS sends a Final CRes message to the 3DS SDK, the 3DS SDK conveys the information to the 3DS Requestor App.
12. The 3DS Requestor App displays the purchase completion information.

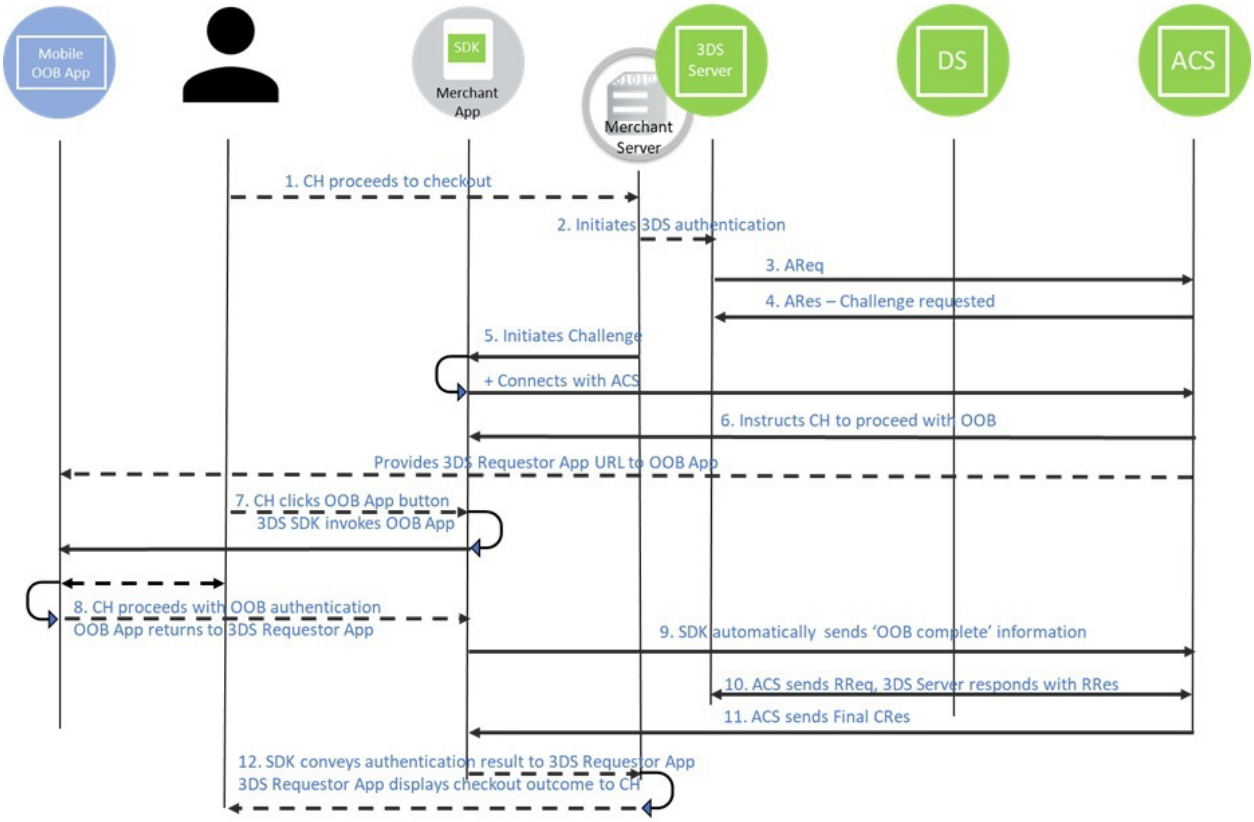
Note: After Step 9, the ACS may continue the challenge if the OOB authentication was not performed or failed, OR send the Final CRes message as shown.

Note: If the ACS receives or knows the result of the OOB authentication (pass or fail) before the Cardholder confirms completion, it may send the RReq before receiving the “OOB complete” information from the 3DS SDK.

Note: It is recommended for the ACS to display the Complete button (refer to the OOB Continuation Label) if the OOB Authentication App is on a different device or if there is a technical issue when the 3DS Requestor App URL and OOB App URL are invoked.

Note: If the 3DS SDK receives the OOB App URL but not the OOB App Label, the 3DS SDK does not return an Error Message to the ACS. The 3DS SDK does not display the OOB App URL button in the user interface, as the OOB App Label is not present. The Cardholder would need to manually switch to the OOB App.

Figure 5.14: OOB Flow App Channel - Automatic Switching to OOB App

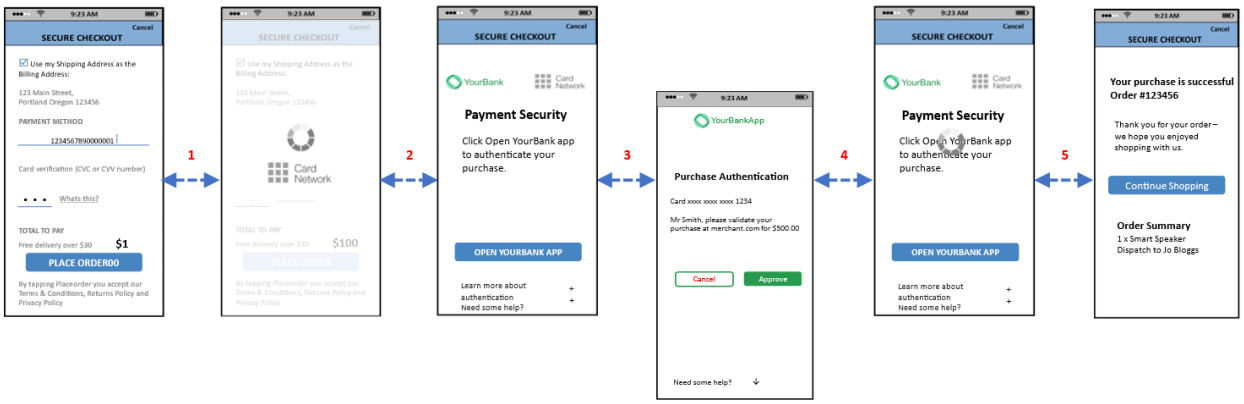


User Experience

1. The Cardholder makes a purchase and proceeds to checkout. The 3DS Requestor App displays the processing screen while waiting for the ACS's response.
2. The 3DS SDK displays the UI provided by the ACS to proceed with an OOB authentication.
3. The Cardholder selects the "Open yourbank app" button, the OOB App automatically opens and comes to the foreground.
4. When the OOB authentication is completed, the OOB App invokes the 3DS Requestor App URL, the Cardholder is automatically taken back to the 3DS Requestor App.
5. The 3DS SDK detects that it is in the UI foreground and automatically sends the OOB complete information.
6. The 3DS Requestor App displays the purchase completion information.

Note: After Step 4, the ACS may continue the challenge if the OOB authentication was not performed or if it failed, OR send the Final CRes message as shown.

Figure 5.15: User Experience: OOB Flow App Channel – Automatic Switching to OOB App



5.6.1 Technical Variant – the Device Operating System Cannot Match the OOB App URL to an Installed App

User Experience

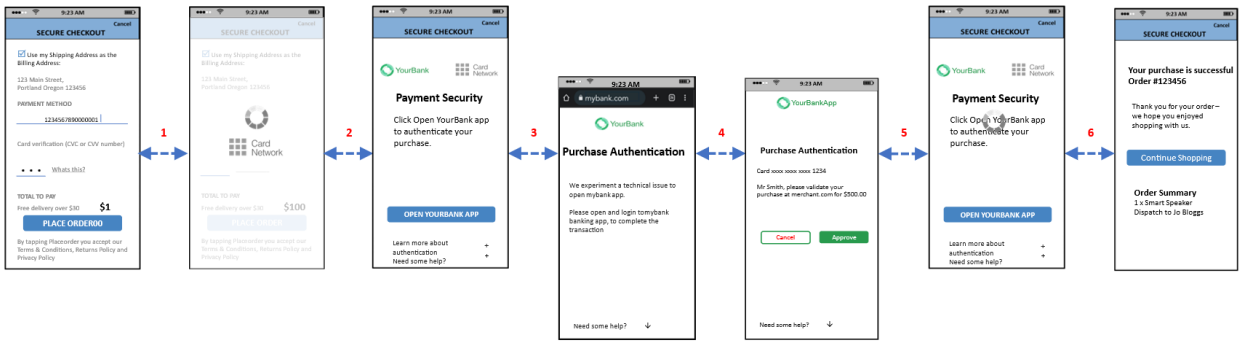
1. The Cardholder makes a purchase and proceeds to checkout. The 3DS Requestor App displays the processing screen while waiting for the ACS's response.
2. The 3DS SDK displays the UI provided by the ACS to proceed with an OOB authentication.
3. The Cardholder selects the "Open yourbank app" button.
   The 3DS SDK invokes the OOB App URL (Universal App Link), but the Device Operating System cannot resolve the URL and opens the default Device Browser.

Note: The ACS would need to provide a landing page to instruct the Cardholder to manually switch to the OOB Authentication App.

4. The Cardholder manually switches to the OOB App.
5. When the OOB authentication is completed, the OOB App invokes the 3DS Requestor App URL, the Cardholder is automatically taken back to the 3DS Requestor App.
6. The 3DS SDK detects that it is in the UI foreground and automatically sends the OOB complete information.
7. The 3DS Requestor App displays the purchase completion information.

Note: The ACS may also display the Complete button if the OOB Authentication App is on a different device.

Figure 5.16: User Experience: Device OS Cannot Match OOB App URL to App



5.6.2 Technical Variant – the OOB App URL Is Invalid or Is Based on a Custom Device Operating System

User Experience

1. The Cardholder makes a purchase and proceeds to checkout. The 3DS Requestor App displays the processing screen while waiting for the ACS's response.
2. The 3DS SDK displays the UI provided by the ACS to proceed with an OOB authentication.
3. The Cardholder selects the "Open yourbank app" button.
   The 3DS SDK invokes the OOB App URL (Universal App Link), but the Device Operating System cannot resolve the URL and returns an error to the 3DS SDK. The 3DS SDK indicates the error to the ACS in a CReq message using OOB App Status (01) and OOB Continuation Indicator (02). The ACS provides a new UI with instructions to the Cardholder to manually switch to the OOB Authentication App.

Note: This error scenario also occurs if the OOB Authentication App is on a different device.

Note: The 3DS SDK needs to interpret the error returned by the Device Operating System in order to indicate the error to the ACS in the CReq message.

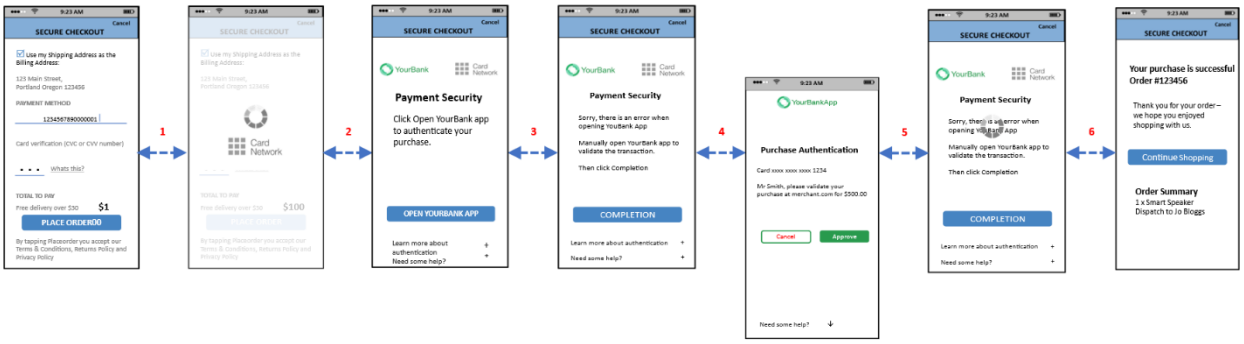
4. The Cardholder manually switches to the OOB App.
5. When the OOB authentication is completed, the OOB App invokes the 3DS Requestor App URL, the Cardholder is automatically taken back to the 3DS Requestor App.

The 3DS SDK detects that it is in the UI foreground and automatically sends the OOB complete information.

6. The 3DS Requestor App displays the purchase completion information.

Note: The ACS should display the Complete button if the OOB Authentication App is on a different device.

Figure 5.17: User Experience: OOB App URL Invalid or Based on Custom Device Operating System



5.6.3 3DS Version 2.3.1 Data Elements

Table 5.4 below lists the data elements that may be provided in relation to OOB – Automatic switching to and from the OOB App.

Table 5.4: 3DS Data Elements Related to OOB – Automatic Switching to and from the OOB App

Data Element	Description	Version
3DS Requestor App URL	3DS Requestor App declaring its URL within the CReq message so that the Authentication App can call the 3DS Requestor App after OOB authentication has occurred. Each transaction would require a unique Transaction ID by using the SDK Transaction ID.	2.3.1
3DS Requestor App URL Indicator	Indicates whether the OOB Authentication App used by the ACS during a challenge supports the 3DS Requestor App URL.	2.3.1 2.2 + Bridging Message Extension
ACS Interface	The ACS interface that the challenge presents to the Cardholder.	2.3.1
ACS UI Template	Identifies the UI Template format that the ACS first presents to the Cardholder.	2.3.1
ACS UI Type	User interface type that the 3DS SDK will render, which includes the specific data mapping and requirements.	2.3.1

Data Element	Description	Version
Authentication Method	Authentication approach that the ACS used to authenticate the Cardholder for this specific transaction.	2.3.1
OOB App Label	Label to be displayed for the link to the OOB App URL.	2.3.1 2.2 + Bridging Message Extension
OOB App Status	Status code indicating the type of problem encountered when using the OOB App URL (fail to open).	2.3.1 2.2 + Bridging Message Extension
OOB App URL	Universal App Link to an Authentication App used in the OOB authentication. The OOB App URL will open the appropriate location within the OOB Authentication App.	2.3.1 2.2 + Bridging Message Extension
OOB App URL Indicator	Indicates if the 3DS SDK supports the OOB App URL.	2.3.1 2.2 + Bridging Message Extension
OOB Continuation Indicator	Indicator notifying the ACS that the Cardholder has selected the OOB Continuation button in an OOB authentication method, or that the 3DS SDK automatically completes without any Cardholder interaction.	2.3.1 2.2 + Bridging Message Extension
OOB Continuation Label	Label to be used in the UI for the button that the Cardholder selects when they have completed the OOB authentication.	2.3.1
SDK Authentication Type	Authentication methods preferred by the 3DS SDK in order of preference.	2.3.1
SDK UI Type	Lists all UI types that the device supports for displaying specific challenge user interfaces within the 3DS SDK.	2.3.1

5.6.4 OOB User Interface for Version 2.3.1

Figure 5.18: OOB User Interface for Version 2.3.1

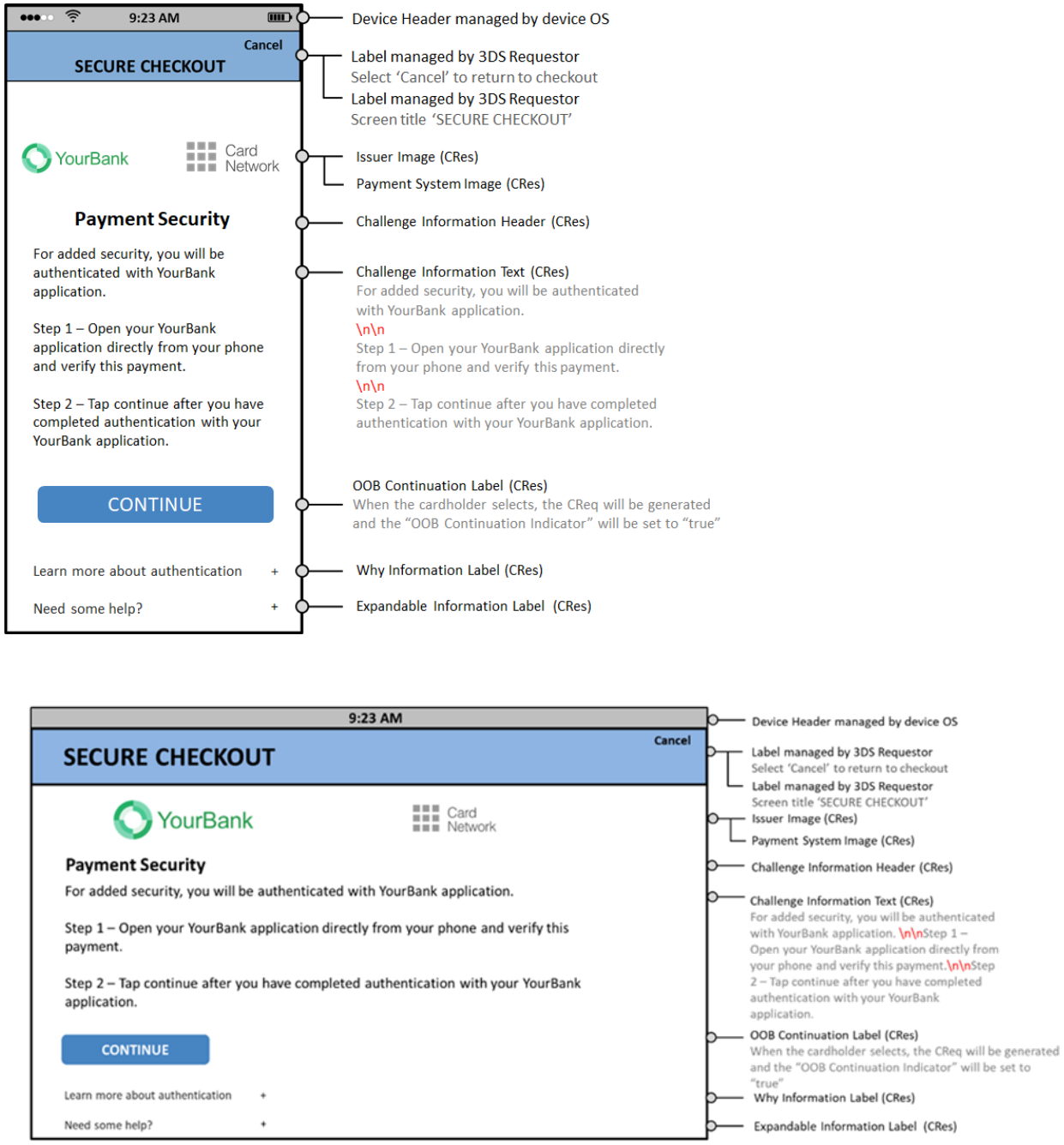


Figure 5.19: Sample OOB Native UI Template with Automatic OOB App URL Link —Portrait + Landscape

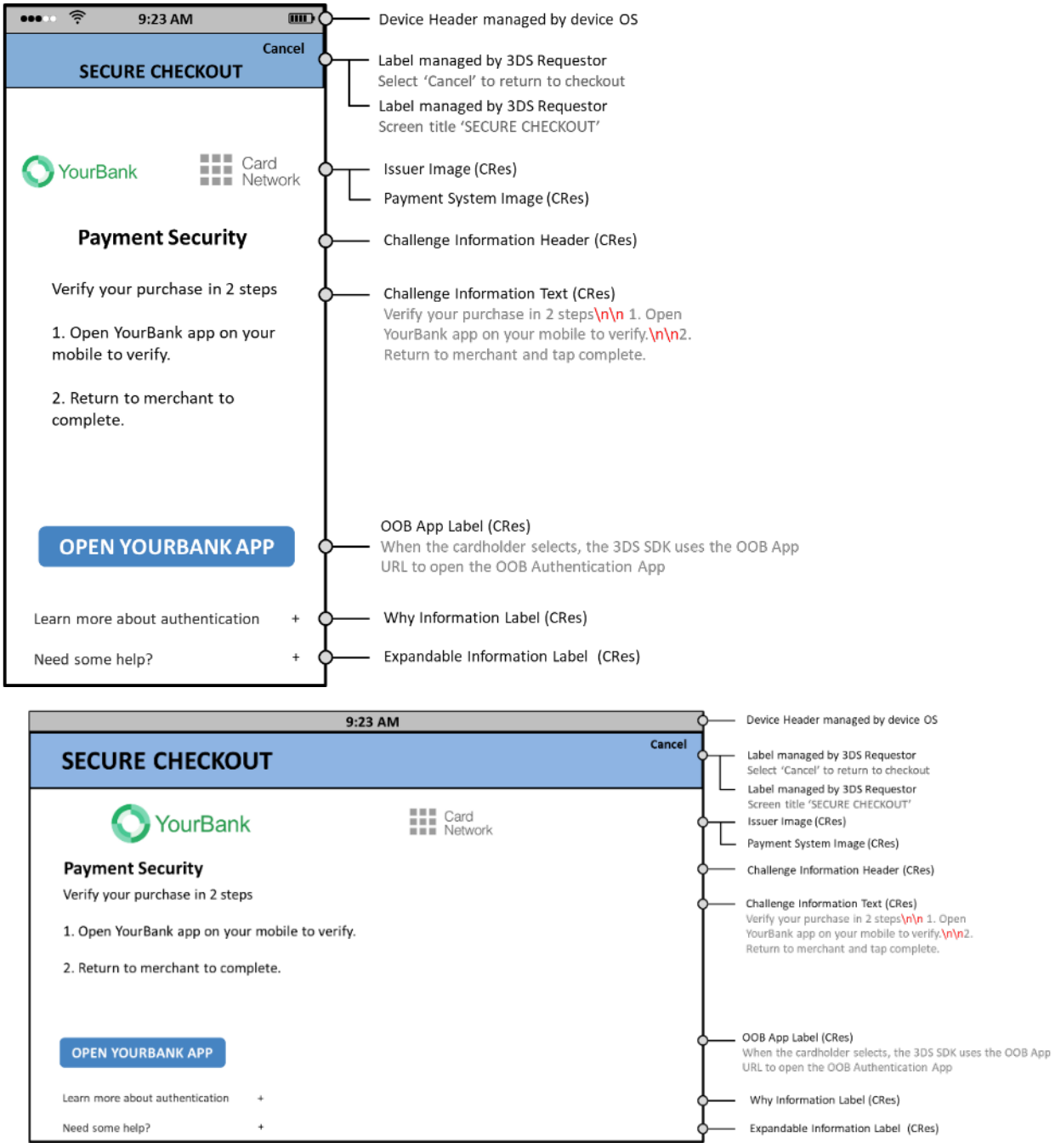


Figure 5.20: Sample OOB HTML UI Template with Complete Button —PA—Portrait + Landscape

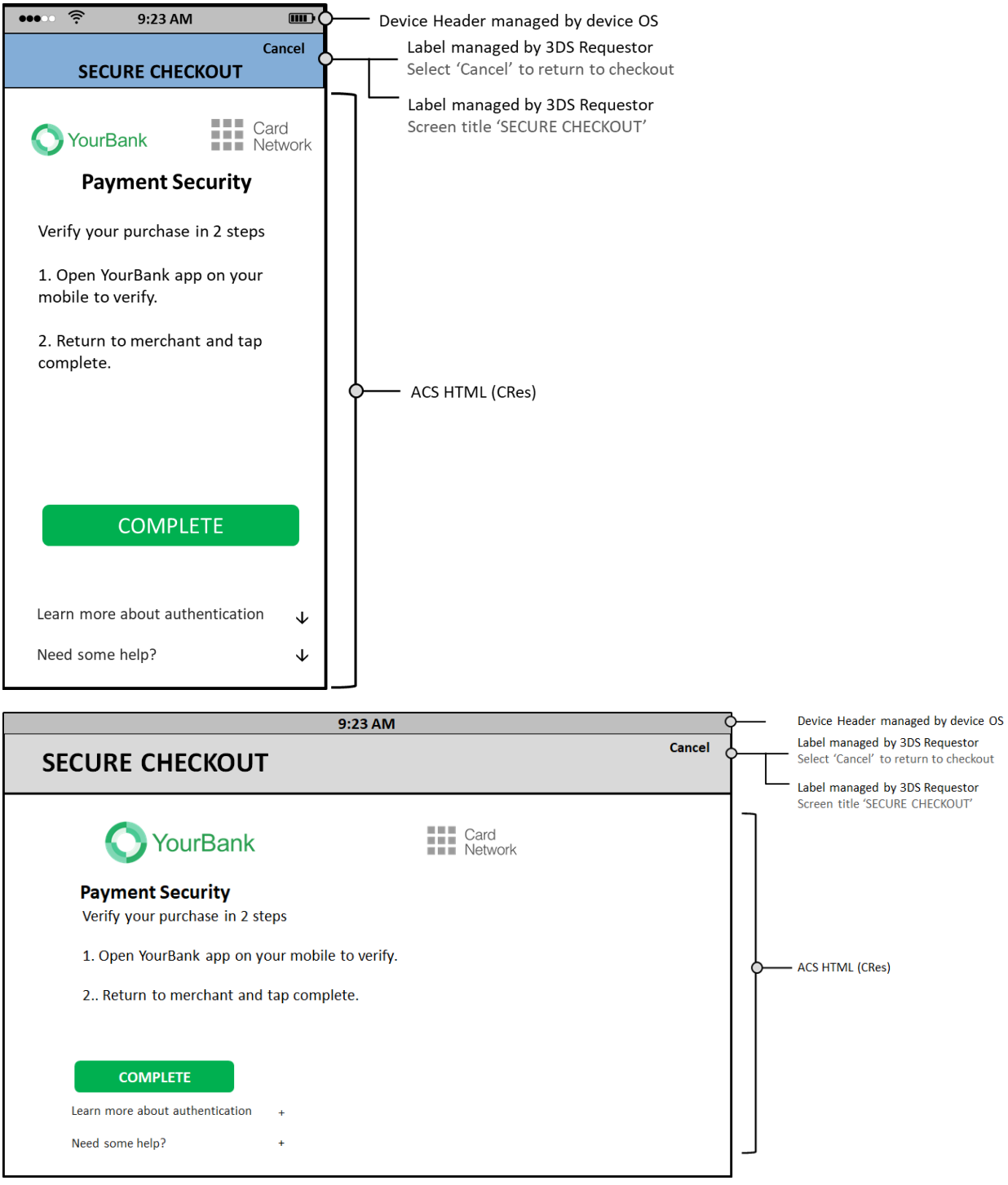
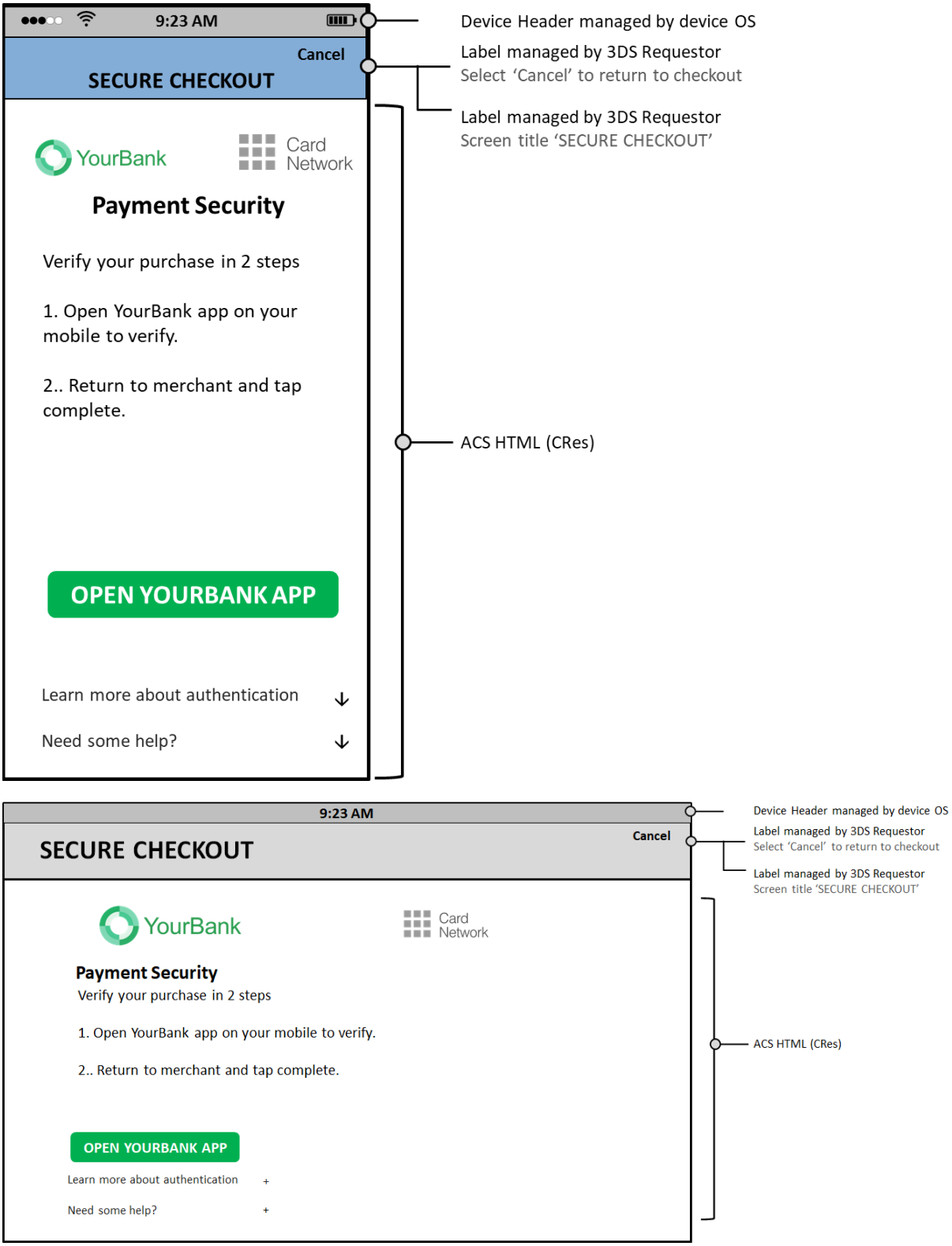


Figure 5.21: Sample OOB HTML UI Template with OOB App URL Button —PA—Portrait + Landscape



5.6.5 OOB v2.2.0 and Bridging Message Extension

Automated switching from the 3DS Requestor App to the OOB Authentication App is possible with version 2.2 of the Core Specification using the Bridging Message Extension.

The flow is similar to version 2.3.1.1 of the Core Specification described in this Section but is limited to ACS UI Type = 04 (OOB).

Note: ACS UI Type = 06 (HTML OOB) is not supported in the Bridging Message Extension.

Preconditions

The ACS and the 3DS SDK shall implement the related OOB requirements as defined in the Bridging Message Extension (refer to the Challenge Data section).

Both the ACS and the 3DS SDK shall support the Challenge Data object in the Bridging Message Extension to make use of this feature.

6 3DS Message Extensions

6.1 Business Overview

As new use cases or requirements that cannot be supported by existing data in the Core Specification emerge, data needed to support these requirements may be carried in a 3DS message extension. Message extensions are most commonly defined and specified by DSs and EMVCo.

A message extension may be defined to:

- Provide additional information about a transaction
- Add functionality to the 3DS ecosystem
- Fix an error in a 3DS protocol version.

The data in a message extension may affect the meaning of the rest of the data such that the entire message can only be understood in the context of the extension data, or it may fix an error in the message. In such cases, the DS may define the message extension as critical, and the value of the criticality attribute shall be 'true'. When a message extension is set as critical, recipients of the message shall recognise and be able to process the extension. If a 3DS component other than the DS receives a message containing a critical extension that it does not recognise, it shall treat the message as invalid.

When a message extension is non-critical (only carries optional or additional information), the receiving 3DS components that cannot process the extension shall ignore the message extension and pass it to the destination system unaltered.

Benefits by Actor

- EMVCo
  - critical changes can be made to 3DS messages in case of a major error
  - ability to enhance 3DS messages with additional data or information relevant to the industry without releasing a new specification version
- DS – ability to enhance 3DS messages with additional information to fulfil specific business requirements/cases
- ACS, 3DS Server – fast implementation of new or DS-specific features

6.2 Technical Features

The Message Extension data element is defined as an array of objects, each object carrying one message extension.

Up to 15 message extensions may be present in the Message Extension data element if required (totalling a maximum of 81920 characters).

EMVCo has currently defined five message extensions:

- Device Acknowledgement Message Extension

- Bridging Message Extension
- Attribute Verification Message Extension
- Travel Industry Message Extension
- Payment Token Message Extension

Note: The following sections focus on the new and more complex message extensions, specifically the Bridging Message Extension and the Attribute Verification Message Extension. Established message extensions (such as the Travel Industry and Payment Token Message Extensions), which have been in use for an extended period, are more straightforward as they facilitate data transfer from the 3DS Server to the DS or ACS without necessitating specific processing or implementation of requirements.

6.3 Device Acknowledgement Message Extension

6.3.1 Business Overview

The Split-SDK was introduced with v2.3.1 of the Core Specification, but some 3DS Requestors were willing to operate the Split-SDK in a 3DS v2.1 or v2.2 environment. The Device Acknowledgement Message Extension enables the 3DS Requestors to communicate to the ACS that the transaction originates from the Split-SDK for a v2.2. authentication.

The Device Acknowledgement Message Extension carries data related to the Split-SDK to the ACS for use in risk decisioning.

In response to the Device Acknowledgement Message Extension in the AReq message, the ACS can acknowledge data received in the EMV 3-D Secure SDK Device Information, particularly for Device Information Data Version 1.3 or higher.

Benefits by Actor

- Merchant – ability to operate a Split-SDK for all 3DS protocol versions
- Issuer – understands that the transaction originates from a Split-SDK, and adjusts the risk analysis and challenge accordingly (for more information, refer to the Split-SDK section).
- DS
  - monitors the use of Split-SDK across the different protocol versions
  - monitors what Device Information versions are supported by the ACS.

6.3.2 Technical Features

The Device Acknowledgement Message Extension is present in the AReq message version 2.2.0 for the App-based flow, when the transaction originates from a Split-SDK.

The Device Acknowledgement Message Extension is only present in the ARes message version 2.2.0 if the ACS:

- acknowledges the supported Device Information, and/or
- provides additional data to the Split-SDK in case of a challenge (Transaction Status = C).

Table 6.1 below lists the data elements that may be provided in relation to the Device Acknowledgement Message Extension.

Table 6.1: 3DS Data Elements Related to the Device Acknowledgement Message Extension

Data Element/Field Name	Description	Version
Extension Version Number	Version number of the message extension.	2.2

Default-SDK Type	Indicates the characteristics of a Default-SDK.
	SDK Variant: SDK implementation characteristics
	Wrapped Indicator: If the Default-SDK is embedded as a wrapped component in the 3DS Requestor App
	Example:
	"defaultSdkType":{
	"sdkVariant":"01",
	"wrappedInd":"Y"
	}
		2.2

SDK Authentication Type	Authentication methods preferred/supported by the 3DS SDK in order of preference.	2.2
SDK Server Signed Content	Contains the JWS object (represented as a string) created by the Split-SDK Server for the AReq message.	2.2
SDK Signature Timestamp	Date and time indicating when the 3DS SDK generated the Split-SDK Server Signed Content converted into UTC.	2.2
SDK Type	Indicates the type of 3DS SDK. This data element provides additional information to the DS and ACS to determine the best approach for handling the transaction.	2.2

Split-SDK Type	Indicates the characteristics of a Split-SDK.
	Split-SDK Variant: Implementation characteristics of the Split-SDK client
	Limited Split-SDK Indicator: If the Split-SDK client has limited capabilities
	Example:
	"splitSdkType":{
	"sdkVariant":"01",
	"limitedInd":"Y"
	}
		2.2

Split-SDK Server ID	DS-assigned Split-SDK Server identifier. Each DS can provide a unique ID to each Split-SDK Server on an individual basis.	2.2

Data Element/Field Name	Description	Version
Authentication Method	Indicates the authentication types that the Issuer will use to challenge the Cardholder when in the ARes message or what was used by the ACS when in the RReq message.	2.2
Device Information Recognised Version	Indicates the highest Data Version of the Device Information supported by the ACS.	2.2
Device User Interface Mode	Indicates the user interface mode that the ACS will present to the Cardholder for a challenge.	2.2

6.4 Bridging Message Extension

6.4.1 Business Overview

The Bridging Message Extension enhances the existing 3DS version 2.2.0 specifications by enabling the implementation of features that are incorporated in the 3DS version 2.3.1.1 specification. In addition, during the migration to version 2.3.1.1, support of the Bridging Message Extension will increase consistency of the key 3DS features provided in version 2.2.0 and version 2.3.1.1.

The Bridging Message Extension covers different version 2.3.1.1 features and defines four sets of data:

- Recurring Data, which enhances the information exchange between the 3DS Requestor and the ACS regarding the recurring transaction information, and a better communication to the Cardholder in case of a challenge.
- Challenge Data, which enables automation of the switching between the 3DS SDK and an Out-of-Band (OOB) authentication application.
- File URL Data, which enables the 3DS Server to retrieve Card Range Data via a file download from the DS rather than from PRes messages.
- Additional Data, which is used by the 3DS Server, the ACS and the DS to share additional information from 3DS version 2.3.1.1 during a 3DS authentication.

6.4.2 Technical Features

Recurring Data, Challenge Data, Additional Data and File URL Data contain new data or new values for existing data in the AReq/ARes, PReq/PRes or RReq messages for version 2.2.0.

Note: As there are multiple versions of the Bridging Message Extension, the Extension Version Number must be kept the same across all message exchanges during a 3DS transaction. When responding, the recipient of the Bridging Message Extension must use the same Extension Version Number for the message pair (AReq/ARes, CReq/CRes, PReq/PRes) as the one received. If the recipient does not support that Extension Version Number, it must not include the Bridging Message Extension in its response.

Recurring Data

Recurring Data is provided for recurring and instalment transactions. The use cases and the technical capabilities of 3DS with recurring data elements are presented in Section 3.

Table 6.2 below lists the recurring data elements that may be provided in relation to the Bridging Message Extension.

Table 6.2: Recurring Data Elements Related to the Bridging Message Extension

Data Element/Field Name	Description	Version
Recurring Amount	Recurring amount in minor units of currency with all punctuation removed.	2.2
Recurring Currency	Currency in which the Recurring Amount is expressed.	2.2
Recurring Currency Exponent	Minor units of currency as specified in the ISO 4217 currency exponent.	2.2
Recurring Date	Effective date of the new authorised amount following the first/promotional payment in a recurring or instalment transaction.	2.2
Recurring Expiry	Date after which no further authorisations are performed.	2.2
Recurring Frequency	Indicates the minimum number of days between authorisations for a recurring or instalment transaction.	2.2
Recurring Indicator	Indicates whether the recurring or instalment payment has a fixed or variable amount and frequency. The Recurring Indicator object contains: • the Amount Indicator • the Frequency Indicator	2.2

Challenge Data

Using Challenge Data, the ACS and 3DS SDK can automate switching from the 3DS Requestor App to the OOB Authentication App.

Challenge Data is only provided in CReq/CRes messages for the App-based flow when the Message Version Number is 2.2.0 (not 2.1.0).

The use cases and the technical details of switching from the 3DS Requestor App to the OOB Authentication App are presented in Section 5.

Note; The ACS shall implement Req 401, limited to ACS UI Type = 04, and the 3DS SDK shall implement Req 399, Req 400, Req 403, Req 404, Req 406, Req 407, Req 408 and Req 409, limited to ACS UI Type = 04 to enable automatic switching from the 3DS Requestor App to the OOB Authentication App.

Challenge Data is only supported for ACS UI Type = 04 (OOB), ACS UI Type = 06 (HTML OOB) is NOT supported.
For the Challenge Data Entry Masking, the ACS requests the Challenge Data Entry to be masked by setting Challenge Data Entry Masking to Y (limited to ACS UI Type = 01). The 3DS SDK shall implement Challenge Data Entry Masking (Figures 44 and 45, Table A.26).

Table 6.3 below lists the challenge data elements that may be provided in relation to the Bridging Message Extension.

Table 6.3: Challenge Data Elements Related to the Bridging Message Extension

Data Element/Field Name	Description	Version
Challenge Data Entry Masking	Indicates that the 3DS SDK shall mask the data entered by the Cardholder.	2.2
OOB App Label	Label to be displayed for the link to the OOB App URL.	2.2
OOB App Status	Status code indicating the type of problem encountered when using the OOB App URL.	2.2
OOB App URL	Universal App Link to an authentication app used in the OOB authentication. The OOB App URL will open the appropriate location within the OOB Authentication App.	2.2
OOB App URL Indicator	Indicates if the 3DS SDK supports the OOB App URL.	2.2
OOB Continuation Indicator	Indicator notifying the ACS that the Cardholder has selected the OOB Continuation button in an OOB authentication method, or that the 3DS SDK automatically completes without any Cardholder interaction.	2.2

File URL Data

File URL Data is used in the PReq/PRes message when the Message Version Number is 2.2.0. If supported by both the 3DS Server and the DS, the 3DS Server downloads a file containing the Card Range data.

Table 6.4 below lists the File URL data elements that may be provided in relation to the Bridging Message Extension.

For additional details and processing requirements refer to Section 5.6 in version 2.3.1.1 of the Core Specification.

Table 6.4: File URL Data Elements Related to the Bridging Message Extension

Data Element/Field Name	Description	Version
Card Range Data Download Indicator	Indicates if the 3DS Server supports Card Range Data from a file. Note: If present, this field contains the value Y.	2.2
Card Range Data File URL	Fully Qualified URL of the DS File containing the Card Range Data for download. Note: When the Card Range Data File URL is present, the file contains the entire Card Range Data, and the 3DS Server ignores any Card Range Data and Serial Number present in the PRes message.	2.2

Additional Data

Using the Additional Data, the ACS, 3DS Server and DS may share additional information to improve the authentication or error reporting, indicating, for example:

- whether the OOB Authentication App used by the ACS supports the 3DS Requestor App URL
- the Acquirer Country Code
- the Authentication Method used by the ACS
- the Card Security Code
- the Device Information Version supported by the ACS
- the Transaction Challenge Exemption applied by the ACS
- the Challenge Error Reporting – detailed information in case of an error in the CReq/CRes messages
- the reason for cancelling the Challenge.

Note: To provide the Challenge Cancelation Indicator, the ACS shall implement the updates to Section 5.9.5 (ACS CReq Message Error Handling—01-APP) in version 2.3.1.1 of the Core Specification.

Table 6.5 below lists additional data elements that may be provided in relation to the Bridging Message Extension.

Table 6.5: Additional Data Elements Related to the Bridging Message Extension

Data Element/Field Name	Description	Version
3DS Requestor App URL Indicator	Indicates whether the OOB Authentication App used by the ACS during a challenge supports the 3DS Requestor App URL.	2.2

Data Element/Field Name	Description	Version
3DS Requestor Authentication Indicator	Indicates the type of Authentication Request. This data element provides additional information to the ACS to determine the best approach for handling an Authentication Request.	2.2
3RI Indicator	Indicates the type of 3RI request. This data element provides additional information to the ACS to determine the best approach for handling a 3RI request.	2.2
Acquirer Country Code	The code of the country where the acquiring institution is located (in accordance with ISO 3166-1). The DS may edit the value provided by the 3DS Server.	2.2
Acquirer Country Code Source	This data element is populated by the system setting the Acquirer Country Code. The DS may edit the value provided by the 3DS Server.	2.2
Authentication Method	Indicates the list of authentication types the Issuer will use to challenge the Cardholder, when in the ARes message, or what was used by the ACS, when in the RReq message. Note: For 03-3RI, only present for Decoupled Authentication.	2.2
Browser Screen Color Depth	Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Obtained from the Cardholder browser using the screen.colorDepth property. Refer to Section A.6 in the Core Specification v2.3.1.1 for more details.	2.2
Card Security Code	Three- or four-digit security code printed on the card.	2.2
Card Security Code Status	Enables the communication of Card Security Code Status between the ACS, the DS and the 3DS Requestor.	2.2
Card Security Code Status Source	This data element will be populated by the system setting the Card Security Code Status.	2.2
Challenge Cancelation Indicator	Indicator informing the ACS and the DS that the authentication has been cancelled. Note: The Additional Data object is not valid for the CReq/CRes messages. Therefore, the Challenge Cancelation Indicator may only be present in the RReq message.	2.2
Challenge Error Reporting	Copy of the Erro Message sent or received by the ACS in case of error in the CReq/CRes messages.	2.2
Device Information Recognised Version	Indicates the highest Data Version of the Device Information supported by the ACS.	2.2

Data Element/Field Name	Description	Version
Transaction Challenge Exemption	Exemption applied by the ACS to authenticate the transaction without requesting a challenge. Note: The accepted values match the values of the 3DS Requestor Challenge Indicator.	2.2
Transaction Characteristics	Indicates to the ACS specific transactions identified by the Merchant. Refer to Merchant Risk Indicator in the *Core Specification v2.3.1.1*	2.2

6.5 Attribute Verification Message Extension

6.5.1 Business Overview

The Attribute Verification Message Extension allows a 3DS Merchant to isolate a specific piece of data about a user and have it verified by a trusted source in the 3DS ecosystem. The purpose of the extension is to satisfy requirements imposed by global regulations mandating that certain types of attributes be individually confirmed to make purchases, set up new subscriptions, access adult content on the internet, or even create new social media accounts. Its intention is to minimise the amount of data necessary to be shared amongst parties in order to satisfy these requirements, while leveraging the existing and widely adopted 3DS infrastructure for ease of implementation.

Examples of verifiable attributes include but are not limited to:

- Age – for purchasing restricted goods such as alcohol and tobacco or creating a new social media account.
- Government ID numbers – for placing online bets via gambling platforms.
- Residency – for purchases where verification of residency is required to set up a recurring subscription.
- Billing/Shipping addresses – for Merchants that may require additional verification on an address for higher purchase amount transactions.

Benefits by Actor

- Merchant – ability to comply with Cardholder attribute verification regulations to complete purchases, set up accounts, view internet content, or place online gambling bets
- Issuer
    - ability to leverage the account data it already has for its Cardholders in order to perform attribute verification
    - ability to use its existing 3DS rails for attribute verification
- Cardholder

- decreases the amount of data about the Cardholder that needs to be shared to complete the transaction
- allows the Cardholder to use their familiar 3DS authentication experience to complete attribute verification

6.5.2 Technical Features

Overview

The Attribute Verification Message Extension is a non-critical message extension, which means that support for this feature is not mandatory. It is applicable for the Browser, App and 3RI message channels. It also applies to both Payment and Non-Payment message categories. The Verification Request Data is sent as part of the AReq message, and the Verification Response Data is sent as part of the ARes and RReq messages if a challenge is required by the ACS.

The requestor may choose from predefined attributes within the extension itself that do not currently exist in the Core Specification. Examples of these attributes are age, date of birth, citizenship, and ID number. Requestors can also refer to a set of existing data elements within the AReq message for attribute verification. Examples of these are cardholder name, billing address, and home phone number. The Attribute Verification Message Extension allows for multiple attributes to be requested within the same message.

The requestor can select the match type, which is the method needed to be performed against the value being requested to be verified. These are the mathematical operators: greater than; less than; greater than or equal; less than or equal; equal and not equal (match type values 01–06, respectively). The final match type is compare (match type value 07). For an attribute request where the value to be verified is numeric, the requestor selects a mathematical operator to be applied to the request. For non-numeric values, the requestor uses the compare match type.

The validating system should return result codes indicating that the verification matched successfully (or true); not matched (or false); attribute requested is supported for verification but could not complete the request; or attribute requested is not supported for verification (result code values 01, 02, 04, 05, respectively) for match types that are numeric operators. The validating system should return result codes that the verification matched successfully (or true); not matched (or false); matched partially; attribute requested is supported for verification but could not complete the request; or attribute requested is not supported for verification (result code values 01, 02, 03, 04, 05, respectively) for the match type compare.

The validating system has the option to respond with the outcome of the Attribute Verification Request in either the ARes or the RReq, depending on whether challenge processing is necessary for a response. All attribute results must be responded to in the same message. If the verifying system chooses to respond frictionlessly, it will set the Verification Response Indicator = N (All verification responses are only provided before challenge processing) in the ARes message along with the Verification Response Object. If the verifying system chooses to respond after a Cardholder challenge, it will set the Verification Response Indicator = Y in the ARes message only, and it will then send the Verification Response Object in the RReq, which will contain the results of the attributes requested to be verified.

In the case of merchant-initiated SPC, where the validating system requires a challenge to respond to the Attribute Verification Request, the validating system will set the Verification Response Indicator = Y in the first ARes message that sets up the SPC challenge for the Merchant. The requestor will repeat the Verification Request Object in the second AReq message. The validating system will then send the Verification Response Object with the attribute verification outcome in the second ARes message of the merchant-initiated SPC challenge.

If the verifying system suspects that an attribute verification request has been submitted with fraudulent intent, it will respond back with a non-successful Result Code in the Verification Response Object and a non-successful Transaction Status in the core authentication message. It will also set the Transaction Status Reason in the core authentication message = 11 (Suspected fraud).

Preconditions

It is presumed that the message extension is supported by the 3DS Server, the ACS, and the DS.

The Frictionless and Challenge Flow sequence diagrams below show the 3DS Server as the system sending the Attribute Verification Request, and the ACS as the system sending the Attribute Verification Response. However, the extension allows for the DS to be the system performing either the request or the verification. In this case, the flow remains the same, except it is the DS that populates the Verification Request Data in the AReq message or the Verification Response Data in the ARes/RReq messages. If the DS is the verifying system, then the Verification Response Indicator will always be set to N in the ARes (all attributes will be verified in the ARes).

Note: Since the attribute verification may involve private or sensitive cardholder data, and to comply with local regulations, the 3DS Requestor and/or ACS should obtain cardholder consent before proceeding with the verification.

3DS Data Elements Related to the Attribute Verification Extension

The data elements listed in Table 6.6, Table 6.7, Table 6.8 and Table 6.9 below are part of the Attribute Verification Message Extension. For additional information, please refer to the EMV 3-D Secure Attribute Verification Message Extension.

Table 6.6: Attribute Verification Data Elements

Data Element/ Field Name	Description	Version
Assigned Extension Group Identifier	A unique identifier for the extension.	2.3.1 2.2
Criticality Indicator	A Boolean value indicating whether the recipient must understand the contents of the extension to interpret the entire message.	2.3.1 2.2
Data	The data carried in the extension.	2.3.1 2.2

Data Element/ Field Name	Description	Version
Extension Name	The name of the extension data set as defined by the extension owner.	2.3.1 2.2

Table 6.7: Data

Data Element/ Field Name	Description	Version
Verification Request Data	The data specific to the Verification Request.	2.3.1 2.2
Verification Response Data	The data specific to the Verification Response.	2.3.1 2.2
Verification Response Indicator	Indicates whether the verification response will be provided after challenge processing. Note: For Merchant-initiated SPC, and if Verification Response Indicator = Y in the initial ARes message, the 3DS Server repeats the verification request in the second AReq message.	2.3.1 2.2
Extension Version Number	Version number of the message extension.	2.3.1 2.2

Table 6.8: Verification Request Data

Data Element/ Field Name	Description	Version
Attribute Name	Indicates the type of attribute verification requested.	2.3.1 2.2
Match Type	Indicates the verification method needed.	2.3.1 2.2
Match Value	Indicates the value against which the verification will be performed	2.3.1 2.2
Core Attribute Field	Indicates the field name present in the core message against which the verification will be performed.	2.3.1 2.2

Table 6.9: Verification Response Data

Data Element/ Field Name	Description	Version
Attribute Name	Indicates the type of attribute verification requested.	2.3.1 2.2
Core Attribute Field	Indicates the field name present in the core message against which the verification was performed.	2.3.1 2.2
Verification Response Source	This data element will be populated by the system setting Verification Response.	2.3.1 2.2
Result Code	Status of the verification request.	2.3.1 2.2
Reason Code	Provides information on why the Result Code field has the specified value.	2.3.1 2.2
Verification Method	Provides information on what method was used by the source for the attribute verification.	2.3.1 2.2

Attribute Verification Frictionless Approval Flow (App-Based, Browser-Based or 3RI Channel)

Overview

In the attribute verification frictionless approval flow, the 3DS Server provides the relevant transaction data and Attribute Verification Request Data in the AReq. The ACS determines that the information provided is sufficient to make a decision on the Attribute Verification Request and sends the applicable Result Code back to the requestor in the Verification Response Data in the ARes. The ACS will also set the Verification Response Indicator = N, signalling to the requesting system that all attributes will be verified in the ARes.

Note: The decision on the Attribute Verification Request is independent of whether the authentication decision is completed frictionlessly or via a challenge. The ACS can send the Attribute Verification Response decision in the ARes, but the authentication is still challenged by the ACS.

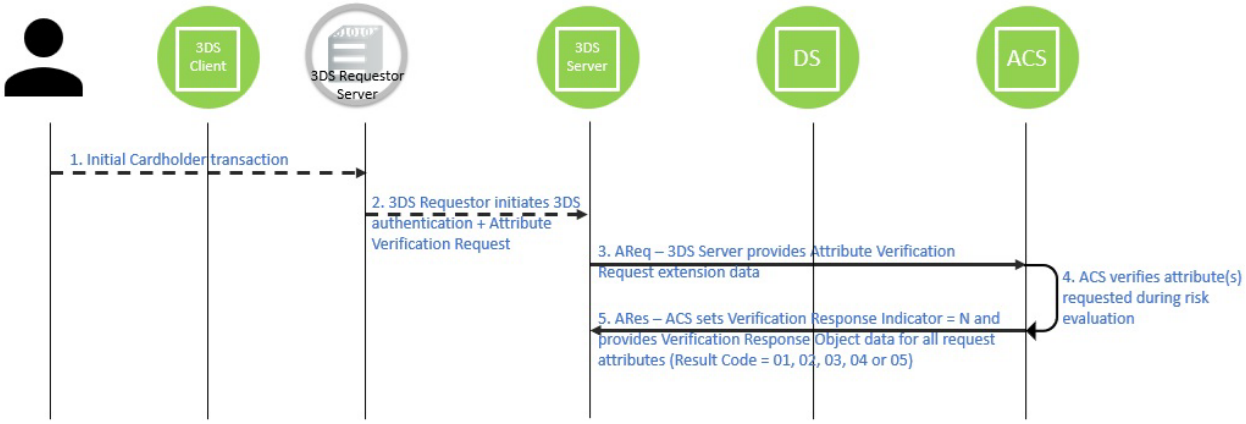
Sequence Diagram

1. The Cardholder makes a purchase or authentication request that requires an attribute verification.
2. The 3DS Requestor initiates a 3DS authentication and provides details about the purchase or non-payment request, along with the attribute to be verified.
3. The 3DS Server sends an AReq message including the Attribute Verification Message Extension populated with the Verification Request Data, specifying the attribute(s) to be verified and the type of verification method to be applied to the attribute(s).

a. If the attribute is a data element that **does not** exist in the Core Specification (i.e., Attribute name = 01/Age, 02/Date of Birth, 03/Citizenship, 04/ID Number), then the Match Value is populated for the attribute.
b. If the attribute is a data element that **does** exist in the Core Specification (i.e., Attribute name = 05/Core Attribute), then the Core Attribute Field is populated indicating the field name present in the core message that the validating system will need to use to perform the verification.

4. As part of its standard risk assessment, the ACS verifies the attribute(s) being requested and responds setting the Verification Response Indicator = N and the Verification Response Data including the result of the attribute verification in the ARes message (Result Code = 01/Matched successfully or true, 02/Not matched or false, 03/Matched partially, 04/Attribute is supported but could not be verified, or 05/Attribute requested is not supported for verification).

Figure 6.1: Frictionless Flow with Attribute Verification Extension



Attribute Verification Challenge Approval Flow (App-Based, Browser-Based or 3RI Channel)

Overview

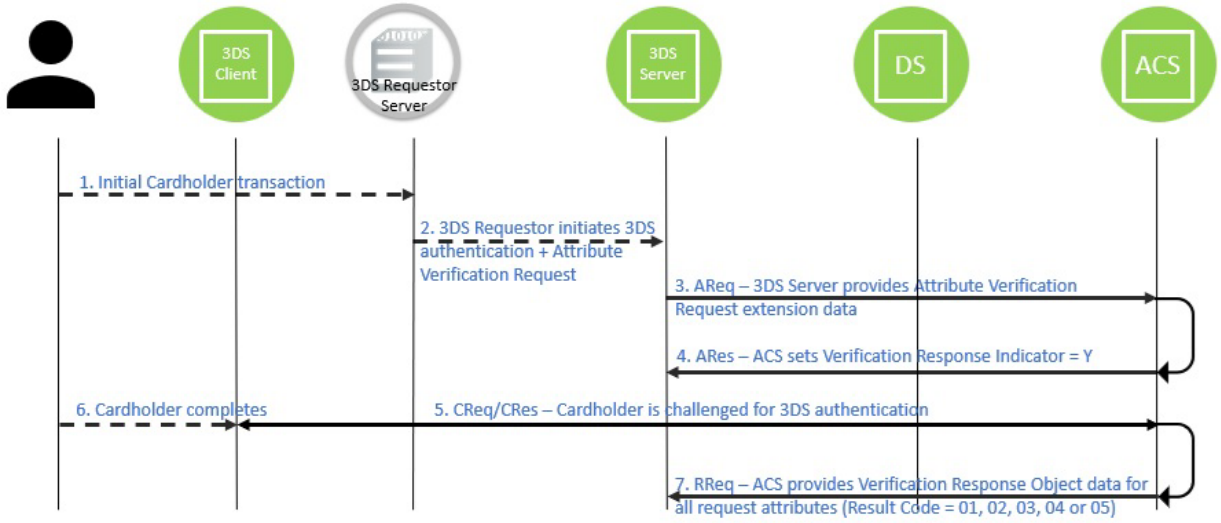
For the attribute verification challenge approval flow, the 3DS Server provides the relevant transaction data and Attribute Verification Request Data in the AReq message. The ACS determines that the Cardholder needs to complete a challenge before confirming the Attribute Verification Request. The ACS returns a response to the requestor in the ARes message that the outcome of the Attribute Verification Request will be provided in the Results Request (RReq) message by setting the Verification Response Indicator = Y. The Cardholder is challenged, and the ACS provides the Verification Response Data object(s) containing the attribute Result Code(s) to the requestor in the RReq message.

Note: If the ACS determines that the Attribute Verification Request must be challenged in order to respond with a decision, then the AReq also goes through the Challenge Flow (Transaction Status = C/D/S in the ARes message).

Sequence Diagram

1. The Cardholder makes a purchase or an Authentication Request that requires an attribute verification.
2. The 3DS Requestor initiates a 3DS authentication and provides details about the purchase or non-payment request, along with the attribute to be verified.
3. The 3DS Server sends an AReq message including the Attribute Verification Message Extension populated with the Verification Request Data specifying the attribute(s) to be verified and the type of verification method to be applied to the attribute(s).
   a. If the attribute is a data element that **does not** exist in the Core Specification (i.e., Attribute name = 01/Age, 02/Date of Birth, 03/Citizenship, 04/ID Number), then the Match Value is populated for the attribute.
   b. If the attribute is a data element that **does** exist in the Core Specification (Attribute name = 05/Core Attribute), then the Core Attribute Field is populated indicating the field name present in the core message that the validating system will use to perform the verification.
4. As part of its standard risk assessment of the AReq, the ACS determines that the attribute(s) being requested require(s) a challenge to be performed to provide a result of the attribute verification. The ACS sends the ARes message with Transaction Status = C/D/S along with the Verification Response Indicator set to Y (all verification responses are provided after challenge processing).
5. The 3DS Requestor Environment and the ACS proceed with the challenge.
6. The Cardholder completes the challenge.
7. The ACS provides the outcome of the authentication in the RReq message and includes the Result Code(s) for the attribute verification in the Attribute Verification Response data (Result Code = 01/Matched successfully (or true), 02/Not matched (or false), 03/Matched partially, 04/Attribute is supported but could not be verified, or 05/Attribute requested is not supported for verification).

Figure 6.2: Attribute Verification Challenge Approval Flow



6.5.3 Use Cases

The tables below specify the data required for each use case.

Preconditions

The verifying system successfully confirmed the attribute(s) requested.

The verifying party is the ACS.

Use Case 1: Age Verification (verifying that the Cardholder is at least 18 years old)

Attribute Verification Request Data	Attribute Verification Response Data
• Attribute Name = 01 (Age) • Match Type = 04 (Greater than or equal) • Match Value = 18 (Minimum age of user to be confirmed)	• Attribute Name = 01 (Age) • Verification Response Source = 01 (ACS) • Result Code = 01 (Matched successfully (or true)) • Reason Code = 01 (Data available and verified successfully)

Use Case 2: Date of Birth Verification (verifying that the Cardholder was born on 13 December 1989)

Attribute Verification Request Data	Attribute Verification Response Data
• Attribute Name = 02 (Date of Birth) • Match Type = 05 (Equal) • Match Value = 19891213 (Date of birth of cardholder to be confirmed in YYYYMMDD format)	• Attribute Name = 02 (Date of Birth) • Verification Response Source = 01 (ACS) • Result Code = 01 (Matched successfully (or true)) • Reason Code = 01 (Data available and verified successfully)

Use Case 3: Citizenship Verification (verifying that the Cardholder is a citizen of the United States)

Attribute Verification Request Data	Attribute Verification Response Data
• Attribute Name = 03 (Citizenship) • Match Type = 05 (Equal) • Match Value = 840 (ISO 3166-1 numeric country code for USA)	• Attribute Name = 03 (Citizenship) • Verification Response Source = 01 (ACS) • Result Code = 01 (Matched successfully (or true)) • Reason Code = 01 (Data available and verified successfully)

Use Case 4: ID Number Verification (verifying a government-issued ID number of the Cardholder)

Attribute Verification Request Data	Attribute Verification Response Data
• Attribute Name = 04 (ID number) • Match Type = 05 (Equal) • Match Value = 1111222233334444 (example value of ID number to be verified)	• Attribute Name = 02 (ID number) • Verification Response Source = 01 (ACS) • Result Code = 01 (Matched successfully (or true)) • Reason Code = 01 (Data available and verified successfully)

Use Case 5: Cardholder Name Verification (verifying that the submitted name matches what is on file for the Cardholder)

Attribute Verification Request Data	Attribute Verification Response Data
• Attribute Name = 05 (Core attribute) • Match Type = 07 (Compare) • Core Attribute Field = `cardholderName` (Field in the Authentication Request message the verifying system will use to perform the verification)	• Attribute Name = 05 (Core attribute) • Core Attribute Field = `cardholderName` (Field in the Authentication Request message the verifying system will use to perform the verification) • Verification Response Source = 01 (ACS) • Result Code = 01 (Matched successfully (or true)) • Reason Code = 01 (Data available and verified successfully)

Use Case 6: Multiple Attributes Requested in a Single Message (verifying a Cardholder's age and name in the same Attribute Verification Request)

Attribute Verification Request Data	Attribute Verification Response Data
Verification Request Object 1 • Attribute Name = 01 (Age) • Match Type = 04 (Greater than or equal)	**Verification Response Object 1** • Attribute Name = 01 (Age) • Verification Response Source = 01 (ACS)

• Match Value = 18 (Minimum age of user to be confirmed) **Verification Request Object 2** • Attribute Name = 05 (Core attribute) • Match Type = 07 (Compare) • Core Attribute Field = `cardholderName` (Field in the Authentication Request message the verifying system will use to perform the verification)	• Result Code = 01 (Matched successfully (or true)) • Reason Code = 01 (Data available and verified successfully) **Verification Response Object 2** • Attribute Name = 05 (Core attribute) • Core Attribute Field = `cardholderName` (Field in the Authentication Request message the verifying system will use to perform the verification) • Verification Response Source = 01 (ACS) • Result Code = 01 (Matched successfully (or true))

6.6 Travel Industry Message Extension

Using the EMV® 3-D Secure Travel Industry Message Extension, the 3DS Server provides additional information about a travel-related transaction (air travel, car rental and hotel). The ACS uses these additional data for its risk-decisioning.

Refer to the EMV 3-D Secure Travel Industry Message Extension for further details.

6.7 Payment Token Message Extension

The EMV® 3-D Secure Payment Token Message Extension enables 3DS components to provide or receive token-related information after a Payment Token has been detokenised.

This message extension is applicable only for Message Version Number 2.2.0.

For Message Version Number 2.3.0 and above, the information is conveyed using the Payment Token Information data element.

Refer to the EMV 3-D Secure Payment Token Message Extension for further details.

7 Split-SDK

7.1 Business Overview

The Split-SDK presents an alternative architecture approach to the Default-SDK. While it functions like the 3DS Default-SDK, as defined in the EMV® 3-D Secure—SDK Specification, what sets the Split-SDK apart is the division of the functionalities between a client-side component (Split-SDK Client) and a server-side component (Split-SDK Server).

The Split-SDK vendor has the choice of the split of functions and interfaces between the server side and client side. This approach not only facilitates development, but also minimises the need for frequent client-side updates, which makes it an ideal choice for Merchants with large-scale mobile application deployment.

Compared to the Default-SDK, the Split-SDK approach streamlines the development and maintenance efforts. For most SDK updates (i.e., bug fixing), the 3DS Requestor should be able to make the changes on the Split-SDK Server, thus reducing the need to push an application update to all Consumer Devices.

The Split-SDK architecture is declined in several variants to facilitate its integration and use in a wide range of e-commerce channels and devices, including IoT devices such as smart speakers. This adaptability ensures a consistent user experience across different platforms and devices, catering to evolving consumer preferences. Businesses can leverage this flexibility to reach new markets and engage with customers in innovative ways.

The only constraint in the split of the SDK functions between the server and the client is that the encryption of Cardholder inputs during a challenge must be performed by the Split-SDK Client. If the Split-SDK Client cannot securely encrypt the Cardholder inputs, the Split-SDK is flagged as Limited. For a Limited Split-SDK, the range of allowed Authentication Methods is limited to those that are dynamic (static Authentication Methods such as password are not supported).

By securing connections with the ACSs and encrypting sensitive Cardholder data, the Split-SDK ensures that transactions are conducted in a secure environment. From an ACS point of view, there are no functional differences between a Default-SDK and a Split-SDK, as message exchanges and challenge screen rendering are identical.

However, implementing the Split-SDK does come with some challenges, particularly involving Device Information. The Merchant and Split-SDK servers must provide accurate and reliable Device Information (Platform Provider-specific Parameters), in particular, Device ID and User ID, to maintain the efficiency of the ACS authentication process.

Benefits by Actor

- Merchant – having most of the updates and/or bug fixing centralised on the Split-SDK Server facilitates the operation as there is no need to update all the Split-SDK clients (and 3DS Requestor App) compared to the Default-SDK.
- Issuer – the ACS directly receives an identifier for the Cardholder and for the Cardholder's device, which facilitates risk analysis.

7.2 Technical Features

Preconditions

The 3DS Requestor is able to identify the Cardholder and his Device, in order to operate a Split-SDK and provide reliable Device information.

The ACS may support the Device acknowledgement message extension to recognise that the authentication is initiated from a Split-SDK for a 3DS version 2.2 authentication.

3DS Data Elements Related to the Split-SDK

The data elements listed in Table 7.1 below are provided by 3DS Servers for authentications initiated from 3DS Requestor Apps based on a Split-SDK.

For additional information, refer to Table A.1 in the Core Specification and to the EMV 3-D Secure Device Acknowledgement Message Extension.

Table 7.1: 3DS Data Elements Related to the Split-SDK

Data Element	Description	Version
Split-SDK Type	Indicates the characteristics of a Split-SDK. Split-SDK Variant: Implementation characteristics of the Split-SDK client. Limited Split-SDK Indicator: If the Split-SDK client has limited capabilities.	2.3.1 2.2 + Device Acknowledgement
Split-SDK Server ID	DS assigned Split-SDK Server identifier. Each DS can provide a unique ID to each Split-SDK Server on an individual basis.	2.3.1 2.2 + Device Acknowledgement
SDK Type	Indicates the type of 3DS SDK. This data element provides additional information to the DS and ACS to determine the best approach for handling the transaction.	2.3.1 2.2 + Device Acknowledgement
SDK Signature Timestamp	Date and time indicating when the 3DS SDK generated the Split-SDK Server Signed Content converted into UTC.	2.3.1 2.2 + Device Acknowledgement
SDK Server Signed Content	Contains the JWS object (represented as a string) created by the Split-SDK Server for the AReq message. The body of the JWS object (represented as a string) will contain the following data elements: • SDK Reference Number • SDK Signature Timestamp • SDK Transaction ID	2.3.1 2.2 + Device Acknowledgement

Data Element	Description	Version
	• Split-SDK Server ID	
Device Information	Device information gathered by the 3DS SDK from a Consumer Device. This is JSON name/value pairs that as a whole is Base64url-encoded. This will be populated by the DS as unencrypted data to the ACS obtained from SDK Encrypted Data.	2.3.1 2.2
SDK Encrypted Data	JWE Object (represented as a string) as defined in Section 6.2.2.1 of the 3DS Specification containing data encrypted by the 3DS SDK for the DS to decrypt.	2.3.1 2.2

7.2.1 Default-SDK and Split-SDK Flow

The transaction flow is identical for a Default-SDK and a Split-SDK. The same messages (CReq/CRes) are exchanged with the ACS during a challenge. For additional information, refer to Section 3.1 – App-based Requirements in the Core Specification.

The Split-SDK may be implemented in 3 variants: Native, Browser and Shell.

7.2.2 Split-SDK Native

The Split-SDK Client functionality is implemented using native platform code of the Consumer Device and is embedded within a 3DS Requestor App (similarly to the Default-SDK).

7.2.3 Split-SDK Browser

The Split-SDK Client functionality is implemented using JavaScript running in a device Browser. The JavaScript is delivered from the Split-SDK Server to the 3-D Secure challenge window opened on the Browser during the authentication.

7.2.4 Split-SDK Shell

The Split-SDK Client functionality is implemented using JavaScript running in a secured WebView opened by the Split-SDK/Shell. The Split-SDK/Shell is a thin client embedded in the 3DS Requestor App (similarly to the Default-SDK). The JavaScript is delivered from the Split-SDK Server during the authentication.

7.2.5 Limited SDK

If the Client cannot securely encrypt the CReq message, then the Split-SDK is considered Limited, as defined in Section 3.3 of the EMV® 3-D Secure Split-SDK Specification. For a

Limited Split-SDK, the range of allowed Authentication Methods is limited to those that are dynamic (static Authentication Methods like password are not supported).

The Limited option for the Split-SDK or Default-SDK applies to devices that are not capable of supporting cryptographic functions such as key generation and encryption of CReq messages.

The Split-SDK Limited option is only applicable to the Native Client.

For the Browser and Shell Client Split-SDK variants, the Client is coded as a JavaScript that executes in a Browser iframe or a WebView. These environments can support the 3DS cryptography functions of the 3DS SDK. Therefore, the Limited option is not applicable.

For additional information, refer to the EMV 3-D Secure Split-SDK Specification.

8 3-D Secure Documentation

The 3DS documentation consists of several key documents that define the three-domain model and overall architecture and provide the technical requirements, as well as a range of supporting documents to help the understanding and implementation of the 3DS protocol.

There are four key types of 3DS documentation:

- Specifications – define the 3DS architecture and requirements
- Specification bulletins – update or complement the specifications
- Message extensions – enable the transport of additional data
- Supporting documents such as guides, white papers or FAQ – provide additional information and guidance.

The 3DS specification suite consists of five key documents listed below.

- EMV 3-D Secure Protocol and Core Functions Specification (the main 3-D Secure specification; Core Specification) – defines the three-domain model, the messages, their data, and the channels for performing a 3DS authentication.
- EMV 3-D Secure SDK Specification – defines the device-side component of 3DS. 3DS Requestors such as Merchants integrate this SDK with their mobile device app and make the app available to end users.
- EMV 3-D Secure Split-SDK Specification – defines a variant of the 3DS SDK for which some of the client functionalities do not run on the device, but on a server component, thus implementing a 3DS SDK with functionalities split between a Split-SDK Client (client side) and a Split-SDK Server (server side).
- EMV 3-D Secure SDK Device Information – defines the device information provided by the 3DS SDK when an authentication is initiated from an app.
- EMV 3-D Secure Specification Bulletin No. 255 Specification Version Configuration – defines the status of the Core Specification, the EMV 3-D Secure SDK—Device Information versions, and the EMV 3-D Secure message extensions.

  The SDK Specification and the Split-SDK Specification are only applicable to mobile device applications or the Split-SDK architecture.

There are currently five EMV 3-D Secure message extensions:

- EMV 3-D Secure Attribute Verification Message Extension – defines a structure for components to request verification and respond to requests for verification of pre-defined attributes or *Core Specification* data elements.
- EMV 3-D Secure Bridging Message Extension – defines how existing 3DS v2.1.0 and v2.2.0 components can provide or consume additional data related to Core Specification v2.3.1.
- EMV 3-D Secure Device Acknowledgement Message Extension – defines how the 3DS Server can provide the Split-SDK-related data to the ACS and the ACS can acknowledge data received in EMV 3-D Secure Device Information.
- EMV 3-D Secure Payment Token Message Extension – defines how 3DS components can provide or receive token-related information.

- EMV 3-D Secure Travel Industry Message Extension – defines how 3DS Servers can provide travel-related data to the ACS.

8.1 3-D Secure Specification v2.2.0

The following documents are applicable to 3DS Specification v2.2.0:

- EMV 3-D Secure Protocol and Core Functions Specification v2.2.0, as amended by EMV 3-D Secure Specification Bulletin No. 214 v3 (June 2023)
- EMV 3-D Secure SDK Specification v2.2.0, as amended by EMV® 3-D Secure Specification Bulletin No. 211 – EMV® 3-D Secure SDK Key Features for version 2.2.0 (December 2018)
- EMV 3-D Secure Specification Bulletin No. 255 v5 – Specification Version Configuration (February 2025)

For this version, the updates to the specifications are contained in the applicable specification bulletins.

8.2 3-D Secure Specification v2.3.1

The following documents are applicable to 3DS Specification v2.3.1:

- EMV 3-D Secure Protocol and Core Functions Specification v2.3.1.1
  - EMV 3-D Secure Specification Bulletin No. 294 – updates, clarifications and errata incorporated since version 2.3.1.0 (May 2023)
  - EMV 3-D Secure Specification Bulletin No. 279 – updates, clarifications and errata incorporated since version 2.2.0 as amended by EMV® 3-D Secure Specification Bulletin No. 214 v3 (August 2023)
- EMV 3-D Secure SDK Specification v2.3.1.1
  - EMV 3-D Secure Specification Bulletin No. 296 – updates, clarifications and errata incorporated since version 2.3.1.0 (May 2023)
  - EMV 3-D Secure Specification Bulletin No. 280 – updates, clarifications and errata incorporated since version 2.2.0 (August 2023)
- EMV 3-D Secure Split-SDK Specification v2.3.1.0
  - EMV 3-D Secure Specification Bulletin No. 271 Split-SDK Specification for version 2.3.1.0 (August 2022)
- EMV 3-D Secure Specification Bulletin No. 255 v5 – Specification Version Configuration (February 2025)

For this version, the specifications include all the latest revisions, and specification bulletins are provided (where applicable) solely as resources reflecting the changes.

8.3 3-D Secure SDK — Device Information

The EMV 3-D Secure SDK – Device Information is applicable to the ACS for all specification versions (refer to EMV 3-D Secure Specification Bulletin 255).

- Data Version 1.0:
  - EMV 3-D Secure SDK – Device Information Version 2.0.0
  - EMV 3-D Secure Specification Bulletin No. 205 v1 (August 2018)
- Data Version 1.1:
  - EMV 3-D Secure SDK – Device Information Version 2.1.0
  - EMV 3-D Secure Specification Bulletin No. 213 v1 (May 2019)
- Data Version 1.3: EMV 3-D Secure Specification Bulletin No. 222 v1 (August 2019)
- Data Version 1.4: EMV 3-D Secure Specification Bulletin No. 223 v1 (October 2019)
- Data Version 1.5
  - EMV 3-D Secure SDK – Device Information Data Version 1.5
  - EMV 3-D Secure Specification Bulletin No. 225 – SDK Device Information Data Version 1.5 Updates, clarifications, and errata (September 2021)
- Data Version 1.6
  - EMV 3-D Secure SDK – Device Information Data Version 1.6
  - EMV 3-D Secure Specification Bulletin No. 285 – SDK Device Information Data Version 1.6 (May 2023)
- Data Version 1.7
  - EMV 3-D Secure SDK – Device Information Data Version 1.7
  - EMV 3-D Secure Specification Bulletin No. 309 – SDK Device Information Data Version 1.7 (February 2025)

All the specifications and specification bulletins listed in this section are available on the EMVCo website.

8.4 Other Supporting Documentation

- EMV 3-D Secure Specifications Frequently Asked Questions – Technical Questions (July 2024)
- EMV 3-D Secure Frequently Asked Questions – General Questions (December 2020)
- EMV 3-D Secure SDK Technical Guide Version 2.1.0 (October 2017)
- EMV 3-D Secure App-based Cryptographic Worked Samples Version 3.0.0 (October 2019)
- EMV 3-D Secure JSON Message Samples Version 2.1.0 (April 2018)
- EMV 3-D Secure White Paper Version 2.0 – Use of FIDO® Data in 3-D Secure Messages to Support Issuer Validation of FIDO® Authentication Data (November 2023)
- EMV 3-D Secure Browser Flow Best Practices (September 2021)
- EMV 3-D Secure and PSD2 Requirements for Strong Customer Authentication (December 2020)
- EMV 3-D Secure UI/UX Design Guidelines (June 2024)

- EMV General Bulletin No. 50 – New EMV 3-D Secure UI Design Guidelines (August 2021)

All the documents listed in this section are available on the EMVCo website.